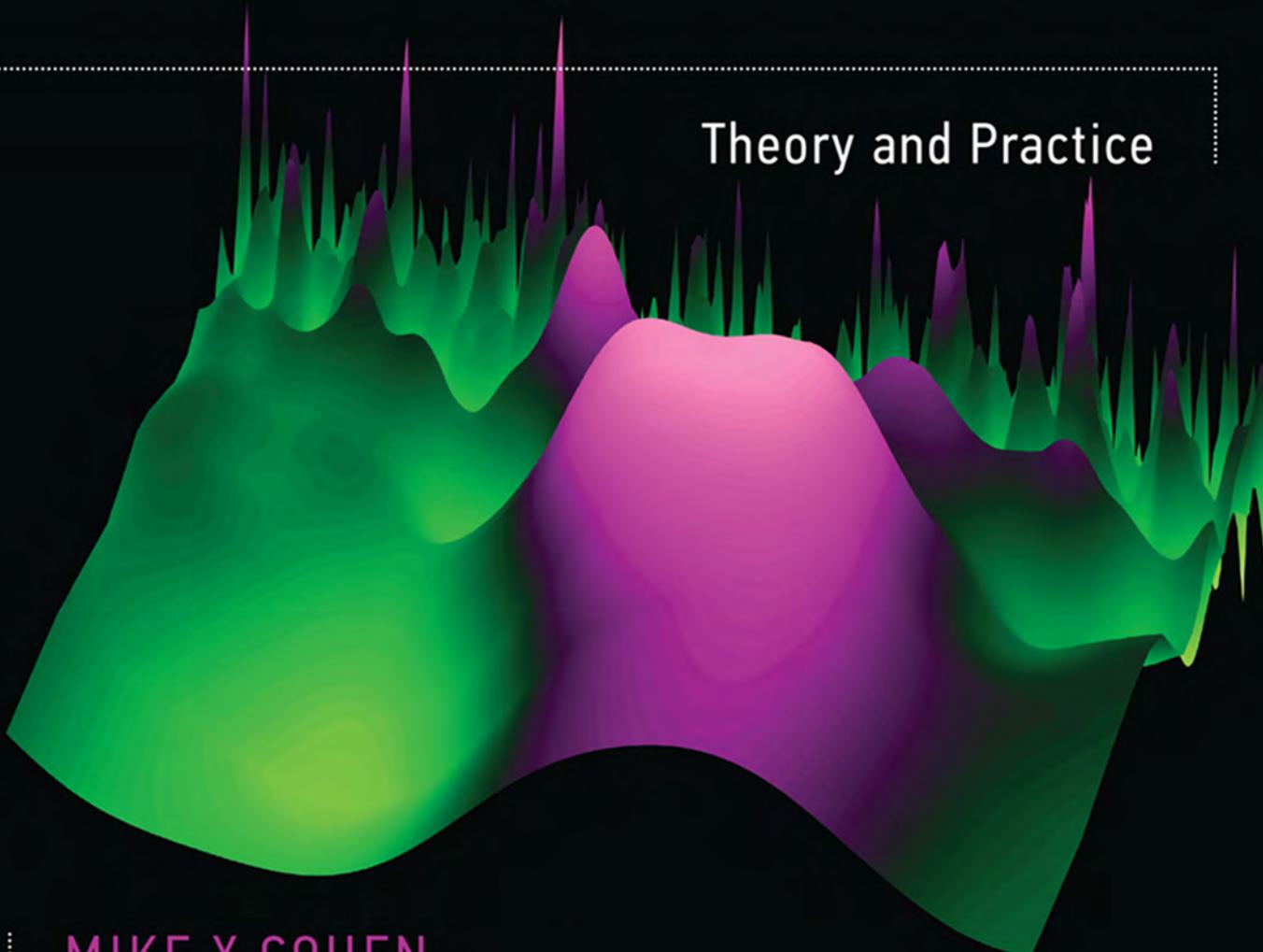


ANALYZING NEURAL TIME SERIES DATA

A 3D surface plot representing neural time series data. The surface has two main peaks, one green and one pink, set against a dark background. The green peak is on the left, and the pink peak is on the right. Both peaks have smaller, sharper peaks (spikes) extending upwards from their surfaces.

Theory and Practice

MIKE X COHEN

Analyzing Neural Time Series Data

Issues in Clinical and Cognitive Neuropsychology

Jordan Grafman, series editor

Patient-Based Approaches to Cognitive Neuroscience

Martha J. Farah and Todd E. Feinberg, editors

Gateway to Memory: An Introduction to Neural Network Modeling of the Hippocampus and Learning

Mark A. Gluck and Catherine E. Myers

Neurological Foundations of Cognitive Neuroscience

Mark D'Esposito, editor

The Parallel Brain: The Cognitive Neuroscience of the Corpus Callosum

Eran Zeidel and Marco Iacoboni, editors

Fatigue as a Window to the Brain

John DeLuca, editor

Patient-Based Approaches to Cognitive Neuroscience, Second Edition

Martha J. Farah and Todd E. Feinberg, editors

Neurogenetic Developmental Disorders: Variation of Manifestation in Childhood

Michèle M. Mazzocco and Judith L. Ross, editors

The Genetics of Cognitive Neuroscience

Terry E. Goldberg and Daniel R. Weinberger, editors

Analyzing Neural Time Series Data: Theory and Practice

Mike X Cohen

Analyzing Neural Time Series Data

Theory and Practice

Mike X Cohen

The MIT Press
Cambridge, Massachusetts
London, England

© 2014 Massachusetts Institute of Technology

All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

MIT Press books may be purchased at special quantity discounts for business or sales promotional use. For information, please email special_sales@mitpress.mit.edu or write to Special Sales Department.

This book was set in ITC Stone Serif Std by Toppan Best-set Premedia Limited, Hong Kong. Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Cohen, Mike X, 1979—

Analyzing neural time series data : theory and practice / Mike X Cohen.

pages cm.—(Issues in clinical and cognitive neuropsychology)

Includes bibliographical references and index.

ISBN 978-0-262-01987-3 (hardcover : alk. paper) 1. Neural networks (Neurobiology) 2. Neural networks (Computer science) 3. Computational neuroscience. 4. Artificial intelligence—Biological applications. I. Title.

QP363.3.C633 2014

612.8'2339—dc23

2013016461

10 9 8 7 6 5 4 3 2 1

Contents

Preface xvii

Acknowledgments xix

Part I: Introduction 1

1 The Purpose of This Book, Who Should Read It, and How to Use It 3

- 1.1 What Is Cognitive Electrophysiology? 3
- 1.2 What Is the Purpose of This Book? 4
- 1.3 Why Shouldn't You Use <Insert Name of M/EEG Software Analysis Package>? 6
- 1.4 Why Program Analyses, and Why in Matlab? 7
- 1.5 How Best to Learn from and Use This Book 8
- 1.6 Sample Data and Online Code 9
- 1.7 Terminology Used in This Book 10
- 1.8 Exercises 11
- 1.9 Is Everything There Is to Know about EEG Analyses in This Book? 11
- 1.10 Who Should Read This Book? 12
- 1.11 Is This Book Difficult? 13
- 1.12 Questions? 14

2 Advantages and Limitations of Time- and Time-Frequency-Domain Analyses 15

- 2.1 Why EEG? 15
- 2.2 Why Not EEG? 16
- 2.3 Interpreting Voltage Values from the EEG Signal 17
- 2.4 Advantages of Event-Related Potentials 18
- 2.5 Limitations of ERPs 19
- 2.6 Advantages of Time-Frequency-Based Approaches 21
- 2.7 Limitations of Time-Frequency-Based Approaches 23
- 2.8 Temporal Resolution, Precision, and Accuracy of EEG 24
- 2.9 Spatial Resolution, Precision, and Accuracy of EEG 26

2.10 Topographical Localization versus Brain Localization	27
2.11 EEG or MEG?	28
2.12 Costs of EEG Research	29

3 Interpreting and Asking Questions about Time-Frequency Results 31

3.1 EEG Time-Frequency: The Basics	31
3.2 Ways to View Time-Frequency Results	35
3.3 tfviewervx and erpviewervx	37
3.4 How to View and Interpret Time-Frequency Results	38
3.5 Things to Be Suspicious of When Viewing Time-Frequency Results	40
3.6 Do Results in Time-Frequency Plots Mean That There Were Neural Oscillations?	41

4 Introduction to Matlab Programming 43

4.1 Write Clean and Efficient Code	43
4.2 Use Meaningful File and Variable Names	45
4.3 Make Regular Backups of Your Code and Keep Original Copies of Modified Code	45
4.4 Initialize Variables	46
4.5 Help!	47
4.6 Be Patient and Embrace the Learning Experience	49
4.7 Exercises	50

5 Introduction to the Physiological Bases of EEG 51

5.1 Biophysical Events That Are Measurable with EEG	51
5.2 Neurobiological Mechanisms of Oscillations	54
5.3 Phase-Locked, Time-Locked, Task-Related	55
5.4 Neurophysiological Mechanisms of ERPs	56
5.5 Are Electrical Fields Causally Involved in Cognition?	57
5.6 What if Electrical Fields Are Not Causally Involved in Cognition?	59

6 Practicalities of EEG Measurement and Experiment Design 61

6.1 Designing Experiments: Discuss, Pilot, Discuss, Pilot	61
6.2 Event Markers	62
6.3 Intra- and Intertrial Timing	63
6.4 How Many Trials You Will Need	65
6.5 How Many Electrodes You Will Need	65
6.6 Which Sampling Rate to Use When Recording Data	66
6.7 Other Optional Equipment to Consider	68

Part II: Preprocessing and Time-Domain Analyses 71

7 Preprocessing Steps Necessary and Useful for Advanced Data Analysis 73

7.1 What Is Preprocessing?	73
7.2 The Balance between Signal and Noise	73

7.3 Creating Epochs	75
7.4 Matching Trial Count across Conditions	78
7.5 Filtering	80
7.6 Trial Rejection	80
7.7 Spatial Filtering	81
7.8 Referencing	82
7.9 Interpolating Bad Electrodes	83
7.10 Start with Clean Data	85

8 EEG Artifacts: Their Detection, Influence, and Removal 87

8.1 Removing Data Based on Independent Components Analysis	87
8.2 Removing Trials because of Blinks	89
8.3 Removing Trials because of Oculomotor Activity	90
8.4 Removing Trials Based on EMG in EEG Channels	92
8.5 Removing Trials Based on Task Performance	93
8.6 Removing Trials Based on Response Hand EMG	94
8.7 Train Subjects to Minimize Artifacts	95
8.8 Minimize Artifacts during Data Collection	96

9 Overview of Time-Domain EEG Analyses 97

9.1 Event-Related Potentials	97
9.2 Filtering ERPs	97
9.3 Butterfly Plots and Global Field Power/Topographical Variance Plots	100
9.4 The Flicker Effect	101
9.5 Topographical Maps	102
9.6 Microstates	104
9.7 ERP Images	105
9.8 Exercises	106

Part III: Frequency and Time-Frequency Domains Analyses 109

10 The Dot Product and Convolution 111

10.1 Dot Product	111
10.2 Convolution	113
10.3 How Does Convolution Work?	113
10.4 Convolution versus Cross-Covariance	118
10.5 The Purpose of Convolution for EEG Data Analyses	118
10.6 Exercises	119

11 The Discrete Time Fourier Transform, the FFT, and the Convolution Theorem 121

11.1 Making Waves	121
11.2 Finding Waves in EEG Data with the Fourier Transform	123

11.3 The Discrete Time Fourier Transform	124
11.4 Visualizing the Results of a Fourier Transform	126
11.5 Complex Results and Negative Frequencies	127
11.6 Inverse Fourier Transform	129
11.7 The Fast Fourier Transform	130
11.8 Stationarity and the Fourier Transform	132
11.9 Extracting More or Fewer Frequencies than Data Points	134
11.10 The Convolution Theorem	135
11.11 Tips for Performing FFT-Based Convolution in Matlab	138
11.12 Exercises	139

12 Morlet Wavelets and Wavelet Convolution 141

12.1 Why Wavelets?	141
12.2 How to Make Wavelets	144
12.3 Wavelet Convolution as a Bandpass Filter	146
12.4 Limitations of Wavelet Convolution as Discussed Thus Far	147
12.5 Exercises	150

13 Complex Morlet Wavelets and Extracting Power and Phase 151

13.1 The Wavelet Complex	151
13.2 Imagining the Imaginary	153
13.3 Rectangular and Polar Notation and the Complex Plane	153
13.4 Euler's Formula	155
13.5 Euler's Formula and the Result of Complex Wavelet Convolution	158
13.6 From Time Point to Time Series	162
13.7 Parameters of Wavelets and Recommended Settings	162
13.8 Determining the Frequency Smoothing of Wavelets	172
13.9 Tips for Writing Efficient Convolution Code in Matlab	173
13.10 Describing This Analysis in Your Methods Section	173
13.11 Exercises	173

14 Bandpass Filtering and the Hilbert Transform 175

14.1 Hilbert Transform	175
14.2 Filtering Data before Applying the Hilbert Transform	178
14.3 Finite versus Infinite Impulse Response Filters	179
14.4 Bandpass, Band-Stop, High-Pass, Low-Pass	179
14.5 Constructing a Filter	180
14.6 Check Your Filters	186
14.7 Applying the Filter to Data	188
14.8 Butterworth (IIR) Filter	189
14.9 Filtering Each Trial versus Filtering Concatenated Trials	190
14.10 Multiple Frequencies	190

14.11 A World of Filters	192
14.12 Describing This Analysis in Your Methods Section	193
14.13 Exercises	193

15 Short-Time FFT 195

15.1 How the Short-Time FFT Works	195
15.2 Taper the Time Series	195
15.3 Time Segment Lengths and Overlap	199
15.4 Power and Phase	200
15.5 Describing This Analysis in Your Methods Section	201
15.6 Exercises	201

16 Multitapers 203

16.1 How the Multitaper Method Works	203
16.2 The Tapers	205
16.3 When You Should and Should Not Use Multitapers	207
16.4 The Multitaper Framework and Advanced Topics	208
16.5 Describing This Analysis in Your Methods Section	208
16.6 Exercises	208

17 Less Commonly Used Time-Frequency Decomposition Methods 211

17.1 Autoregressive Modeling	211
17.2 Hilbert-Huang (Empirical Mode Decomposition)	211
17.3 Matching Pursuit	212
17.4 P-Episode	213
17.5 S-Transform	214

18 Time-Frequency Power and Baseline Normalizations 217

18.1 1/f Power Scaling	217
18.2 The Solution to 1/f Power in Task Designs	220
18.3 Decibel Conversion	220
18.4 Percentage Change and Baseline Division	223
18.5 Z-Transform	224
18.6 Not All Transforms Are Equal	224
18.7 Other Transforms	227
18.8 Mean versus Median	227
18.9 Single-Trial Baseline Normalization	230
18.10 The Choice of Baseline Time Window	230
18.11 Disadvantages of Baseline-Normalized Power	234
18.12 Signal-to-Noise Estimates	234
18.13 Number of Trials and Power Estimates	237
18.14 Downsampling Results after Analyses	239

18.15 Describing This Analysis in Your Methods Section	239
18.16 Exercises	239

19 Intertrial Phase Clustering 241

19.1 Why Phase Values Cannot Be Averaged	241
19.2 Intertrial Phase Clustering	242
19.3 Strength in Numbers	246
19.4 Using ITPC When There Are Few Trials or Condition Differences in Trial Count	248
19.5 Effects of Temporal Jitter on ITPC and Power	249
19.6 ITPC and Power	251
19.7 Weighted ITPC	253
19.8 Multimodal Phase Distributions	257
19.9 Spike-Field Coherence	257
19.10 Describing This Analysis in Your Methods Section	257
19.11 Exercises	258

20 Differences among Total, Phase-Locked, and Non-Phase-Locked Power and Intertrial Phase Consistency 259

20.1 Total Power	259
20.2 Non-Phase-Locked Power	259
20.3 Phase-Locked Power	260
20.4 ERP Time-Frequency Power	262
20.5 Intertrial Phase Clustering	262
20.6 When to Use What Approach	263
20.7 Exercise	264

21 Interpretations and Limitations of Time-Frequency Power and ITPC Analyses 265

21.1 Terminology	265
21.2 When to Use What Time-Frequency Decomposition Method	266
21.3 Interpreting Time-Frequency Power	267
21.4 Interpreting Time-Frequency Intertrial Phase Clustering	268
21.5 Limitations of Time-Frequency Power and Intertrial Phase Clustering	269
21.6 Do Time-Frequency Analyses Reveal Neural Oscillations?	270

Part IV: Spatial Filters 273

22 Surface Laplacian 275

22.1 What Is the Surface Laplacian?	275
22.2 Algorithms for Computing the Surface Laplacian for EEG Data	278
22.3 Surface Laplacian for Topographical Localization	283
22.4 Surface Laplacian for Connectivity Analyses	286

22.5 Surface Laplacian for Cleaning Topographical Noise	288
22.6 Describing This Analysis in Your Methods Section	289
22.7 Exercises	289

23 Principal Components Analysis 291

23.1 Purpose and Interpretations of Principal Components Analysis	291
23.2 How PCA Is Computed	293
23.3 Distinguishing Significant from Nonsignificant Components	297
23.4 Rotating PCA Solutions	300
23.5 Time-Resolved PCA	300
23.6 PCA with Time-Frequency Information	301
23.7 PCA across Conditions	303
23.8 Independent Components Analysis	304
23.9 Describing This Method in Your Methods Section	305
23.10 Exercises	305

24 Basics of Single-Dipole and Distributed-Source Imaging 307

24.1 The Forward Solution	307
24.2 The Inverse Problem	310
24.3 Dipole Fitting	311
24.4 Nonadaptive Distributed-Source Imaging Methods	312
24.5 Adaptive Distributed-Source Imaging	313
24.6 Theoretical and Practical Limits of Spatial Precision and Resolution	314

Part V: Connectivity 317

25 Introduction to the Various Connectivity Analyses 319

25.1 Why Only Two Sites (Bivariate Connectivity)?	319
25.2 Important Concepts Related to Bivariate Connectivity	320
25.3 Which Measure of Connectivity Should Be Used?	322
25.4 Phase-Based Connectivity	323
25.5 Power-Based Connectivity	324
25.6 Granger Prediction	324
25.7 Mutual Information	325
25.8 Cross-Frequency Coupling	325
25.9 Graph Theory	326
25.10 Potential Confound of Volume Conduction	326

26 Phase-Based Connectivity 333

26.1 Terminology	333
26.2 ISPC over Time	333

26.3 ISPC-Trials	337
26.4 ISPC and the Number of Trials	340
26.5 Relation between ISPC and Power	341
26.6 Weighted ISPC-Trials	342
26.7 Spectral Coherence (Magnitude-Squared Coherence)	342
26.8 Phase Lag-Based Measures	346
26.9 Which Measure of Phase Connectivity Should You Use?	351
26.10 Testing the Mean Phase Angle	352
26.11 Describing These Analyses in Your Methods Section	355
26.12 Exercises	355

27 Power-Based Connectivity 357

27.1 Spearman versus Pearson Coefficient for Power Correlations	357
27.2 Power Correlations over Time	361
27.3 Power Correlations over Trials	363
27.4 Partial Correlations	366
27.5 Matlab Programming Tips	367
27.6 Describing This Analysis in Your Methods Section	370
27.7 Exercises	370

28 Granger Prediction 371

28.1 Univariate Autoregression	372
28.2 Bivariate Autoregression	373
28.3 Autoregression Errors and Error Variances	375
28.4 Granger Prediction over Time	376
28.5 Model Order	377
28.6 Frequency Domain Granger Prediction	380
28.7 Time Series Covariance Stationarity	383
28.8 Baseline Normalization of Granger Prediction Results	384
28.9 Statistics	385
28.10 Additional Applications of Granger Prediction	387
28.11 Exercises	388

29 Mutual Information 389

29.1 Entropy	389
29.2 How Many Histogram Bins to Use	391
29.3 Enjoy the Entropy	395
29.4 Joint Entropy	396
29.5 Mutual Information	396
29.6 Mutual Information and Amount of Data	398
29.7 Mutual Information with Noisy Data	400
29.8 Mutual Information over Time or over Trials	402

29.9 Mutual Information on Real Data	402
29.10 Mutual Information on Frequency-Band-Specific Data	402
29.11 Lagged Mutual Information	404
29.12 Statistics	405
29.13 More Information	406
29.14 Describing This Analysis in Your Methods Section	407
29.15 Exercises	407

30 Cross-Frequency Coupling 409

30.1 Visual Inspection of Cross-Frequency Coupling	409
30.2 Power-Power Correlations	410
30.3 A Priori Phase-Amplitude Coupling	410
30.4 Separating Task-Related Phase and Power Coactivations from Phase-Amplitude Coupling	418
30.5 Mixed A Priori/Exploratory Phase-Amplitude Coupling	419
30.6 Exploratory Phase-Amplitude Coupling	421
30.7 Notes about Phase-Amplitude Coupling	422
30.8 Phase-Phase Coupling	425
30.9 Other Methods for Quantifying Cross-Frequency Coupling	426
30.10 Cross-Frequency Coupling over Time or over Trials	426
30.11 Describing This Analysis in Your Methods Section	426
30.12 Exercises	427

31 Graph Theory 429

31.1 Networks as Matrices and Graphs	429
31.2 Thresholding Connectivity Matrices	432
31.3 Connectivity Degree	433
31.3 Clustering Coefficient	436
31.4 Path Length	440
31.5 Small-World Networks	441
31.6 Statistics	444
31.7 How to Describe These Analyses in Your Paper	446
31.8 Exercises	446

Part VI: Statistical Analyses 447

32 Advantages and Limitations of Different Statistical Procedures 449

32.1 Are Statistics Necessary?	449
32.2 At What Level Should Statistics Be Performed?	450
32.3 What <i>p</i> -Value Should Be Used, and Should Multiple-Comparisons Corrections Be Applied?	452
32.4 Are <i>p</i> -Values the Only Statistical Metric?	453
32.5 Statistical Significance versus Practical Significance	454
32.6 Type I and Type II Errors	455

32.7 What Kinds of Statistics Should Be Applied?	456
32.8 How to Combine Data across Subjects	457

33 Nonparametric Permutation Testing 459

33.1 Advantages of Nonparametric Permutation Testing	459
33.2 Creating a Null-Hypothesis Distribution	461
33.3 How Many Iterations Are Necessary for the Null-Hypothesis Distribution?	462
33.4 Determining Statistical Significance	463
33.5 Multiple Comparisons and Their Corrections	468
33.6 Correction for Multiple Comparisons Using Pixel-Based Statistics	470
33.7 Corrections for Multiple Comparisons Using Cluster-Based Statistics	471
33.8 False Discovery Rate for Multiple-Comparisons Correction	475
33.9 What Should Be Permuted?	475
33.10 Nonparametric Permutation Testing beyond Simple Bivariate Cases	477
33.11 Describing This Analysis in Your Methods Section	477

34 Within-Subject Statistical Analyses 479

34.1 Changes in Task-Related Power Compared to Baseline	479
34.2 Discrete Condition Differences in Power	481
34.3 Continuous Relationship with Power: Single-Trial Correlations	482
34.4 Continuous Relationships with Power: Single-Trial Multiple Regression	485
34.5 Determining Statistical Significance of Phase-Based Data	487
34.6 Testing Preferred Phase Angle across Conditions	490
34.7 Testing the Statistical Significance of Correlation Coefficients	490

35 Group-Level Analyses 493

35.1 Avoid Circular Inferences	493
35.2 Group-Level Analysis Strategy 1: Test Each Pixel and Apply a Mapwise Threshold	494
35.3 Group-Level Analysis Strategy 2a: Time-Frequency Windows for Hypothesis-Driven Analyses	495
35.4 Group-Level Analysis Strategy 2b: Subject-Specific Time-Frequency Windows for Hypothesis-Driven Analyses	499
35.5 Determining How Many Subjects You Need for Group-Level Analyses	501

36 Recommendations for Reporting Results in Figures, Tables, and Text 503

36.1 Recommendation 1: One Figure, One Idea	504
36.2 Recommendation 2: Show Data	504
36.3 Recommendation 3: Highlight Significant Effects Instead of Removing Nonsignificant Effects	506
36.4 Recommendation 4: Show Specificity (or Lack Thereof) in Frequency, Time, and Space	508
36.5 Recommendation 5: Use Color	509
36.6 Recommendation 6: Use Informative Figure Labels and Captions	511
36.7 Recommendation 7: Avoid Showing “Representative” Data	512
36.8 A Checklist for Making Figures	514

36.9 Tables	514
36.10 Reporting Results in the Results Section	515

Part VII: Conclusions and Future Directions 519

37 Recurring Themes in This Book and Some Personal Advice 521

37.1 Theme: Myriad Possible Analyses	521
37.2 Advice: Avoid the Paralysis of Analysis	522
37.3 Theme: You Don't Have to Program Your Own Analyses, but You Should Know How Analyses Work	522
37.4 Advice: If It Feels Wrong, It Probably Is	523
37.5 Advice: When in Doubt, Plot It Out	523
37.6 Advice: Know These Three Formulas like the Back of Your Hand	524
37.7 Theme: Connectivity over Trials or over Time	524
37.8 Theme: Most Analysis Parameters Introduce Bias	525
37.9 Theme: Write a Clear Methods Section so Others Can Replicate Your Analyses	525
37.10 Theme: Use Descriptive and Appropriate Analysis Terms	526
37.11 Advice: Interpret Null Results Cautiously	527
37.12 Advice: Try Simulations but Also Trust Real Data	528
37.13 Advice: Trust Replications	529
37.14 Theme: Analyses Are Not Right or Wrong; They Are Appropriate or Inappropriate	529
37.15 Advice: Hypothesis Testing Is Good/Bad, and So Is Data-Driven Exploration	530
37.16 Advice: Find Something That Drives You and Study It	531
37.17 Cognitive Electrophysiology: The Art of Finding Anthills on Mountains	532

38 The Future of Cognitive Electrophysiology 535

38.1 Developments in Analysis Methods	535
38.2 Developments in Understanding the Neurophysiology of EEG	537
38.3 Developments in Experiment Design	538
38.4 Developments in Measurement Technology	539
38.5 The Role of the Body in Brain Function	540
38.6 Determining Causality	541
38.7 Inferring Cognitive States from EEG Signatures: Inverse Inference	542
38.8 Tables of Activation	543
38.9 Disease Diagnosis and Predicting Treatment Course and Success	543
38.10 Clinical Relevance Is Not Necessary for the Advancement of Science	544
38.11 Replications	545
38.12 Double-Blind Review for Scientific Publications	547
38.13 ?	548
References	549
Index	573

Preface

I began learning about time-frequency decomposition of EEG data in January of 2007, after a memorable end to a New Year's party led to a particular 30-day resolution. Before then, I knew the basics of interpreting time-frequency power plots from reading publications and online tutorials, but it was mostly magic to me; I had no idea *how* those red and blue blobs came to be and what they really meant. With the extra time on my hands, I decided to learn how to create a time-frequency power plot. I wanted to understand how the data went from wiggly lines to colorful two-dimensional (2-D) plots. It was hard in the beginning—I thought, for example, that a Morlet wavelet was some kind of breakfast dish—but after some trial and error, I managed to work out how to construct a wavelet in Matlab, how to perform convolution using the Matlab `conv` function (although I didn't really understand what convolution was), how to compute a decibel, and so on. And it worked! I turned a wiggly line into a colorful 2-D time-frequency power plot.

I initially thought that I would be satisfied with that bit of knowledge and that I would go back to my ERP and fMRI studies. I was wrong. Perhaps there was some afterburn of my New Year's resolution, or perhaps some piece of sanity inside me withered away, but I couldn't stop there. I was unsatisfied with what I knew about time-frequency-based analyses of EEG data, and I had to learn more. Even as I finish writing this book—exactly 6 years after making that fateful New Year's resolution (I am writing this on December 31, 2012)—I remain unsatisfied with what I know about time-frequency-based analyses of EEG data. What started off as a means to satisfy an idle curiosity and pass a few hours in the evenings turned into a career- and life-absorbing obsession that has not abated since early 2007. That obsession is to understand the mathematical and Matlab-implementational mechanics of EEG data analyses and to figure out ways to explain how those analyses work to people who are intelligent but who lack a PhD in mathematics (including myself, at least concerning the latter category).

I suspect that many people who apply time-frequency-based analyses are like me before 2007: they know the basics of how to interpret the colorful plots, but they also feel that some magic is involved, or at least, some very complicated math that they don't, won't, and can't understand. This is unfortunate for a number of reasons because, as I outline in chapters 1 and 37, learning the methods behind the magic will help scientists be better scientists.

Each year, I teach a class on analyzing EEG data in Matlab, and nearly every person in that class, ranging from advanced bachelor's students to full professors, starts off knowing very little about time-frequency-based analyses and, within 2 months, can program and talk intelligently about some of the most advanced analyses applied to EEG data. This is not due to any special quality of their teacher, but rather, it is because the material is not that difficult to learn; there is simply a lack of good resources from which to learn it.

The difficult part is explaining the math and theory in a language that nonmathematicians can understand. No offense to those with a PhD in mathematics or physics, but I think that the math experts who develop the analyses forget what it's like not to have their impressive amount of background knowledge. They forget what it's like not to know what convolution means or how a Fourier transform works, and they forget that to many people—even highly educated psychologists and neuroscientists—an imaginary number is something out of a trippy 1960s cartoon, and a complex space is a description of an M. C. Escher drawing. Thus, the overarching goal of this book is to demystify time-frequency-based analyses, to take the magic out of those red and blue blobs, and to try to make this wealth of powerful and insightful data analysis tools accessible to anyone with sufficient motivation and some spare time, regardless of their background in math, Matlab programming, or EEG research.

Writing this book was an incredibly rewarding experience for me, much more so than I initially thought it would be. I do not know whether it will be as good and accessible a resource as I intend it to be, but I hope that you find working through this book educational and rewarding. Good luck and have fun. And be careful what you wish for on New Year's Eve!

Acknowledgments

Although I wrote all of the words and the Matlab code in this book on my own, this work is the culmination of many years of lessons learned from myriad experiences and from myriad people. I am and will continue to be shaped by the stimulating, encouraging, and, at times, disillusioning and frustrating interactions I have with colleagues, friends, mentors, and students, by lectures I have heard and symposia I have attended, and by the scientific papers and manuscripts I have read (even the ones that could and should have been better). I will not list names here because the list would be too long and the danger of accidentally omitting names is too great. If you ever had a discussion with me, if you ever gave a talk when I was in the audience, if you ever asked me a difficult question during one of my talks or poster presentations, if you ever reviewed one of my manuscripts or grant proposals, or if you ever wrote a paper that I read or reviewed: Thank you.

I Introduction

1 The Purpose of This Book, Who Should Read It, and How to Use It

1.1 What Is Cognitive Electrophysiology?

Cognitive electrophysiology is the study of how cognitive functions (including perception, memory, language, emotions, behavior monitoring/control, and social cognition) are supported or implemented by the electrical activity produced by populations of neurons.

Cognitive electrophysiology is a broad field that contains a spectrum of researchers. At one end of the spectrum are researchers who are mainly interested in cognitive processes. For these scientists electrophysiology is useful because it is more sensitive than behavioral measures such as reaction time or introspective self-report and therefore is better able to dissociate cognitive processes and their subcomponents. At this end of the spectrum task design and theory development are crucial, and sophisticated data analyses with precise neurophysiological interpretations are less important. Understanding neural mechanisms is relevant, but ultimately the goal of the research is to dissect and understand the cognitive components of behavior rather than the physiological properties of the brain.

At the other end of the spectrum are researchers who are mainly interested in discovering the functional properties of neural networks and who use cognitive paradigms as tools to elicit specific patterns of neural activity. For these scientists electrophysiology is useful because it is a direct measure of neural population-level activity, can link research in humans to computational models of neurobiological and neurophysiological processes, and offers an opportunity for cross-species comparisons. At this end of the spectrum sophisticated data analyses and neurophysiological interpretations of results are most important, and cognitive tasks (if used at all) are generally as simple as possible, containing few conditions and many trials. Cognitive theories are useful as interpretational frameworks, but ultimately the goal of the research is to understand how the brain works rather than to dissect components of behavior.



Figure 1.1

Cognitive electrophysiology is a field defined by a spectrum from cognitive to electrophysiology. As a cognitive electrophysiologist, where on this spectrum you consider yourself will help guide your experiments, hypotheses, data analyses, target journals and conferences, and career choices.

Probably you find yourself somewhere between these two extremes (figure 1.1). It is useful to think about where you place yourself in this spectrum because it will help guide how much time and energy you devote to reading cognitive psychology papers versus neuroscience papers, the kinds of analyses you should perform on your data, and the types of journals and conferences that will serve as outlets for your findings and ideas. You should not criticize other scientists, or allow yourself to be criticized, for being on different positions in this spectrum.

1.2 What Is the Purpose of This Book?

The purpose of this book is to teach you the conceptual, mathematical, and implementation (via Matlab programming) bases of time-, time-frequency-, and synchronization-based analyses of magnetoencephalography (MEG), electroencephalography (EEG), and local field potential (LFP) recordings from humans and nonhuman animals. If you go through this book chapter by chapter and implement the examples in Matlab using the provided sample data or your own data, you will develop an understanding of why and how analyses are performed, how to interpret results, what are the methodological and practical issues of these analyses, and how to perform single-subject-level and group-level statistical analyses.

Noninvasive human electrophysiology has been around for a long time, at least since the 1960s and of course dating back to Hans Berger's recordings in the late 1920s. However, in the past decade there has been a surge of publications using advanced analysis techniques for EEG and MEG data (figure 1.2). These analysis techniques include spectral analysis, time-frequency decomposition, and synchronization analyses. This is in large part thanks to increases in computing power, software programs such as Matlab with reasonably easy scripting languages, and the development of open-source toolboxes (such as eeglab, fieldtrip, spm, nutmeg, cartool, and BrainStorm) and commercial software (such as Brain Vision Analyzer, Curry, and BESA) that can perform those techniques. These developments have allowed researchers to explore their data and to link electrophysiological dynamics to behavior and cognition in ways not previously possible.

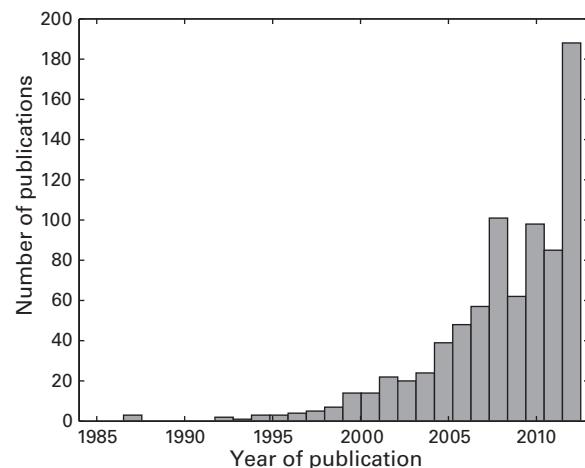


Figure 1.2

This histogram shows that the number of publications utilizing time-frequency analyses has increased considerably over the past years. The data were obtained from a pubmed search of EEG or MEG or LFP and time-frequency, and were accessed in December 2012.

However, despite the increase in availability of automated programs to perform advanced analyses, an understanding of the mathematical and methodological details and the knowledge of the pitfalls associated with these analyses are lacking in many researchers who use the automated programs. That is, the analysis methods available are becoming increasingly sophisticated, but many of the researchers who apply those methods do not fully understand what happens “under the hood” when they click buttons in a software program or run some lines of Matlab script that they were given.

People often ask whether there are tutorials or books from which to learn what those analyses mean and how they work. Unfortunately, the answer has always been: No, you have to be a good Matlab programmer, be willing to learn math on your own, and spend lots of time struggling in Matlab until these concepts become internalized and you can apply them flexibly. Widespread use of advanced data analysis techniques in cognitive electrophysiology is limited by a lack of resources from which to learn those techniques. This is unfortunate because cognitive electrophysiology will benefit from researchers who, for example, have a strong background in psychological theories but less formal mathematical training.

There are good resources for learning more generally about EEG and ERP research (Handy 2004; Luck 2005; Regan 1989), and there are online tutorials (e.g., on the eeglab and fieldtrip websites) that provide general introductions about using software tools and interpreting results from time-frequency-based analyses. But these resources are less useful

for understanding the math or the potential pitfalls of such analyses, and they provide insufficient instructions for people who want to implement the methods themselves or further adapt existing methods. On the other side, there are resources that assume a considerable amount of background knowledge in mathematics or physics. These resources often leave variables in equations undefined, have little useful information about practical implementation matters such as how to perform statistics or how to deal with limited data, and they are largely inaccessible to those without a graduate-level mathematical or physics education.

Thus, what is missing is a resource designed for cognitive scientists, neuroscientists, cognitive neuroscientists, psychologists, and the like, who are intelligent and motivated to understand and implement data analyses but who lack the formal training in mathematics and computer science to utilize the mathematically dense analysis resources or to read the raw computer code behind software packages. This book is designed to provide such a resource.

1.3 Why Shouldn't You Use <Insert Name of M/EEG Software Analysis Package>?

You can be confident that the algorithms in commercial software packages and Matlab toolboxes such as Brain Vision Analyzer, eeglab, fieldtrip, spm8, nutmeg, ELAN, cartool, BrainStorm, and others are correct and valid. If you want to perform the analyses that these toolboxes are specifically designed to perform (often, these are the analyses that the research groups developing the toolboxes find most relevant for their research), then there is nothing wrong with using point-and-click kinds of analysis programs. But if you want to understand the details of the methods, or if you want to have the flexibility to perform analyses not included in these packages, you will need an intimate understanding of the math and logic behind time-frequency decomposition as well as an understanding of the practical implementation-level details, effects of parameter selection, and potential methodological issues of the analyses. The more you understand how the analyses work and how they are implemented in Matlab, the more flexibility and freedom you will have to analyze your data in a manner best tailored to your specific needs, research questions, and experiment design. Ultimately, this book is about maximizing your freedom, at least when it comes to analyzing cognitive electrophysiology data.

You should not feel discouraged from using existing analysis packages, and you should not feel pressured to write all of your own code for your analyses. However, it is important to understand what happens to your data from raw form to publication-quality figures. Writing your own code will ensure that you understand each step, but even if you never write your own code, and even if you never open Matlab, you should have at least a basic working

understanding of what happens to your data when you click on the “analyze now” button in your analysis software.

It does not matter if you use point-and-click graphical user-interface programs or run some lines of Matlab code; neither guarantees that you will know what you are doing. Graphical user interfaces and Matlab scripts can be equally well used, abused, or misunderstood. The hope is that, by going through this book, you will understand what happens to your data when you click that button or run a few lines of code that someone gave you.

1.4 Why Program Analyses, and Why in Matlab?

Perhaps you do not need to spend time programming analyses and instead can understand how time-frequency-based analyses work simply by looking at equations and illustrative figures. If this strategy works for you, then that is great; you can consider yourself in the upper echelon of gifted mathematically minded people. For the rest of us, mathematical equations may seem to make sense while we are looking at them, but this level of understanding is fairly superficial and rigid. In my experience teaching this material, most students look at an equation and slowly utter “okay,” as if they hope that by declaring the equation sensible, they will somehow understand what it means. But then they have a bewildered expression when it is time to turn to Matlab and implement the equation. Once they have worked through the implementation and can see visually what the function does to input data, their eyelids raise, their backs straighten, and they smile, because now they finally realize what the equation means. Experiences like these have convinced me that most people learn time-frequency-based analyses by implementing them, not by looking at equations. This is the reason that this book comes with over 10,000 lines of Matlab code, and this is the reason why you will learn best from the book if you go through the book in front of a computer with Matlab open and work through the equations in computer code as you read the book (more on this point in section 1.6).

Then there is the question of why one specific programming language—Matlab—is highlighted in this book over other programming languages. Matlab is a high-level programming environment that is relatively easy to learn and use. Several of the most widely used M/EEG analysis packages are Matlab toolboxes. Because it is so widely used, Matlab code can easily be shared with people in other labs and in other countries. Matlab has a command interface that can store and make accessible large amounts of data. This is advantageous because you can easily inspect data during each stage of processing and analysis, and you can easily inspect results from different subjects. Matlab also makes very nice-looking plots that are customizable and that can be exported as pixel-based image files (jpg, bmp, png,

tiff), vector files (eps), or movies, which can be used to make presentations and publication-quality figures.

Despite the focus on Matlab, the analyses and algorithms presented in this book are platform independent, and you could translate the code that accompanies this book to any other programming language, such as Python, C++, and maybe even BASIC. If you do not have access to Matlab, Octave is an alternative program you can use. Octave is a free software that runs on Windows, Mac, and Unix systems and can interpret most Matlab code. Octave is not quite as fast as Matlab and does not have many bells and whistles, but it works. You should be able to use almost all of the sample code in Octave with little or no modification. Translating the code to other languages such as Python will take more time and effort but should be possible. If anyone is courageous enough (and interested enough) to translate the online Matlab code into another language, I would be happy to post it online for anyone else to download.

For the remainder of the book I will simply write “Matlab” without implying that Matlab is the only software package or programming language that could be used. When specific Matlab commands, functions, or syntax are referred to, Courier typeface is used. Most of the analysis code presented in this book does not rely on any extra Matlab toolboxes. In cases when functions from toolboxes are used, alternatives will be suggested when available. The three most helpful Matlab toolboxes for EEG data analyses are, in order of likely use, the signal-processing toolbox, the statistics toolbox, and the image-processing toolbox.

1.5 How Best to Learn from and Use This Book

This book is organized in what I hope is a logical fashion, meaning you can start on page 1 and continue reading until the end. The best way to go through this book is in front of a computer with Matlab installed and the code and sample data accompanying this book downloaded, so you can run the code and generate the figures that appear in the book. However, concepts are also explained in plain English and with pictures, so you should be able to develop an intuitive, semantic understanding of the analyses, their mathematical bases, and their interpretations even without going through the Matlab code.

The book is organized in a somewhat monotonic fashion such that the ordering of book chapters corresponds roughly to the order in which you perform analyses, progressing from simple to advanced single-subject analyses and finally group-level statistics. Because the material is cumulative, the information in each chapter is built on the information discussed in previous chapters. Thus, you will maximize your learning if you read the chapters in the order in which they are presented.

1.6 Sample Data and Online Code

There are sample data and Matlab code provided with the book, downloadable from www.mikexcohen.com/book. Almost every figure in this book was made using the online code and sample data. Running the code as you follow the book will not only reproduce the figures but will also help you to understand each step of each analysis and will facilitate further exploration and development of the methods. Note that in many cases the code for one figure relies on variables that were created when previous figures from that chapter were made. Thus, if you open the Matlab file for chapter 12 and try to generate figure 12.6, Matlab might crash because of undefined variables. If this happens, start from the beginning of that chapter and run the code to create figure 12.1, then figure 12.2, and so on. I checked all the code for each chapter using Matlab versions 2011a and 2012 on 64-bit Windows and on 64-bit Mac. If you have a different Matlab version or a different operating system, it is possible that you might need to make minor adjustments to the code. Some features of the code will crash on older versions of Matlab (such as using a tilde to suppress output of a function); these situations and their solutions are presented on first use in the code.

I thought carefully about which EEG dataset to include as sample data in this book. I decided not to use the cleanest and most ideal dataset I could find. Rather, I decided to use a dataset that I think represents the signal-to-noise characteristics of a typical subject in a scalp EEG experiment. Thus, the single-subject results shown in this book reflect the quality of results you can expect for single-subject data analyses. Labels and locations of electrodes are shown in figure 1.3. The dataset is one condition from one subject taken from Cohen and Ridderinkhof (2013). The data have been cleaned by manual trial rejection and independent components analysis using the eeglab software package (Delorme and Makeig 2004). Some intracranial EEG data are also provided, specifically for cross-frequency coupling analyses.

You can use your own data instead of the sample data, but keep in mind that you might need to change some code depending on the format and nature of your data compared to the sample data.

I encourage you to use and adapt the Matlab code for your data. However, keep in mind that the scripts were written specifically for the analyses presented in the book and for the sample dataset. If you modify the code it is your responsibility to do so correctly and appropriately. Using the code without reading this book may result in inappropriate or incorrect analyses or misinterpretations of results. Again, it is your responsibility to use the code appropriately. If you do not feel comfortable adapting the code to your own data, then it is best to use the online code only for the provided sample data when learning from this book



Figure 1.3

Electrode names and two-dimensional topographical locations of EEG electrodes in the sample data provided with the book.

and to use the analysis software package that you are comfortable with when analyzing your own data.

1.7 Terminology Used in This Book

Most of the material in this book can be used for EEG, MEG, electrocorticogram (ECoG, also called intracranial EEG), LFP, electromyography (EMG), functional magnetic resonance imaging (fMRI), near-infrared spectroscopy, or any other discretely sampled time-varying signal. Some of the analyses and statistics in this book are geared toward human scalp EEG and MEG measurements; using the material provided here for other purposes may require some adaptation.

For simplicity and consistency throughout this book, I write “EEG,” but this does not imply that the analyses are inappropriate for MEG or LFP or any other time-varying signal. In some cases there are differences between the treatment of MEG and EEG data (for example, differences arise concerning volume conduction and some spatial filters); these situations are specified. For the same reason, the term “electrode” is used for convenience, but this is not meant to exclude those analyses from being applied to MEG sensors.

Data analyses have terms. Analysis terms are useful short-hand references for a set of assumptions and mathematical equations that are applied to data. For example, the terms “correlation,” “factor analysis,” and “general linear model” refer to specific data analysis

procedures. In the cognitive electrophysiology literature, the same or very similar mathematical analyses have been given different terms by different groups. This is unfortunate because it leads to confusion about what analysis was actually applied to the data. In many cases terms are ambiguous and could refer to very different analyses, with different interpretations and different neurophysiological or cognitive significances. The position taken in this book is that analysis terms should be concise descriptions of the data analysis rather than interpretations of the results or hypothesized neural mechanisms. This argument is further developed in chapters 21 and 37. When many terms for the same analysis exist, I have tried to list all of them when introducing the method and then justify the term preferred in this book.

1.8 Exercises

Many chapters contain exercises at the end. These exercises will reproduce and further develop the material covered in that chapter. These could be used as homework assignments in a class or simply as a personal challenge.

Most of the exercises ask you to take Matlab code presented in that chapter and further develop the code to perform additional analyses that were introduced. Thus, the results you produce from the exercises will, in many cases, look similar to the figures in that chapter. For this reason it will be helpful to use the provided sample data for the exercises rather than using your own data.

Some exercises have right and wrong answers; others do not. Some exercises, for example, ask you to perform an analysis using a variety of parameters. In these cases the point is for you to appreciate the impact (or perhaps lack thereof) of the parameters of different techniques on the results and to consider what range of parameter settings is appropriate and what range of parameter settings is inappropriate and should not be applied to your data.

Answers to the exercises are not provided, but pictures showing what the correct solution might look like are available online along with the code and data. In some cases your solution might look slightly different from mine if you used different parameters or because the exercises involved using randomly generated numbers. Nonetheless, in most cases, if your solution produces a plot that looks like what you can see in the exercises solution online, it is likely that you solved the exercises correctly.

1.9 Is Everything There Is to Know about EEG Analyses in This Book?

Certainly not. There are limitless possibilities for analyzing data, and publications detailing new ways to analyze data appear almost monthly in peer-reviewed journals. Included in

this book are what I consider the most useful, promising, and accepted approaches for linking EEG dynamics to cognitive processes. Others may disagree. There are simply too many possible methods to detail in one book. To those scientists whose published methods are not included in this book but who think that their methods are relevant for understanding neurocognitive function, please accept my apologies and do not take the omission of your method as a negative scientific evaluation of your work.

That said, by going through the methods in this book, you will be well prepared to learn and develop other analysis approaches that are not presented here or are not discussed in depth. Many of the basic methods detailed in this book (including convolution, the Fourier transform, Euler's formula, wavelets, circular variance, and permutation testing) are fundamental and form the groundwork for other advanced data analysis methods.

None of the methods presented in this book is my invention, although in some cases, I suggest minor adaptations or improvements to existing methods (the only equations to which I can claim partial credit are 14.1, 19.3, and 26.10). Rather, this book explains in plain English, math, and Matlab how analyses that are commonly used in the literature can be performed, how the results can be interpreted, and what pitfalls and potential methodological concerns arise with the use of certain methods. Not all of the methods are appropriate for all datasets and all experiments. Whether an analysis method is appropriate for your data and your experiment depends on many factors. Although I include some suggestions for the circumstances or research questions for which particular analyses are well suited, it is ultimately your responsibility to determine whether an analysis method is appropriate for your data and for your experiment.

1.10 Who Should Read This Book?

This book is written specifically for individuals who have little formal mathematical training but who are motivated to learn the conceptual, mathematical, and implementational bases of EEG data analyses. The book is appropriate for advanced undergraduates up to full professors but is probably of most practical use for graduate students and postdoctoral researchers who are conducting or would like to conduct EEG research. Some experience with EEG and Matlab (or other programming language) is useful but not necessary.

Scientists with formal and extensive training in mathematics, engineering, or physics will find some of the material in this book below their level (e.g., the mechanisms of convolution and the Fourier transform). Other material, however, particularly regarding practical matters of dealing with statistics, trial count, data interpretation, and so on, should be useful to even the most mathematically savvy readers. There are other books that are more dense with math

and Matlab for advanced users (Chatfield 2004; van Drongelen 2006), although resources such as these books tend to have less information about practical matters and implementation details and, in some cases, are not specifically about EEG analyses, and thus, the reader must still have some additional expertise to adapt the material to EEG data analyses. References are cited throughout the book for further reading.

This book may also be useful to individuals who do not plan on analyzing EEG data but who would like to learn more about how the analyses work and how to evaluate results of time-frequency analyses. If you are completely new to EEG, the general introduction to EEG and time-frequency results in chapters 2, 3, and 5 should be a useful starting point.

The book is written specifically with a view toward scalp-recorded EEG and MEG recorded in humans. Although some of the discussions on preprocessing (chapters 6–8) and spatial filtering (chapters 22 and 24) apply primarily to scalp EEG and MEG data, the sections on time-frequency decomposition methods, connectivity, and statistics can be applied to any time series data of sufficient sampling rate (generally, over 200 Hz), regardless of the spatial scale or species from which the data were obtained.

Finally, this book is not meant to repudiate any existing EEG analysis package, nor do I suggest abandoning EEG analysis packages in favor of writing 100% of your own analysis code. Understanding the concepts and math behind EEG analyses is important whether you write your own code, modify someone else's code, supplement existing packages with custom-tailored code, or use user-friendly point-and-click software programs. Understanding the mathematical bases of advanced data analyses will help you correctly perform those analyses, appropriately interpret results, get more out of your data, avoid errors or suboptimal analysis choices, and critically evaluate results you see in publications, talks, and conference posters. The sample data are given in eeglab format for convenience, but this is not meant as an explicit endorsement of eeglab, nor is it meant as an implicit repudiation of any other Matlab toolbox for analyzing EEG data. There are advantages and disadvantages to each analysis package, and I have tried to make this book as package-independent as possible. There are format-conversion utilities from eeglab to other major packages, and it should be fairly straightforward to modify the code in this book to use with the data storage formats of other Matlab toolboxes.

1.11 Is This Book Difficult?

Probably. It starts out easy but becomes progressively more challenging. How difficult it is for you depends on your background level of math, programming, and EEG analysis. Unfortunately, most advanced analyses involve concepts, math, and programming implementations

that are likely to be unfamiliar to many psychology and neuroscience students. But the learning curve is steep, and almost anyone with sufficient motivation should be able to learn the material covered in this book. As mentioned earlier, the book was designed for you to progress through chapters in monotonic order, and cumulative knowledge is expected. Therefore, the book may be more difficult if you skip over some chapters. The chapters you should certainly not skip over are chapters 10–13; these chapters explain the basics of the Fourier transform, convolution, and Euler’s formula, which play central roles in nearly all other data analysis techniques.

1.12 Questions?

If you do not understand some of the material in the book, the first thing you should do is go back to the last section you understood, and then work forward again. Make sure you are doing all the exercises in Matlab. Work with other people, and ask colleagues who might understand the material better than you.

Science is and should be a challenge that you relish and look forward to each morning. Probably you find the experience of learning to be rewarding, which is why you are in science. I hope this book gives you double pleasure: you get to learn something new (how to analyze electrophysiology data), and you get to use that knowledge to learn other new things about how the brain works. Good luck and have fun.

2 Advantages and Limitations of Time- and Time-Frequency-Domain Analyses

2.1 Why EEG?

There are several reasons why high-temporal-resolution techniques such as EEG are exceptional tools for studying neurocognitive processes (Cohen, 2011b). The first reason is that these methods capture cognitive dynamics in the time frame in which cognition occurs. Cognitive, perceptual, linguistic, emotional, and motor processes are fast. Most cognitive processes occur within tens to hundreds of milliseconds. Furthermore, cognitive events occur in a temporal sequence that may span hundreds of milliseconds to a few seconds. High-temporal-resolution techniques are well suited to capture these fast, dynamic, and temporally sequenced cognitive events. By comparison, the temporal precision of the hemodynamic response is 2–3 orders of magnitude slower than that of the electrophysiological response.

For example, theta-band (4–8 Hz) oscillations are implicated in several cognitive functions, including memory and cognitive control. Neuroscientists generally consider theta to be a relatively “slow” brain rhythm, but for our conscious experience of the world, theta is quite fast. You can see this for yourself right now. Clap your hands as fast as you possibly can. That is around the lower edge of theta (4–5 times per second, or Hz). You probably cannot clap at theta for more than a few seconds. A faster—but not the fastest—brain rhythm speed implicated in many perceptual processes is the gamma band (30–80 Hz). How fast is gamma? The next time you are in a car, listen to the sound of the engine; at 3000 rpm a four-cylinder car engine cycles in the gamma band (around 50 Hz). It is so fast that you will first hear the engine as a continuous hum, and it takes some effort to parse that hum into discrete beats. Thus, cognition is fast, brain rhythms are fast, and brain-imaging technologies such as EEG can capture these rapid dynamics.

A second reason why tools such as EEG are advantageous for studying neurocognitive processes is that they directly measure neural activity. The voltage fluctuations that are measured by EEG (or magnetic field changes in the case of MEG) are direct reflections of

biophysical phenomena at the level of populations of neurons. Furthermore, oscillations that can be observed in the EEG signal are direct reflections of neural oscillations in the cortex. The neurophysiological mechanisms that give rise to population-level oscillations are well understood and can be modeled fairly accurately (Buzsaki and Wang 2012; Wang 2010). Although it is not fully known exactly what neurophysiological factors contribute in what proportion to the EEG signal that is measured from outside the head, it is generally not debated that EEG measures the meso- and macroscopic neural dynamics produced by populations of cortical neurons. This can be seen as an advantage over currently used MRI-based functional measures such as BOLD, which do not measure neural events directly and have a more complex relationship between what they actually measure (hemodynamic activity) and what can be inferred in terms of the kinds of neural dynamics that produce or correlate with a hemodynamic response (Singh 2012).

Third, the EEG signal is multidimensional. Although you might conceptualize EEG data as two-dimensional (voltage changes over time and space, where space is measured through different electrodes), in fact EEG data comprise at least four dimensions: time, space, frequency, and power (the strength of frequency-band-specific activity) and phase (the timing of the activity; power and phase are discrete elements of a dimension because they provide largely independent information). This multidimensionality provides many possibilities for specifying and testing hypotheses that are rooted both in neurophysiology and in psychology. The brain can be conceptualized as an enormously complex biological system that uses a multidimensional space for information processing, representation, and transfer. The multidimensionality of EEG data allows for analyses that are inspired by known physiological mechanisms. This offers the opportunity to link findings obtained through noninvasive recordings in humans to invasive recordings in nonhuman animals as well as to biophysical models of neural ensemble activity (Cohen 2011b).

2.2 Why Not EEG?

The brain-imaging tool you use should be the one best suited to answer your research question. EEG is a powerful and insightful brain-imaging technique, but it is not well suited for addressing all research questions. In general, EEG is not well suited for studies in which precise functional localization is important, and EEG is not well suited for testing hypotheses about deep brain structures (although activity from deep brain structures can be measured with EEG in some cases). If your research question involves testing whether different striatal subregions are preferentially involved in two different cognitive processes, EEG will be of little use to your research. However, if your research question requires dissociating activity

in parietal cortex from activity in frontal cortex, the spatial accuracy of EEG is likely to be sufficient to confirm functional/anatomical dissociations at this scale. Nonetheless, EEG is likely to be a suboptimal method for any research questions involving “where in the brain does process X occur or is information Y stored.”

Another set of research questions for which EEG may be a suboptimal brain-imaging technique includes those questions concerning cognitive processes that are slow and that have an uncertain and variable time course. For example, imagine you are studying how the brain generates hypotheses based on abstract information, and each trial in your experiment lasts 10 s. This cognitive event unfolds over several seconds and is likely to have a different and possibly unpredictable time course on each trial because the depth and timing of the hypothesis-generation process will vary trial to trial. In this case the extremely high temporal precision of EEG might be disadvantageous. In fact, many EEG analyses become unreliable when the cross-trial jitters in timing are longer than a few tens or hundreds of milliseconds. In this example the relatively low temporal precision of fMRI is better suited to studying slow cognitive processes. Some social and emotional processes might also have slow and uncertain time courses that can be more difficult to study with EEG.

2.3 Interpreting Voltage Values from the EEG Signal

The unit of measurement of EEG is volts (typically, microvolts or μ V); the unit of MEG is tesla (typically, femtotesla, fT, which is 10^{-15} tesla). For EEG, microvolts are relative values, such that the microvolt recorded from an electrode is actually the change in the measured electrical potential between that electrode and a reference electrode placed somewhere else on the head (section 7.8 further discusses referencing issues).

Although many EEG studies, particularly time-domain studies but also frequency- and time-frequency-domain studies, report microvolt values, these can be difficult to interpret in an absolute sense for the following two sets of reasons. First, the microvolt values will change depending on data processing and analysis decisions, including the choice of reference and the time period used for baseline subtraction. That is, if you apply a baseline subtraction using a baseline window of -200 to 0 ms, the voltage values will be different compared to a baseline window of -400 to -100 ms, although the brain events that those potentials reflect did not change. The voltage values may be even more different when you are comparing a preresponse to a prestimulus baseline period. Furthermore, temporal and spatial filters will change the microvolt values. The second reason that raw voltage values can be difficult to interpret is that microvolt values differ across subjects according to factors that may be of no interest to cognitive electrophysiologists, including skull shape and thickness, scalp

preparation during the recording, the orientation of the **dipole** in the brain that generated the scalp-measured activity, **cortical folding**, whether the subject washed his or her head that morning, and other factors. These are not global factors, meaning that there can be **local differences in voltage values across the head**. The voltage value might also be influenced by the **quality** of the EEG **electrodes** and **amplifier** as well as by built-in **hardware** filters. (The above arguments are valid for the MEG signal as well, except those related to skull and scalp properties, because the magnetic signal passes through the skull and scalp unimpeded.)

Of course, all of the factors listed above influence all trials and all conditions equally, so relative changes in voltage values within a subject can be meaningfully compared and validly interpreted. But **raw** voltage values are difficult to compare and should not be over-interpreted. Thus, if a publication from the 1980s reports an EEG effect that is $1.4 \mu\text{V}$ in magnitude, and you replicate the finding but with an effect size of $3.7 \mu\text{V}$ in magnitude, it is not appropriate to conclude that the effect has increased since the 1980s. There are many possible causes for the difference in voltage values across the two studies, and you should interpret the general pattern of effects and the time-frequency-electrode characteristics of the effects rather than the difference in microvolt values.

For this reason, you should not be concerned that many time-frequency-based analyses transform the original data scale. In fact, analyses that involve **scale transformations** (such as **decibel** normalization for time-frequency power data) offer an **advantage** over analyses that retain the original data scale: **individual differences in raw values are eliminated**, and thus the results can be more easily compared across electrodes, subjects, recording equipment, and publications. An increase in frequency-band-specific power of 2.313 decibels in condition A compared to condition B is an interpretable number that can be compared across individuals and across studies, whereas an increase in the potential of $2.313 \mu\text{V}$ in condition A compared to condition B is difficult to compare across individuals or across studies.

2.4 Advantages of Event-Related Potentials

There are four main advantages of event-related potentials (ERPs). First, ERPs are **simple** and fast to compute and require few analysis assumptions or parameters. If the point of your experiment is to test whether brain processing is different between two conditions, and you have no preconceptions about what kinds of electrophysiological dynamics those differences might reflect, comparing ERPs across the two conditions is the fastest and easiest approach to address this question.

The second advantage of ERPs is their high **temporal** precision and accuracy. Because ERPs generally involve very little data processing and gentle or no temporal filters, the estimate of electrical activity at each millisecond is very precise. As you will learn in chapters 10–14

of this book, time-frequency decomposition involves some temporal smoothing, which decreases the temporal precision of the data. This occurs because the data at each time point become weighted averages of data at previous and subsequent time points. If the purpose of your experiment is to determine at what poststimulus latency the brain can distinguish different visual categories, ERPs will provide a more accurate estimate of that latency than time-frequency results. Note that when low-pass filters or high-pass filters are applied to ERPs, their temporal precision decreases. (On the other hand, many ERP studies neither require nor utilize its high temporal precision, as illustrated in section 2.7.)

The third advantage of ERPs is that there is an extensive and decades-long literature of ERP findings in which to contextualize and interpret your results. Furthermore, this literature and the theories that have been developed around ERP findings provide ample opportunities to generate new and testable hypotheses. In contrast, there is less published research on time-frequency electrophysiological characteristics of cognitive processes. In part this is because the methods are relatively new to cognitive electrophysiology, and fewer people are using time-frequency-based analyses compared to ERPs.

The fourth advantage of ERPs is related to the first advantage: because ERPs are fast to compute and easy to look at, they provide a quick and useful data quality check of single-subject data. For example, if your experiment involves making decisions about visual stimuli, stimulus-locked ERPs should show a visual evoked response over posterior occipital and parietal electrodes, followed by a P3-type component over central and parietal electrodes. If you do not see a spatial-temporal progression of ERP activity that conforms to this general pattern, it is likely that something is wrong with the data or with experiment event markers. Therefore, ERPs should be visually inspected for each subject, even if you have no hypotheses concerning condition effects on ERPs.

2.5 Limitations of ERPs

There are two main limitations of ERPs. The first concerns interpretational issues, particularly with regard to interpreting null results. Null results can often be difficult to interpret in science, but a lack of significant condition differences in ERPs is more difficult to interpret compared to null results in time-frequency representations. This is because ERPs reveal relatively little of the information present in EEG data, as will become clear throughout this book. In other words there are many kinds of dynamics in EEG data that do not have a representation in the ERP.

Figure 2.1 illustrates how task-related information can be lost during ERP averaging. Single trials were simulated by creating a 6-Hz sine wave and adding a 10-Hz stimulus response and some noise. The event-related response is visually identifiable in each individual trial (left

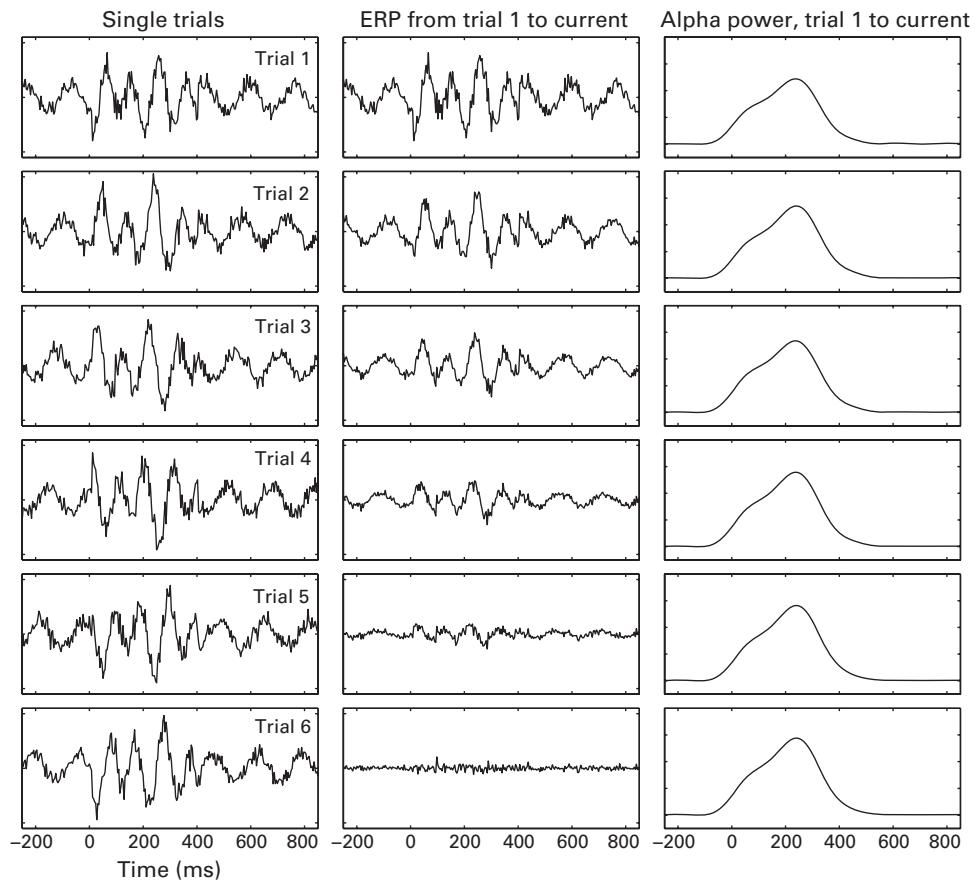


Figure 2.1

Simulated data showing how time-locked but not phase-locked activity (left column) is lost in ERP averaging (middle column) but is visible in band-specific power (right column). Each row in the left column shows a different trial, and each row in the middle and right columns shows averages from the first until the current trial.

column), but the ERP is almost completely flat after only six trials (middle column). The right column shows the recovery of the event-related response by extracting alpha-band power. Of course, this simulation was designed specifically to make this point, but it is also easy to demonstrate in real data that non-phase-locked dynamics are task modulated but are not observable in the ERP. This is shown, for example, in chapter 20.

The second limitation of ERPs is that they provide limited opportunities for linking results to physiological mechanisms. This is because the neurophysiological mechanisms that produce ERPs are less well understood compared to the neurophysiological mechanisms that produce oscillations. Two examples of neural phenomena that have been linked to brain function and cognition across a range of species are interregional synchronization and cross-frequency coupling; these nonlinear neural activity patterns have been studied in nonhuman animals in vivo and in vitro experiments and are believed to be fundamental mechanisms of neural computation and interregional communication. Both of these analyses are not possible with ERPs. Indeed, many ERP papers discuss the role of networks and interactions among brain regions but do not test these hypotheses (indeed, many of these hypotheses cannot be tested with ERPs). This is unfortunate because in some cases within the ERP literature, novel and cleverly designed experiment tasks are developed, and then only superficial analyses are applied that leave the reader with an incomplete sense of the cortical network dynamics underlying task performance.

2.6 Advantages of Time-Frequency-Based Approaches

Time-frequency-based analysis approaches are not guaranteed to give beautiful results. They are also not necessarily the best way to analyze EEG data in all situations. However, conceptualizing and analyzing EEG data as a multidimensional signal that contains frequency as a prominent dimension provide many opportunities to link EEG data to experiment manipulations, ongoing subject behavior, patient groups, and other neurophysiological processes.

There are three major advantages of analyzing and interpreting EEG data using time-frequency-based approaches. The first advantage is that many results from time-frequency-based analyses can be interpreted in terms of neurophysiological mechanisms of neural oscillations. Oscillations appear to be a ubiquitous and fundamental neural mechanism that supports myriad aspects of synaptic, cellular, and systems-level brain function across multiple spatial and temporal scales (Varela et al. 2001). This advantage may not be very important to you if you are more interested in cognitive than in neural processes.

A second major advantage is related to the first: at present, oscillations are arguably the most promising bridge that links findings from multiple disciplines within neuroscience and

across multiple species. Neural oscillations are studied using highly biophysically detailed computational models, in vitro single-cell recordings, in vivo microelectrodes in nonhuman animals, intracranial EEG in human patients (e.g., epilepsy or deep-brain-stimulation patients), and healthy awake humans using scalp EEG and MEG. Other neuroimaging methods that are common in some species are uncommon in others, such as fMRI (rare in nonhuman animals) and single-cell recordings (rare in humans).

A third advantage concerns more practical **statistical** matters. If you accept that EEG data capture a dynamic and multidimensional space of brain processing, then ERPs reveal a fraction of that space, and time-frequency-based analyses reveal more (although not all) of that space. Thus, there may be **many task-relevant dynamics in EEG data that are retrievable using only time-frequency-based approaches** (see figure 2.2 for an example). Even if you are more interested in cognitive processes than in neurophysiological mechanisms, time-frequency-based analyses will likely give you a more accurate answer to

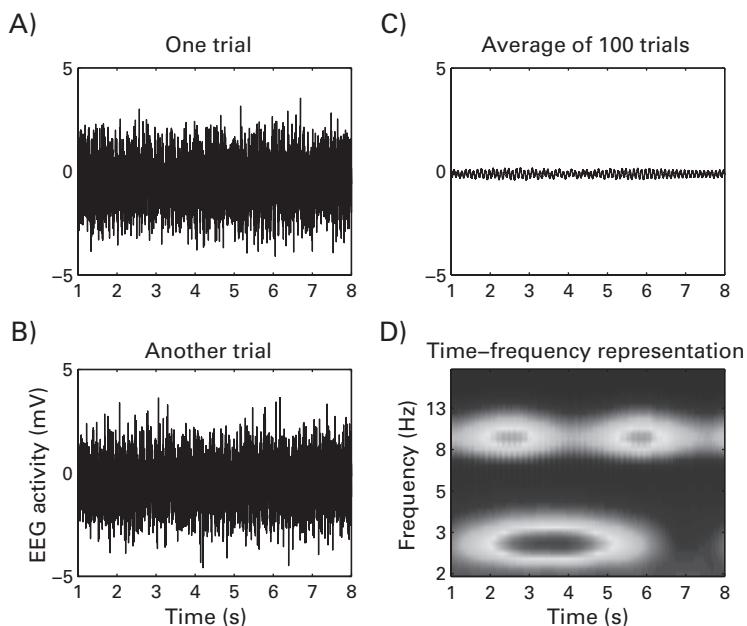


Figure 2.2

Simulated data showing that complex and multifrequency information contained in EEG data may have no representation in the ERP, if that information is non-phase-locked. One hundred trials were simulated; panels A and B show example trials. Panel C shows the ERP of those 100 trials, and panel D shows the time-frequency power. (This figure is adapted from Cohen 2011b).

a question such as, “Does the brain distinguish between two similar types of grammatical errors?”

2.7 Limitations of Time-Frequency-Based Approaches

There are two main limitations of time-frequency-based approaches. The first is the decrease of temporal precision resulting from time-frequency decomposition. This is a general property of rhythmic signal analysis and is not specific to EEG. How much temporal precision is lost depends on the frequency band (lower frequencies generally suffer from more loss of temporal precision) and the parameters used in the analyses. You will learn why the temporal precision decreases in chapters 13 and 14. Although the loss of temporal precision can be mitigated by analysis strategies, the temporal precision of time-frequency-based approaches will almost always be worse than that of ERPs, although it will almost always be better than that of fMRI.

Although ERPs theoretically have a higher temporal precision than time-frequency-based approaches, many ERP studies do not utilize their temporal precision. For example, the P3 component can peak anywhere between 250 and 600 ms poststimulus, and many P3 studies report the peak or mean amplitude from a time window of tens to hundreds of milliseconds. In this type of analysis the EEG data could have been recorded at 100 Hz (or perhaps even 30 Hz) with no loss of information or difference in interpretation. Low-pass filtering of ERP data further diminishes its temporal precision. To be sure, some ERP studies, particularly those focused on early sensory components, do utilize the high temporal precision of ERPs. But it is arguably the case that the temporal precision utilized in many cognitive ERP studies is in the same range as that for time-frequency analyses. This is illustrated in figure 2.3.

A second limitation of time-frequency-based analyses is that the large number of analyses that can be applied to EEG data, and the seeming complexity of those analyses, can be intimidating. New-comers to the field may feel overwhelmed by the large number of analysis possibilities with little guidance on when different analyses should be used and which sets of analysis parameters are appropriate in which situations. Such a diverse range of analyses increases the possibility of performing suboptimal, improper, or incorrect analyses or making inappropriate interpretations of poorly analyzed results. Relatedly, as the number of analyses and comparisons increases, so does the seeming complexity of statistical analyses and correcting for multiple comparisons. The danger of “the paralysis of analysis” increases, meaning that researchers get stuck, constantly performing new analyses with different parameters, while losing sight of the purpose of the study.

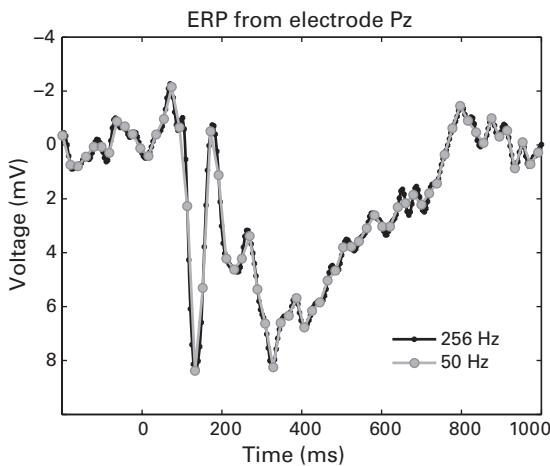


Figure 2.3

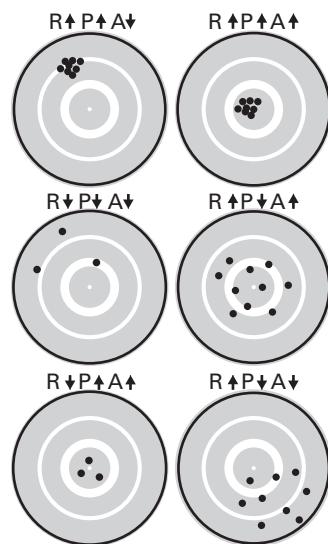
Some ERP components (particularly later ERP components) would not be affected if the sampling rate were much lower (in this case, 50 Hz). ERP data were low-pass filtered at 40 Hz.

This limitation is exacerbated by the relatively small literature on linking time-frequency dynamics to cognitive processes compared to the literature linking ERPs to cognitive processes. Although this field is rapidly growing (as shown in figure 1.2), there are still many cognitive processes for which the time-frequency electrophysiological characteristics are not known or are understudied. Although this provides ample opportunities for exploratory data analyses, it also means that hypothesis-driven research is more difficult, simply because there are fewer findings and even fewer theories from which to make predictions concerning the time-frequency characteristics underlying neurocognitive processes.

2.8 Temporal Resolution, Precision, and Accuracy of EEG

The differences among precision, resolution, and accuracy are subtle but important (see figure 2.4 for illustrative definitions). **Resolution** refers to the **number** of data samples per unit time (the number of dots in figure 2.4), **precision** refers to the **certainty** of the measurement at each time point (the spatial distance among dots in figure 2.4), and **accuracy** refers to the **relationship between the timing of the EEG signal and the timing of the biophysical events that lead to the EEG signal** (the distance of the dots to the center of the bull's-eyes in figure 2.4).

The temporal **resolution** of EEG is determined by the **sampling rate** of the acquisition. It is generally between hundreds and a few thousands of samples per second. Temporal resolution

**Figure 2.4**

Bull's-eye illustration of the differences among resolution (R), precision (P), and accuracy (A). Up-and-down arrows indicate high and low levels. Resolution is illustrated by the number of dots, precision is illustrated by the spread of the dots, and accuracy is illustrated by the distance of the dots away from the center of the bull's-eye.

is what allows you to extract frequency-band-specific information. Some analyses do not require a high temporal resolution (e.g., extracting delta-band power), whereas other analyses require a higher temporal resolution (e.g., cross-frequency coupling). For most analyses, temporal resolutions between 250 and 1000 Hz are sufficient and appropriate.

The temporal precision, in contrast, depends on the analysis applied. Unfiltered ERPs have the highest temporal precision because each millisecond of the ERP was sampled from brain activity at only that millisecond. In contrast the temporal precision of 1-Hz bandpass-filtered activity is much lower because each millisecond of the 1-Hz activity is a weighted average of temporally surrounding activity (up to several seconds of activity). This means that there is some temporal leakage, and 1-Hz events that occur at time 200 ms will affect the 1-Hz results at time 600 ms. You will learn why this is the case in chapters 10–14. Thus, the temporal precision depends on the analysis applied, the parameters selected, and the frequency band (higher frequency bands generally have greater temporal precision).

The distinction between temporal resolution and temporal precision is an important one: time-frequency analyses require high-temporal-resolution data, but after the time-frequency

results have been obtained, the temporal precision is often low enough that the results can be downsampled to, say, 50 Hz (this was also argued to be the case for some ERP components in figure 2.3). In other words, when the temporal precision of the results is decreased by analyses, the temporal resolution can often be decreased to match the temporal precision. This idea is further discussed in chapters 18 and 27.

The temporal accuracy of EEG is extremely high, but for a different reason than its precision and resolution are high. Whereas the temporal precision depends on the analysis and parameters, temporal accuracy is extremely high because brain electrical activity travels instantaneously (within measurement possibilities) from the neurons generating the electrical field to the electrodes that are measuring those fields. Imagine, for example, that electrical fields became “stuck” in the skull such that the latency between brain electrical activity and its measurement at the electrodes could vary from 10 ms to 500 ms. In this untrue scenario EEG would have low temporal accuracy.

2.9 Spatial Resolution, Precision, and Accuracy of EEG

Although EEG has high temporal precision, resolution, and accuracy, its spatial precision, resolution, and accuracy are all relatively low compared to high-spatial-resolution imaging techniques such as fMRI.

The spatial resolution of EEG is determined by the number of electrodes. Most researchers use at least 32 electrodes, and 64 is common. Up to 256 (or 304 for MEG) is increasingly common. The spatial resolution that you need depends on the kinds of analyses you will perform. For example, results from beamforming analyses (a brain source reconstruction spatial filter) improve in accuracy with more electrodes, whereas the P3 to an oddball stimulus is unlikely to be better measured with 256 compared to 32 electrodes.

The spatial precision of EEG is fairly low but can be improved by spatial filters such as the surface Laplacian or adaptive source-space-imaging techniques. Simulation studies show that with many electrodes (>200) and anatomically precise forward models of subject-specific brain and skull anatomy, the spatial precision of source reconstruction can be fairly high. In practice, however, such high spatial precision is more difficult to obtain.

The spatial accuracy of EEG is low. Activity recorded from one electrode does not reflect only activity from neurons directly below that electrode, but rather, from a complex mixture of activities from many brain regions close to and distant from that electrode. Furthermore, the extent to which one brain region contributes to the signal recorded from each EEG electrode depends on cortical anatomy and to what extent that brain region is active at a given point in time. Thus, if you had data from only one electrode,

you would not be able to determine with any reasonable accuracy where in the brain that signal was generated (unless your measure of accuracy is “above the neck” vs. “below the neck”).

Brain networks are organized on several spatial scales. In the context of a **noninvasive EEG**, it is useful to differentiate three spatial scales of neural organization. The first is the **microscopic** scale. This refers to spatial areas of less than a few cubic millimeters and comprises neural columns, neurons, synapses, and so forth. Dynamics happening at this scale are most likely **invisible** to EEG, either because events at this scale do not produce electrical field potentials or because the field potentials they produce are not powerful enough to be recorded from the scalp. The second spatial scale is the **mesoscopic** scale. This scale refers to patches of cortex of several cubic millimeters to a few cubic centimeters. Dynamics occurring at this spatial scale can be resolved with EEG, although it may require high-density recordings (64 or more electrodes) and spatial filtering techniques such as the surface Laplacian or source space imaging. Finally, there is the **macroscopic** scale, which refers to relatively large regions of cortex that span many cubic centimeters. This spatial scale is easily measurable with EEG, even with only a few electrodes.

2.10 Topographical Localization versus Brain Localization

Topographical localization refers to identifying the **electrodes** that show the **maximum effect** under investigation. For example the error-related negativity can be measured from many electrodes, but it can be topographically localized to electrode Cz or FCz. Topographical localization does not necessarily mean that the activity was generated by neural populations directly underneath that electrode, although this may be the case, for example, with radial EEG dipoles after the surface Laplacian has been applied.

Brain localization refers to identifying the **locations in the brain that generated the activity measured from the scalp**. Brain localization is more difficult to determine and involves greater uncertainty than topographical localization, in part because **brain localization** relies on several assumptions about **brain shape, electrical conductivity**, and so on, and in part because the **inverse problem has many possible solutions** that are equally likely (that is, a large number of plausible brain states could produce the same topographical distribution of activity).

Thus, the main difference between these two terms is that topographical localization is a description of an observation, whereas brain localization is an interpretation of a result that is supported by some combination of theory, previous research, and data analyses in combination with spatial filters.

2.11 EEG or MEG?

EEG and MEG measure similar physiological properties and often but not always produce similar results. There are some differences in what EEG and MEG measure, and there are practical considerations, but choosing between EEG and MEG is a less difficult decision than choosing between EEG and fMRI.

In terms of measurement, EEG can detect both radial and tangential sources (this refers to the orientation of the activity dipole with respect to the skull) and is maximally sensitive to radial dipoles on gyral crowns. MEG is maximally sensitive to tangential sources and has low sensitivity to radial sources. For experiments in which localized dipole-like activity can be expected (e.g., early sensory or motor mapping), this EEG-MEG distinction may be a concern, and it is possible that null effects could be obtained with one method and not the other. For cognitive experiments in which a larger patch of cortex might be activated, that patch is likely to extend over cortical folding and thus be measured by EEG and MEG. One example of a difference between EEG and MEG is midfrontal theta, which some studies suggest is easier to measure using EEG than using MEG (Srinivasan, Winter, and Nunez 2006; Stemmer, Vihla, and Salmelin 2004), possibly because the theta emanates from radial dipoles. MEG sensors over frontal regions may also be further away from the brain if the subject leans with his or her head against the back of the MEG helmet. This may lower the signal-to-noise ratio of the signal coming from frontal brain regions.

MEG is better than EEG at detecting high-frequency activity (e.g., above 60 Hz). This is because magnetic fields pass through the skull and scalp, whereas the electrical fields are volume conducted through these tissues, which decreases signal-to-noise ratio at higher frequencies.

In terms of practical aspects of measurement, EEG has several advantages over MEG. EEG is portable and can easily be transported to another lab, hospital, or home. Indeed, a complete EEG recording setup can fit into a small suitcase. Most universities and hospitals have several EEG setups. MEG, in contrast, requires dedicated and specially built rooms and more intensive and expensive maintenance. Most universities and hospitals do not have an MEG scanner, or they may have only one that is shared for clinical and scientific use. One practical disadvantage of EEG is that electroconductive gel must be applied into each electrode, which can be time consuming and annoying for subjects. MEG is advantageous in this respect because it does not require gel or other “wet” preparation (except if a few electrodes are used to track head position or eye movements). Some EEG systems (e.g., those from the company EGI) may not require gel, and future technological improvements may further relieve EEG researchers and their subjects of this gooey annoyance.

In terms of source reconstruction accuracy, it is often said that MEG is better for source localization than EEG. In theory this is not true: with the same number of electrodes and highly anatomically accurate forward models, source reconstruction accuracy is similar for EEG and MEG and can be better with EEG for radial sources. In practice, however, it is often the case that source reconstruction accuracy is higher with MEG than with EEG. This is because most EEG setups have fewer electrodes compared to MEG sensors and because the precise positions of the EEG electrodes are rarely measured (instead, standard template locations are used). This decreases the accuracy of the forward model. Finally, the electrical conductances of the skull and scalp are difficult to measure and can be different for different individuals, and different across the skull for the same individual. More is presented on these topics in chapter 24.

2.12 Costs of EEG Research

Most people will say that EEG research is less expensive than MEG or MRI research. This is a strange justification for favoring EEG research over MRI research, and anyway, it is not necessarily true, particularly not if you use high-quality equipment that provides high spatial resolution and high signal-to-noise data. MEG and MRI scans cost around \$200–600 per hour or per subject. EEG costs are seemingly small per subject—a few dollars for supplies and perhaps \$15–30 to pay the subject for participating—but you should factor in that new EEG caps/electrodes can cost up to \$5,000 each (and you will need several caps for different head sizes) and may last for 100–200 subjects. New amplifiers can also be expensive. Consider this: if a new EEG system is purchased with 128-channel caps and it is used for 400 subjects, the total cost can be $\$100,000/400 = \250 per subject, which is about the same price as MEG or MRI. And if you add powerful analysis computers, an eye tracker, data storage/backup devices, and electrode localization equipment, the cost of an EEG setup can easily exceed \$150,000, or \$375 per subject. Even if you already have the amplifier, consider that 256-electrode caps can cost around \$9,000 (for two sizes), subjects are paid around \$20 for their participation, and wet supplies cost around \$3 per subject. Thus, 150 subjects can cost \$83 per subject.

Thus, EEG research can be cheap if the equipment is already present and if 32 electrodes are used. But new EEG equipment with improved signal quality and high spatial resolution can lead to similarly high costs compared to MEG and MRI. If you are starting a research lab and considering an equipment budget, you should focus on which method will be best for your research questions, not on what method will be the cheapest.

3 Interpreting and Asking Questions about Time-Frequency Results

The purpose of this chapter is to provide an introduction to time-frequency-based approaches of EEG data analyses to people who are new to EEG or who have some experience with ERPs but are new to time-frequency-based analyses.

3.1 EEG Time-Frequency: The Basics

EEG data contain rhythmic activity. Even in raw, unfiltered, unprocessed data you can see rhythmic activity (figure 3.1). This rhythmic activity reflects **neural oscillations**, which are fluctuations in the excitability of populations of neurons (more on this in chapter 5). Some oscillations are fast, some are slower, and some are very slow. Some oscillations last for a long time (seconds to tens of minutes), and some are more transient (tens to hundreds of milliseconds). Some oscillations change according to task events, and some oscillations seem unrelated to task events.

Oscillations are described by three pieces of information: frequency, power, and phase (figure 3.2). **Frequency** is the **speed** of the oscillation and has units of hertz (Hz), which refers to the number of cycles per second and is the reciprocal of time. Thus, 2 Hz means two cycles per second, 40 Hz means 40 cycles per second, and 0.1 Hz means one cycle every 10 s. **Power** is the amount of **energy** in a frequency band and is the squared amplitude of the oscillation. **Phase** is the **position** along the sine wave at any given time point and is measured in radians or degrees. (There is another use of the term phase, which is the angle at which the sine wave crosses a zero point, but this is less commonly used for EEG data analyses; more on this in chapter 15.) Power and phase are independent of each other, meaning that neural dynamics measured through power are different from those measured through phase. One exception is that as power becomes very small, phase becomes increasingly difficult to estimate, but in many cases this is not a major practical concern.

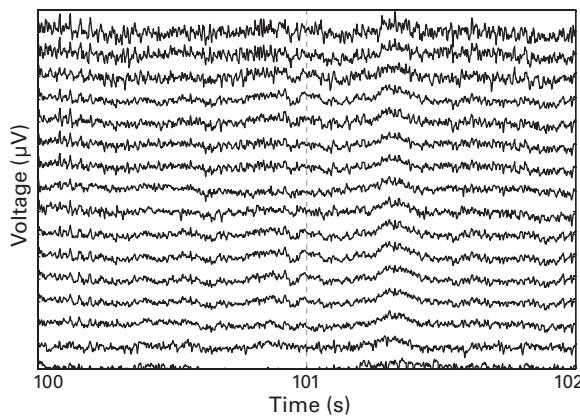


Figure 3.1

Raw EEG data (after 0.1-Hz high-pass filtering) showing oscillations at different speeds and for different lengths of time. Each line corresponds to an electrode.

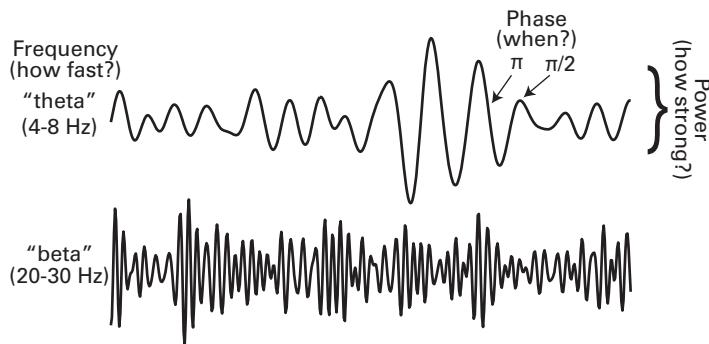


Figure 3.2

The three dimensions that define oscillations: frequency, power, and phase.

The oscillations measured by EEG reflect fluctuations in dendritic electrical activity of populations of neurons. When hundreds to hundreds of thousands of neurons are synchronously active (typically for EEG, these neurons are cortical pyramidal cells), the small and weak electrical fields generated by individual neurons sum and become large enough and powerful enough to travel through brain tissue, fluid, skull, and skin and can be recorded from electrodes placed on or near the scalp that monitor electrical (EEG) or magnetic (MEG) activity. Synchronous activity from small populations of neurons, asynchronous activity, activity from brain structures located below the cortex, and activity from transsulcal and geometrically opposing populations can be difficult or impossible to measure with EEG. Chapter 5 discusses the neurophysiology in more detail.

Brain rhythmic activity contains multiple frequencies simultaneously, which can be separated through signal-processing techniques. This can be understood by an analogy to radio: many radio stations broadcast simultaneously but can be isolated according to frequency bands in which each station transmits information. Similarly, different cognitive processes and neural functions seem to utilize different frequency ranges or conjunctions of frequency ranges.

Brain rhythms are grouped into bands that are defined by logarithmically increasing center frequencies and frequency widths. Brain rhythm frequency bands include delta (2–4 Hz), theta (4–8 Hz), alpha (8–12 Hz), beta (15–30 Hz), lower gamma (30–80 Hz), and upper gamma (80–150 Hz). These are not the only frequency bands; there are oscillations in the subdelta and omega (up to 600 Hz) ranges, but the frequency bands that are most typically associated with cognitive processes in the literature are between 2 Hz and 150 Hz. This grouping is not arbitrary but, rather, results from neurobiological mechanisms of brain oscillations, including synaptic decay and signal transmission dynamics (Buzsaki 2006; Buzsaki and Draguhn 2004; Kopell et al. 2010; Steriade 2005, 2006; Wang 2010). However, there are no precise boundaries defining the bands. You might see theta referred to as 3–9 Hz or 3–7 Hz or 4–7 Hz. Furthermore, individual differences in peak frequencies have been linked to a number of individual characteristics, including brain structure, age, working memory capacity, and brain chemistry (more on this topic in section 35.4).

Changes in rhythmic activity correlate with task demands, including perceptual, cognitive, motor, linguistic, social, emotional, mnemonic, and other functional processes. Indeed, patterns of rhythmic brain activity have been associated with every cognitive process with which they have been examined (to my knowledge). These patterns of activity seem to reflect some aspects of the neurophysiological implementation of the cognitive functions. For this reason EEG is a promising tool to bridge research on humans and research on nonhuman

animals using *in vivo* and *in vitro* techniques and computational models of neurophysiological dynamics.

It can be difficult to localize oscillatory activity precisely and simultaneously in frequency and in time. In general, analysis parameters allow you to choose better precision in one domain (time or frequency) at the expense of poorer precision in the other domain. For example, if you choose analysis parameters that allow you to identify transient activity, it will be difficult to determine whether that activity occurred at 20 Hz or 20.5 Hz. In contrast, if you choose analysis parameters that allow you to identify the precise frequency of activity, it will be difficult to determine whether 6.234 Hz activity differs between conditions starting at 200 versus 230 ms. This is not a problem of EEG data analyses but, rather, a problem of detecting time-varying oscillating signals. Consider another periodic signal: your heartbeat. You cannot measure your resting heart rate by pressing against an artery for less than 1 s (unless your resting heart rate is over 120 bpm, in which case you might have more important things to worry about than temporal vs. frequency precision). On the other hand, if you want to measure changes in heart rate in 5-s steps, you have to accept some uncertainty in the precise rate in order to measure changes over time. For the same reason it is not possible to record 20 ms of EEG and extract information about 5-Hz brain activity. Generally, you can expect uncertainties on the order of tens of milliseconds and several hertz, although this depends on the type of analysis and on the parameters you choose for that analysis. When you read in a paper about time-frequency results that there was “activity at 15 Hz,” this actually means “activity in a weighted range of frequencies such that 15 Hz activity maximally contributed to the result and other frequencies contribute less, according to their distance away from 15 Hz.” The same qualification holds for the timing of activity—the activity reported at 346 ms is actually a weighted sum of activity from preceding and succeeding time points.

There is a distinction between background and task-related activity. Background activity refers to activity that is present in the data but is unmodulated by task events. Neural dynamics without any clear identifiable behavioral or task-related modulation may reflect general properties of neural architecture or may support neural computation in more complex ways than our current analytic approaches can uncover. Either way, background activity is difficult to interpret in terms of neurocognitive processes. This is why some form of baseline normalization is useful: baseline normalization removes or strongly attenuates patterns of activity that are present in the data but are unrelated to the task. Within task-related activity, a distinction can be made between phase-locked and non-phase-locked activity. Activity is phase-locked when its phase is the same or very similar on each trial, whereas activity is non-phase-locked when its phase is different on each trial, even if it is still time-locked to the trial.

events. Figure 2.2 shows an example of non-phase-locked activity, and all ERPs are examples of phase-locked activities. Section 5.3 has further discussion and illustration of this point.

Each electrode measures activity from multiple neural populations. In general, electrodes are maximally sensitive to the brain tissue directly underneath the electrode when the activity is generated by radial-oriented dipoles located in gyral crowns. But neural populations from more distant cortical and, in some cases, subcortical regions contribute to the signal recorded by that electrode in a complex manner that depends on physical distance and activity strength as well as on geometric orientation of the neural sources and, for EEG, skull thickness and shape.

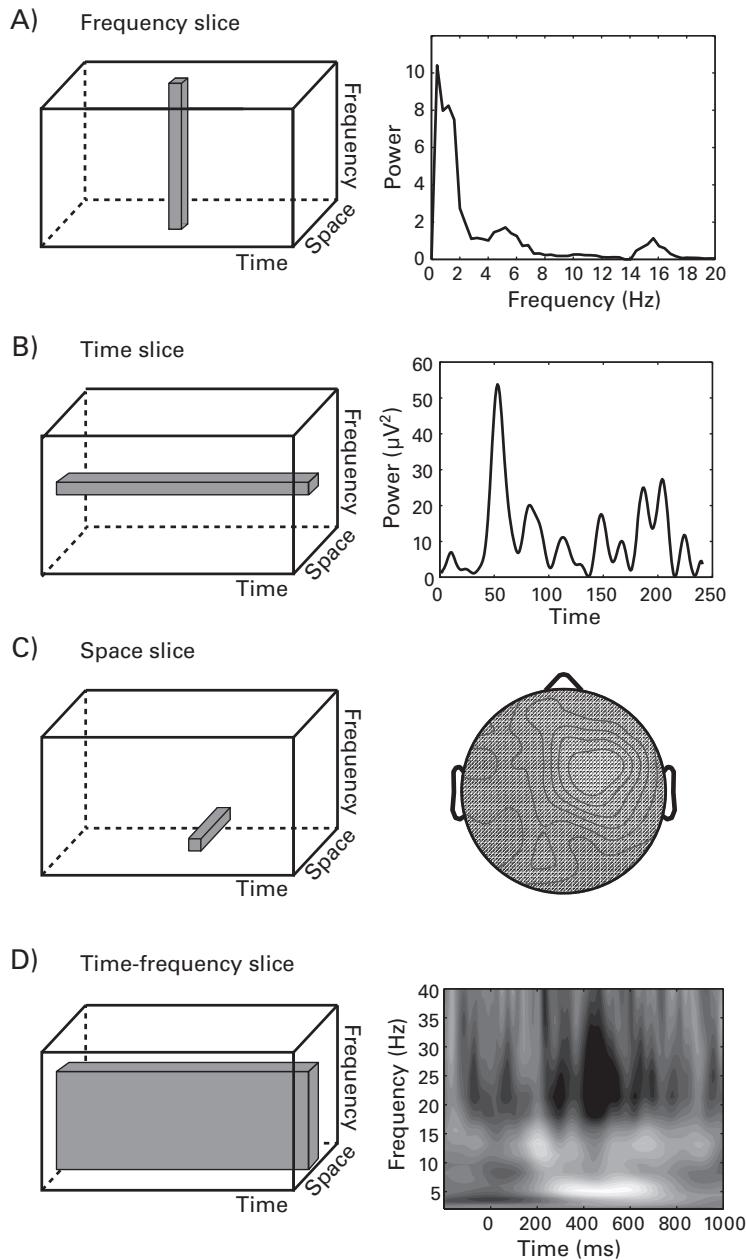
Similarly, many electrodes will record activity from the same brain sources. This introduces spatial autocorrelation (activity at neighboring electrodes is strongly correlated, and the strength of the correlation decreases with distance), which limits spatial precision and may cause spurious or inflated results for some kinds of connectivity analyses. There are several types of spatial filters that can attenuate spatial autocorrelation and thus improve topographical or brain localization. These spatial filters also make the data more appropriate for connectivity analyses.

3.2 Ways to View Time-Frequency Results

Imagine time-frequency results as a three-dimensional (3-D) cube (figure 3.3), with the dimensions being time, frequency, and space (space is measured by electrodes). In practice, there are more dimensions in this hypercube, corresponding to task condition, analysis type, subject, and so on, but 3-D cubes are easier to conceptualize visually. Because it is difficult to view a solid 3-D cube as a 2-D representation on a piece of paper or computer monitor (it is even more difficult to view a 5-D hypercube), in practice, 2-D slices of this cube are shown in papers and in presentations.

The frequency slice (figure 3.3A) This shows power (energy at each frequency band on the y-axis) as a function of frequency (x-axis), collapsing over a period of time that could be hundreds of milliseconds to tens of minutes long. Time is “lost” in these plots in the sense that you cannot determine whether and how the frequency characteristics change over time. These plots are most useful when no or little time-varying changes in the frequency characteristics are expected, such as in a resting state or a sleep stage.

The time slice (figure 3.3B) In this slice, one frequency band is selected, and activity at that frequency band is plotted over time. These plots look a bit like an ERP in the sense that you see a line plotted as a function of time, but unlike an ERP, the activity is specific to one

**Figure 3.3**

The data cube, containing information over time, frequency, and space, is difficult to view or conceptualize and therefore is sliced in different ways to illustrate 1-D or 2-D snapshots of the results.

frequency band. These plots are useful for comparing activity across multiple conditions or electrodes and when there is an a priori reason to focus on a specific frequency band.

The space slice (figure 3.3C) Here, data are shown at one time-frequency point, or the average over multiple adjacent time-frequency points, over electrodes in a topographical plot. This helps visualize the topographical distribution of the effect and facilitates topographical localization.

The time-frequency slice (figure 3.3D) Finally, data can be shown in a time-frequency plot. In time-frequency plots, time is on the *x*-axis and frequency is on the *y*-axis. Typically, frequencies are plotted such that higher frequencies are toward the top, although you may occasionally see the *y*-axis flipped such that higher frequencies are plotted toward the bottom of the plot. The color of the plot (also known as the *z*-axis or the depth) reflects some feature of the time-frequency data, such as power, phase clustering, connectivity, or correlation coefficient. Note that similar-looking plots are sometimes used to show very different kinds of results such as ERP images or activity over time and space (e.g., in cortical layers from laminar recordings). Always inspect the *y*-axis label and legend to be sure.

Other slices Figure 3.3 illustrates the most common ways of slicing the data cube, but they are not the only ways. You might see, for example, a 2-D plot showing activity at each electrode over time, thus a space-time slice; or a 2-D plot showing activity over different frequencies as a function of physical distance of the subject to a target (Vyssotski et al. 2009), thus a distance-frequency plot. The important concepts are that time-frequency EEG results form a data cube and that any illustration of time-frequency results is a line or slice through this cube.

3.3 tfviewerx and erpviewerx

To better explore the data cube, I wrote a small Matlab utility that shows simultaneously the time-frequency plot from one electrode and the topographical map at a selected time-frequency point. To use the utility, you can import a data cube (a time \times frequency \times electrode matrix) when calling the function, or you can simply call the function, and it will open with preloaded results. You can click around on the time-frequency plot and the topographical map to update the display, or you specify time-frequency-space points. The utility is called “tfviewerx” and comes with the online Matlab code. A screen shot of this utility is shown in figure 3.4. You are free to use and modify this utility as you like, including adding or removing features. The tfviewerx utility does not require any Matlab toolboxes or third-party toolboxes. It was programmed in Matlab 2011b and works well on Mac and on Windows computers. It should work in other versions of Matlab and in Unix operating

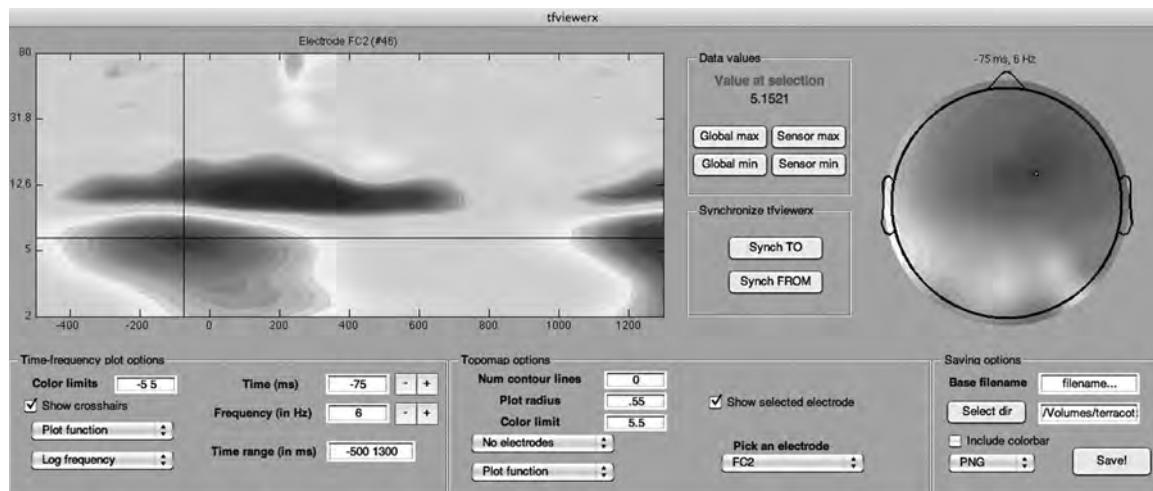


Figure 3.4

A screen-shot of the data-cube-viewing utility `tfviewerx`, which is available online with the Matlab code. Mouse clicks on the time-frequency plot update the topographical map to show the scalp distribution at that time-frequency point, and mouse clicks on the topographical map update the time-frequency plot to show the time-frequency dynamics at the nearest electrode. Multiple `tfviewerx` windows can be opened (e.g., to view results from different conditions) and can be synchronized to show the same time-frequency-electrode point across plots. Type “`help tfviewerx`” in Matlab to learn about how to use this utility.

systems, but I have not tested it extensively. There is also a complementary utility called `erpviewerx`, which is the same viewer but optimized for ERPs rather than time-frequency results.

3.4 How to View and Interpret Time-Frequency Results

If you are new to time-frequency-based analyses, the following five-step plan should help you interpret and talk intelligently about results presented in time-frequency plots.

Step 1: Determine what is shown in the plot The most common data feature to plot is power, but phase clustering, connectivity, and correlation with behavior are also possible. Make sure you understand conceptually what is being plotted (e.g., the difference between power and phase clustering) before moving forward. When in doubt, ask.

Step 2: Inspect the ranges and limits of the plot What are the time and frequency ranges (*x*- and *y*-axes, respectively), and is there activity that seems to be cut off by the boundaries of the

plot? What are the color limits, and are they symmetric (e.g., -3 to +3) or asymmetric (-2 to +4), or bounded by zero (0 to 0.5)? Symmetric color scales can be easier to interpret; asymmetric color scales can be used to highlight increases or decreases in activity.

Step 3: Inspect the results Is there activity at multiple frequencies and time windows, or is the activity all centered in one time-frequency “blob”? Is the activity duration short or long, and is it frequency band-limited or does it span multiple frequency bands? Is there activity during the prestimulus period (this period is often used for baseline normalization)? If results are shown from several electrodes or on topographical maps, is there topographical specificity; that is, are effects present selectively at some parts of the scalp? If no spatial information is presented, from which electrode(s) are results shown, and why?

Step 4: Link the results to the experiment (or to patient groups or gene or drug treatment, or whatever the independent variable is) What does time = 0 refer to? Are there multiple events in the experiment (e.g., first stimulus, second stimulus, response), and how are these different events represented in the time-frequency results? Do the results make sense given the experimental design, and are they consistent with previous research using similar designs? Are there prominent features of the results that are not mentioned by the authors/presenters (this could be due to an priori hypothesis about specific time-frequency windows)? What do the results suggest about the cognitive process under investigation and its potential neural implementation? Finally, **do the results provide any new information about brain function, or could one simply substitute a parameter such as reaction time in place of the results shown and reach the same conclusion?**

Step 5: Understand the statistical procedures used to support the interpretations Is any statistical thresholding used? If not, the results should be interpreted qualitatively and not quantitatively. Is the analysis **hypothesis-driven** or **exploratory and data-driven**? This affects the interpretation of the results as well as statistical approaches that are appropriate. Exploratory data-driven approaches generally require conservative statistical thresholds and corrections for multiple comparisons over time, frequency, and electrodes (or voxels, if the data were analyzed in brain space instead of electrode space). Hypothesis-driven analyses, on the other hand, increase sensitivity and theoretical relevance, and less stringent thresholds such as $p < 0.05$ may be acceptable. Exploratory analyses might lack the sensitivity to detect subtle features of the results, and hypothesis-driven analyses might miss important features of the results that were not predicted by the theory. If the analyses were hypothesis-driven, how were time-frequency-space windows selected for statistical analyses?

At this point, more knowledgeable readers might ask detailed questions about analysis choices and parameter settings that might have affected the results or might highlight specific features of the data.

3.5 Things to Be Suspicious of When Viewing Time-Frequency Results

Below is a list of features of results that should make you suspicious. Seeing one of these features does not necessarily mean that there is something wrong with the analysis or that the data contain artifacts; rather, the following observations deserve special attention because they could reflect analysis errors or data artifacts.

Horizontal or vertical stripes (figure 3.4A) in a time-frequency plot may reflect ripple artifacts from poor filter construction. This can happen if the filter widths are too narrow or if the filter was applied to too little data, thus causing edge artifacts. Chapter 14 has more discussions about how to design bandpass filters and how to detect poorly designed filters.

Brief and large-power effects at high frequencies (figure 3.4B) could be driven by EEG artifacts such as amplifier saturation or a noise spike from a bad electrode. A very large artifact in even one out of one hundred trials can cause such brief and large power high-frequency effects.

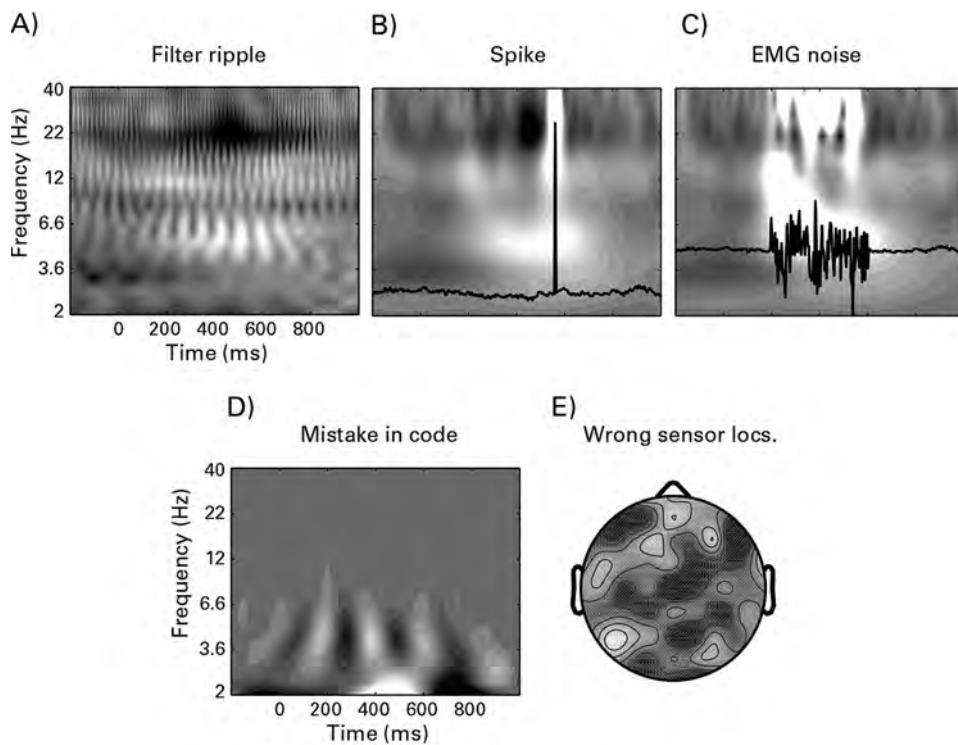
Broadband effects (figure 3.4c) that span many “classical” frequency ranges could be due to mechanical noise or excessive muscle activity from the jaw or neck.

Very fast color changes over time or frequency (figure 3.4D) could be a mistake in the analysis (in this case, the real part of the analytic signal was plotted rather than power). Fast changes in lower frequencies are more suspect than fast changes in higher frequencies because of increased temporal smoothing at lower frequencies. In the case illustrated in figure 3.4D, there is no artifact, but time-frequency plots of the real part of the signal are less informative and thus are generally not shown.

Strange topographical distributions (3.4E) that contain many peaks might be due to noisy or bad electrodes or to an incorrect mapping between electrode label and physical location (this is the case for figure 3.4E). Note that high-pass spatial filters such as the surface Laplacian will increase topographical localization and highlight local spatial features.

High-frequency activity (over 100 Hz) has a low signal-to-noise ratio and may require many trials and special analysis techniques to enhance signal to noise. Note that MEG has better signal to noise at higher frequencies due to decreased signal distortion from the skull and scalp.

For activity in very low frequencies, below 1 Hz, many modern amplifiers have built-in high-pass filters that attenuate activity in very low frequencies, although there are DC-coupled amplifiers that are well suited to measure slow fluctuations. Furthermore, researchers often apply a high-pass filter of 0.1 or 0.5 Hz to eliminate slow fluctuations. Sweat potentials, for example, can cause slow drifts in the data.

**Figure 3.5**

Some features of time-frequency results that should arouse suspicion, although they are not necessarily artifacts. In panels B and C, the offending single trial (out of 99 otherwise good-data trials) is superimposed on the time-frequency plot (EEG trace amplitude is arbitrarily scaled). The topographical map in panel E was produced by randomly swapping electrode label-location mappings.

3.6 Do Results in Time-Frequency Plots Mean That There Were Neural Oscillations?

This is actually a tricky question to answer. “Believers” will be quick to say yes, and skeptics will be quick to say no. The real answer is “It is difficult to know for sure.” On the one hand, EEG measures summed field potentials of populations of neurons, which are strongly oscillatory. Thus, meso- and macroscopic neural processes that generate oscillations will dominate EEG measurements. On the other hand, Fourier’s theorem specifies that any signal can be represented using sine waves, and thus, even nonoscillatory signals have a representation in a time-frequency plot. Chapter 21 has a more detailed discussion of this point.

4 Introduction to Matlab Programming

This chapter is not intended to be a complete guide to Matlab programming. Rather, the code accompanying this chapter introduces you to most of the commands used in this book, and the text provides some general advice and code-writing tips. There are many general Matlab introduction tutorials online. There are also several books from which to learn Matlab, including those written specifically for neuroscientists (Wallisch 2009) and behavioral scientists (Ramsay, Hooker, and Graves 2009; Rosenbaum 2007). However, it is a good idea even for readers with modest Matlab experience to go through the code accompanying this chapter: the code starts simple but gets increasingly complex. You should go through each of the three scripts accompanying this chapter on your own in Matlab, line by line and in order.

4.1 Write Clean and Efficient Code

Perhaps you think that as long as your code works, it does not matter how it looks or whether it is efficient. To some extent, this is true: cognitive electrophysiology studies are and should be evaluated based on the quality of the theories, hypotheses, experiment design, analyses, and interpretations; they should not be evaluated on the aesthetics of the programming code used to analyze the data.

That said, there are three reasons to try to write clean and efficient code. First and foremost, clean code is easy to read and understand. Having clean and easy-to-read code will help you prevent making programming errors, and when there are programming errors, it will help you identify and remedy those errors. Having clean code will also help you adapt existing scripts to new analyses and new datasets. If you write confusing and sloppily written code, you can probably read and interpret it when you first write it, but after weeks, months, or even years without looking at that code, you may get lost and not understand what you wrote. On the other hand, if your code is clean and properly commented, you will be able to return to the code after months or years of not looking at it and still understand what the

code does and how to work with it. This is even more important if you share your code with other people and if you want other people to be able to understand and adapt your code. Again, clean and efficient code will help make their lives and data analyses easier.

There are three strategies to keep in mind that will help you write clean code. (1) Use comments. Write comments before a line of code or collection of lines of code to indicate succinctly what those lines do. Also use comments to specify the size of large matrices and the order of the dimensions in those matrices (e.g., time, frequency, condition, electrodes). Keep comments brief and to the point; do not use comments excessively, otherwise the code may become difficult to find from within a sea of comments. (2) Group lines of code by their common purpose. This is analogous to how you would group sentences into a paragraph. Think of a line of code as a sentence; several lines of code that have one purpose, or one message, should be grouped into a paragraph, separated by one or several blank lines. Relatedly, if there are many functions or commands on one line, consider breaking it up into two or more lines to be easier to read. (3) Use sensible and interpretable variable names (more on this point in section 4.2).

The second reason to write clean and efficient code is that efficient code runs faster. If your code runs faster, you will spend less time waiting for analyses to finish and more time looking at results and working on new analyses. To write efficient code, avoid redundancies (performing the same computations multiple times, e.g., evaluating an equation inside a loop when it could be evaluated outside the loop), pieces of code that are never called or that do nothing, separating into multiple files what would better be done in one file, or keeping in one file what could better be done in multiple files. Whenever possible, perform matrix manipulations instead of loops. Loops should be used only when necessary. Using informative comments will help you figure out ways to make the code more efficient.

The third reason to write clean and efficient code is that clean and efficient code promotes clear and organized thinking. Programming is problem solving. To program, you must first conceptualize the problem, then break it down into subcomponents, and then break those subcomponents further down into individual digestible computer commands. Programming involves taking an abstract idea for an analysis or figure you would like to show and turning that idea into a series of logical and concrete statements. These are also important skills for science in general: science involves taking an abstract idea about how some aspect of the brain or behavior works and turning that idea into a series of logical and concrete experiments, statistical analyses, and theory-relevant interpretations. There are overlapping skills that are useful both for programming and for being a scientist, and it is likely that working on one set of skills will transfer to the other set of skills.

Most of the hard work in programming should be done in your head and on a piece of paper, not in Matlab. Particularly if you are new to programming, start writing your script

on a piece of paper with a pencil. Write down what the script should do and in what order. After you have a plan for how to write the script, then turn to your computer, open Matlab, and start programming.

After you consider the advice in the previous five paragraphs, take this one additional piece of advice: do not obsess over having the cleanest and most efficient code possible. It is not the most important part of being a cognitive electrophysiologist, and you can certainly be a great scientist without ever writing a single line of Matlab code. But as you write code, whether you are programming all of your own analyses from scratch or writing a script to call eeglab or fieldtrip commands, try to keep your code efficient and clear. Your future self will thank you.

4.2 Use Meaningful File and Variable Names

Give files useful names. For example, name a file “Flankers_task_TF_analyses.m” instead of “Untitled84.m.” There is no shame in having long file names. Alternatively (or additionally), write commented notes at the top of each script that explain what the purpose of that script is. Note that some versions of Matlab on some operating systems will give errors when calling files that have spaces in the file names; you can use underscores instead.

It is even more important to give meaningful names to variables. Variables should have names that will allow you to identify and disambiguate the purpose of those variables from other variables. For example, if you have EEG data to store in a variable, it is better to call the variable something like `EEG_data` than something like `variable_name`. Avoid using variable names like `a`, `b`, `c`, `d` except if they are used only in small loops or as temporary variables that will be used only for a few lines of code. You can also develop your own style for naming variables. For example, I always put “`i`” at the end of counting variables in loops (“`i`” for index). This is particularly useful when using multiple nested loops over subjects (looping variable `subi`), channels (looping variable `chani`), frequencies (looping variable `freqi`), and trials (looping variable `triali`). In Matlab, variable names cannot start with numbers but may contain numbers, cannot have many nonalphanumeric characters (e.g., &, *, %, \$, #), and cannot have spaces (underscores can be used instead).

4.3 Make Regular Backups of Your Code and Keep Original Copies of Modified Code

Spending hours on an analysis script only to accidentally delete or overwrite the file is not an enjoyable experience. Fortunately, Matlab automatically creates backup files (.asv or .m~ on Windows/Mac), which should help minimize accidental script loss, although these backups might be tens of minutes behind the original version. Matlab .m files used for scripts

and functions contain only text and therefore occupy very little disk drive space, so there is little need to delete a file. You can, for example, email scripts to yourself to maintain time-stamped backups.

Avoid working on multiple copies of the same analysis script. For example, if you keep a time-frequency decomposition script for an experiment on the server computer, and then you back up that script on your portable USB drive, make sure you are modifying only one of those scripts. Otherwise you might end up making different changes to different versions of the script.

If you modify functions that come with Matlab or a third-party toolbox such as eeglab or fieldtrip, it is a good idea to save the original function with a different file name. For example, you can save the file as “filename.m.orig” and then modify “filename.m.” The other option is to modify the original file and comment each line you modify or add. This latter option is suboptimal when you make many modifications because keeping track of every change you make may get cumbersome.

4.4 Initialize Variables

Initializing variables means that you **reserve space in the Matlab buffer** for that variable by **creating the variable before populating it with data**. Typically, the variable is set to contain **all zeros**, or all ones, or all NaNs (not-a-number). You do not need to initialize all variables, particularly smaller variables or variables that you use for only a short period of time. However, **larger or more important variables**, such as those that contain data you intend to analyze or save, should be initialized before use. In Matlab, unlike in some other programming languages, it is permitted to add elements or dimensions to variables without initializing them first (this is demonstrated in the online Matlab code [script “c”] that accompanies this chapter), but this behavior should be avoided when possible. There are three reasons why you should initialize variables.

First, initializing variables, particularly for large matrices, helps **avoid memory crashes**. Second, initializing variables that will be populated inside a loop helps **prevent data from previous iterations of the loop contaminating current iterations**. For example, imagine you have a script that imports and processes behavioral data, and the script contains a loop over subjects. There is a variable called `trialdata` that stores data from each trial within a subject before computing cross-trial averages. If the first subject has 500 trials and the second subject has 400 trials, without your initializing `trialdata` at each iteration of the loop over subjects, the data for the second subject will contain 500 trials, the last 100 of which were left over from the first subject. Obviously, this is a situation you want to avoid. Programming

errors like this can be difficult to become aware of because they are unlikely to produce any Matlab errors or warnings. In some cases, you may not know how big a variable will be and thus cannot initialize it to the final size. If there is any danger of cross-loop-iteration contamination, you can **clear the variable** in an appropriate place (typically, the beginning of the loop), or you can **initialize the variable to be bigger** than necessary and then remove unused parts of the matrix afterwards. In this case it might be **better to initialize the matrix to NaNs** instead of zeros; if you accidentally forget to remove the unused part of the matrix, you do not want to average zeros into the data (the function `nanmean` will take the average of all non-NaN values in a matrix).

Another potential mistake that may be difficult to find because it will not produce a Matlab error or warning is the **location of the initialization**. Going back to the example from the previous paragraph, imagine further that you have another variable that contains trial-averaged data from each subject, called `subjectdata`. You should initialize `subjectdata` before the loop over subjects, and you should initialize `trialdata` within the loop over subjects. If you initialize `subjectdata` within the subject-loop, you will end up with a matrix of all zeros except for the last subject because data from previous subjects will be reinitialized at each iteration.

The third reason to initialize variables is that it will help you to think in advance about the sizes, dimensions, and contents of large and important variables. As noted at the end of section 4.1, the more thinking you do before and during programming, the cleaner, more efficient, and less error-prone your scripts are likely to be.

4.5 Help!

Even the most experienced and savvy programmers get stuck sometimes. There will certainly come a time when you need help, at least with Matlab programming. Matlab programming issues generally fall into one of three categories.

You know the function name but don't understand how it works. Start by typing `help <function name>` in the Matlab command. In some cases you can type `doc <function name>` to get a more detailed help file. Many functions have help files that contain examples; try running the example lines of code. Most functions are simply text files; you can open the function with `edit <function name>` and look through the code to try to understand how it works. This option is more useful when you develop some experience with programming and reading other people's code. Not all functions are viewable; some functions are compiled for speed. Try running the code with simpler inputs for which you can better understand the output. For example, if you have a large four-dimensional matrix and are unsure which

dimension is being averaged in the mean function, try creating smaller matrices of only a few numbers, for which you can easily compute the means, and compare these against the function outputs. You can also plot the data before and after calling the function to see what effect the function had on the data. You can search the Internet to see if there are additional discussions of that function or additional examples of how to use the function. You can also ask colleagues for help. However, try to figure it out on your own before asking someone else. It may initially seem like a waste of time to spend 30 min understanding a function when a colleague could explain it to you in 30 s, but if you figure it out on your own, you are likely to learn more from that experience, and therefore, you are likely to avoid making that kind of error in the future. This is how you become a better programmer.

You know what you want Matlab to do, but you can't figure out the command or function for it. This is a frustrating problem. The three ways to solve this issue are by reading the help file of similar functions (in particular, look for the “see also” part at the end of the help file), searching on the Internet for what you want Matlab to do, and asking a colleague.

You know what you want Matlab to do and you know the command to do it, but there are errors when you run the code. Newcomers to Matlab seem to spend most of their time and frustration on this kind of Matlab issue. If your Matlab command window contains more red than black, don't give up hope; errors become less frequent over time as you learn from mistakes. Here are some tips for resolving this kind of Matlab error.

First, find the function or command that produces the error. This may sound trivial but can be tricky if there are multiple functions called in one line. For example, if you get an error with the following line of code:

```
abs(mean(exp(1i*angledata(:,trials4analysis)),2)),
```

you need to determine whether the error resulted from one of the three functions called (`abs`, `mean`, `exp`) or from one of the two variables (`angledata` or `trials4analysis`). To locate the error, start from the innermost (or most deeply embedded; most likely in the middle of the most parentheses or brackets) variable or function, and evaluate this in the Matlab command. In this case, start by evaluating `trials4analysis` and see if that produces an error. If not, move to the next function or variable—in this example, `angledata(:,trials4analysis)`. Eventually, you will find where the error occurs.

Now that you have located the error, read the error message (the red text). Sometimes, error messages seem to be written in a foreign language. If you do not understand the error, look for keywords. For example, if the error message contains “matrix size” or “matrix dimensions,” use the `size` function to examine the dimensions of the variable that produced

the error. If the error message contains “subscript indices must be real positive integers” or “index exceeds matrix dimensions,” then probably your index variable (in the above example, `trials4analysis`) has zeros, negative numbers, fractions, or numbers greater than the size of the matrix being indexed. Some error messages are more self-explanatory, such as “incorrect number of input arguments.”

If you still cannot solve the error, try plotting all of the inputs and outputs of the functions that precede the error; perhaps you will notice something strange or obvious in the plots. Missing data points in plots are likely to be NaNs or Infs (infinity)—these can cause errors in some functions or when used as indexing variables. You can also use the step-in option, which will halt the function that produced the error at the offending line. This is beyond the scope of this chapter, but you can read more about stepping-in on the Internet or in Matlab tutorials.

Other possible causes of an error include that the function is not in Matlab’s path (and thus Matlab does not know the function exists), that the function is contained in a toolbox that you do not have, or that the function is compiled for or relies on libraries that are specific to a version of Matlab (e.g., 32-bit vs. 64-bit) or operating system. Errors can also occur if you use a variable name that is the same as a function name. For example, if you write `mean=my_data_part`; Matlab will recognize `mean` as a variable instead of the function. Using variable names that are existing functions is bad practice. If you want to know whether a name is already used by a function or existing variable, type `which <name>`.

4.6 Be Patient and Embrace the Learning Experience

Debugging Matlab code can be an infuriating and humiliating experience that makes you want to quit science and sell flowers on the street. But don’t give up hope—it gets better. Embrace your mistakes and learn from them. Remember: no one is born a programmer. The difference between a good programmer and a bad programmer is that a good programmer spends years learning from his or her mistakes, and a bad programmer thinks that good programmers never make mistakes. I get annoyed when people think I am a good programmer because I can find and fix their bug in 30 s when they could not do it in 2 h. What they do *not* know is that I spent much more than 2 h finding and fixing that exact same bug in my own code, probably several times in the past. Eventually I learned to recognize what that bug is, where in the code it is likely to be found, and how to fix it.

Remember—time spent locating and fixing programming errors is *not* time lost; it is time invested.

4.7 Exercises

4.7.1 Exercises for Script A

1. Create a 4×8 matrix of randomly generated numbers.
2. Loop through all rows and columns, and test whether each element is greater than 0.5.
3. Report the results of the test along with the value of the matrix element and its row-column position. For example, your Matlab script should print The 3rd row and 8th column has a value of 0.42345 and is not bigger than 0.5.
4. Make sure to add exceptions to print out 1st, 2nd, and 3rd, instead of 1th, 2th, and 3th.
5. Put this code into a separate function that you can call from the command line with two inputs, corresponding to the number of rows and the number of columns of the matrix.

4.7.2 Exercises for Script B

6. Import and plot the picture of Amsterdam that comes with the online Matlab code.
7. On top of the picture, plot a thick red line from “Nieuwmarkt” (near the center of the picture) to “Station Amsterdam Centraal” (near the top of the picture).
8. Plot a magenta star over the Waterlooplein metro station (a bit South of Nieuwmarkt).
9. Find the maximum value on each color dimension (red, green, or blue) and plot a circle using that color. There may be more than one pixel with a maximum value; if so, pick one pixel at random.

4.7.3 Exercises for Script C

10. From the function you wrote for exercise 5, generate a 32×3 number matrix in which the three numbers in each row correspond to the row, column, and result of the test (1 for bigger than 0.5; 0 for smaller than 0.5).
11. Write this 32×3 matrix to a text file that contains this matrix along with appropriate variable labels in the first row. Make sure this file is tab-delimited and readable by a spreadsheet software such as Microsoft Excel or Open Office Calc.

5 Introduction to the Physiological Bases of EEG

It seems obligatory for a resource on EEG analyses to discuss the neurophysiological events that produce the EEG signal. There are many excellent resources available on this topic. It would be redundant to rewrite what is already explained in these resources, so this chapter instead provides a brief overview of the important points, along with references for reading more about each specific topic. The following references (among many others) are good starting points for deeper and approachable discussions of the neurophysiology and biophysics that underlie the EEG and MEG signal as well as the neurophysiological mechanisms that underlie oscillations (Buzsaki, Anastassiou, and Koch 2012; Nunez and Srinivasan 2006; Steriade 2006; Wang 2010).

5.1 Biophysical Events That Are Measurable with EEG

EEG reflects mainly the summation of excitatory and inhibitory postsynaptic potentials at the dendrites of ensembles of neurons with parallel geometric orientation. As neurotransmitters activate ion channels on the cell membrane, ions flow into and out of the neuron from and to the extracellular space. This change in potential generates electrical fields that surround the neuron. The electrical field generated by one neuron is too weak to be measured from an EEG electrode several centimeters away, but as neural activity becomes synchronous across hundreds, thousands, or tens of thousands of neurons, the electrical fields generated by individual neurons sum, and the resulting field becomes powerful enough to be measured from outside the head. It has been estimated that between 10,000 and 50,000 neurons, mainly in superficial cortical layers, dominate the EEG signal (Murakami and Okada 2006; Wang et al. 2005). Although the electrical conductivity differs among brain tissue, skull, and scalp, there is sufficient conductivity for the electrical fields to travel from the neural population that generated that field to the top layer of the scalp. Because the electrical conductivity of air is almost zero, a physical electrical bridge must be formed between the skin of the scalp

and the EEG electrodes. This is why scalp EEG electrodes require electroconductive gel, paste, or salt-water-soaked sponges. Magnetic fields are perpendicular to electrical fields and pass through the skull and scalp unimpeded. This is why MEG sensors must be close to the head but do not need to make direct contact with the scalp.

EEG cannot measure all neural events. In fact, most of the events in the brain are not measurable with EEG. If you find that assertion disappointing or disheartening, perhaps you will take some comfort in reading that the same could be said of all brain-imaging techniques, from single-unit recordings to local field potentials, from voltammetry to MR-based imaging. No single brain-imaging technique can record most of the events in the brain. Different techniques are well suited for measuring certain kinds of brain events. This is why you should choose the method that will best answer your research question, and this is why you should interpret results from one method in the context of what can be measured with that method.

EEG cannot measure individual molecular or synaptic events, nor can it isolate events that are produced by a specific neurotransmitter or neuromodulator. EEG cannot measure action potentials or local field potentials generated by small ensembles of a few or dozens of neurons. These spatially small-scale events produce either no electrical fields or weak electrical fields that can be measured only with invasive recordings. Some of these small-scale events may include cortical layer-specific oscillations (Kral et al. 2000; Scheffer-Teixeira et al. 2012; Sun and Dan 2009; Vreugdenhil, Bracci, and Jefferys 2005). If the small-scale events modulate meso- or macroscopic populations that produce large field potentials, EEG may measure the indirect effects of those small-scale events but not those events themselves.

Even some electrical fields that should be powerful enough to be measured from the scalp may not be measured with EEG. For example, field potential fluctuations on opposing sides of a sulcus would cancel if they had similar strengths, and thus not be measurable from the scalp (figure 5.1).

Deep brain sources such as the thalamus, basal ganglia, hippocampus, and brainstem are difficult but not impossible to measure from the scalp. There are two reasons for this difficulty. The first is that field strength decreases as an exponential function of distance, so even very powerful fields generated by deep brain structures have only a small impact on the voltage recorded from the scalp. To measure deep brain activity from the scalp, those deep brain sources must produce powerful fields, and there should be many trials for averaging. For example, studies on brainstem-generated potentials generally have thousands of trials to obtain sufficient signal-to-noise (Stone et al. 2009). This can be compared with the dozens of trials that are typically sufficient for cortex-generated potentials.

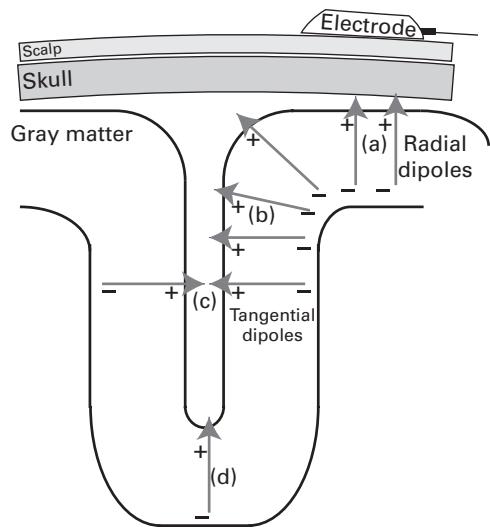


Figure 5.1

Illustration of dipoles in different orientations with respect to the skull. The dipoles illustrated in (a) will contribute the strongest signal to EEG, whereas the dipoles illustrated in (b) will contribute the strongest signal to MEG. The dipoles illustrated in (c) are unlikely to be measured because the dipoles on opposing sides of the sulcus produce electrical fields that are likely to cancel each other. The dipole illustrated in (d) will make a smaller contribution to EEG than dipole (a) because it is further away from the electrode. (This figure is inspired by figure 1 of Scherg 1990.)

The second reason why activity from deep brain structures is difficult to measure from the scalp is that populations of neurons in subcortical structures are not often arranged in a geometrically parallel orientation. This means that when there is synchronous population-level activity, the electrical fields generated by individual neurons are likely to cancel each other out at the macroscopic scale rather than summing and becoming powerful enough to be measurable from the scalp. If you would like to measure activity from deep brain structures using scalp EEG, you should use a task that is known to elicit activity in that brain region (e.g., as demonstrated with fMRI), have many trials to increase signal-to-noise ratio (hundreds or thousands, if possible), and, if you will perform source localization to provide support for the origin of the deep brain source, use many electrodes and subject-specific anatomical MRIs to increase spatial accuracy of the source reconstruction results.

Very slow fluctuations (<0.1 Hz) can be difficult but not impossible to measure with EEG. This is not due to the biophysics of the brain but rather to limitations of most modern EEG amplifiers and data acquisition systems. Most amplifiers have built-in high-pass filters that

attenuate very slow fluctuations because they may cause amplifier saturations. However, there are DC-coupled amplifiers that are suitable for recording fluctuations below 1 Hz. If you would like to measure very slow activity, check whether your amplifier is well suited for this and check whether there are high-pass filters that are applied during the recording.

Very fast fluctuations (>100 Hz) are also difficult but not impossible to measure. High-frequency activity generally has low power and thus is more difficult to distinguish from noise. MEG is better than EEG at recording high-frequency activity because the magnetic fields are unimpeded by the differing conductances across the brain, skull, and scalp. For EEG, activity above around 80 Hz should be carefully inspected to make sure it is not driven by noise spikes, EMG, eye movements, or other artifacts.

5.2 Neurobiological Mechanisms of Oscillations

An oscillation is a rhythmic alternation of states. Oscillations can occur in time or in space, and are commonly seen in physical and biological systems. In the brain, the term oscillation refers to rhythmic fluctuations in the excitability of neurons or populations of neurons. Neural oscillations are observed on many spatial and temporal scales (Varela et al. 2001) and have been linked to many neurobiological events ranging from long-term potentiation to conscious perception (Buzsaki 2006; Engel, Fries, and Singer 2001; Herrmann, Frund, and Lenz 2010; Kistler, van Hemmen, and De Zeeuw 2000; Klimesch et al. 2008; McBain and Kauer 2009). Oscillations can also be seen in unfiltered (“raw”) scalp EEG data. As discussed in section 2.6, one of the major advantages of conceptualizing EEG as comprising oscillations is that it provides a link to a large literature on *in vitro*, *in vivo*, and computational mechanisms of oscillations.

The neurobiological mechanisms that produce oscillations are fairly well understood (Buzsaki, Anastassiou, and Koch 2012; Wang 2010), although uncertainties remain in the extent to which different factors contribute to the signal recorded by EEG, in part due to the complexity of the models and the difference in spatial scale between individual neurons and scalp EEG (Coombes 2010; Deco et al. 2008; Whittingstall and Logothetis 2009). There are three basic physiological mechanisms that produce oscillations of hundreds to hundreds of thousands of neurons. One mechanism involves interactions between inhibitory (GABAergic) interneurons and excitatory pyramidal cells. When a population of pyramidal cells becomes active (e.g., from a volley of inputs from the thalamus or other cortical area), their excitation increases as they continue exciting each other. Interneurons within this population also become activated, and as the activity of the inhibitory interneurons increases, the

excitatory cells become inhibited. The activity of the interneurons then decreases, allowing excitation of the pyramidal cells to increase again. This shifting balance between states of excitation and inhibition produces oscillations. This process provides the basic skeleton of an oscillation; there are many additional factors that modulate the frequency, amplitude, and phase of the oscillations. This alternating balance between excitatory and inhibitory neurons is thought to be the mechanism that dominates the oscillations observed with EEG. Oscillations can also be produced by purely excitatory networks and by purely inhibitory networks. If you are interested in reading more about the mechanisms of neural oscillations and their involvement in neural computations, you could start with the references cited above, in the opening paragraph of this chapter, and the book *Rhythms of the Brain* (Buzsaki 2006).

5.3 Phase-Locked, Time-Locked, Task-Related

A distinction can be made among phase-locked, non-phase-locked, and background activity (figure 5.2). Phase-locked activity (also sometimes called “evoked”) is phase-aligned with the time = 0 event and will therefore be observed both in time-domain averaging (the ERP) and in time-frequency-domain averaging. Non-phase-locked activity (also sometimes called “induced”) is time-locked but not phase-locked to the time = 0 event. Simulated examples of non-phase-locked activity were shown in figures 2.1 and 2.2. Non-phase-locked activity can be observed in time-frequency-domain averaging but not in time-domain averaging (thus, it has no representation in the ERP). It is not entirely clear what mixture of physiological dynamics would result in phase-locked versus non-phase-locked activity (David, Kilner, and Friston 2006), although in general, non-phase-locked activity is taken as stronger evidence for the presence of oscillations (Donner and Siegel 2011; Gray and Singer 1989; Tallon-Baudry and Bertrand 1999). Methods to compute phase-locked versus non-phase-locked activity are discussed in chapter 20.

Phase-locked and non-phase-locked activities are task related in that their time and/or frequency characteristics change as a function of engagement in task events. Background activity, in contrast, does not change as a function of the task (Freeman 2004). Background activity is used in resting-state studies; however, in most cognitive electrophysiology studies, background activity provides little useful information. Furthermore, background activity can be distracting if it has a large amplitude or is present in the same frequency band as the task-related effects. Because the background activity is unrelated to task events, applying a baseline normalization will remove most or all of the background activity, thus allowing you to focus on task-related dynamics.

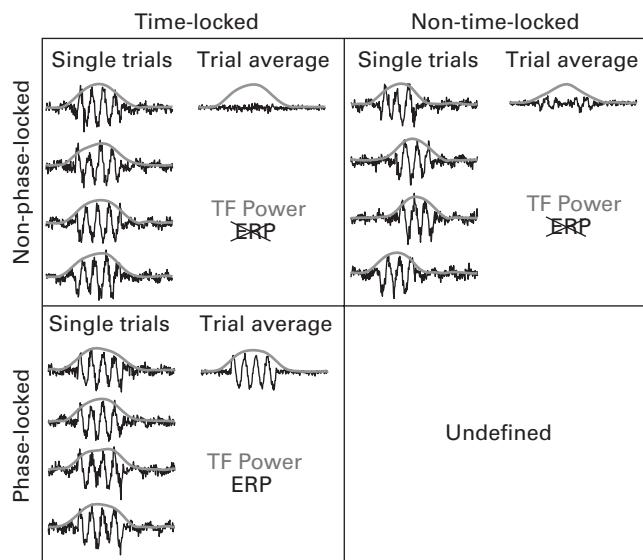


Figure 5.2

Illustration of whether time-frequency (TF) power and the ERP can measure phase-locked, non-phase-locked, time-locked, and non-time-locked activity. The left column of each cell shows four trials of simulated data, and the right column of each box shows the average of those four trials. Black lines show the raw time series, and gray lines show the time course of 10-Hz power. The ERP captures only phase-locked and time-locked activity. Time-frequency power can measure time-locked activity regardless of whether it is phase-locked or non-phase-locked. Activity that is not time-locked can be measured with time-frequency power, although the results will be smoothed and thus less temporally precise.

5.4 Neurophysiological Mechanisms of ERPs

Compared to oscillations, there is less empirical and theoretical work on the neurophysiological mechanisms that produce event-related potentials (ERPs)—that is, why there are positive and negative polarity peaks at somewhat regularly spaced intervals following experiment events. There are recent simulations that suggest mechanisms of ERPs through complex additive and nonlinear effects (David, Kilner, and Friston 2006) or rapid changes in frequencies (Burgess 2012), but there is less *in vitro* confirmation of ERP mechanisms compared to oscillation mechanisms. Other models about how ERPs can emerge from ongoing or oscillatory activity are described briefly below.

Additive This model proposes that the ERP reflects a signal that is elicited by an external stimulus such as a picture or a sound or by an internal event such as the initiation of a

manual response and is added to ongoing background oscillations. Because the oscillations are not related to the stimulus, they are attenuated in trial averaging. This model assumes that there is a distinction between the neurophysiological events that produce oscillations and the neurophysiological events that produce ERPs.

Phase reset This model proposes that ERPs result at least partially from a sudden alignment of the phases of ongoing oscillations (Makeig et al. 2002). That is, when a stimulus appears, the ongoing oscillation at a particular frequency band is reset to a specific phase value, which may reflect a return to a specific neural network configuration. This model may not account for all ERP components, particularly the later “cognitive” ERP components (Fell et al. 2004).

Amplitude asymmetry or baseline shift Although the electrical currents generated by neurons are polarity balanced, it is possible that outward-going currents are less detectable from the scalp (Mazaheri and Jensen 2008). This would produce an asymmetry in the oscillations measured by scalp EEG electrodes such that peaks and troughs are not equally distributed. It might also produce a baseline shift, which would also effectively produce asymmetries between peaks and troughs of oscillations (Nikulin et al. 2010). Changes in overall power could thus produce asymmetries in ongoing oscillations, which, when averaged over trials, might appear as a slow ERP (de Munck and Bijma 2010; Jensen, van Dijk, and Mazaheri 2010; Nikulin et al. 2007).

Having more mathematically precisely specified models might help resolve this debate (de Munck and Bijma 2010), but it seems that these different models are difficult to disentangle empirically at the level of the scalp. Indeed, models and simulations do not seem to be able to produce unambiguous evidence for the veracity of any one of these models at the expense of the other models (Krieg et al. 2011; Yeung et al. 2007). Ultimately, this debate may never be fully resolved until the neurophysiological mechanisms of ERPs are better understood, regardless of the mathematical precision of the competing hypotheses. One issue that complicates matters is that different ERP components may have different neural origins, and so general explanations about the mechanisms of all ERPs may be doomed to fail to capture all ERPs. If you would like to read more about this debate, the following references, in addition to those listed above, will help you to get started (Burgess 2012; David, Harrison, and Friston 2005; Makeig et al. 2004; Makinen, Tiitinen, and May 2005; Mazaheri and Jensen 2006; Penny et al. 2002; Sauseng et al. 2007; Shah et al. 2004).

5.5 Are Electrical Fields Causally Involved in Cognition?

Anyone reading this book probably hopes the answer is “yes.” However, at present there is limited causal evidence to state with certainty whether and how electrical fields are causally

involved in cognitive processes. Nonetheless, several lines of evidence suggest that electrical fields are causally involved in neural computation and information transfer.

One line of evidence comes from in vitro studies on the relationship between local field potential oscillations and synaptic events that are thought to underlie learning and memory. For example, long-term potentiation in the hippocampus, which is thought to be a basis of Hebbian learning and thus memory formation, occurs preferentially at specific phases of theta-band (4–8 Hz) oscillations (see Axmacher et al. 2006, for a review).

Another line of evidence comes from studies on the relationship between the timing of action potentials and the phase of the local field potential oscillation. In general, these studies show that the timing of many but not all neurons is constrained by the local field potential, such that neurons are more likely to emit an action potential during some phases of the local field potential oscillation. The synchronization between action potential timing and field potential phase has led to theories of phase coding (Lisman and Otmakhova 2001; Yamaguchi et al. 2007).

Other theories suggest that interregional oscillatory synchronization is a mechanism underlying the transmission of information across neural networks and that this synchronization-mediated connectivity is crucial for perceptual and cognitive processes (Akam and Kullmann 2012; Fries 2005; Singer 1993). The idea is that spatially disparate neural networks can most efficiently cooperate and transfer information when they are phase synchronized. Partly for this reason, phase-based synchronization methods are the most widely used approaches for studying connectivity in electrophysiology data and are discussed in chapter 26.

Recent studies on ephaptic coupling provide compelling evidence for a causal role of oscillations in brain function. Ephaptic coupling refers to interactions among neurons that occur via the transmission of ions, which are transmitted via an electrical field through the extracellular space. Ephaptic coupling of individual neurons is impeded by the relatively small magnitude of individual neuron electrical fields compared to extracellular distances and insulation properties of myelin, but it is likely to occur if neurons are densely packed (Bokil et al. 2001) or if the field potential is relatively powerful. Local field potentials generated from networks of neurons produce larger fields and have been shown to entrain spike timing via ephaptic coupling in the cortex (Anastassiou et al. 2011), particularly in the delta and theta frequency bands.

Two promising methodological approaches to determining whether oscillations are causally involved in brain computation are optogenetics in mice and rats and transcranial alternating-current stimulation in humans. Optogenetics involves shining a light at a

particular wavelength into the brain of a mouse. The mice (rats are also sometimes used) are from a genetic line that allows specific channels on specific types of neurons to be activated with millisecond precision when those channels are exposed to a light of a particular wavelength (Kravitz and Kreitzer 2011; LaLumiere 2011). By the use of optogenetics, oscillatory activity at a desired frequency can be exogenously introduced into a specific region of the brain. For example, optogenetics studies have demonstrated that gamma-band oscillations can be exogenously enhanced, and this enhancement facilitates signal transmission and noise suppression (Sohal et al. 2009).

In humans, transcranial alternating-current stimulation (TACS) involves passing an electrical current between two electrodes placed on the scalp. The frequency of stimulation can be specified, typically between 0.1 and 100 Hz. Thus, a frequency-band-specific electrical current can be exogenously introduced into the human brain. This is advantageous over transcranial magnetic stimulation, which can evoke oscillations but in a transient manner that depends on the brain region to which the stimulation is applied (Rosanova et al. 2009; Thut et al. 2011). Using TACS, researchers have demonstrated, for example, that 20-Hz stimulation over motor cortex increases motor evoked potential magnitude, whereas stimulation at a range of other frequencies does not (Feurra et al. 2011). Another study showed that stimulation at an individual subject's alpha peak increased alpha power (Zaele, Rach, and Herrmann 2010). Furthermore, large-scale frontal-parietal network entrainment by theta-band TACS facilitates cognitive processing (Polania et al. 2012). TACS is not often used in cognitive electrophysiology but may become an important tool for testing hypotheses about the role of specific oscillation frequencies in cognition (see also chapter 38).

5.6 What If Electrical Fields Are Not Causally Involved in Cognition?

Perhaps all of the empirical work on the causal role of neural oscillations is misguided or flawed. Perhaps all of the theories on the role of oscillations in cognition are wrong, and perhaps the computational models providing putative mechanisms of oscillations in cognition rely on incorrect assumptions. If electrical fields were not causally involved in cognition, would this be the end of cognitive electrophysiology?

Not at all. Electrical fields produced by neural populations are undeniably powerful and insightful indices of brain function. If they are not causally involved in cognition but rather are epiphenomenal curiosities that result from the mechanisms that truly underlie cognition, the use of field potential oscillations in the study of brain organization is still valid. Along a similar vein, it is not widely believed that the blood oxygenation-level-dependent (BOLD)

signal measured by fMRI is a causal mechanism of neural information processing, but it is widely believed that the BOLD response is a powerful and useful indirect index of brain function. The lack of causality does not stop thousands of scientists around the world from using the BOLD signal to understand brain functional organization. Similarly, conclusive evidence against a causal role of oscillations in cognition would not stop cognitive electrophysiologists from using EEG to make important discoveries about the functional organization of cognition and the brain.

6 Practicalities of EEG Measurement and Experiment Design

This chapter covers some details about how to design experiments so they are appropriate for time-frequency-based analyses. It also contains some suggestions for additional equipment that might facilitate your research. General information about setting up an EEG lab and additional advice for designing EEG and ERP experiments can be found in the books by Luck and Handy (Handy 2004; Luck 2005).

6.1 Designing Experiments: Discuss, Pilot, Discuss, Pilot

Do not underestimate the importance of good experiment design. Beautiful, clean data and clever, properly done analyses combined with a poorly designed task will give results that are difficult to interpret and may see rejections at peer-reviewed journals. On the other hand, a well-designed experiment is likely to produce meaningful and interpretable results that have implications for theories and may inspire new research, even if the data are noisy and only basic analyses are performed.

Before collecting data for your experiment, discuss your experiment design with colleagues and have them do the task (if there is a task) to give you feedback on its design. Pilot test your experiment behaviorally to make sure you can obtain the predicted behavioral effect before proceeding to collect brain data. You should then collect EEG data from one or two subjects and analyze those datasets completely, including the analyses corresponding to the main hypotheses. You may find flaws, suboptimal design features, extraneous conditions, or the need for additional conditions only after analyzing these pilot datasets. It is better to discover suboptimal and fixable design features during piloting than after collecting all of the data. If you are satisfied with the design and do not plan on changing any experiment parameters, you can keep those datasets for the final group-level analyses. If you change the task after piloting and cannot use the data you have already collected in the final results, then at least you had to throw out only one or two datasets rather than 20 datasets.

6.2 Event Markers

Experiment event markers, or triggers, are square-wave pulses that are sent from the stimulus-delivering computer to the EEG amplifier and are usually recorded as a separate channel in the raw data file (or sometimes, multiple channels). The amplitude of the pulse is used to encode specific events such as stimulus onset or response. Most systems have eight-pin cables that allow up to 255 unique codes (zero is the no-marker default). During data importing, the markers are converted to labeled time stamps that indicate when different events in the experiments occurred. An example may be seen in figure 6.1.

Event markers are critical because they are used to time-lock the EEG data offline. They are also used to reconstruct different conditions and responses. Therefore, it is better to have markers encode too much rather than too little information. It is easy to collapse multiple markers during the analyses but can be difficult to reconstruct specific trial events when they were not encoded in the data file. Combining data from a text file with the EEG data becomes even more difficult after preprocessing and trial rejection. Although most systems allow only 255 possible marker values, there is no practical limit to how much information can be encoded in the EEG data using 255 markers. For example, a pair of markers spaced 10 ms apart provides 65,025 possible unique markers (255 times 255).

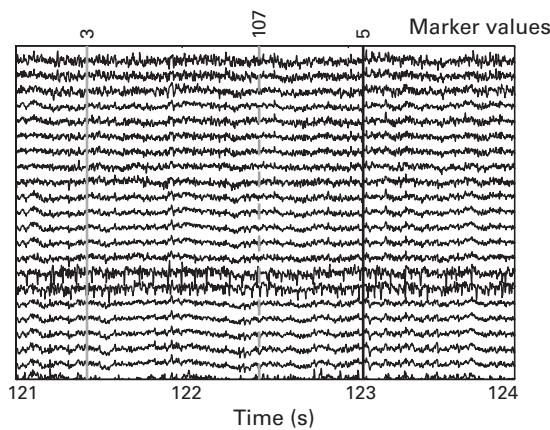


Figure 6.1

Example EEG data showing 3 s of data and three experiment markers. The experiment markers are represented as vertical lines, and the numbers on top of the vertical lines correspond to particular events. In this case the numbers 3 and 5 refer to two response buttons being pressed by the subject, and the number 107 corresponds to a particular stimulus. This picture was made using the eeglab lab function `eegplot`.

Many data acquisition systems do not allow multiple markers to be sent simultaneously, and when there are overlapping markers, only one is registered. Dropped markers can be detrimental because those trials may be lost. Simultaneous markers occur occasionally, for example, if the subject accidentally presses a button at the exact same time a stimulus marker is sent. This may happen once every few thousand trials and therefore is unlikely to have a significant negative impact on the analyses or on the results. If there are many simultaneous and blocked markers, there is a problem with your experiment, and you should investigate and resolve the issue before collecting additional data.

The temporal duration of each marker (this refers to the length of time in which the marker channel has a nonzero value) should be at least a few samples; a 0.5-ms marker might not be recorded if the sampling rate is 500 Hz. On the other hand, if the markers have a duration of 200 ms, they are likely to overlap with other markers. In general, 5 ms should be a sufficient duration.

There are many software packages that deliver stimuli to the subject and event markers to the EEG acquisition system. Some software programs are commercial (e.g., Presentation or E-Prime), and others are toolboxes for Matlab (e.g., Psychophysics toolbox or Cogent). Make sure the software you use can integrate with the computer hardware to maximize timing precision and stimulus delivery.

Because the timing and the values of the event markers are so critical to the experiment, if the timing is off or if the values are not properly encoded, there is little you can do with the data. Therefore, when using new software or hardware, it is a good idea to confirm the timing and event marker values. You can test the event marker values by sending codes 1–256 with 10 ms spaces between markers. You should be able to reconstruct these markers from the EEG data. If not, there is a problem somewhere, perhaps with the computer output port or the cable.

6.3 Intra- and Intertrial Timing

Partly because of some temporal smoothing introduced by time-frequency decomposition, and partly because time-frequency responses may linger for hundreds of milliseconds after an experiment event has ended, it is ideal to have experiment events within a trial separated by at least several hundred milliseconds (note that stimulus offsets also produce EEG perturbations). This will allow the brain response to one event to subside before the response to the next event begins. For some kinds of experiments this can be difficult to achieve, for example, if a button press quickly follows a stimulus, or if the stimuli need to appear closely temporally spaced to each other, as in the attentional blink task. If

the approximate duration of the electrophysiological response to a stimulus is not known, you could pilot the task using long interevent durations to see how long the electrophysiological response to each event lasts. How long the brain response to an event will last depends on the kind of task and on the cognitive process that is elicited. The time-frequency dynamics elicited by a simple and neutral visual stimulus, such as a Gabor patch, will likely decay quickly after stimulus offset. But the time-frequency dynamics elicited by a picture of a sick child in a poor country might take longer to subside because of lingering reactions to that picture.

Concerning the duration of time between the end of one trial and the start of the next trial (also called the *intertrial interval*), consider carefully what period of time you will use for baseline normalization of task-related data and what frequencies you want to analyze. Keep in mind that baseline time period for time-frequency decomposition should end before trial onset, for example –500 to –200 ms. This is different than ERPs, for which the baseline period typically ends at the time = 0 event. The reason for the earlier baseline period with time-frequency decomposition is that temporal filtering may cause some early poststimulus activity to “leak” into the prestimulus baseline period, and this temporal leakage can be worse at lower frequencies. You will learn more about why this is the case in chapters 10–14, and chapter 18 contains deeper discussions on baseline normalization and the choice of the pretrial baseline period. For most cognitive electrophysiology experiment designs, intertrial intervals of at least 1000 ms should be sufficient.

Intertrial intervals can be constant (e.g., 1000 ms on each trial) or variable (e.g., randomly selected on each trial from an interval between 800 ms and 1200 ms). There are advantages and disadvantages of each of these choices. On the one hand, if subjects can predict the temporal onset of the upcoming trial, they can prepare for the trial, and this preparation may be reflected in the EEG, for example, as changes in posterior alpha-band activity (Rohenkohl and Nobre 2011). On the other hand, with variable intertrial intervals, subjects may try to guess the next trial onset, which can introduce temporal expectations, surprise, and variable task preparation. These processes may vary in nonlinear ways as a function of the duration of the intertrial interval (e.g., as a hazard function). Indeed, both behavioral performance and EEG measures are affected by the duration and variability of intertrial intervals (Appelbaum et al. 2012; Egner, Ely, and Grinband 2010; Gonsalvez and Polich 2002). Subjects will nearly always generate temporal expectations about when the next trial will occur, and these expectations are likely to affect EEG and behavior. The question is whether to constrain these expectations in the experiment or leave them unconstrained. Note that the presence of pretrial activity does not hinder baseline normalization or the ability to examine condition differences in pretrial activity; this is discussed in chapter 18.

6.4 How Many Trials You Will Need

The number of trials you need for each condition depends on the signal-to-noise characteristics of the data (that is, how clean vs. how noisy the data are), how big the effect is, and the type of analysis you will perform. There is no magic number of trials that will guarantee good results or sufficient signal-to-noise ratios. That said, for many analyses a minimum of 50 trials per condition per subject is a reasonable number of trials that should lead to a sufficient level of signal-to-noise ratio, but this is not a strict rule. For example, the error-related negativity—an ERP component elicited by response errors—seems to be statistically robust and with reasonable reliability with as few as 14 trials (Larson et al. 2010). The reliabilities of other EEG dynamics such as band-specific power, measures of connectivity, and cross-frequency coupling are understudied or unknown (although this would be useful practical information). Further issues related to the number of trials necessary for analyses are presented in some analysis-specific chapters (for example, chapters 18 and 19).

6.5 How Many Electrodes You Will Need

The number of EEG electrodes or MEG sensors you need depends on what analyses you plan on performing and what inferences you plan on drawing. If you want to perform brain source reconstruction analyses and make inferences about brain localization, more electrodes (>100) are useful to increase the signal-to-noise ratio and accuracy of the spatial filters. On the other hand, if all you will do with the data is measure the P3 amplitude, you technically need only three electrodes (one placed over central parietal cortex to record the P3, one for a reference, and one for a ground). Unless you have compelling reasons otherwise (e.g., if you are testing a special clinical population that cannot tolerate many electrodes), use at least 64 electrodes if possible. This will allow you to apply spatial filters, perform connectivity analyses, and examine topographical distributions of the results. Sixty-four electrodes should be sufficient for nearly all analyses without significantly adding to data analysis time or hard disk data storage.

In addition to analysis considerations there are also practical considerations for the number of electrodes you will need, including preparation time and data storage. Most EEG electrode caps require electroconductive gel placed into each electrode to form a physical electroconductive bridge with the scalp. This procedure takes time and may take longer if the subject requires extra scalp preparation (asking your subjects to wash their hair and avoid using hairspray or gel will help provide a clean EEG signal). The preparation time increases with the number of electrodes. EEG caps with 64 gel-filled electrodes can be prepared in less

than 30 min by two trained workers, but 256 gel-filled electrodes may take considerably longer. There are also gel-free EEG caps that use sponges that are soaked in a salt-water solution; these sponge-based caps can decrease preparation time considerably.

Having more electrodes also has the practical implication of increasing data storage and processing time. Whether having more than 64 electrodes is worth the extra cost of data storage and computation time depends on the cost of data storage and backup and the computer systems available for analyses. Writing efficient Matlab code will help decrease computation time.

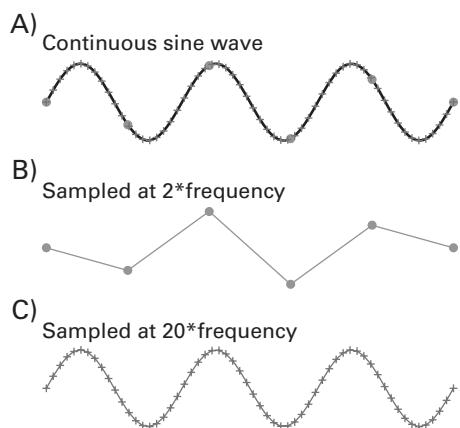
6.6 Which Sampling Rate to Use When Recording Data

Sampling rate refers to the number of times per second that data are acquired from all electrodes. This defines the temporal resolution of the data. The choice of sampling rate depends on several factors, including the kinds of analyses you plan on performing, the frequencies you plan on analyzing, and the available disk space and processor speed/type.

Technically, you need to sample at least twice the highest frequency of interest. This means that you need to use a sampling rate of at least 100 Hz if you want to test for 50-Hz activity. This is because of the Nyquist theorem, which, applied to time-frequency dynamics, states that only frequencies below half the sampling rate can be recovered (you will learn about why this is the case in chapter 11).

In practice, however, you should sample more than twice the highest frequency you will analyze. Having more data points per oscillation cycle increases signal-to-noise ratio and therefore allows for better estimation of high-frequency activity, as shown in figure 6.2. This is particularly the case if your analyses will involve using the phase-angle time series because the phase-angle time series is more susceptible to loss of temporal information compared to the power time series. Also keep in mind that there is an upper limit beyond which little additional information can be obtained. For example, sampling EEG data at 20,000 Hz is unlikely to provide any information not available in 1000-Hz-sampled data. In practice, sampling rates between 500 Hz and 2000 Hz are likely to be sufficient for all analyses. (The data provided with this book were originally recorded at 2048 Hz and were downsampled to 256 Hz to minimize download and analysis times.) Sampling rates higher than 2000 Hz are useful only in some situations where large artifacts must be precisely measured, such as during simultaneous EEG and fMRI recording.

Higher-sampling-rate data will take up more disk space, increase analysis time, and increase file read/write times. These issues may or may not be a concern depending on whether and

**Figure 6.2**

A continuous sine wave (panel A) and an illustration of the effect of subsampling that sine wave. Panel B shows that sampling the sine wave at twice its frequency (see gray dots along the sine wave in panel A) can reconstruct some features of the sine wave but fails to reconstruct the finer features, in particular the precise peak and trough times and the ongoing phases. Panel C shows that sampling at 20 times the frequency (see gray plus signs in panel A) can reconstruct the time-varying features of the sine wave with much higher accuracy.

how much you pay for data storage and backup and what types of computers you use for analyses. Less powerful computers running 32-bit Matlab may underperform or may crash due to insufficient memory if the sampling rate is too high. You can downsample the data after recording them, so it is better to record the data using a high sampling rate (e.g., 1000–2000 Hz) and then downsample the data prior to analyses if necessary. This is better than recording the data at 128 Hz and then having a poor signal-to-noise ratio for power and phase estimates in the beta and gamma frequency ranges.

From a convenience-of-analysis perspective, 1000 Hz is the optimal sampling rate. At 1000 Hz, there is a one-to-one conversion between time in milliseconds and time in samples. That is, 14 ms is also 14 samples. 500 and 2000 Hz are the next-most convenient sampling rates (14 ms is, respectively, 7 and 28 samples). After that, converting time to samples becomes difficult to do in your head (e.g., how many samples at 256 Hz fit into 350 ms?). This should not be a major guiding force behind your data acquisition because it is very easy to convert between time in samples and time in milliseconds. But given the choice between sampling rates of 1024 Hz and 1000 Hz, I would choose the latter.

6.7 Other Optional Equipment to Consider

Obviously, the EEG electrodes are the most important equipment components of an EEG lab. But there is more than just brain activity to consider; monitoring other behavioral and physiological events may improve the quality of your research. These improvements could come from removing trials that contain artifacts that are difficult to identify in the EEG data, or the improvements could come from providing additional insights into neurocognitive processes such as interactions between the brain and the body. Following is a list of possible add-ons that can be integrated with the EEG setup.

Response EMG or force grips There are several reasons why precise information about the motor actions of the subject during the task might be useful. Recording simultaneous EMG from the muscles used to indicate responses will allow you to examine cortical-muscular connectivity (Lattari et al. 2010) and will allow you to identify trials in which subjects twitched the muscles of the incorrect hand and then pressed the button of the correct hand (more on these “partial errors” in section 8.6). Force grips can also be used and provide largely consistent findings compared to EMG (Hoozemans and van Dieen 2005).

Eye tracker An eye tracker can be used for several purposes. It will facilitate preprocessing and cleaning the dataset by allowing you to remove trials in which the subject looked away from a fixation spot. It will allow you to use saccades and looking times as dependent measures, which can facilitate comparison with nonhuman primate research that uses saccades, and which are arguably underutilized dependent measures in cognitive neuroscience (Hannula et al. 2010). EEG data during the saccade will contain oculomotor artifacts that must be carefully removed or avoided before the data can be interpreted (Keren, Yuval-Greenberg, and Deouell 2010). Eye trackers can also be used to measure changes in pupil dilation, which can provide insight into cognitive processes (Granholm and Steinhauer 2004). Although the pupil has a relatively sluggish time course, there have been recent deconvolution methods that improve the temporal precision of the pupil response (Wierda et al. 2012). Finally, information from the eye tracker can be used to improve the quality of oculomotor artifact removal (Plochl, Ossandon, and Konig 2012).

Electrode localization equipment It is common in MEG research to record the precise location of the head with respect to the sensors, but this is infrequently done in EEG research. Most EEG research relies on standard templates of electrode positions. With careful EEG cap placement, the template locations are likely to be accurate to within 1–2 cm. Considering the spatial smoothing resulting from volume conduction, uncertainties of electrode positions within 2 cm are unlikely to have significant negative consequences for the results. However,

these uncertainties decrease the spatial precision of EEG, which may be detrimental when some spatial filters are applied such as the surface Laplacian or beamforming. If you plan on performing source localization analyses, and precise localization is important for the conclusions you hope to make from the data, using electrode localization equipment will be beneficial.

A comfortable chair for the subject to sit in This is more important than it may initially seem. If you have an uncomfortable chair, your subjects may shift around as they try to find a more comfortable position. This will cause movement artifacts in the EEG data. If the subjects are very uncomfortable, they may become distracted from the task. If the chair promotes bad posture, subjects might strain their back, shoulder, and neck muscles, which could introduce EMG artifacts in the EEG data.

A good response device Choose a response device that has good timing and is comfortable and intuitive to use. The response device should be easy to hold with an intuitive layout of the buttons. You do not want subjects looking down at their hands to figure out where the correct button is, and you do not want subjects to press the wrong button accidentally. Standard computer keyboards are poor choices for response devices: subjects may have to look down to confirm that they are pressing the correct button, and most keyboards have timing uncertainties on the order of several tens of milliseconds (some keyboards that are specially designed for gaming have millisecond-precision responses). The response buttons should not be too easy to press, otherwise subjects might not know whether their response was registered. On the other hand, if the buttons are too difficult to press, subjects may get tired. Some response devices make soft clicks when the buttons are pressed or when the circuit is closed. These clicks can be useful for subjects to know that their response was registered, although this may introduce confounds in auditory experiments.

II Preprocessing and Time-Domain Analyses

7 Preprocessing Steps Necessary and Useful for Advanced Data Analysis

7.1 What Is Preprocessing?

Preprocessing refers to any transformations or reorganizations that occur between collecting the data and analyzing the data. Some preprocessing steps merely organize the data to facilitate analyses without changing any of the data (e.g., extracting epochs from continuous data), other preprocessing steps involve removing bad or artifact-ridden data without changing clean data (e.g., removing bad electrodes or rejecting epochs with artifacts), and some preprocessing steps involve modifying otherwise clean data (e.g., applying temporal filters or spatial transformations).

This chapter is not a cookbook for preprocessing data but rather contains discussions of different preprocessing steps that you should consider before you start preprocessing your data. Many preprocessing choices and steps depend on details of the experiment design, the equipment used to collect the data, the analyses you plan on performing, and idiosyncratic protocols and preferences that you or your research group have developed.

It is a good idea to keep track of all the details of preprocessing for each subject, such as which trials were rejected, which electrodes were interpolated, and which independent components were removed from the data. This way, your results can be replicated from the raw data if necessary. Further, because some preprocessing choices may introduce biases to the data, you should use the same preprocessing procedures for all conditions to minimize the possibility that any biases that may have been introduced will spuriously cause condition differences.

7.2 The Balance between Signal and Noise

EEG data contain signal and noise. Appropriate preprocessing will attenuate the noise in the data. Unfortunately, however, signal and noise are often mixed together and may be difficult

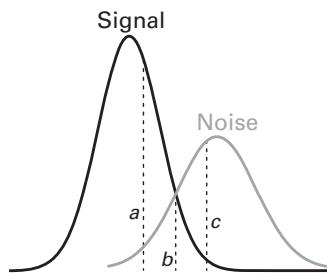


Figure 7.1

Theoretical depiction of signal and noise in EEG data as a signal detection problem. Different preprocessing strategies lead to different balances between the amount of noise versus signal that is retained in data. At one extreme (dashed vertical line a), the data contain nearly no noise, but this comes at the expense of losing signal. At the other extreme (dashed vertical line c), nearly all of the signal in the data is retained, but this comes at the expense of keeping noise in the data as well. There is no optimal position to take in this theoretical function; it depends on the kinds of analyses you are performing, how many trials you have, and how tolerant your analyses are to noise.

to disentangle completely. This leads to a trade-off between signal and noise (figure 7.1): removing a lot of putative noise is also likely to remove some signal, and leaving as much signal in the data as possible will likely mean that noise remains as well. In some cases noise can clearly and unambiguously be dissociated from signal, but in many other cases, the distinction is less clear. Amplifier saturations, for example, produce large spikes that are several orders of magnitude bigger than the neurally generated EEG. There is no ambiguity that this noise spike should be removed from the data.

In other cases, whether a feature of the EEG data is considered noise or signal depends on the researcher and the goal of the study. For example, some researchers apply a low-pass filter of 30 Hz because higher-frequency activity is likely to contain noise and muscle artifacts. Other researchers have built their careers focusing only on activity above 30 Hz. Another example is that many researchers spend their entire careers studying ERPs, whereas other researchers routinely subtract the ERP from the data because it is a potential artifact for some analyses that assume local stationarity. Thus, in some cases, one scientist's noise is another scientist's signal.

Your criterion for balancing signal and noise will influence each step of your preprocessing protocol, ranging from temporal filtering to trial rejection to independent components analysis-based rejection to oculomotor-based trial rejection. Where you set this criterion also depends on how much data you have and on how difficult the data are to acquire. If you have hundreds or thousands of trials, you have the luxury of setting stringent criteria to

remove many trials that may contain noise. On the other hand, if the data were difficult to obtain, for example intracranial recordings in humans, or if the experiment design precludes having many trials in some conditions, you might be willing to tolerate some noise in order to retain as much signal as possible.

Fortunately, time-frequency-based analyses tend to increase the signal-to-noise characteristics of the data, particularly for single-trial analyses and particularly for relatively low frequencies (below around 20 Hz). This is shown in section 18.12.

7.3 Creating Epochs

EEG data are recorded continuously and therefore are represented as a two-dimensional (2-D) matrix (time and electrodes). To facilitate investigating task-related changes in the EEG, the continuous data are cut into segments surrounding particular experiment events. Typically, this corresponds to the start of the trial. After epoching, the time-domain data are stored in a 3-D matrix (electrodes, time, and trials). Epoching is not necessary for resting-state datasets, although the continuous data can be segmented into nonoverlapping segments of a few seconds to facilitate analyses.

When epoching, you must decide which event to use for time locking—that is, what to call “time = 0.” In some experiment designs this is straightforward: the stimulus onset at each trial is time = 0. Other experiment designs might require some decision making at this step. Experiments in which stimulus-related activity and also response-related activity (which is temporally variable with respect to stimulus onset) are of interest, or experiments in which several stimuli are presented with variable delays, have multiple events that could be used as the time = 0 event. In these cases you could time-lock the data to the earliest event in each trial, which might be convenient for baseline normalization, or you could time-lock the data to the event on which you will focus most of the analyses. The time series data can be temporally shifted during analyses, so the decision of what to use as the time = 0 event will not necessarily limit your analyses. For example, if you time-lock the epochs to stimulus onset, it is possible during the analyses to re-time-lock the data to the response button press without going back to the raw data. If there are events in the trial that have variable timing with respect to the time = 0 event, make sure that all events are in the epoch and that there is sufficient buffer zone (discussed below) for edge artifacts at the end of the epoch.

The other decision you need to make when epoching—which may have significant consequences for the quality of the time-frequency decomposition, particularly for lower frequencies—is how much time to include before and after the time = 0 event. How long the epochs should be depends, of course, on the experiment—the epochs need to be at least as long as

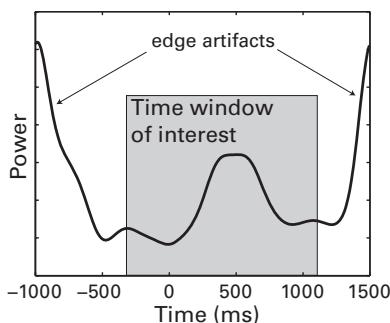


Figure 7.2

Edge artifacts resulting from discontinuous breaks in the time series between trials can contaminate the results if there are insufficient buffer zones to allow those edge artifacts to subside. In this case the edge artifacts are easily identifiable, and it is also clear that those artifacts subside before the time window of interest (gray area). In general, edge artifacts will contaminate up to three cycles of activity, but this could be less or more depending on the magnitude of the edges.

the duration of the trial. But the length of the epochs is also related to the kinds of analyses you want to perform. If you will compute only ERPs, the epochs can be as long as the time period you want to analyze plus a baseline period. For example, -200 ms until 800 ms relative to stimulus onset might suffice.

However, if you plan on performing time-frequency-based analyses, you should create longer epochs. The reason is to avoid contaminating your results with edge artifacts. Edge artifacts result from applying temporal filters to sharp edges such as a step function, and produce a high-amplitude broadband power artifact that can last hundreds of milliseconds. These artifacts will always be present when there are noncontinuous breaks, which happens at the first and last points of the EEG epochs (see figure 7.2 and chapter 14). At the first and last points of each epoch, the time series theoretically goes to zero and continues at zero for infinity. Thus, the transition between the outer “zeros” and the real data are sharp edges.

One way to avoid contaminating your results with edge artifacts is to have long epochs that allow the edge artifacts to subside before and after the experiment events in which you are interested. These extra “buffer zones” at the start and end of the trial will be discarded after the time-frequency decomposition. For example, if you intend to analyze the time period from 0 to 1000 ms, you might take epochs from -1500 to +2500 ms (thus, an extra 1500 ms on either end). How much of a buffer zone you need depends mainly on the frequencies that you intend to extract from the data. Edge artifacts typically last two or three cycles, although this depends on the magnitude of the edge. Thus, the lower the frequency

band that you will extract from the data, the more buffer zone you will need to be confident that the edge artifact has subsided. For example, at 0.5 Hz, the edge artifact can be several seconds long, whereas at 100 Hz, the edge artifact will be only a few tens of ms long. Fortunately, edge artifacts are easy to identify in time-frequency power results. If you are unsure how much of a buffer zone to include, you can analyze one subject and inspect the results carefully to see whether edge artifacts are contaminating the time period of interest. As a general rule of thumb three cycles at the lowest frequency you will analyze should be sufficient for a buffer zone (e.g., 1500 ms for 2-Hz activity).

One caveat to taking large epochs is that, depending on the amount of time between trials, you may have overlapping (and redundant) data in each epoch. This is not problematic for analyses because the overlapping data will be discarded, but it can introduce biases if you perform independent components analysis because the independent components analysis will be run on some time points more often than on other time points. If you have epochs with overlapping data, and if you plan on using independent components analysis to analyze data (that is, analyzing the component time courses instead of the electrode time courses), be sure not to expose the independent components analysis to the same data more than once.

Having sufficient buffer zones is necessary if you will perform time-frequency decomposition via complex Morlet wavelet convolution or the filter-Hilbert method. For time-frequency decomposition via the short-time FFT or multitaper, long buffer zones are not necessary because the time-frequency decomposition occurs on a more temporally local scale (this is explained in chapter 15). Nonetheless, it is still a good idea to take larger epochs than you would for ERP-only analyses.

If you are reanalyzing a dataset that has already been epoched and cannot be reepoching from the continuous data, and if you are concerned that the epochs are too short and the time-frequency results might be contaminated by edge artifacts, you can use a “reflection” approach, whereby the EEG data from each trial and electrode are reversed and put in the beginning and end of the trial. This makes the epoch three times as long, thus creating a buffer zone for edge artifacts. Remember to discard the reflected data after analyses, because the reflected data are backward versions of real data and cannot be interpreted as “pretrial” or “posttrial” activity. See figure 7.3 for an overview of how time series reflection works.

Reflection should be used as a measure of necessity, not as a substitute for including buffer zones in epoching. The reason is that time-frequency analyses involve some temporal smoothing, which you will read more about in chapters 10–14. The temporal smoothing may cause activity to leak out of the reflected data and into the time region of interest. Imagine, for example, that your epochs were cut from -100 to +1000 ms and there is low-frequency activity from around 200–600 ms. When the data are reflected, the low-frequency

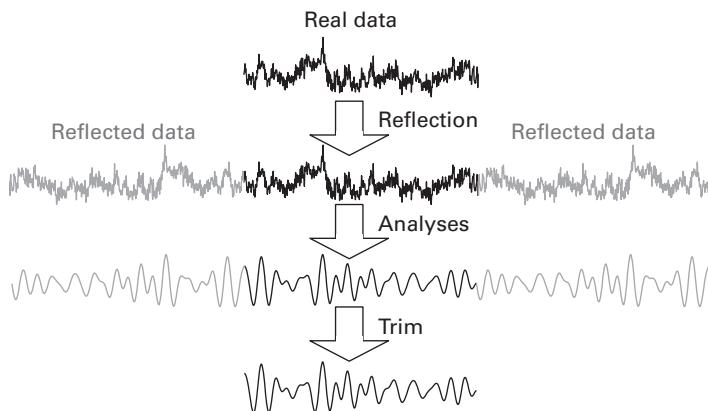


Figure 7.3

Data reflection procedure. Data are reversed in time, concatenated to both ends of the real-data time series, analyses are performed, and then the reflected data are trimmed. This procedure attenuates edge artifacts and can be useful if the epochs are cut too short for planned analyses.

activity toward the beginning of the trial may spuriously enhance the low-frequency activity in the nonreflected data. If this leaks into the prestimulus period, this could lead to a misinterpretation of the results. Thus, when epochs are created during preprocessing, long epochs with buffer zones are best, but reflecting short epochs is a better strategy than contamination due to edge artifacts.

You might think that instead of taking long epochs with buffer zones or reflecting data you can simply apply a taper to attenuate the data at the beginning and at the end of each trial (similar to the way time segments are tapered for the short-time FFT or multitaper; chapters 15–16). This is generally not a good idea. If you attenuate the data at the start of the epoch, you will decrease the time-frequency power during the baseline period, which will spuriously and inappropriately enhance the trial-related power in the task period. Even if the attenuation occurs before the start of the baseline period, because temporal filtering (done either through wavelet convolution or the filter-Hilbert method) involves setting the activity at each time point to be weighted sum of activity from previous time points, the filtered signal during the baseline period will still be artificially attenuated. Thus, tapering the entire epoch time period is not recommended.

7.4 Matching Trial Count across Conditions

It is ideal for all conditions to have the same number of trials. Depending on the analysis performed, differences in trial count may have little impact or may have a significant impact

on the results. In general, analyses based on phase are more sensitive to trial count than are analyses based on power or on the ERP. For phase-based analyses, a small number of trials will introduce a positive bias in the results, such that conditions with fewer trials are more likely to show larger effects. Power-based analyses may also have some positive bias because raw power values can only be positive, and thus noise is more likely to increase than decrease power. The ERP is not specifically negatively or positively biased by low trial count because the time-domain voltage values can take negative or positive values. However, low trial count decreases signal-to-noise ratio and therefore makes the ERP more sensitive to outliers or nonrepresentative data. For low trial count with ERPs, taking the mean amplitude in a time range is more robust to noise compared to peak times (Clayson, Baldwin, and Larson 2012; Luck 2005).

If there are small differences in trial count across conditions (e.g., 84 trials in condition A and 87 trials in condition B), you do not need to be concerned about potential biases introduced by condition differences in trial count. However, if there are large differences in trial count across conditions, particularly if one condition has fewer than 30 trials (e.g., 84 trials in condition A and 23 trials in condition B), you should be concerned. Whenever possible, try to design the experiment such that there are similar numbers of trials in each condition. However, this is not possible with all experiment designs; if your experiment necessarily entails an unbalanced trial count across trials, you can consider matching trial count across conditions.

Matching trial count generally involves identifying the condition with the fewest trials and selecting trials from other conditions such that all conditions end up with an equal number of trials. You will have to decide whether removing some good data is worth minimizing potentially spurious results due to trial count differences. There are three ways to match trial count across conditions.

One way to match trial count is to select the first N trials from each condition, where N is the number of trials in the smallest condition. This approach generally should not be taken because it biases some conditions toward having more trials earlier in the experiment when subjects are less tired, more motivated, and less practiced on the task.

A second way to match trial count is to select trials at random. This has the advantage that there is no bias in terms of when the trial occurred in the experiment but has the disadvantage that reanalyzing the same data multiple times may yield slightly different results. Thus, if you randomly select trials, it would be a good idea to store in a variable which trials were selected.

A third way to match trial count is to select trials based on some relevant behavioral or experiment variable such as reaction time. Here the idea is to select a subset of trials from all conditions such that the distributions of reaction times from the retained trials are similar

across conditions. This approach helps to equalize time on task and other general cognitive factors that contribute to reaction time (e.g., attention, engagement, motivation) and may thus help to rule out some alternative explanations of condition differences. The main disadvantage is that if there are reaction time differences between conditions, matching reaction times across conditions may bias trial selection from different regions of the reaction time trial distribution in different conditions. Selective sampling may be based on any other relevant behavioral measure such as saccade speed, pupil response, subjective difficulty rating, or the selective sampling may be based on a stimulus property such as luminance or location of a stimulus.

Trial count matching is not necessary across subjects unless you compare different groups of subjects such as patients versus controls or if you want to correlate the EEG results across subjects with a behavioral variable that might be related to trial count. Chapters 18, 19, and 26 show examples of the effects of trial count on time-frequency results.

If you match trial count across conditions, you should report this in the Methods section of the paper and, if relevant, also report the behavioral results before and after trial selection.

7.5 Filtering

Filtering data can help remove high-frequency artifacts and low-frequency drifts, and notch filters at 50 Hz or 60 Hz help attenuate electrical line noise. Most time-frequency decomposition methods (wavelet convolution, FFT, filter-Hilbert) involve applying a set of temporal filters to the data. Thus, filtering the data might not be necessary if you will focus on time-frequency dynamics. For example, there is no need to low-pass filter the time-domain data at 40 Hz if you will then perform time-frequency analyses to extract power from 2 to 20 Hz. Applying a high-pass filter at 0.1 or 0.5 Hz to the continuous data is useful and recommended to minimize slow drifts. High-pass filters should be applied only to continuous data and not to epoched data. This is because the edge artifact of a 0.5-Hz filter may last up to 6 s, which is probably longer than your epochs. Filtering ERPs is discussed in chapter 9, and details about how to design and implement bandpass filters in Matlab are presented in chapter 14.

7.6 Trial Rejection

Removing trials that contain artifacts prior to analyses is an important preprocessing step. It is also perhaps the preprocessing step most open to interpretation and idiosyncratic preferences. Some trials should unambiguously be rejected, and other trials will incite disagreement about whether they should be rejected. Some researchers argue for manual trial rejection

based on visual inspection, whereas others argue for automatic trial rejection so that the researcher never needs to look at the data. Automatic rejection procedures are fast and free of user bias, and will result in the same trials being rejected regardless of who handles the data. On the other hand, automatic procedures may use criteria that are appropriate for some subjects but not for others and may produce both Type I and Type II errors (that is, trials you think should be retained are rejected, and trials you think should be rejected are retained). I have tried several automatic algorithmic procedures and found them unsatisfactory and therefore prefer manual trial rejection based on visual inspection. But this is based on my experiences and my preferences for seeing raw data. Your decision of whether to use manual or automatic rejection should be based on your experiences, preferences, and data. Chapter 8 has more discussion about identifying and removing artifacts. If you will perform time-frequency decomposition of the data, you should be aware that sharp edges in the data are more detrimental to time-frequency decomposition than they are to ERPs. Relatively small sharp edges may be undetected by automatic algorithms but can have adverse effects on time-frequency decomposition results.

7.7 Spatial Filtering

There are three main reasons to apply spatial filters to your data. The first is to help localize a result. For example, if you want to confirm that an activity peak corresponds to left motor cortex, you can apply a surface Laplacian or fit a single dipole.

The second reason to apply spatial filters is to isolate a topographical feature of the data by filtering out low-spatial-frequency features. For example, if your task involves visual stimuli that require spatial attention, it may be difficult to separate visual processing in occipital cortex from attention-related processing in parietal cortex. In this case the surface Laplacian or distributed source imaging may help minimize the spatial overlap between occipital and parietal responses, thereby increasing confidence in functional/anatomical distinctions.

The third reason to apply spatial filters is as a preprocessing step for connectivity analyses. Some spatial filters such as the surface Laplacian or distributed source imaging help to minimize volume conduction—a possible artifact that may contaminate some kinds of connectivity analyses. Not all spatial filters (e.g., PCA) will address volume conduction artifacts, and not all connectivity measures require spatial filters.

The stage of preprocessing in which you should apply the spatial filter (e.g., before or after time-frequency decomposition) depends on what kind of filter you will use and what the goal of filtering is. Here are a few examples. If you are performing an ERP study on response preparation and want to see whether an ERP peak is consistent with a source in the motor cortex,

a single dipole can be fit to the grand-averaged ERP. If you are using the surface Laplacian to minimize volume conduction for connectivity analyses, the Laplacian should be applied to the single-trial time-domain data before time-frequency decomposition is performed and connectivity is computed. If you are performing a PCA on time-frequency power, the PCA can be performed either on single trials within a subject or on the trial-averaged power across subjects, depending on whether you are performing a subject-level or group-level PCA.

7.8 Referencing

Referencing is an issue only for EEG; MEG is a reference-free measurement. The issue is that the voltage values recorded from each electrode are relative to a voltage value recorded elsewhere. Where should this “elsewhere” electrode be? In theory, it could be anywhere, such as the subject’s toe (or the experimenter’s toe) or a wall in the experiment room. These are poor choices because the activity recorded from the reference electrode will be very different from the activity recorded from the electrodes on the scalp. Any activity present in the reference electrode will be reflected as activity in all other electrodes. Thus, choose your reference electrode(s) carefully and make sure the reference electrodes are properly placed and have a good, clean signal during subject preparation. Noise in the reference electrodes will turn into noise in the scalp electrodes. Note that the surface Laplacian is reference independent.

Averaged mastoids (the bone behind the ear) or earlobes are typical reference electrodes. (The sample data online are referenced to linked earlobes.) These are good choices because the reference electrodes are close to the other electrodes but record less brain activity. But mastoids and earlobes are not perfect references because they are close enough to the brain to measure neural activity, for example, from lateral temporal areas. For a large number of electrodes (>100), or if electrodes are placed on the neck and face, an average reference is often recommended. Referencing to one lateralized site is not recommended because this will introduce a lateralization bias in the data. The reference electrode should not be close to an electrode where you expect your main effects. For example, Cz is a poor choice of reference if your analyses involve response errors (which elicit maximal activity around electrode Cz or FCz). An example of the same data with different referencing schemes is shown in figure 7.4.

For intracranial recordings there are several options for a reference electrode, including an electrode in the bone if available, the average of all electrodes in each grid/probe, or another intracranial electrode. Using an external scalp electrode as a reference for intracranial data might be a suboptimal choice because any scalp activity measured by the reference electrode (possibly including muscle activity) will contaminate the intracranial recordings.

Because referencing is a linear transformation of the data, data can be re-referenced offline. Thus, the electrode that serves as the reference during recording is not very important. On

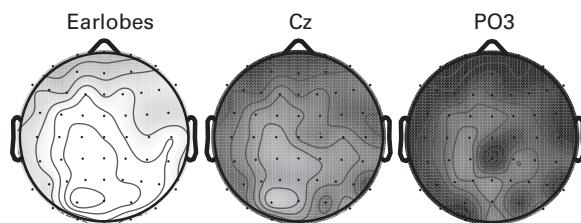


Figure 7.4

The effect of different reference electrodes on the same data. Earlobes refers to the average of electrodes placed on the two earlobes. In many situations, using one of the scalp electrodes as the reference is suboptimal.

the other hand, make sure you know which electrode is used as the online reference; otherwise you might think that you have a bad electrode that records only a flat line.

Some electrodes have a bipolar reference, meaning one electrode is measured relative to another. This is often the case for eye electrodes (one above the eye and one below the eye are referenced to each other, leaving only one signal; and one to the left of the eye and one to the right of the eye, again referenced relative to each other leaving only one signal), EMG, or EKG.

The choice of referencing is a widely discussed topic in EEG research with varied opinions, many of which conclude by noting that no reference is perfect, and the one you should use depends on a variety of factors including how many electrodes you have, where the electrodes are placed, what analyses will be performed, and what kinds of cognitive tasks you use, and thus, from which brain regions you hope to elicit activity (Dien 1998; Junghofer et al. 1999; Schiff 2005).

7.9 Interpolating Bad Electrodes

Interpolation is a process by which data from missing electrodes are estimated based on the activity and locations of other electrodes (see figure 7.5). Most interpolation algorithms use a weighted distance metric such as nearest-neighbor, linear, or spline. The more electrodes you have, the more accurate the estimation will be. Interpolation is used when there is a bad electrode that is either completely flat or that measures noise that is many orders of magnitude larger than real brain signal.

Ideally, you would not need to interpolate because all of your electrodes provide good data. In reality, however, interpolation is often necessary in a minority of cases, particularly with high-density recording systems or with old and overused electrodes. While recording the data during the experiment, it is a good idea to scan the data and look for bad electrodes.

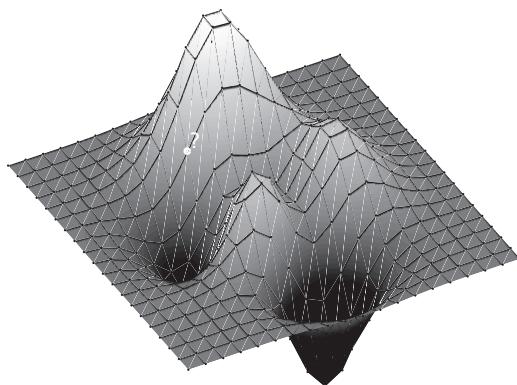


Figure 7.5

Topographical illustration of interpolation. Displayed is a smooth topographical landscape (analogous to a scalp-measured voltage) that is discretely sampled (black dots; analogous to electrodes). Interpolation involves estimating the activity at the white “electrode,” given the activities and distances of all other electrodes. This topography was generated with the Matlab peaks function.

If you notice a bad electrode during the experiment, pause the experiment if possible and try to fix or replace the bad electrode. It is preferable to reject a few trials but keep all of your electrodes than to interpolate a bad electrode for the duration of the experiment.

The problem with interpolated electrodes is that they do not provide unique data; they are a perfect weighted sum of the activity of other electrodes. This reduces the rank of the data matrix, which may lead to problems in analyses that require the matrix inverse (taking the pseudoinverse is usually an appropriate solution).

The alternative to interpolating bad electrodes is simply to ignore them by removing them from the data or by setting their activities to NaNs and using Matlab’s nanmean function during group-level analyses. Removing an electrode entirely from the dataset may cause some confusion when averaging across subjects because, for example, one subject will have 63 electrodes while another subject will have 64 electrodes.

Interpolation can be particularly important for some spatial filters such as the surface Laplacian or source reconstruction or if you will re-reference to the average of all electrodes: the activity of one bad electrode may contaminate the clean signal of other electrodes.

When deciding whether to interpolate an electrode, inspect the data carefully. In some cases there is a true brain signal recorded by the electrode but a lot of noise on top of that signal. If this is the case, try to filter out the noise without interpolating the electrode. One way to determine whether a noisy electrode contains brain signal is to apply a low-pass filter to the data at 30 Hz. If the low-frequency activity from that electrode looks similar to that of

the surrounding electrodes, it is likely that the unique signal from that electrode can be salvaged, and interpolation may not be necessary. If the low-frequency activity from that electrode looks completely different from activity of surrounding electrodes, particularly if the magnitude of the data is much smaller or much larger than that of surrounding electrodes, it is unlikely that any real brain signal was recorded by that electrode, and then it could be interpolated.

Most or all EEG analysis packages have routines for interpolating electrodes. The online Matlab code for this book also provides a spherical spline method, which is based on the same method used for the surface Laplacian, presented in chapter 22.

7.10 Start with Clean Data

There is no data analysis substitute for clean data. The fanciest and most sophisticated data analyses will give you bad results if you start with bad data. On the other hand, if you have very clean and high-signal-to-noise data, even modest analyses can provide compelling and insightful results.

Preprocessing can help turn good data into very good data, but no amount of preprocessing will turn low-quality and noisy data into very good data. Do not rush into a recording if you are unsatisfied with the data quality, and do not be afraid to pause the experiment if the data quality suddenly decreases during a recording and you think you can fix it. Explain to subjects (if they are human and awake) the importance of collecting clean data; they will try to help give you clean data. The next chapter discusses additional strategies for acquiring clean data.

8 EEG Artifacts: Their Detection, Influence, and Removal

If you ask EEG researchers to list the types of artifacts that contaminate EEG data, most will identify blinks, muscle movements, brief amplifier saturations, and line noise as the main artifacts that plague EEG data. These are true sources of artifacts, and they should be removed from the data before analyses are performed, but they are not the only artifacts that contaminate EEG data. There are also cognitive artifacts that should be identified and removed from your data. These artifacts might not have any noticeable correlate in the EEG, but they may contaminate your results nonetheless. This chapter provides an overview and discussion of the different types of artifacts in your data, how to identify them, and some ways to deal with them. It is not necessarily the case that you need to reject all the types of trials mentioned in this chapter. Some of these suggestions may be appropriate for your data, and others may be of no concern for your data. However, you should consider these points and think carefully about the procedures you use for identifying and removing artifacts in the data.

One important concept that is worth stressing here—and which newcomers to EEG often struggle with—is that EEG is *not* a noise-free measurement. There is no way to remove all of the noise from EEG, in part because it is unknown what all of the sources of noise are. Noise can also be difficult to identify. For example, high-frequency activity may look like noise at the single-trial level, and non-phase-locked activity may look like noise when one is computing an ERP. Certainly, you should try to minimize noise, but you should also realize and accept that your data will never be completely noise free. Most EEG data analyses are robust to the levels of noise that are typically observed in EEG data (after preprocessing), particularly with dozens or hundreds of trials per condition.

8.1 Removing Data Based on Independent Components Analysis

Independent components analysis is a source-separation technique that decomposes the EEG time series data into a set of components that attempt to identify independent sources of

variance in the data. It is often described in terms of audio recordings at a cocktail party: imagine there are many voices talking and many microphones placed around the room. By considering weighted combinations of the microphones' recordings, you could isolate the sound coming from individual voices. In this case the result of an independent components analysis is a set of weights for each microphone such that the weighted sum of all microphones best isolates the voice from one person. In the context of EEG the independent components analysis provides a set of weights for all electrodes such that each component is a weighted sum of activity at all electrodes, and the weights are designed to isolate sources of brain electrical signals (Jung et al. 2000; Makeig et al. 2004). Some aspects of independent components analysis are similar to principal components analysis, which you will learn more about in chapter 23.

Independent components analysis can be used either to clean EEG data by identifying components that isolate artifacts and then subtracting those components from the data, or as a data reduction technique by analyzing component time series instead of electrode time series. Nearly all of the time series analysis methods presented in this book can be applied to independent components in addition to electrode time series.

When independent components analysis is used as a preprocessing tool, components can be judged as containing artifacts based on their topographies, time courses, and frequency spectra. Components containing blink artifacts are probably the easiest to identify. They have an anterior distribution, and their time courses are largely flat with occasional very high-amplitude spikes, corresponding to the artifacts of the eye muscles as they close and open (figure 8.1). Other components might identify EMG or line noise.

Keep in mind that the decomposition of the data is based purely on statistical properties; an independent components analysis cannot determine which properties of the data to consider "signal" and which properties to consider "noise." In practice, components are likely to contain both signal and noise. This can make it difficult to know which components to remove from the data. In general, if the component time course shows a task-related ERP-looking deflection, it may contain signal. However, any non-phase-locked signal would also not be apparent in the ERP, so the absence of an ERP is not proof that the component contains no signal. You should be cautious about removing components that seem to contain signal. In general it is good to take a conservative approach and remove components from the data only if you are convinced that those components contain artifacts or noise and no or very little signal. In the best-case scenario you would remove only one component corresponding to blink artifacts.

The maximum number of components that can be isolated in the EEG data is the number of electrodes you have. If you have many electrodes (more than 100), it might be useful to

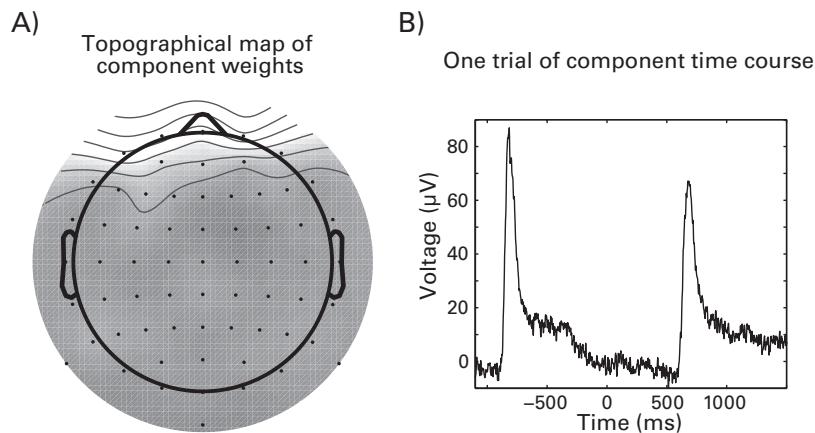


Figure 8.1

Example topographical map and example single-trial time course of an independent component that isolated a blink artifact. Panel A shows that the weights from this component are maximal at anterior electrodes, and panel B shows the time course of this component from one trial. You can see that on this trial, the subject blinked before and after the trial. The time course is a weighted sum of the activity of all electrodes, and the weights are defined by the results of the independent components analysis.

extract a smaller number of components than there are electrodes, in part because it will speed the analysis and in part because there are unlikely to be over 100 independent sources in the brain that are active and that can be statistically isolated at the level of scalp EEG.

Many EEG analysis programs offer utilities for performing independent components analysis. The Matlab toolbox eeglab is the toolbox that provides the most active development of independent components analysis.

8.2 Removing Trials because of Blinks

Should you reject trials that contain blinks? Blinks clearly introduce artifacts in the EEG data, but there are arguments for and against removing these trials.

Blink artifacts do not destroy the brain-generated EEG signal but, rather, linearly sum on top of the brain-generated EEG. There are several methods that successfully attenuate the oculomotor artifacts while sparing brain activity. The two most commonly used methods are independent components analysis (Jung et al. 2000) and regression-based techniques (Gratton, Coles, and Donchin 1983). Independent components analysis seems to work better for removing blinks and other oculomotor artifacts (Hoffmann and Falkenstein 2008; Plochl,

Ossandon, and Konig 2012). The success of these algorithms at removing oculomotor artifacts suggests that trials containing blinks should not be rejected.

However, there might be situations in which blink artifacts can be successfully removed but the trial should be rejected anyway. For example, relatively long blinks can be removed by independent components analysis, but if the subject closed his or her eyes for several hundred milliseconds, she or he may have been too tired on that trial to be focused on the task. Another example is tasks that involve briefly presented visual stimuli. A poorly timed blink might mean that the subject did not see the stimulus. Thus, depending on your experiment, trials with blinks could be rejected based on specific criteria such as the duration of the blink or the proximity of the blink to the visual stimulus presentation.

Some researchers specifically instruct subjects to inhibit blinking during the trial and to blink only during specified times or during the intertrial interval. This approach eliminates the need for correcting the data or rejecting trials, but it may introduce other problems. First, active suppression of blinking is a demanding task that relies on cortical oculomotor networks (Berman et al. 2012). Inhibiting blinks may thus introduce task-unrelated but stimulus-locked activity in frontoparietal oculomotor circuits. Second, inhibiting blinks can be cognitively demanding and distracting, particularly with long trials or a long experiment. Subjects may become so preoccupied with inhibiting their blinks that they sacrifice attention to the task. Third, if subjects are permitted to blink during the intertrial interval, then this time period might not be suitable as a baseline for normalization of time-frequency dynamics.

If you allow subjects to blink whenever they need to, they should avoid time-locking their blinks to experiment events such as button presses or stimulus offsets to facilitate a clean statistical isolation of blink artifacts from the experiment events to which those blinks are time-locked. For example, if subjects always blink whenever they press a button, it will be difficult for an independent components analysis to isolate the blink activity from the response-related activity (because these will not be independent components).

8.3 Removing Trials because of Oculomotor Activity

Blinks are not the only source of artifacts introduced by the eyes. There are also saccades and microsaccades that can contaminate EEG data, particularly at frontal and lateral frontal electrodes (or at posterior electrodes if the reference electrode is on the face). These artifacts can be minimized through experiment design by having visual stimuli at a central location on the experiment monitor, thus minimizing the need for subjects to look around. Having an easy-to-see fixation spot on the monitor at all times will also help subjects prevent eye

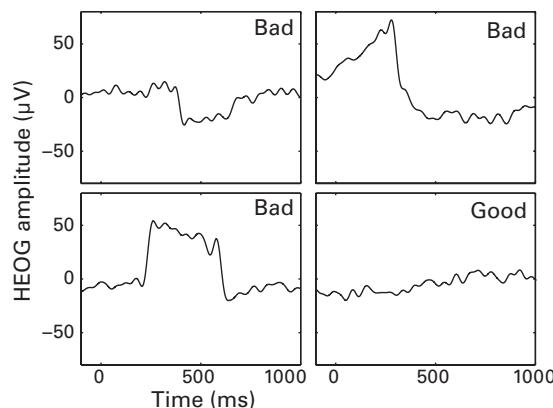


Figure 8.2

Horizontal EOG activity indicates eye movements after stimulus onset (time = 0). The first three panels are taken from trials that were removed prior to analyses.

movements. Finally, instructing subjects that small eye movements cause artifacts in the data will help minimize the frequency of saccades.

Having an eye tracker is ideal for trial rejection based on eye movements. If you do not have an eye tracker, you can use horizontal and vertical electrooculogram (EOG) electrodes. These electrodes are good at detecting relatively large eye movements, although they may lack the sensitivity of eye trackers to detect small eye movements or microsaccades. Figure 8.2 shows examples of horizontal EOG electrode activity. The panels marked “Bad” show trials that were removed prior to analyses.

Microsaccades can be more difficult to detect without an eye tracker because the small eye movements may be indistinguishable from noise in all but the cleanest EOG data. The potential impact of microsaccades on high-frequency EEG activity is a recent topic of discussion (Fries, Scheeringa, and Oostenveld 2008; Hassler, Barreto, and Gruber 2011; Yuval-Greenberg and Deouell 2009; Yuval-Greenberg et al. 2008). Microsaccades can be minimized by having small stimuli such that subjects do not need to saccade to see the entire stimulus, and by presenting the stimulus on the screen for only a short period of time (this may not be feasible in all experiments). There are algorithmic approaches for detecting and removing microsaccade artifacts from EEG data (Hassler, Barreto, and Gruber 2011; Keren, Yuval-Greenberg, and Deouell 2010; Nottage 2010).

How detrimental eye movements are to the data also depends on the referencing scheme and on your planned analyses. If you use a nose reference, eye movements might influence

the data more than if you use an earlobe reference. If your main hypotheses concern anterior frontal or lateral frontal regions, EOG artifacts are a serious concern; if your main hypotheses concern midcentral electrodes, EOG artifacts are less of a concern because they are less likely to be measured at these electrodes.

Some spatial filtering techniques can help isolate potential EOG artifacts. The surface Laplacian, for example, will help prevent the spread of oculomotor artifacts to activity at other electrodes. This is discussed further in chapter 22.

You might argue that artifacts from horizontal and vertical eye movements can, like blinks, be identified and removed with independent components analysis. Although this is likely to be the case if subjects produce enough eye movements, there is a larger concern than the oculomotor artifact itself: if the subjects are supposed to be fixating throughout the experiment, trials in which subjects broke fixation indicate that the subjects were not fully engaged in the task on that trial.

8.4 Removing Trials Based on EMG in EEG Channels

Trials with excessive EMG activity in the EEG channels should be removed. EMG is noticeable as bursts of 20- to 40-Hz activity, often has relatively large amplitude, and is typically maximal in electrodes around the face, neck, and ears. EMG bursts are deleterious for EEG data if you plan on analyzing activity above 15 Hz. EMG bursts also indicate that the subject moved, sneezed, coughed, or giggled during that trial. Thus, even if you will not examine frequencies above around 15 Hz, you might still want to remove trials with large EMG bursts because the subject may have been engaged in activities other than the task during that trial. A burst of EMG activity during a trial is shown in figure 8.3 (this trial was removed prior to analyses).

Some subjects show low-amplitude EMG activity continuously throughout the experiment, even after you instruct them to relax their face, neck, and shoulders. What to do with these datasets depends in part on your experiment. If you have specific hypotheses about beta-band activity, you might be unwilling to tolerate a dataset with continuous EMG and therefore will exclude this dataset from group analyses. Sometimes the EMG activity is well localized to lateral temporal or anterior electrodes; if you plan on testing beta-band activity over motor electrodes that show little EMG artifact, you may be willing to keep that dataset in the group analyses.

As you will learn in chapter 18, baseline normalization of time-frequency power is based on relative changes in power before versus after trial onset. This means that if the EMG activity has constant amplitude before and after trial onset, that EMG activity will be removed

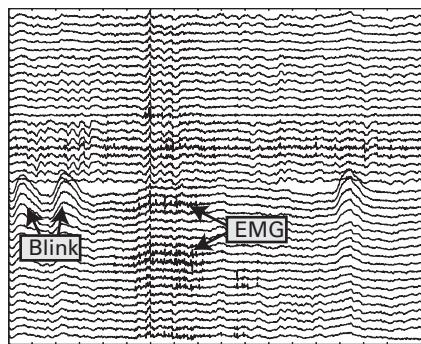


Figure 8.3

One trial of EEG data showing EMG activity on several electrodes. EMG activity is recognizable as brief bursts of high-frequency activity. This snippet of EEG data also shows blink artifacts.

during baseline normalization. Furthermore, if the continuous EMG activity is present in all conditions, it will subtract out during condition comparisons.

Sometimes continuous EMG activity can be well isolated with independent components analysis. If you plan on subtracting an EMG-containing component from your data, make sure the component does not also contain a brain-generated signal.

Note that if you create long epochs to have a buffer zone for edge artifacts (as discussed in section 7.3), it is not necessary to reject trials based on artifacts present in the buffer zone because these data will be discarded after time-frequency decomposition.

8.5 Removing Trials Based on Task Performance

So far, this chapter has discussed artifacts that contaminate the EEG signal or that are visually recognizable in the EEG data. There are other artifacts that might have little or no visually recognizable correlate in the EEG data but that may nonetheless have negative consequences for the results. These artifacts can be called *cognitive noise* or *cognitive artifacts*, and you should consider removing these trials from your analyses.

If your task involves responses that can be accurate or not, you will likely want to remove or at least separate error trials. You might also want to remove posterror trials if you are concerned about posterror behavior changes that might influence task performance and associated brain activity. You should also consider removing trials in which subjects do not make a response if they were instructed to do so, trials with more responses than were required, trials with very fast reaction times (for a finger button press, less than 200 ms), or trials with very

slow reaction times (for example, reactions times that are slower than three standard deviations from each subject's median reaction time). In these trials it is likely that the subject was not fully engaged in the task.

Depending on your task there might be additional justifications for removing trials that are suspected to contain cognitive noise. For example, if you have rest breaks that last several tens of seconds, subjects might not be fully reengaged in the task on the first trial after the break. Another example is if subjects perform a few tens of trials with one set of instructions and then switch to another set of instructions; the first trial after each switch will involve a cognitive set shift and a switch cost. These trials generally have longer reaction times and lower accuracy rates. If you are not studying these switch costs, you might want to exclude switch trials from the data.

8.6 Removing Trials Based on Response Hand EMG

If your experiment and lab setup allow, consider recording EMG from the muscles subjects use to indicate the response (e.g., from the fingers used to press buttons). This will allow you to identify partial error trials in your data. Partial errors occur when the subject twitches the muscle of the incorrect response, although he or she pressed the correct button. Correct trials that contain partial errors elicit patterns of brain activity that look more like errors than they do like correct responses (Cohen and van Gaal 2012; Coles et al. 1985; Coles, Scheffers, and Fournier 1995). In other words, correct trials containing partial errors contaminate “pure” correct trials. Partial errors can also be identified using force grip response devices. Figure 8.4 shows an example of a partial-error EMG response.

Here is the method we use to identify partial errors (Cohen and van Gaal 2012); this algorithm may need adjustment depending on the type and quality of your data. First, the Z-transform of the derivative of the EMG signal from each hand is taken and then rectified (that is, taking the absolute value); this eliminates hand- and subject-specific differences in impedance and signal amplitude. A partial error is identified when this Z-derivative signal of the hand not used to make the response exceeds two standard deviations (that is, a Z-score of 2) in the time between stimulus onset and the actual button press. The magnitude of this EMG peak must be more than two times larger than the largest EMG peak from -300 ms to stimulus onset (this eliminates trials in which noisy EMG produces apparent partial errors). Regardless of how you identify partial errors, you should visually inspect trials that are and are not identified as partial errors to confirm that the algorithm works well for your data.

Recording EMG from the thumb muscle tends to be easier than EMG from the other fingers. This is partly because the thumb muscles are bigger and therefore easier to find. You can

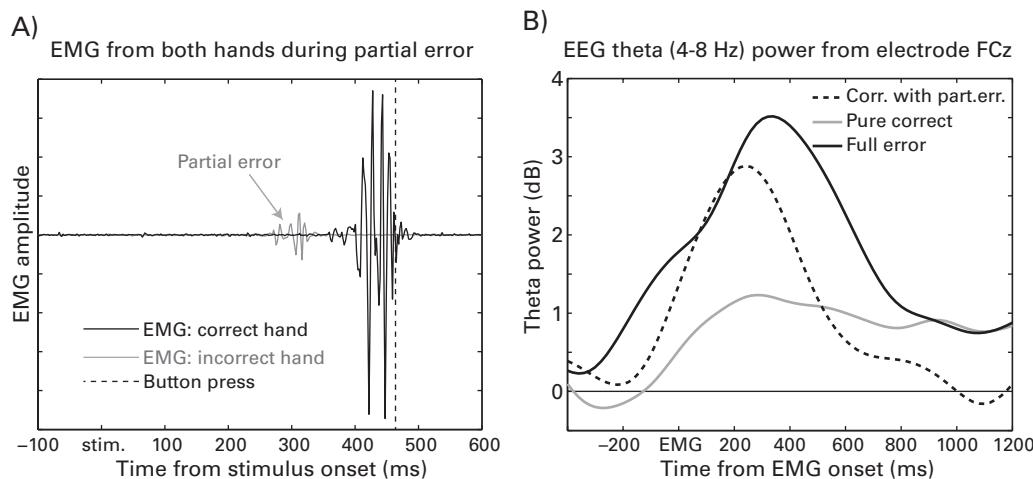


Figure 8.4

Partial errors can be detected with EMG recordings and are useful to identify correct trials with error-like brain responses. Panel A shows example EMGs from right and left thumbs showing a partial error—a muscle twitch of the incorrect hand although only the correct button was pressed in this trial. Correct trials that contain partial errors elicit error-like brain processes, as can be seen in panel B. Theta-band power from electrode FCz is time-locked to EMG onset; each line is the average of all trials from one subject. Partial errors can be identified and removed from the dataset or separately analyzed.

also improve the quality of the EMG data by having subjects use response buttons that take some effort to press. This will require more muscle engagement, which will thus produce bigger and cleaner EMG responses. In contrast, a mouse button requires little physical exertion to press and will produce a small EMG activation.

8.7 Train Subjects to Minimize Artifacts

It is safe to assume that your human volunteer subjects want to provide you clean, high-quality data, particularly if you stress the importance of their data to the overall research. If the subjects are causing artifacts in the EEG data, most likely they are not aware that they are causing artifacts. Many EEG artifacts can therefore be minimized with proper training. After setting up the EEG cap, show the subject her EEG data in real time on a computer monitor that she can see. Explain that EEG data contain both brain activity and noise from muscles. You can have her blink, clench her jaw, tense her neck/shoulder muscles, talk, smile, wiggle her ears, and so on. When subjects know what kinds of behaviors produce EEG artifacts, they can minimize those behaviors during the task.

8.8 Minimize Artifacts during Data Collection

As mentioned in the previous chapter, there is no preprocessing or analysis trick that will turn low-quality, noisy data into beautiful results. The best way to get your preprocessing protocol to turn good data into very good data is to start with good data. Training subjects before the task is part of ensuring that you will have good data.

While running the experiment, keep an eye on the real-time EEG data as they are being acquired. At least once every 30 s you should quickly glance at the EEG data and check that they look OK. If you notice artifacts in the data, first check whether there is a rest period (large artifacts during rest breaks are usually OK because these periods are typically not analyzed). If the artifacts are present during the task, consider pausing the experiment at the next possible opportunity to determine what is causing the artifacts. If the artifacts are coming from the subject being tired or uncomfortable, turning the lights on and talking to the subject for a minute before resuming the task might help to reengage the subject's attention and reduce the source of the artifacts.

9 Overview of Time-Domain EEG Analyses

9.1 Event-Related Potentials

The logic underlying the computation of an ERP is straightforward: each trial contains signal and noise; the signal is similar on each trial, whereas the noise fluctuates across trials. Because the noise fluctuations are randomly distributed around zero, noise cancels out when many trials are averaged, thus leaving the signal (the ERP). To create an ERP, simply align the time-domain EEG to the time = 0 event (this was probably already done during preprocessing) and average across trials at each time point.

Figure 9.1A shows a few randomly selected trials from one electrode. Figure 9.1B shows all trials from this electrode and the average of all trials superimposed. The average is considerably smaller in magnitude than the individual trials. This is because all non-phase-locked activity, which tends to have larger amplitude, is subtracted out during averaging (see also figures 2.1 and 5.2). Figure 9.1C shows only the ERP with a tighter y -axis limit, which accentuates the smaller fluctuations in the ERP.

Thus, the mathematical basis of an ERP is simple: sum the voltage at each time point over trials and then divide by the number of trials. If you will use ERPs mainly as a data quality inspection tool, that is about all you need to know. If you plan on using ERPs to make inferences about cognitive processes, you should familiarize yourself with issues related to component overlap, component quantification, appropriate interpretation, and statistical procedures. See the books by Luck and Handy (Handy 2004; Luck 2005) for more discussions of these issues.

9.2 Filtering ERPs

Keep in mind that time-domain signal averaging over trials is itself a low-pass filter. This is because non-phase-locked activity does not survive time-domain averaging, and frequencies

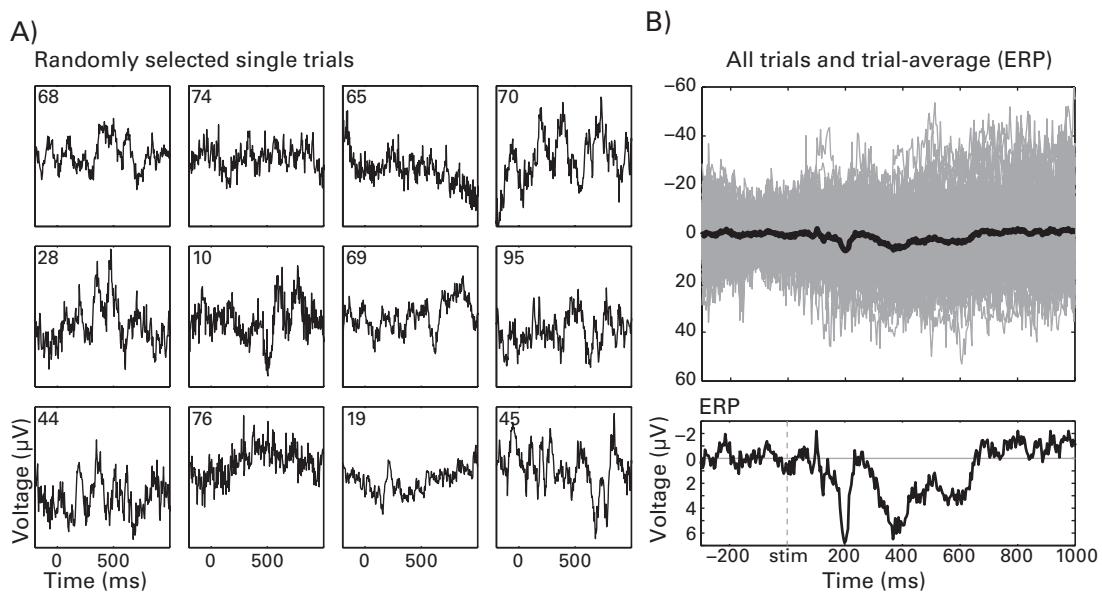


Figure 9.1

Panel A shows single-trial EEG traces from 12 randomly selected trials (number inside plot indicates trial number). Data are from electrode FCz. Panel B shows 99 single trials in gray and their average—the ERP—in black. Panel C shows the same ERP with focused y -axis scaling.

above around 15 Hz tend to be non-phase-locked. Averaging ERPs across multiple subjects provides further low-pass filtering because brief neural events are likely to have some temporal jitter across subjects, and thus the average will be smoothed in time. Nonetheless, it is common to apply additional low-pass filters when computing ERPs. Filtering the ERP minimizes residual high-frequency fluctuations, makes the ERPs look smoother, and facilitates peak-based component quantification by reducing the possibility that the peak is a noise spike or an otherwise nonrepresentative outlier. Filtering ERPs is not always necessary, particularly if there are many trials or if you are focusing on later ERP components that tend to be extended in time.

Filtering ERPs is a topic of some debate. Despite the advantages listed in the previous paragraph, poorly designed filters can introduce ripples in the time domain. These ripples (also called ringing artifacts) result from having poorly designed filters, such as filters with very narrow transition zones. The reason these artifacts occur is similar to the reason that edge artifacts occur (figure 7.2). You will learn more about this in chapter 14. The danger of these artifacts when interpreting ERPs is that large ripples in the time domain can appear to

be oscillations. Thus, if you have a poorly constructed filter, you might interpret your results to indicate that there are brain oscillations during your task, when in fact you are interpreting an artifact of the filter. In practice, this situation is easily avoided by constructing filters with gentle transition zones. Ripple artifacts are not the only concern, however. Some filter settings (again, settings that are generally associated with poor filter construction) may introduce systematic biases in ERP components, such as the use of forward-only or causal filters (Acunzo, Mackenzie, and van Rossum 2012; Rousselet 2012). Again, chapter 14 discusses proper filter construction and also provides qualitative and quantitative measures of whether a filter is properly constructed. A final concern of filtering ERPs is that applying the low-pass filter reduces temporal precision because the voltage value at each time point becomes a weighted average of voltage values from previous and subsequent time points. Thus, one of the main advantages of ERPs (its high temporal precision) is reduced. The lower the cutoff frequency of the filter, the more temporal precision is lost. Because of these considerations, you should carefully consider whether a filter is necessary for your ERPs. Figure 9.2 illustrates the effect of different filters on a single-subject trial-averaged ERP.

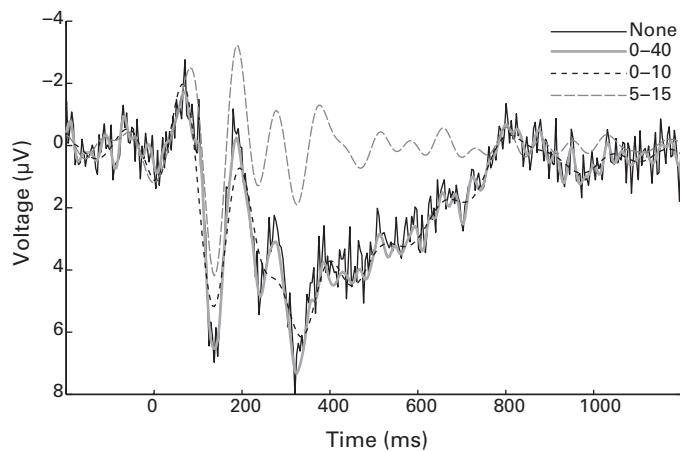


Figure 9.2

An ERP from electrode P7 with no filtering (black line) and with different filter settings (numbers in the legend indicate lower and upper frequency bounds in hertz). Note that some filter settings can have dramatic effects on the interpretation of specific ERP features. For example, the 5–15 Hz filter seems to have accentuated the first negative-going peak at around 100 ms and removed the later P3-type component, and the 0–10 Hz filter removed the negative-going peak at around 280 ms. The wide-band 0–40 Hz filter had the least effect on the larger ERP fluctuations while removing the high-frequency fluctuations. This plot is an illustration of why you should carefully consider the frequency range of the filter used for interpreting ERPs, particularly if you use a narrow frequency range.

ERPs are often filtered using a frequency cutoff of around 20 or 30 Hz, and occasionally the frequency cutoffs are as low as 5 or 10 Hz. Note that applying a filter to single trials and then computing the ERP is identical to computing the ERP and then applying a filter (demonstrated in chapter 14). Filtering the data before trial averaging is thus not necessary, although it can be useful for data inspection or independent components analysis.

9.3 Butterfly Plots and Global Field Power/Topographical Variance Plots

A butterfly plot shows the ERP from all electrodes overlaid in the same figure (figure 9.3A). If the data are referenced to the global average, the butterfly plot will show ERPs symmetric about the zero horizontal line. Butterfly plots are useful for detecting bad or noisy electrodes.

The global field power is the standard deviation of activity over all electrodes. Global field power can be obtained simply by computing the standard deviation over all electrodes at each point in time, and is best interpreted when using the average reference (Murray, Brunet, and Michel 2008). Thus, during periods of cortical quiescence, the global field power will

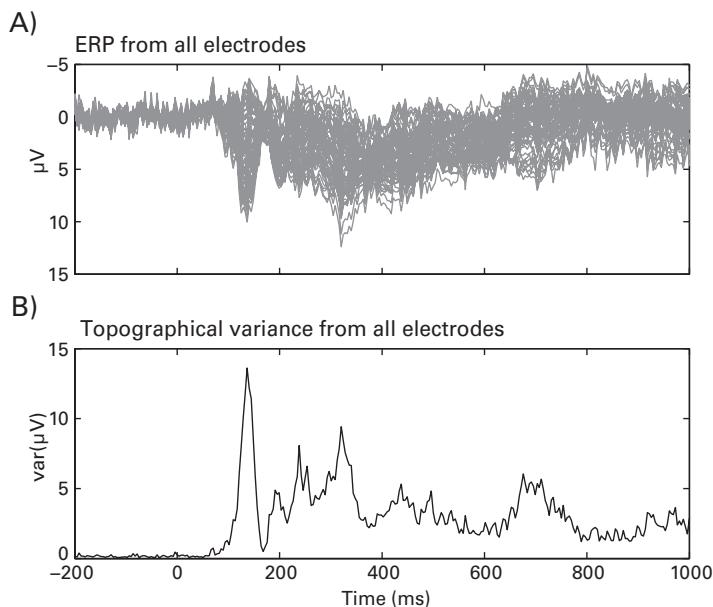


Figure 9.3

An example butterfly plot (panel A) and a topographical variance plot (panel B). Although they lack spatial information, these plots are useful for data inspection and provide an overview of the time periods with cortically diverse events, including approximately 180 ms, 220 ms, 320 ms, and 700 ms.

be small. In contrast, as different brain regions become active, global field power increases. Note that global field power is not redundant with average amplitude—a large-amplitude, spatially broad component such as a P3 may have relatively low global field power if most electrodes have the same activity levels.

Topographical variance accentuates the global field power and thus facilitates visual inspection (figure 9.3B). Butterfly plots and topographical variance plots are useful as data quality indices and to confirm the timing of the representations of task events in the data.

9.4 The Flicker Effect

The flicker effect in EEG research refers to entrainment of brain activity to a rhythmic extrinsic driving factor (Regan 1977; Vialatte et al. 2010). For example, if you look at a strobe light that flickers at 20 Hz, there will be rhythmic activity in your visual cortex at 20 Hz, phase-locked to each light flash from the strobe light. This can be measured as a narrow-band increase in power or phase alignment at the frequency of the visual flicker, or it can be measured in the ERP. This effect is also referred to as steady-state evoked potential, frequency tagging, SSVEP (steady-state visual evoked potential), SSAEP (auditory evoked potential), or something similar.

The flicker effect is arguably an underutilized tool in cognitive electrophysiology. The main benefit of the flicker effect is that it allows you to “tag” the processing of a specific stimulus. For example, by having multiple stimuli that flicker at different frequencies on the screen at the same time, you can independently track processing of each tagged stimulus, measured by changes in the EEG power at the stimulus frequency. This is a powerful technique because the spatial precision and accuracy of EEG are normally much too low to examine the representations of specific stimuli. The flicker effect allows you to “fake” a very high spatial resolution, as if you could isolate populations of neurons that respond to one stimulus. One limitation of the flicker effect is its poor temporal precision: it takes several hundred milliseconds for the flicker effect to stabilize, and longer periods of time provide an increased signal-to-noise ratio. Thus, this approach is ideal for stimuli that can remain on screen for several seconds.

One may wonder in which brain regions the flicker will entrain neural network activity. Evidence suggests that stimulus flicker can entrain brain activity beyond the stimulated sensory area (e.g., early visual cortex in the case of visual flicker). For example, theta-band visual flicker increases hemodynamic activity in medial prefrontal regions (Srinivasan et al. 2007a). EEG and MEG studies suggest that spatially widespread networks, including frontal and parietal areas, can exhibit a flicker effect that, in some cases, appears to depend on the flickering

frequency (Ding, Sperling, and Srinivasan 2006; Srinivasan, Bibi, and Nunez 2006). If neural entrainment to a rhythmic exogenous stimulus extends beyond primary sensory areas, it is possible that the flicker may facilitate cognitive processes that utilize those frequencies. Indeed, there is some evidence that performance on cognitive tasks can be modulated by stimulus flicker frequency (Thut, Schyns, and Gross 2011; Williams, Ramaswamy, and Ouhahaj 2006) and that entrainment can modulate processing of stimuli that are presented in phase compared to stimuli presented out of phase with the entrainment stream (Mathewson et al. 2012). Furthermore, the brain may retain a memory trace of the frequency at which items were tagged (Wimber et al. 2012).

Stimulus flicker frequencies up to 100 Hz can evoke a flicker effect (Herrmann 2001), although lower frequencies generally elicit a stronger flicker effect (that is, larger in magnitude). Other stimulus factors such as size, luminance, and contrast will impact the signal-to-noise ratio of the flicker effect.

It remains debated whether the stimulus flicker effect reflects a neural oscillation or repeated ERPs to each stimulus presentation (Capilla et al. 2011; Moratti et al. 2007; Thut, Schyns, and Gross 2011). This debate has implications for the interpretation of the flicker effect in terms of neurophysiological mechanisms of oscillations but does not impact its usefulness as a tool for isolating the processing of specific stimuli.

There are no special analysis techniques for studying the flicker effect; you can examine the ERP or perform frequency decomposition or time-frequency decomposition as described throughout this book. In general, a peak in the frequency domain at the flicker frequency should be readily observed in a spectral plot, and the magnitude of this frequency peak can be compared to the same frequency before the flickering stimulus began, or it can be compared to the power of neighboring frequencies for which there was no flicker.

9.5 Topographical Maps

Topographical maps are an excellent and nearly ubiquitously used method for showing the spatial distribution of EEG results. Topographical maps are fairly straightforward to construct. It is useful to understand how they are made, although in practice it is easier to use plotting routines that come with a data analysis package rather than to write your own topographical map routines. Creating a topographical map is conceptually similar to interpolating an electrode (see figure 7.5), except that instead of estimating the activity at one point in space corresponding to a missing electrode, activity is estimated at many points in space between electrodes. Figure 9.4 illustrates why it is useful to interpolate activity in the space between electrodes.

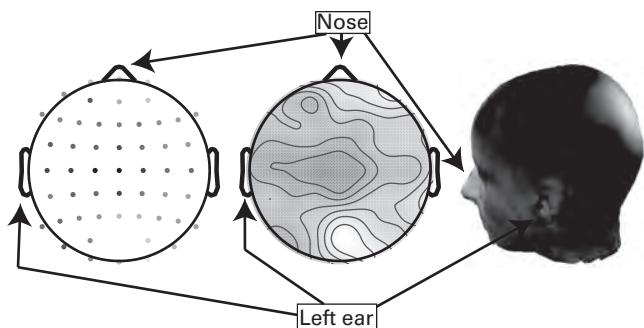


Figure 9.4

The same data shown by coloring dots on a topographical map (left) or interpolating those values over a surface (middle and right). Clearly, the interpolated maps are easier to interpret. Two-dimensional plots show data from all or most electrodes; 3-D plots (right) look more realistic but obscure data from part of the scalp. Locations of the nose and left ear are indicated to facilitate orientation and comparison.

In Matlab, creating a topographical map can be done by defining a set of grid points on a surface on which to interpolate data, based on the data from known topographical positions. The more grid points (that is, the finer the surface lattice), the smoother the topographical plot will be. However, increased smoothness comes at the expense of increased computation time, and smoother topographical plots do not contain any more information than less smooth topographical plots. The online Matlab code goes through, step-by-step, how topographical maps are created. In practice, however, it is easiest to use the topographical plotting routines that come with eeglab, fieldtrip, or other EEG analysis programs. For the remainder of this book, topographical maps are created using the eeglab function `topoplot`.

Most topographical plots are two-dimensional (2-D). To interpret 2-D topographical plots, look at the plot as if you are looking down on the top of someone's head, with the nose on the top of the plot and the left and right ears on the left and right sides, respectively (fortunately, topographical plots are always shown in neurological convention).

Sometimes 3-D topographical plots are shown. There are advantages and disadvantages of 2-D and 3-D topographical plots. Two-dimensional plots are a bit less intuitive to interpret in an anatomical sense but show activity simultaneously from all or most electrodes (if you have many electrodes, those electrodes on or close to the face and neck may be excluded from the plots). Three-dimensional plots are easier to interpret and look nicer in figures, but they show activity from only a third of the head; multiple views would be required to show the full topographical distribution.

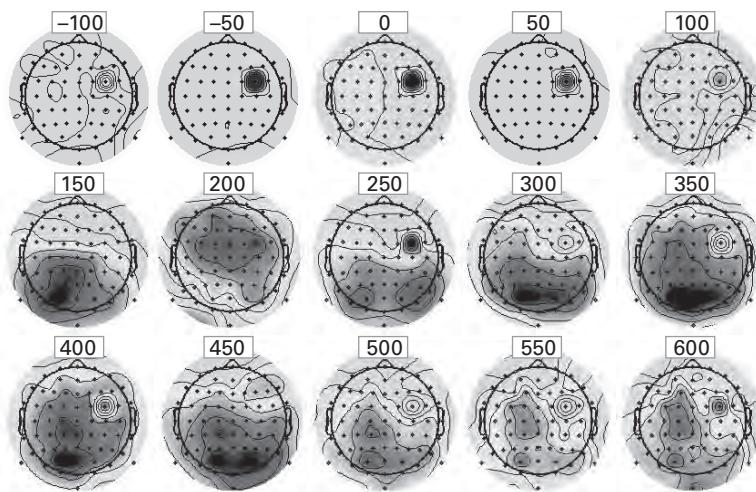


Figure 9.5 (plate 1)

Plotting topographical maps over time facilitates rapid data quality inspection. The numbers in white boxes indicate the latency at which the topographical data are plotted (in milliseconds) with respect to trial onset. These plots show, among other things, that there is one bad electrode. In this case the bad electrode was generated by replacing the true EEG activity at electrode FC4 with randomly generated numbers.

Inspecting topographical maps of ERPs for each subject is a good idea, even if you have no hypotheses that can be tested with ERPs. Topographical maps of ERPs provide excellent and rapid data inspection possibilities. Topographical maps allow you to confirm the timing of task events, and they allow to you detect bad or noisy electrodes. An example is shown in figure 9.5 (plate 1).

9.6 Microstates

A good description of microstates is available on Scholarpedia (scholarpedia.org/article/EEG_microstates; accessed in late 2012), so it is copied here: “In EEG as well as ERP map series, for brief, subsecond time periods, map landscapes typically remain quasi-stable, then change very quickly into different landscapes” (Lehmann 1971). The durations of these “landscapes,” as well as their topographical characteristics, vary over time and as a function of task demands. Durations tend to be around the alpha range (70–130 ms), and topographical distributions tend to fit into four or five distinct patterns. Microstates have been linked to

cognitive processes from perception to memory to language (Britz and Michel 2010; Lehmann et al. 2005; Muller et al. 2005; Pitts and Britz 2011), including microstates during the prestimulus interval that predicts upcoming stimulus processing (Britz and Michel 2011; Lehmann et al. 1994).

The following is a brief description of how microstates are identified; more details and algorithms can be found in publications by Brunet, Murray, and Michel (2011) and Murray, Brunet, and Michel (2008). To identify microstates, consider that when there are temporally stable topographical distributions, the difference between the topographical maps at times t and $t + 1$ is small. For example, in the case that the voltage does not change over two successive time points, the temporal difference is zero. In contrast, rapid changes in the topographical distribution of activity (the cortical landscape of electrophysiological activity) from one time point to the next would result in a large temporal difference. Thus, when the temporal difference (also called global map dissimilarity) remains low for a period of tens to hundreds of milliseconds and then suddenly becomes relatively large, this sharp increase is considered to be a transition from one state to another. The stable maps are then used in a hierarchical clustering analysis in order to identify a small number of topographical distributions that best characterize the topographical maps during periods of stability. These are called cluster maps. Finally, the topography at each time point is labeled according to the cluster map to which it is most similar. This produces a time course of map topographies and can be used in task-related and statistical analyses. The software package cartool is designed for studying microstates (sites.google.com/site/fbmlab/cartool).

9.7 ERP Images

An ERP image is a 2-D representation of the EEG data from a single electrode. Rather than all trials averaged together to form an ERP, the single-trial EEG traces are stacked vertically and then color coded to show changes in amplitude as changes in color. They are also useful as single-subject data inspection tools because trials with large-amplitude data (which likely contain artifacts) can easily be seen. ERP images can also be used to link trial-varying task parameters or behaviors to the time-domain EEG signal. This is done by sorting the EEG trials according to values of the aligning event, such as the reaction time or the phase of a frequency-band-specific signal at a certain time point. Thus, the trial sequence no longer corresponds to chronological order but rather to the values of the aligning event.

Two examples of ERP images are shown in figure 9.6. Figure 9.6A shows an ERP image from electrode FCz aligned to the reaction time on each trial. The reaction times are plotted

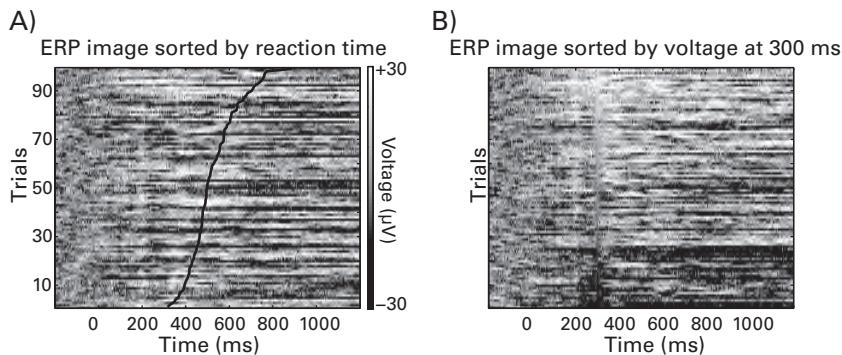


Figure 9.6

Example ERP images, both using data from electrode FCz. Time is on the *x*-axis, trials are on the *y*-axis, and the grayscale intensity (or color if you create this figure on your computer) corresponds to the EEG voltage values over time and trials. In panel A the trials are re-sorted according to the reaction time on each trial, and in panel B the trials are re-sorted according to the voltage value at 300 ms. The black line in panel A corresponds to the reaction time on each trial.

on top of the EEG data as a black line. You can see that the reaction time line goes monotonically to the right. This shows that the trials were re-sorted from their initial chronological configuration to one that follows the reaction time distribution. In this example there is no clear visual relationship between the single-trial EEG and reaction time. Figure 9.6B shows an ERP image that is aligned to the voltage value at 300 ms poststimulus.

ERP images are often smoothed, for example by convolving the image with a 2-D Gaussian, to facilitate interpretation and to minimize the influence of noise or other nonrepresentative single-trial fluctuations. The program eeglab has easy-to-use options for creating and smoothing ERP images.

Despite their name, ERP images are not limited to time-domain EEG data; they could also be made from frequency-band-specific activity, such as the filtered signal, power, or phase. ERP images may sometimes look like time-frequency plots because time is on the *x*-axis and the data are colored. However, the *y*-axis of ERP images contains no frequency information.

9.8 Exercises

1. Compute the ERP at each electrode. Select five time points at which to show topographical plots (e.g., 0 to 400 ms in 100-ms steps). In one figure, make a series of topographical plots at these time points. To increase the signal-to-noise ratio, make each plot show the average of

activity from 20 ms before until 20 ms after each time point. For example, the topographical plot from 200 ms should show average activity from 180 ms until 220 ms. Indicate the center time point in a title on each subplot.

2. Loop through each electrode and find the peak time of the ERP between 100 and 400 ms. Store these peak times in a separate variable and then make a topographical plot of the peak times (that is, the topographical map will illustrate times in milliseconds, not activity at peak times). Include a color bar in the figure and make sure to show times in milliseconds from time 0 (not, for example, time in milliseconds since 100 ms or indices instead of milliseconds). What areas of the scalp show the earliest and the latest peak responses to the stimulus within this window?

III Frequency and Time-Frequency Domains Analyses

10 The Dot Product and Convolution

In this section of the book (chapters 10–21), you will learn several techniques for extracting time-varying frequency-band-specific information from EEG data. Many techniques for extracting frequency information from a time-domain signal rely on a mathematical procedure called *convolution*, which in turn relies on a procedure called the *dot product*. The better you understand how convolution and the dot product work, the better you will understand how time-frequency-based analyses work, and the more flexibly you will be able to adapt and program time-frequency-based analyses.

10.1 Dot Product

Before learning about convolution, you need to become familiar with the basic step of convolution, which is called the dot product. There are several interpretations of the dot product. It can be thought of as a sum of elements in one vector weighted by elements of another vector (a signal-processing interpretation) or as a covariance or similarity between two vectors (a statistical interpretation) or as a mapping between vectors (the product of the magnitudes of the two vectors scaled by the cosine of the angle between them; a geometric interpretation). In different situations, you might find different interpretations to be more useful. But the algorithm is always the same, and the result is always the same: a single number that is computed based on two vectors of equal length.

To compute the dot product, you simply multiply each element in one vector by the corresponding element in the other vector (that is, the first element in vector A by the first element in vector B; the second element in vector A by the second element in vector B, and so on), and then sum all of the points.

$$\text{dotproduct}_{ab} = \sum_{i=1}^n a_i b_i \quad (10.1)$$

a and b are vectors and n is the number of elements in a (or b ; the two vectors must have the same length). Translated into Matlab code, equation 10.1 can be expressed as a loop and then a sum, or it can be expressed as a single matrix multiplication. The online Matlab code shows you several different but equivalent methods to compute the dot product. The dot product may seem like a very simple and trivial expression, but do not underestimate its utility: the dot product is the basic building block of many data analysis techniques, including convolution and the Fourier transform.

The geometric interpretation of a dot product is particularly useful for time-frequency decomposition because this interpretation will facilitate understanding how to extract power and phase angles from complex numbers. Geometrically, the dot product is a mapping of one vector onto another. This is easiest to illustrate visually using two two-element vectors because two-element vectors can be conceptualized on a two-dimensional Cartesian plane. First, consider that the vectors $[a_1 \ b_1]$ and $[a_2 \ b_2]$ can be expressed as lines from the origin of the plot to the points described by $[a_1 \ b_1]$ and $[a_2 \ b_2]$ (figure 10.1). Now imagine drawing a line from the end of one vector to the other vector such that they intersect with a 90° angle. The length of the projection onto the vector is the dot product. Figure 10.1 illustrates situations in which the dot product can be greater than zero, equal to zero, and less than zero. Note that the dot product is zero when the vectors are orthogonal to each other (that is, when they intersect with a 90° angle).

A vector with two elements can be conceptualized as a point in a 2-D space. A vector with three elements can therefore be conceptualized as a point in a 3-D space (a cube), and so on for as many dimensions as there are elements in the vector. However, the principle for computing the dot product is the same as illustrated in figure 10.1, regardless of how many

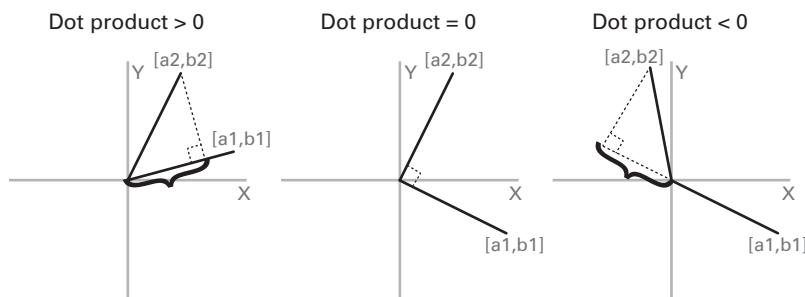


Figure 10.1

Graphical illustration of the geometric interpretation of the dot product between two two-element vectors. Curly brackets illustrate the magnitude of the projection of one vector onto the other (this is the dot product).

dimensions there are. Thus, an N -element vector can be conceptualized as a single point in an N -dimensional space, and the dot product between two N -element vectors is the mapping between two hyperlines from the origin of the N -dimensional space to the points defined by the elements (or coordinates) of those vectors. Thus, you can conceptualize an EEG signal that is 640 time points long as a single point in a 640-dimensional space, and the location of that point is defined by the values (coordinates) of the EEG signal at each time point. As you will learn over the next several chapters, several frequency and time-frequency decomposition methods involve computing the mapping between two high-dimensional vectors, such as a sine wave at a particular frequency and an EEG time series.

10.2 Convolution

Convolution in the time domain (convolution in the frequency domain is discussed in the next chapter) is an extension of the dot product, in which the dot product is computed repeatedly over time. Convolution can also be performed over space, but for EEG analyses, convolution over time is used.

Similar to the dot product, convolution can have several interpretations: you can think of convolution as a time series of one signal weighted by another signal that slides along the first signal (a signal-processing interpretation) or as a cross-covariance (the similarity between two vectors over time; a statistical interpretation, but see section 10.4 concerning convolution vs. cross-covariance), as a time series of mappings between two vectors (a geometric interpretation), or as a frequency filter (more on this in chapter 11). Again, different interpretations might be helpful in different situations, but the algorithm is the same: convolution is performed by computing the dot product between two vectors, shifting one vector in time relative to the other vector, computing the dot product again, and so on.

In convolution, one vector is considered the signal, and the other vector is considered the kernel. In practice it is possible to get the same result no matter which vector is labeled signal and which is labeled kernel. Thus, the distinction is mainly one of convenience. In this book the EEG data is called the signal, and the wavelet or sine wave is called the kernel.

10.3 How Does Convolution Work?

To introduce you to convolution, it is useful to consider a simple example. A square wave will be the signal (figure 10.2A), and a five-point linear decay function will be the kernel (figure 10.2B). Imagine taking the kernel, flipping it backward, and then dragging it forward in time so it passes over the signal. As the kernel passes over the signal, it drags on the signal, and the

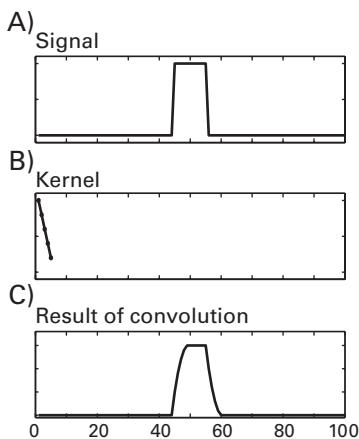


Figure 10.2

Example of convolution between two vectors, one labeled the signal (panel A) and one labeled the kernel (panel B). Panel C shows the result of the convolution between these two vectors.

result of the drag is something that is neither the signal nor the kernel, but looks a bit like each of them. Thus, conceptually, convolution entails sliding the kernel along the data and obtaining a result that shows what features the signal and the kernel have in common. This “commonality” at each time point is computed as the mapping between the kernel and the signal, that is, the dot product.

Hopefully figure 10.2 gave you an intuitive feel for what would happen when you drag the kernel along the signal. Figure 10.3 illustrates how convolution works, step by step. First, flip the kernel backward and line it up with the data as in figure 10.3A (the alignment actually starts before the signal for reasons that will be explained in a few paragraphs). Then, compute the dot product by multiplying each point in the kernel by each corresponding point in the data and summing over all of the multiplications. The dot product between the kernel and the corresponding part of the signal is then placed in a new vector in the position corresponding to the center of the kernel (figure 10.3A). It is convenient but not necessary to use kernels that have an odd number of data points, so that there is a center point.

This is the first step of convolution. Now the kernel is shifted to the right by one time step, but the signal does not move. Repeat the dot product procedure and store this result, again in the position corresponding to the center of the kernel, which is now one time step to the right. Notice that the dot product is computed with the same kernel but a slightly different part of the signal. This process of computing the dot product and then moving the kernel one step to the right continues until the kernel has reached the end of the data (figure 10.3B).

The resulting vector is a time series of dot products between the kernel and temporally corresponding equal-length windows of the signal.

There is one more thing to explain about convolution. You may have noticed that if you line up the kernel and the data such that they have their leftmost points overlapping and continue computing dot products until the kernel and the signal have their rightmost points overlapping, the result of the convolution contains fewer points than the signal (figure 10.3B). In this example with a short kernel, this is not a big loss, but imagine if the kernel and the data were equally long—the result of convolution would be one single dot product. This is not good. Therefore, convolution actually begins with the kernel aligned to the left such that the rightmost point of the kernel (after it is reversed) overlaps with the leftmost point of the data (10.3C). Because the two vectors must have the same number of points in order to compute a dot product, zeros are added before the start of the signal. The number of zeros added is the length of the kernel minus one (because the rightmost point of the kernel overlaps with the first point of the signal). These extra zeros do not contribute to the result because zero times a kernel is zero. Then, convolution runs each step toward the right, and it stops when the leftmost point of the kernel is aligned with the rightmost point of the signal. Again, zeros are added after the end of the signal to be able to perform a dot product.

As you can see in figure 10.3C, this means that the result of the convolution between the kernel and the signal is longer than the signal. How much longer? The result of convolution is one-half of the length of the kernel too long in the beginning and one-half of the length of the kernel too long at the end. The formula for how long the result of the convolution will be is $\text{length}(\text{signal}) + \text{length}(\text{kernel}) - 1$. Thus, after you run convolution, you need to trim the result by removing one-half of the length of the kernel from the beginning and one-half of the length of the kernel plus one from the end. This leaves you with a result of convolution equal to the length of the signal.

When written in equations, convolution is often denoted using the asterisk (*), as in: the convolution between a and b is $a * b$. This can get confusing because in many programming languages, including Matlab, the asterisk indicates multiplication.

The Matlab code online takes you through time-domain convolution step by step. You should take your time going through this code until you understand how time-domain convolution works. It may seem tedious to go through this code and perform convolution in the time domain when you could simply use the Matlab function `conv`. However, as mentioned at the beginning of this chapter, convolution is the basic step of many time-frequency decomposition methods, and so it is important to understand how it works.

Figure 10.4 shows that the result of this “manual” convolution is the same as the result of using the Matlab `conv` function, which performs convolution in the frequency domain

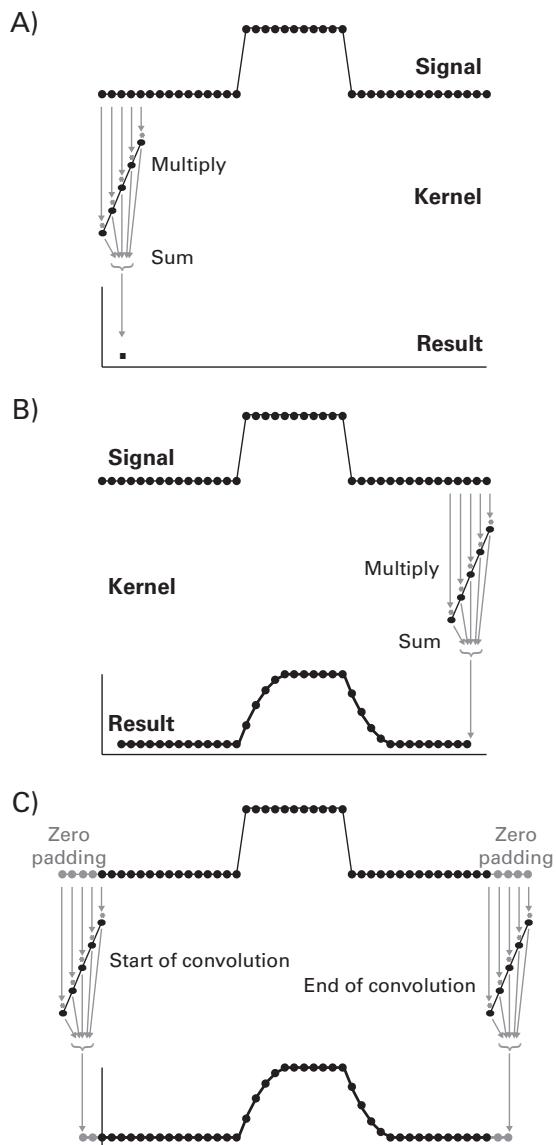
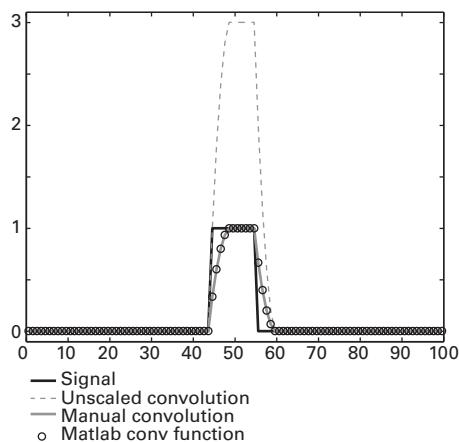


Figure 10.3

Overview of convolution. Panel A shows one step of convolution in which each point in the kernel is multiplied by each corresponding point in the signal, the multiplications are summed, and that value—the dot product—is placed in a position corresponding to the center of the kernel. Note that the kernel is reversed (compare with figure 10.2B). The kernel is then moved one time point to the right, and the dot product is computed again with a slightly different part of the signal. Panel B shows the result after many steps of convolution. Note that by lining up the leftmost point of the kernel with the leftmost point of the data at the start of convolution (and the rightmost point of the kernel with the rightmost point of the data at the end of convolution), the result of convolution is shorter than the signal. For this reason convolution actually begins with the rightmost point of the flipped kernel aligned to the leftmost point of the data, and convolution ends when the leftmost point of the kernel is aligned with the rightmost point of the data, as seen in panel C. The data are zero-padded on either end to compute the dot product with an equal number of points. The result of convolution therefore contains one-half of the length of the kernel too many points at the beginning and at the end. These extra points are discarded, leaving the result of convolution with as many points as the signal.

**Figure 10.4**

Overlap of results of manual time-domain convolution and the Matlab frequency-domain convolution function `conv`. Also shown is the result of convolution after normalization, which is not necessary when the sum of the kernel points is zero, as is the case with wavelets.

rather than the time domain (more on how this works later). To compare the result of convolution with the original signal, it is useful to scale the result of the convolution by the sum of the kernel points. Keep in mind that postconvolution scaling is not the same thing as mean-centering the kernel before convolution—the mean-centered kernel will produce a different convolution result because it will have negative numbers.

For completeness, equation 10.2 shows the mathematical formula for convolution.

$$(a * b)_k = \sum_{i=1}^n a_i b_{k-i} \quad (10.2)$$

In this equation, a and b are the two vectors for which the dot product is computed, k indicates that this equation refers to the convolution at time point k , and i corresponds to the elements in the equal-length parts of vectors a and b . Vector b is the kernel because it is flipped backward (this is the effect of $k - i$). You can see clearly from equation 10.2 that each step of the convolution is simply the dot product between the vector a and the time-reversed vector b .

10.4 Convolution versus Cross-Covariance

Convolution and cross-covariance (or cross-correlation; the cross-correlation is simply the cross-covariance scaled by the variances) are similar in concept but slightly different in algorithm. Both are methods to compute the time-varying mapping between vectors, but with convolution the kernel is reversed, whereas with cross-covariance the kernel is kept in its original orientation. If the convolution kernel is temporally symmetric—and thus is the same whether flipped backward or not—convolution and cross-covariance yield identical results.

10.5 The Purpose of Convolution for EEG Data Analyses

In EEG data analyses, convolution is used to isolate frequency-band-specific activity and to localize that frequency-band-specific activity in time. This is done by convolving wavelets—time-limited sine waves—with EEG data. As the wavelet (the convolution kernel) is dragged along the EEG data (the convolution signal), it reveals when and to what extent the EEG data contain features that look like the wavelet. When convolution is repeated on the same EEG data using wavelets of different frequencies, a time-frequency representation can be formed. Before learning about how this works and how to implement it yourself in Matlab, you will need first to learn about the Fourier transform and the convolution theorem. These are the topics of the next chapter.

10.6 Exercises

1. Create two kernels for convolution: one that looks like an inverted U and one that looks like a decay function. There is no need to be too sophisticated in generating, for example, a Gaussian and an exponential; numerical approximations are fine.
2. Convolve these two kernels with 50 time points of EEG data from one electrode. Make a plot showing the kernels, the EEG data, and the result of the convolution between the data and each kernel. Use time-domain convolution as explained in this chapter and as illustrated in the online Matlab code. Based on visual inspection, what is the effect of convolving the EEG data with these two kernels?

11 The Discrete Time Fourier Transform, the FFT, and the Convolution Theorem

The Fourier transform is an incredibly important signal-processing technique in time-series data analysis as well as in many other branches of science, engineering, and technology. It is the computational backbone of most EEG data analyses. In turn, the backbone of the Fourier transform is the dot product, which is why the previous chapter was dedicated to that topic.

The Fourier transform works by computing the dot product between the signal (such as EEG data) and sine waves of different frequencies (the kernels). Remember from chapter 2 that sine waves have three characteristics: frequency (how fast, measured in cycles per second or hertz), power (or amplitude; power is amplitude squared, so these terms are sometimes used interchangeably), and phase (the timing of the sine wave, measured in radians or degrees). Thus, the result of the Fourier transform is a three-dimensional (3-D) representation of the time series data in which the three dimensions are frequency, power, and phase. This 3-D Fourier representation contains all of the information in the time series data, and the time series data can be perfectly reconstructed from the result of its Fourier transform (this will be demonstrated later in this chapter).

To perform a Fourier transform, you need to know how to compute the dot product, and you need to know how to create a sine wave. You already know how to compute the dot product; now it is time to learn about creating sine waves.

11.1 Making Waves

Sine waves can be represented mathematically using the following formula:

$$A \sin(2\pi ft + \theta) \tag{11.1}$$

in which A is the amplitude of the sine wave, π is pi, or $3.141\ldots$, f is the frequency of the sine wave, t is time (this could also be space, but only time-varying sine waves are discussed here), and θ is the phase angle offset, which is related to the value of the sine wave

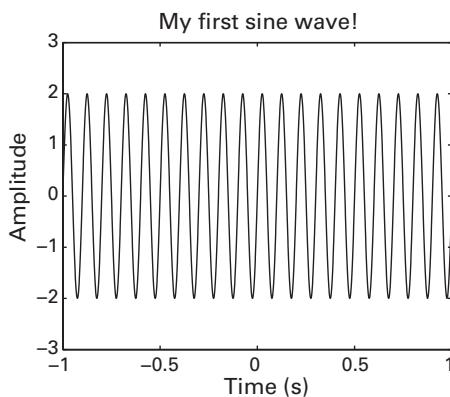


Figure 11.1

A sine wave that was created using the expression $2\sin(2\pi 10t + 0)$ (thus, amplitude of 2 and frequency of 10 Hz).

at time = 0. If the amplitude is 1 and the phase is zero, equation 11.1 reduces to the perhaps more familiar-looking $\sin(2\pi ft)$. Figure 11.1 shows a sine wave created using equation 11.1.

Take a few minutes to explore the online Matlab code that created this figure by changing the sampling rate, the frequency, the phase, and the amplitude. Try to get a feel for the effects those parameters have on the resulting sine wave. You should notice that the frequency parameter changes the speed of the oscillation, and the number indicates the cycles per second and has units of hertz. Changing the sampling rate has little effect except that the sine wave will look more jagged when the sampling rate becomes small relative to the frequency. Changing the amplitude affects the height of the peaks and troughs of the sine wave. If left undefined, the amplitude is implicitly set to 1.

The most obvious feature of a sine wave is that it goes up and down over time. Why does it go up and down? You will learn in the next chapter that sine waves can be better conceptualized as a spiral in a 3-D space and that what you see in figure 11.1 is a projection onto two of the three dimensions. This gives the appearance of going up and down.

By adding together several sine waves of different frequencies, amplitudes, and phases, the resulting time series looks more complicated than any of the individual sine waves. Figure 11.2A shows five individual sine waves, and figure 11.2B shows the result of adding all of those sine waves together. Again, spend a few minutes going through the online Matlab code that generated this figure. Try changing the parameters of the sine waves to see how the summed time series looks, and try adding a few more sine waves to see what the effect is on the resulting sum. Note that if you want to add more sine waves, the variables `frequencies`, `phases`, and `amplitudes` must have the same number of elements (one for each sine

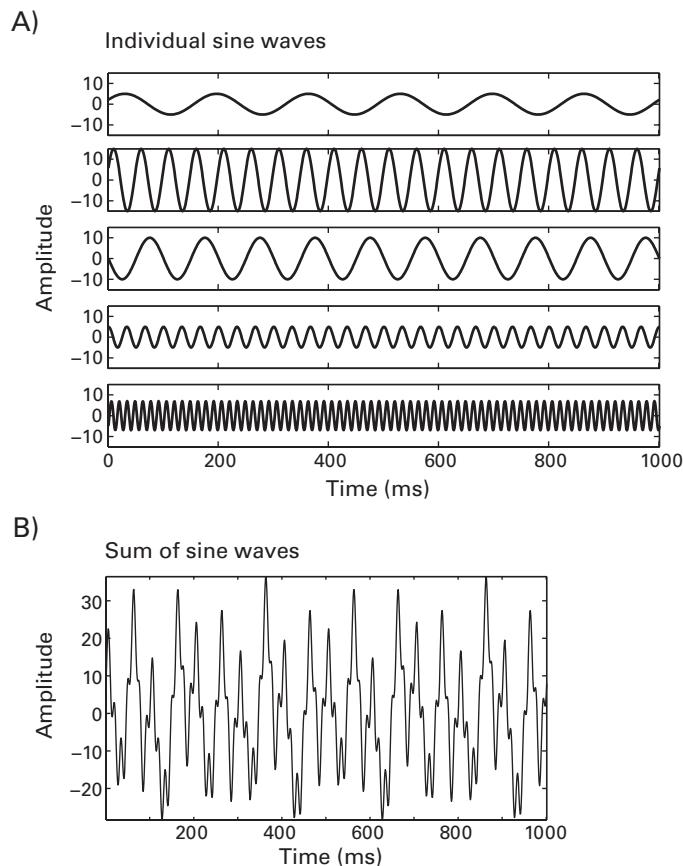


Figure 11.2

Several sine waves of differing amplitudes, frequencies, and phases, plotted separately (panel A) and after being added together (panel B).

wave). Try making `frex` longer than `phases` and `amplitudes`. Where does Matlab crash and why?

Although it is easy to distinguish the time series in figure 11.2B from real EEG data, adding some noise makes the signal look more like real data, as shown in figure 11.3.

11.2 Finding Waves in EEG Data with the Fourier Transform

In the previous examples, time series were created by summing collections of sine waves. However, this does not help you analyze real EEG data. With real data you have the opposite

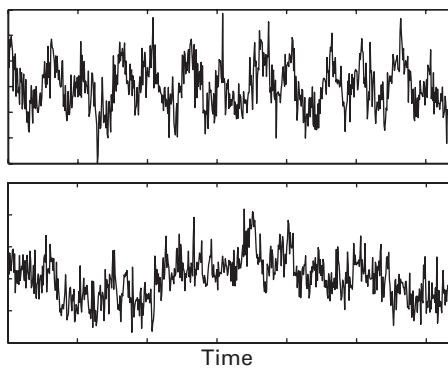


Figure 11.3

One of these time series was generated by summing different sine waves and adding noise; the other time series is real EEG data. Can you guess which is which? What evidence do you use to support your decision?

problem: you already have the time series, and you want to know which sine waves with which frequencies, amplitudes, and phases will reconstruct that time series. This is the purpose of the Fourier transform. Essentially, the Fourier transform works by computing the dot product between sine waves of different frequencies and the EEG data.

The best way to learn how the Fourier transform works is by programming the discrete time Fourier transform. It may seem tedious to program the Fourier transform from scratch when you could simply use the Matlab function `fft`. However, the Fourier transform is one of the two most important and fundamental mathematical expressions for time-frequency-based data analyses (the other is Euler's formula, which you will learn about in chapter 13). The better you understand how the Fourier transform works, the better you will understand the mechanics of the more advanced analyses.

11.3 The Discrete Time Fourier Transform

The idea behind the discrete time Fourier transform is fairly simple: create a sine wave and compute the dot product between that sine wave and the time series data. Then create another sine wave with a different frequency and compute the dot product between that sine wave and the same time series data. The number of sine waves you create, and the frequency of each sine wave, is determined by the number of data points in the time series data (more on this in a few paragraphs). The following Matlab code will perform a discrete time Fourier transform of random data.

```
N = 10; % length of sequence
data = rand(1,N); % random numbers
% initialize Fourier coefficients
fourier = zeros(size(data));
time = (0:N-1)/N; % time starts at 0; dividing by N normalizes to 1
% Fourier transform
for fi=1:N
    % create sine wave
    sine_wave = exp(-1i*2*pi*(fi-1).*time);
    % compute dot product between sine wave and data
    fourier(fi) = sum(sine_wave.*data);
end
```

There are two steps that occur within each iteration of this for-loop. The first step is that a sine wave is created. The frequency of that sine wave is defined by the looping index, and the looping index in turn is defined by the number of points in the data (in this case, the variable N). The equation for the sine wave looks different here compared to equation 11.1—you can still see the $2\pi ft$ part, but there is no “sin” function, and instead there is an `exp` and a `-1i`. This makes a complex sine wave, which is something you will learn about in chapter 13; for now, just keep in mind that this is a sine wave. Note that there are other equivalent ways of expressing this equation, such as a cosine wave plus an imaginary sine wave. The expression used above is preferred here because it will help you understand the link between the Fourier transform and Euler’s formula.

You can also see that the frequency of the sine wave is defined by one less than the looping index ($fi-1$), whereas you might expect it to be defined by fi . The sine wave at each iteration corresponds to a frequency one step slower than that iteration. Thus, the first iteration of this loop produces a sine wave of zero frequency. This is also sometimes called the “DC component,” where DC stands for direct current. A sine wave with zero frequency is a flat line, and thus the zero frequency captures the mean offset over the entire signal. For example, if you subtract the mean of the signal before computing its Fourier transform, the DC or zero frequency component will be zero.

The second step inside the loop of the Fourier transform computes the dot product between the sine wave and the data. As you learned in chapter 10, the dot product computes the mapping between two vectors, which can be thought of as a way to measure the similarity between those two vectors. In figure 10.1 you saw how the dot product works visually on two 2-D vectors; here, the dot product is computed between two 10-D vectors. Thus, the

Fourier transform works by computing the dot product between the signal and sine waves of different frequencies.

The frequencies are defined by a looping index, but it would be convenient to know what those frequencies are in hertz. To convert the frequencies to hertz, consider that the number of unique frequencies that can be extracted from a time series is exactly one-half of the number of data points in the time series, plus the zero frequency. This is due to the Nyquist theorem: you need at least two points per cycle to measure a sine wave, and thus half the number of points in the data corresponds to the fastest frequency that can be measured in those data. Thus, the number of unique frequencies that can be extracted from a time series of length N is $N/2 + 1$ (the $+1$ is for the zero frequency component). This also means that $N/2 + 1$ is the frequency resolution of the data. Because frequencies start at zero and go up to the Nyquist frequency, to convert frequencies from indices to hertz, you create linearly equally spaced numbers between zero and the Nyquist frequency in $N/2 + 1$ steps. This can be done in Matlab using the function `linspace`; the online Matlab code shows you how to do this.

For completeness, below is the formula for the discrete Fourier transform of variable x at frequency f .

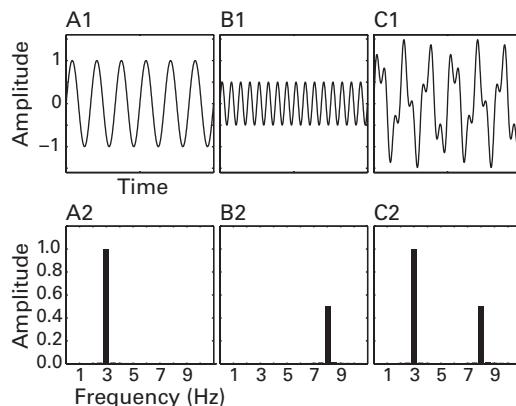
$$X_f = \sum_{k=1}^n x_k e^{-i2\pi f(k-1)n^{-1}} \quad (11.2)$$

in which n refers to the number of data points in vector x , and the capital letter X_f is the Fourier coefficient of time series variable x at frequency f . Again, do not worry too much about the exponential and imaginary number; these are explained in detail in chapter 13. You might also sometimes see this equation written with the summation variable from $k = 0$ to $n - 1$, but equation 11.2 is written with indices starting at one instead of zero to facilitate comparison with the implementation in Matlab.

11.4 Visualizing the Results of a Fourier Transform

Now that you have computed the Fourier transform, how can you view the results? Remember that the Fourier transform provides a 3-D representation of the data in terms of frequency, power, and phase. Very often, the phase information is ignored when results of a Fourier analysis are shown, thus leaving two dimensions to show (the 3-D representation is considered below). Thus, the most typical way of showing results from a Fourier analysis is a 2-D plot with frequency on the x -axis and power (or amplitude) on the y -axis, as illustrated in figure 11.4.

Notice that figure 11.4 uses bars instead of lines to plot the results of the Fourier transform. Because frequencies are discretely sampled, it is appropriate to use bars or dots to plot

**Figure 11.4**

Two sine waves of 3 Hz and 8 Hz (panels A1 and B1), and their sum (panel C1) have a Fourier power spectrum that can be represented in plots of frequency (x -axis) by amplitude (y -axis) (see plots in the bottom row). The frequency of the sine wave corresponds to the position on the x -axis, and the amplitude of the sine wave corresponds to the position on the y -axis.

Fourier results instead of lines, which would imply a smooth transition between sampled points. However, with long time series, the discretization becomes very fine, and bar plots can be impractical for viewing, particularly for viewing several results on the same plot. Thus, in practice, line plots can be used because they increase visibility and the possibility for direct comparisons across Fourier results from different time series.

As mentioned earlier, although the result of a Fourier transform is often presented in 2-D plots for convenience, the result of a Fourier transform actually occupies a 3-D space. This is shown in figure 11.5. The Fourier transform was taken from randomly generated data (shown in figure 11.5A). Figure 11.5B shows that the Fourier transform of that time series data can be represented in a frequency-power-phase space. Note that the phase here refers to the position of the sine wave at each frequency when it crosses time = 0; this is slightly different from the phase angle time series. You will learn more about this in chapter 13. Figure 11.5C–D shows that rotating the 3-D space to show only two dimensions at a time allows power and phase spectra to be plotted in a format that you might be more used to seeing.

11.5 Complex Results and Negative Frequencies

How does this 3-D information come from the single variable `fourier` in the Matlab code shown earlier? If you look at the contents of this variable in Matlab, you will see that it has the form $a + ib$. This is the representation of a complex number and is crucial for extracting

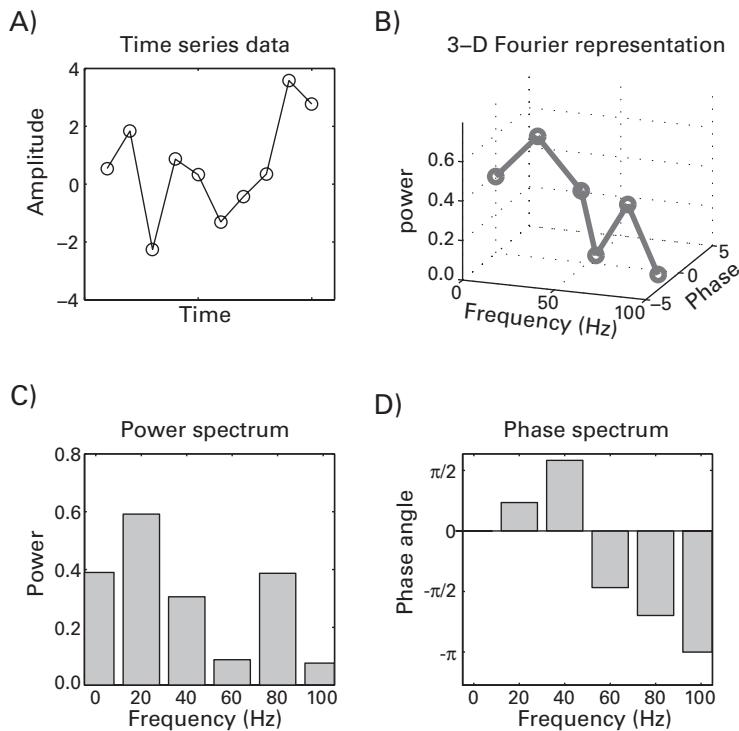


Figure 11.5

Panel A shows a time series of randomly generated data (sampled at 200 Hz). Panel B shows a frequency representation of the data as a line through 3-D Fourier space (frequency, power, and phase; see axis labels). Panels C and D show the projections of the 3-D Fourier result onto two dimensions at a time. Panel C shows only the frequency and power information from the Fourier transform, and panel D shows only the frequency and the phase information from the Fourier transform.

power and phase information, and will be covered in greater detail in chapter 13. For now, you can just plot `abs(fourier)` in Matlab, which gives you the amplitude of the Fourier series.

If you inspect the online Matlab code that generates figure 11.5, you may notice that only the first half of the results of the Fourier transform are plotted. The first half of the Fourier series contains the “positive” frequencies, and the second half of the Fourier series contains the “negative” frequencies. The negative frequencies capture sine waves that travel in reverse order around the complex plane compared to those that travel forward (positive frequencies). This feature of the Fourier transform will become relevant when you learn about the Hilbert transform (section 14.1) but is not further discussed here. For a signal with only real numbers (as is the case with EEG data), the negative frequencies mirror the positive frequencies; you

can thus ignore the negative frequencies and double the amplitudes of the positive frequencies. However, do not remove the negative frequencies from the Fourier series because you will need them to compute the inverse Fourier transform.

11.6 Inverse Fourier Transform

The Fourier transform allows you to represent a time series in the frequency domain. It is often stated that the Fourier transform is a perfect frequency-domain representation of a time-domain signal. If this is true, it must be possible to reconstruct the original time series data with only the frequency domain information. This is done through the inverse Fourier transform.

Once you understand how the Fourier transform is computed, the inverse Fourier transform is easy to understand and compute. Remember that the result of a Fourier transform is a series of coefficients that represent the mapping (the dot product) of each complex sine wave to the data. The inverse Fourier transform reverses this procedure. In fact, figure 11.2 was conceptually similar to an inverse Fourier transform: you started with sine waves of different frequencies, amplitudes, and phases, and you summed them together to form a single time series. Thus, to compute the inverse Fourier transform, you build sine waves of specific frequencies, multiply them by the respective Fourier coefficients at those frequencies, sum all of these sine waves together, and then divide by the number of sine waves (which is also the number of points in the time series). This will result in a perfect reconstruction of the original time series data. The online Matlab code shows you how to perform an inverse Fourier transform, and figure 11.6 shows the time series reconstructed from the Fourier coefficients and the original time series data.

For completeness, equation 11.3 shows the formula for the inverse Fourier transform, using the same format that was used in equation 11.2.

$$x_k = \sum_{k=1}^n X_k e^{i2\pi f(k-1)n^{-1}} \quad (11.3)$$

Notice that in equation 11.3, the dot product (here, a single multiplication) is computed between the complex sine wave and the Fourier coefficients X instead of the time series data x . The other difference between equations 11.2 and 11.3 is that the exponential contained a negative sine in equation 11.2, which is absent in equation 11.3.

This exercise shows that the original time series can be perfectly reconstructed based on the Fourier transform. This becomes important for learning about the convolution theorem and why you should perform convolution in the frequency domain.

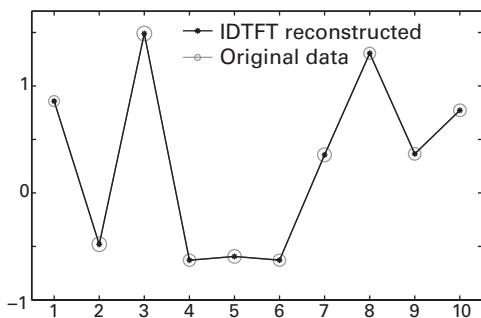


Figure 11.6

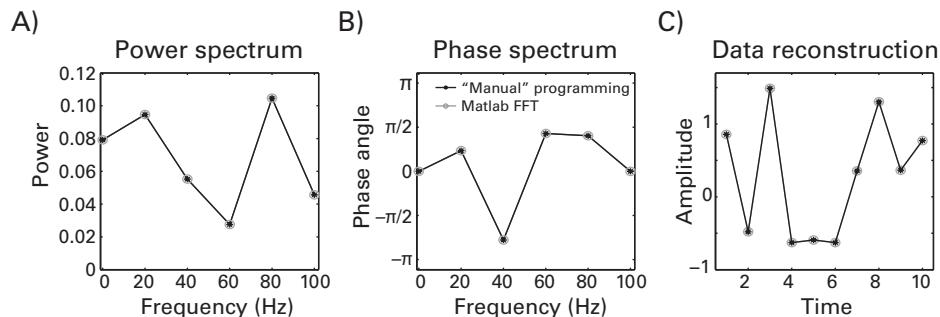
Time series data of randomly generated numbers and the reconstruction of the data using the inverse discrete time Fourier transform (IDTFT).

For the statistically oriented readers, there is also a statistical explanation of why the Fourier transform can perfectly represent a time series. If the time series is modeled using the same number of sine waves (analogous to regressors) as data points, there are no remaining degrees of freedom, and thus, the model must explain 100% of the variance of the data.

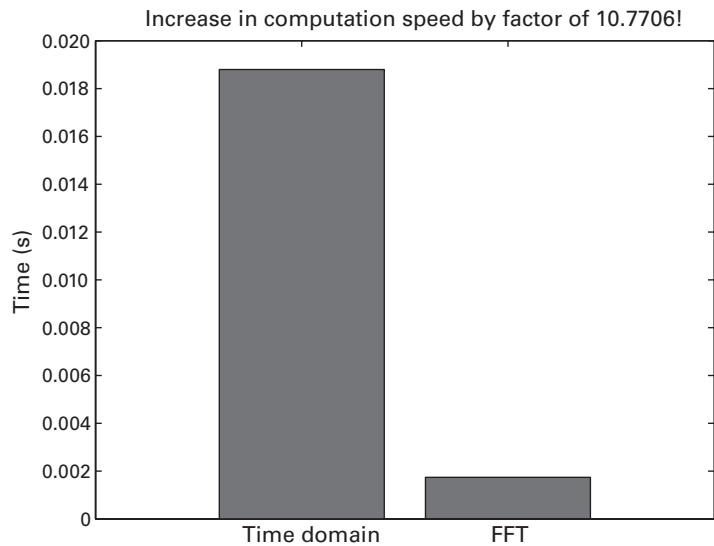
11.7 The Fast Fourier Transform

It is important to understand how the Fourier transform works at both the conceptual and the implementational levels. But, didactic purposes notwithstanding, you should never use the discrete-time Fourier transform as implemented in the online Matlab code for this chapter when analyzing your data. There is a much faster, more efficient, and more elegant way to compute the Fourier transform. It turns out that if you break up the Fourier transform into all of its individual multiplication and addition steps, many of these individual elements are redundant and can be eliminated, thus allowing the Fourier transform to be computed much faster and with no loss of information. There are several methods for increasing the efficiency of the Fourier transform, including the fast Fourier transform (FFT) that is implemented in Matlab using the function `fft`. The Matlab function for the inverse Fourier transform is called `ifft`. Figure 11.7 shows that the FFT and the “manual” discrete time Fourier transform produce identical results.

How much faster is the FFT compared to the discrete time Fourier transform? Figure 11.8 shows results of a computation time test. In this test the Fourier transform was computed on 640 time points of data using the discrete time Fourier transform and the FFT. The FFT was

**Figure 11.7**

The fast Fourier transform (FFT) produces identical results as the discrete time Fourier transform and is much faster. (Note that due to very small computer rounding errors, you might see 0 or 2π , or $-\pi$ or $+\pi$, for the phase value at the Nyquist frequency.)

**Figure 11.8**

Computation times for the discrete time Fourier transform versus the FFT.

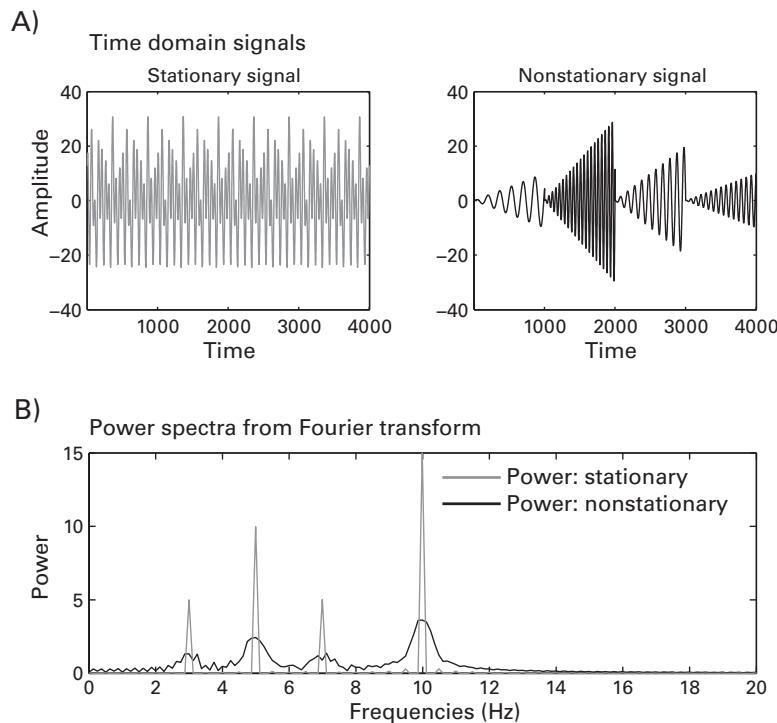
around ten times faster than the discrete time Fourier transform. In practice, though, this is a huge underestimate: the more data you have, the bigger the increase in speed. Furthermore, the speed of the FFT can be further decreased by using vectors with the number of elements corresponding to a power of two. In real data analyses, you may have millions of data points, and if the number of points is a power of two, the decrease in speed for the FFT compared to the discrete time Fourier transform can be around 3 orders of magnitude. This could be the difference between 1 min and 1000 min (17 h) of computation time. Technically, the speed of a discrete time Fourier transform is related to the number of computations as N^2 , where N is the number of data points; the speed of the FFT is related to the number of computations as $N \log N$.

11.8 Stationarity and the Fourier Transform

One assumption of the Fourier transform is that your data are stationary, which means that the statistics of the data, including the mean, variance, and frequency structure, do not change over time (that is, the time series is “well behaved”). This is clearly not the case for EEG data: the frequency structure of neurophysiological activity changes over time both because of task events and because of endogenous processes. In this sense one assumption of the Fourier transform is likely to be violated in real data, particularly data collected during cognitive tasks. Although the lack of stationarity of EEG data reflects dynamic properties of the brain (Kaplan et al. 2005), it can also introduce difficulties for some analyses, including the Fourier transform and Granger prediction (chapter 28).

Violations of stationarity can decrease the peakiness of the results of the Fourier transform. This can be seen in figure 11.9. Two time series were created from the same sine waves of frequencies 3, 5, 7, and 10 Hz (figure 11.9A). One signal was stationary, and the other was nonstationary—variations were introduced by having the frequency structure change over time and also by having the amplitude and therefore also the variance increase over time within each sine wave. As seen in figure 11.9B, the power spectrum clearly shows the spectral peaks for both the stationary and nonstationary time series. However, the spectral peaks for the nonstationary data are less well defined, and there seems to be power at many other frequencies between the peaks, even though those frequencies were not explicitly defined in the simulated data. This is not an artifact but rather a feature: the nonstationary time series has a more complicated structure and therefore requires energy at a larger number of frequencies in order to represent the time series in the frequency domain.

This is one of two main reasons to perform temporally localized frequency decomposition methods such as wavelet convolution, filter-Hilbert, or short-time FFT. With temporally

**Figure 11.9**

Violations of stationarity (here introduced by changes in frequency and amplitude over time) result in energy at frequencies that were not explicitly generated when creating the data. Nonetheless, spectral peaks can clearly be observed for both the stationary and the nonstationary time series. Note that for the stationary time series, the power peaks appear to be carrot shaped, but this is due to using lines instead of bars to represent discretely sampled frequencies. If you look closely, you can also see small but non-zero power at many frequencies for the stationary time series; these are artifacts resulting from the sharp edges at the beginning and ends of the time series.

localized methods, you assume that the data are stationary within relatively brief periods of time (generally a few hundred milliseconds for task data), which seems to be a reasonable assumption for EEG data (Florian and Pfurtscheller 1995).

The second reason to perform a temporally localized frequency decomposition was mentioned earlier: The Fourier transform does not show how dynamics change over time. That is, although the Fourier transform contains all temporal information of the time series data, as demonstrated by the ability of the inverse Fourier transform to reconstruct a time series, time-varying changes in the frequency structure cannot be observed directly in power or phase plots.

11.9 Extracting More or Fewer Frequencies than Data Points

Earlier you learned that the number of frequencies you get from a Fourier transform is $N/2 + 1$, where N is the number of time points in the data, and $+1$ is for the DC or zero-frequency component. Thus, the frequency resolution of the Fourier analysis is specified by the number of time points in the data. However, it is possible to get more or fewer frequencies from a Fourier transform (that is, it is possible to increase or decrease the frequency resolution). Because the number of frequencies is related to the number of data points, you can simply add data points to get more frequencies. However, because you may not have more real data to add, you can add zeros after the end of the time series. This is called zero padding. Zero padding increases the N of the Fourier transform without changing the data, thus allowing you to get more frequencies from the Fourier transform. However, because zero padding does not increase the amount of information in the data, it does not increase the frequency *precision* of the Fourier transform, only the frequency *resolution* (see section 2.8 for a discussion of resolution versus precision).

The Matlab function `fft` has an optional second input, which is the order of the FFT (N), which determines the number of frequencies that are extracted. By default, N is equal to the number of data points, but you can specify this number to be larger or smaller. If N is larger than the number of data points, the time series will be zero-padded, and the FFT will return more frequencies (Matlab adds the zeros; you do not need to do this yourself). Conversely, if N is smaller than the number of data points, data points will be removed from the end of the time series. Although zero padding does not provide any additional information, it can make frequency-domain convolution more convenient and faster to perform. Frequency-domain convolution is discussed in the next section, and the methods to utilize the power-of-two feature to decrease the speed of convolution are shown in chapter 13.

11.10 The Convolution Theorem

Learning how a Fourier transform works is important for several reasons, one of which is that the Fourier transform provides a means to perform convolution in an efficient and elegant manner. This is because of the convolution theorem, which states that convolution in the time domain is the same as multiplication in the frequency domain. (The opposite is also true, although it is less relevant for EEG data analyses: convolution in the frequency domain is the same as multiplication in the time domain.)

Because convolution in the time domain is the same as multiplication in the frequency domain, you can perform convolution in two ways. The first way is the time-domain version of convolution shown in the previous chapter: flip the kernel backward, slide it along the signal, and compute the dot product at each time step. The second way to perform convolution is by taking the Fourier transforms of the signal and the kernel, multiplying the Fourier transforms together point-by-point (that is, frequency-by-frequency), and then taking the inverse Fourier transform (figure 11.10). Although this may not seem as if it should make a difference, it does: time-domain convolution is slow, and frequency-domain convolution is fast.

When you perform the frequency-by-frequency multiplication of the Fourier transforms of the kernel and the signal, you are scaling the frequency spectrum of the signal by the frequency spectrum of the kernel. In other words, the result of the multiplication (and hence, the result of the convolution) is the frequency structure that is common to both the kernel and the signal. This is why convolution can be conceptualized as a frequency-domain filter, as was mentioned in section 10.2. Figure 11.11 shows an example illustrating how

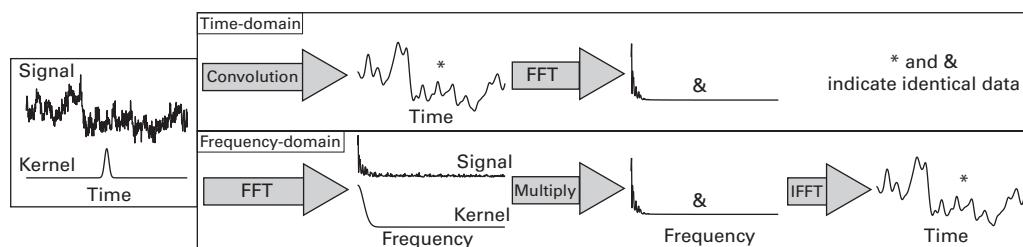


Figure 11.10

Illustration of the convolution theorem and the interchangeability of time-domain convolution and frequency-domain multiplication. The two time series with asterisks are identical, as are the two frequency spectra with ampersands.

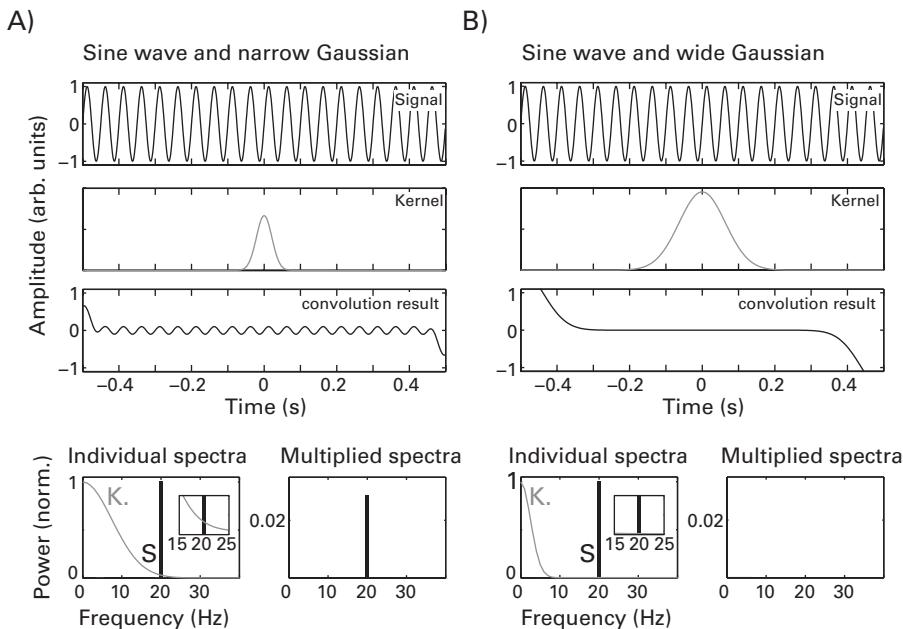


Figure 11.11

The convolution between a 20-Hz sine wave and a narrow Gaussian (panel A) dampens the sine wave, whereas the convolution between the same sine wave and a wide Gaussian (panel B) obliterates the sine wave. This is because the power spectrum of the sine wave (bottom row) overlaps slightly with the power spectrum of the narrow Gaussian at 20 Hz, but the power spectrum of the sine wave does not overlap with the power spectrum of the wide Gaussian. The gray line corresponds to the kernel (K), and the black bar corresponds to the signal (S). The insets in the power plots highlight the overlap (or lack thereof) between the frequency representations of the Gaussians and the frequency representation of the sine wave. The power spectra were normalized to 1 to facilitate visual comparison. The sharp rise and drop at the beginning and end of the result of convolution are edge artifacts.

convolution acts as a frequency filter, using simulated data. A sine wave of 20 Hz was convolved with two Gaussians—one with a narrow width and one with a wide width (the frequencies of the Gaussian widths were 15 Hz and 5 Hz; you'll learn more in the next chapter about specifying the widths of Gaussians). The result of the convolution of the sine wave and the narrow Gaussian simply dampened the amplitude of the sine wave, whereas the result of the convolution with the wide Gaussian obliterated the sine wave.

To understand why this is the case, consider the frequency spectra of the sine wave and the two Gaussians, and consider the overlap of their spectra. The sine wave has a very narrow spectral peak at 20 Hz, and the Gaussians have a smoother slope-looking shape. What

you should notice is that the power spectrum of the narrow Gaussian has nonzero values that overlap with the nonzero values of the power spectrum of the 20-Hz sine wave (see small inset plots in the bottom row of figure 11.11). In contrast, the power spectrum of the wide Gaussian is zero at frequencies where the power spectrum of the sine wave is nonzero. The resulting frequency spectrum multiplications will produce an amplitude-attenuated sine wave for the narrow Gaussian and no sine wave at all for the wide Gaussian.

Thus, convolution acts as a filter such that the frequency profile of the signal is passed through the frequency profile of the kernel. As you will see in the next several chapters, this is the basis of wavelet convolution: you pass the EEG data through a set of filters (wavelets) that are tuned for specific frequencies, and the result of convolution is the frequency-band intersection between the EEG data and the wavelet.

One more example is given of how convolution acts as a frequency-band filter, this time using real EEG data. Figure 11.12 shows one trial of EEG data, a Gaussian, and the result of convolution between them. This figure is laid out in the same way as figure 11.11 is. Thus, in

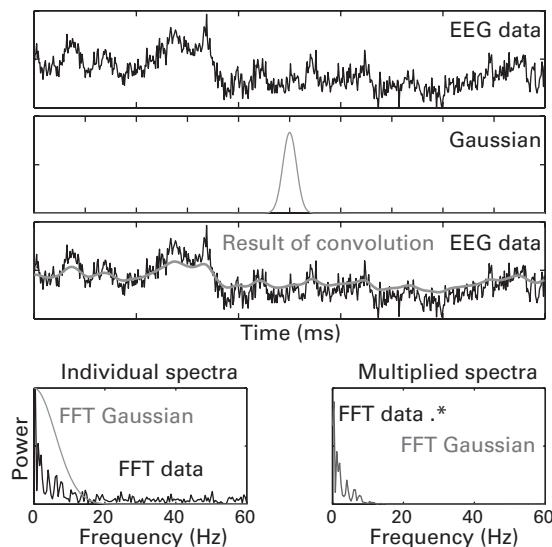


Figure 11.12

Convolving an EEG time series from one trial with a Gaussian low-pass-filters the data. This results from the frequency spectrum of the Gaussian tapering the higher frequencies in the EEG data. Note that this example here is meant for illustration of the convolution theorem and how convolution acts as a frequency filter; convolution with a Gaussian is not necessarily the best method for filtering EEG data. Chapter 14 contains more in-depth discussions of how to construct and apply bandpass filters to EEG data. The y -axis scaling of the power spectra is arbitrary to improve visibility.

the bottom row of figure 11.12 you can see the frequency characteristics of the data and the Gaussian (left-side plot), and you can see the result of multiplying their spectra frequency-by-frequency (right-side plot). The result of convolving the EEG data with a Gaussian is a time series comprising the frequency characteristics of the EEG data weighted by the frequency characteristics of the Gaussian. Thus, the Gaussian kernel is a low-pass filter because its spectral characteristics are dominated by low frequencies, and thus, the high frequencies of the EEG data are severely attenuated.

11.11 Tips for Performing FFT-Based Convolution in Matlab

If you try to perform FFT-based convolution by taking the inverse Fourier transform of the products of the Fourier transforms of the EEG data and the wavelet (or any signal and kernel), you will get an incorrect result. In fact, you may get a Matlab error if the signal and the kernel are not the same length. Even if the signal and the kernel are the same length, the following Matlab command will not produce a valid convolution:

```
result = ifft(fft(signal) .* fft(kernel));
```

The reason the above line of code will not produce a valid convolution is that the result of convolution must be equal to the length of the signal plus the length of the kernel minus one (the reason for this is discussed in section 10.3). Thus, you will need to make the inverse FFT return the correct number of time points, and to do this, you will need to make sure to compute the FFTs of the signal and the kernel using the appropriate number of time points (in other words, the length of the signal plus the length of the kernel minus one). Remember that the result of the FFT can be greater than the number of time points in the input if the `fft` function is called with a second argument that specifies the desired number of frequency points. After you compute the inverse Fourier transform, you will then need to remove the appropriate number of time points from the beginning and from the end of the time series. This is the correct way to perform a convolution using the `fft` and `ifft` functions.

This is a tricky point, and it is very important. If you do not use the `fft` function appropriately, you will get a result (if the signal and the kernel have the same number of time points), but that result will not be a convolution! You can easily double-check your code to make sure you've done it correctly by using the Matlab `conv` function:

```
result = conv(signal,kernel,'same');
```

This result should be identical to the result of the FFT-based convolution. If you use the ‘full’ option instead of the ‘same’ option, the result will be the length of the result of convolution, that is, the length of the signal plus the length of the kernel minus one.

In theory, you can always use the `conv` function instead of performing convolution using `fft` and `ifft`. However, using the `conv` function is inefficient for wavelet convolution with many frequencies because you will redundantly recompute the FFT many times. This can slow data analysis times by an order of magnitude or more. This is explained more toward the end of chapter 13.

11.12 Exercises

1. Reproduce the top three panels of figure 11.12 three times. First, perform time-domain convolution using the “manual” convolution method shown in chapter 10. Second, perform frequency-domain convolution using the discrete time Fourier transform presented at the beginning of this chapter. Finally, perform frequency-domain convolution using the Matlab functions `fft` and `ifft` (do not use the function `conv`). (You can optionally reproduce the bottom panel of figure 11.12 for the frequency domain analyses; keep in mind that the power scaling is for display purposes only.)
2. From the three sets of Matlab code you have for reproducing figure 11.12, run a computation time test. That is, time how long it takes Matlab to perform 1000 repetitions of each of the three methods for computing convolution that you generated in the previous exercise (do not plot the results each time). You can use the Matlab function pairs `tic` and `toc` to time a Matlab process. Plot the results in a bar plot, similar to figure 11.8.
3. Generate a time series by creating and summing sine waves, as in figure 11.2B. Use between two and four sine waves, so that the individual sine waves are still somewhat visible in the sum. Perform a Fourier analysis (you can use the `fft` function) on the resulting time series and plot the power structure. Confirm that your code is correct by comparing the frequencies with nonzero power to the frequencies of the sine waves that you generated. Now try adding random noise to the signal before computing the Fourier transform. First, add a small amount of noise so that the sine waves are still visually recognizable. Next, add a large amount of noise so that the sine waves are no longer visually recognizable in the time domain data. Perform a Fourier analysis on the two noisy signals and plot the results. What is the effect of a small and a large amount of noise in the power spectrum? Are the sine waves with noise easier to detect in the time domain or in the frequency domain, or is it equally easy/difficult to detect a sine wave in the presence of noise?

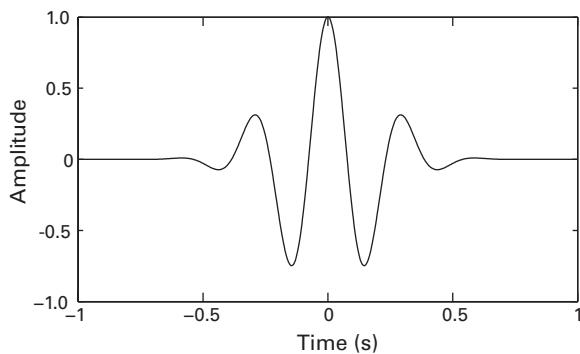
12 Morlet Wavelets and Wavelet Convolution

In the previous chapter you learned that the Fourier transform can be used to obtain a frequency-domain representation of the EEG data. Two limitations of the Fourier-transform-based frequency representation were also discussed: changes in frequency structure over time are difficult to visualize, and EEG data violate the stationarity assumption of Fourier analysis. These two limitations provide motivation to apply time-resolved frequency decomposition of EEG data. The next six chapters introduce you to several of the most commonly used methods for obtaining time-resolved frequency representations (or time-frequency representations) of EEG data. A time-frequency representation of EEG retains advantages of both the time and frequency domains while making only small and—depending on your experiment and hypotheses—possibly inconsequential sacrifices in temporal and frequency precision. This and the next chapters introduce Morlet wavelet convolution. Jean Morlet, for whom this procedure is named, was a French geophysicist, and so “Morlet” can be pronounced either as “more let” or, for the French-oriented readers, “more lay.” The Morlet wavelet is also sometimes called a Gabor wavelet (Sinkkonen, Tiitinen, and Naatanen 1995). An example Morlet wavelet is shown in figure 12.1.

A Morlet wavelet looks like a sine wave in the middle but then tapers off to zero at both ends. Morlet wavelets are useful for localizing changes in the frequency characteristics over time. Before learning how to use wavelets for time-frequency analyses, it is important first to learn why wavelets are useful and how to create them.

12.1 Why Wavelets?

The reason the Fourier transform does not show whether and how the frequency characteristics change over time is that the kernel used in the Fourier transform (a sine wave) has no temporal localization; the amplitude of the sine wave continues to fluctuate over its entire time series (theoretically, from negative infinity to positive infinity). From inspecting

**Figure 12.1**

A Morlet wavelet, which is created by windowing a sine wave by a Gaussian.

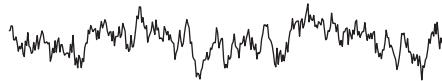
In figure 12.2, it should be sensible that to obtain temporally localized frequency information, the dot product with the EEG signal should be computed with only part of a sine wave, specifically the part of a sine wave that occurs at a specific window of time.

What is the best way to temporally localize dynamic patterns of the frequency structure in a time series? You might initially think that using one sine wave cycle is good (figure 12.2C). Unfortunately, this is not the best option because although the temporal precision is good, the frequency precision is poor. This is related to the trade-off between time and frequency precisions and is discussed in more detail in chapter 13. Another possibility would be to have a “box-car” window around the sine wave (figure 12.2D). This is also a poor choice because it weights all data points in the box-car area equally and because it will produce edge artifacts in the data (due to edges in the sine wave). A good option is to use a Gaussian taper to window the sine wave (figure 12.2E). Gaussian windows have no sharp edges that produce artifacts, dampen the influence of surrounding time points on the estimate of frequency characteristics at each time point, and allow you to control the trade-off between temporal precision and frequency precision. A sine wave windowed with a Gaussian is called a Morlet wavelet.

Morlet wavelets are not the only kinds of wavelets. There are many different kinds of wavelets, and you can even create your own wavelets. Wavelets must have values at or very close to zero at both ends, and they must have a mean value of zero. However, not all wavelets are useful for time-frequency decomposition of EEG data. In this book only Morlet wavelets are used because they are well suited for localizing frequency information in time.

There is one final reason to use wavelets. In the previous chapter it was stated that the Fourier transform assumes stationarity of the signal. This assumption is violated in EEG data,

A) EEG data



B) No temporal weighting (Fourier transform)



C) Strong temporal weighting



D) Boxcar temporal weighting



E) Gaussian temporal weighting

**Figure 12.2**

In order to extract time-varying frequency-specific information from EEG data (A), the data must be convolved with a sine wave. Without tapering of the sine wave (B), the result reflects frequency-specific information from the entire time series. Use of only one cycle (C) maximizes temporal precision but at the expense of frequency precision. A uniform boxcar tapering (D) is suboptimal because of decreased temporal specificity and potential artifacts from sharp edges. A Gaussian tapering (E) (also known as a Morlet wavelet) provides an adequate balance between temporal and frequency precision without introducing edge artifacts.

in which the frequency structure of the signal changes over time. The stationarity assumption of wavelet convolution is that the signal is stationary only during the time period in which the wavelet looks like a sine wave. This is a more reasonable assumption that is likely to be valid: EEG data often remain stationary for hundreds of milliseconds (Florian and Pfurtscheller 1995; Jeong, Gore, and Peterson 2002). Keep in mind that violations of local stationarity do not invalidate the results of a wavelet convolution, although they will decrease the accuracy of the frequency information, as illustrated in figure 11.9.

From figure 12.2E, it is apparent that the frequency information you obtain at each time point is a weighted sum of the frequency information of surrounding time points, with the weight decreasing with increasing distance away from the center of the wavelet. This should be kept in mind when interpreting time-frequency results: the activity at each time point is an estimate of the instantaneous activity and is influenced by activity from neighboring time points. This is not a unique feature of Morlet wavelet convolution but instead is a feature of all time-frequency decomposition methods, including temporal filtering.

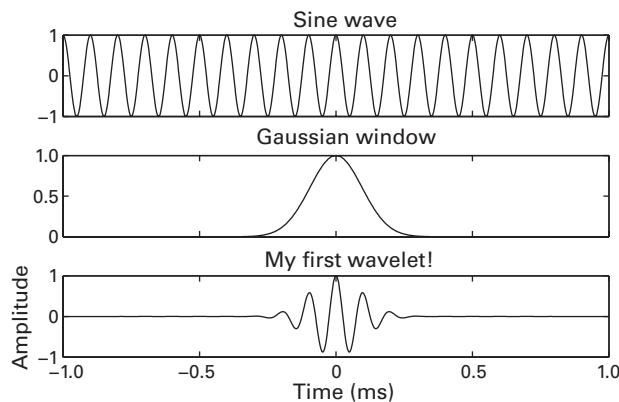


Figure 12.3

A Morlet wavelet (bottom) is a sine wave (top) windowed by a Gaussian (middle). This is called a real-valued Morlet wavelet because it comprises all real numbers. Complex Morlet wavelets are discussed in chapter 13.

12.2 How to Make Wavelets

To make a Morlet wavelet, create a sine wave, create a Gaussian, and multiply them point by point (figure 12.3). The sine wave and the Gaussian must have the same number of time points and the same sampling rate (this also needs to be the same sampling rate as the EEG data). The frequency of the wavelet is the frequency of the sine wave. The frequency of a Morlet wavelet is called its *peak frequency*. Sometimes *center frequency* is also used, but the term peak frequency is preferred here because wavelets, unlike sine waves, do not contain energy in only one frequency band but, rather, in a range of frequency bands for which the frequency of the sine wave is the peak. This will be clearer in chapters 13 and 14.

You already know from the previous chapter how to create sine waves. The Gaussian window (also sometimes called a bell-shaped or normal curve) is created according to the following formula:

$$GaussWin = ae^{-(t-m)^2/(2s^2)} \quad (12.1)$$

The variable a refers to amplitude (the height of the Gaussian), t is time, m is an x -axis offset (this is not relevant for EEG analyses and can always be set to zero and thus left out of the equation), and s is the standard deviation or the width of the Gaussian. The standard deviation of the Gaussian is defined according to equation 12.2:

$$s = \frac{n}{2\pi f} \quad (12.2)$$

The variable f is frequency (in hertz), and n refers to the number of wavelet cycles. The parameter n defines the trade-off between temporal precision and frequency precision. It is a nontrivial parameter that has implications for what kinds of results you can obtain from the data, and it should be carefully selected when you perform your analyses. Appropriate values and ranges for this parameter are discussed in depth the next chapter; for this chapter, a value of 6 is used.

Similar to the way a Fourier transform uses many sine waves of different frequencies, time-frequency decomposition via wavelet convolution involves using many wavelets of different frequencies. Unlike Fourier analysis, however, the frequencies of the wavelet can be specified by you rather than being specified by the number of data points in the time series. Furthermore, the number of wavelets that can be used is not constrained—you can use as many wavelets as you want, and those wavelets can have almost any frequency you specify. A group of wavelets that share the same properties but differ in frequency is called a family of wavelets. There are a few theoretical and practical limits to constructing a family of wavelets:

1. You cannot use frequencies that are slower than your epochs. That is, if you have 1 s of data, you cannot analyze activity lower than 1 Hz. In practice, you should have several cycles of activity (for example, if you have 1 s of data, use wavelets that are 4 Hz and faster).
2. The frequencies of the wavelets cannot be above the Nyquist frequency (one-half of the sampling rate).
3. Because of frequency smoothing from time-frequency precision trade-offs (this is discussed and demonstrated in the next chapter), frequencies that are very close to each other will likely provide similar or nearly identical results. For example, if you have a wavelet at 15.0 Hz, a wavelet at 14.9 Hz is unlikely to provide any unique information. More frequency bins may produce smoother-looking plots but will also increase computation time without increasing the information in the results. In general, between 15 and 30 frequencies (spanning, say, 3 Hz to 60 Hz) should be sufficient for most experiments.

Figure 12.4 shows a few individual wavelets and a time-frequency plot of all wavelets in this family.

In figure 12.4B the y -axis corresponds to the peak frequency of the members of the wavelet family, the x -axis corresponds to time, and the color of the plot corresponds to the amplitude of the wavelet (here in grayscale, but on your computer screen, red for positive deflections of the wavelet and blue for negative deflections). This plot is conceptually similar to the ERP-image introduced in chapter 9 (figure 9.6), in that many individual time courses are

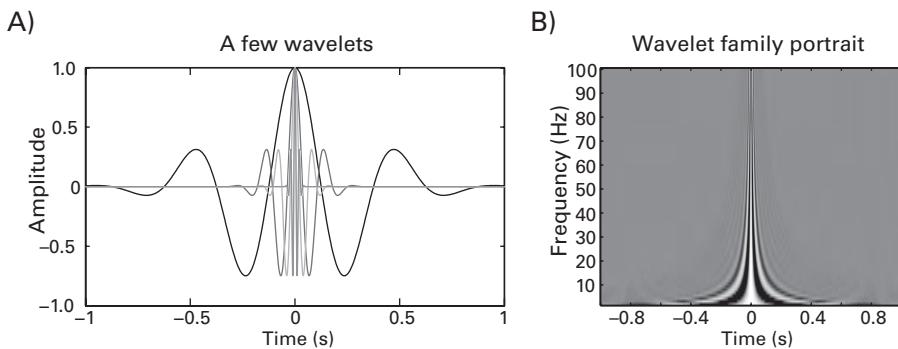


Figure 12.4

Different members of this wavelet family were created by changing the frequency of the sine wave while leaving other parameters unchanged.

“rotated,” stacked on top of each other, and then color coded for visualization. If you run the online Matlab code that generates figure 12.4, you will be able to see a movie of a 2-D colored wavelet turning into one line of an image of a family of wavelets. This should help you conceptualize how a wavelet becomes one line in the wavelet image plot in figure 12.4B. After the movie finishes, you can left-click on the image to spin it around in three dimensions. This is an important concept because this same principle of converting ups and downs of a time series into colors on a line is also used to create time-frequency maps.

12.3 Wavelet Convolution as a Bandpass Filter

You learned in chapters 10 and 11 about convolution and how convolution can be interpreted as a vector that expresses the time-varying mapping between a kernel and a signal (here: the wavelet and the EEG data). Figure 12.5 shows some raw EEG data and how those data look when convolved with a Morlet wavelet that has a peak frequency of 6 Hz. For comparison, the same data are shown having applying a bandpass filter from 4 Hz to 8 Hz.

Figure 12.5 shows that convolving EEG data with a wavelet at a specific frequency is similar to bandpass filtering the data around that same frequency. In fact, it is shown in chapter 14 that a Morlet wavelet is a special case of a bandpass filter in which the frequency response is Gaussian-shaped. In other words, wavelet convolution is bandpass filtering. To see why this is the case, consider figure 12.6, which is similar to figure 11.12 but uses a Morlet wavelet with a peak frequency of 6 Hz instead of a Gaussian. You can see that the spectral representation of a Morlet wavelet has a Gaussian shape around the peak frequency of the wavelet

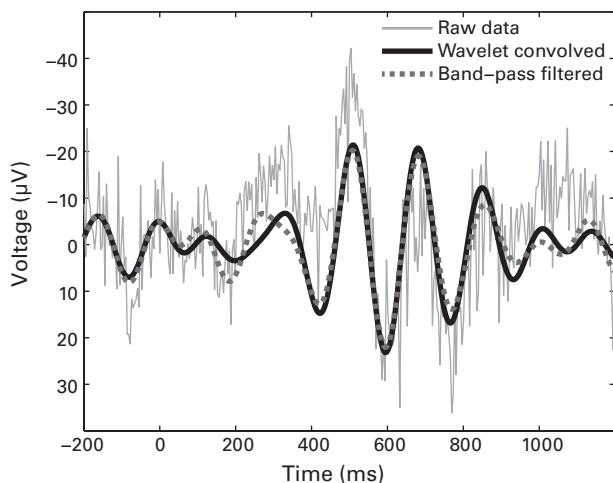


Figure 12.5

One trial of EEG data plotted before any filtering (gray line), after being bandpass filtered between 4 and 8 Hz (dashed line), and after being convolved with a 6-Hz Morlet wavelet (black line). This shows that wavelet convolution and bandpass filtering produce similar results (minor discrepancies are due to wavelet and filter characteristics and could be further minimized).

(figure 12.6B). When the frequency spectrum of the wavelet is multiplied by the frequency spectrum of the EEG data and the inverse Fourier transform is computed, the resulting time series contains the frequency characteristics of the EEG data that are tapered by the frequency characteristics of the wavelet (that is, a Gaussian around the peak frequency). This is why convolution with a wavelet is bandpass filtering.

12.4 Limitations of Wavelet Convolution as Discussed Thus Far

There are two limitations with real-valued Morlet wavelet convolution as it has been presented in this chapter. First, figures 12.5 and 12.6 show that convolution with a Morlet wavelet acts as a bandpass filter, but for time-frequency analyses, power and phase information are needed, and these features of the data are not readily apparent in the bandpass-filtered signal. (In theory, this problem can be addressed by taking the Hilbert transform of the bandpass-filtered signal, as discussed in chapter 14, but there is a simpler and more convenient solution presented in chapter 13.)

Second, the result of convolution with a Morlet wavelet depends on phase offsets between the wavelet and the data. Figure 12.7 illustrates this issue. Figure 12.7A–C shows a wavelet

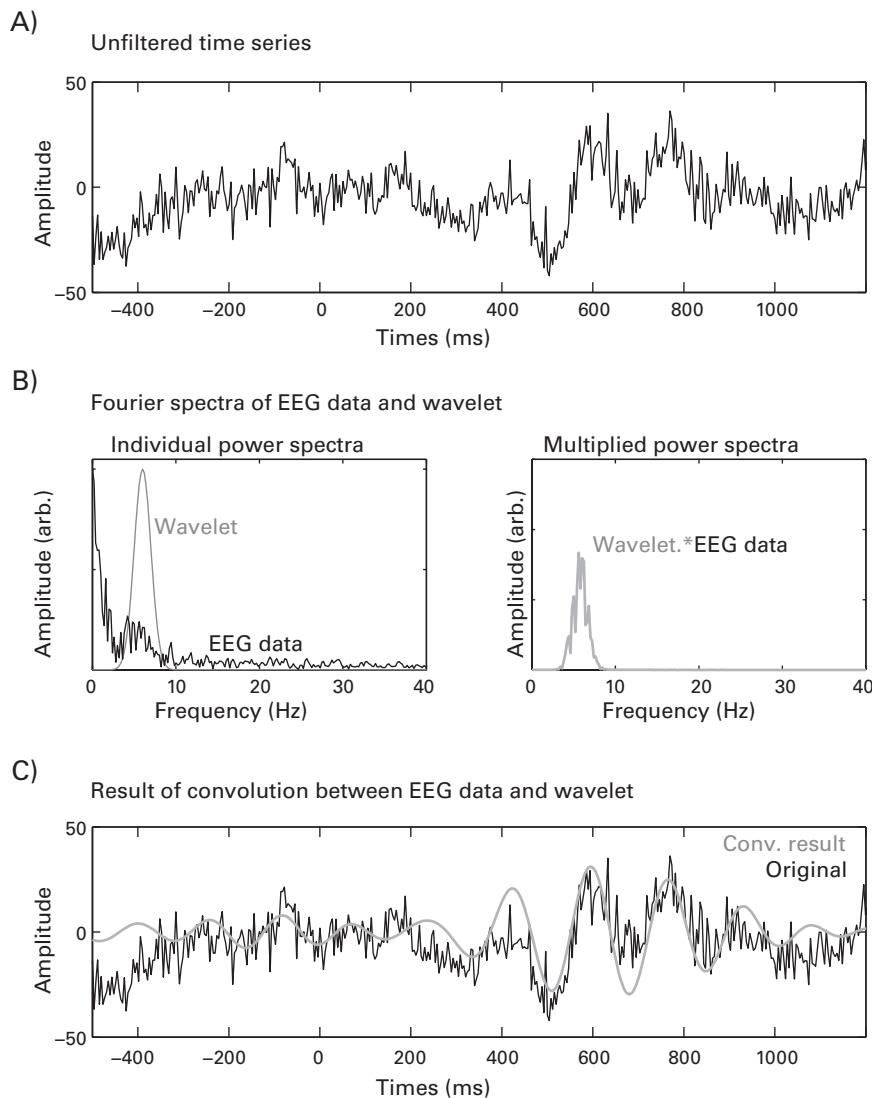
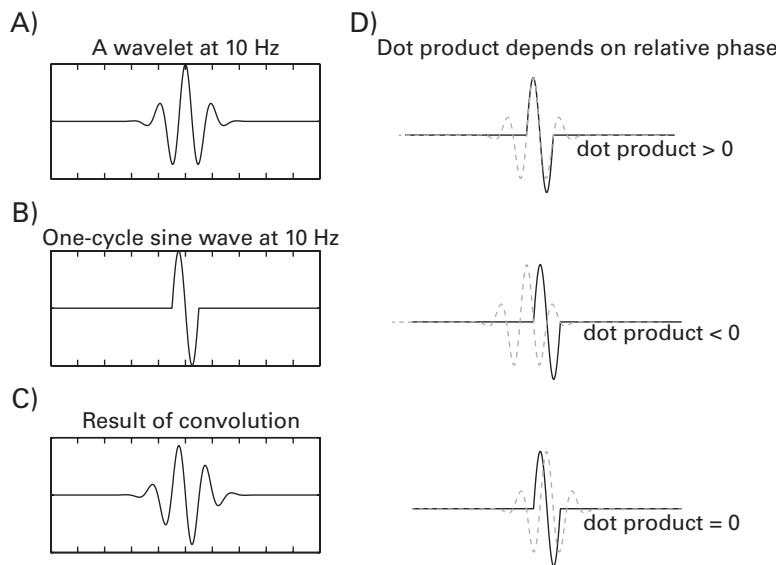
**Figure 12.6**

Illustration of why wavelet convolution acts as a bandpass filter. Panel A shows raw EEG data from one trial from electrode FCz. Panel B shows the power spectra of the Morlet wavelet and the EEG data. Note that the wavelet has a Gaussian shape in the frequency domain. The left-hand plot in panel B shows the point-by-point multiplication of the frequency spectra of the wavelet and the EEG data. Panel C shows the result of convolution overlaid on top of the original EEG data. This figure also illustrates how the result of convolution reflects activity that is maximal at the peak frequency of the wavelet (here, 6 Hz), but also activity from a weighted combination of surrounding frequencies (here ranging from around 3 Hz to 9 Hz).

**Figure 12.7**

The result of each step of convolution (the dot product between the wavelet and the data) depends on the phase relationship between the kernel and the signal at that time point. This issue is resolved by using complex wavelets, as discussed in chapter 13.

with a peak frequency of 10 Hz, a one-cycle sine wave at the same frequency as the wavelet, and the result of convolution between them. Panel D shows the result of the dot product between the wavelet and the one-cycle sine wave at different points of overlap (i.e., different phase offsets). Although the sine wave should be very similar to the wavelet—in fact, they are both made from the same sine wave; they just have different taperings—the result of convolution seems to indicate that there are points in time where the two vectors are orthogonal (dot product of zero) and other points in time where the two vectors have a negative dot product.

Intuitively, this seems to make little sense: if the two signals are so similar to each other, why do their dot products sometimes indicate otherwise and only at particular phases? You can see in figure 12.7D why this happens: The dot product is zero when there is a 90° phase lag between the wavelet and the one-cycle sine wave, and the dot product is negative when there is a 180° phase lag between the two signals. Thus, if you want to know how much energy there is in the EEG signal at a specific frequency band and at a specific time point, you would have to align the wavelet so that it has a 0° phase lag with the EEG data at the time point of interest and then compute the dot product. This is not what you want—you want to

determine the relationship between the wavelet and the EEG data at all time points and all phase lags, not just at some phase lags.

In order to resolve both of the limitations of real-valued Morlet wavelets, EEG data are convolved with *complex Morlet wavelets*—wavelets that have both a real component and an imaginary component. With complex Morlet wavelet convolution, the mapping between the two vectors does not depend on phase lags but, rather, is represented in a 2-D space that allows you to extract not only the bandpass-filtered signal as in figure 12.5 but also time-frequency power and phase information. In the next chapter, you learn what a complex wavelet looks like, then you learn about imaginary numbers, the complex plane, and Euler’s formula, and, finally, you will come to understand why complex wavelets are necessary and how to use them to extract estimates of power and phase from EEG data.

12.5 Exercises

1. Create a family of Morlet wavelets ranging in frequency from 2 Hz to 30 Hz in five steps.
2. Select one electrode from the scalp EEG dataset and convolve each wavelet with EEG data from all trials from that electrode. Apply the Matlab function `real` to the convolution result, as in `convol_result=real(convol_result)`. This will return the EEG data bandpass filtered at the peak frequency of the wavelet. You learn more about why this is in the next chapter.
3. Average the result of convolution over all trials and plot an ERP corresponding to each wavelet frequency. Each frequency should be in its own subplot.
4. Plot the broadband ERP (without any convolution). Thus, you will have six subplots in one figure. How do the wavelet-convolved ERPs compare with the broadband ERP? Are there dynamics revealed by the wavelet-convolved ERPs that are not apparent in the broadband ERP, and are there dynamics in the broadband ERP that are not apparent in the wavelet-convolved ERPs? Base your answer on qualitative visual inspection of the results; statistics or other quantitative comparisons are not necessary.

13 Complex Morlet Wavelets and Extracting Power and Phase

13.1 The Wavelet Complex

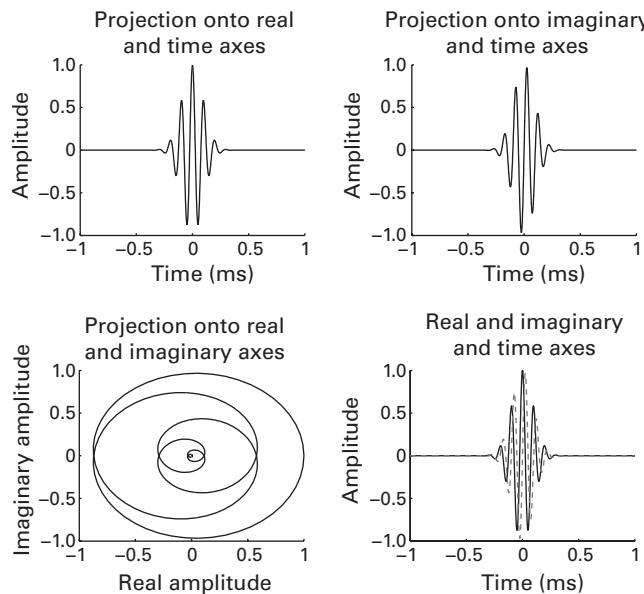
In the previous chapter you saw that Morlet wavelet convolution can be used as a bandpass filter for EEG data. In this chapter you will learn how to use complex Morlet wavelets to extract estimates of time-varying frequency band-specific power and phase from EEG data.

To extract power and phase information from EEG data, *complex* Morlet wavelets are necessary. They are called complex because they have a real part and an imaginary part. A complex wavelet thus occupies a three-dimensional (3-D) space: time, real, and imaginary (figure 13.1). You can conceptualize real and imaginary dimensions the way you conceptualize x and y planes on a Cartesian graph.

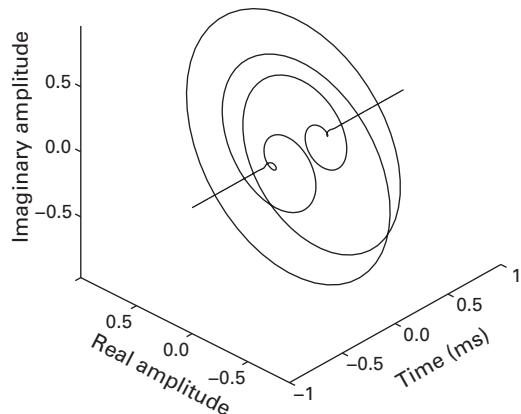
After examining figure 13.1, try to conceptualize wavelets not as curvy lines that go up and down but rather as corkscrews that go around in a circle over time. This is illustrated in figure 13.2. If you run the online Matlab code to generate figure 13.2, you will be able to rotate the wavelet on all three dimensions with the mouse. The online Matlab code will also generate a movie that allows you to see both the Cartesian and the polar representations of a complex wavelet evolving over time.

Now that you appreciate what a wavelet looks like visually, it is time to take a closer look at the numbers. First examine the size of the wavelet. If you type `size(wavelet)` in the Matlab command window, you will see that the variable `wavelet` is a 1×1001 matrix, in other words, a vector, or a single stream of numbers. However, if you look at the contents of the variable `wavelet` by typing `wavelet'` (the transpose will print a column vector for visual convenience), you can see that each element in the wavelet actually comprises two numbers, the second of which has an “*i*” behind it. What does this mean? Is this the new iWavelet?

This is the representation of a complex number. A complex number has the form “ $a + ib$.” The first part of the number (a) is the real component, and the second part of the number

**Figure 13.1**

A complex wavelet is a 3-D (time, real, imaginary) function. Plotted here are projections onto various pairs of those dimensions.

**Figure 13.2**

Three-dimensional view of a complex wavelet.

(b) is the imaginary component. The i is the imaginary operator; multiplying a number by i makes that number imaginary. Over the next few pages you will learn how to work with complex numbers. Complex numbers are fundamental to most time-frequency decomposition methods. Thus, the better you understand how to work with complex numbers, the more flexibly you will be able to perform, adapt, and develop advanced EEG data analyses.

13.2 Imagining the Imaginary

If you remember middle school, you may remember your math teacher telling you that you can only take the square root of positive numbers (and preferably of convenient numbers such as 4, 16, and 25) because the square root of a number X is a number that, when multiplied by itself, gives you X . You cannot take the square root of a negative number because no number (even a negative number) times itself can be negative. The imaginary number, however, is the square root of -1 . That is, a number multiplied by itself that gives you -1 . Do not worry too much about the interpretation of an imaginary number per se; try to focus on how imaginary numbers are used to extract power and phase information from EEG data.

The imaginary number is indicated with an i or a j . Mathematicians tend to use i and engineers tend to use j to differentiate it from current. Matlab accepts both. Matlab also accepts $1i$ and $1j$, which allows you to use i and j as looping indices or other variables.

13.3 Rectangular and Polar Notation and the Complex Plane

Polar notation is a mathematical framework that is useful for characterizing circular and spherical data. For EEG data analyses, polar notation provides a convenient and succinct language to describe properties of frequency-band-specific activity, including the bandpass-filtered signal, power, and phase.

Rectangular notation (also called Cartesian notation) is what you probably think of when you think about points on a 2-D plane described by x and y coordinates. Consider a complex number as a location on a 2-D plane, except that this plane comprises a real axis and an imaginary axis instead of an x -axis and a y -axis. The two parts of the complex number tell you where that point is on the plane. Consider, for example, the complex number $4 - 8i$. The real component of this number is 4, and the imaginary component is -8 . On a complex plane this number refers to a point that is 4 units over on the real axis and 8 units down on the imaginary axis (figure 13.3A).

Converting between rectangular and polar notations is straightforward, but you might need to remember those high school trigonometry classes that you tried so hard to forget.

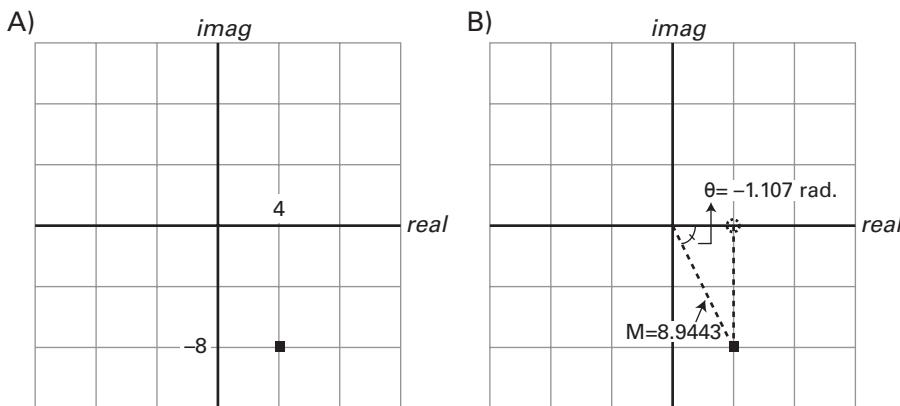


Figure 13.3

The same point in a complex space (with a real axis and an imaginary axis) can be represented using Cartesian (A) or polar (B) notations.

The point on the complex plane obtained by going 4 units to the right and 8 units down from the origin can also be located by knowing $M = 8.9443$ and $\theta = -1.107$ radians. Where do those numbers come from, and what do they mean?

Imagine drawing a right triangle that would include the origin (zero on both axes) and the complex number as two of the points. The third point would be the projection onto the real axis at zero on the imaginary axis (see dotted lines in figure 13.3B). The Pythagorean theorem states that the hypotenuse of a right triangle (here, the line from the origin to the point representing the complex number) is the square root of the squared lengths of the opposite and adjacent lines. In other words, 4^2 (real part squared) is 16, -8^2 (imaginary part squared) is 64, their sum is 80, and the square root of that is about 8.9443. That is how you compute the hypotenuse, or magnitude (M), of the line. This number represents the distance from the origin to the point represented by the complex number. θ is the angle formed by the hypotenuse and the positive (right side of the plot) real axis and can be computed as the arctangent of the length of the opposite over the length of the adjacent (there are other equivalent trigonometric methods to compute this angle). Using M and θ to locate a complex number on a 2-D plane is polar notation.

The magnitude can never be less than zero. The two components (real and imaginary) are squared, and the sum of squared numbers cannot be negative. This should also make intuitive sense—the length of a line from zero to any point cannot be negative, even when the points themselves have negative coordinates.

More formally, the following equations can be used to obtain the magnitude and angle in polar notation when you already have the rectangular notation.

$$M = \sqrt{(\text{real}^2 + \text{imag}^2)} \quad (13.1)$$

$$\theta = \arctan(\text{imag} / \text{real}) \quad (13.2)$$

real and *imag* refer to the real and imaginary components of the complex number. The Matlab function `atan` computes the arctangent and has an exception for when the real part (the denominator) is zero, which prevents θ from being infinity.

You can also obtain the real and imaginary parts in rectangular notation when you already have the polar notation.

$$\text{real} = M \cos(\theta) \quad (13.3)$$

$$\text{imag} = M \sin(\theta) \quad (13.4)$$

You can see that the real part of a complex number corresponds to a cosine, and the imaginary part of a complex number corresponds to a sine. In the following equations the individual components shown in equations 13.3 and 13.4 are combined and then reorganized using algebra. Equation 13.7 shows how to convert a complex number between the rectangular and polar notations.

$$\text{real} + \text{imag} = M \cos(\theta) + M \sin(\theta) \quad (13.5)$$

$$\text{real} + \text{imag} = M [\cos(\theta) + \sin(\theta)] \quad (13.6)$$

$$a + ib = M[\cos(\theta) + i \sin(\theta)] \quad (13.7)$$

13.4 Euler's Formula

In addition to the Fourier theorem, Euler's formula is an important tool for time-frequency decomposition that forms the backbone of many advanced EEG data analysis techniques. Euler's formula allows you to represent complex numbers as points on a circle. The formula follows from the polar notation of a complex number:

$$Me^{i\theta} = M[\cos(\theta) + i \sin(\theta)] \quad (13.8)$$

where e is the base of the natural logarithm. It is a constant that begins 2.71828. . . . The Matlab function `exp` computes the exponential of the input [that is, `exp(3)` is e^3]. θ is any real number (in this book, θ will always represent an angle in radians). You can see that $Me^{i\theta}$

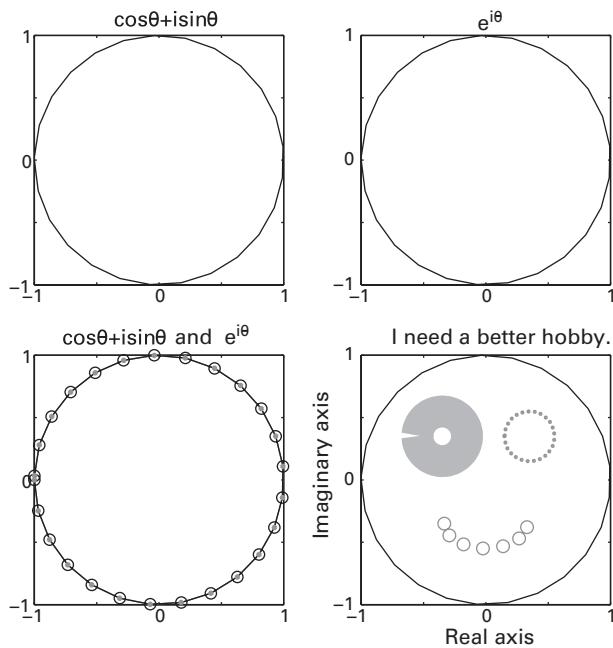


Figure 13.4

Illustration of equality of Euler and trigonometric representations and another illustration (lower right panel) of how the real component corresponds to cosine while the imaginary component corresponds to sine.

is a convenient notation for representing a point in a complex plane and thus can also be used to represent a vector in a complex plane defined by the origin and that point. M is the magnitude of that vector, and θ is its angle. Thus, the complex number $4 - 8i$ can be represented using Euler's formula as $8.9443e^{-1.107i}$.

It is worth showing that the equality of the two sides of equation 13.8 is true. Figure 13.4 shows circles that were generated using trigonometric and Euler notations. If you inspect the Matlab code that generated this figure, you can see that in the code that generates the lower right subplot of figure 13.4, changing the cosine part in the equation affects the position along the real axis, whereas changing the sine part in the equation affects the position along the imaginary axis.

If you look at the imaginary part of the wavelet in figure 13.1, you will notice that it looks a bit lopsided. This is because the real part of the wavelet corresponds to a cosine wave, and the imaginary part of the wavelet corresponds to sine wave. The difference between cosine and sine is one-quarter of a cycle, or $\pi/2$. This is illustrated in figure 13.5, which shows the

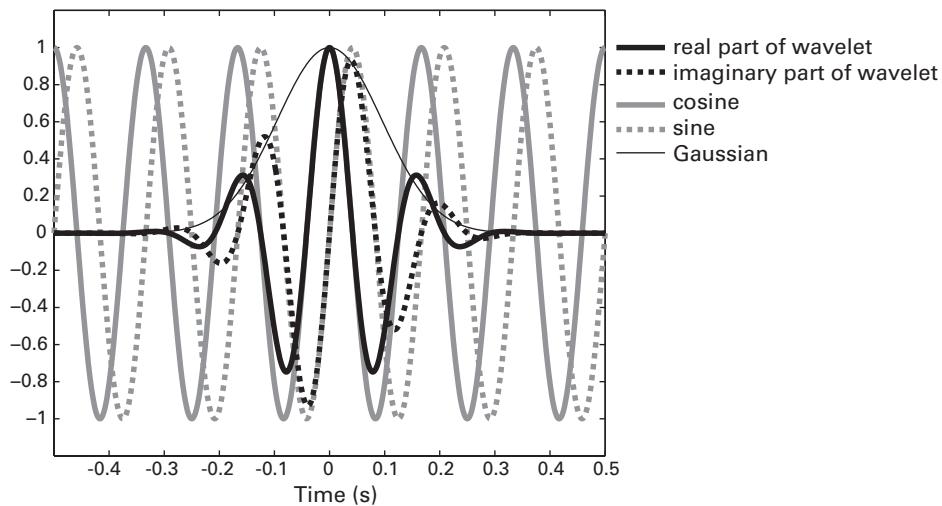


Figure 13.5

Overlaying the real and imaginary parts of a complex wavelet along with a cosine wave and a sine wave. This shows that the real part of the wavelet corresponds to a cosine wave, and the imaginary part of the wavelet corresponds to a sine wave.

real and imaginary parts of a wavelet, as well as cosine and sine waves, and a Gaussian. Thus, the difference between the real and the imaginary parts of a complex wavelet is the difference between a cosine and a sine. Furthermore, cosine and sine are related to each other as a one-quarter counterclockwise rotation in complex space. That is, in the complex plane, when you rotate from the real axis to the imaginary axis, a cosine becomes a sine. This feature becomes relevant for understanding how the Hilbert transform (chapter 14) works.

After learning about polar notation, complex numbers, and Euler's formula, it is now possible to understand the equation for a complex Morlet wavelet. Remember from chapter 12 that a real-valued Morlet wavelet is created by multiplying a sine wave by a Gaussian; a complex Morlet wavelet (cmw) is created in the same way, except that the sine wave is a complex sine wave.

$$cmw = Ae^{-t^2/2s^2} e^{j2\pi ft} \quad (13.9)$$

$$A = \frac{1}{(s\sqrt{\pi})^{1/2}} \quad (13.10)$$

The first part of equation 13.9 (the first exponential) is a Gaussian and should look familiar from equation 12.2 (some of the unused components of the Gaussian formula, for example

the x -axis offset, are omitted here for simplicity). The second part of the equation (the second exponential) is a complex sine wave. You should recognize that this complex sine wave is formed by combining Euler's formula and the $2\pi ft$ part of a sine wave. That is, the θ in equation 13.8 is replaced with $2\pi ft$, in which t is time. The s in equations 13.9 and 13.10 is the standard deviation of the Gaussian (defined in equation 12.2), and f is the peak frequency of the wavelet. A is a frequency band-specific scaling factor. If you plan to apply a baseline normalization such as percentage change or decibel, this scaling factor is not necessary (more on this in chapter 18). Furthermore, if you are performing complex wavelet convolution only to obtain the phase angle time series, the scaling factor is also not necessary. It is possible to shorten equation 13.9 by adding the exponents (because the base numbers are the same), although keeping them separate facilitates interpretation.

If you look back at equation 11.2, you can see that the formula for the discrete Fourier transform contains Euler's formula. Thus, to perform a Fourier transform, a dot product is computed between the time series data and a complex sine wave. It was mentioned several times in this chapter that a complex sine wave comprises both a cosine (real) and sine (imaginary) component. Thus, the formula for the Fourier transform can also be written and computed using two equations, one for sine and one for cosine (as shown by Euler's formula, see equation 13.8).

13.5 Euler's Formula and the Result of Complex Wavelet Convolution

In section 12.4 you saw that the dot product between a wavelet and a one-cycle sine wave could be negative, positive, or zero, depending on slight shifts in relative phase between the signal and the kernel (figure 12.7). It was also mentioned that this seemingly counterintuitive feature of the dot product would be resolved by using complex wavelets. This problem can now be revisited.

When you compute the dot product between a complex wavelet and a signal, the result of the dot product is a complex number; it contains both a real part and an imaginary part because the complex wavelet has a real part and an imaginary part. Euler's formula can be applied to represent that complex number as a point in polar space, and that point can be further conceptualized as the endpoint of a vector from the origin of the polar space to that point.

This is shown in figure 13.6. Figure 13.6A shows that dot product will be computed between a complex wavelet (only the real part is shown here) and a one-cycle sine wave, and the phase lag between them will vary. The polar plots in figure 13.6B show the results of the dot products as vectors in polar space from the origin to the complex number that resulted

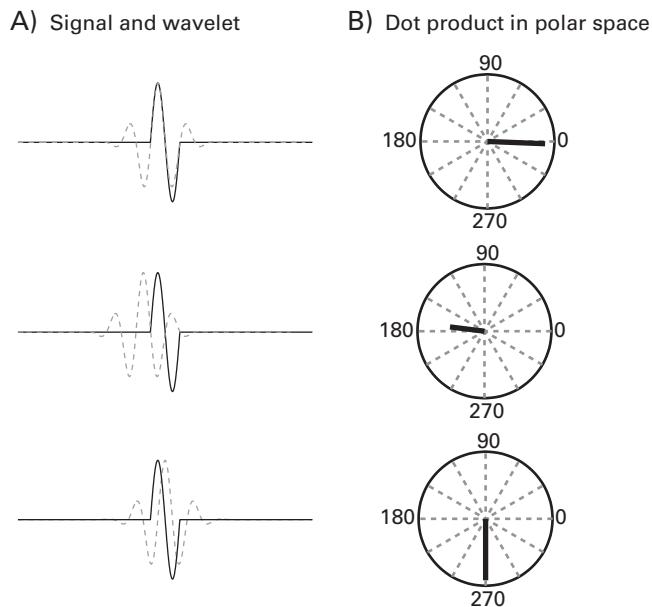


Figure 13.6

The dot product (one step of convolution) between a complex Morlet wavelet (real part shown here using a dotted line) and a one-cycle sine wave produces a complex number, here represented as a vector in a polar plot using the magnitude and angle. Note that the more the wavelet and the one-cycle sine wave overlap, the longer the vector is in complex space, regardless of the phase angle of that vector. This figure can be compared with figure 12.7.

from the dot product of the complex wavelet and the piece of the sine wave. In these plots the real axis is a line from 180° to 0° , and the imaginary axis is a line from 270° to 90° .

Now you can directly compare the results here with the results presented in figure 12.7. Notice that when the dot product with the real-valued wavelet from figure 12.7D was less than zero, the dot product with the complex wavelet has the vector pointing to the left (negative on the real axis and close to zero on the imaginary axis). Notice further than when the dot product with the real-valued wavelet was zero, the dot product with the complex wavelet has a vector pointing down (zero on the real axis and negative on the imaginary axis). Finally, when the dot product with the real-valued wavelet was positive, the dot product with the complex wavelet has a vector pointing toward the right (positive on the real axis and zero on the imaginary axis). In other words, the result of dot products with the real-valued wavelet maps onto the real axis in the result of dot products with complex wavelets; the imaginary axis is ignored in the dot product with the real-valued wavelet.

One final feature to notice in figure 13.6 is that the more the wavelet and the one-cycle sine wave overlap, the longer is the line in complex space. Conversely, in the middle row of figure 13.6, the wavelet and the one-cycle sine wave overlap relatively little, and the line in complex space is relatively short. This means that the length of the line—the vector from the origin to the point described by a complex number resulting from the convolution of a complex wavelet and a signal—provides information regarding the similarity or overlap between the kernel and the signal. Importantly, the length of this line does not depend on the phase relationship between the kernel and the signal. In contrast, the phase relationship between the kernel and the signal is characterized by the angle of the vector.

There are three pieces of information that can be extracted from the complex dot product (illustrated in figure 13.7). The first piece of information is the projection onto the real axis. This is the bandpass-filtered signal (figure 13.7D1) and was plotted in figures 12.5 and 12.6. This value can be positive or negative, depending on the phase relationship between the kernel and the signal (this was the limitation of real-valued Morlet wavelets shown in figure 12.7). You can also extract the projection onto the imaginary axis, but this is information not commonly used in EEG analyses.

The second piece of information that you can extract from the complex dot product is the magnitude of the vector from the origin to the point in complex space defined by the result of the dot product (equation 13.1 and figure 13.7D2). The length of this vector is related to the similarity or overlap between the kernel (in the case illustrated in figure 13.7, a 15-Hz complex Morlet wavelet) and the signal (EEG data). Thus, when the EEG data contain a lot of energy at 15 Hz, the result of a dot product with a 15-Hz complex wavelet will produce a point in complex space that is far away from the origin (the magnitude of the line is large). The length (magnitude) of this vector is called the amplitude, and the length squared is called the power. This is the estimate of the instantaneous power at the point in time corresponding to the center of the wavelet with respect to the EEG data, and at the peak frequency of the wavelet. In Matlab, you can use the function `abs` to extract the magnitude of the complex number.

The third piece of information that can be extracted is the angle of that vector with respect to the positive real axis. This angle is an estimate of the phase angle at the point in time corresponding to the center of the wavelet and at the peak frequency of the wavelet. In Matlab you can use the function `angle` to extract the phase angle of the complex number. Note that the power and phase values are called estimates because they are influenced by activity at neighboring time points.

A brief aside on computational efficiency: power can be extracted either by squaring the length of the complex vector [in Matlab: `abs(X)^2`] or by multiplying the complex vector

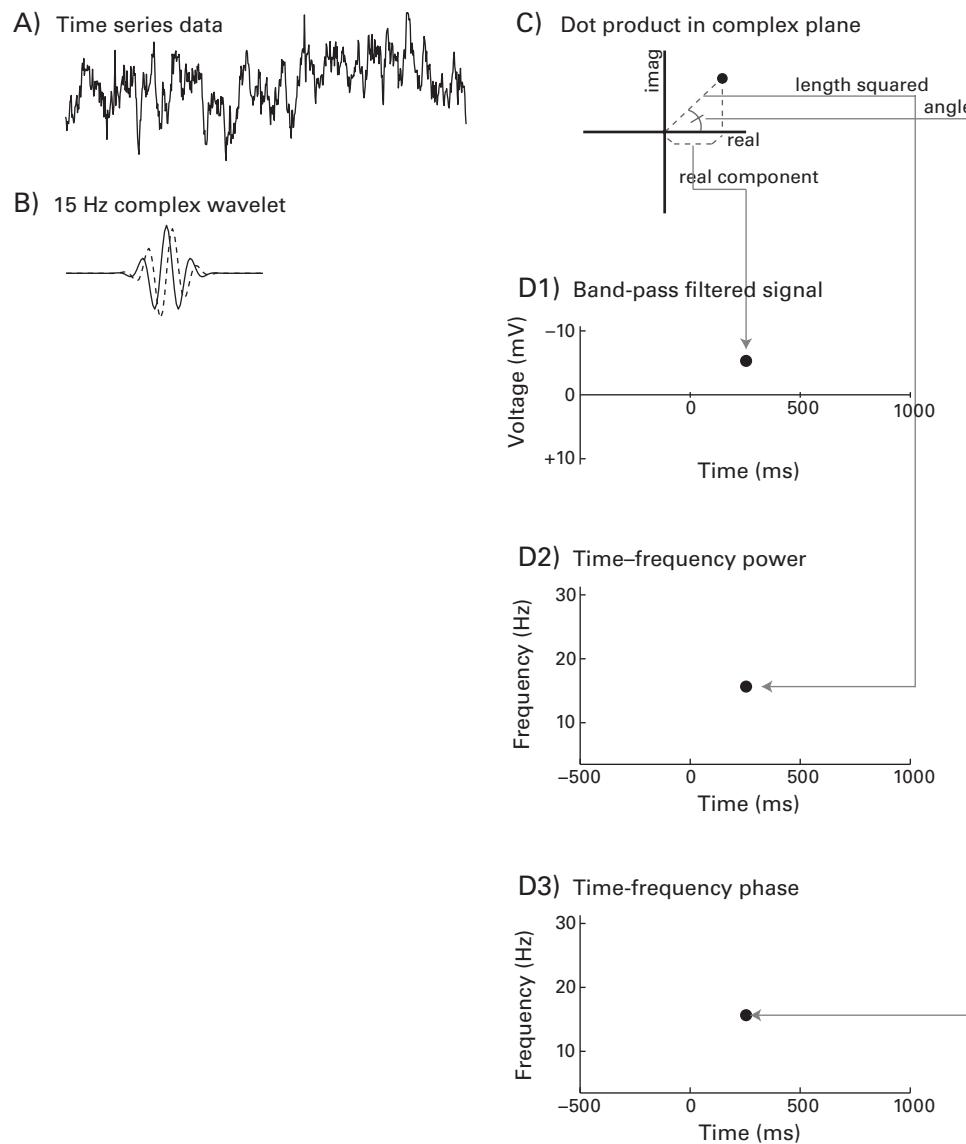


Figure 13.7

Graphical overview of how to extract the bandpass-filtered signal, power, and phase from one step of the result of complex convolution. Each step of convolution between the time series data (panel A) and a complex wavelet (panel B) is a dot product, which can be conceptualized as a point in a complex plane (panel C). The projection of that point onto the real axis is the bandpass-filtered signal (panel D1), the squared length of the vector from the origin to the dot product is power (panel D2), and the angle of the vector with respect to the positive real axis is the phase angle (panel D3). These values are then placed into time-frequency matrices at the frequency corresponding to the peak frequency of the wavelet and at the time point corresponding to the center time point of the wavelet and its corresponding temporal position in the EEG data (in the case illustrated here, the wavelet would be centered at 250 ms). Note that panels C and D illustrate general procedures for extracting power and phase from a complex signal resulting from any time-frequency decomposition method that produces an analytic signal, including the filter-Hilbert and short-time FFT methods.

by its conjugate [in Matlab: `x .* conj (x)`]. The conjugate of a complex number is obtained by multiplying the imaginary part by minus one. Thus, the conjugate of $a + ib$ is $a - ib$. The squared length of the complex vector is conceptually more straightforward and has an intuitive geometric interpretation, as shown in figure 13.7. This notation is thus preferred for learning about the mechanics of time-frequency decomposition. However, for large matrices, multiplying the complex vector by its conjugate is faster (around twice as fast, depending on the size of the matrix) and thus should be used in practice with real data.

13.6 From Time Point to Time Series

Remember that time-domain convolution involves computing time-sliding dot products between the kernel and the signal. The previous section thus illustrated one step of convolution between a complex Morlet wavelet and EEG data. Consider that when the M and θ in Euler's formula can vary over time ($M_t e^{i\theta_t}$), you can represent a complex sine wave as spiraling through a circle on a complex plane over time, similar to how the complex wavelet was like a corkscrew over time (figure 13.2). And by extracting the magnitude and phase angle of the result of convolution, you can construct a time series of power or phase values from one frequency band. This is shown in figure 13.8 for one trial of EEG data (dealing with many trials will be addressed later and in chapter 18). This process would then be repeated over multiple frequency bands to obtain a time-frequency map.

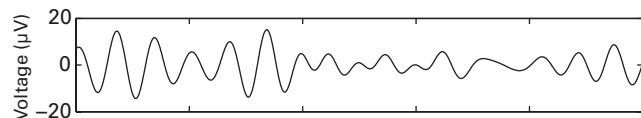
In figure 13.2, you saw that a complex Morlet wavelet is like a corkscrew over time, real, and imaginary axes. You can conceptualize the result of complex wavelet convolution the same way—existing in three dimensions of time, real, and imaginary. Thus, the results in figure 13.8 show projections onto two of the dimensions at a time. The 3-D representation of the result of that convolution can be seen in figure 13.9A.

The result of convolution can then be transformed to a new set of axes. Rather than time, real, and imaginary axes, you can represent the result of convolution in time, amplitude (or power), and phase axes. This is illustrated in figure 13.9B. The seemingly large discrete jumps on the phase axis are due to discontinuous borders in the plot between $-\pi$ and $+\pi$. Figure 13.9C,D shows projections onto two of the three dimensions. If you run the online Matlab code that generates this figure, you will be able to rotate the 3-D plots yourself with the mouse.

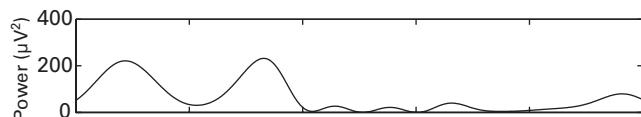
13.7 Parameters of Wavelets and Recommended Settings

Now that you know how to create and use complex Morlet wavelets, it is time to discuss parameters of wavelets and the consequences those parameters will have on your results.

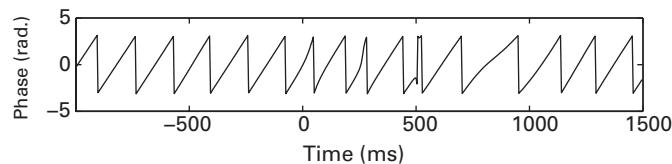
A) Projection onto real axis (filtered signal)



B) Squared magnitude of vector (power)



C) Angle with respect to positive real axis

**Figure 13.8**

Different pieces of frequency-band-specific information extracted from EEG via complex wavelet convolution.

Some of the parameters discussed below are also applicable to other time-frequency decomposition methods such as the filter-Hilbert method.

13.7.1 How Low Should the Lowest Frequency Be?

This depends on your expectations about the results, on the task design and timing of trial events, and on the length of your data epochs. If your study is hypothesis-driven and concerns only alpha-band activity, there is no need to analyze data down to 1 Hz; you could probably stop at, say, 5 or 6 Hz (it is a good idea to analyze activity a bit below and a bit above the alpha band to determine whether the effect is frequency band specific). You should have at least several cycles worth of time to extract robust band-specific information. Thus, if you have 1-s epochs, estimates of activity below 4 Hz will have relatively low signal-to-noise ratio.

13.7.2 How High Should the Highest Frequency Be?

This also depends on your expectations about the results as well as on the sampling rate. You cannot examine frequencies higher than the Nyquist frequency, and in practice, it is good

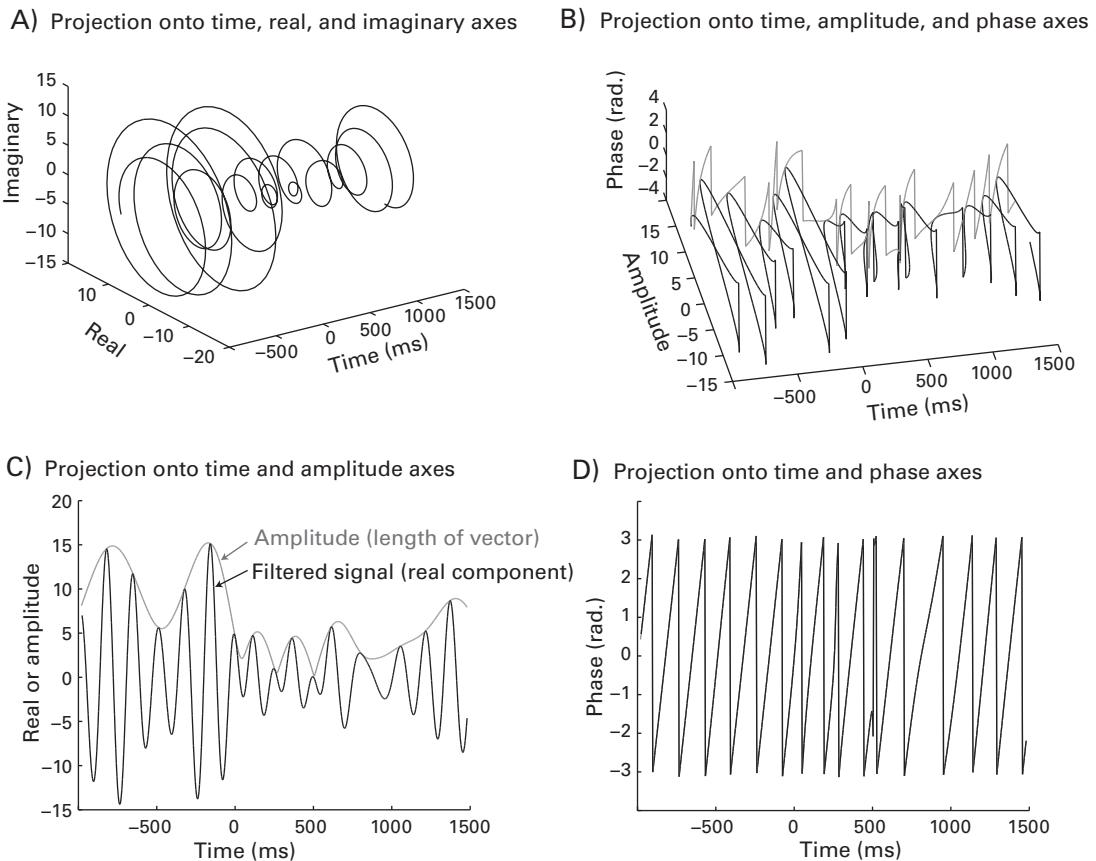


Figure 13.9

Try to conceptualize the result of complex Morlet wavelet convolution as a line in a 3-D space. After convolution, the three dimensions are time, real, and imaginary (panel A). For most EEG data analyses, the real and imaginary components are transformed to amplitude (or power) and phase (panel B). Panels C and D show the 3-D representation from panel B projected onto the time and amplitude axes (C) and the time and phase angle axes (D). Panel C also shows the real component from panel A, as a comparison with the amplitude. If you use the online Matlab code to generate this figure, you can freely rotate these images, which might help you conceptualize the 3-D nature.

to have many data points per cycle to increase signal-to-noise ratio (figure 6.2). For example, if the sampling rate is 500 Hz, you can use a maximum frequency of 125 Hz (thus providing four sample points per cycle). There are also practical considerations. If your study concerns alpha activity, analyzing up to 150 Hz will increase analysis time and may produce time-frequency plots in which only a small portion of the plot is relevant. Furthermore, having additional frequencies that are not of interest will lead to more stringent statistical significance thresholding when correcting for multiple comparisons. If you do not have specific expectations about the frequency band of the results, you could use a range of 4 Hz to 60 Hz. This range is likely to capture the main time-frequency dynamics in most cognitive tasks. If you see task-related activity close to the lowest or highest frequencies, you could reanalyze the data using a broader frequency range.

13.7.3 How Many Frequencies Should Be Used?

If you have a clear hypothesis about 10-Hz activity, you technically need only one wavelet. However, this might be a suboptimal analysis strategy because, for example, there might be individual differences in the exact frequency of the alpha activity. Having more frequencies gives you the freedom to select the appropriate frequency ranges based on the data, and it will also allow you to perform exploratory and post-hoc analyses in addition to the hypothesis-based analyses. On the other hand, more frequencies means larger results matrices, increased computation time, and increased requirements for multiple comparisons corrections.

Furthermore, because of frequency smoothing inherent in wavelet convolution, there is autocorrelation in the time-frequency plots. That is, a 6.3-Hz wavelet and a 6.6-Hz wavelet are unlikely to provide independent information. This is illustrated in figure 13.10 for two wavelets with peak frequencies of 9 Hz and 10 Hz. You can see that, both in the time domain and in the frequency domain, these two wavelets have a lot of overlap, meaning that they will extract similar (although not identical) time-frequency information from EEG data.

On the other hand, using more frequencies produces smoother time-frequency plots that are easier to inspect visually. In general, 20–30 frequencies provide a reasonable number to cover a broad frequency range (e.g., 4–60 Hz) while also making nice-looking plots.

13.7.4 Should Frequencies Be Linearly or Logarithmically Spaced?

You can specify the peak frequencies of wavelets to increase linearly (e.g., 5, 13, 22, 30 Hz), or logarithmically (e.g., 5, 9, 16.5, 30 Hz). Both are correct, although given that frequencies are often conceptualized on a logarithmic scale, it makes sense to use logarithmically spaced wavelet peak frequencies. That is, although the difference between 5 and 13, and between 22 and 30, is the same in integer units, the difference between 5 and 9, and between 16.5

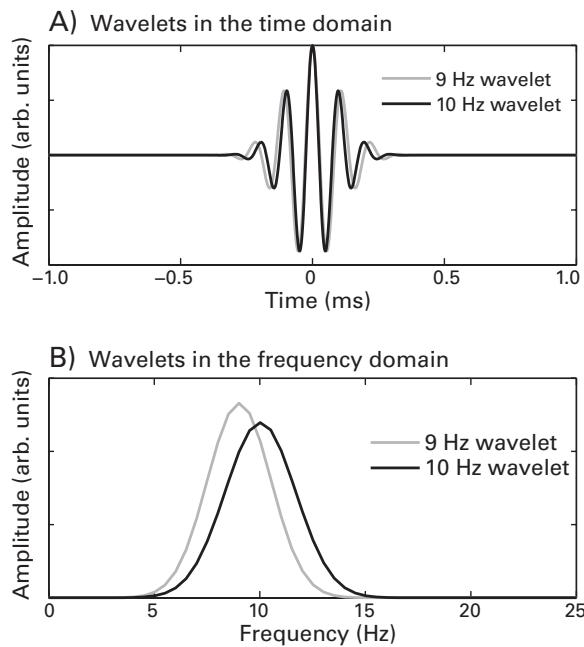


Figure 13.10

Time-domain (panel A) and frequency-domain (panel B) illustration of how wavelets with close frequencies have similar and largely overlapping time and frequency profiles.

and 30, is the same in logarithm-base-10 units. If you use logarithmic frequency scaling, the width between frequencies will be approximately equal over the range of frequencies extracted (thus, 4–8 Hz and 30–60 Hz would take up a similar amount of *y*-axis coverage in time-frequency plots).

The choice of linear versus logarithmic scaling does not have major consequences for statistical analyses or interpretations of the results, but it does affect the visual appearance of time-frequency plots. Consider the two time-frequency power plots in figure 13.11 (plate 2), which show the same result, but the *y*-axis is scaled either linearly (figure 13.11A) or logarithmically (figure 13.11B). Take a minute to find the same time-frequency features in both plots (hint: they might be seemingly far apart from each other on the *y*-axis).

If your main results concern lower-frequency activity, it is advisable to use logarithmic scaling because this will highlight the lower-frequency end of the spectrum (figure 13.11A); on the other hand, if your main results concern higher-frequency activity, it is advisable to use linear scaling because this will highlight the higher frequencies and visually minimize

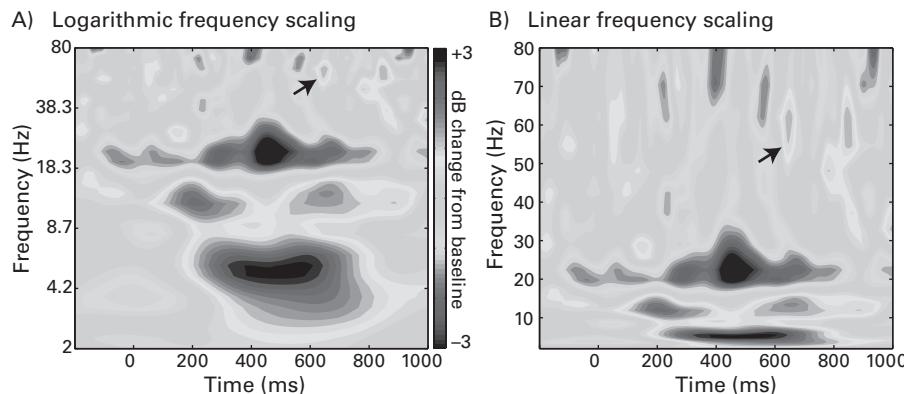


Figure 13.11 (plate 2)

Time-frequency results of the same analyses applied to the same data can look different depending on whether the frequencies in the *y*-axis are scaled logarithmically (panel A) or linearly (panel B). The black arrows, for example, show the same gamma power burst in both plots.

the lower frequencies (figure 13.11B). This figure also shows that the choice of linear versus logarithmic scaling can be made after the analyses have been performed. That is, you can extract logarithmically spaced frequencies and then plot them on a linear scale.

On this note, it is worth briefly following a tangent here to discuss *y*-axis scaling in the Matlab function `image` (and the related function `imagesc`, which automatically color-scales the plot). When using this function, Matlab will always scale the *y*-axis in linear increments, regardless of the *y*-axis labels you input. But in figure 13.11 (color plate 2) the *y*-axis (frequency) is scaled in logarithmic increments. If you define logarithmically spaced frequencies in the analysis and then plot the data using the following code in Matlab:

```
imagesc(EEG.times,frequencies,tf_data)
```

then your plot will have the wrong *y*-axis labels! This is a subtle but very important point. If you have a linear frequency scaling, you can safely enter “frequencies” as the *y*-axis label (second input) into the function `imagesc`. But if you use nonlinear frequency scaling, it is best not to enter the second input (that is, leave it blank by using square brackets: []) and then manually specify the *y*-axis ticks and their labels. The online Matlab code shows you how to label the *y*-axis correctly to display nonlinearly spaced frequencies. The function `contour` (and related functions such as `contourf`) will correctly scale the *y*-axis regardless of whether the frequencies are spaced linearly or logarithmically.

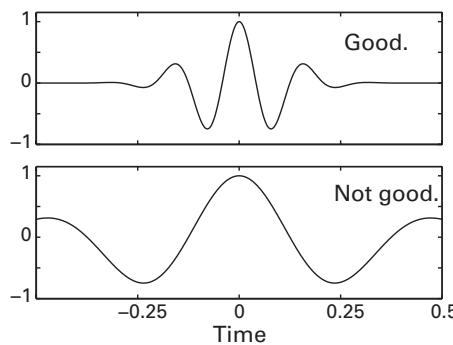


Figure 13.12

The lowest-frequency wavelet should taper cleanly to zeros. If your wavelet looks like the wavelet in the lower plot, increase the amount of time (in this example, -2 to $+2$ s would be enough time). There is no theoretical limit on how long wavelets can or should be, so it is best to err on the side of making them too long compared to making them too short.

13.7.5 How Long Should Wavelets Be?

This is an important issue, but it has an easy answer. Wavelets should be long enough such that the lowest-frequency wavelet tapers to zero (or extremely close to zero) at both the negative and positive ends of time. You can plot the real part of the lowest-frequency wavelet to make sure it is long enough. If the wavelet is cut off as in the lower panel of figure 13.12, edge artifacts will be present in the results. There is no limit on the maximum length of the wavelet because the wavelets taper to zero; multiplying the data by lots of zeros does not affect the result of wavelet convolution. Although wavelets at different frequencies may have different lengths, this will complicate your Matlab code and is not necessary. Unless you use wavelets with very low frequencies, defining wavelets using a time range of -2 s to $+2$ s should be long enough.

A related and also important detail is that the wavelets should be centered in the time window. The easiest way to ensure that the wavelet is centered in time is to create the wavelet using a time vector from a negative to a positive number (e.g., -2 seconds to $+2$ seconds). The center of the wavelet will thus be located at time = 0. This procedure has the added benefit that it will give the wavelet an odd number of data points, which is convenient for convolution. Finally, make sure the wavelet is created using the same sampling rate as the EEG data.

13.7.6 How Many Cycles Should Be Used for the Gaussian Taper?

The number of cycles of the Gaussian taper defines its width, which in turn defines the width of the wavelet. This is a nontrivial parameter and may influence the kinds of results you can

obtain. This parameter controls the trade-off between temporal and frequency precisions. This is sometimes called the trade-off between temporal and frequency resolution, but this is technically incorrect: both temporal resolution and frequency resolution are determined entirely by the data-sampling rate and do not change with analysis parameters; in contrast, it is their precisions that are affected by analysis parameters. See figure 2.4 for a review of the differences among precision, resolution, and accuracy.

As is shown below, a larger number of cycles gives you better frequency precision at the cost of worse temporal precision, and a smaller number of cycles gives you better temporal precision at the cost of worse frequency precision. This is the parameter that controls the Heisenberg uncertainty principle applied to time-frequency analysis: the more you know about *when* something happened, the less you know about *where* (i.e., at which frequency) it happened, and vice versa.

Looking at wavelets of the same frequency with a different number of cycles should help you understand why this is the case. A wavelet of three cycles (figure 13.13A) has many more zero values compared to the wavelet of seven cycles (figure 13.13B). This difference has several related implications: (1) the three-cycle wavelet is better suited for detecting transient activations whereas (2) the seven-cycle wavelet is more sensitive to longer activations at specific frequencies; (3) the three-cycle wavelet is more precise at

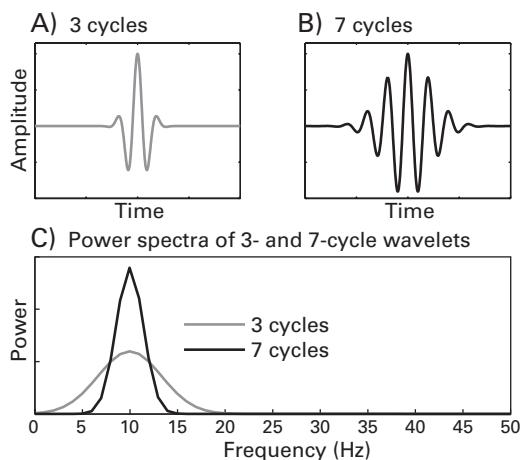


Figure 13.13

Two wavelets at the same frequency that differ only in number of cycles have different temporal and frequency characteristics and therefore different sensitivities to temporal and frequency features of the data. This has practical implications for the results, as shown in figure 13.14 (color plate 3).

localizing dynamics in time whereas (4) the seven-cycle wavelet is more precise at determining the frequency of the dynamic. This latter point is also clear from inspecting the power spectra of the two wavelets (figure 13.13C): although the peak frequencies are the same, the widths differ. At the extreme case of an infinite number of cycles, the wavelet becomes a sine wave, and the wavelet convolution reduces to one step of the Fourier transform.

How many wavelet cycles should you use with real data? As with many other parameters, there is no single correct answer; it depends on the goal of the analysis. If you are looking for transient changes in activity, a smaller number of cycles (around three or four) will be better. If you have a relatively long trial period in which you expect frequency-band-specific activity (this could be the case during an extended visual presentation or a working memory delay), a larger number of cycles (seven to ten) will facilitate identifying temporally sustained activity. If you would like to distinguish activity at frequencies within a narrow range (for example, separating lower alpha from upper alpha), you should use a larger number of cycles and accept the decreased temporal precision. On the other hand, if you have hypotheses about how quickly a neural response can dissociate condition A from condition B, you should use a smaller number of cycles and accept the decreased frequency precision. If both temporal precision and frequency precision are important for your study, you can perform the analysis twice: once with a smaller number of cycles to determine the timing of the neural events and once with a larger number of cycles to determine the frequency bands of the neural events. (I have never seen this done in practice, but there is nothing wrong with this approach as long as the results are appropriately interpreted.)

You can observe the influence of the number of wavelet cycles in wavelet convolution with real data in figure 13.14 (plate 3). Figure 13.14A shows the results of convolution with three-cycle wavelets, which maximize temporal precision. There seem to be two power increases, one at 390 ms and one at 600 ms, that span a fairly broad lower frequency range, from around 3 Hz to 12 Hz. Furthermore, there seem to be several pulses of power suppression that span frequencies of 17 Hz to 35 Hz. Figure 13.14B, in contrast, shows the results of a convolution with the same EEG data and wavelets with the same peak frequencies, but this time with 10 cycles. These results highlight slightly different features of the same data. The two separate low-frequency peaks have joined together, but it now can be seen that there are power increases in two distinct frequency bands, one in the theta range and one in the upper alpha range. Furthermore, the pulses of beta-band power suppression have congealed into a temporally sustained response.

Which of panels A and B (figure 13.14, plate 3) show the “correct” result? Both. They are both correct, and they are both equally valid descriptions of the time-frequency dynamics in

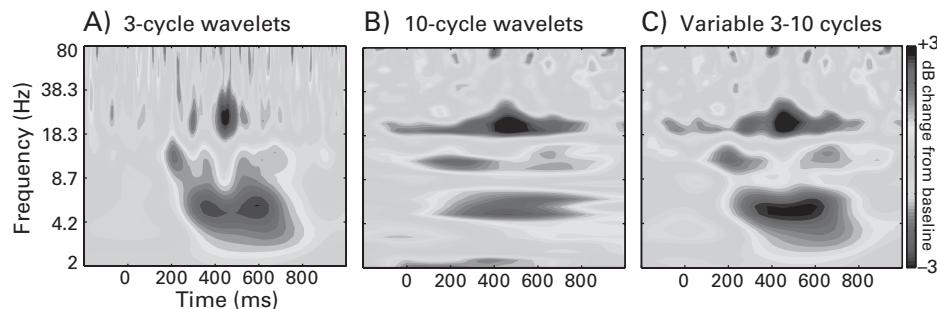


Figure 13.14 (plate 3)

The width of the Gaussian that is used to create the wavelets affects the features of the results that will be obtained from complex Morlet wavelet convolution. Different numbers of cycles can be used to highlight temporal precision (panel A) or frequency precision (panel B). Panel C shows that the balance between temporal and frequency precisions can change as a function of frequency; this increases temporal precision at lower frequencies and increases frequency precision at higher frequencies. The results were transformed to decibel (dB) change relative to a pretrial baseline. Baseline transformations are discussed in depth in chapter 18.

the data. The difference is that the results in panel A highlight the temporal dynamics of the data, and the results in panel B highlight the frequency dynamics of the data.

Figure 13.14A and B shows cases in which the number of wavelet cycles is constant over frequency. This need not be the case; the number of wavelet cycles can also change as a function of frequency. This can be accomplished by increasing the number of cycles with the frequency of the wavelet. For example, the number of cycles could increase from 3 to 10 in the same number of steps used to increase the frequency of the wavelets from 2 Hz to 80 Hz (figure 13.13C). This is a useful procedure because it adjusts the balance between temporal and frequency precisions as a function of the frequency of the wavelet. It will also change the frequency spread of the wavelets as a function of frequency, which is discussed in section 13.8. In general, you should use at least 3 cycles, and at most 14 cycles, unless you have a specific reason to do otherwise. If you use more than seven cycles, it is a good idea to check that the wavelets taper to zero, as was shown in figure 13.12.

One final theoretical point to consider when choosing the number of cycles for a wavelet is that the data should ideally be stationary during the period in which the wavelet is nonzero. The more cycles you use, the longer the wavelet's nonzero components, and thus, the longer the period of time for which data should be stationary. As shown in figure 11.9, violations of stationarity do not invalidate the results, but they will decrease the accuracy of the estimate of the frequency characteristics.

13.8 Determining the Frequency Smoothing of Wavelets

It might be relevant to know how much frequency smoothing the wavelets apply to your data. Because the wavelet has a Gaussian shape in the frequency domain (as shown in, e.g., figures 13.10 and 13.13), the extent to which neighboring frequencies contribute to the result of wavelet convolution can be reported in terms of full width at half-maximum (FWHM). FWHM refers to the frequency width for which the power is at 50% on the left and right sides of the peak (that is, the lower and upper 50% attenuation frequencies). The formula to compute the FWHM follows.

$$\text{FWHM} = 2\sqrt{2 \ln 2}\sigma \quad (13.11)$$

where σ is the standard deviation of the frequency response. This formula is valid only for Gaussian distributions. In practice, it might be better to estimate the FWHM from the data rather than compute it algorithmically because the precise standard deviation will depend on the number of sample points, which in turn depends on the length of the wavelet and the sampling rate of the data. The way to estimate FWHM is first to normalize the power spectrum of the wavelet such that it has a minimum value of 0 and a maximum value of 1. From the normalized frequency-domain response, identify the frequency points prior to and following the spectral peak that are closest to 0.5 and then subtract those two frequencies. Figure 13.15A illustrates this procedure, and figure 13.15B shows the FWHM for a range of frequencies using a fixed number of wavelet cycles (3 or 10) and a variable number of wavelet cycles as a function of frequency.

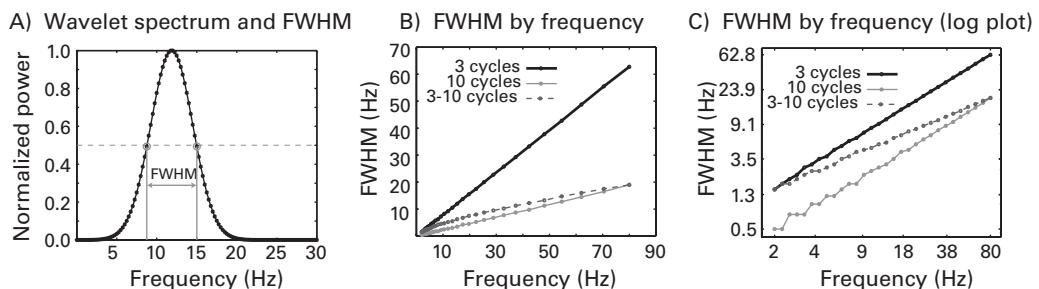


Figure 13.15

FWHM of wavelets, shown for one wavelet (panel A), and for a family of wavelets (panel B). Panel A illustrates how the FWHM can be computed as the range of frequencies corresponding to 50% of the power of the peak of the spectral representation. Panel B shows the FWHM as a function of frequency for the time-frequency plots shown in figure 13.14 (plate 3). Panel C shows the same information using logarithmic x - and y -axis scaling.

13.9 Tips for Writing Efficient Convolution Code in Matlab

Because the Fourier transform of the EEG data does not change as a function of frequency, the FFT of the data needs to be computed only once before looping over frequencies. This can save considerable time, particularly if you have a lot of data and many frequencies. In contrast, if you use the Matlab `conv` function, the FFT of the data will be redundantly recomputed at each iteration inside the frequency loop.

You can further decrease analysis time by using FFTs of an order that is a power of two. That is, the FFT of a 1024-point vector is faster to compute than the FFT of a 1023-point vector. Usually, data time series must be zero-padded to get to a length that is a power of two, so make sure you remove these extra points after convolution. The online Matlab code for this chapter shows you how to zero-pad FFTs and then remove the extra points after convolution.

Finally, you do not need to perform convolution on each trial separately; it is faster and more elegant to concatenate all trials into one long time series, perform one convolution with all trials, and then reshape the result back to a time-by-trials matrix. This procedure is demonstrated in section 14.9.

13.10 Describing This Analysis in Your Methods Section

If you perform a complex Morlet wavelet convolution to analyze your data, there are several important details to include in the Methods section, including the minimum, maximum, and number of frequencies of the wavelets; whether frequencies increased linearly or logarithmically; and the number of wavelet cycles and whether this changed as a function of frequency. Morlet wavelets are commonly used in the literature, so it is not necessary to include the equations in the Methods section unless you modify the standard equations. On the other hand, the equations for Morlet wavelets are brief, so there is little concern about exceeding journal space or word count limitations. You could also report the frequency-domain FWHM of the wavelets, although this is not typically done.

13.11 Exercises

1. Create a family of complex Morlet wavelets, ranging in frequencies from 2 Hz to 30 Hz in five steps.
2. Convolve each wavelet with EEG data from all electrodes and from one trial.
3. Extract power and phase from the result of complex wavelet convolution and store in a time \times frequency \times electrodes \times power/phase matrix (thus, a $640 \times 5 \times 64 \times 2$ matrix).

4. Make topographical plots of power and phase at 180 ms at all frequencies (hint: you may need to use the `squeeze` function to remove singleton dimensions). Arrange the plots in one figure with five columns for frequency and two rows for power/phase. Put labels in the plot so it is clear which topographical maps correspond to which frequencies.
5. Repeat step 4 for activity at 360 ms.
6. Are there any prominent topographical features in power or in phase? Do these differ for different frequencies? Do power and phase have similar topographical distributions? Is there any reason to suspect that they might have similar or different topographies?
7. (Optional) Because phase values are circular ($-\pi$ and $+\pi$ are identical), most color maps are inappropriate because they suggest that $-\pi$ and $+\pi$ are very different values (represented, e.g., by blue and red colors). Create a circular colormap that can be used for phase values. You can do this by setting the red, green, and/or blue values to be a cosine function rather than a linear function. Recreate the phase topographical maps. Do they look any different with the new color maps?

14 Bandpass Filtering and the Hilbert Transform

In this chapter you will learn another method for time-frequency decomposition: bandpass filtering and applying the Hilbert transform (hereafter referred to as the filter-Hilbert method). This method is also commonly used in the literature and can give similar results as wavelet convolution and the short-time FFT (Bruns 2004; Le Van Quyen et al. 2001). The near-equivalence of bandpass filtering and wavelet convolution was shown in figure 12.5 and is demonstrated in this chapter. After filter-Hilbertring an EEG signal, the resulting analytic signal is a complex time series from which you can extract power and phase values using the same methods as described in the previous chapter for complex wavelet convolution. The result of the Hilbert function, like the result of complex wavelet convolution, is called the analytic signal.

The main advantage of the filter-Hilbert method over wavelet convolution is that the filter-Hilbert method allows more control over the frequency characteristics of the filter, whereas the frequency shape of a Morlet wavelet is always Gaussian. There are two minor practical disadvantages to the filter-Hilbert method, which are that the Matlab filter kernel construction functions are in the Matlab signal-processing toolbox and that bandpass filtering is a bit slower than wavelet convolution.

14.1 Hilbert Transform

The Hilbert transform allows you to extract a complex signal from a signal that contains only a real part. Remember from chapter 13 that to extract time-varying estimates of power and phase from an EEG time series, you need a signal that can be represented using Euler's formula: $Me^{i2\pi ft}$, which can also be written as $M\cos(2\pi ft) + iM\sin(2\pi ft)$. This is the analytic signal. Without any processing, EEG data have the form $M\cos(2\pi ft)$, which is to say, an oscillatory signal that has only a real (cosine) component and no imaginary (sine) component. Convolution with a complex Morlet wavelet is one method for extracting the imaginary

component [that is, the $iM\sin(2\pi ft)$ part] because the complex wavelet contains both a cosine and an imaginary sine function.

The Hilbert transform is an alternative approach for extracting the imaginary part, $iM\sin(2\pi ft)$, of a real-valued signal, $M\cos(2\pi ft)$. This is done by creating and adding the phase quadrature component to the $M\cos(2\pi ft)$ part. The phase quadrature (or one-quarter-cycle) component is created by rotating parts of the complex Fourier spectrum of a real-valued signal (Smith 2007).

The following procedure will accomplish the Hilbert transform, and the online Matlab code will show you this procedure step by step. First, compute the Fourier transform of a signal and create a copy of the Fourier coefficients that have been multiplied by the complex operator (i). This turns the $M\cos(2\pi ft)$ into $iM\cos(2\pi ft)$. Next, identify the positive and negative frequencies. The positive frequencies are those between but not including the zero and the Nyquist frequencies, and the negative frequencies are those above the Nyquist frequency (throughout the Hilbert transform, the zero and Nyquist frequencies are left untouched). The next step is to convert the $iM\cos(2\pi ft)$ to $iM\sin(2\pi ft)$. Remember that cosine and sine are related to each other by one-quarter cycle; thus, to convert a cosine to a sine, you rotate the positive-frequency coefficients one-quarter cycle counterclockwise in complex space (-90° or $-\pi/2$) (think about the complex plane: rotating from the positive real axis to the positive imaginary axis turns a cosine into a sine). It was mentioned in section 11.5 that negative frequencies capture sine waves that travel in reverse order around the complex plane. Thus, to convert a cosine to a sine in negative frequencies, you rotate the negative-frequency coefficients one-quarter cycle clockwise (90° or $\pi/2$).

Rotating the positive-frequency Fourier coefficients counterclockwise can be achieved by multiplication with $-i$. Notice that the $iM\sin(2\pi ft)$ transformation for the positive frequencies involves multiplying i with $-i$, which is 1. This means that when the rotated positive-frequency Fourier coefficients are added to the original positive-frequency coefficients, the effect is to double the original positive-frequency Fourier coefficients.

Rotating the negative-frequency Fourier coefficients can be achieved by multiplication with i . Notice that this involves multiplying i with i , which is -1. Thus, when the rotated negative-frequency coefficients are added back to the original negative-frequency coefficients, the effect is a subtraction from themselves; that is, the negative-frequency coefficients become zero.

The final step is to take the inverse Fourier transform of the modulated Fourier coefficients. The result is the analytic signal, which can be used in the same way that you use the result of complex Morlet wavelet convolution. Thus, you can substitute the Hilbertized

signal in panel C of figure 13.7 and follow the procedures outlined in the previous chapter to obtain the real component, the power, and the phase-angle time series.

In practice, the Hilbert transform can be computed without explicitly rotating the positive- and negative-frequency Fourier coefficients, simply by doubling the positive-frequency coefficients and zeroing the negative-frequency coefficients. This is what the Matlab function `hilbert` does. You can see in the online Matlab code that the explanation in the previous paragraphs produces identical results to the Matlab `hilbert` function, although the Matlab function is slightly faster. This is not the only method for computing the Hilbert transform—for example, the Hilbert transform can also be computed through the Hartley transform or the discrete cosine transform (Olkonen, Pesola, and Olkkonen 2010).

The Hilbert transform does not affect the real part of the signal. This can be shown by plotting a cosine wave and the real component of that cosine wave after Hilbert transform (figure 14.1).

Important! The Matlab function `hilbert` will accept a matrix as input. This is useful to apply the Hilbert transform to many trials or many electrodes in one command. However, the Matlab `hilbert` function will always compute the FFT over the first dimension. Thus, you should make sure that the first dimension of your matrix contains time; you do not want to compute the Hilbert transform over trials or over electrodes. If you plot only the real part

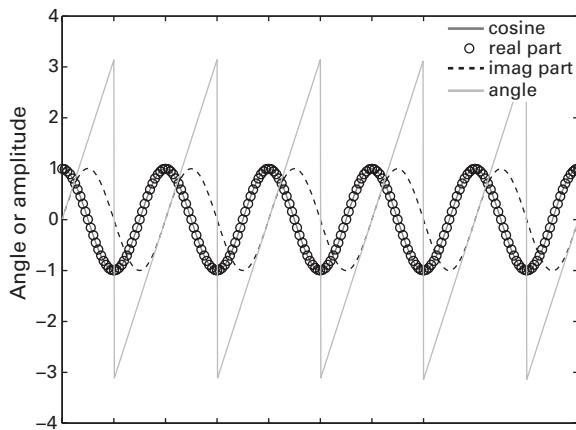


Figure 14.1

The Hilbert transform is used to extract complex information from a real-valued signal. Here, a cosine wave was created (solid gray line), and its Hilbert transform was taken. The real part of the result of the Hilbert transform is the original signal, while the imaginary part is a sine wave (which is what happens when a cosine is phase-shifted 90°).

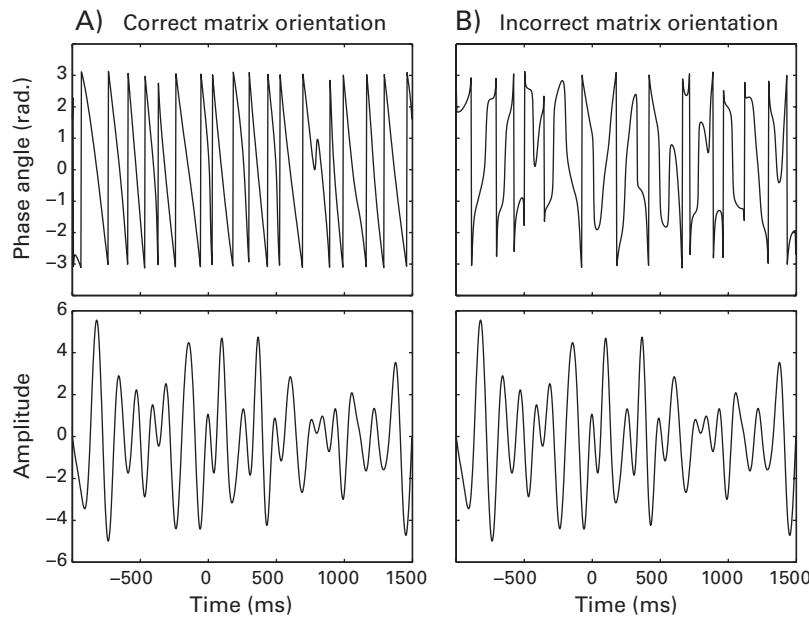


Figure 14.2

Applying the Matlab `hilbert` function to matrix data can produce incorrect results if the input data matrix is incorrectly oriented. This is easily identified by plotting the angle of the result of the Hilbert transform (top row). As mentioned in the text, the Hilbert transform does not change the real component (bottom row).

of the Hilbert transform, you will not know whether it was computed on the correct dimension because the Hilbert transform does not affect the real part of the signal. A fast and easy method to make sure the `hilbert` function was applied correctly is to plot the phase angles over time; either they will look as you expect phase angles to look (figure 14.2A) or they will look strange (figure 14.2B). If the phase angles look incorrect, do not analyze your data further until you resolve the issue (most likely, time was not in the first dimension of the matrix input to the `hilbert` function).

14.2 Filtering Data before Applying the Hilbert Transform

Bandpass filtering is not necessary; you can simply apply the Hilbert transform to the broad-band data. However, the resulting analytic signal may be difficult to interpret because all frequencies present in the data will contribute to the result, and frequencies with more power

(generally lower frequencies because of power-law scaling) will contribute more compared to frequencies with less power (generally higher frequencies). Thus, you should filter the data into separate frequency bands before applying the Hilbert transform. This way, results can be interpreted in a frequency-band-specific manner.

This leads to the main advantage of the filter-Hilbert method over complex wavelet convolution: you have more control over the characteristics of the filter, compared to the amount of control you have over the characteristics of Morlet wavelets. In fact, you have no control over the shape of the Morlet wavelet; it is always Gaussian shaped in the frequency domain. A bandpass filter, on the other hand, can be Gaussian or plateau shaped. The rest of this chapter therefore focuses on constructing and applying bandpass filters to EEG data for use in time-frequency decomposition.

14.3 Finite versus Infinite Impulse Response Filters

Filters can be classified as FIR or IIR (finite or infinite impulse response). These terms describe how a filter responds to an impulse—a single input. FIR filters have a response that ends at some point (that is, its response is finite), whereas IIR filters have a response that never ends (its response is infinite).

For most situations when using the filter-Hilbert method for time-frequency decomposition of EEG data, FIR filters are preferred over IIR filters. FIR filters are more stable and less likely to introduce nonlinear phase distortions. Although the computational cost is a bit higher compared to IIR filters because of FIR's increased filter order (filter order is discussed in section 14.5), in practice this is not a major limitation for modern computers. Most of the rest of this chapter covers FIR filter design and application because of the advantages of FIR over IIR filters. The Butterworth IIR filter is introduced briefly in section 14.7.

14.4 Bandpass, Band-Stop, High-Pass, Low-Pass

There are four ways of using filters to highlight some frequencies of the data while attenuating other frequencies. The terms bandpass and band-stop are intuitive (respectively, keep and remove activity between the specified frequencies). The terms high-pass and low-pass refer, respectively, to filters that retain high or low frequencies while attenuating low or high frequencies (thus, a high-pass filter allows higher frequencies to pass through). For time-frequency decomposition, bandpass filters are the most useful type of filters. Morlet wavelets, as already mentioned in earlier chapters and demonstrated again in this chapter, are bandpass filters.

14.5 Constructing a Filter

Recall that when you convolve a wavelet with EEG data, the real component of the result of the convolution is the EEG data filtered around the peak frequency of the wavelet. From the convolution theorem, you can say that the result of the convolution is a weighted combination of the frequency structure of the EEG data and the frequency structure of the wavelet (figures 11.10 and 11.11). Bandpass filtering works the same way: a kernel is constructed based on ideal frequency characteristics that you define, and when the kernel is convolved with EEG data, the requested frequencies are preserved while the undesired frequencies are attenuated. You can see in figure 14.3A that the filter kernel looks a bit like a wavelet in that it appears to be a tapered sine wave (or a tapered sum of sine waves), but the waveform is more complex than the Gaussian-tapered sine wave used for Morlet wavelets.

The power spectrum of a Morlet wavelet is Gaussian shaped. In contrast, the power spectrum of an FIR filter can be plateau shaped (figure 14.3B). In practice, you can have the FIR

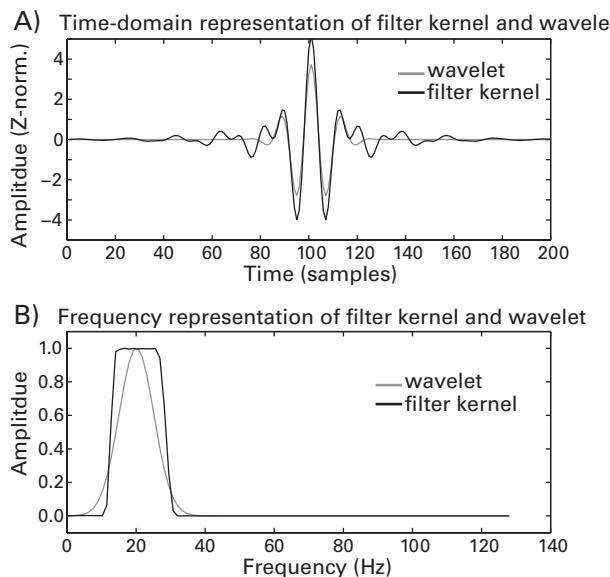


Figure 14.3

Comparison of the time-domain and frequency-domain representations of a Morlet wavelet and an FIR filter kernel. Both the wavelet and the filter kernel were designed to isolate 20-Hz activity. Data were amplitude-normalized to facilitate visual comparison. Panel B also illustrates why it is informative to refer to the “peak” frequency of a wavelet and the “center” frequency of a bandpass filter.

filter take any shape you want in the frequency domain, although plateau-shaped filters are recommended. This is why bandpass filtering can provide extra frequency specificity compared to wavelets, and this is why wavelet convolution will generally produce time-frequency plots that appear a bit smoother over the frequency axis compared to time-frequency plots produced by the filter-Hilbert method.

Now that you have a conceptual idea of what a filter kernel looks like, it is time to learn how to create a filter kernel in Matlab. Broadly speaking, the way to create a filter kernel is by specifying the ideal filter shape and the frequencies that define that shape. Frequencies are specified as a fraction of the Nyquist frequency. There are several functions in Matlab to construct filter kernels, most of which are contained in the signal-processing toolbox. The two functions that are discussed here are `firls` and `fir1`. These functions create finite-impulse-response filters via least squares (`firls`) and a windowed linear phase filter with tight transition zones (`fir1`). After a discussion of how to construct a filter kernel, differences between `firls` and `fir1` are discussed. Other Matlab functions for creating filter kernels include `fir2` (frequency-sampling-based filter construction), `firrcos` (raised cosine-shaped filter), `gaussfir` (Gaussian-shaped filter), and `firpm` (Parks-McClellan). You can also create your own filter kernels, but this is not recommended unless you are knowledgeable about the mechanics of proper filter kernel construction.

There are three inputs to the Matlab function `firls` (several of the other filter construction functions take the same inputs, although `fir1` takes only the first two of the following inputs). The first input is the order parameter, which defines the length of the filter kernel (the length of kernel is the order plus one). The filter order determines the precision of the filter's frequency response. Although the order parameter is less important for filter kernel design compared to the second and third inputs, it deserves some attention. Larger orders will produce kernels with relatively better frequency precision, although they will also increase computation time. There is a lower bound to how short the filter kernel can be: if you want to resolve activity at a particular frequency, the filter kernel must be long enough to contain at least one cycle at that frequency. For example, if the lower frequency bound is 10 Hz, the filter kernel must be at least 100 ms long. In practice, it is good to have somewhere between two and five times the lower frequency bound (thus, between 200 and 500 ms for a 10-Hz filter). Eeglab, for example, uses three times the lower frequency bound as the default order. The order may change as a function of the center frequency of the bandpass filter (more specifically, the lower bound of the filter for each frequency band). In theory, the filter order must be an even number to exclude a filter representation for the Nyquist frequency. In practice, however, the Matlab filter construction functions `fir1` and `firls` will automatically add one to the order if the inputted order is an odd number. When

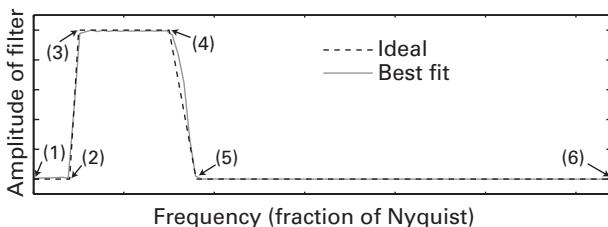


Figure 14.4

The second input into the Matlab filter-construction function `firls` defines the ideal frequency response. This figure illustrates how to define an ideal frequency response for a plateau shaped FIR filter (see dotted line), and the numbers show which frequency points will be used as the input to `firls`. The units are scaled by the Nyquist frequency (thus, the first input is 0 and the last input, which is the Nyquist frequency, is 1). If you use the `fir1` function, only the frequencies corresponding to (3) and (4) are used as inputs. The slopes from (2) to (3), and from (4) to (5), are called the transition zones. The actual frequency response of the filter kernel (solid line) is a least-squares approximation of the ideal response.

calling filter-construction Matlab functions, make sure you input sample points for the order parameter, not time in milliseconds or seconds.

The second input to `firls` is a vector of frequencies that defines the shape of the response. For a bandpass filter, you can use six numbers: the zero frequency, the frequency of the start of the lower transition zone, the lower bound of the bandpass, the upper bound of the bandpass, the frequency of the end of the upper transition zone, and finally the Nyquist frequency (figure 14.4). These numbers are scaled such that the Nyquist frequency is 1.0; thus, you can define this vector in frequencies in Hz and then divide the vector by the Nyquist frequency. For the function `fir1`, you enter only two numbers, the lower and upper bounds of the filter, because transition zones are set to zero (differences between `firls` and `fir1` are discussed later in this section).

Finally, the third input to `firls` is the “ideal” filter response amplitude. This is a vector comprising as many numbers as the second input and contains zeros for the frequencies you want to attenuate and ones for the frequencies you want to keep. For a bandpass filter, you can use [0 0 1 1 0 0], where the ones correspond to the lower and upper frequency bounds of the bandpass plateau, the first and last zero corresponds to the DC and Nyquist frequencies, and the second and fifth zeroes correspond to the frequency bounds of the transition zones (see figure 14.4). You can also use numbers between zero and one if you want to suppress, for example, 48.23% of a frequency range, but in practice it is best to keep the filter shape simple.

You are not limited to inputs of six numbers. You can use as many numbers as you want to maximize the control over the precise shape of the filter. For example, by making the

inputs 100 elements long and shaped like a Gaussian around the desired peak frequency (imagine smoothing the frequency function in figure 14.4 so it looks more like the wavelet frequency representation in figure 14.3B), `firls` will return a filter kernel that looks very similar to a Morlet wavelet. This is why wavelet convolution and the filter-Hilbert method can produce nearly identical results. In theory, you can make more complicated filters, for example those containing two spectral peaks at different frequencies. However, complicated filter shapes are difficult to design and may introduce artifacts in the time-domain data. In the case of multipeak filters, those artifacts may be difficult to detect. Again, it is best to keep the filter design simple, as illustrated in figure 14.4.

The frequency width of the bandpass filter (that is, the length of the plateau in the frequency representation of the filter kernel) and the transition zones define the trade-off between temporal precision and frequency precision. As the plateau becomes narrower, the frequency precision increases, but this decreases the temporal precision because narrow frequency filters require longer kernels to resolve. This can be seen by inspecting the time-domain filter kernels: narrower bandpass filters produce filter kernels that have nonzero values that extend for longer periods of time (see figure 14.5). This means that the estimate of band-specific activity at one time point is relatively more influenced by data at surrounding time points, which thus decreases the temporal precision. Consider the extreme situation: as the filter width becomes narrower and narrower, toward being infinitesimally narrow, the result is a perfect sine wave with an amplitude that never dampens. Conversely, as the filter plateau becomes wider, the amount of time required to construct the filter kernel becomes smaller. Thus, the temporal precision increases while the frequency precision decreases. You can also refer back to figure 9.2: the filter setting that involved the least amount of loss of temporal precision was the 0–40 Hz filter.

Recall from section 7.3 that sharp edges in the time domain can produce artifacts in the frequency domain. The reverse is also true: sharp edges in the frequency domain can produce artifacts in the time domain. These artifacts take the form of ripples, which can look like oscillations in the time-domain-filtered signal. Consider an extreme example: a power spectrum that is entirely flat except for a sharp peak at one frequency produces a perfect ringing effect in the time domain—a sine wave. Sharp edges in the frequency domain can be avoided by using transition zones. If you use `firls`, the transition zones should be between 10% and 25% of the lower and upper frequency bounds. There is a trade-off for the width of the transition zones: sharper transition zones give a better frequency response but increase the risk of introducing time-domain ringing artifacts; gentler transition zones decrease the risk of time-domain ringing artifacts but also have less frequency specificity. If you plan on using a transition zone of less than 10%, you should either smooth the filter kernel using a Hann or Hamming window to minimize edge artifacts when the filter is applied to EEG data

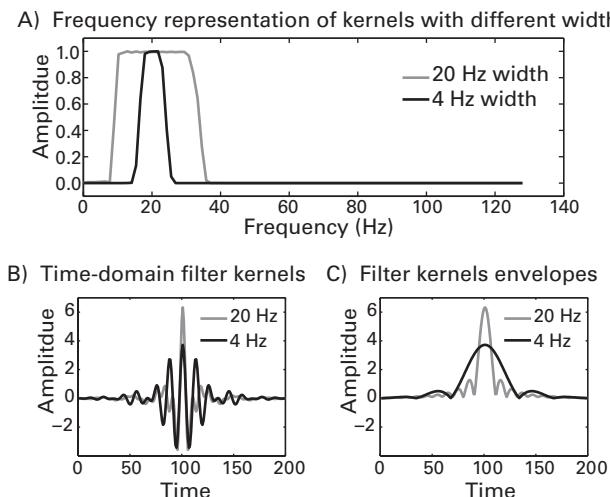


Figure 14.5

Illustration of how the width of the filter controls the time-frequency precision trade-off in bandpass filtering. Panel A shows the frequency representations of two filter kernels with the same center frequency (20 Hz) but with either 4-Hz or 20-Hz width (respectively, 2 Hz or 10 Hz on either side of the center frequency). Panel B shows the time-domain version of the same two filter kernels, and panel C shows their amplitude envelopes (the magnitude of the Hilbert transform of the time-domain kernel). Filters with wider pass-bands have kernels that taper to zero after fewer time points (gray lines in panels B and C, corresponding to a filter with a 20-Hz width), thus improving temporal precision at the expense of worse frequency precision. In contrast, filters with narrow pass-bands have kernels that taper to zero after more time points (black lines in panels B and C, corresponding to a filter with a 4-Hz width), thus improving frequency precision at the expense of worse temporal precision.

or use the function `fir1`. Increasing the order of the filter is also useful with sharp transition zones because the filter kernels will have more time to taper to zero, thus reducing the edge artifacts.

The main difference between the filter kernels created by `firls` and `fir1` is the transition zone: `firls` is a general function for filter kernel construction that allows you to define your own transition zones; `fir1` automatically sets the transition zones to zero and then smoothes the resulting filter kernel to minimize ringing artifacts. This smoothing effectively creates a nonzero transition zone, which can be seen in figure 14.6A,B. In fact, by using `firls` with transition zones of zero and then smoothing the resulting filter kernel with a Hamming window, you can nearly perfectly reconstruct the filter kernel created by `fir1` (figure 14.6C). Thus, the added flexibility of `firls` can lead to a better or worse filter design, depending on

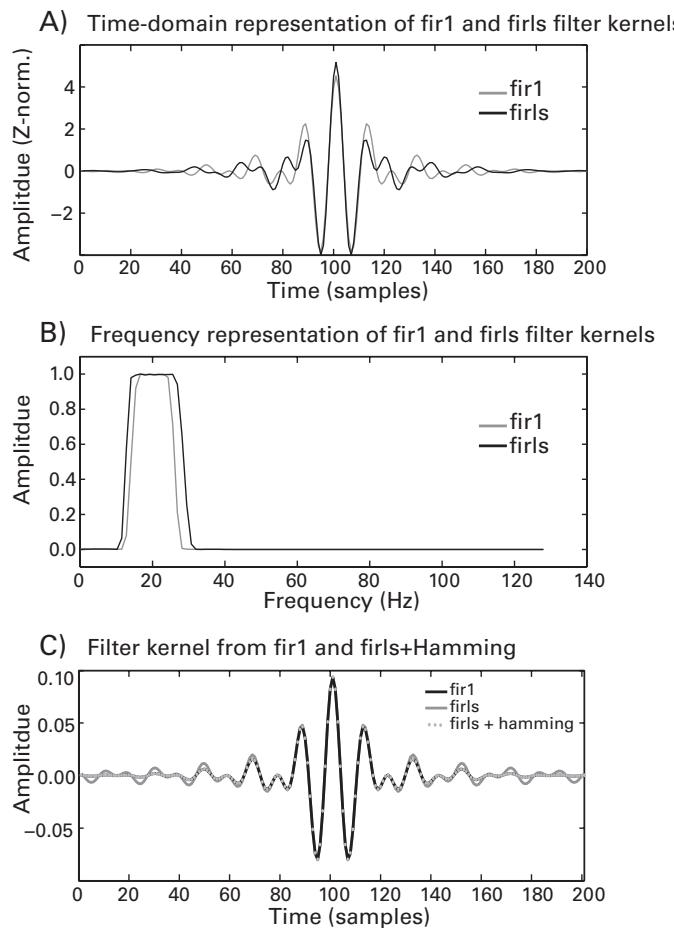


Figure 14.6

Comparisons of the time (panel A) and frequency (panel B) representations of FIR filter kernels created through Matlab functions `fir1` and `firls`. Panel C shows that the filter kernel computed by `fir1` is nearly the same as that produced by `firls` if you set the transition zones to zero and then smooth the filter kernel with a Hamming window.

the input parameters you specify and the goal of the filtering. If you want to make the filter as narrow-band as possible, it is best to use `fir1`. An example is an experiment utilizing the flicker effect (section 9.4), for which you would want to isolate the specific stimulus flicker frequency. If you are concerned about potential ringing artifacts in the time-domain data, or if you are using fairly wide pass bands, `firls` might be advantageous because you can specify gentle transition zones. If you are unsure which filter kernel to use, you can plot the kernel and its frequency representation (as in figure 14.6A,B), carefully inspect some sample data before and after applying different filters, and make an informed decision.

14.6 Check Your Filters

Check your filters carefully before applying them. You might have a poorly constructed filter without knowing it, and this may have negative consequences for your results. A poorly constructed filter means that there is a poor match between the frequency representation of the filter kernel and the ideal filter that you specified. In general, the fit of the actual filter to the ideal filter is a function of the center frequency, the pass-band width, the transition zones, and the filter order. The “goodness” of a filter can be quantified as the similarity between the actual frequency characteristics of the filter and the frequency characteristics of the ideal filter that you specified. Formally, this can be measured as a sum of squared errors.

$$sse = \sum_{i=1}^n (ideal_i - actual_i)^2 \quad (14.1)$$

in which sse is the sum of squared errors, n is the number of frequencies that were specified in the ideal filter (in the example shown in figure 14.4, this would be six), and *ideal* and *actual* refer to the power spectra of the ideal filter (the third input to the `firls` function) and the filter kernel. Note that the ideal filter has (in this case) six numbers, whereas the actual filter power spectrum has as many numbers as there are points in the filter kernel (that is, the order parameter plus one). Thus, you will have to match the frequency points appropriately. The online Matlab code for this chapter shows you how to do this.

The sse should be very close to zero, and you should not use any filter that has an sse above one. Examples of good and bad filters are shown in figure 14.7A,B. In this case the only difference between the ideal filters used to create these two filter kernels was the pass-band width—10 Hz for the good filter and 15 Hz for the bad filter. Figure 14.7C,D show sse for a broader range of filter characteristics. Figure 14.7C shows sse as a function of the center frequency and pass-band width, with the transition zone held constant at 20%; and figure

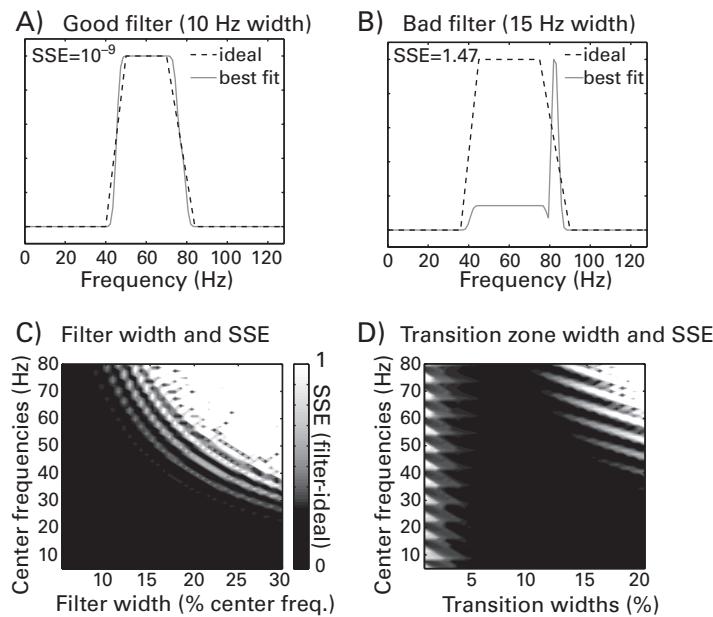


Figure 14.7

Examples of filter parameters that created poorly constructed filters. SSE is the sum of squared errors between the ideal response and the actual frequency response of the filter. In all cases the filter order was set to 200, and the Nyquist frequency was 128 Hz.

14.7D shows *sse* as a function of the center frequency and the transition width, with the pass-band width held constant at 20% of the center frequency. You can see that there are combinations of parameter settings that will produce bad filters.

The filter order and Nyquist frequencies were fixed for all simulations shown in figure 14.7. The *sse* will change with different filter orders, and the entire landscape will change with a higher Nyquist frequency. It is therefore important not to interpret this figure as an absolute reference for what parameters should and should not be used in constructing filters. Instead, you should be aware that some filter parameter settings will produce bad filters, and you should also be aware that it is easy to compute a metric that indicates whether a filter is bad (using equation 14.1) and thus should be avoided. For some extremely poorly designed filters, Matlab will issue a warning. However, other poorly designed filters will not be recognized by Matlab as requiring a warning. For example, the filter shown in figure 14.7B produced no Matlab warning. Without inspecting the filter kernel, you might not know that you are applying such poorly constructed filters to your data.

14.7 Applying the Filter to Data

The function `firls` (and most other filter construction functions including `fir1`) returns a vector of length $N + 1$, where N is the order. This is the filter kernel. You can plot it, as was shown in figure 14.3A and several other figures. The next step in filtering EEG data is to apply that filter kernel to the data. This can be done using the Matlab function `filtfilt`. The inputs to the function `filtfilt` are the filter kernel, a scalar or vector of weighting coefficients, and the data time series. The filter kernel is the output of `firls`, and the weighting coefficient can be set to 1.0 unless you would like to apply a weighting to the filter kernel, which is not typically necessary for time-frequency decomposition of EEG data. The output of `filtfilt` is the filtered data.

How does filtering work? It works in a conceptually similar way to wavelet convolution. Compare, for example, the output of Matlab's filter function `filtfilt` with the result of a convolution between the EEG data and this same filter kernel (figure 14.8).

The Matlab function `filtfilt` calls a function called `filter`. The function `filter` introduces a phase delay in the filtered signal because it is a causal filter, meaning that the filtered signal at any point in time is based only on previous values, not on future values. These phase delays can be reversed by refiltering the already-filtered data after reversing the

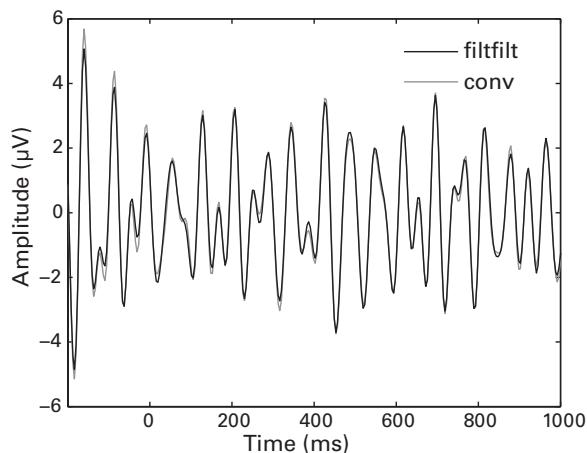


Figure 14.8

Convolution and zero-phase-shift filtering using the same filter kernel produce very similar results. A filter kernel was created using the Matlab function `firls`, and then this filter kernel was used to bandpass-filter EEG data using the Matlab function `filtfilt` (a zero-phase-shift filter; black line) and using convolution (gray line).

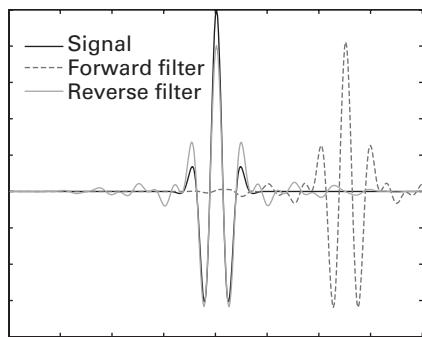


Figure 14.9

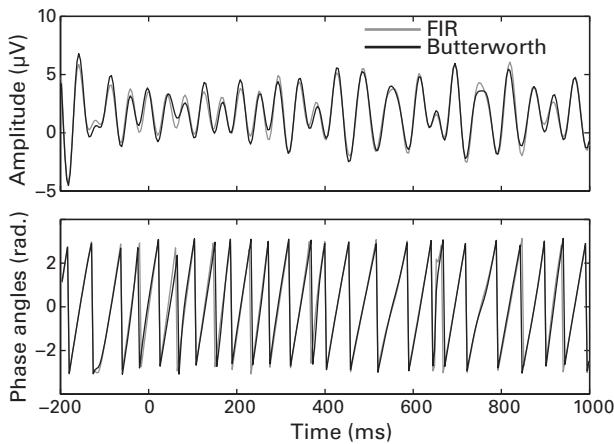
The Matlab function `filter` introduces a phase delay, which can be reversed by filtering the filtered data again after being flipped backward in time. The reverse filtered result should then be reversed again to arrive at the original orientation of time (in most cases, time originally goes forward).

filtered data in time. Thus, the phase delay that was introduced in the forward-going filter is reversed, and the final result contains no phase delays. You can see this in figure 14.9. Wavelet convolution does not introduce phase delays because convolution is noncausal, meaning that the filtered signal at each time point is based both on previous and on future values of the signal. Thus, convolution does not need to be repeated with time reversed. Although the Matlab function `filtfilt` is in the signal-processing toolbox, the function that it calls, `filter`, is not part of the signal-processing toolbox. Thus, if you do not have access to the signal-processing toolbox and would like to apply the filter-Hilbert method, you can use the function `filter`, but you will need to make sure you apply the filter twice, once forward in time and once backward in time.

After you have applied the filter to the data, the result is a real-valued signal that contains only those frequencies specified by the frequency response of the filter kernel, as illustrated for wavelet convolution (e.g., see figures 11.11, 11.12, and 12.6). The Hilbert transform can then be applied to the bandpass-filtered signal, and the analytic result of the Hilbert transform can be used to extract power and phase information, as shown in figure 13.7.

14.8 Butterworth (IIR) Filter

Perhaps the most common type of IIR filter used for EEG is the Butterworth filter. Constructing a Butterworth filter is similar to how FIR filters are constructed and produces similar results (figure 14.10). The online Matlab code shows you how to construct and apply a Butterworth filter.

**Figure 14.10**

Comparison of results after FIR and IIR (fifth-order Butterworth) filter.

14.9 Filtering Each Trial versus Filtering Concatenated Trials

It is faster to concatenate all trials into one long time series, filter that concatenated time series, and then cut out the separate trials, compared to filtering each trial separately. Not only is it faster, it will also facilitate writing cleaner code, in part because it renders a loop over trials unnecessary. The general procedure is shown in figure 14.11, and comparisons of results and computation times are shown in figure 14.12.

The main concern of trial concatenation is edge artifacts. If you use the `filtfilt` function, edge artifacts are less influential when performing single-trial filtering because Matlab's `filtfilt` function reflects the time series data (see figure 7.3) to minimize edge artifacts. If you have sufficient buffer time at the start and end of trials, edge artifacts will subside by the time period you will analyze. You should avoid filtering concatenated trials if you have short epochs right around the time period in which you want to analyze (e.g., if your epochs are -200 to +800 ms around the event of interest); in this case, it is also possible to reflect the data on your own and then concatenate the reflected trials.

14.10 Multiple Frequencies

As with wavelet convolution, you will probably want to examine activity over many frequency bands. This is done by looping over frequencies and refiltering the raw data into successive frequency bands. You might consider having some frequency overlap in the bands

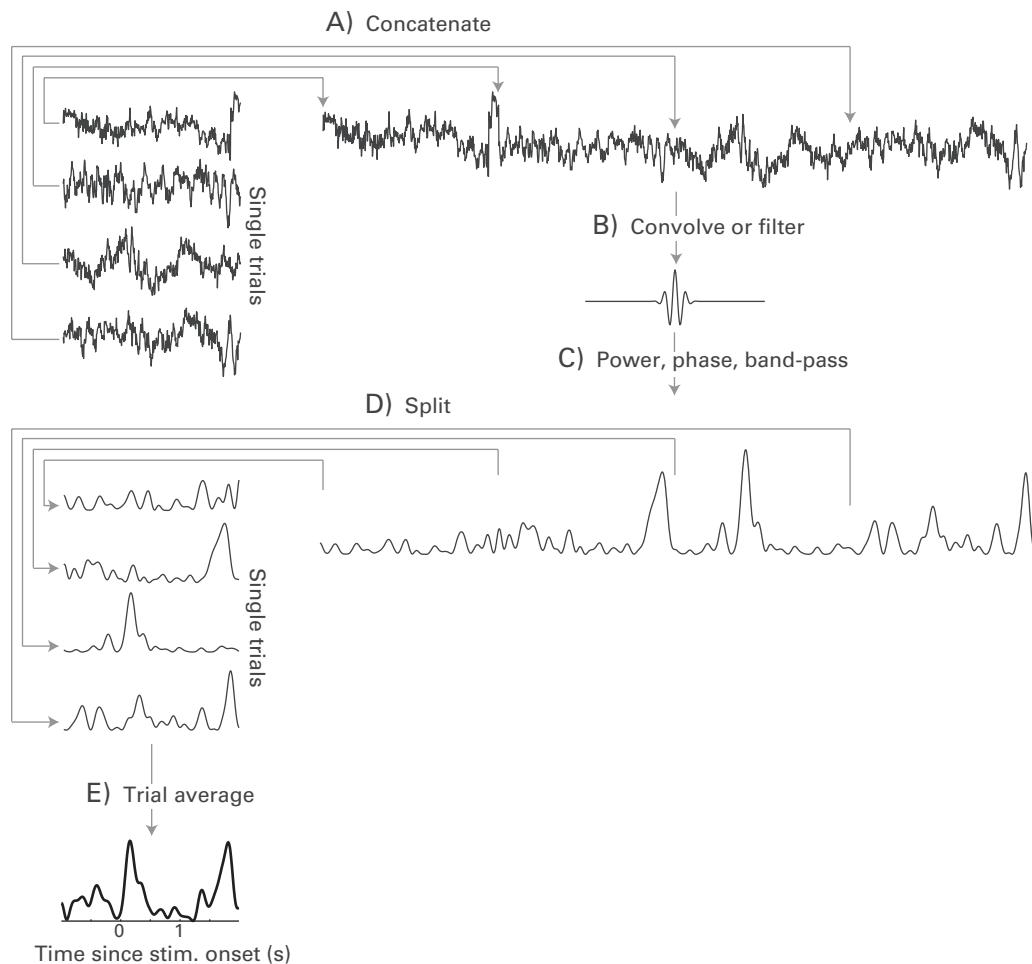
**Figure 14.11**

Illustration of concatenation procedure to improve performance of time-frequency decomposition. This method is appropriate for wavelet convolution and the filter-Hilbert method. Single trials are concatenated to produce one long time series (panel A), which is then convolved with a complex Morlet wavelet or is filter-Hilbertized (panel B). If convolution is used, you should zero-pad the concatenated time series so it has a length of a power of 2 to decrease FFT computation time. Power, phase, or the bandpass-filtered signal is then extracted from the entire time series (panel C), and thereafter the concatenated time series is reshaped back to individual trials (panel D) for trial averaging (panel E) or for any other analyses. Sharp transitions at the boundaries between trials will produce edge artifacts, but if there is sufficient buffer zone at the beginning and end of each trial, the edge artifacts will not contaminate the results (edge artifacts may also be present in nonconcatenated trials).

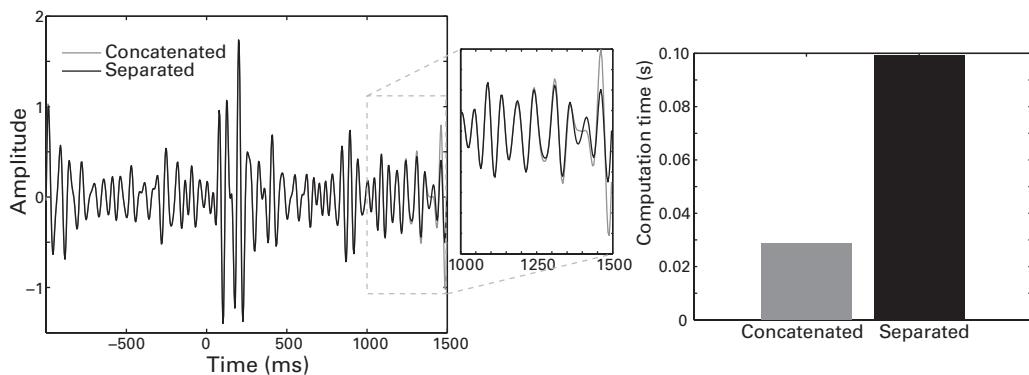


Figure 14.12

Filtering concatenated trials is more elegant and efficient than filtering single trials but produces minor artifacts toward the edges of the data (these are avoided in single-trial filtering due to signal reflection in the Matlab `filtfilt` function). The more trials you have, the bigger the increase in speed.

to make smoother and more visually interpretable time-frequency plots. Frequency overlaps of between 25% and 75% are acceptable. For example, a 50% frequency overlap would mean that the upper and lower bounds for adjacent frequency bands are 4–8 Hz and 6–10 Hz. Keep in mind that a lot of overlap will increase computation time without providing any new information (analogous to the frequency overlap of the 9-Hz and 10-Hz wavelets shown in figure 13.10). Similar to Morlet wavelets, you can create the filters using linear or logarithmic steps. With increasing frequencies, the widths of the filters should also increase. That is, a filter with a center frequency of 6 Hz might have lower and upper bounds of 4.5 and 7.5 Hz, and a filter with a center frequency of 60 Hz might have lower and upper bounds of 45 and 75 Hz.

14.11 A World of Filters

Digital signal filtering is a major topic in engineering and science. There are several types of digital temporal filters, and entire books and online resources are devoted to the advantages and disadvantages of different types of filters. This chapter covers slightly more than the minimum you need to know to construct and apply bandpass filters to use in combination with the Hilbert transform for time-frequency analysis of EEG data. Filtering ERPs is a topic of additional debate, and there are several opinions about whether and how filtering affects ERP features such as peak latency and peak-to-peak quantification. If you would like to read more about this topic, see Luck (2005), chapter 5, the references cited in section 9.2 of this book, and the following references (Acunzo, Mackenzie, and van Rossum 2012; Rousselet 2012).

14.12 Describing This Analysis in Your Methods Section

If you used the filter-Hilbert method to analyze your data, make sure to include enough detail in the Methods section so that someone else can replicate your analyses. Important pieces of information include the minimum, maximum, and number of frequency bands; the pass-bandwidths of the frequencies and whether the bandwidths changed as a function of frequency; the order of the filter and whether the order changed as a function of frequency; and the transition zones of the filter shape. You should also mention which Matlab (or other program) function you used to create the filter kernel (e.g., `fir1` or `firls`). Because poorly constructed filters can have negative consequences for the quality of the filtered data, you should check that the filters were well constructed and note in the paper whether the filters had a good fit to the ideal shape and whether any parameters needed to be modified because of poor filter construction. This can be done through visual inspection or by computing equation 14.1.

If you tested a range of filter parameter settings and found that the results depend on those settings, report which range of parameters you tried and which settings you used for the final analyses.

14.13 Exercises

1. Pick two frequencies (e.g., 5 Hz and 25 Hz) and one electrode and perform complex Morlet wavelet convolution and filter-Hilbert using those two frequencies as the peak/center frequencies for all trials. Plot the resulting power and the bandpass-filtered signal (that is, the real component of the analytic signal) from each method. Plot one single trial (you can choose the trial randomly but plot the same trial for both methods) and then plot the average of all trials. Describe some similarities and differences between the results of the two time-frequency decomposition methods.
2. Modify the wavelet and filter settings (but keep the peak/center frequencies the same) until these two methods produce very similar results. Next, modify the wavelet and filter settings (except the peak/center frequencies) to make the results different (stay within a reasonable range of parameter settings; they do not need to look dramatically different). Which parameters did you change to make the results look more similar versus more different? How different are the results, and would you consider this a meaningful difference? What does this difference tell you about when to use specific parameter settings for wavelet convolution and the filter-Hilbert method?

15 Short-Time FFT

The short-time FFT is an alternative method for extracting time-frequency power and phase information. The short-time FFT is a simple extension of the Fourier transform that addresses the two main limitations of the Fourier transform for EEG time-frequency analyses that were identified in chapter 11: the Fourier transform obscures time-varying changes in the frequency structure of the data, and the Fourier transform assumes that the data are stationary for the duration of the time series. Any method for computing the Fourier transform could be used, but the fast Fourier transform (FFT) is the only method discussed in this chapter.

15.1 How the Short-Time FFT Works

Once you understand how a Fourier transform works and how to extract power from a complex analytic signal, the short-time FFT method is straightforward: use the FFT to extract the frequency structure of brief segments of data (time windows) rather than the entire time series (Makeig 1993; Welch 1967). Figure 15.1 shows an overview of the short-time FFT method. You can see in figure 15.1D that this method produces a time-frequency map that is similar to what you can obtain from complex Morlet wavelet convolution or the filter-Hilbert method.

From this two-dimensional (2-D) result (time by frequency), you can extract the time course of power at one frequency (figure 15.2A) or the power spectrum at one time point (figure 15.2B). The results presented in figure 15.2 are averaged across trials.

15.2 Taper the Time Series

Before computing the Fourier transform of each time segment, you should taper the data in that segment. This will attenuate the amplitude of the data at the beginning and at the end

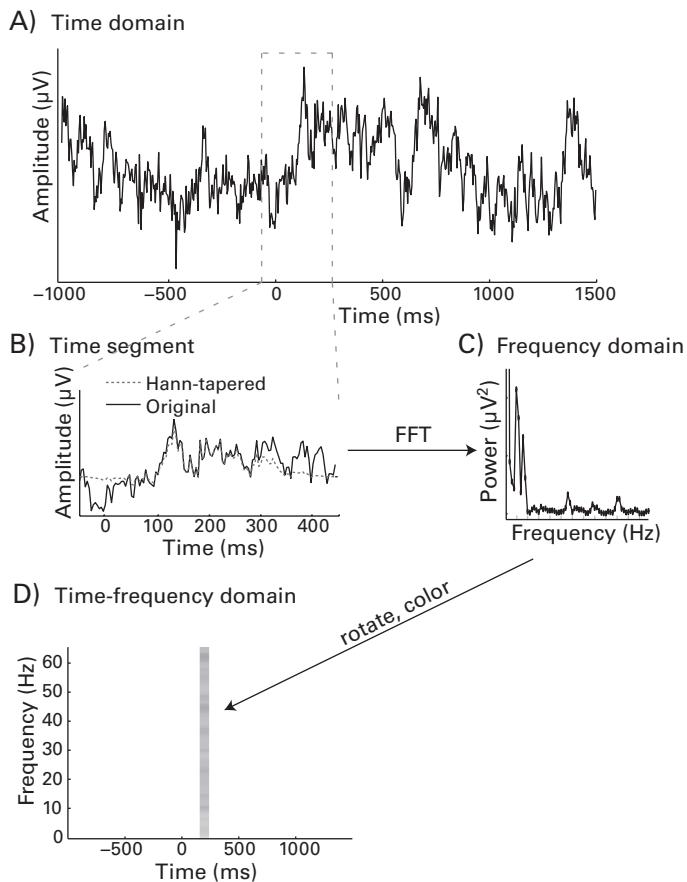


Figure 15.1

Overview of the short-time FFT method. From the time series data (panel A), a small segment of the data, comprising a few hundred milliseconds, is taken (panel B). A windowing taper is applied to that segment to minimize the possibility of edge artifacts, and then the Fourier transform of the tapered time series is taken (panel C). The power spectrum of that segment is then placed into a time-frequency space with the frequencies corresponding to that of the FFT and the time point corresponding to the center time point of the time segment from panel B. It is not necessary to perform these procedures separately for each trial because the Matlab function `fft` can accept a 2-D input (e.g., time by trials). Make sure you take the `fft` over time, not over trials. The FFT returns as many frequencies between DC and the Nyquist frequency as there are time points in the segment, although you can keep only the frequencies of interest, such as 4 Hz to 60 Hz. For resting-state data, the same procedure could be applied, but the FFT results would be averaged over time in panel D (e.g., see figure 2 in Allen and Cohen 2010).

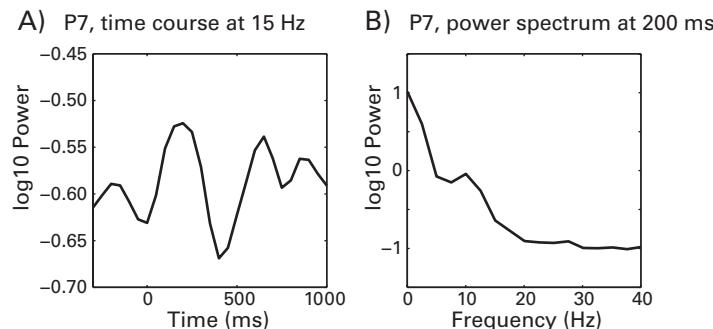


Figure 15.2

Power over time (panel A) and power structure at one time segment (panel B) obtained from the short-time FFT method. No baseline transformation was applied in these plots.

of the time segment (figure 15.1B), which is important for preventing edge artifacts from contaminating the time-frequency results. Note that this means that large buffer zones at the start and end of each trial are not necessary for the short-time FFT method. The downside of tapering is that it attenuates valid EEG signal, but this effect can be mitigated by using temporally overlapping segments. This is discussed in section 15.3.

There are several possible tapers to use, including Hann (also sometimes called Hanning although it is named after someone named Hann), Hamming (named after someone named Hamming), and Gaussian (named after Gauss). The Hann window is advantageous because it tapers the data fully to zero at the beginning and end of the time segment. This eliminates any possibility of even minor edge artifacts. The Gaussian can be made to taper to zero, but the window becomes fairly narrow, which may excessively taper the data. The Hamming window does not fully taper the data to zero. A comparison of the Hann, Hamming, and Gaussian windows is shown in figure 15.3. There are Matlab functions that generate these tapering windows, although the formulas are fairly simple and are presented in the online Matlab code.

Other tapers that could be used for the short-time FFT include Kaiser, cosine, and Blackman. You can test different windows exhaustively, but it is unlikely that they will provide significantly different results compared to the Hann window.

Because the number of frequencies returned by the FFT is equal to $N/2 + 1$, where N is the number of time points in each time segment, there will probably be more frequencies than you want to include in the analyses. For example, if you have segments that are 500 ms long and the sampling rate is 1000 Hz, the FFT at each time segment will contain

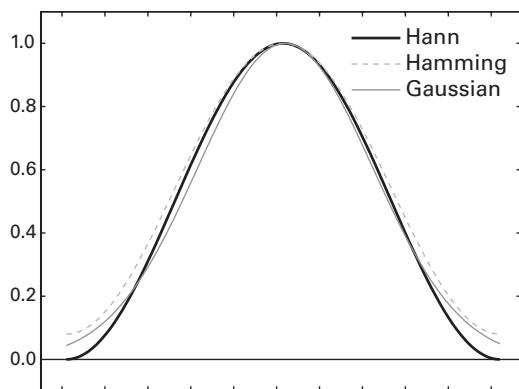


Figure 15.3

Different tapering functions often used for the short-time FFT in EEG data. The two features to look for in a taper are their width—narrow tapers attenuate more signal—and how close they are to zero at the beginning and at the end of the window.

250 frequencies spaced between zero and 500 Hz (the Nyquist frequency). Thus, you can select only those frequencies that you want to analyze, for example, from 4 Hz to 60 Hz in 30 steps (the “requested frequencies”). Sampling the requested frequencies is necessary if the length of the time segment changes as a function of frequency (discussed below) because the indices corresponding to each frequency will vary with the length of the time segment.

When you extract the requested frequencies, you can either extract the single closest frequency bin to each requested frequency or you can take the average of several frequency bins surrounding each requested frequency (figure 15.4). Taking an average is a good strategy because it increases the signal-to-noise ratio and minimizes the possibility that the frequency estimate is driven by outliers or nonrepresentative data. Note that taking the average power from several neighboring frequencies makes the result of the short-time FFT method more similar to results from the filter-Hilbert method. To make the results from the short-time FFT method more similar to the results of complex wavelet convolution, you could average Gaussian-weighted neighboring frequencies.

It was mentioned earlier that buffer zones are not necessary for the short-time FFT method. However, you still need to create longer epochs than you would for ERP-only analyses because of the time segment lengths. For example, if you want to estimate power at -200 ms (e.g., to estimate baseline power), you might need the epochs to start at -600 ms (assuming a time segment length of 800 ms for lower frequencies).

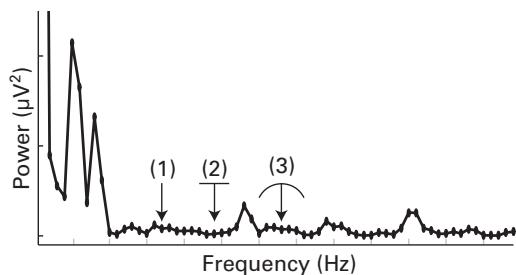


Figure 15.4

Illustration of different ways of extracting frequencies from the FFT of each time segment. Because the FFT returns more frequencies than you probably would want to include in the analyses, you can select “requested frequencies” from the full FFT result. To obtain the requested frequencies you can either select the single frequency bin closest to each requested frequency (arrow 1), average neighboring frequencies (arrow 2), or apply a distance weighting such as a Gaussian to neighboring frequencies to compute a weighted average (arrow 3). The latter two options increase the signal-to-noise ratio of the results of the short-time FFT. In practice any one of these methods can be used, but it should be consistently applied to all frequencies and all data within a study.

15.3 Time Segment Lengths and Overlap

The length of the time segment offers the much-discussed trade-off between temporal and frequency precision and resolution: shorter time segments provide better temporal precision at the expense of frequency precision and resolution, whereas longer time segments provide better frequency precision and resolution at the expense of decreased temporal precision. Remember from chapter 11 that the frequency resolution of a Fourier transform is defined by the number of points in the time series. Therefore, the larger the time segment, the more frequencies can be extracted, and thus the greater the frequency resolution. The temporal resolution is always defined by the data sampling rate and thus does not depend on the length of the time segment.

The choice of time segment length is also related to the lowest frequency you want to analyze. The segment must be long enough to capture at least one cycle of the lowest frequency, and preferably more than one cycle to increase signal-to-noise ratio. Thus, if you want to analyze 3-Hz activity, the window must be at least 333 ms long, and preferably 667 ms or 999 ms to capture two or three cycles (or any length; time segment lengths need not be integer multiples of cycle durations). On the other hand, 999 ms might be so long as to average over and thus fail to capture transient higher-frequency activity.

To adapt the short-time FFT method such that the trade-off between temporal precision and frequency precision changes as a function of frequency, you can change the time segment length as a function of frequency: lower frequencies can have longer time segments while higher frequencies can have shorter time segments. This procedure will have a similar effect to wavelet convolution with a variable number of cycles (the width of the Gaussian) as a function of frequency. In this case the FFT will be performed several times on time segments around each time point. Each time the FFT is performed, a range of frequencies is extracted. For example, for a center time point of 300 ms, the FFT can be computed on the data from -100 to +700 ms, and frequencies between 4 and 20 Hz are extracted. Next, the FFT is computed on the data from 0 to 600 ms, and frequencies between 20 Hz and 40 Hz are extracted. And so on. This is a simplified example; you could also change the length of the time segment for each requested frequency.

A second parameter of the short-time FFT is the amount of overlap between successive time segments. Having temporal overlap is useful for three reasons: it improves the temporal precision, it mitigates the loss of signal due to tapering, and it smoothes the time-frequency plots, thus facilitating visual inspection of the time courses of activities. Although there are no rules or strict guidelines for how much overlap to use, values between 50% and 90% of the length of the segment are acceptable. For example, steps of 75 ms for a 300-ms time segment would result in a 75% overlap.

15.4 Power and Phase

Similar to complex wavelet convolution and the filter-Hilbert method, the short-time FFT method provides information about power and phase. Power values can be used and interpreted in the same way as power values from wavelet convolution and filter-Hilbert. Phase values from the short-time FFT, however, are not the same as the phase values from wavelet convolution or filter-Hilbert. Whereas the phase-angle-time series from wavelet convolution or filter-Hilbert is an estimate of the instantaneous phase value at each time point, the phase value from the short-time FFT method is the phase parameter of the sine wave at each frequency, in other words, the *y*-axis intercept of the sine wave at zero on the *x*-axis (figure 15.5). This concept was also discussed around figure 11.5.

Despite the different meaning of the phase values from the short-time FFT compared to wavelet convolution or filter-Hilbert, the phase values from the short-time FFT can often be used in similar ways as the phase angle time series extracted from complex wavelet convolution or the filter-Hilbert method. One example is with intertrial phase clustering, which is a measure of the consistency of band-specific activity over trials and is the main topic

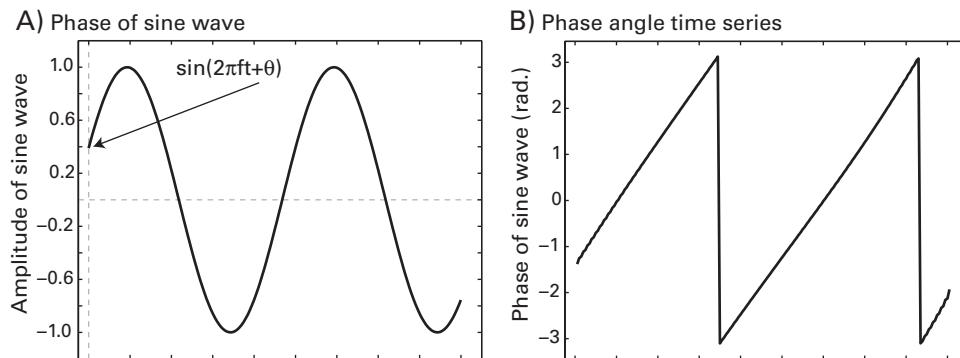


Figure 15.5

The difference between the phase values returned from the short-time FFT method (panel A) and wavelet convolution or filter-Hilbert (panel B). The short-time FFT returns one phase value per time window, which corresponds to the phase parameter of a sine wave. The phase angle time series, on the other hand, is a vector of phase angle values (one per time point) corresponding to the position along the sine wave at that time. These two values have different interpretations but can often be used in the same way in analyses, as shown in figure 15.6 and discussed chapter 19.

of chapter 19 (figure 15.6; how this analysis was done and how it can be interpreted are discussed in chapter 19).

15.5 Describing This Analysis in Your Methods Section

If you use the short-time FFT approach to analyze your data, make sure the Methods section in your paper clearly states the relevant parameters so that someone else could replicate your analyses. The relevant parameters include the overlap between successive time segments, how many frequencies were extracted (that is, the requested frequencies), whether the time segment length changed as a function of requested frequency, whether neighboring frequencies were averaged together to increase signal-to-noise ratio, and finally, which function was used to taper the data. If you used a software package or toolbox such as eeglab, write which functions were used, and any parameters that were changed from their default settings.

15.6 Exercises

1. Compute the short-time FFT at each electrode and make topographical maps of theta-band (around 6 Hz) power and alpha-band (around 10 Hz) power at 150 ms and 700 ms.

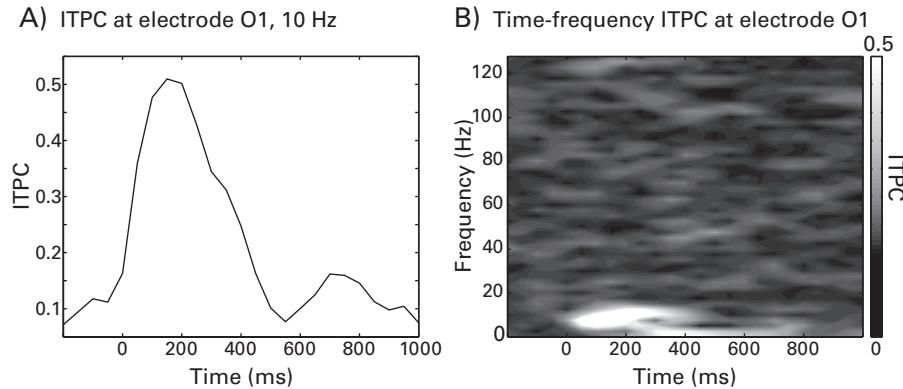


Figure 15.6

Intertrial phase clustering (ITPC), a measure of the consistency of phase values at each time-frequency point over trials, computed using the phase information from the short-time FFT method. Panel A shows a time course of ITPC at 10 Hz from electrode O1, and panel B shows a time-frequency plot of ITPC from O1. Details of the methods and interpretation of ITPC are presented in chapter 19.

2. Select one electrode and one frequency and compute power over time at that electrode and that frequency using complex wavelet convolution, filter-Hilbert, and the short-time FFT. Plot the results of these three time-frequency decomposition methods in different subplots of one figure. Note that the scaling might be different because no baseline normalization has been applied. How visually similar are the results from these three methods? If the results from the three methods are different, how are they different, and what parameters do you think you could change in the three methods to make the results look more or less similar?

16 Multitapers

In chapter 15 you learned that the short-time FFT has advantages over the FFT in terms of its ability to temporally localize changes in frequency structure. But because edge artifacts will contaminate the frequency representation, each time segment must be tapered such that the signal at the start and at the end of the segment is attenuated to zero. Although this entails a loss of signal, performing FFTs over a moving sequence of overlapping segments helps mitigate the signal degradation. The multitaper approach is an extension of the short-time FFT method that is designed to increase the signal-to-noise ratio of the frequency representation by applying several tapers that have slightly different temporal (and thus spectral) characteristics (Mitra and Pesaran 1999; Thomson 1982).

The multitaper method is particularly useful for situations of low signal-to-noise ratio, such as higher-frequency activity or single-trial estimates of power. For frequencies lower than around 30 Hz, the multitaper method may be less appropriate because the signal-to-noise ratio is already relatively high and because the spectral smoothing resulting from the multitaper procedure may impede frequency isolation, meaning that activities from multiple frequency bands become averaged together.

16.1 How the Multitaper Method Works

Figure 16.1 shows an overview of the multitaper method and a comparison between the multitaper method and the short-time FFT. The multitaper method begins similarly to the short-time FFT: from the entire trial period, one time segment is extracted (in this case, a 400-ms time segment). This time segment is shown in figure 16.1A. Next, the data from that time segment are multiplied by a series of tapers (panel B), resulting in several tapered time series (panel C; section 16.2 discusses the tapers in more detail). You can see that the different tapers concentrate signal in different regions of time. Next, the FFT of each tapered time series is taken (panel D), and the resulting spectra are averaged together (panel E). This

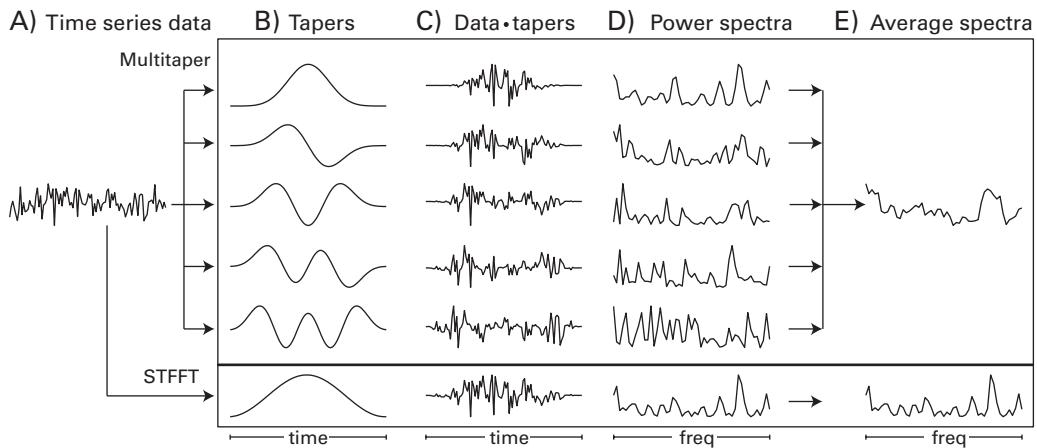


Figure 16.1

Overview of the multitaper method and comparison with the short-time FFT approach. The y-axes were arbitrary scaled to facilitate visual comparison. Note that the x-axes show time in columns A, B, and C and frequency in columns D and E. The dot between data and tapers in the label of panel C indicates the pointwise multiplication.

procedure can be contrasted with the short-time FFT approach, depicted in the bottom row of figure 16.1: the short-time FFT approach uses only one taper (in this case, a Hann window) and one FFT. In this example it is clear that the power spectrum from the multitaper is smoother than the power spectrum obtained from the short-time FFT. It is also clear that conceptually, the short-time FFT approach and the multitaper method differ only in that the multitaper approach uses multiple tapers.

Figure 16.1 is one step of the multitaper method. It would replace panels B and C in figure 15.1. To compute a time-frequency map, you would follow the rest of the procedures outlined in figure 15.1. For example, figure 16.2 shows the results from the same data as shown in figure 15.2, but from the multitaper method.

The multitaper method can be used to extract both power and phase information. Because the phase value per frequency may change with each taper, the phase values obtained from the multitaper approach reflect an average phase from the tapered data time series. As with the short-time FFT method (chapter 15), the phase values here refer to the phase of the sine wave in that time segment at each frequency rather than the ongoing phase-angle time series. Also as with the short-time FFT, phase values from the multitaper method can be used for many phase-based analyses.

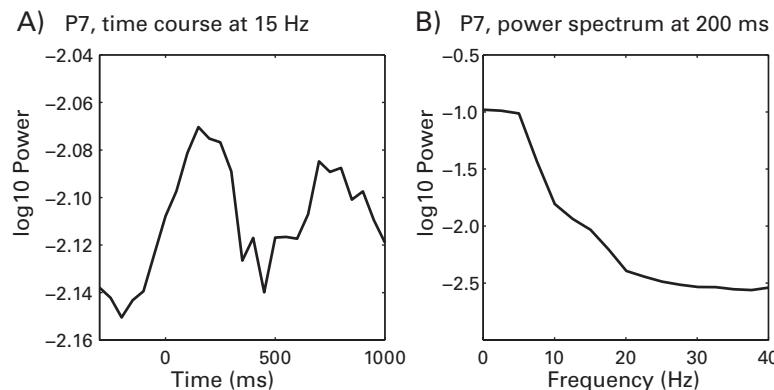


Figure 16.2

Same data as shown in figure 15.2, but the results were obtained via the multitaper method instead of the short-time FFT method.

16.2 The Tapers

The tapers used for the multitaper method are called discrete prolate spheroidal sequences and can be obtained via the Matlab function `dps` (in the signal processing toolbox). They are also often called Slepian tapers (Slepian 1978). Slepian tapers are orthogonal to each other (that is, the dot product of any one taper with any other taper is 0), and they have slightly different frequency characteristics, thus focusing the spectra of the resulting tapered time series on different parts of the spectrum.

The multitaper method introduces some smoothing on the frequency axis. The amount of smoothing depends on the number of tapers, such that the more tapers that are used, the smoother the results. This should be apparent from figure 16.1. Frequency smoothing may be beneficial for higher-frequency activity, in which the spectrum tends to be broader than at lower frequencies. Frequency smoothing is also beneficial to account for individual differences. That is, if two subjects in your experiment have gamma-band peaks at 50 Hz and 65 Hz in the same task, the frequency precision of a complex Morlet wavelet with 10 cycles may identify the unique bands of these two subjects so precisely that the cross-subject averaging fails to identify the task-related gamma band activity at the group level. In this situation it is possible that frequency smoothing will facilitate the group-level identification of the higher-frequency activity.

The multitaper method also introduces temporal smoothing, particularly when compared to wavelet convolution or the filter-Hilbert method. Although all time-frequency

methods involve temporal smoothing because the estimate of frequency-band-specific activity at each time point reflects a weighted average of activity at surrounding time points, the multitaper method involves more temporal smoothing because of the multiple tapers with different temporal foci within each time segment. This can be seen by considering figure 16.1C. Similar to the frequency smoothing, this temporal smoothing might be beneficial in the identification of high-frequency activity, particularly in cross-subjects group-level averaging. Consider that if increases in gamma activity occur in bursts that are neither time-locked nor phase-locked to the time = 0 event, those bursts may not survive cross-trial and cross-subject averaging. If this is the case, temporal smoothing resulting from applying multiple tapers will facilitate the identification of the non-phase-locked gamma-power increases, because those gamma-power bursts will be smoothed over time and thus more easily identifiable and quantifiable.

The number of tapers that the Matlab function `dpss` will return is computed based on the desired spectral smoothing and the length of the time segment. Specifically, the product of the desired segment length (in sample points) and the frequency bandwidth (as a fraction of the sampling rate) is the second input argument to the Matlab `dpss` function, and the number of tapers is twice this product, rounded to the nearest integer. The product of the segment length and the desired spectral smoothing can be kept at a constant (generally, this product is kept between 2 and 4), or it can be allowed to vary as a function of frequency band. This latter option improves the time-frequency trade-off between temporal and frequency precision as a function of frequency. This also means that the length of the time segment will change as a function of frequency. In the former situation, you can simply specify this integer in the second input to the Matlab `dpss` function.

The last taper of the sequence generally has a relatively poor spectral representation and thus can be omitted from the analysis (Mitra and Pesaran 1999). This can be seen by plotting the second output of `dpss`, which is related to the energy in the specified band and ideally should be close to 1.0. This vector typically has $n - 1$ numbers that are close to 1, and a final number that is lower, around 0.7 or 0.8. Avoid using tapers that have these low ratio values.

How much spectral smoothing is appropriate? There is no firm answer. You can choose smoothing parameters based on published research with similar data and a similar experiment design as your study, or based on visual inspection of the results (Mitra and Pesaran 1999). There is a trade-off between smoothness and accuracy, such that smoother results (thus, more tapers) decrease the accuracy of the frequency characteristics of the data. The results should be smoothed enough to increase the signal-to-noise ratio, but not at the expense of losing sensitivity to detect time-frequency features. The same smoothing criteria

should be applied to all electrodes and conditions to minimize the possibility that a bias in frequency estimation will affect the results.

As with the short-time FFT method, the multitaper method will return as many frequency bins between zero and the Nyquist frequency as there are time points in each segment. Thus, you have the option of selecting requested frequencies based on a single frequency bin or the average of neighboring frequency bins (figure 15.4). However, because the multitaper method already involves spectral smoothing and averaging, it may not be necessary to further average over frequency bins when extracting power from the multitaper-averaged spectra.

The function `dpss` is in the signal-processing toolbox. Thus, you will need access to this toolbox to perform a multitaper analysis. The Matlab signal-processing toolbox also includes a function called `pmtm` (Power via MultiTaper Method), which performs a multitaper analysis on inputted time series data, similar to the analysis shown in this chapter.

16.3 When You Should and Should Not Use Multitapers

There are three situations in which the multitaper approach may be beneficial to your results. The first situation is if you have noisy data or a small number of trials and are concerned about the influence of noise on the results. The second, and related, situation is if you are performing single-trial analyses, particularly if you would like to analyze frequencies above around 30 Hz. The third situation is if you would like to focus on high-frequency power, particularly above around 60 Hz. The multitaper approach will help increase the signal-to-noise ratio of high-frequency activity, and the temporal and frequency smoothing may facilitate cross-subject averaging. Several EEG analysis toolboxes will perform a multitaper-based time-frequency decomposition. However, the Matlab toolbox `fieldtrip` has the most active development of this method, including, for example, options to create frequency-band-selective tapers that further enhance their sensitivity beyond the basic multitaper application discussed in this chapter.

There are two situations in which the multitaper approach might not be the preferred method. The first situation is if you will focus the analyses on frequencies below around 30 Hz. The smoothing at these low frequencies can make it difficult to isolate discrete time-frequency events, particularly if there are several events that are close to each other in time-frequency space. The second situation is if you will perform analyses in which precise timing of high-frequency activity is required. Because the temporal precision of the multitaper method is relatively low compared to wavelet convolution or the filter-Hilbert method (as discussed in section 16.2), the timing of high-frequency activity is more difficult

to determine. Although this can be beneficial for identifying non-phase-locked power, it can be detrimental to, for example, cross-frequency coupling (chapter 30), for which temporal precision of high-frequency power should be as high as possible.

16.4 The Multitaper Framework and Advanced Topics

Multitapers should be seen as a general framework for a set of possible analysis approaches. The material presented in this chapter provides a basic introduction to the multitaper framework. More sophisticated applications of multitaper include, for example, varying the lengths of the time segments, the number of tapers, and the center frequency of the spectral representation of the tapers, as a function of the center frequencies (some of these features are included in the fieldtrip toolbox; Oostenveld et al. 2011). Further, there are more advanced methods for estimating the appropriate frequency smoothing and number of tapers to use, adaptively weighting the contribution of different frequency bands to optimize averaging, and performing statistical evaluation of whether spectral peaks are statistically significant (Thomson 1982). For further reading, consider the book *Observed Brain Dynamics* (Mitra and Bokil 2007). For more information about the multitaper approach more generally, see the book *Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques* (Percival and Walden 1993).

16.5 Describing This Analysis in Your Methods Section

There are several important details about the multitaper method that should be mentioned in the Methods section, including the length of the time segment, the number of frequencies that were extracted, the number of Slepian tapers, and the resulting smoothing. If the analysis was done in a Matlab toolbox such as fieldtrip, state whether the default parameters were used and list any nondefault parameters otherwise.

16.6 Exercises

1. Pick one electrode and compute a time-frequency map of power using both the multitaper method and the short-time FFT. Store all of the power values for all of the trials.
2. Next, compute a time-frequency map of signal-to-noise ratio. The signal-to-noise ratio of power is discussed more in chapter 18, but it can be estimated as the average power at each time-frequency point across trials, divided by the standard deviation of power at each time-frequency point across trials.

3. Make time-frequency plots of power and signal-to-noise ratio from the two methods. Make another plot in which you subtract the signal-to-noise plots between the two methods. Are there any noticeable differences between the signal-to-noise results when the multitaper method versus the short-time FFT is used?
4. Select two frequencies, one relatively low and one relatively high (e.g., 8 Hz and 60 Hz), and compare the power time series and signal-to-noise time series in these frequency bands from the two methods in a separate figure, using line plots. Comment on the differences if there are any.

17 Less Commonly Used Time-Frequency Decomposition Methods

As the title suggests, this chapter introduces you to the ideas behind some additional approaches for estimating time-frequency dynamics of EEG data. These methods are not widely used but may be of interest for specific or exploratory applications. References are provided for readers interested in learning more or applying these methods. Topics are listed alphabetically.

17.1 Autoregressive Modeling

Autoregressive modeling is the basis of Granger prediction and is discussed in more detail in chapter 28. An *autoregressive model* is one in which values of a signal are predicted from previous values of that signal. The number of previous values is called the order and must be carefully selected because models with too small or too large an order may cause poor fits of the model to the data. Rhythmic patterns in temporal lags can be converted from sample points to frequencies in hertz. To estimate frequency-band-specific power from the autoregressive model, the time series autocovariance is examined as a function of lag; the strength of the autocovariance is related to power at that frequency band. When it was first introduced into the EEG analysis literature, one advantage of autoregressive modeling was that, unlike the FFT, its frequency resolution is not limited by the number of data points in a time segment. Wavelet convolution and the filter-Hilbert method also share this advantage. Autoregressive modeling is infrequently used, and has largely been replaced by other methods such as wavelet convolution. If you would like to learn more about autoregressive modeling for time-frequency analyses, you can start with Ding et al. (2000) and Florian and Pfurtscheller (1995).

17.2 Hilbert-Huang (Empirical Mode Decomposition)

The Hilbert-Huang method was developed for detecting time-frequency events in nonstationary data (Huang et al. 1998; Lin and Zhu 2012; Pigorini et al. 2011). It does not rely

on covariance-with-template procedures such as wavelet convolution, filtering, or Fourier transform. Rather, it acts as an adaptive filter by using a data reduction technique called empirical mode decomposition to decompose the raw EEG signal into a series of fundamental components (also called intrinsic mode functions). Briefly, empirical mode decomposition involves identifying local minima and maxima of a time series, creating new time series by interpolating across the local minima and maxima (after some adjustments known as “sifting”), and subtracting the mean from the original signal. This mean-subtracted signal is called an intrinsic mode function. After the subtraction, local minima and maxima are identified from the intrinsic mode function, and the process is repeated until there are only a few minima and maxima left in the signal. The “sifting” procedure is slightly more involved than described here, and you can read more about it in any of the papers cited in this section, and in Sweeney-Reed and Nasuto (2007). Intrinsic mode functions can be conceptualized as basis time series, to which the Hilbert transform can be applied, and from which phase, power, and frequency information can be extracted.

The Hilbert-Huang method is not often used in cognitive EEG research, although it has been applied to brain-computer interfaces (Panoulas, Hadjileontiadis, and Panas 2008), detecting epileptic spikes (Oweis and Abdulhay 2011), and testing the possibility that ERPs arise in part due to phasic changes in the frequency structure of the EEG (Burgess 2012). The main advantage of empirical mode decomposition is that you do not need to specify in advance which frequencies to search for in the data; rather, the frequencies that are naturally present in the data will emerge through the empirical decomposition (Sweeney-Reed and Nasuto 2007). It is therefore a useful approach for identifying changes in instantaneous frequency (by scaling the temporal derivative of phase by the sampling rate), particularly in data that are non-stationary (empirical mode decomposition does not assume signal stationarity). Some disadvantages are that the frequency structure of the decomposition is likely to differ across subjects, and that the result may be influenced by low-pass filtering the data.

Empirical mode decomposition is not further discussed in this book, but the online Matlab code contains a function (`emdx.m`) that will perform an empirical mode decomposition analysis on 1-D, 2-D, or 3-D EEG data.

17.3 Matching Pursuit

Matching pursuit is an adaptive algorithm that, like the Hilbert-Huang transform, is based on decomposing the signal into more basic components. Matching pursuit, unlike the Hilbert-Huang, does involve a template-matching procedure, although it works differently from how

wavelet convolution or the filter-Hilbert works. Rather than convolve all wavelets with the data, matching pursuit involves finding which template from the “dictionary” best matches the data at that time window and computing the residuals between the data and all templates. The templates are also called atoms and are often Morlet wavelets or similar kinds of tapered frequency-band-specific signals. Similar to empirical mode decomposition, it may be most useful in experiments in which rapid changes in frequency is expected (e.g., Ray and Maunsell 2010). For more information about Matching pursuit and its use in neuroscience, see Benar et al. (2009), Jmail et al. (2011), and Mallat and Zhang (1993).

17.4 P-Episode

The idea of the *p-episode* (or P_{episode}) technique is to detect the occurrence and duration of oscillatory events. This is done by filtering the data into frequency bands (with wavelet convolution or filter-Hilbert) and detecting, on a trial-by-trial basis, whether a power fluctuation exceeds an amplitude threshold (figure 17.1). A duration threshold may also be applied. This method does not explicitly measure the amplitude of power, although the amplitude must be large enough to exceed the threshold. The threshold can be computed from all time points in the experiment, a baseline period, or a control condition. In figure 17.1, the threshold was selected as 95% of the distribution of power at all time points from all trials.

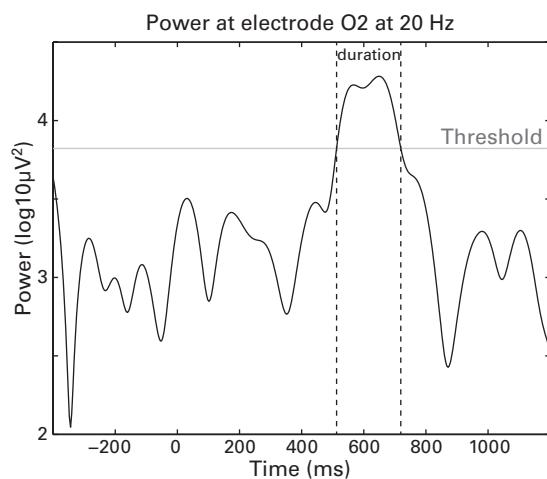


Figure 17.1

Illustration of the p-episode technique. The power time course is plotted from trial 50 (this trial was selected for illustration based on visual appearance of the results).

The main advantage of the p-episode method is that it detects the temporal duration of a band-specific increase in power. This is unique among time-frequency decomposition methods, which all focus exclusively on the amplitude of the signal. Thus, the p-episode method is particularly useful if you have hypotheses concerning the duration of power increases, particularly at the single-trial level. The p-episode technique may also be useful for detecting high-frequency power. For example, if gamma power occurs in bursts that are task related but neither time-locked nor phase-locked to the time = 0 event, those bursts may not survive cross-trial and cross-subject averaging. The p-episode technique has increased sensitivity to detecting such brief but high-amplitude events.

There are three limitations of this technique. First, because it is based on detection of suprathreshold events, the choice of threshold will influence the results. Second, because noise will produce transient increases in power, excessively noisy data might not be appropriate for the p-episode technique. Computing condition differences should help in these situations because the noise would ideally be equally distributed across conditions. Third, increases in power that have a modest amplitude but are consistent across trials might not be detected by the p-episode technique, although they would be apparent in trial averaging of, for example, wavelet convolution. For more information about this method, see Caplan et al. (2001), Montez et al. (2009), and van Vugt, Sederberg, and Kahana (2007).

17.5 S-Transform

The S-transform was developed as an adaptation of the short-time FFT, in particular to address the limitation that the short-time FFT may have limited sensitivity for detecting transient events that are shorter than the length of the FFT time segment (Stockwell, Mansinha, and Lowe 1996). The S-transform works the same way as a wavelet convolution but uses a slightly different kernel. The kernel of the S-transform is a tapered sine wave, similar to the Morlet wavelet. The taper of the S-transform has a Gaussian-like shape, although it differs slightly from a formal Gaussian as defined by equation 12.1. By selecting a specific number of wavelet cycles, Morlet wavelets as described in this book and the S-transform provide nearly identical wavelets and thus will provide nearly identical results (see figure 17.2). It is therefore appropriate for use in time-frequency decomposition (Jones et al. 2006; Pinnegar, Khosravani, and Federico 2009). The main disadvantage of the S-transform for EEG time-frequency analyses is that it does not contain a parameter to change the width of the wavelet as a function of frequency (see equation 12.2).

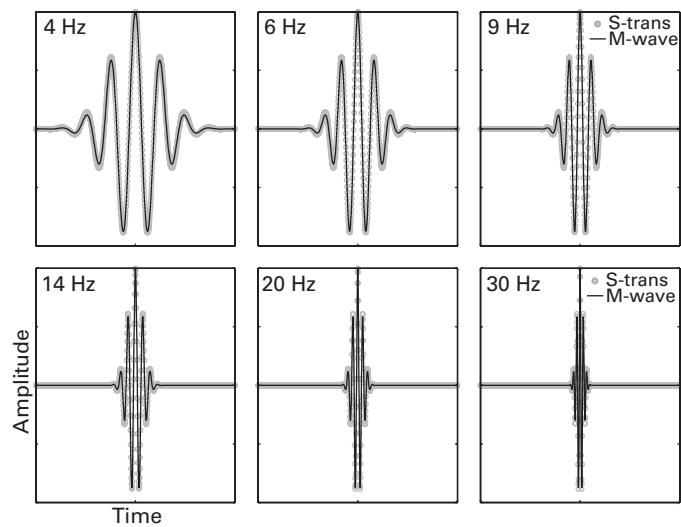


Figure 17.2

Comparison of the real component of the S-transform ("S-trans"; gray dots) and the Morlet wavelet ("M-wave"; black lines) for several frequencies. Morlet wavelets were created using six cycles for all frequencies.

18 Time-Frequency Power and Baseline Normalizations

Now that you know how to extract estimates of time-varying frequency-band-specific power from EEG data using wavelet convolution, filter-Hilbert, short-window FFT, multitaper, or any other method, it is time to learn what to do with those data and how to link fluctuations in power over time and frequency to task events. If you created two-dimensional (2-D) time-frequency power plots in previous chapters, you may have noticed that the plots were difficult to interpret visually because the magnitude of power changed considerably from the lowest frequency to the highest frequency, perhaps even by several orders of magnitude. The purpose of this chapter is to discuss methods for converting time-frequency power data to a scale that is amenable to qualitative visual inspection and quantitative statistical analysis.

18.1 $1/f$ Power Scaling

The reason time-frequency power plots in previous chapters were difficult to interpret is that the color scaling that looked good for one frequency band did not look good for frequency bands that were lower and higher. This is because the frequency spectrum of data tends to show decreasing power at increasing frequencies. This is not specific to EEG data but also characterizes the relationship between power and frequency of many signals, including radio, radiation from the Big Bang, natural images, and many more. This decrease in power as a function of an increase in frequency follows a “ $1/f$ ” shape, which is illustrated in figure 18.1.

The more general form is c/f^x , where c is a constant and x is an exponent (in figure 18.1, $c = 1$ and $x = 1$; you can change c and x to observe the effects on the plot). It is also called a *power law*, because one variable (in this case, EEG time-frequency power) is a power function of another variable (frequency) (note that “power” here refers to raising a number to a power of another number, not the squared magnitude of a complex number). Because

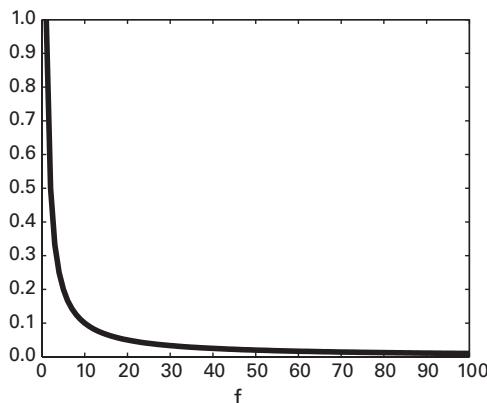


Figure 18.1

A $1/f$ function that characterizes, among other things, EEG power data.

EEG time-frequency power obeys a $1/f$ phenomenon, the power at higher frequencies (e.g., gamma) has a much smaller magnitude than the power at lower frequencies (e.g., delta).

This is why it is difficult to visualize activity from a large range of frequency bands simultaneously. The $1/f$ phenomenon entails five important limitations to interpreting and working with time-frequency power data. Fortunately, as we see later in this chapter, certain baseline normalization methods overcome all of these limitations.

The first limitation is that it is difficult to visualize power across a large range of frequency bands. This was already mentioned and is shown again below. The second limitation is that it is difficult to make quantitative comparisons of power across frequency bands. You might want to determine, for example, whether the power increase in gamma is significantly bigger than the power decrease in alpha. Because the raw power values change in scale as a function of frequency, lower frequencies will usually show a seemingly larger effect than higher frequencies in terms of the overall magnitude. The third limitation is that aggregating effects across subjects can be difficult with raw power values. This is because individual differences in raw power are influenced by skull thickness, sulcal anatomy, cortical surface area recruited, recording environment (e.g., scalp cleanliness and preparation and electrode impedance), and other factors that are independent of the neurocognitive process under investigation (this issue was discussed in section 2.3). The fourth limitation is that task-related changes in power can be difficult to disentangle from background activity (this is also shown below). This is particularly the case for frequencies that generally tend to have higher power or frequencies that tend to have higher power during baseline periods, such as alpha activity over posterior parietal and occipital electrodes. Finally, the fifth limitation is that

raw power values are not normally distributed because they cannot be negative and they are strongly positively skewed. This limits the ability to apply parametric statistical analyses to time-frequency power data.

There are methods that address some of these limitations. For example, the $1/f$ shape can be attenuated by taking the logarithm of the power values (Kiebel, Tallon-Baudry, and Friston 2005) or by modeling the $1/f$ shape and removing that fitted function from the data (Freeman 2006). However, these methods do not address several of the other limitations discussed above.

Figure 18.2 shows time-frequency power plots of the same data using different color scalings. It is clear that no color scale is adequate to show changes in all or even most frequency bands. You can also see in figure 18.2 that computing the logarithm to the base 10 of power helps minimize the $1/f$ scaling a bit but does not eliminate all of the scaling differences over a large range of frequencies. Further, taking the logarithm of the power does not help distinguish background activity from task-related activity.

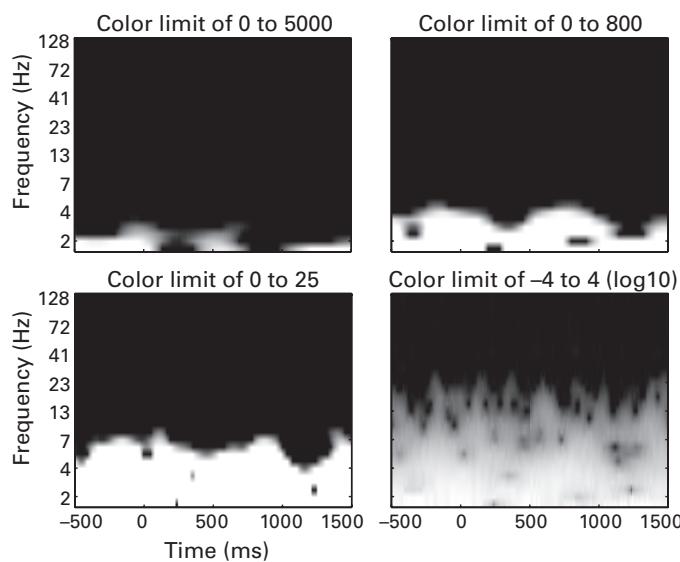


Figure 18.2

Time-frequency power plots showing that without any correction, no color scaling works optimally for all frequency bands. Taking the logarithm to base 10 of the data (log10; lower right panel) helps but does not fully eliminate the difference in scaling over frequencies. Furthermore, it is difficult to distinguish background (dynamics that do not change before and after time = 0) from task-related (dynamics that change as a function of stimulus onset) power.

18.2 The Solution to 1/f Power in Task Designs

Fortunately, there is a solution that addresses all five of the limitations of raw power values identified above. The solution is to use one of several kinds of baseline normalizations, and these are discussed below. All share four advantages:

1. They transform all power data to the same scale. This allows you to compare, visually and statistically, results from different frequency bands, electrodes, conditions, and subjects.
2. Because the normalizations are computed with respect to a (typically pretrial) baseline, task-related time-frequency dynamics become disentangled from background or task-unrelated dynamics.
3. Baseline normalizations put all power results into a common and easily numerically interpretable metric.
4. Because baseline-normalized power data are often normally distributed, parametric statistical analyses may be appropriate to use. This facilitates quantitative group-level analyses and also facilitates integration with other kinds of data, such as behavioral performance, questionnaire results, and demographics.

Below you will learn about the three baseline normalization procedures that are most commonly used in cognitive electrophysiology.

18.3 Decibel Conversion

The decibel (dB) is a ratio between the strength of one signal (frequency-band-specific power) and the strength of another signal (a baseline level of power in that same frequency band). The decibel scale is convenient because it is robust to many limitations of raw power, such as 1/f power scaling and subject-specific and electrode-specific idiosyncratic characteristics. The base unit is called a bel and is the logarithm of a ratio of numbers. Typically, tens of bels are used, hence decibels.

$$dB_{tf} = 10 \log 10 \left(\frac{\overline{activity}_{tf}}{\overline{baseline}_f} \right) \quad (18.1)$$

The horizontal bar over *baseline* indicates the mean across the baseline time period, and *t* and *f* are time and frequency points. Note that the baseline has no *t* subscript, indicating that all time points within a frequency band use the baseline period. After decibel conversion,

the scale and the interpretation of frequency-band-specific power become the change in power relative to baseline rather than the absolute level of power. Because the activity and the baseline within each frequency band are equally affected by $1/f$ scaling, any frequency-band-specific activity or amplitude level that is constant over time, including background activity, will be removed.

What is “the baseline”? The baseline is a period of time, typically a few hundred milliseconds before the start of the trial, when little or no task-related processing is expected. The choice of baseline period is a nontrivial one and influences the interpretation of your results. There are several important issues to consider when selecting a baseline period, which are discussed later in this chapter (section 18.10). For now, a baseline period of -500 to -200 ms before trial onset is used. Figure 18.3 (plate 4) shows an example of decibel-normalized power. These are the same time-frequency power data that were shown in figure 18.2.

Figure 18.4 further illustrates the advantage of using a decibel scale. Power spectra from the same time point are shown separately for the raw power and decibel-corrected power.

The power spectrum in the lower panel of figure 18.4 is easily interpreted, but the power spectrum in the upper panel is not. Note that figures 18.3 and 18.4 show results from one

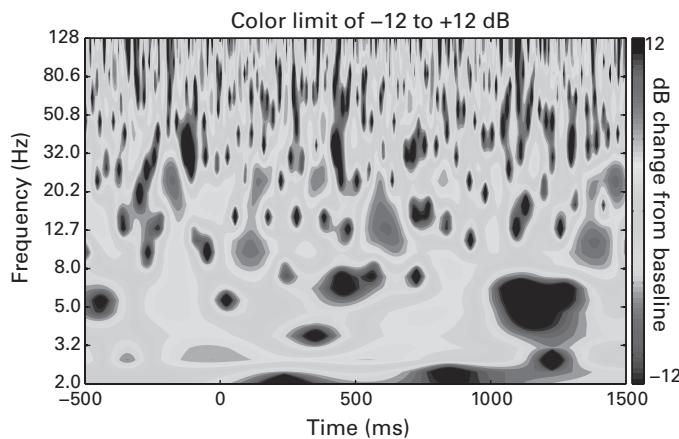


Figure 18.3 (plate 4)

Same time-frequency results that were shown in figure 18.2 but now decibel-transformed relative to a pretrial baseline period of -500 to -200 ms. Note that power across the entire range of frequencies can be compared.

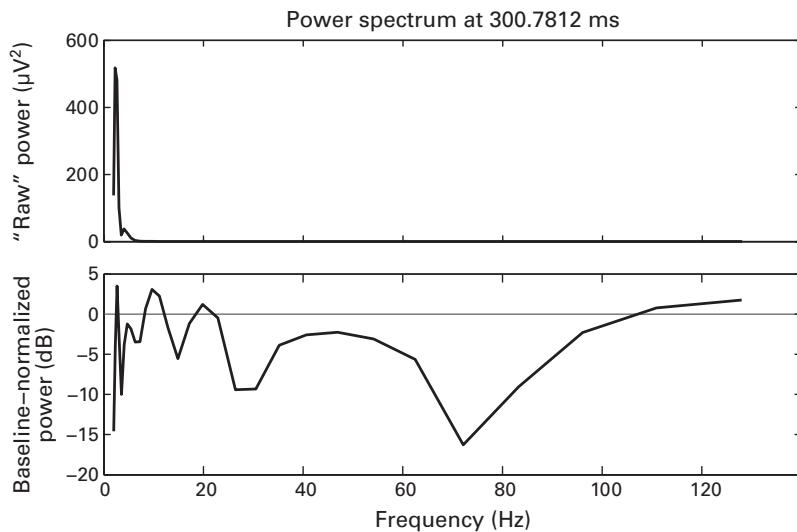


Figure 18.4

Comparison of raw and decibel-converted power at one time point. A horizontal line at 0 dB facilitates comparison of activity with respect to baseline power.

trial, and thus the decibel changes are fairly large for EEG data. Typical decibel values after trial averaging are in the range of $\pm 1\text{--}4$ dB. As is discussed below, for real data analyses, decibel conversion should be done on the trial-average power. That is, first average trials together and then transform to decibels; do not transform each trial to decibels separately and then average.

Figure 18.3 (plate 4) used a symmetric color scaling, meaning that the minimum and maximum colors were mapped to the same magnitude below and above zero (-12 and +12). Although there is no rule for how to set the color scaling, symmetric color scaling should be preferred in most situations. That is, you should use -3 to +3 or -1.5 to +1.5 rather than -3 to +1. In some situations asymmetric color scaling is appropriate and can be used to highlight specific features of the time-frequency plots. However, when a plot has asymmetric color scaling, it is difficult to compare the relative decreases with the relative increases in activity. This is illustrated in figure 18.5 (plate 5), which shows the same time-frequency plots that differ only in the color scaling.

No matter what color scale you use, you should always plot a color bar with numerical labels for the lower and upper color limits alongside the plots in your figures, as is done in figure 18.5 (plate 5). You should also label the color bar or write in the legend what scale the color bar indicates, so readers know how to interpret the color.

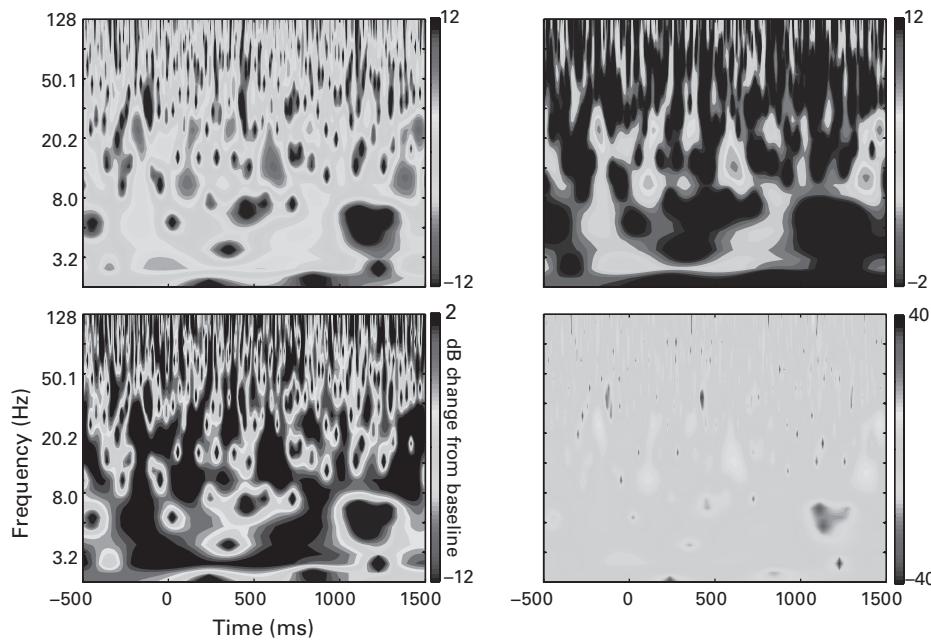


Figure 18.5 (plate 5)

Different color scaling applied to the same time-frequency power results. Symmetric color scaling should be preferred in most situations. Asymmetric color scaling can highlight or obscure different features of the data. Values in the color bars correspond to decibel change from a pretrial baseline of -500 to -200 ms. These data are taken from one trial and one electrode.

18.4 Percentage Change and Baseline Division

Percentage change is another option for baseline normalization. Similar to decibels, after percentage change, the results must be interpreted as changes in power relative to the power during the baseline period.

$$prctchange_{tf} = 100 \frac{activity_{tf} - \overline{baseline}_f}{\overline{baseline}_f} \quad (18.2)$$

As with equation 18.1, you can see that there is no t subscript in the average baseline activity. Thus, the percentage change value at each time-frequency point is computed relative to a baseline power that is frequency-band specific but not time-point specific.

A related transform to percentage change is dividing power during the task by power during the baseline time period. The unit is a ratio and is a linear transform of percentage

change (you can see this from equation 18.2). Dividing power by the baseline is also present in the decibel conversion (equation 18.1), and thus the baseline division ratio is a logarithmic transformation of the decibel. The relationships among decibels, percentage change, and baseline division ratio are discussed in section 18.6.

18.5 Z-Transform

The Z-transform works slightly differently from decibel or percentage change but still retains the same advantages that it corrects for $1/f$ power law scaling and transforms power data to be comparable across frequencies, electrodes, conditions, and subjects. With the Z-transform, power data are scaled to standard deviation units relative to the power data during the baseline period. The units are normal Z values and so can be easily interpreted and converted to p values (for example, $Z = 1.96$ corresponds to a two-tailed $p = 0.05$)

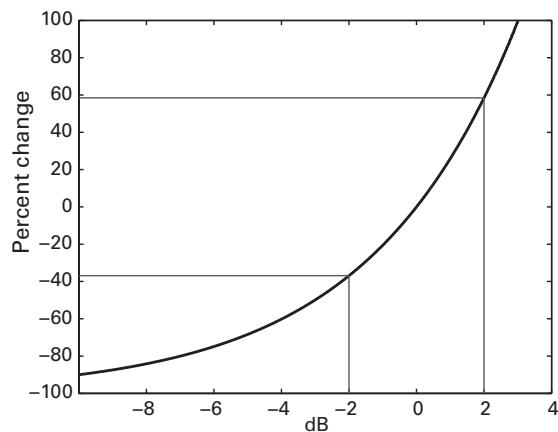
$$Z_{tf} = \frac{activity_{tf} - \overline{baseline}_f}{\sqrt{n^{-1} \sum_{i=1}^n (\overline{baseline}_{if} - \overline{baseline}_f)^2}} \quad (18.3)$$

in which n is the number of time points in the baseline period. The denominator in equation 18.3 is the formula for the standard deviation. The Z-transform differs from decibel and percentage change because the latter two methods are based only on the average baseline power, whereas the Z-transform is based both on the average baseline power and on the standard deviation of the baseline power over time. Because of this, estimates of stimulus-related power may be adversely affected by highly variable data in the baseline period. This may be an issue if you have noisy data or few trials. An example of this is shown below.

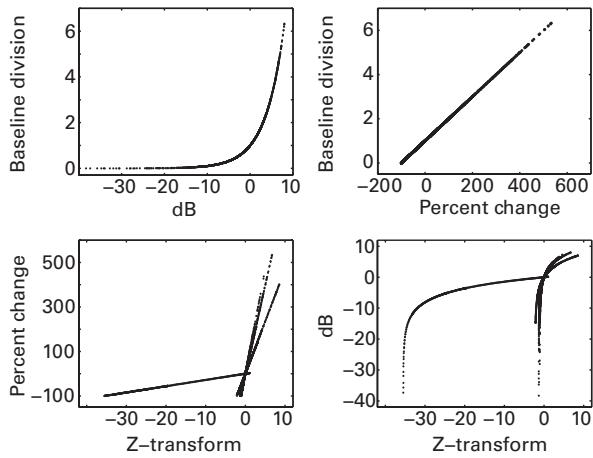
18.6 Not All Transforms Are Equal

Different baseline transformations should yield similar but not necessarily identical results. An important difference between decibel and percentage change, for example, is that decibels provide a logarithmic scale, whereas percentage change is a linear scale. At decibel and percentage change values near zero, both transforms will give highly similar results because at this small scale, decibels approach a linear function. As the values become further away from zero, however, the scales begin to differ. Decibel values become steeper at increasing negative values compared to percentage change values (see figure 18.6A). For example, a decibel range of -2 to $+2$ corresponds to a percentage change range of -36.9% to $+58.5\%$.

A) dB vs. percent change in simulated data



B) dB, percent change, and Z in real data

**Figure 18.6**

Comparisons of different time-frequency power baseline normalization methods. Panel A shows a comparison of percentage change and decibels, based on simulated data. Panel B shows the relationships among baseline division, percentage change, decibels, and Z-transform in real data. The Z-transformed data contain two frequencies with large negative outliers due to large variance during the baseline time period.

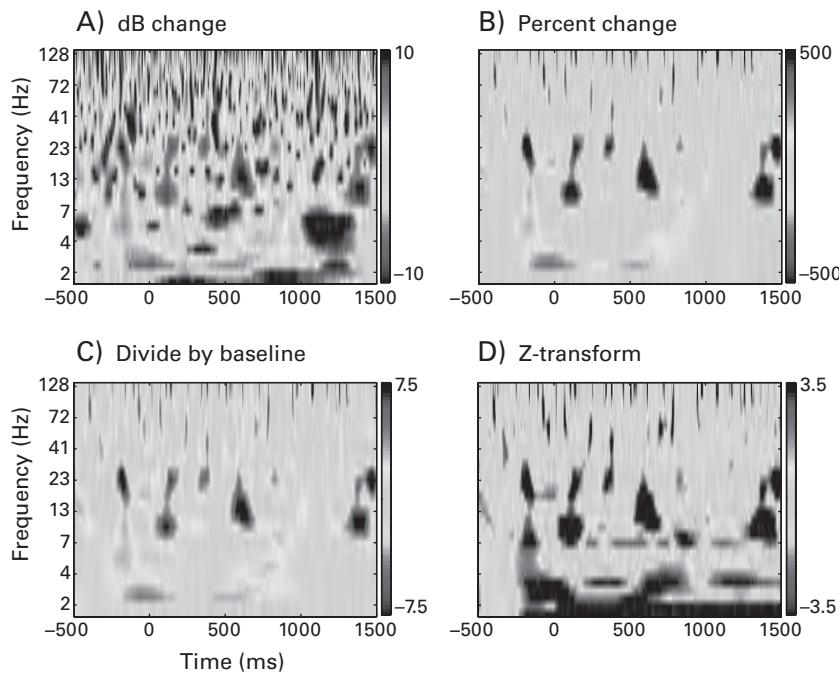


Figure 18.7 (plate 6)

Results from the same data and same analysis, but with different baseline normalization methods applied. As with previous figures in this chapter, the color scaling is larger here than what you would normally expect because data from one trial were used.

The Z-transform, however, has a more complex relationship to decibel and percentage change that cannot be analytically derived because it depends on the characteristics of the data (figure 18.6B). This is because the Z-transform is influenced by the standard deviation of power data during the baseline period.

Figure 18.7 (plate 6) shows time-frequency power plots of the same data using decibels, percentage change, baseline division, and Z-transform normalization. The main features of the time-frequency power dynamics are visible in all plots, but there are also some differences in what features are highlighted by the different baseline normalizations. The plots showing decibel change, percentage change, and baseline division could be made more visually comparable to each other by using asymmetric scales. However, most researchers use symmetric color scaling regardless of the type of baseline normalization, so it is useful to compare them this way. In figure 18.7D you can also see that the lowest two frequencies have very large negative values for the Z-transformation. This was due to a large standard

deviation during the baseline period, possibly from noise or outliers. Although the influence of noise is attenuated with more trials (these results were based on one trial), this highlights a potential disadvantage of the Z-transform, particularly if the data are noisy or if there are few trials.

18.7 Other Transforms

The methods discussed above are not the only baseline transformations that you could use to normalize the data. Other transforms could be based on empirical mode decomposition or on the Frobenius norm or other matrix norms. Although it is best to use a normalization approach that is commonly used, any procedure that is equally applied to all conditions, electrodes, and subjects, and that addresses the limitations of raw power discussed above, could be justifiable. For time-frequency power, linear baseline subtractions should be avoided because they do not address $1/f$ power-law scaling.

18.8 Mean versus Median

The typical strategy for aggregating activity over many trials is to compute the average, or the arithmetic mean. This is a ubiquitously used measure of central tendency in many branches of science. But is it the best measure of central tendency for time-frequency power? There are two related reasons why the mean across trials may be inappropriate in some situations. First, power values cannot be negative, which means that the distribution of power values across trials is likely to be positively skewed. Second, for the same reason, outliers and other nonrepresentative data are more likely to be larger than the mean, compared to smaller than the mean, and therefore will bias the mean toward larger values. This is further exacerbated by the fact that power is amplitude squared. Thus, somewhat large amplitude values become larger power values, and very large amplitude values, which may be outliers, become extremely large power values. Therefore, there is a danger that with a relatively low trial count and noisy data, outliers in one condition can skew the results to incorrectly suggest a condition difference (Handelsman 2002).

Computing the median is an acceptable alternative to the mean for some non-Gaussian or skewed distributions, particularly in situations in which outliers may influence the mean. The median is also easy to interpret because it is the number for which about half of the values are higher and about half are lower (“about half” because the median is not necessarily a number in the dataset when there is an even number of trials). If the data are clean, there are no outliers, and there is a sufficient number of trials, mean and median should

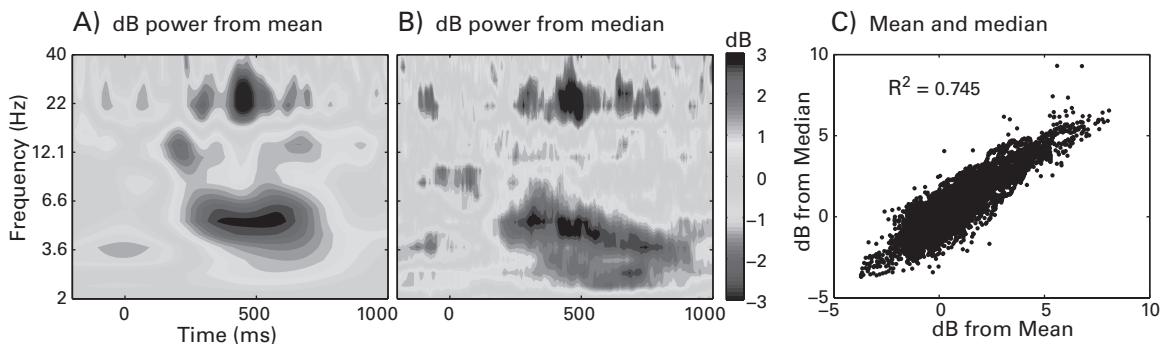


Figure 18.8 (plate 7)

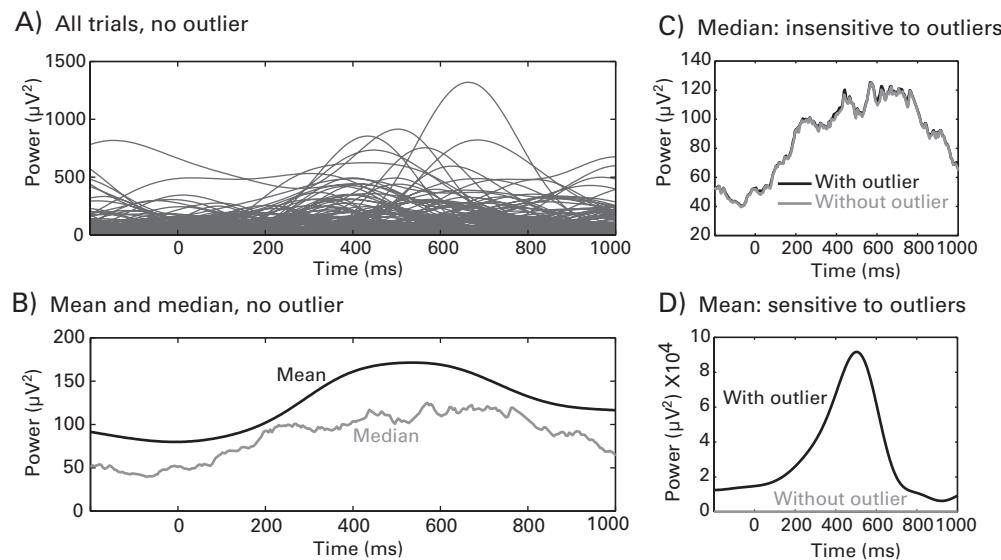
Time-frequency power plots of the same data using mean (panel A) or median (panel B) to combine data across trials. Panel C shows the correlation between the results using mean and using median.

give similar results. A comparison of the mean and median time-frequency power from one electrode can be seen in figure 18.8 (plate 7).

It is clear from figure 18.8 (plate 7) that the mean and median provide very similar results. Indeed, they have almost 75% of their variances in common, indicating that these two central tendency measures are largely overlapping although not entirely redundant. But visual inspection of the time-frequency plots suggests some differences. For example, a relative suppression in 8–10 Hz power from around −100 to +100 ms can be seen in the median but not the mean data. It is likely there were a few outlier trials with large power in this time-frequency region that pulled the mean up. Another difference is that the median plot shows a more sustained and slightly lower-frequency increase in theta power compared to the mean plot. Of course, these differences are based on visual qualitative inspection, and the extent to which results using the mean versus the median will differ (if at all) depends on the characteristics of the data.

The performance of the mean and median can be compared in the presence of an artificially created outlier. Thus, one additional trial was added to the sample dataset. This trial was created by multiplying one of the existing trials by a factor of one hundred. Ideally an outlier this large would be identified and removed during preprocessing. Nonetheless, this extreme situation provides a useful illustration of the potential influence of one outlier trial (out of 99 nonoutlier trials) and how the median is insensitive to outliers. Results are shown in figure 18.9.

Should you use the mean or the median as a measure of central tendency over trials? The mean is most commonly used in the cognitive electrophysiology literature and is likely to be

**Figure 18.9**

Panel A shows power from one frequency band and one electrode over all trials (each line is a trial). Panel B shows the mean and median power over all trials. Thereafter, one outlier trial was created by multiplying the time series data from one trial by the number 100. Panel C shows that the median power time course is unaffected by the presence of a single outlier trial. In contrast, panel D shows that the mean is very sensitive to the presence of an outlier. The line corresponding to the mean power without the outlier seems to be a flat gray line at the bottom of the plot; this is due to the large difference in scaling for the mean with the outlier included.

appropriate in most situations in which there are clean data and a sufficient number of trials. One situation in which the median might be preferred over the mean, or in addition to the mean, is when there are few trials and noisy data. This might be the case in special populations (for example, clinical populations or children) in which it is not possible to acquire many trials and/or when it is difficult to obtain the level of clean data that you might obtain with volunteer undergraduate students who have participated in experiments before. Or it might be the case for some experiment designs in which, even with patient and well-trained subjects, only a small number of trials can be obtained for a condition because, for example, the design requires that those conditions occur infrequently. In these situations, using the median will help protect the results from being spuriously driven by noise or nonrepresentative data.

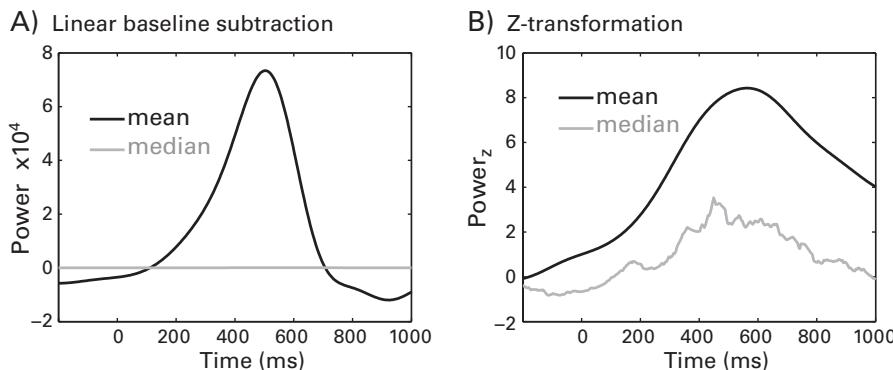


Figure 18.10

Comparison of mean and median power after linear baseline subtraction (left) and single-trial Z-transform (right), in the presence of an outlier trial.

18.9 Single-Trial Baseline Normalization

Typically, power data are averaged across all trials, and then a baseline normalization is applied. Performing baseline normalization on single trials prior to averaging helps minimize the influence of outlier trials but does not completely mitigate their impact, particularly for large outliers. Figure 18.10 shows results of two approaches for single-trial baseline normalization (Grandchamp and Delorme 2011): linear baseline subtraction (as is done on time-domain EEG traces to compute ERPs) and single-trial Z-transformation. The single-trial Z-transform is the same approach presented in section 18.5 but applied to each individual trial rather than to the trial average. You can see in figure 18.10A that the linear baseline subtraction fails to mitigate the effect of the one large outlier trial. The single-trial Z-transform (figure 18.10B) helps to minimize the impact of the outlier but still provides a somewhat positively skewed representation of the central tendency of power fluctuations. One potential limitation of the single-trial Z-transformation is that noisy single-trial baseline periods may lead to nonrepresentative estimates of power fluctuations, as could be seen in figure 18.7D (plate 6) for the low-frequency power.

18.10 The Choice of Baseline Time Window

Because baseline normalization means that activity is rescaled to be relative to the activity during the baseline period, the choice of baseline time period is an important one that may have implications for the results. Figure 18.11 (plate 8) illustrates this point. Consider first

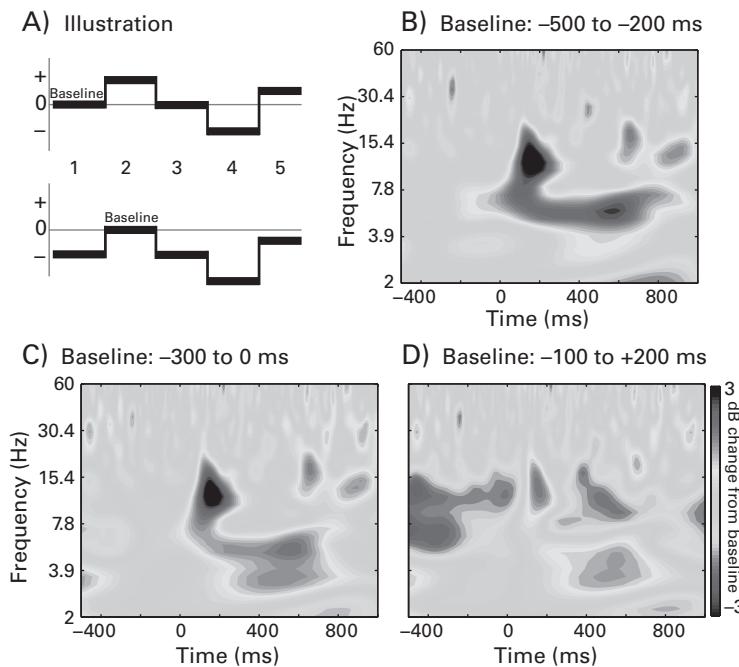


Figure 18.11 (plate 8)

The choice of baseline time period used for power normalization has implications for interpreting the results. Panel A illustrates conceptually that the same activity can be interpreted differently depending on what is used as a baseline period. Panels B–D illustrate this effect in real data. Panel D shows an extreme and unrealistic choice of baseline because it includes poststimulus time but demonstrates that activity present in the baseline period will appear as the inverse of that activity at other time points.

panel A, which shows that the same data could be interpreted differently depending on which time period is used for the baseline. When time period 1 is used as the baseline time period, the activity at time period 3 would be interpreted as a return to the baseline level of activity. However, when time period 2 is used as the baseline time period, activity in time period 3 would be interpreted as a relative suppression of activity. In these two situations the data themselves are not changed, only their interpretation with respect to data at other time periods.

The other three plots in figure 18.11 (plate 8) illustrate this point using real data. All plots show time-frequency power from the same data and the same wavelet analysis parameters but with different time windows used for baseline normalization (in this case, a decibel scale was used, but the illustration applies equally to any baseline normalization method).

Although in ERP analyses it is common to have the baseline end at time = 0, for time-frequency analyses this can be suboptimal because temporal smoothing from time-frequency decomposition can produce some temporal leakage of trial-related activity to the pretrial time period, particularly if the activity occurs shortly after the time = 0 event. Thus, a good baseline time period might be -500 to -200 ms, or -400 to -100 ms. This can be seen in figure 18.11 (plate 8) by comparing panels B and C: some of the theta-band (6-Hz) power extends a bit before time = 0 due to temporal smoothing, and thus, the time-frequency plot in panel C shows a less robust task-related increase in theta power compared to panel B.

Your choice of baseline time period also depends on what happened prior to time = 0. If, for example, a trial-start warning stimulus appeared 300 ms before time = 0, then you should avoid including this event in the baseline time period because its activity will become reflected in trial-related activity. This point is demonstrated in figure 18.11D, which shows power normalized to a baseline period of -100 to +200 ms. Obviously, this is a poor choice of baseline time period, but the point is that the trial-related *increase* in 14-Hz power becomes a pretrial *decrease* in relative power. An example of this phenomenon using real data is shown in figure 28.7. In that case, Granger prediction results during the baseline period changed in the interpretation of the poststimulus results after the data were transformed to percentage change from baseline. It is not necessary to understand how Granger prediction analyses work to notice that the pretrial increase in Granger prediction in figure 28.7C becomes reflected as a mild task-related decrease in Granger prediction in figure 28.7D.

Because any activity present in the baseline period will be transferred to the task-related activity, it is a good idea to use the pretrial period as a baseline even if you are not examining stimulus-locked activity. For example, if there are multiple stimuli and a response in each trial, and your hypotheses concern response-related activity, a preresponse baseline window may include stimulus-related activity.

A baseline period of a few hundreds of milliseconds prior to trial onset is the most common time period. Before learning about wavelet convolution or filtering, you might think that 300 ms is too short a baseline period for low-frequency activity because 300 ms might contain one or even less than one cycle for estimating power. However, remember that power is estimated at each time point and reflects a weighted sum of activity at surrounding time points (how strongly and for how much time depends on the wavelet or filter parameters). Thus, it is not the case that power in a 300-ms window reflects exclusively the activity within those 300 ms. Instead, power in those 300 ms reflects activity from a wider window of time in which the activity within those 300 ms is maximally weighted. For this reason even a single time point could be used as a frequency-band-specific baseline.

However, this is not recommended; a single time point could contain nonrepresentative data, and averaging over a few hundred milliseconds increases the signal-to-noise ratio of the baseline activity.

A pretrial baseline period is optimal. Whenever possible, try to design your experiment such that a pretrial baseline period of several hundred milliseconds can be used. However, this might not always be possible; some experiment designs may preclude using a pretrial baseline period. This might happen, for example, if you have rapidly presented trials and thus insufficient time for a clean baseline period.

There are three alternative options for baseline normalization that you could consider. One option is to use a separate rest period as the baseline. This could be a break of several tens of seconds every few minutes or after a block of trials. However, keep in mind that differences in cognitive mindset between task performance and a long resting baseline period might make the rest period an inappropriate baseline as a comparison for phasic task-related activity. It would be good to inform subjects of when the baseline period occurs; otherwise there may be too many artifacts in the data for it to be usable as a clean baseline period. A second option is to use a control condition as the baseline. The potential danger with this approach is that if the timing of events differs between the conditions, condition differences may become difficult to interpret. For example, imagine it takes 200 ms to process the stimulus in the control condition and 300 ms to process the stimulus in the active condition, but the strength of time-frequency power at 200 ms in the control condition is the same as power at 300 ms in the active condition. Using the control condition as a baseline in a time-point-by-time-point fashion will show a relative suppression and then enhancement of task-related activity, even if the strength of activity is the same at the periods of maximum stimulus processing. Finally, a third option for an alternative baseline period is to use the time series from the entire trial as the baseline. This approach is appropriate for detecting phasic changes in activity. However, sustained changes in power throughout the trial period may become difficult to detect.

One final consideration for baseline normalization is whether to use a condition-specific or a condition-average baseline. The former means that each condition is baseline-normalized independently of all other conditions, and the latter means that the conditions are first averaged together, then the baseline activity is computed, and then that average baseline activity is applied to each condition. There are two advantages of the condition-average baseline. First, the signal-to-noise ratio of the baseline power will increase because more trials will be included in the average. Second, any condition differences in baseline activity will be retained, which is advantageous if you have hypotheses about condition

differences in pretrial activity. Consider an experiment in which subjects anticipate and prepare for an upcoming trial in condition A but not in condition B. With a condition-averaged baseline, pretrial preparation-related activity that is present in condition A and not in condition B will show up as a relative increase in pretrial activity in condition A and a relative decrease in pretrial activity in condition B. However, with a condition-specific baseline, the pretrial differences will show up as a trial-related decrease in activity in condition A compared to condition B.

No matter what baseline period you select, it is important to interpret the results appropriately: increases and decreases in power should be interpreted as changes in power during the task relative to the power during the baseline time period, and zero change in activity means that there was the same amount of activity as there was in the baseline period.

18.11 Disadvantages of Baseline-Normalized Power

The main disadvantage of the baseline normalizations discussed in this chapter is that the activity becomes relative to the activity during the baseline period. This has two implications, as discussed above: “zero activity” means that there is no difference in activity compared to the baseline time period you selected; and activity that occurs during the baseline period becomes inversely reflected as task-related activity.

These disadvantages should be considered alongside the advantages of baseline normalization, namely that data from all conditions, electrodes, frequencies, and subjects are on the same scale and are thus visually and quantitatively comparable. Baseline normalization is almost always a good idea but is not always necessary.

Finally, note that the conversions discussed in this chapter can be applied to any data, not only time-frequency power. However, decibels cannot be used on data that can have zero or negative values because the logarithm of zero is undefined, and the logarithm of a negative number is not a real number. This will never occur with time-frequency power because raw power values cannot be negative, but this might occur with other data you want to baseline correct. If you have zero or negative values in your data, you can use percentage change or Z-transform, or rescale the data such that they are all positive real numbers and can therefore be converted to decibel.

18.12 Signal-to-Noise Estimates

Signal-to-noise ratio (SNR) is a useful although not commonly used measure of the quality of the data. To compute the true SNR, you need to be able to dissociate the signal from

the noise. In real EEG data this is not possible because signal and noise are mixed together. Thus, SNR can be estimated as the ratio of the mean signal to the standard deviation of the signal. This formula, even though simple, is open to interpretation in terms of how it is computed. For example, in the ERP literature, SNR is often estimated as the ratio of a component peak voltage to the temporal variance during the pretrial baseline period. This procedure can lead to SNRs of 10 or more when many trials are averaged and can be further enhanced by low-pass filtering (because low-pass filtering will decrease the temporal variance). For time-frequency data, SNR can be computed over time and frequency (SNR_{tf}).

$$SNR_{tf} = \frac{\mu_{tf}}{\sigma_{tf}} \quad (18.4)$$

where μ is the mean power over trials, σ is the standard deviation of the power over trials, and tf indicates one time-frequency point. This formula results in an estimate of SNR at each time-frequency point over trials. Alternatively, you can compute SNR_{base} , which is the power at each time-frequency point relative to the variance in the pretrial baseline period. This is similar to Z-transformation, which was used for baseline normalization in section 18.3.

Time-frequency maps of SNR_{tf} and SNR_{base} are shown in figure 18.12A,C (plate 9). Figure 18.12B,D (plate 9) shows that power and SNR provide unique information.

If you are accustomed to SNR of ERP components for which SNRs of 10 or higher can be reported, you might think the range of SNRs shown in figure 18.12 (plate 9) is unacceptably low. This is due to differences in how SNR is quantified. With ERPs, SNR is often defined as the peak of a trial-averaged component (e.g., the P3) compared to the temporal variance during the baseline period (e.g., -200 ms to trial onset). A comparable analysis on trial-averaged time-frequency power yields SNR estimates of up to 100, which is an order of magnitude larger than the range typically reported for SNR of ERP components. Likewise, computing the time course of the ERP SNR in the same way that SNR was computed in figure 18.12C,D yields average SNR values of around 0.1 for the ERP. This can be compared to average values of around 0.8 for time-frequency power. These findings are illustrated in the online Matlab code. Thus, when computing, evaluating, and comparing SNRs, it is important to make sure that you understand what method was used to create the SNR.

SNR_{tf} can be useful for two reasons. First, it can be used to help determine the quality of single-subject data when you are interpreting the robustness of a finding. That is, changes in power over time, across electrodes, or among conditions can be considered robust if the SNR is “big enough.” What is “big enough”? Unfortunately, this is difficult to answer because there are no guidelines for appropriate SNR ranges of EEG time-frequency power data. Such guidelines are further complicated by the fact that time-frequency power noise

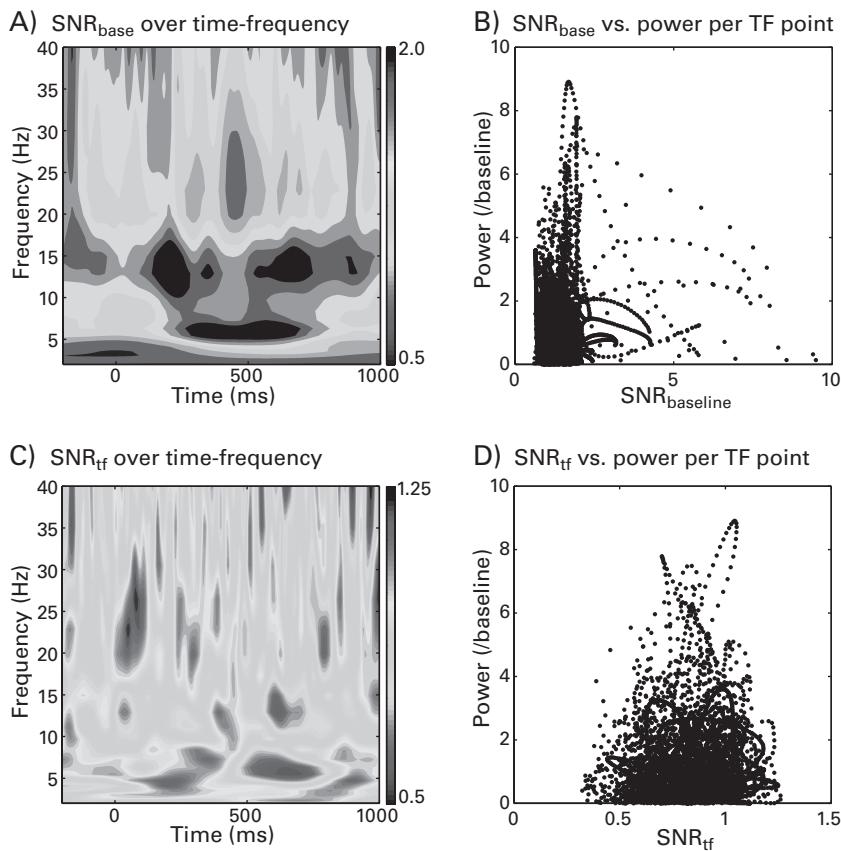


Figure 18.12 (plate 9)

Time-frequency SNR results, using the same time-frequency data that were shown in figure 18.11. Panels A and B show SNR_{base} (SNR relative to the baseline time period), and panels C and D show SNR_{tf} (SNR across trials at each time-frequency point).

is not normally distributed. For example, SNR estimates might be lower for median power compared to mean power in the presence of outliers, although the former might be a better measure of central tendency over trials in datasets with low trial count or high noise. A rigorous investigation of this issue with practical advice would be useful to the field.

The second reason why SNR_{tf} might be useful is in testing hypotheses about within-subject, cross-trial variability. For example, some developmental disorders such as autism and ADHD have been associated with increases in within-subject variability in behavioral

and EEG measures (Geurts et al. 2008; Milne 2011). SNR_{tf} might also be useful in an experiment in which greater cross-trial variability is expected in one condition compared to another.

18.13 Number of Trials and Power Estimates

How many trials should you have in each condition in order to obtain robust estimates of time-frequency power? There is no magic number of trials, and there is no guarantee that an arbitrary number of trials such as 50 will be a sufficient number for your experiment, the size of your effects, and your data characteristics. Relatively larger effects generally require fewer trials—the error-related frontal theta response, for example, seems robust with as few as 20 trials. How many trials you need also depends on the frequency of interest and on the analysis parameters you use. Low frequencies generally have higher SNR and therefore require fewer trials, and analysis parameters that involve more temporal or spectral smoothing generally have higher SNR.

One way to test whether you have enough trials is by estimating the reliability of the trial-averaged power using random subsets of trials. The idea is the following. If there is no noise in the data, the time course of frequency-band-specific power on one trial will correlate perfectly with the time course of frequency-band-specific power averaged over all trials. As noise increases, the correlation between an individual trial time course and the trial-average time course will decrease. To determine whether, for example, 10 trials provide sufficient reliability, the power time course averaged over 10 randomly selected trials can be correlated with the power time course averaged over all trials. This procedure can be repeated for any number of trials between one and the total number of trials. When this analysis is repeated over frequencies, the result is a frequency-by-trial-count map of correlation coefficients. This entire procedure can be repeated over several iterations in order to select different random subsets of trials. Spearman correlation coefficients should be used because the non-baseline-transformed power data are nonnormally distributed. The online Matlab code will show you how to perform this reliability analysis, and the results from electrodes FCz and P7 are shown in figure 18.13. (The online Matlab code here uses a fast implementation of computing correlations via least-squares fit; you will learn more about this in chapters 28 and 35.)

Keep in mind that these plots are from two electrodes and one subject. Thus, figure 18.13 should not be interpreted as an absolute reference for the required number of trials but, rather, as a tool for assessing whether you have enough trials in your experiment. The best

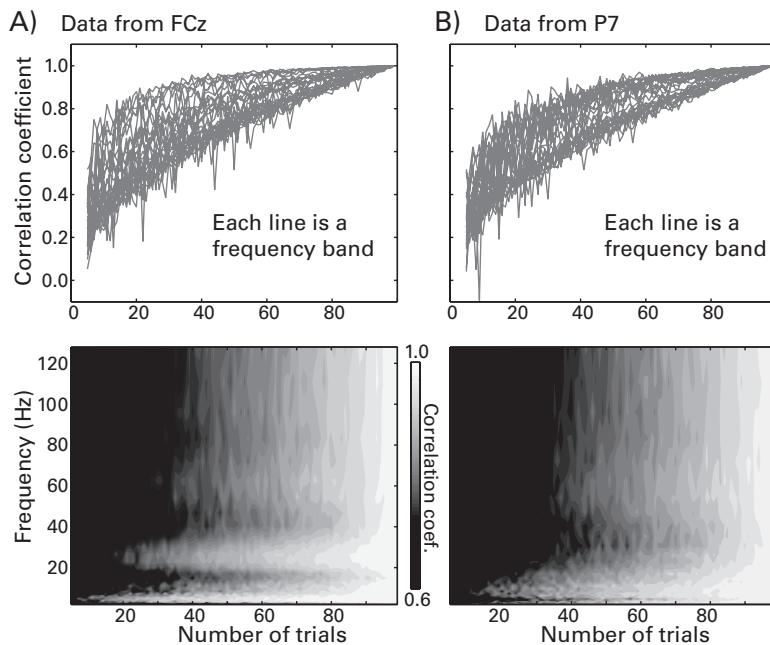


Figure 18.13

To examine how many trials is a reasonable minimum for each condition and frequency band, the average power time course of a randomly selected number of trials can be correlated with the average power time course of all trials. The plots show average correlation coefficients across ten iterations of randomly selected trials as a function of the number of trials. In an ideal situation with no noise and minimal cross-trial variability, the correlations would all be very close to 1.0. Panel A shows results from electrode FCz, and panel B shows results from electrode P7. The number of trials it takes to produce a correlation coefficient of 0.7 is a reasonable minimum number of trials per condition. Note that the appropriate number of trials in each dataset will depend, to some extent, on the frequency band, the time-frequency decomposition parameters, the electrode, and the task characteristics.

way to do this would be to pool all trials from all conditions and then examine the correlations at trial counts corresponding to the number of trials in the smallest condition. Based on other measures of reliability (such as Cronbach's α , for example), trial counts that correspond to correlations of around 0.7 should provide a reasonable number of trials per condition. For example, based on the results presented in figure 18.13, around 50 trials seems to be a reasonable minimum number of trials, but this could be as few as 20 trials for power in the theta band (the low-frequency elongated white correlation from electrode FCz is around 4–7 Hz).

18.14 Downsampling Results after Analyses

Recall from figure 2.4 that there is a distinction between precision and resolution. The high temporal resolution of EEG is necessary to apply time-frequency decomposition methods. However, after time-frequency decomposition, the temporal precision decreases because the estimate of power at each time point is a weighted combination of temporally surrounding time points. This means that activity from neighboring time points is autocorrelated and thus provides redundant information. In other words, after time-frequency decomposition, the temporal resolution is greater than the temporal precision. Thus, you can often downsample the results after the time-frequency decomposition with little or no loss of information. In many cases the results can be downsampled to 40 or 50 Hz (that is, one estimate of activity each 20 or 25 ms). Downsampling the results will decrease computation time, decrease file read/write time, and decrease disk space usage. Whether and how much you should downsample the results depends on the analyses you plan on performing. For example, if you expect transient effects, particularly in relatively high frequencies, downsampling might misrepresent or even eliminate some effects. This issue is discussed further, along with examples using real data, in section 27.5.

18.15 Describing This Analysis in Your Methods Section

Whether and how you compute baseline normalization is a small but critical detail of the Methods. Make sure you write which transform you used and the time period used for baseline activity. If no baseline normalization was applied, state whether any transformations were applied to the data, such as computing the logarithm of the power, prior to statistical analyses and plotting. Furthermore, you should justify the reason for applying or not applying baseline normalization. The main justifications for applying baseline normalization were listed in sections 18.1 and 18.2.

18.16 Exercises

1. Select three frequency bands and compute time-varying power at each electrode in these three bands, using either complex wavelet convolution or filter-Hilbert. Compute and store both the baseline-corrected power and the raw non-baseline-corrected power. You can choose which time period and baseline normalization method to use.
2. Select five time points and create topographical maps of power with and without baseline normalization at each selected time-frequency point. You should have time in columns and

with/without baseline normalization in rows. Use separate figures for each frequency. The color scaling should be the same for all plots over time within a frequency, but the color scaling should be different for with versus without baseline normalization and should also be different for each frequency.

3. Are there qualitative differences in the topographical distributions of power with compared to without baseline normalization? Are the differences more prominent in some frequency bands or at some time points? What might be causing these differences?

19 Intertrial Phase Clustering

In chapter 13 you learned that the analytic signal resulting from convolution between an EEG time series and a complex wavelet, or resulting from the filter-Hilbert method, can be conceptualized as a vector in a complex polar plane with a magnitude (length of the vector) and a phase angle (the angle in radians relative to the positive real axis). These phase angles provide information about the timing of frequency-band-specific activity and are the main focus of this chapter.

If you want to compute the consistency of time-domain EEG traces over trials, you simply average the activity at each time point across trials to form an event-related potential (ERP). Computing the consistency of time-frequency power over trials is the same as for ERPs: average the frequency-band-specific power at each time point across trials. However, computing the consistency of time-frequency phase values over trials is not so simple because phase values cannot be averaged together in the same way that voltage values or power values can be averaged. Learning how to average phase values and compute the consistency of phase values over trials is useful not only for examining the timing of frequency-band-specific activity but also forms the basis of several phase-based connectivity methods.

19.1 Why Phase Values Cannot Be Averaged

It would seem easy simply to average together the phase angle time series across trials, the way you would average single-trial EEG traces to form an ERP. Unfortunately, however, this is inappropriate. Phase angles are circular, which means that, for example, 0.05 and 6.2332 might seem far apart from each other when considered as numbers on a number line, but are actually very close to each other when considered as radians. Indeed, averaging these two numbers together gives a result close to π , which is in the opposite side of polar space from angles 0.05 and 6.2332 radians. Thus, it is not appropriate to average radian values together as if they were normal numbers (figure 19.1).

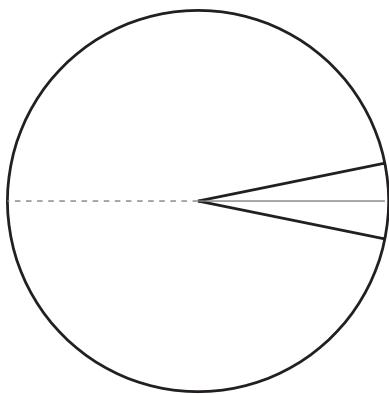


Figure 19.1

Two vectors (black lines) have similar angles; the average vector (solid gray line) reflects their proximity, but a vector formed by averaging their angles in radians (dashed gray line) does not reflect the two vectors. This is an illustration of why averaging phase angle radian values is inappropriate.

Remember that phase angles can be represented as vectors with unit length on a circle, and that Euler's formula (e^{ik}) provides a convenient mathematical description of those vectors in a complex plane. Thus, a population of phase angles can be represented as a population of vectors on a circle. For example, figure 19.2A,C shows phase angles from all trials at 200 ms and at 800 ms post-stimulus onset. Each vector (shown as a gray line) was formed by taking the phase angle from one trial and setting the vector length to 1 (rather than the vector length reflecting the similarity between the time series and the wavelet). The histograms on the bottom row show these same phase angle distributions in a different and perhaps more familiar way. You can see that at 200 ms, the vectors are clustered around one region of the circle, whereas at 800 ms they are more scattered. Another way to describe these distributions of phase angles is that the distribution of phase angles over trials is less uniform at 200 ms compared to the distribution at 800 ms. The extent to which these vectors are clustered (or nonuniformly distributed) is the measure of phase clustering across trials. This should make sense conceptually: if the timing of an oscillatory process is the same or similar at each repetition of a stimulus or other experiment event, their phase angles should take on similar values across trials.

19.2 Intertrial Phase Clustering

The measure of phase angle clustering illustrated conceptually in figure 19.2 is called intertrial phase clustering (ITPC). ITPC measures the extent to which a distribution of phase

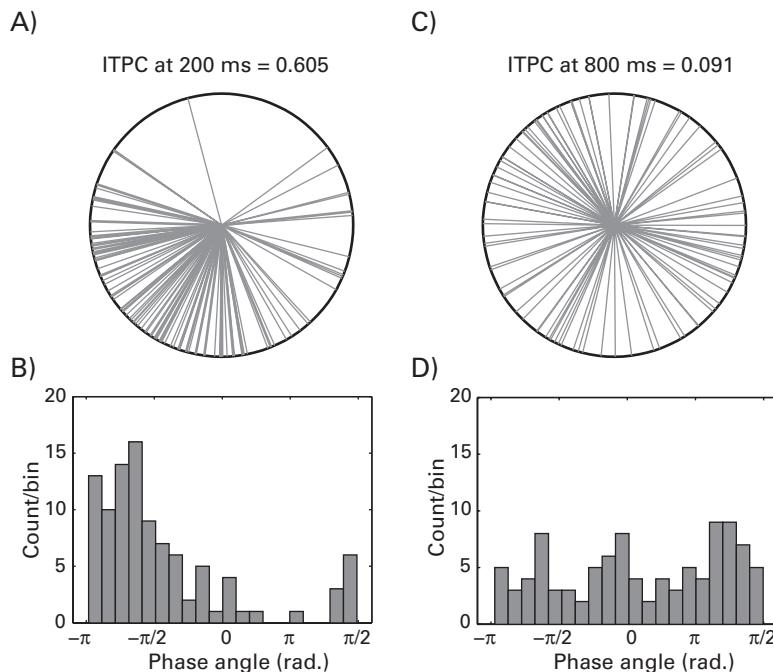
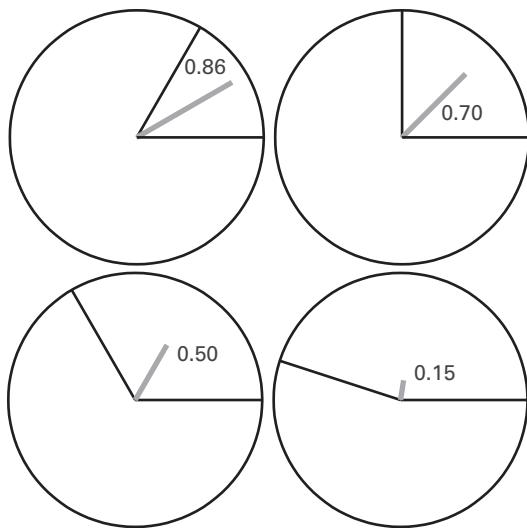


Figure 19.2

Phase-angle distributions at two time-frequency points. Each line in panels A and C corresponds to one trial, and the histograms in panels B and D show counts of trials per phase bin. It is clear that the phase angles are more clustered at 200 ms (panels A and B) compared to 800 ms (panels C and D).

angles at each time-frequency-electrode point across trials is nonuniformly distributed in polar space. In the literature you may see this measure referred to as “phase-locking value,” “phase-locking factor,” “phase resetting,” “phase coherence,” “intertrial phase coherence,” or “cross-trial” instead of “intertrial.” The term intertrial phase clustering is preferred here because it is a description of the analysis rather than an interpretation of the result (analysis terminology is further discussed in section 21.1).

How is the uniformity of the distribution of phase angles measured? This is done by computing the average vector (remember that the vectors are averaged, not their phase angles in radians) and then taking the length of that average vector. Figure 19.3 shows four examples of pairs of vectors, their average vector, and the length of the average vector. Note that the individual vectors always have unit length (length of 1.0), but the average vector has a length less than 1. The further apart the two vectors are from each other, the smaller the length of the average vector. Thus, the length of the average vector reflects the closeness of the two unit-length vectors. ITPC is this length.

**Figure 19.3**

Example pairs of unit-length vectors (black lines) and their averages (gray lines). The numbers inside each circle indicate the length of the average vector. This number is the ITPC for those two vectors.

ITPC is bound between zero and one, with zero indicating completely uniformly distributed phase angles and one indicating completely identical phase angles. This should make sense from figure 19.3. Consider two extreme situations: at one extreme, all phase angles are perfectly uniformly distributed, and the average vector has a length of 0.0; at the other extreme, all phase angles are identical, and the average vector has a length of 1.0. It is not possible for ITPC to be negative or greater than one.

At this point, you should have an intuitive understanding of ITPC. The next step is to quantify ITPC mathematically.

$$\text{ITPC}_{\text{tf}} = \left| n^{-1} \sum_{r=1}^n e^{ik_{tr}} \right| \quad (19.1)$$

The double vertical bars indicate the absolute value or, in this case, the length of the average vector. This is necessary because the result of the averaging is a complex number (because the vectors are described by complex numbers) that contains both the length and the angle of the average vector. n is the number of trials; n^{-1} is a convenient shorthand for $1/n$, and, combined with the summation operator, indicates an average. e^{ik} is from Euler's formula and provides the complex polar representation of a phase angle k on trial r at time-frequency

point tf . You can see that the M from equation 13.8 is not present in equation 19.1. This means that magnitude information is not taken into account when computing ITPC, and thus, the lengths of all vectors are implicitly set to 1.0. This formula can be expressed in Matlab code as:

```
abs (mean (exp (1i*k) ))
```

where k is a vector of phase angles at one time-frequency point over trials. Notice that this equation does not average phase angles in radians; it averages complex vectors whose angles are defined by the phase angles in radians. Make sure you write `abs (mean (. . .))` and not `mean (abs (. . .))`. The former is the length of the mean vector (this is what you want), whereas the latter is the mean of the individual vector lengths. Because each vector has a length of 1, the mean of all vector lengths is exactly 1.

The angle of the average vector is the average phase, or the “preferred” position of the angles at that time-frequency point over trials (in figure 19.2A, the preferred phase angle would point down and to the left). In practice, the average phase angle is not often used. However, it would be possible for two experimental conditions to elicit comparable ITPC strength but significant differences in the preferred phase angle. Statistical methods for testing differences in preferred phase angle over conditions or time points are discussed in chapter 34.

Equation 19.1 will generate an ITPC value for a single time-frequency-electrode point. This equation is then repeated over many time points and many frequencies to generate a time-frequency map of ITPC at each electrode. An example ITPC time course and a time-frequency plot are shown in figure 19.4.

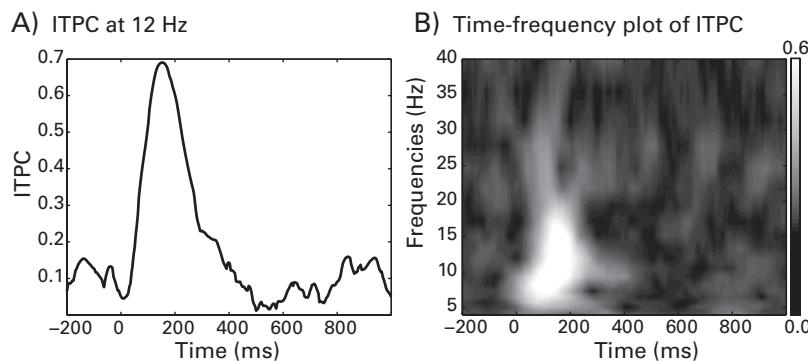


Figure 19.4

ITPC from electrode Pz at 12 Hz (panel A) and over time-frequency space (panel B).

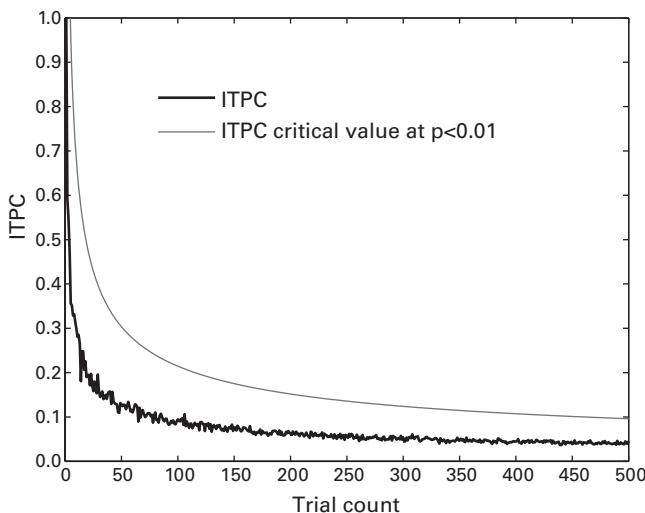


Figure 19.5

ITPC as a function of trial count for simulated random phase angles averaged over 50 simulations. Also shown is the critical ITPC value for each trial count, corresponding to a p -value threshold of 0.01.

19.3 Strength in Numbers

Trial count influences ITPC. Because ITPC cannot be below zero, noise and sampling errors are more likely to increase rather than decrease ITPC, particularly when there are few trials. This is illustrated in figure 19.5, which shows ITPC as a function of the number of data points (trials) using randomly generated phase angles. You can see that even with random numbers, ITPC values are fairly high with small sample sizes. The gray line shows the critical ITPC value corresponding to $p < 0.01$; any ITPC value above this line would be considered statistically significant (section 34.5 discusses how to compute and interpret this critical value). You can also see that even with randomly generated phase angles, ITPC values do not reach 0 after 500 trials.

Figure 19.6 shows ITPC as a function of trial count using real data (electrode FCz at 6 Hz). From a total of 99 trials, random subsets of trials were selected, ITPC was computed, and then the average ITPC between 100 ms and 450 ms was taken. Figure 19.6A shows ITPC as a function of the number of randomly selected trials, similar to the plot in figure 19.5. The value of ITPC stabilizes with around 20 trials, although it becomes statistically significant only after around 45 trials (using a p -value threshold of 0.01). Figure 19.6B shows the time course of ITPC at electrode FCz for all 99 trials compared to 20 randomly selected trials.

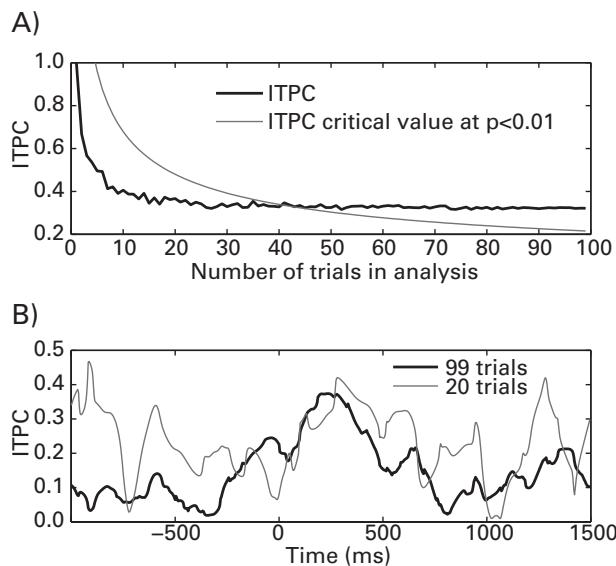


Figure 19.6

Relationship between ITPC and trial count in real data. Panel A shows the same analysis as was shown in figure 19.5 (ITPC as a function of trial count) but for real data instead of randomly generated phase angles. Each point is the average of 50 iterations of random trial selection. Panel B shows the time courses of ITPC for 20 randomly selected trials and all 99 trials. Phase angles were extracted via complex wavelet convolution with a 6-Hz wavelet.

Figure 19.6A shows the relationship between ITPC and trial count for one frequency band; figure 19.7 shows results of this analysis computed over a wider range of frequencies. You can see that the effect of trial count on ITPC depends on the frequency. For example, ITPC values for frequencies below 15 Hz seem to stabilize after around 20 trials, whereas ITPC values from higher frequencies seem to require more trials to stabilize. Keep in mind that figures 19.6 and 19.7 are based on data from one electrode, one subject, and one time window; the results of this test are likely to vary depending on the characteristics of your data. As with figure 18.13, you should interpret this figure not as an absolute reference for how many trials you need but rather as a way to examine in your own data whether you likely have enough trials for a stable estimate of ITPC within each condition. You can perform this analysis on your data by pooling all trials from all conditions together and examining whether the ITPC-by-trial-count function stabilizes with the number of trials corresponding to the number of trials in each condition.

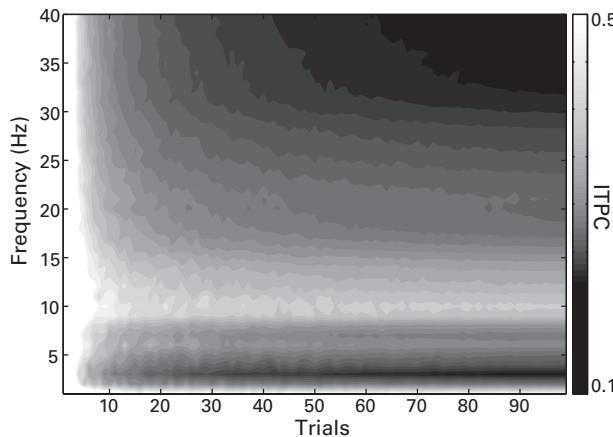


Figure 19.7

ITPC as a function of trial count and frequency. Note that where there is strong ITPC (here, around 10 Hz), trial count seems to have less influence on the strength of ITPC. Results are averaged over 50 iterations of randomly selected trials.

Low trial count can be more deleterious when ITPC is compared between conditions that differ in trial count. Consider figure 19.6A, and imagine comparing ITPC between a condition with 15 trials and a condition with 60 trials. The condition with 15 trials is likely to have larger ITPC simply because of lower trial count, regardless of what neurocognitive processes may have occurred during those two conditions. On the other hand, if your two conditions have trial counts of 60 and 70, there is less cause for concern about possible spurious results due to trial count differences.

19.4 Using ITPC When There Are Few Trials or Condition Differences in Trial Count

If possible, try to avoid this issue all together by designing the experiment such that each condition has at least 30 trials and all conditions have roughly the same number of trials. However, this is not always possible. If you have differences in trial count across conditions and would like to compare condition differences in ITPC, there are three strategies you can apply.

The first strategy is to match conditions for trial count within subject. The procedures, advantages, and limitations of selecting trials across conditions are discussed in section 7.4.

The second strategy is to apply a condition-specific baseline subtraction or percentage change transformation. The positive bias in ITPC due to trial count affects all time points

(indeed, this bias is even present with random data, as shown in figure 19.5), and thus, subtracting a baseline ITPC value will help minimize the bias. Because ITPC is not affected by $1/f$ power-law scaling, linear baseline subtraction is appropriate. Decibels, percentage change, and Z-transform are also acceptable baseline normalizations. However, this is not necessarily an ideal strategy. Low trial count not only increases ITPC, it also increases the susceptibility of ITPC to noise or nonrepresentative data. Thus, the baseline ITPC may be noisy, as can be seen in figure 19.6B.

The third strategy is to transform the ITPC to $ITPC_Z$, also known as Rayleigh's Z. The formula for $ITPC_Z$ follows:

$$ITPC_Z = n * ITPC^2 \quad (19.2)$$

in which n is the number of trials (see figure 19.8). Note that $ITPC_Z$ cannot be interpreted as a standard statistical Z value (that is, a value drawn from a distribution with mean of zero and variance of one) because it cannot be below zero, unless you somehow have a negative number of trials. However, a p -value can be computed from $ITPC_Z$ to determine statistical significance. This is discussed in section 34.5.

19.5 Effects of Temporal Jitter on ITPC and Power

Because ITPC reflects the precise timing of band-specific activity, it is sensitive to temporal jitter or temporal uncertainty of experiment events. This temporal jitter can arise, for

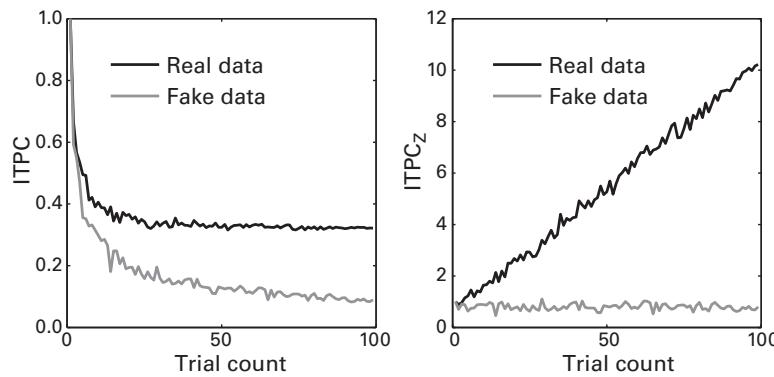


Figure 19.8

Comparison of ITPC and $ITPC_Z$ for randomly generated phase angles (gray lines) and real data (black lines). You can see that $ITPC_Z$ remains flat with increasing trial count for random data, whereas $ITPC_Z$ increases with trial count for real data, even though the "raw" ITPC values remain stable. This reflects increased reliability of $ITPC_Z$ with additional data.

example, from uncertainties in monitor display rates or time locking to events with uncertain onset times, such as EMG activations or pupil responses.

The negative impact of temporal jitter becomes worse at higher frequencies because the cycles become shorter. Imagine a temporal jitter of 10 ms; this jitter corresponds to only 2% of a cycle at 2 Hz but 40% of a cycle at 40 Hz. The negative effects of temporal jitter on ITPC can be demonstrated by introducing temporal jitter into real data. In the plots in figure 19.9 (plate 10), a random time lag between 4 and 40 ms was added to each trial, and then ITPC and power were computed. Temporal jitters had a strong negative impact on ITPC above 6 Hz and obliterated the increase in ITPC at 100 ms from 22 to 35 Hz. In contrast, power was largely unaffected by the temporal jitters. This result shows that phase values are more temporally precise measurements of frequency-band-specific activity compared to power values.

If you use a 60-Hz monitor and there is uncertainty as to when the stimuli are drawn on the monitor with respect to when the experiment marker is recorded in the EEG acquisition

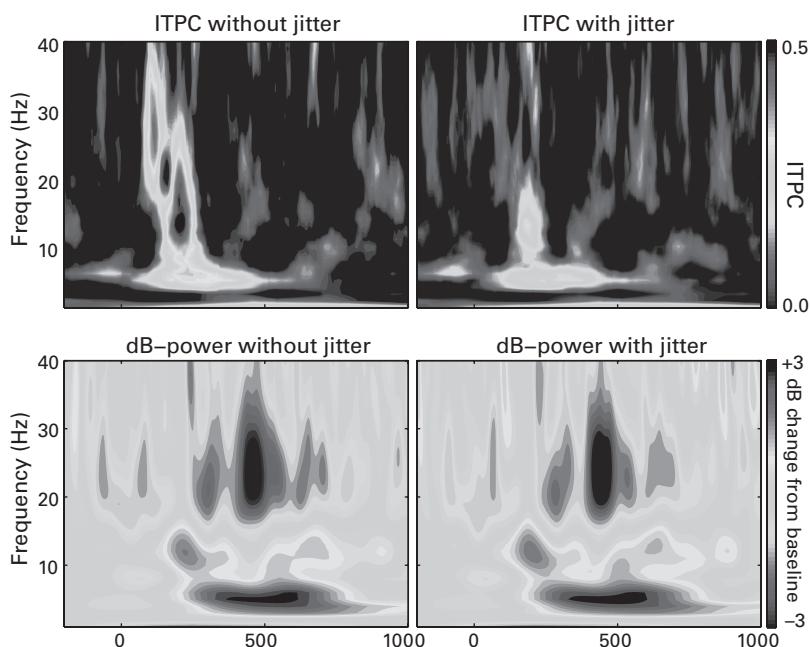


Figure 19.9 (plate 10)

Temporal jitters of less than 40 ms can have deleterious effects on ITPC (top row), particularly at frequencies above 10 Hz. In contrast, temporal jitters have little noticeable effects on power.

system, this could produce jitters up to 34 ms, which is the same range used in figure 19.9 (plate 10). Therefore, if you want to examine ITPC in the alpha band or higher, it would be a good idea to compare the timing of the experiment markers with the timing of the actual stimulus onset, with an oscilloscope if possible. If ITPC in relatively high frequencies is important for your analyses, you might also consider using a monitor with at least a 120-Hz refresh rate or using LEDs or auditory tones that are more temporally precise. The timing uncertainty related to monitor refresh rates can be further compounded if you have stimuli appear in different locations on the monitor. Monitor pixels are drawn in a loop that lasts the duration of the refresh rate; if stimulus locations differ randomly across trials and this is not taken into consideration in the analyses, temporal jitters of up to one refresh (e.g., 17 ms for a 60-Hz monitor) could be introduced into the data. If there are unavoidable and large temporal jitters in the experiment events, it might be best not to analyze ITPC except at low frequencies.

19.6 ITPC and Power

Several times in this book it is noted that phase angles are independent of power except in situations of very low power. The issue is that with decreasing power, phase becomes more difficult to estimate. Consider the extreme case, in which there is zero power at a particular frequency band: the band-specific activity is zero, there are no oscillations, and therefore phase is undefined. This extreme situation of zero power is unlikely to occur in real EEG data, partly because the brain generates broadband activity in addition to frequency-band-specific activity and partly because noise often increases broadband power. However, decreased power may nonetheless affect the signal-to-noise ratio of phase estimates, and so ITPC may in fact be affected by power. This point was illustrated using simulated data (Muthukumaraswamy and Singh 2011), suggesting that phase-based measures can be influenced by power when the signal-to-noise ratio is very low (less than -10 dB; in that simulation, power had little effect on phase-based measures for simulated data with higher signal-to-noise ratios).

Figure 19.10 shows the relationship between ITPC and power using real data. ITPC was computed over time for the original data and the data tapered with a 1-Hz sine wave with an amplitude fluctuation between 0 and 2. This caused the amplitude of the data to fluctuate between zero and double its original amplitude. This was done on the broadband signal and on the signal after bandpass filtering from 10 to 20 Hz. The broadband ITPC shows some differences between the unmodulated and modulated signals around times when the amplitude modulation was close to zero but seems intact otherwise, including when the amplitude was

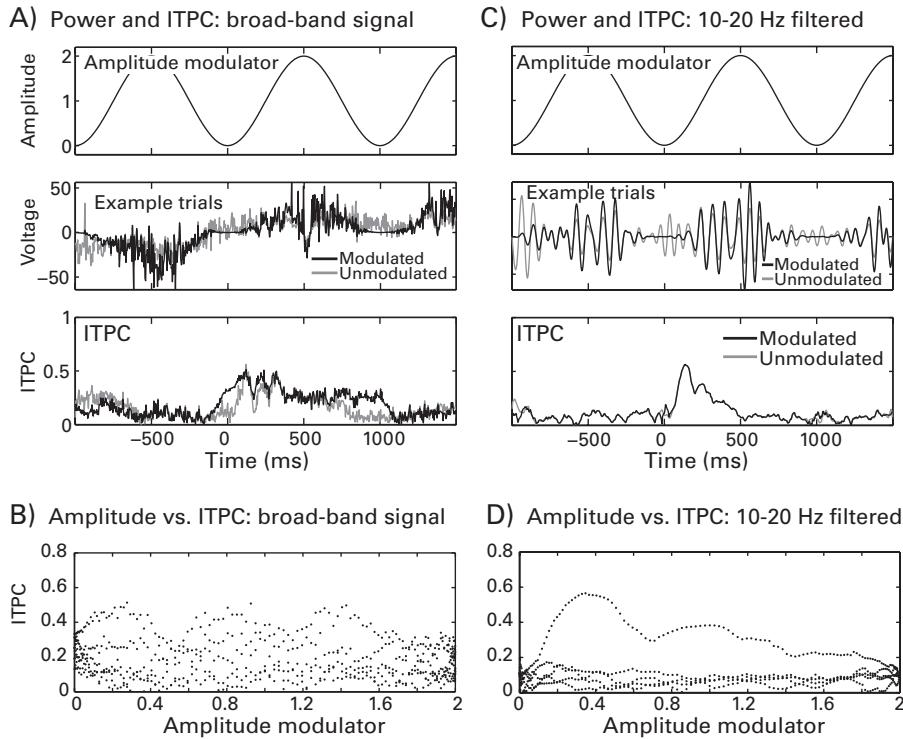


Figure 19.10

The relationship between power and ITPC. Real data were modulated by a 1-Hz sine wave (panels A and C, top row), resulting in significant power decreases and increases over time (panels A and C, middle row). ITPC based on broadband activity was somewhat affected by power being attenuated to zero, whereas ITPC based on band-limited activity was less affected by power (third row; the black and gray lines in panel C are mostly overlapping). Panels B and D show the relationship between ITPC and the modulating power signal.

doubled. Thus, extreme signal attenuation can have an adverse effect on ITPC, but signal enhancement had little effect in this example. Figure 19.10B shows that there was no clear relationship between the modulating signal amplitude and ITPC over time.

The bandpass-filtered data showed very similar ITPC time courses for the unmodulated and modulated activity, consistent with the idea that the phase information remains intact when the amplitude information is strongly attenuated. Although it may appear from figure 19.10D that there is a relationship between the modulating signal and ITPC (e.g., the increase in ITPC when the amplitude modulator is around 0.4), this is because a trough of the

1-Hz amplitude modulator coincided with a stimulus-related increase in ITPC. For example, in the online Matlab code, if you change the amplitude-modulating signal from a sine to a cosine, the relationship between modulating amplitude and ITPC will reverse because the phase of the modulating amplitude will shift.

The negative effect of very low power on ITPC was observed only in the broadband signal; the amplitude of the data had relatively little impact on the ITPC of bandpass-filtered data. This is because the time periods of zero power were relatively brief and because the phase values at each time point of a wavelet convolution could still be estimated from surrounding points (because stationarity is assumed during the wavelet period). Thus, longer periods of sustained zero (or near-zero) power might hinder estimation of frequency-band-specific phase values. The more important point of figure 19.10, however, is that for all nonzero points of the amplitude modulator period, power modulation had no effect on ITPC. You can try this yourself in the online Matlab code by increasing the magnitude of the power modulator envelope. It will have no effect except at power values of zero. Thus, changes in power within realistic ranges of EEG data do not necessarily lead to spurious inflations or deflations of ITPC except in the unusual case when power is exactly zero for an extended period of time (chapter 26 further illustrates this point).

In general, these findings suggest that power and ITPC are not necessarily coupled and that ITPC can be safely interpreted independently of power. Nonetheless, the potential relationship between power and ITPC should not be dismissed because of this one illustration. It is a good idea to examine the relationship between ITPC and power in your data before confidently interpreting ITPC and power in different ways. This could be done, for example, by showing qualitative differences between power and ITPC over time, frequency, electrodes, or conditions.

19.7 Weighted ITPC

Weighted ITPC (wITPC) is an extension of ITPC that provides a more direct link between phase angles and trial-varying behavior or experiment parameters (Cohen and Cavanagh 2011; Cohen and Voytek 2013). The wITPC addresses two limitations of interpreting ITPC with respect to cognitive processes. The first limitation is that ITPC can arise due to several task-related but not condition-specific factors, including stimulus-evoked responses and general orienting or attention responses. A second limitation of ITPC is that it precludes discovery of phase dynamics that are related to the task but are not clustered within the same range across trials. That is, if phase values are related to trial-varying behavior or experiment properties but are not related to stimulus onset, there may not be a significant ITPC.

Unlike ITPC, in which the vectors all have a magnitude of 1.0, with wITPC the magnitude of each vector is scaled according to a behavioral or experiment variable on that trial, such as reaction time, stimulus luminance, expected reward value, or pupil response. From here, calculation of wITPC is similar to ITPC: the length of the mean vector of the (weighted) phase angle distribution is computed:

$$\text{wITPC}_{\text{tf}} = \left| n^{-1} \sum_{r=1}^n b_r e^{ik_{rf}} \right| \quad (19.3)$$

The only difference between this equation and equation 19.1 is the addition of the trial-varying vector b , which is a variable that is unique to trial r . Thus, the lengths of the vectors are modulated by b rather than having unit length. This vector b might need to be scaled if it contains negative numbers because vectors should have positive length.

Perhaps you can already detect an issue with wITPC: because the average vector length scales with the magnitude of b , wITPC does not have an upper bound of 1. Instead, its magnitude depends on the scale of the data, which is arbitrary. For example, imagine that the b vector is the duration of a stimulus-preceding task-preparation time: wITPC will be larger if time is computed in milliseconds instead of seconds. This is an awkward property for a brain activity measure and hinders comparison of wITPC across different b variables and different subjects. You might initially think that a solution would be to scale the b elements so they have a maximum of 1 and a minimum of 0 or scale the mean vector length by the sum of the individual vector lengths. However, these transforms do not address a second issue with wITPC, which is that the mean vector length will be related not only to values of b but also to any nonuniform clustering of phase values. In fact, wITPC will have higher values during periods in which there is strong ITPC, even if those phase values are unrelated to the b vector. This is shown below and is also explained further in section 30.3 on phase-amplitude cross-frequency coupling.

The solution is to use nonparametric permutation testing. This resolves both the b -scaling issue and the phase-clustering issue. Nonparametric permutation testing has an additional advantage, which is that it transforms the wITPC to a value that is amenable to within- and across-subject statistical evaluation and comparison. Permutation testing is explained in detail in chapter 33 and is briefly described here. Permutation testing of wITPC involves comparing the observed wITPC value (that is, the value from evaluating equation 19.3) against a distribution of wITPC values that were computed under the null hypothesis of no relationship between phase and b . The null hypothesis distribution is created by randomly shuffling the trial mapping between b and phase angles (chapter 33 shows how this procedure produces an appropriate test of the null hypothesis). Once the distribution is created, wITPCz

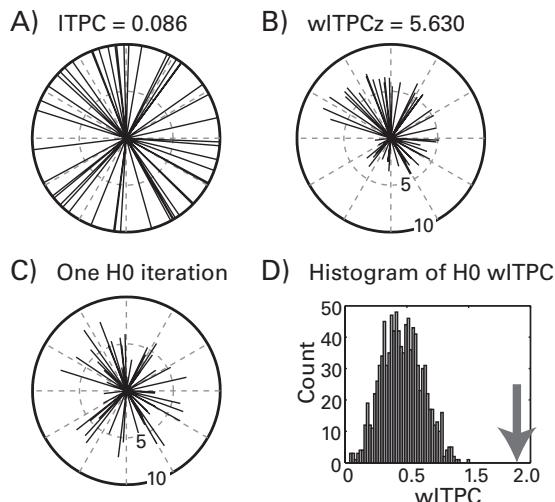


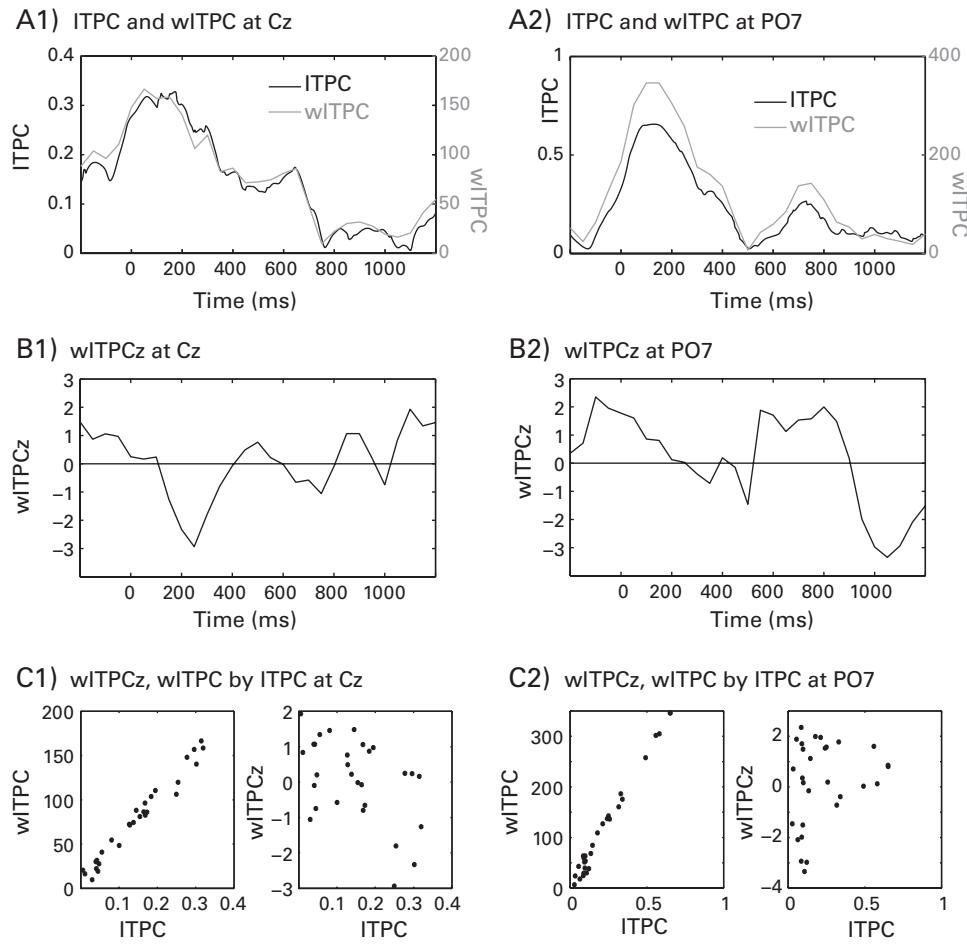
Figure 19.11

Illustration of the procedure to compute wITPCz from wITPC. Panels A and B show the dissociation between ITPC and wITPCz (each line corresponds to a trial at one time-frequency point). Note that when the lengths of the individual vectors are scaled, there is a clear relationship between the distribution of phase angles and the trial-varying behavior even though the distribution of the phase angles themselves is uniform. Panel C shows one iteration of a null hypothesis shuffling (H0), in which the mapping of the *b* vector and phase values was shuffled over trials. Panel D shows a histogram of null-hypothesis vector lengths, along with the observed vector length (large arrow) from the distribution shown in panel B.

can be computed as the normalized distance of the observed wITPC from the distribution of null hypothesis wITPC values. The normalized distance is computed by subtracting the mean of the null hypothesis distribution and then dividing by the standard deviation of that distribution. The wITPCz can then be interpreted in the same way that you would interpret a normal Z-score. The procedure is illustrated using simulated data in figure 19.11.

Figure 19.11 was generated from simulated data for illustration purposes. Figure 19.12 shows wITPC and wITPCz in real data using reaction times as the *b* vector. The phases were extracted via complex wavelet convolution with a 6-Hz wavelet. The importance of computing wITPCz instead of interpreting wITPC is evident in the similarity between wITPC and ITPC and their lack of similarity to wITPCz. The confound is that wITPC is influenced by the distribution of phase values, which is what ITPC measures.

These two analytic approaches—ITPC and wITPCz—are complementary and provide different kinds of information regarding neurocognitive processing. ITPC provides insights into the overall stimulus- or response-related timing of frequency-band-specific activity over

**Figure 19.12**

Comparison of wITPC and wITPCz with ITPC at electrodes Cz (left-hand plots) and PO7 (right-hand plots) for 6-Hz phase angles. The trial-varying modulating variable was reaction time. Note that Cz and PO7 have similar time courses of ITPC but very different time courses of wITPCz.

trials and could be driven by a number of cognitive factors, some of which might have little relevance to the purpose of the experiment (e.g., general task orienting, working memory access, attention). The $wITPC_z$ is specific to the behavior or experiment feature used as the b weighting vector and is not related to phase clustering per se.

19.8 Multimodal Phase Distributions

ITPC is designed for unimodal distributions—that is, phase distributions that have one peak. If there are multiple peaks, ITPC will be reduced (because it will reflect the average of two peaks) and will tend toward zero if the two peaks have opposite preferred phase angles. It is unknown how often bimodal (or multimodal) phase distributions occur in the brain or in the literature because they are rarely examined. There are methods for detecting and quantifying multimodal peaks in phase distributions. One method relies on comparing the shape of the phase angle distribution to a null-hypothesis distribution. This method was developed for measuring cross-frequency coupling but can also be applied to ITPC (counts per bin would be used instead of high-frequency power) (Tort et al. 2010). Other possibilities include using phase bifurcation to test for divergence of ITPC strength over two conditions (Busch, Dubois, and VanRullen 2009) and computing ITPC separately for an arbitrary number of sub-bins and then examining differences in preferred angle (Drewes and VanRullen 2011).

19.9 Spike-Field Coherence

If you have single-unit and local field potential activity simultaneously recorded, you can compute spike-field coherence using a similar method as for ITPC: for each frequency band, identify the phase angle of the field potential at which the spike occurs and generate a distribution of phase angles according to spike times.

19.10 Describing This Analysis in Your Methods Section

Although ITPC is a mathematically straightforward method, it is less commonly applied in the cognitive electrophysiology literature. Therefore, you should justify why you included ITPC in the analyses. Typically, ITPC is used to investigate the timing and consistency of the timing of frequency-band-specific activity because ITPC has more frequency-band specificity compared to the ERP (this is shown and further discussed in chapter 20). Including the details of this analysis in your Methods section involves simply including equation 19.1 and perhaps explaining that ITPC is bound by 0 and 1, with 0 indicating random phases and 1

indicating perfect phase clustering. However, be clear about the term you use for this analysis because in the literature you will find the same terms used for different analyses (issues of terminology are discussed in sections 21.1 and 37.10). The wITPCz is not a commonly used technique, so it should be explained in more detail, including how the distribution of wITPC values under the null hypothesis was generated.

19.11 Exercises

1. Pick three electrodes. Compute time-frequency plots of ITPC and decibel-corrected power for these electrodes, using either complex Morlet wavelet convolution or the filter-Hilbert method. Plot the results side by side for each electrode (power and ITPC in subplots; one figure for each electrode). Are the patterns of results from ITPC and power generally similar or generally different? Do the results look more similar at some electrodes and less similar at other electrodes?
2. For each of these three electrodes, compute wITPCz using reaction time as the trial-varying modulator. Perform this analysis for all time-frequency points to generate time-frequency maps of the relationship between phase and reaction time. Do the time-frequency maps of wITPCz look different from the time-frequency maps of ITPC? Do you see any striking patterns in the ITPCz results, and do the results differ across the different electrodes (don't worry about statistics, base your judgment on qualitative patterns)? How would you interpret the results if they were statistically significant?

20 Differences among Total, Phase-Locked, and Non-Phase-Locked Power and Intertrial Phase Consistency

The concepts of phase-locked and non-phase-locked activities were introduced in chapter 5 (e.g., figure 5.2). This chapter illustrates some of the methods used for separating phase-locked from non-phase-locked power in EEG data.

20.1 Total Power

This is the easiest to understand because “total power” is what you have already been learning about in chapters 12–18. Total power is what you get when you take the time-frequency decomposition (through any method) of each trial and then average together the time-frequency power from all trials. This is the time-frequency approach most commonly used in the cognitive electrophysiology literature. An example is shown in figure 20.1A (plate 11).

20.2 Non-Phase-Locked Power

Non-phase-locked power (also sometimes called induced power) is the time-frequency representation of the data after the phase-locked components of the EEG signal are removed.

Although there is general agreement about the definition of non-phase-locked power, there are several suggestions for how to compute it (David, Kilner, and Friston 2006; Donner et al. 2008; Klimesch et al. 1998; Martinez-Montes et al. 2008; Tallon-Baudry and Bertrand 1999; Truccolo et al. 2002). The approach used here is to subtract the ERP from each trial and then perform a time-frequency decomposition of the single trials as you would for total power. Note that this means that the ERP of the non-phase-locked activity will be—by definition—zero. For the same reason, the ITPC of non-phase-locked activity will be very close to zero. Thus, the time-frequency representation after subtraction of the ERP from each trial will show the dynamics that are task-related (if baseline normalization is applied) but do

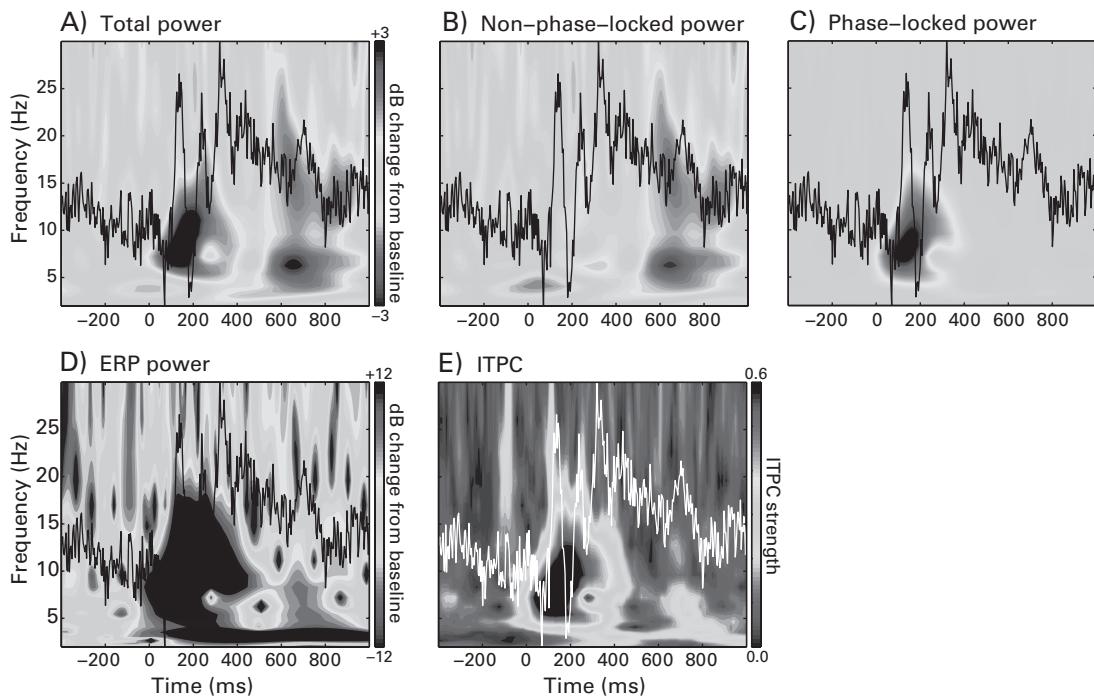


Figure 20.1 (plate 11)

Example results from different methods of computing time-frequency power (panels A–D) and ITPC (panel E) at electrode O1. The ERP is overlaid on the time-frequency plots to facilitate comparison between the ERP and the time-frequency features (the phase-locked ERP is shown in panel B rather than the non-phase-locked ERP because the latter is zero). The ERP is colored white in panel E for increased visibility in the grayscale version of this figure. The color scaling in panels B and C is the same as that in panel A.

not contribute to the ERP. An example of non-phase-locked power is shown in figure 20.1B (plate 11). This method produces results that, in this case, are very similar to the total time-frequency power, except for the dynamics around 0–200 ms in the theta-alpha range (figure 20.2A).

20.3 Phase-Locked Power

If total power comprises both phase-locked and non-phase-locked activity, then it is logical that subtracting the non-phase-locked power from the total power yields the phase-locked power. Phase-locked power is also sometimes called evoked power.

Any activity that is not phase-locked to the time = 0 event will have no representation in the phase-locked power. For example, it is unlikely that phase-locked power will contain activities in frequencies higher than around 20 Hz, particularly after around 200 ms after the time = 0 event, because it is unlikely (although not impossible) that high-frequency activity will maintain phase locking to an event after many cycles. This is in part because small temporal jitters will have large negative effects on phase clustering, particularly in higher frequencies figure 19.9 (plate 10).

An example of phase-locked power is shown in figure 20.1C (plate 11). Here, the subtraction (total power minus non-phase-locked power) was done after the total and the non-phase-locked power were separately converted to decibel scale. This is a sensible approach because the baseline activity from the non-phase-locked signal is nearly identical to the baseline activity from the total signal (the ERP in the baseline time period is close to zero and thus subtraction from single trials has little effect), and because subtracting non-baseline-corrected power will produce negative values, which cannot be decibel-transformed. Phase-locked power defined in this way is closely related to ITPC (figure 20.2B), but is different from the time-frequency representation of the ERP (figure 20.2C), which is discussed next.

20.4 ERP Time-Frequency Power

To obtain the time-frequency representation of the ERP, you compute first the ERP and then its time-frequency representation via any method (wavelet convolution, filter-Hilbert, etc.), rather than computing the time-frequency representation of each trial and averaging those results together.

The raw power values (that is, the values before baseline normalization) from an ERP time-frequency power analysis are likely to be at least an order of magnitude smaller than the raw values from the total power or the non-phase-locked power analyses. This is because the voltage values of the ERP are generally at least an order of magnitude smaller than the voltage values of single trials (figure 9.1B). On the other hand, the baseline-normalized ERP-derived power will generally have larger decibel (or percentage) change compared to total power because the ERP baseline activity is likely to contain little variability due to time-domain averaging.

The baseline period power from the total or non-phase-locked analysis should not be used to normalize the ERP time-frequency power, in part because of the difference in signal magnitude as discussed above, and in part because then the baseline would come from a different part of the signal than what is used for the ERP time-frequency power (in fact, it is the part of the signal that was removed from the data). Therefore, it is appropriate to use

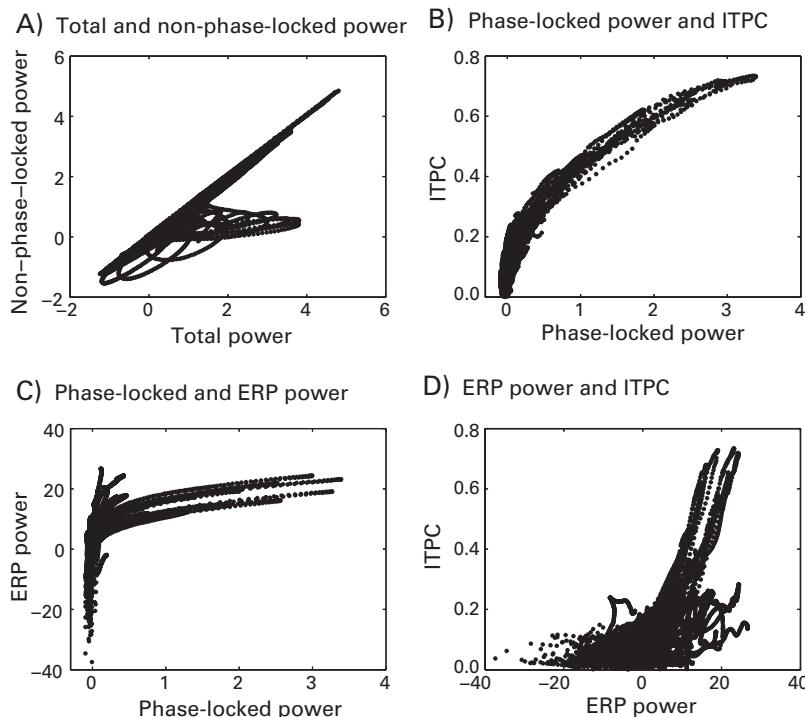


Figure 20.2

Direct comparison of results for the different methods of computing time-frequency power discussed in this chapter. Each dot corresponds to a single time-frequency point. The scale for all axes is dB change from baseline, except for ITPC in panels B and D.

the ERP baseline to normalize the ERP time-frequency power. Although it may initially seem awkward to have dB values in the range of 10–15 (or higher, if the ERPs have large magnitude and the baseline is very flat), this is an accurate depiction of the spectral representation of the ERP signal.

An example of ERP time-frequency power is shown in figure 20.1D (plate 11), and a comparison between ERP time-frequency power and ITPC and phase-locked power can be seen in figure 20.2C–D.

20.5 Intertrial Phase Clustering

This was the topic of chapter 19. ITPC measures the clustering in polar space of phase angles over trials at each time-frequency point. An example is shown in figure 20.1F (plate 11).

20.6 When to Use What Approach

Unless you have a reason otherwise, it is best to use total power. This is the most commonly used approach; it does not involve altering data by removing part of its signal; and its interpretation does not rely on assumptions regarding the neurophysiological events that may produce phase-locked versus non-phase-locked activity (see section 5.2 for further discussion of this point).

ERP time-frequency power does not seem to have many advantages. As specified by the Fourier theorem, there is no information available in the ERP time-frequency power that is not present in the ERP, and the ERP has higher temporal precision. Furthermore, the ERP may be simpler to interpret because there is likely to be a large literature to which you can link the ERP findings. ERPs are unlikely to have a “clean” time-frequency representation with well-defined frequency components due to mixing in the time domain during averaging and due to the violation of stationarity during the ERP, which decreases the interpretability of the spectral reconstruction (see section 11.8). Consider, for example, that the frequency structure at around 200 ms in figure 20.1D extends from around 2 Hz to almost 20 Hz in the ERP time-frequency power, whereas the total power in figure 20.1A has a frequency structure that is more contained within the alpha band. ERP time-frequency power may be most useful for attempts to dissociate components of the ERP by their frequency representation rather than by their latency and polarity.

There is an interesting observation in the comparison of the ERP and its time-frequency representation in figure 20.1D with panels A–C in that figure. The slow potential in the ERP, which is visible as a slow fluctuation from around 0 ms until around 800 ms, is captured by the ERP time-frequency response as an increase in delta-band (<4 Hz) power, but is not readily apparent in the time-frequency power plots in panels A–C. It is possible that this ERP fluctuation is generated by a mild ITPC (panel E), or that it may be generated by mechanisms that are not well captured by time-frequency decomposition methods, for example an amplitude asymmetry in oscillations (Mazaheri and Jensen 2008).

Non-phase-locked power is useful in two situations. The first is to examine the relationship between the information available in time-frequency power compared to the ERP. For example, comparing panels A and B in figure 20.1 (plate 11) shows that the increase in theta power around 200 ms reflects a phase-locked response, whereas the later theta power increase reflects a non-phase-locked response. Thus, a time-frequency feature that is present in the non-phase-locked time-frequency power suggests that that finding could not be obtained by analyzing the ERP alone. Interpreting differences in results between phase-locked and the total power in terms of neurophysiological mechanisms relies on the assumption that the

neurophysiological processes that generate the ERP are distinct from the neurophysiological processes that generate non-phase-locked activity. Some authors make this assumption (e.g., Pfurtscheller and Lopes da Silva 1999; Tallon-Baudry and Bertrand 1999), although other simulations suggest that this issue is more complex (David, Kilner, and Friston 2006). Compelling neurophysiological evidence for a clear dissociation between biological processes underlying phase-locked versus non-phase-locked activity would be helpful for interpreting the differences between phase-locked and non-phase-locked activity.

The second situation in which subtracting the ERP may be useful is to help make the single-trial EEG data more likely to remain stationary for some analyses such as autoregression modeling in Granger prediction. As discussed in chapter 28, this strategy is sometimes successful but is not guaranteed to provide stationarity in single-trial EEG data (e.g., figure 28.7).

ITPC is best to use when there are hypotheses regarding the timing of band-specific activity over trials with respect to an experiment event such as stimulus onset or response. ITPC is advantageous over the ERP because ERPs often have relatively little frequency-band specificity. On the other hand, the temporal precision of ITPC is lower than that of ERPs, as discussed in chapter 2.

20.7 Exercise

Pick two frequencies and compute total and non-phase-locked power from each electrode over time at these two frequencies. Pick two time windows, one early and one late, of several hundreds of milliseconds each (e.g., 100–300 ms and 500–800 ms) and show topographical maps of total power, non-phase-locked power, and phase-locked power from the average of all time points within these windows. Are there striking topographical differences among these results? If so, are the differences bigger or smaller in the early or the late time window? Why might this be the case?

21 Interpretations and Limitations of Time-Frequency Power and ITPC Analyses

21.1 Terminology

Unfortunately, there are different terms used in the literature for the same analyses. This is unfortunate because it impedes quick and efficient comparison of findings across studies and across research groups. The terms preferred in this book are those that are brief descriptions of the mathematical analysis that was performed rather than terms that are ambiguous about the analysis performed or terms that are interpretations of what neurophysiological events may have occurred. Below are a few specific examples, and section 37.10 continues this discussion.

The term “power,” for example, indicates the squared length of the complex vector and is an estimate of the amount of energy in that frequency band at that point in time. Other terms such as synchronization, event-related synchronization/desynchronization, time-frequency representation (TFR), and event-related spectral perturbation can be ambiguous and may lead to confusion. For example, the term synchronization (and event-related synchronization/desynchronization) can be confused with synchronization between electrodes or between frequency bands. Further, using the term synchronization to refer to power is an interpretation about an underlying neurobiological mechanism rather than a succinct description of the analysis (more on this term below in section 21.3). It is therefore a suboptimal term. Event-related spectral perturbation implies a change in the spectral characteristics due to an experiment event, but this could refer to changes in power, phase, connectivity, cross-frequency coupling, frequency-band-specific network dynamics, band-specific topographical distribution, and other parameters. Time-frequency representation (TFR) could refer to power or phase (in fact, a complete TFR requires both power and phase from as many frequencies as there are data points, although TFR generally refers only to power and generally uses many fewer frequencies than data points).

Another example of a term preferred here is intertrial phase clustering (ITPC). The terms phase-locking value or factor, or intertrial coherence, are sometimes used, but these terms sometimes refer to ITPC and sometimes refer to phase-based connectivity. Furthermore, the term is an interpretation of a presumed mechanism—oscillations become “locked” to an external event or to the phase values of another electrode—but the neurophysiological mechanisms underlying phase locking are less clear. The term intertrial coherence is also ambiguous because the term coherence often implies that two systems are operating in temporal simultaneity, which is inconsistent with the mathematical formulation of phase clustering per time-frequency point across trials. Thus, ITPC is a succinct description of the analysis, which is that phase values become clustered over trials.

21.2 When to Use What Time-Frequency Decomposition Method

There are few qualitative differences among various methods for performing time-frequency decomposition, and certainly not among the most commonly used methods (complex wavelet convolution, filter-Hilbert, and short-time FFT). With few exceptions the method you use is largely a matter of the method you prefer and feel most comfortable with. Below are some (mostly minor) advantages and disadvantages that might lead to preferences for one method over another for a specific application.

Complex Morlet wavelet convolution requires no Matlab toolboxes and therefore can easily be run on non-Matlab platforms such as Octave or Python. Further, because of the Gaussian shape of the frequency response of Morlet wavelets, wavelet convolution tends to produce smooth-looking and therefore easily visually interpretable time-frequency plots. Wavelet convolution is easily adaptable to balancing the trade-off between temporal and frequency precision as a function of frequency, and provides power and phase angle time series information. Finally, wavelet convolution is fast because it can be broken down into individual components that are run only when necessary (e.g., separately computing the FFT of the data and the FFT of the wavelets).

The filter-Hilbert method requires the Matlab signal-processing toolbox unless you write your own or can obtain a suitable filter kernel construction routine. A zero-phase-shift filter function is also necessary, but this can be written based on a causal filter function (by refiltering the time-flipped filtered signal and then time-flipping again). The Hilbert transform can easily be written in Matlab (it is included in the online Matlab code for chapter 14). The filter-Hilbert method also provides power and phase-angle time series information. Compared to wavelet convolution, bandpass filtering allows better control over the filter characteristics because the width of the frequency domain plateau and transition zones can be specified. This increased flexibility can be advantageous, or it can be a nuisance or even

lead to poor filter construction (as shown in figure 14.7 and discussed in section 14.6) and possible artifacts in the time-domain reconstruction. For this reason the filter-Hilbert method may be preferred for broadband analyses (e.g., 30–50 Hz in one time series) compared to wavelet convolution.

The short-time FFT produces power estimates that can be very similar to those produced by wavelet convolution or the filter-Hilbert method. The short-time FFT can be adapted to have the time-frequency precision trade-off change as a function of frequency (this was discussed but not explicitly shown in chapter 15). The phase values from the short-time FFT require a different interpretation compared to phase-angle time-series values, but this does not prevent their use in many phase-based analyses such as ITPC. The only situation in which the short-time FFT should not be preferred is for analyses in which power or phase values should be estimated with the same temporal resolution as the sampling rate (this is the case, for example, with cross-frequency coupling). In these cases, wavelet convolution or filter-Hilbert should be preferred.

The multitaper approach requires the Matlab signal-processing toolbox. It is the slowest and most computationally intensive of the time-frequency decomposition methods surveyed here because it involves applying several tapers to the data and computing the FFT for each tapered time series. With fast computers or distributed computing solutions, however, this should not be a limiting factor. The multitaper method is not necessary for capturing high-frequency power but it should enhance the signal-to-noise ratio of high-frequency power, particularly if there are few trials or noisy data. The implementation of the multitaper method in the fieldtrip toolbox offers the most flexibility and sensitivity for highlighting high-frequency activity.

It is best to use one of the four approaches discussed above because they are widely used and generally accepted time-frequency decomposition methods. Other methods (e.g., those discussed in chapter 17) should be used only if the more standard methods are less appropriate for your particular application. This is not because those methods are necessarily worse, but simply because many scientists will be less able to critically evaluate or replicate your analysis approach. When alternative methods are used instead of the more commonly used methods, they should be justified (for example, empirical mode decomposition is more appropriate than wavelet convolution if the hypothesis involves rapid changes in instantaneous frequency).

21.3 Interpreting Time-Frequency Power

It is likely the case that the neurobiological events that contribute most to an increase in time-frequency power include synchronization at the local neural level. Imagine that

10,000 neurons each oscillate at 10 Hz but that they each oscillate independently of each other, meaning that there are no consistent phase relationships among the neurons. At the mesoscopic level their electrical fields would cancel and would have very low power when measured from the scalp. On the other hand, synchronization among these neurons would lead to synchronous field potential oscillations, which would produce a field potential powerful enough to be detected by scalp electrodes. This is probably why the term synchronization is occasionally used to refer to power. Clearly, local neuronal synchronization is required to have a measurable EEG signal. On the other hand, it is also possible that an increase in power does not reflect a change in the strength of local synchronization but, rather, a change in the number of neurons that are synchronized. Indeed, even a population-level decrease in neural synchronization could produce an increased field potential response if the number of neurons that are recruited increases.

Power increases and decreases are frequency band specific, and different frequency bands seem to have different neurobiological mechanisms (Neuper and Pfurtscheller 2001). Gamma, for example, is often interpreted as reflecting spatially local processing, whereas lower frequency bands such as delta and theta are generally interpreted as reflecting coordination of larger-scale networks (von Stein and Sarnthein 2000). These should be taken as useful guidelines but not restrictive interpretations. For example, long-range gamma-band synchronization has been reported in humans and nonhuman primates (Nicol et al. 2012; Pesaran, Nelson, and Andersen 2008; Siegel et al. 2008). Alpha is thought to correlate negatively with cortical activation, suggesting that alpha reflects active and selective inhibition (Jensen, Bonnefond, and VanRullen 2012; Klimesch, Sauseng, and Hanslmayr 2007). Beta-band activity over motor areas is linked with motor responses (Lattari et al. 2010; Neuper, Wortz, and Pfurtscheller 2006). Theta-band activity over prefrontal regions has been implicated in working memory and top-down cognitive control processes (Cavanagh, Zambrano-Vazquez, and Allen 2012; Cohen and van Gaal 2012; Sauseng et al. 2010). This is merely a fraction of the extant literature linking specific brain rhythms in specific brain regions to cognition. Reviewing this entire (and rapidly growing) literature could generate an entire book on its own.

21.4 Interpreting Time-Frequency Intertrial Phase Clustering

ITPC is, mathematically, the clustering of the timing of band-specific activity over trials. Because it is mostly independent of power, phase-based results deserve an interpretation that is separate from the interpretation of power. Because phase reflects the timing of population-level activity, it can be conceptualized as a “functional configuration” or a “functional state.” That is, at each instance of an event that elicits ITPC, the neural networks contributing to

the ITPC return to the same or similar functional configuration, as measured by phase. This functional configuration could be used to maximize information processing, or it could be used to facilitate interregional coupling. These functional configurations in turn might be driven by consistency in the timing of afferent signals from lower areas that reset the ongoing oscillations to the appropriate functional configuration.

21.5 Limitations of Time-Frequency Power and Intertrial Phase Clustering

There are at least six limitations of time-frequency power and ITPC analyses. Some of these limitations generally concern noninvasive brain imaging and therefore also apply to ERPs and fMRI.

1. *The potential influences of analysis parameter selection on the interpretations of the results* The main analysis parameters concern the balance between temporal and frequency precisions, which affect how the data are “stretched” over time or over frequency (figure 13.13). Other parameters include the number of wavelets/shape of the band-pass filter, selection of baseline time period and baseline normalization method, and the number and range of frequencies. It is important to keep parameter settings constant across all conditions, electrodes, and subjects; thus, any biases introduced by parameter settings will affect all data equally. The influence of processing and analysis parameters on final results is an underdiscussed topic in cognitive electrophysiology. For example, it has been reported that the many preprocessing and analysis options in fMRI can produce different results (Carp 2012).

2. *The decreased temporal precision of EEG resulting from time-frequency decomposition* As discussed in chapters 2 and 3, quantifying oscillatory signals requires measuring several cycles (in the same way that it is not possible to measure your resting heart rate by holding your finger against an artery for 500 ms), and thus, temporal precision must be sacrificed in order to resolve frequency-band-specific activity. Although the decrease in temporal precision can be mitigated through appropriate analysis parameter selection, time-frequency methods will generally always have a lower temporal precision compared to ERPs (unless the ERPs are stringently low-pass filtered). However, many ERP components, particularly the “later” ERP components often associated with higher-level cognition, do not require temporal precision greater than 40–100 Hz (figure 2.3). Thus, the temporal precision of time-frequency results matches the temporal precision of some ERP components. If your hypotheses require temporal dissociations of a few tens of milliseconds or less, it might be better to test those hypotheses with ERPs; otherwise, standard time-frequency decomposition methods can provide a similar amount of temporal precision as ERPs, particularly for the later cognitive components.

3. *ITPC is more likely to show results in lower compared to higher frequencies* This is because ITPC requires precise timing of experiment events. The sensitivity of ITPC to temporal uncertainty increases with frequency, as the phase cycles become shorter. Temporal jitters may result not only from uncertainties in the timing of EEG experiment markers or hardware delays but also from the temporal smoothing that results from wavelet convolution or filtering. Thus, the results of an ITPC analysis may fail to detect true ITPC in the brain, particularly in higher frequency bands.

4. *EEG has limited spatial precision at the electrode level due to mixing of projections from source activities and (in the case of EEG but not MEG) from volume conduction through the skull/scalp* Strategies for improving spatial precision are discussed in chapters 22 and 24.

5. *Results of time-frequency-based analyses have increased multiple comparisons problems* This increased multiple comparisons problem is largely due to the increased dimensionality of the data compared to ERPs (or fMRI, for that matter), but it is also partly due to the relative novelty of time-frequency-based analysis approaches for investigating cognition. There are many fewer investigations into the time-frequency dynamics of cognitive processes compared to investigations into the ERP dynamics of cognitive processes. This means that there are fewer opportunities for hypothesis-driven analyses of time-frequency-based results, which means that exploratory data analyses are often required. Exploratory data analyses require more conservative statistical corrections for multiple comparisons and have an increased risk of failing to identify theoretically relevant but relatively subtle effects. This limitation should decrease over the coming years as more becomes known about the time-frequency characteristics of different cognitive processes and as theories begin to develop that account for frequency-band-specific dynamics that support cognitive processes.

6. *Whether time-frequency features can be interpreted as neural oscillations is debated* This issue is discussed below.

21.6 Do Time-Frequency Analyses Reveal Neural Oscillations?

This is often a contentious issue that some ERP-only researchers will use to question the validity or the usefulness of time-frequency-based analyses. The reasoning is that because the Fourier theorem dictates that all signals can be represented in the frequency domain, results of time-frequency analyses cannot be unambiguously interpreted as reflecting oscillations; instead, they may simply reflect the spectral representation of ERPs.

It is important in this discussion to make a clear distinction between two orthogonal but often confused points: (1) the usefulness of a large array of data-analytic approaches based on conceptualizing EEG data as containing time-frequency information; and (2) interpreting

results from time-frequency-based analyses as unambiguously indexing neurophysiological mechanisms of oscillations.

For the first point, the answer is a clear and unambiguous Yes—there is a large amount of information available in EEG data, and ERPs capture only a fraction of that information. Time-frequency-based analyses do not reveal all of the information in EEG data, but they will help link brain dynamics to behavior dynamics in ways not possible using ERPs (Cohen 2011b). It is not difficult to find examples in the literature where time-frequency dynamics predicted experiment conditions or behavior although the ERP did not. Any study involving non-phase-locked activity, activity over 20 Hz, prestimulus activity, resting-state activity, or connectivity results likely contains findings that were accessible only through time-frequency-based analysis techniques. Thus, from a purely practical perspective concerning the link between the EEG signal and cognition, time-frequency-based approaches should be used in addition to or instead of ERPs for cognitive electrophysiology.

As for the second point—whether features in time-frequency space can be interpreted as reflecting the known neurophysiological mechanisms of oscillations—this issue has little practical relevance for performing cognitive electrophysiology research other than governing the terms you use when speaking about and writing about your findings. The issue is that calling a pattern of EEG activity an oscillation carries some weight. The term “oscillation” within neuroscience implies certain neurophysiological processes that drive oscillations, such as interactions between inhibitory interneurons and excitatory pyramidal cells (chapter 5).

Determining whether your finding reflects oscillatory or nonoscillatory activity is difficult. You might hear the argument that nonoscillatory activity will be present for only one or two cycles, whereas oscillations will be present for many cycles, but there are three problems with this argument: (1) even one-cycle activity may be temporally extended due to temporal smoothing (which depends on time-frequency decomposition parameters; see figure 13.13); (2) there is no minimum number of cycles for there to be a true neural oscillation, and there is no reason why the neurophysiological mechanisms that generate neural oscillations cannot last one or two cycles; (3) temporally extended changes in band-specific activity are not necessarily oscillatory; several temporally delayed transient events will have a frequency representation that might appear continuous due to temporal smoothing, particularly when methods are used that utilize temporal smoothing to increase signal-to-noise ratio, such as the multitaper method. In other words, it is not possible to look at a time-frequency plot and know with certainty that some time-frequency feature does or does not reflect a neural oscillation, as defined by known physiological mechanisms that produce neural oscillations.

Another way to approach this issue is to consider that there is a latent variable (a neural oscillation) and there is a manifest variable (frequency-band-specific EEG activity). Calling a pattern of EEG activity an oscillation means you are inferring the existence of a latent variable based on a manifest variable. This is similar to the interpretation of the fMRI BOLD response as “neural activity” (an interpretation widely used in cognitive neuroscience), and it is similar to assuming that subjects have experienced a cognitive state called “conflict” because reaction times were longer in one condition than they were in another condition. In all of these examples an inference is made about a hidden neural or cognitive event (“oscillation,” “neural activity,” or “conflict”) based on what was actually measured (frequency-band-specific EEG activity, a hemodynamic response, or relatively increased reaction times). Anyone who is skeptical of interpreting time-frequency features as neural oscillations is justified in his or her skepticism. But if you are skeptical that observed time-frequency features reflect neural oscillations, then you should be even more skeptical that reaction time speed reflects the cognitive state of conflict and that changes in signal intensity values from echo-planar MRI scans reflect neural activity. Again, this is an issue of interpreting results that, though important, is and should be decoupled from the validity and usefulness of applying time-frequency-based analyses to neural time series data.

The main advantage of interpreting a pattern of time-frequency activity as an oscillation is that it links that activity to a large array of cross-species, *in vivo* and *in vitro*, empirical, theoretical, and computational studies of the roles of neural oscillations in brain function and cognition. After many decades of cognitive ERP research, there remains little understanding of the neurophysiological events that produce the temporal sequence of polarity-reversing but purportedly nonoscillatory voltage fluctuations that comprise ERPs.

In the end, it comes down to how you want to interpret the results. Calling the finding “band-limited” or “frequency-band-specific” is more conservative because it is an accurate description of the finding, whereas calling the result a “neural oscillation” is an interpretation of a hypothesized neurophysiological process that produced the band-specific activity. My advice is to use terms such as “band-specific” or state the center frequencies when describing empirical results (e.g., “a relative suppression of ~10 Hz power in condition A compared to condition B” or “theta-band power was maximal over midfrontal electrodes”) and to reserve the term “oscillation” when interpreting or speculating about the results (e.g., “oscillations in the alpha band have been implicated in active inhibition of processing” or “theta-band oscillations in prefrontal cortex might be used to coordinate large-scale networks”).

IV Spatial Filters

22 Surface Laplacian

Temporal filters such as wavelet convolution and FIR filters highlight features of the data by setting each data point to be a weighted sum of surrounding data points, and those weights are constructed in a way that maximizes selectivity to specific frequency bands. By applying temporal filters to the data, you can highlight aspects of the signal that are present but difficult to isolate in the unfiltered data.

Spatial filters are similar in concept. Spatial filters are created by defining weights for each electrode such that the weighted sum of activity at all electrodes helps highlight aspects of the signal that are present but difficult to isolate in the spatially unfiltered data. Differences among spatial filters arise from the type of information that is used when constructing the weights. The surface Laplacian defines weights according to interelectrode distances. Principal components analysis (chapter 23) defines weights according to patterns of interelectrode covariances. Some source estimation procedures (chapter 24) such as beamforming define weights based both on electrode activity and on physical electrode positions.

This chapter introduces you to the computations, uses, and interpretations of the surface Laplacian. The most relevant reference for this chapter is chapter 8 of the book *Electrical Fields of the Brain* (Nunez and Srinivasan 2006).

22.1 What Is the Surface Laplacian?

The surface Laplacian is a spatial bandpass filter that increases topographical selectivity by effectively filtering out spatially broad features of the data. Spatially broad topographical features are likely to reflect volume-conducted potentials (see figure 22.1A) or distributed but highly coherent cortical sources. Surface Laplacian-transformed data estimate the potential at the dura (that is, if you could have electrodes under the skull instead of on top of the scalp), focus the results on high-spatial-frequency components while attenuating low-spatial-frequency components, and are more appropriate for investigating interelectrode connectivity

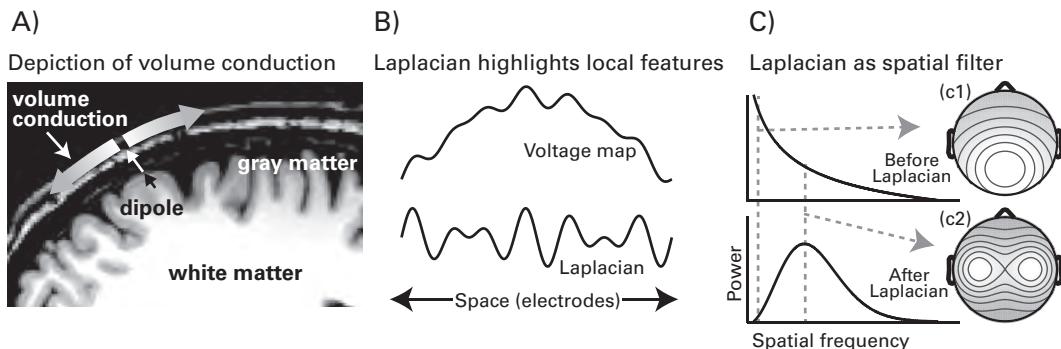


Figure 22.1

Conceptual overview of the motivations for and consequences of the surface Laplacian applied to EEG data. Panel A illustrates volume conduction whereby electrical fields spread tangentially at the boundary between the skull and the scalp. Panel B illustrates how low-spatial-frequency components in scalp EEG (top) are filtered out in the surface Laplacian (bottom). The surface Laplacian therefore highlights local spatial features that are present in the data but may be difficult to observe when summed with large-amplitude low-spatial-frequency components. Panel C illustrates that the surface Laplacian can be conceptualized as a spatial bandpass filter. In the illustrative power plots, spatial frequency is on the x -axis, and power is on the y -axis. Topographical maps c1 and c2 illustrate low and high spatial frequencies. The Laplacian will attenuate c1 while leaving c2 intact. Figure 22.4 demonstrates this phenomenon using simulated data.

(Srinivasan et al. 2007b; Winter et al. 2007). After application of the surface Laplacian transform, high-spatial-frequency activity—activity that is observable at only a small cluster of electrodes—is preserved, whereas low-spatial-frequency activity—activity that is observable at most or all electrodes—is attenuated (figure 22.1B).

In the literature you may also see the terms “surface current density,” “current source density,” or “current scalp density” (SCD or CSD). The term “source” here refers to sources and sinks of electrical activity at the level of the skull, not the putative brain regions that produce the scalp-recorded data. The CSD can be obtained via the surface Laplacian or by one of several other methods. The main difference between calling the spatial filter a CSD or a surface Laplacian is that with the former you are implicitly assuming that the resulting topographical distribution identifies locations of sources and sinks of electrical currents on the head, whereas with the latter you are simply describing the mathematical transform that was applied to the data. In other words the CSD and the surface Laplacian can be (and often are) the same thing, but the term surface Laplacian is preferred here because it is a brief description of the method rather than an interpretation of what the results of that method

might reflect. The surface Laplacian should be applied only to EEG (not MEG) data, and preferably to EEG data with 64 or more electrodes.

There are several advantages of computing and basing analyses on the surface Laplacian. First, the surface Laplacian improves topographical localization (see section 2.9 for the difference between topographical localization and brain localization). Second, the surface Laplacian minimizes volume-conduction effects, which makes the surface Laplacian attractive for electrode-level connectivity analyses, including spectral coherence (Srinivasan et al. 2007b; Winter et al. 2007), phase-based synchronization measures (Lachaux et al. 1999), and Granger prediction (Cohen and van Gaal 2012; Seth 2010a). Third, the surface Laplacian is reference independent because it is computed based on the second spatial derivative of the potentials. Finally, the surface Laplacian requires few parameters or assumptions, including assumptions about the locations of brain sources that are measured by scalp electrodes. The surface Laplacian also does not require assumptions about the conductivity of the different head tissues. This can be contrasted, for example, with brain source estimation procedures, which often require a large number of assumptions and parameters that can affect the results.

The Laplacian is more sensitive to radial dipoles than it is to tangential dipoles. Pyramidal cells that generate EEG are generally aligned perpendicular to the cortex regardless of cortical folding with respect to the skull. Thus, dipoles located in the sulci are farther away from the skull and will produce field potentials that have broader spatial distributions (see dipole d in figure 5.1). Thus, activity seen in the surface Laplacian is dominated (although not necessarily entirely driven) by radial dipoles in gyral crowns.

The main disadvantage of the surface Laplacian is a by-product of its advantage—spatially broad activities are attenuated and may become undetectable. For example, the P3b ERP component tends to have a fairly broad spatial distribution: it has a spatial peak at parietal electrodes but can be measured from many electrodes over the scalp. The P3b will be diminished in the surface Laplacian. This has two implications: first, results obtained via the surface Laplacian are likely to have been generated by relatively local and superficial sources (that is, regions of the cortex close to the skull, such as gyral crowns); second, the surface Laplacian should not be applied if you expect the results to be driven by deep sources such as the insula or by spatially distributed and highly temporally coherent generators, because deep sources and spatially distributed coherent sources generally produce broad spatial topographies that are attenuated after the Laplacian.

The Laplacian must be applied to time-domain data. That is, you should compute the Laplacian first and then perform time-frequency analyses, not compute the Laplacian of time-frequency results (Tenke and Kayser 2005). If you use the surface Laplacian only for improved topographical localization of an ERP effect, you can apply the surface Laplacian to

the ERP rather than to the single-trial data because the ERP is a simple linear transformation of the single-trial data. The equivalence of applying the surface Laplacian to single-trial data and then averaging, versus applying the surface Laplacian to averaged data, is demonstrated in the online Matlab code. Finally, the surface Laplacian must be applied to all of the data or to none of the data; it cannot be applied to parts of the data or only to the data from some conditions or from some subjects. The surface Laplacian is a filter and therefore changes both the features and the scale of the data.

The “raw” units of the surface Laplacian are microvolts per square millimeter ($\mu\text{V}/\text{mm}^2$), which is often converted to square centimeters. This is because the surface Laplacian is the second spatial derivative of the topographical activity. However, the magnitude of these values should not be overinterpreted, partly for reasons outlined in section 2.3 and partly because the smoothing parameters used to compute the surface Laplacian (discussed below) will change the values. For many time-frequency analyses that involve phase values or that use baseline normalization, the unit is not important because the data become rescaled to percentage change, decibels, or other values.

Because the surface Laplacian is based on spatial sampling and spatial weighting, the number of electrodes you have is important for the accuracy of the surface Laplacian. In general, having more electrodes is better. Sixty-four electrodes is a reasonable minimum, and >100 electrodes will provide more accurate results. That said, Perrin et al. (1987) used only 16 electrodes, although they were demonstrating a method for computing the surface Laplacian rather than using it to interpret EEG results. Typically, the surface Laplacian is computed using standard electrode locations, but there are methods to use individual skull shape and precise electrode positions to improve the accuracy of surface Laplacian (Deng et al. 2011, 2012).

22.2 Algorithms for Computing the Surface Laplacian for EEG Data

A basic approximation to the surface Laplacian is to subtract from each electrode the averaged activity of immediately surrounding electrodes (Hjorth 1975; Shepard 1968). This is not the most elegant solution, however: volume conduction does not spread only to nearest neighbors but rather to many electrodes up to tens of centimeters away, and the effect of volume conduction does not affect all neighboring electrodes equally but, rather, as a function of the distance between each electrode and the “source” electrode (that is, the skull/surface source of the electrical activity).

There are several algorithms to compute the surface Laplacian that are more accurate than nearest-neighbor subtraction. If the head were a completely flat two-dimensional (2-D)

surface, computing the second spatial derivative would be easy. Unfortunately, however, the head is not a flat plane but rather is a complex 3-D object that can be approximated by a sphere (although this is obviously a simplification). Thus, the computation of the surface Laplacian must be related to the spherical spatial derivative. Some solutions, often call deblurring or dura imaging, rely on three- or four-shell spherical or realistic head models with specified electrical conductivities at the different head tissues (Cadusch, Breckon, and Silberstein 1992; Le and Gevins 1993). Other solutions are based on spherical spline interpolations and make no assumptions about the conductivity of head tissues (one of these methods is highlighted below). It can be shown that many solutions to the surface Laplacian provide qualitatively similar and in some cases nearly identical results (an example of this is shown later in this chapter). Here, the spherical spline method by Perrin and colleagues (Perrin et al. 1987, 1989a, 1989b) is presented. This method is faster than the “New Orleans” method discussed in the Nunez and Srinivasan (2006) book while giving nearly identical results (online Matlab code is provided for both methods). Computation times for the surface Laplacian of the sample EEG dataset (63,360 time points over all trials) were 0.18 and 250 s on my laptop for the Perrin and New Orleans implementations, respectively, while providing nearly identical results (in fairness, I did not work on optimizing the Matlab code for the New Orleans method, so it may be possible to bring down its computation time to similar speeds as the Perrin method).

The first step in computing the Laplacian is to compute the G and H matrices. These are electrode-by-electrode weighting matrices that are used in the Laplacian computation:

$$G_{ij} = (4\pi)^{-1} \sum_{n=1}^{\text{order}} \frac{(2n+1)P_n(\cos dist_{ij})}{(n(n+1))^m} \quad (22.1)$$

$$H_{ij} = (4\pi)^{-1} \sum_{n=1}^{\text{order}} \frac{-2(n+1)P_n(\cos dist_{ij})}{(n(n+1))^{m-1}} \quad (22.2)$$

in which i and j are electrodes, and m is a constant positive integer that relates to the smoothness of the result. Values of m from 2 to 6 are reasonable, and it can be fixed at 4 (Perrin et al. 1987, 1989a), or 3 for high-density (>100 electrodes) recordings. Smaller values will produce results that contain only very high spatial frequencies, whereas larger values will produce results that contain only very low spatial frequencies.

P is the Legendre polynomial, which is often used for spherical coordinate distances; n is the order term for the Legendre polynomial (from one until order), which defines the spatial harmonic frequencies with respect to each electrode (see figure 22.2). A larger order parameter will increase the accuracy of the G and H matrices. Perry et al. recommend an order of 7. With 64 electrodes the accuracy seems to improve until an order of around 10, with little

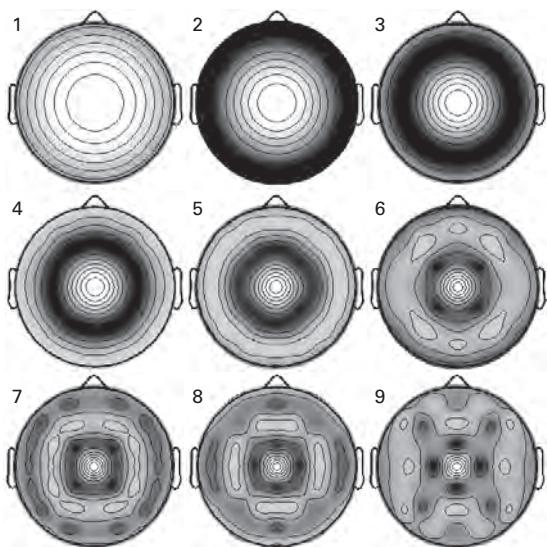


Figure 22.2

Topographical illustration of the Legendre polynomial “seeded” from electrode Cz. The Legendre result is related to distance and spatial frequency and is used to compute the G and H matrices. The first nine polynomials are shown here. With larger order, the spatial frequencies will eventually exceed the spatial sampling frequency, which is determined by the number of electrodes, thus providing no additional information. The surface Laplacian of EEG data with more than 64 electrodes may benefit to some extent from use of a larger order parameter than 10.

improvement or change in the G and H matrices (but longer computation time) beyond 10. You can either leave this parameter at 10 (the default setting in the function provided with the online Matlab code for 64 or fewer electrodes) or test for optimal values for your EEG cap and number of electrodes.

Think of this parameter as being related to the frequency precision of the results. If the order is small, you will extract only low-frequency components, which is probably the opposite of what you want from the surface Laplacian. With 64 electrodes, order values above 10 mean that the spatial frequency precision of the Laplacian exceeds the spatial resolution of the EEG cap, which is why values higher than 10 have little effect for 64 electrodes. For more than 100 electrodes, and, in particular, more than 200 electrodes, the results will benefit from larger Legendre polynomials, perhaps up to 13 or 15. However, as the order becomes large, only very high spatial frequencies can pass through the filter. This may impede cross-subject averaging and comparisons. Thus, the Legendre polynomial order should be set to a value that is high enough to isolate high-spatial-frequency activity but not so high

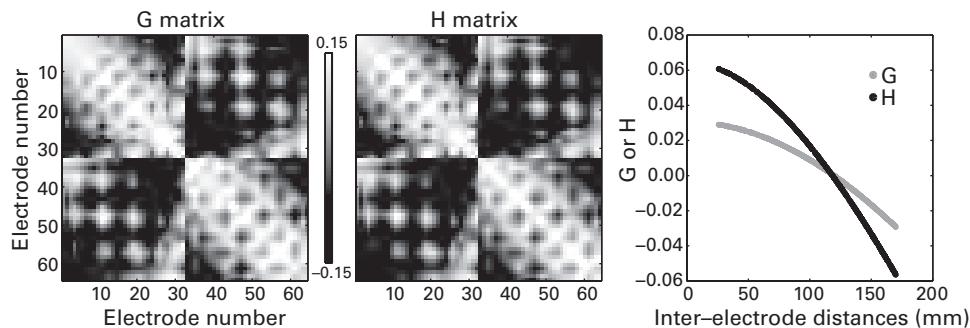


Figure 22.3

G and *H* weighting matrices. These matrices are the weights that are applied to the data to attenuate low-spatial-frequency components.

as to isolate only extremely high spatial frequencies that will reflect activity only at single electrodes.

The *cosdist* term in the Legendre polynomial is the cosine distance among all pairs of electrodes (electrode locations are normalized to a unit-radius sphere because the Legendre requires distances between -1 and 1) and is computed according to the following equation:

$$\text{cosdist}_{ij} = 1 - \frac{(X_i - X_j)^2 + (Y_i - Y_j)^2 + (Z_i - Z_j)^2}{2} \quad (22.3)$$

The *G* and *H* matrices can be thought of as the weights that are applied to the data such that the activity at each electrode becomes a weighted sum (weighted by *G* and *H*, thus weighted by distance) of activity at all other electrodes. The details are provided below, but this is a useful simplified conceptualization. Figure 22.3 shows the *G* and *H* matrices as well as the elements of the matrices plotted as a function of interelectrode distances.

You may have noticed that, so far, no data have been introduced to the equations in this chapter. The *G* and *H* matrices are based entirely on interelectrode distances and specified spatial frequency ranges, and are not related to the time-series data. This means that, technically speaking, the *G* and *H* matrices need to be computed only once per electrode montage, and as long as the number and locations of the electrodes do not change across subjects, the same matrices can be used for all time points and all datasets. In practice, however, the *G* and *H* matrices can be computed very quickly, so there is no need to precompute and save the matrices for each electrode montage.

Once the *G* and *H* matrices are created, they are then used to compute the Laplacian. The formulas are below; note that these formulas may seem different from those presented in

Perrin et al. (1989a, 1989b), but they are simply rearranged solutions and are numerically identical.

$$\text{lap}_i = \sum_{j=1}^{\text{nelec}} C_i H_{ij} \quad (22.4)$$

lap_i is the Laplacian for electrode i at one time point, and j refers to each other electrode (nelec is the total number of electrodes). H_{ij} is the element in the H matrix corresponding to electrodes i and j . The C matrix is where the data are finally introduced.

$$C_i = d_i - \frac{\sum_{j=1}^{\text{nelec}} d_j}{\sum_{j=1}^{\text{nelec}} Gs_j^{-1}} Gs^{-1} \quad (22.5)$$

$$d_i = \text{data}_i^{-1} Gs \quad (22.6)$$

$$Gs = G + \lambda \quad (22.7)$$

where λ is a smoothing parameter added to the diagonal elements of G (hence, Gs is the smoothed G matrix). Values between 10^{-5} and 10^{-6} are reasonable. Larger values (e.g., 10^{-3}) will produce overly smooth results. Smaller values (e.g., 10^{-7}) may produce more precise topographical results, but this may come at the expense of impeding cross-subject spatial averaging. For more than 128 electrodes, very small values may also cause instabilities in the matrix inversions. In general, you should use a λ of 10^{-5} for 64 electrodes; for more than 64 electrodes (e.g., 256), smaller values such as 10^{-6} or 5^{-6} might be appropriate. If you are using a high-density EEG setup, it would be a good idea to test a range of λ values on sample data before applying that value to the entire dataset. Whatever smoothing parameter you choose should be applied to all data (from all conditions and all subjects). Note that different smoothing parameter values will alter the “raw” data values ($\mu\text{V}/\text{cm}^2$) because the spatial derivative will change more or less quickly. However, because this will affect all data, it will not change condition differences, and it will have no effect on transformations such as decibels or percentage change, or on the phase angle time series from time-frequency decompositions.

These equations may look a bit confusing, but the general point to notice is that a weighted sum of activity at all electrodes is subtracted from the activity of each electrode. This is done first using the G matrix (equation 22.5), and then the result is scaled again using the H matrix (equation 22.4). Hence, it is a second spatial derivative.

These equations are presented as if the Laplacian is computed separately for each electrode and for each time point, but this is merely to facilitate explanation. In fact, the Laplacian can be computed simultaneously for all electrodes and for all time points using matrix algebra. Eliminating the double loop leads to a >1000-fold decrease in computation time compared to implementations that compute the surface Laplacian for each electrode and time point separately. This decreased computation time may also be of practical benefit for applications in which data must be processed rapidly, such as neurofeedback or brain-computer interfaces. One potential drawback of the loopless method is that if the datasets are very large and the analysis computer is memory challenged, Matlab might have an out-of-memory crash resulting from the matrix transpositions. If you encounter this problem, try performing the surface Laplacian on smaller time segments of data. Because the Laplacian transform is based on space and not on time, there are no edge artifacts as there are for temporal filtering. The online Matlab code that performs the surface Laplacian (`laplacian_perrinX.m`) will accept 2-D (electrodes by time) or 3-D (electrodes by time by trials) input matrices.

22.3 Surface Laplacian for Topographical Localization

The surface Laplacian will help isolate topographical features to within several square centimeters. This is because spatially broad topographical components that are driven by deep or powerful volume-conducted sources are generally outside the spatial frequency range of the Laplacian filter and are therefore attenuated.

The effect of computing the surface Laplacian on simulated data can be seen in figure 22.4. In this simulation, topographical data were generated by placing spatial Gaussian “activities” over electrodes Pz, C3, and C4. Panels A and B show, respectively and separately, low- and high-spatial-frequency features. Panel C shows the sum of the features in panels A and B. Remarkably, panel D shows that the Laplacian removes the low-spatial-frequency component while preserving the two high-spatial-frequency components. Figure 22.5 shows an example using real EEG data.

Because the surface Laplacian attenuates low-spatial-frequency features, it will also attenuate artifacts that contaminate your data if those artifacts are characterized by low spatial frequencies. The blink and other EOG artifacts are examples of this. EOG artifacts have a fairly broad spatial topography and are therefore attenuated after the surface Laplacian. This might be useful in situations in which an independent components analysis could not cleanly isolate all of the EOG activity. This feature of the surface Laplacian is particularly

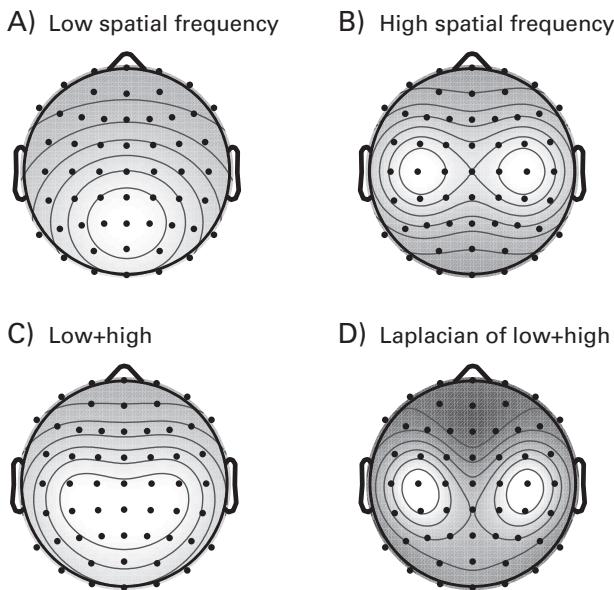
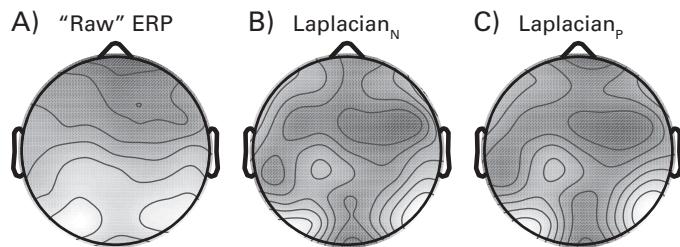


Figure 22.4

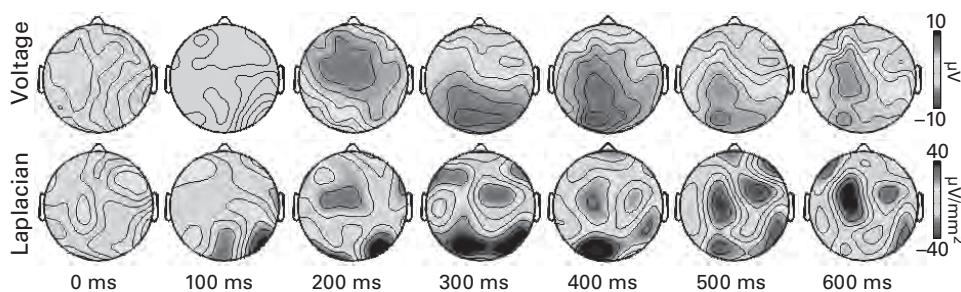
Simulated data showing the effect of the surface Laplacian on features of different spatial frequencies. Panel A shows one low-spatial-frequency feature, and panel B shows two high-frequency features (features were created as two-dimensional Gaussians centered on electrodes Pz, C3, and C4, with varying standard deviations). Panel C shows the sum of the topographies in panels A and B. It is difficult to see any individual feature; in fact, the topography appears to be one broad parietal component. However, the surface Laplacian of the sum (panel D) filters out the low-spatial-frequency feature shown in panel A and therefore recovers the two high-spatial-frequency features shown in panel B. Grayscale scaling is arbitrary for each map to facilitate visual comparison. The online Matlab code has an additional example of how the surface Laplacian can help isolate and disambiguate spatially nearly overlapping features. You can also use the online Matlab code to modify the spatial proximities and Gaussian variances to observe the performance of the surface Laplacian under different spatial configurations.

beneficial if you will examine activity at anterior electrodes and want to be confident that EOG activity does not contaminate the results. The online Matlab code that generates figure 22.4 will show an example of this using simulated data.

Spatially unfiltered data and the surface Laplacian show different cortical electrical features: spatially unfiltered topographies tend to be dominated by large-amplitude, low-spatial-frequency (and thus, likely volume-conducted) sources, whereas surface Laplacian topographies are driven by local sources that may be smaller in amplitude (as illustrated in figure 22.1B). For this reason, it might be useful to compare your results before and after

**Figure 22.5**

The effect of the surface Laplacian on real data (the ERP from 246 ms post-stimulus onset). Panels B and C show that two different algorithms to compute the surface Laplacian—N corresponds to the method discussed in the Nunez and Srinivasan (2006) book, and P corresponds to the method presented in Perrin et al. (1987)—produce highly similar results, in this case resulting in a spatial correlation coefficient of 0.9798. By modifying the smoothing parameters in the two algorithms, it would be possible to increase the spatial correlation coefficient.

**Figure 22.6 (plate 12)**

Comparison of spatially unfiltered voltage topographies (top row) and surface Laplacians (bottom row) of the same data. The color scaling is the same for all time points within each row. Note, for example, the topographical maps at 200 ms, for which the clearly visible right lateralized occipital component seen in the surface Laplacian is difficult to observe in the unfiltered data.

computing the surface Laplacian (figure 22.6; plate 12). This comparison will provide insight into whether the topographical patterns of activity are likely to have been generated by local sources or whether they are likely to reflect volume-conducted effects.

In the comparison of spatially unfiltered and surface Laplacian topographies, there are three possible results. First, the topographical activity appears similar in both the spatially unfiltered data and the surface Laplacian. This suggests that the activity was locally generated and was not overpowered by larger, volume-conducted components. An example of this can be seen in figure 22.6 (plate 12) for the left frontocentral component at 600 ms. In these

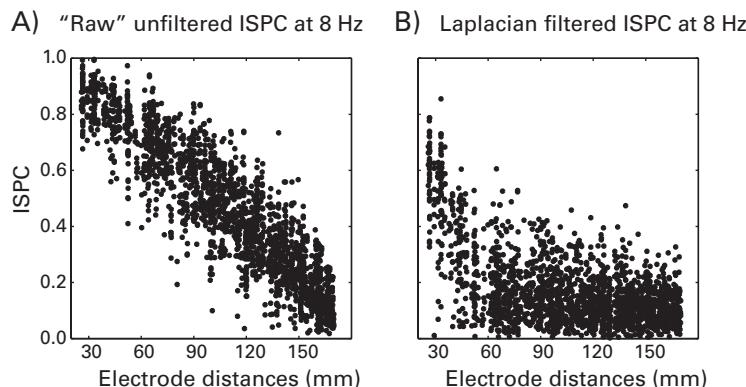
situations the time courses of EEG activity from spatially unfiltered and Laplacian data will look similar at the “source” electrode. The second possible result is that there are topographical patterns visible in the surface Laplacian that are not visible, or are difficult to visualize, in the spatially unfiltered data. This is the most typical situation and suggests that there are local cortical dynamics that are overshadowed by volume-conducted effects, and those local dynamics are brought to light by the surface Laplacian. There are several examples of this in figure 22.6 (plate 12), including the right lateralized occipital component at 200 ms, the bilateral occipital components at 300 ms, and the relatively negative component at right frontocentral areas from 300 to 600 ms. Finally, there can be topographical patterns visible in the spatially unfiltered data that are not visible in the Laplacian. This suggests that that spatial pattern reflects volume-conducted activity.

When the topographies of the spatially unfiltered data and the surface Laplacian differ, the time courses of activity at the same electrode will be different. This is because the surface Laplacian filters the data to highlight activity at a specific spatial frequency range. Analogously, if you filter the same broadband time series data at 2 Hz and at 20 Hz, those filtered time courses will be different even though they were taken from the same data.

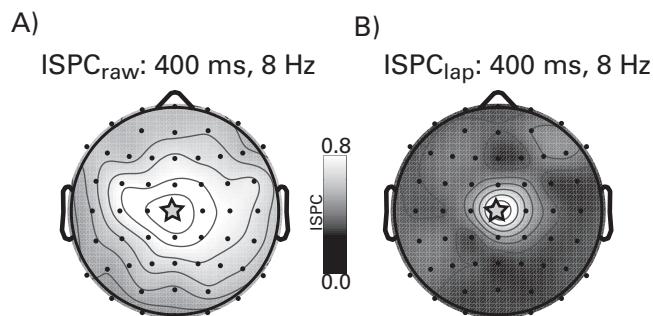
22.4 Surface Laplacian for Connectivity Analyses

Applying the surface Laplacian to EEG data helps render the data appropriate for connectivity analyses. This is because the surface Laplacian minimizes volume-conducted effects that are a potential source of spurious connectivity results (this is discussed further in section 25.10). Figure 22.7 illustrates how, before the Laplacian is computed, connectivity is driven almost entirely by interelectrode distance. Indeed, around 80% of the variance in connectivity in this example can be explained purely by distance (figure 22.7A). In contrast, after the surface Laplacian has been computed, connectivity is strongly correlated with interelectrode distance only for immediate neighboring electrodes (figure 22.7B).

Figure 22.8 further illustrates this point by showing “seeded” connectivity from one electrode. The topographical maps show the strength of connectivity between each electrode and Cz. It is clear that there is a strong relationship between connectivity strength and distance from Cz in figure 22.8A. In contrast, the relationship between interelectrode distance and connectivity strength seems to hold only for nearest neighbors after the surface Laplacian has been applied (figure 22.8B). In general, it is safe practice to avoid interpreting connectivity from electrodes within 5 cm of the seed electrode, which, for EEG caps with 64 electrodes, corresponds to one neighboring electrode.

**Figure 22.7**

Panel A shows that without any spatial filtering, interelectrode connectivity strength (here measured through ISPC, a phase-based measure of connectivity that you will learn about in chapter 26) is strongly predicted by interelectrode distance; panel B shows that this connectivity is largely decoupled from distance after application of the surface Laplacian, except at very short distances. Interelectrode distance accounted for 80.69% of the variance (r^2 from a Spearman correlation between distance and connectivity strength) in connectivity for non-spatially filtered data, whereas it accounted for 18.66% of the variance for surface Laplacian-transformed data. When immediate neighboring electrodes were removed, these percentages dropped to 75% and 7%, respectively.

**Figure 22.8**

Seeded connectivity from electrode Cz. The grayscale intensity shows the strength of connectivity between Cz and each other electrode. Similar to figure 22.7, connectivity strength decreases gradually as a function of distance for non-spatially filtered data (panel A), whereas connectivity strength drops precipitously after the immediate nearest-neighbor electrodes for surface Laplacian-filtered data (panel B).

It is important to stress that the surface Laplacian does not guarantee that all volume-conduction effects are eliminated. Thus, it is not the case that you can simply apply the surface Laplacian and then be confident that volume conduction will never produce spurious connectivity results. Rather, the surface Laplacian is one of several strategies that will minimize the possibility that connectivity effects were spuriously caused by volume conduction. The issue of volume conduction in connectivity analyses is discussed in more depth in section 25.10.

22.5 Surface Laplacian for Cleaning Topographical Noise

Although the surface Laplacian is sometimes referred to as a high-pass spatial filter, it is actually a bandpass spatial filter. This means that the surface Laplacian will remove not only low-spatial-frequency components but also very-high-spatial-frequency components. For a 64-electrode EEG, there are few or no very-high-spatial-frequency components, simply because the electrode spacing is relatively large (in other words, the spatial Nyquist frequency is low, so high-spatial-frequency features cannot be measured). Thus, for 64 electrodes, the Laplacian can effectively be a high-pass spatial filter.

However, with high-density EEG (>100 electrodes and, in particular, 256 electrodes), the spatial resolution is high enough that the bandpass spatial filter property of the surface Laplacian can be observed. This means that very-high-spatial-frequency components of the data will be attenuated in addition to the low-spatial-frequency components. This can be useful for cleaning some topographical noise. Because single-electrode noise spikes have a high spatial frequency, they are attenuated by the surface Laplacian. Figure 22.9 (plate 13) shows an example of this using data recorded from a 256-electrode EEG net. The topography is of a single time point before any data processing or cleaning has been performed. You can see that the Laplacian attenuated some of the high-spatial-frequency components, which are reflected as single-electrode events. Some of these single-electrode noise spikes would be removed during normal preprocessing such as temporal filtering.

Applying the surface Laplacian should not be relied on as an excuse for collecting noisy data, nor should it replace normal data-cleaning and preprocessing protocols. However, this feature of the Laplacian is another advantage of its application as a processing tool for use in data analyses, particularly for high-density EEG caps. As mentioned earlier in this chapter, the surface Laplacian must be applied to all data or to no data; you cannot apply the surface Laplacian to only a few noisy time points.

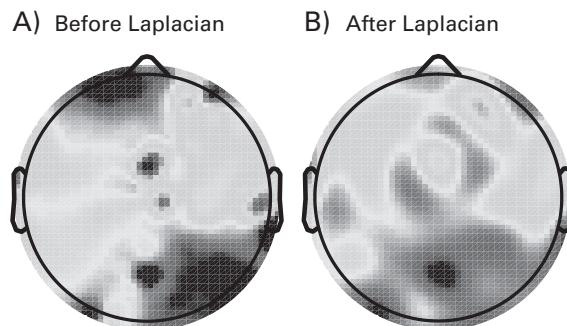


Figure 22.9 (plate 13)

The surface Laplacian attenuates high-spatial-frequency features in addition to low-spatial-frequency features. One time point of activity from a 256-electrode net is shown before (panel A) and after (panel B) application of the surface Laplacian. Because the single-electrode fluctuations (which are probably due to noise) have a very high spatial frequency representation, they are attenuated due to the bandpass filtering of the Laplacian.

22.6 Describing This Analysis in Your Methods Section

Perhaps most importantly, you should state the purpose of applying the surface Laplacian to your data. The most common motivations for using the surface Laplacian are to increase topographical localization, to facilitate electrode-level connectivity analyses, and to attenuate volume-conducted features of the data that overshadow the effects you might be interested in investigating. If you examined the results with and without the surface Laplacian, state whether any patterns of results changed in qualitative ways.

There are several algorithms for computing the surface Laplacian; be clear about which algorithm you used. It is not necessary to write out all of the equations for the Laplacian unless you make modifications to the algorithm. If you used nondefault parameters, for example for smoothing the G matrix, state the values of the parameters you used and justify the departure from commonly used parameter values.

22.7 Exercises

1. Based on the topographical maps of ERPs in figure 22.6 (plate 12), select one electrode whose activity you think might look similar before and after computing the surface Laplacian, and one electrode whose activity you think might look different before and after the surface Laplacian.

2. Perform a time-frequency decomposition of the data from those two electrodes both before and after computing the surface Laplacian (that is, compute the surface Laplacian on the raw data before applying a time-frequency decomposition). Compute both power (decibels normalized using a baseline period of your choice) and ITPC.
3. Plot the results using the same color scaling for before and after the surface Laplacian.
4. Are there any salient differences in the time-frequency power or ITPC results before versus after application of the surface Laplacian, and do the differences depend on the frequency? How would you interpret similarities and differences at different frequency bands?

23 Principal Components Analyses

23.1 Purpose and Interpretations of Principal Components Analysis

Chapter 10 introduced the idea that the EEG data from one time point can be conceptualized as a location in a high-dimensional space, with each dimension corresponding to each time-point (or, for purposes of this chapter, each dimension corresponds to each electrode). In this framework, EEG data over time can be conceptualized as a line traveling through this high-dimensional space. This conceptualization is useful for principal components analysis (PCA). The goal of PCA is to construct sets of weights (called principal components) based on the covariance of a set of correlated variables (here, electrodes) so that components explain all of the variance of the data such that the components (1) are uncorrelated with each other and (2) are created so that the first principal component explains as much variance as possible for one variable, the second principal component explains as much of the residual variance as possible for one variable while being orthogonal to the first component, and so on, for as many components as there are electrodes (variables). Another way to explain PCA is that it rotates axes in an N -dimensional space (where N is the number of variables or electrodes) such that the variance along each dimension is minimized, and the axes are orthogonal to each other. An example of principal components for two-dimensional (2-D) data is represented visually in figure 23.1.

There are several ways to interpret the result of a PCA on EEG data. One interpretation is geometric—each principal component is a vector in a 64-dimensional space (one dimension for each electrode) that characterizes the “direction” of the data distribution (as seen in figure 23.1). Another interpretation is that PCA can be used to create a set of spatial filters in which the weights for each electrode are defined by patterns of interelectrode temporal covariance. The PCA therefore highlights specific features of the data that might be difficult to identify in the spatially unfiltered data, because they are created by weighted combinations of all electrodes. PCA can also be interpreted as a data reduction technique whereby high-dimensional

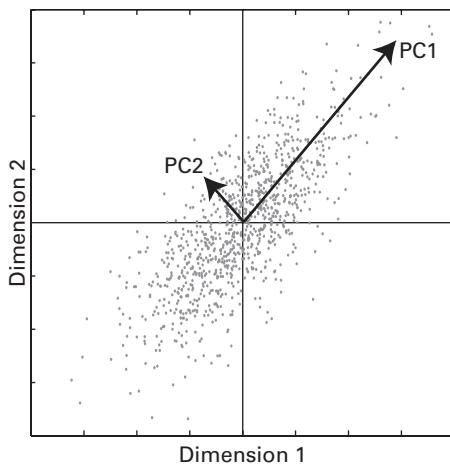


Figure 23.1

Conceptual illustration of PCA. Two-dimensional data (gray dots) are correlated; PCA returns new axes such that the first axis (principal component; PC1) accounts for as much variance as possible along one dimension. The eigenvector is related to the direction of the dimension, and the eigenvalue is related to the length of that vector and thus to the amount of variance accounted for. The second axis (PC2) accounts for as much of the residual variance as possible while being orthogonal to the first axis (the orthogonality constraint can be relaxed in some applications of PCA). You can also see that the variance about axes PC1 and PC2 is smaller than the variance about axes Dimension 1 and Dimension 2 (the original x - and y -axes). Components are also sometimes called modes.

data are reduced to a smaller number of dimensions by assuming that the components that account for a relatively large proportion of variance reflect true signal, whereas components that account for relatively little variance reflect noise.

Recall from the previous chapter that the surface Laplacian computes weights based on interelectrode distances. For PCA, weights are defined only according to statistical properties of the data; physical electrode locations are not considered. The PCA and the surface Laplacian can be conceptualized as spatial filters that provide information about opposite sides of the spatial spectrum: whereas the surface Laplacian explicitly attenuates low-spatial-frequency activity and therefore highlights local features of the data, PCA highlights global spatial features of the data by identifying patterns of large-scale covariance. Local topographical features contribute relatively little to the covariance matrix and are thus less likely to contribute to the first few principal components that are typically analyzed.

PCA can be computed over any set of variables and observations; in this chapter, PCA is presented with electrodes as variables and time points as observations. However, PCA is a

general framework for identifying patterns of covariance among variables regardless of the dimension used to define the covariance matrices. For example, PCA can be computed from frequency information (Miller et al. 2009) or from time-frequency results. It can be used within subject, thus treating trials as observations (as presented here), or it can be used across subjects, thus treating subjects as observations. Applications of PCA not discussed here (e.g., over space, frequency, subject, condition) may change the interpretation of the results but not the underlying mathematical procedures or Matlab implementations.

23.2 How PCA Is Computed

The first step of computing a PCA is to construct a covariance matrix. A covariance matrix is like a correlation matrix, but it retains the scale of the data. It can be obtained using matrix algebra by subtracting the mean and multiplying the data (electrodes \times time points) by its transpose.

$$\text{covariance} = (n - 1)^{-1} (X - \bar{X})(X - \bar{X})^T \quad (23.1)$$

in which X is an electrodes-by-time-points matrix (or, more generally, a variables-by-instances matrix), and \bar{X} is the mean signal of each electrode over time. If your data are time-by-electrodes, you would need to transpose the first matrix, not the second matrix (or transpose the data to be electrodes by time). Subtracting the mean before computing the covariance is necessary for the PCA to follow patterns of variance; otherwise, the first component will reflect the mean signal. Because this is a covariance matrix, the diagonals will contain the variances of each electrode. n is the number of instances, or time points (not the number of variables). Dividing by $n - 1$ is a normalization factor. Sometimes you will see the denominator as n instead of $n - 1$; $n - 1$ should be used when sampling from a population, which is usually the case in cognitive electrophysiology—EEG data are samples of a theoretical population of time points and trials. Dividing by n can introduce a bias in the covariance at small n (e.g., less than 50). For larger n , the bias is likely to be negligible. EEG data covariance matrices are often computed from hundreds or thousands of observations, so whether n or $n - 1$ is used is unlikely to have a significant negative impact on the PCA results.

There are three ways to organize the data to apply equation 23.1. First, the covariance matrix can be computed from the ERP. Second, it can be computed from all time points from all trials, concatenated into a large 2-D matrix (in this case, $64 \times 63,360$ points, corresponding to 640 time points and 99 trials). Finally, the covariance matrix can be computed separately for each trial, and then the covariance matrices from each trial can be averaged together. This third approach is advantageous because it increases the signal-to-noise ratio

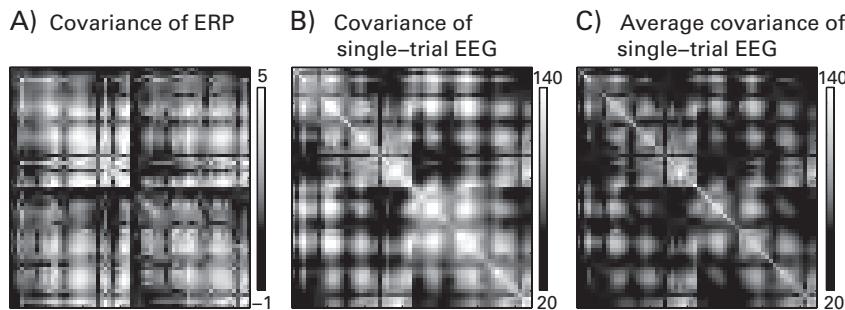


Figure 23.2

Matrices showing covariance (units are μV) among all pairs of electrodes computed in slightly different ways. Panel A shows the covariance of the ERP; panel B shows the covariance of all time points (that is, all time points and trials were concatenated, and one covariance matrix was computed across all 63,360 time points); and panel C shows the average covariance of single trials (that is, the covariance was computed separately for each trial, and then 99 covariance matrices from 99 trials were averaged together). In most cases the procedure illustrated in panel C should be preferred because it increases signal-to-noise ratio while being sensitive to both phase-locked and non-phase-locked activity.

of the covariance matrix. The covariance matrices resulting from these three approaches are shown in figure 23.2. Although the latter two methods generally produce similar results, there is a distinction between the first and the latter two covariance matrices: the covariance of the ERP can be conceptualized as reflecting the phase-locked (evoked) covariance, whereas the covariance of the single-trial EEG data can be conceptualized as reflecting the total (phase-locked and non-phase-locked) covariance.

Once a covariance matrix is computed, the next step is to perform an eigendecomposition via the Matlab function `eig` (you can also perform a singular value decomposition using the Matlab function `svd`, which should yield identical results to the eigenvalue decomposition in most cases). An eigendecomposition is a matrix decomposition that returns eigenvectors and their associated eigenvalues, which characterize the patterns of interelectrode covariance. Thinking back to figure 23.1, the eigenvectors are the new rotated axes, and the eigenvalues are the lengths of those axes. In the parlance of PCA the eigenvectors are the principal components or the weights for each electrode that, when applied to the electrode time series, produce the PCA time courses. The eigenvalues can be scaled to percentage variance accounted for by dividing each eigenvalue by the sum of all eigenvalues and multiplying by 100. Converting eigenvalues to percentage variance accounted for provides an easily interpretable metric and has the added benefit that it puts all eigenvalues in the same scale and thus comparable across conditions, time windows, subjects, and so on regardless of the

scale of the original data. Note that the Matlab function `eig` returns eigenvalues and eigenvectors in ascending order; you might find it more intuitive to have the components sorted in descending order (thus, the biggest component is the first rather than the last).

The results of the eigenvalue decomposition are square matrices with as many rows/columns as there are variables (electrodes). Each column in the eigenvector matrix is a principal component, and each row stores the weights (also called component loadings) of each electrode. Because this is a square matrix, it is easy to confuse which dimension is for components and which dimension is for electrode weights. This is a critical distinction, so make sure you are using the correct dimension. If you are unsure, make a topographical map of the first principal component—it most likely has a topographical distribution dominated by one low spatial frequency component (see online Matlab code for an example).

The electrode weights for each component can be plotted as topographical maps, and time courses of the components can be obtained by multiplying the weights by the electrode time series data. Note that this means that each component has one associated time course, and that time course is a weighted sum of the activity of all electrodes.

The difference between performing the PCA on a covariance matrix defined from the ERP versus from single trials can be seen in the PCA maps in figure 23.3. Panel A shows the first six PCA topographical maps and time courses from the covariance matrix based on the ERP, and panel B shows the first six components from the covariance matrix based on the single trials (the covariance matrix shown in figure 23.2C). Although the two analyses were based on the same time-domain dataset, there are noticeable differences between them. One qualitative difference, for example, is that PC4 in panel B appears to be a bilateral sensory-motor component that is not apparent in the first six components of panel A. This may be because the data are stimulus locked; because the response time varies across trials, response-related activity has little representation in the stimulus-locked ERP, although it is present in each trial.

Once the component time courses are generated via the weighted sum of all electrode activities, this time course can be analyzed in a similar fashion to the way you would analyze electrode time courses. Figure 23.4 (plate 14), for example, shows ERPs and time-frequency power plots from the first two components (based on the single-trial covariances) that are shown in figure 23.3B. Remarkably, the first two components seemed to have identified a component for early sensory processing (PC2) and a component for cognitive processing (PC1). This interpretation is based on their topographical maps and on the timing of their activity that is visible in the ERP and in the time-frequency plots in figure 23.4 (plate 14). Remember that these dynamics are not localizable to any one electrode; they reflect the

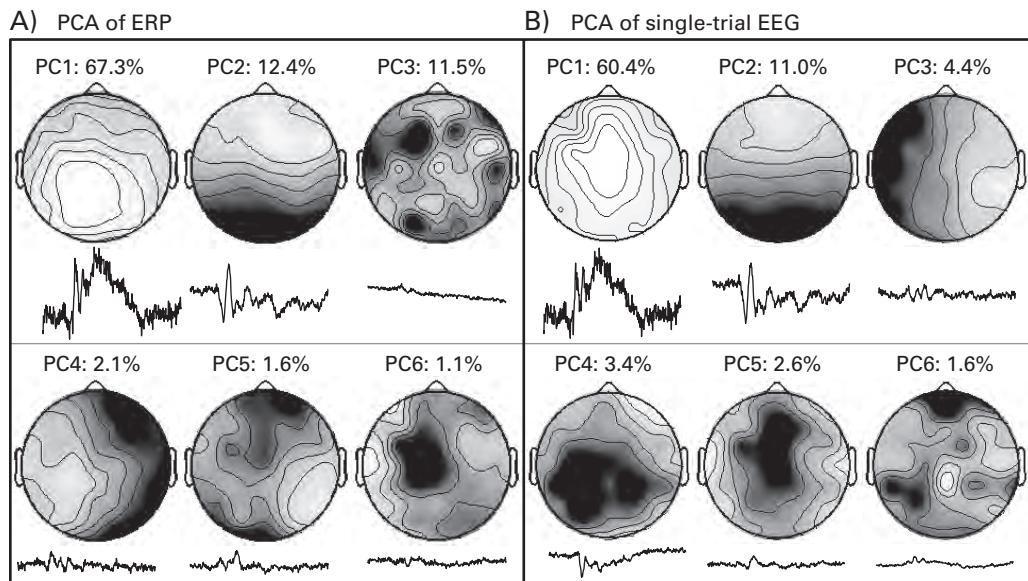


Figure 23.3

The first six principal components (PC), their percentage variance accounted for, and their associated time courses, plotted separately for the PCA based on the covariance matrix from the ERP (panel A) and from the single-trial EEG (panel B). Time courses are all scaled to the same magnitude, and thus, their vertical deflections are comparable; grayscaling of each map is arbitrary, and thus the shades of gray are not comparable across maps.

processing of large-scale networks that are defined based on patterns of time-domain and broadband covariance matrices.

The sign of the PCA weights is somewhat arbitrary, meaning that it is possible for the electrode weights to be inverted. Slightly different algorithms (e.g., eigenvalue decomposition vs. singular value decomposition, or the present analyses vs. the Matlab function `princomp`) may produce identical or nearly identical results but with some components of opposite signs. The Matlab function `princomp`, for example, forces the largest component to be positive regardless of its original sign. Depending on what you plan on doing with the PCA results, potential flips in sign may not matter. If the sign is important, there are several methods to ensure a nonarbitrary PCA sign, including setting the sign to be positive for the largest component (as the Matlab function `princomp` does) or setting the sign according to the direction of the data distributions (Bro, Acar, and Kolda 2007).

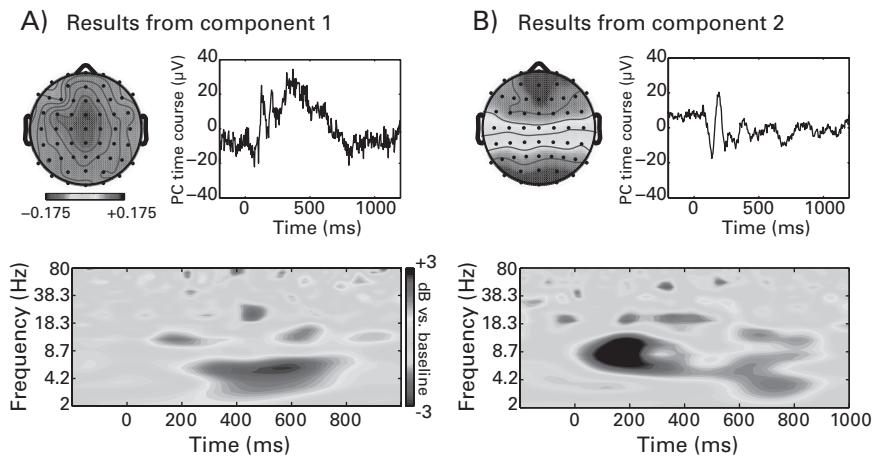


Figure 23.4 (plate 14)

Activities from the first two PCs. The plots show the topographical map, trial-averaged component time course (this is the ERP of that component), and the time-frequency power (both panels use the same color scaling for the topographical maps and for the time-frequency power plots). These two components reflect large-scale spatial processing modes and seem to have dissociated early sensory processing from later cognitive processing, although they were both computed from the same data. Note that these two components are, by definition, orthogonal to each other; whether the neurocognitive dynamics they capture are orthogonal to each other is a separate issue.

23.3 Distinguishing Significant from Nonsignificant Components

The number of components in the PCA is equal to the number of variables (electrodes). However, when considering a plot of the percentage variance accounted for over components, you will notice that only the first several components account for “a lot” of variance, and most of the later components account for relatively little variance. Often in the PCA the first several components will be considered signal, whereas the later components will be considered noise. Keep in mind that PCs are defined based on strengths of interelectrode covariance. Thus, weak interelectrode covariance would result in components with small eigenvalues even if they contain meaningful signal. In other words, local and topographically constrained activity will be reflected in a PC that accounts for relatively little variance in the covariance matrix, although it may be theoretically and empirically significant (thus, it may be signal, not noise). Indeed, the previous chapter pointed out that spatially local features of the data can be meaningful and can be obscured by spatially broad features. This

is why labeling components as “signal” or “noise” based only on their eigenvalues should be done cautiously. For the rest of this section, components are referred to as significant or nonsignificant.

How do you know which components are significant? (If you decide to rotate the components, as described in section 23.4, the question becomes: How do you know which components to retain for rotation?) Distinguishing significant from nonsignificant components involves determining a percentage variance threshold and considering components that explain variance above that threshold to be significant. There are several ways to determine the threshold. One way is to compute the percentage explained variance that would be expected from each component if all electrodes were uncorrelated with each other. In this case, each component would explain $[100 \times 1/M]$ percent of the variance, where M is the number of electrodes/components (figure 23.5).

Another way to determine a threshold is to use permutation testing, whereby the data are randomly shuffled, a PCA is computed on the shuffled data, and the amount of variance explained in shuffled data, averaged over many repetitions, is taken as the threshold (also shown in figure 23.5). The advantage of permutation-based thresholding over the analytic

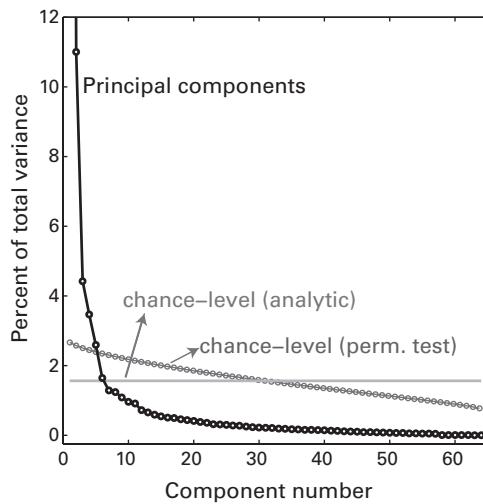


Figure 23.5

Percentage of total variance accounted for by each of 64 components, along with thresholds based either on an analytic solution (the expected amount of variance accounted for assuming that each electrode contributes completely independent information) or on permutation testing (shuffling data points and recomputing the PCA 1000 times). The first component is cut out of the plot to facilitate visual inspection.

method described in the previous paragraph is that the data distributions and characteristics are retained. This method is shown in the online Matlab code, and more details concerning permutation testing are discussed in chapter 33. Other methods for determining whether a component is significant include examining changes in the slope of the eigenvalue function or visually inspecting the plot of variance accounted for by each PC (this method may be impractical if you are performing PCA over many time segments and frequency bands). Different thresholding procedures should yield fairly similar though not necessarily identical results. For example, in figure 23.5, the two different thresholding procedures described here would result in labeling one component as significant or nonsignificant.

Keep in mind that if you performed independent components analysis and subtracted some components from the data, the rank of the data may be lower than the number of electrodes. The rank of a matrix refers to the number of linearly independent dimensions in the data. For example, imagine that you created a new electrode by summing together three other electrodes. Although you would now have 65 electrodes, the rank of the data matrix has not increased because that new electrode is a perfect linear combination of other electrodes. If you subtract independent components from the data during preprocessing, it is possible that the rank of the data will decrease (this depends on how many components were subtracted and how much variance those components accounted for). The sample online EEG data has a rank of 62. The reduced rank of the covariance matrices means that the last few PCs will likely contain no information. Indeed, the last two PCs of the online EEG data account for less than $10^{-6}\%$ of the variance of the data, which is basically zero percent plus some computer rounding error.

Once the significant components are identified, there are several subsequent analysis options. You can treat PCA as a dimensionality reduction technique and analyze only the significant components rather than all 64 electrodes (for example, as shown in figure 23.4; color plate 14). This should help minimize noise and reduce multiple comparison problems. Or you can treat PCA as a way to denoise the data by projecting back only the significant components. As mentioned earlier, keep in mind that this procedure selects features of the data that have large topographical covariances; local data features may be excluded from such a component selection process, even if they reflect meaningful patterns of brain activity. Another option is to examine how much variance is explained in the data by one or all signal components (an example is shown in section 23.5).

There are several other pieces of information that could be extracted from PCA. For example, you could compute the variance accounted for by all components with below-threshold eigenvalues to estimate the “noise” of the global EEG response (this measure will likely be correlated but not redundant with the amount of variance explained by the first component).

Furthermore, there are PCA-based measures of the complexity of a system. Consider that a complex system contains several independent elements, and thus more components will account for a suprathreshold amount of variance; in contrast, relatively simple systems will have a small number of independent elements, and thus the eigenvalue function will decline more steeply. This can be quantified as the ratio of the first to the second eigenvalue (or the first to the average of all other suprathreshold eigenvalues) or as the exponential steepness of the decrease in eigenvalues. You can also quantify the “PCA coherence,” defined as the ratio of the squared first eigenvalue to the sum of squared eigenvalues (Mitra and Pesaran 1999) (this is also correlated with the percentage variance of the first component).

23.4 Rotating PCA Solutions

PCA as presented so far forces all components to be orthogonal to each other. This might be too strong an assumption; it is reasonable to expect that topographical patterns of covariance are intercorrelated.

Components can be “rotated” to allow them to be correlated (i.e., oblique) or to allow them to capture additional variance (that is, components might not necessarily optimally capture variance due to the orthogonality constraint). Rotation involves selecting a number of significant components and applying iterative algorithms to further minimize variance along the selected dimensions. There are several methods for rotating principal components that make different assumptions about whether and how components can be correlated. Note that a rotated PCA solution will change if you rotate a different number of components, whereas the unrotated solution will always be the same.

Whether you should rotate the PCA solution depends largely on the reason for performing the PCA. If you want to isolate overlapping ERP components, for example, rotation will facilitate a cleaner interpretation of the components (Dien 2012; Dien, Beal, and Berg 2005). To rotate PCA solutions, you can try the PCA toolbox for Matlab (Dien 2010), which is designed specifically for handling EEG data. If the goal is to try to unmix the disparate sources measured by EEG, ICA might provide a better solution. The differences between PCA and ICA are discussed in section 23.8.

23.5 Time-Resolved PCA

So far, the covariance matrix used to compute the PCA was made from all time points. It may seem inappropriate to include all time points into the covariance matrix, particularly

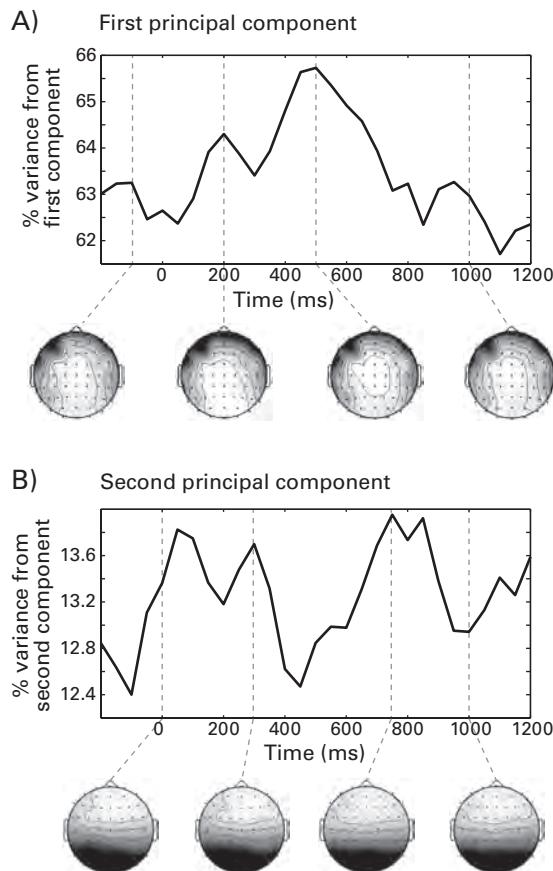
the buffer zones (for edge artifacts) or intertrial intervals. Furthermore, different time periods within the trial might be associated with different neuroelectrical modes or covariance configurations. Thus, PCA can be performed over successive time windows. Figure 23.6 shows an example of PCA results from the first and second components resulting from repeated PCAs over time. The PCA was performed on covariance matrices defined in 400-ms time windows centered from -200 to +1200 ms relative to stimulus onset. The percentage of variance accounted for by the first component was maximal at around 500 ms, and the topographical representation of this first component also shifted from having a spatial peak over parietal/occipital electrodes at 200 ms, to a spatial peak over frontocentral electrodes at 500 ms. The second component, in contrast, had peaks early and late in the trial. These findings may suggest that one spatial processing mode dominated the EEG activity between 400 and 600 ms.

If you perform a time-resolved PCA, you will need to decide how much time to include when computing the covariance matrix. There is a trade-off between temporal specificity (shorter periods of time help isolate temporal dynamics, particularly temporally brief dynamics) and signal-to-noise ratio (longer periods of time produce more stable estimates of covariance). If you perform time-resolved PCA on frequency-band-specific activity, the width of the temporal window also depends on the center frequency. This is discussed further in section 23.6. In most cases, a time window of around 400 ms should be sufficient. Time windows less than 100 ms should be avoided unless you have a specific justification; such short windows may lead to low signal-to-noise covariance matrices. For resting-state data, you might improve the signal-to-noise ratio by segmenting the data into nonoverlapping windows of 1–2 s.

23.6 PCA with Time-Frequency Information

PCA can be combined with temporal bandpass filtering to highlight frequency-band-specific spatial features. This can be done by first bandpass filtering the time-domain signal (either via FIR filtering or via the real part of the result of wavelet convolution) and then following the same procedures as already outlined in this chapter. An example frequency-band-specific PCA result is shown in figure 23.7.

The time courses in figure 23.7 were generated by multiplying the PCA weights by the 12-Hz-filtered data. In theory, it is also possible to apply the 12-Hz-defined PCA weights to broadband data or to data filtered at a different frequency, but these time courses may be more difficult to interpret because the PCA weights are being applied to data other than what they were computed from.

**Figure 23.6**

Results of PCA performed over time. At each time point (spaced 50 ms apart), the percentages of variance accounted for by the first (panel A) and second (panel B) PCs are plotted on the *y*-axis. Topographical maps below each time course show the projection of that component onto the scalp electrodes at selected time points (grayscale is the same across time points but is different between the first and second components). If you want to compare directly the changes in percentage variance accounted for by the first and second components, the percentage change time courses could be normalized to the baseline time period.

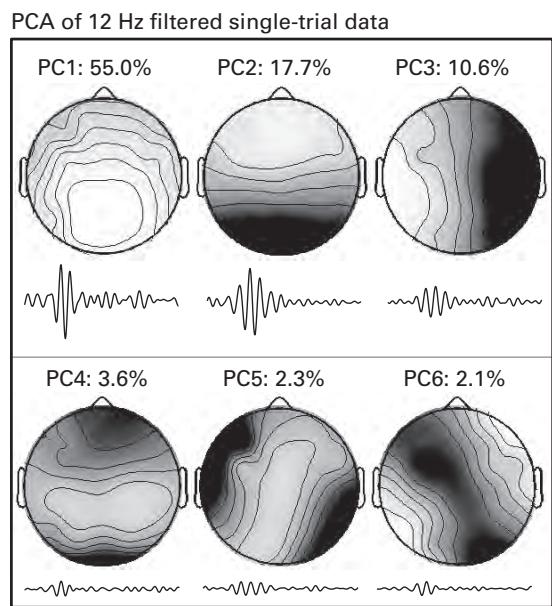


Figure 23.7

Same as figure 23.3B but for data filtered around 12 Hz.

23.7 PCA across Conditions

PCA can be used for testing condition differences in part because PCA can be used as a data reduction technique. Rather than comparing the electrode-level activity across conditions, you can compare the PC-defined activity across conditions. For condition data, you can compute the PCA based on either the covariance from all conditions (as shown in figure 23.8) or the covariance from each condition separately. Advantages of computing the covariance based on all trials from all conditions include that (1) the signal-to-noise ratio of the covariance is maximized because many trials contribute to the covariance matrix and that (2) it facilitates interpreting condition differences, because those differences cannot be due to topographical differences in PC weights. The main advantage of computing the covariance matrices separately for each condition is that the PCA results will have increased sensitivity to identifying condition differences because the weights are tuned to the properties of each condition.

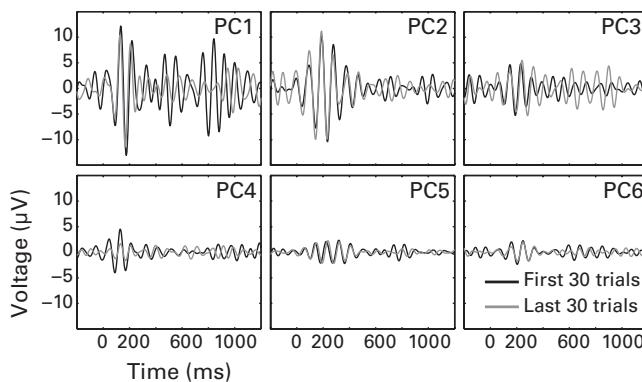


Figure 23.8

PCA was computed based on all trials (from the 12-Hz-filtered data shown in figure 23.7), and then the PCA weights were applied separately to the first 30 and the last 30 trials. Trials are ordered chronologically, so any differences could reflect task practice and/or fatigue effects. One qualitative difference, for example, is the enhancement of 12-Hz activity from around 600–1000 ms in the first 30 compared to the last 30 trials in PC1. This shift reverses in PC3.

23.8 Independent Components Analysis

Some aspects of independent components analysis (ICA) are conceptually or mathematically similar to PCA. Here is the key difference: PCA is used to decorrelate and reduce the dimensionality of a multivariate signal; ICA is used to demix independent sources that are embedded in multivariate signals. Thus, PCA decorrelates and ICA demixes.

There are other important differences between PCA and ICA. PCA is computed using only second-order statistics (variances), whereas ICA also uses higher-order statistics such as skew and kurtosis. ICA uses iterative methods based on minimizing cost functions (typically via mutual information) to adjust weights, whereas PCA does not iterate to adjust weights except during some rotation methods. PCA components are constrained to be orthogonal to each other (except in some methods of rotation), whereas ICA does not assume that sources are orthogonal to each other.

ICA is a standard and widely used technique in EEG data analyses. It is used to clean data prior to analyses (as discussed in section 8.1), and it is also used directly in analyses by analyzing component time courses instead of electrode time courses. Eeglab provides a range of ICA routines that are continuously being developed. You can learn more about ICA on the eeglab website (<http://sccn.ucsd.edu/eeglab/>) and the publications referenced there.

Some aspects of PCA and ICA can be difficult to compare across subjects because the topographical projections can differ across subjects. Picking components from each subject can be done based on visual inspection, matching to spatial or temporal templates, or via clustering procedures, as can be done in eeglab for ICA. Other uses of PCA can be easier to compare across subjects. For example, the analyses presented in figure 23.5 could easily be amalgamated over subjects because the important piece of information is the change in the percentage variance accounted for over time, not the topography of the maps.

23.9 Describing This Method in Your Methods Section

Because PCA can be used for many different purposes, it is important to state clearly the purpose of performing the components analysis. This will help readers interpret the results, and it will also help them evaluate the appropriateness of the analysis. Some PCA choices may be appropriate or inappropriate, depending on why you are performing the PCA (factor rotation is one example). Be clear about what data were used to compute the covariance matrices, whether filtering was applied, and what time windows were used. If you used software packages to perform the analyses, note which one was used and whether you changed any parameters from the default settings.

23.10 Exercises

1. Perform PCA on broadband data using two time windows, one before and one after trial onset (e.g., -500 to 0 ms and 100 to 600 ms).
2. Plot topographical maps and time courses of the first four components. To construct the PCA time courses, multiply the PCA weights defined by the pre- and posttrial time windows with the electrode time courses from the entire trial. Do you notice any differences in the topographical maps or time courses from before versus after stimulus onset? How would you interpret differences and/or similarities?
3. Repeat this exercise but after bandpass filtering in two different frequency bands. Make sure there are no edge artifacts in the pretrial time window (consider using reflection, if necessary, as described in figure 7.3). Justify your decision of frequency bands and time window width(s). Comment on any qualitative similarities and differences you observe between frequency bands and time windows and similarities and differences between the frequency-band-specific and broadband signal from the results obtained in the previous exercise.

24 Basics of Single-Dipole and Distributed-Source Imaging

Source-space imaging approaches, ranging from single-dipole fitting to distributed adaptive source imaging, define sets of weights per electrode such that the weighted sum of all electrodes is an estimate of activity emanating from some physical location in the brain (Michel et al. 2004). Two important concepts for source imaging are the forward solution and the inverse problem. The forward solution is an estimate of the topographical map that would result from activity in a dipole in a specific region of the brain with a specific orientation. The inverse problem is the estimate of what dipoles with what orientations and with what magnitudes could produce an observed topographical map. Figure 24.1 provides a graphical overview of these two concepts, and they are discussed further below.

Many source-imaging procedures can be conceptualized in terms of a regression or general linear model. The forward solution is analogous to the independent variables, and the observed electrode-level data are the dependent variables. The goal of the analysis is to define a set of weights, which are analogous to regression coefficients, that describe the relationship between the independent and dependent variables. Further analogous to regression, weights are created by minimizing the error between the fits of the forward model and the data. Differences among methods arise from different assumptions and approaches to estimating the relationship between putative brain generators and observed scalp activity.

24.1 The Forward Solution

Imagine you could place a small probe into the brain that would transmit a radio signal. When you activated the probe, what would that signal look like from the electrodes on the scalp? The answer to this question is the forward solution. It can be solved analytically without any EEG data, and its accuracy depends on a number of factors, including whether the head is modeled as a perfect sphere or as an anatomically accurate MRI-derived boundary

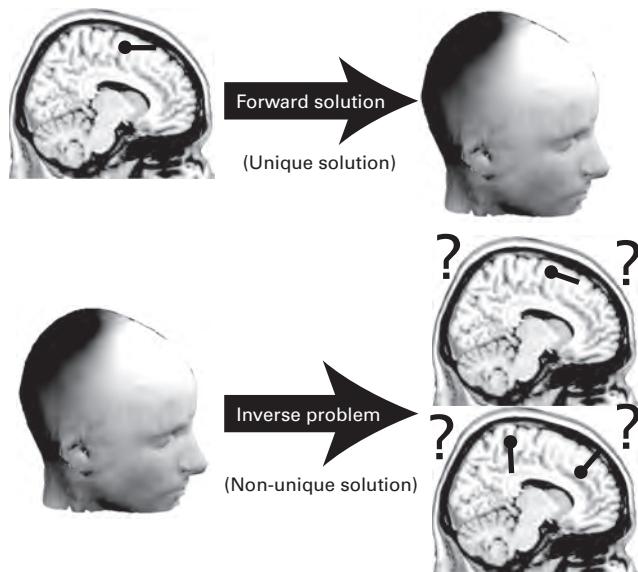


Figure 24.1

Graphical illustration of the forward solution and the inverse problem. The forward solution (What would activity at the scalp look like, given activation of one dipole in the brain?) has a unique solution, and can be obtained analytically without EEG data. The inverse problem (What configurations of brain sources could produce the observed pattern of topographical activity?) has no unique solution because many different configurations of dipoles could produce the same pattern of topographical activity. Shown here are two of an infinite number of possible brain-source dipole configurations that could produce the same topography.

element model and whether precise electrode positions are measured from each subject or whether standard electrode positions are assumed (Hallez et al. 2007).

This imaginary radio-signal-emitting probe does not broadcast in all directions equally but, rather, in a dipolar fashion. Thus, there can be a forward solution for different orientations in space. For some forward solutions, three fixed orientations (along the cardinal x -, y -, and z -axes) are modeled, as shown in figure 24.2. For other forward solutions only dipoles that are perpendicular (normal) to the cortex are modeled. With high confidence in (1) the precision of the measurements of electrode positions on the scalp, (2) the forward model including cortical folding, and (3) coregistration between the electrode locations and the MRI, constraining the analysis to cortex-perpendicular dipoles can improve localization precision. But with uncertainty in any or all of these factors, modeling of all three orientations may improve the accuracy of the reconstructed signal (Sekihara et al. 2004; Steinstrater et al.

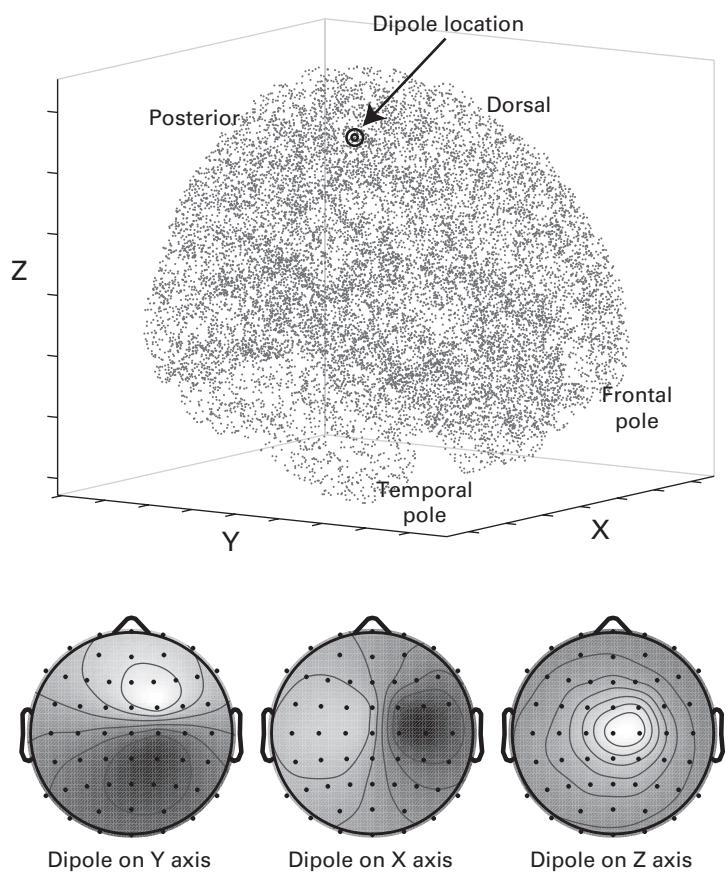


Figure 24.2

Example topographical maps expected from different dipole orientations: 15,028 brain sources were constructed from an MRI (each gray dot is a location), and the forward model for 64 EEG electrodes was computed for three dipole orientations from each source location. The forward model is a prediction about what the topographical map would look like if there were activity at only one dipole with one orientation. Topographical maps show the projection of the forward model from a selected location (see bull's-eye in brain) for dipoles oriented in the three cardinal directions. Using the analogy of a general linear model, this figure illustrates the independent variables; the problem of source-space estimation can be conceptualized as finding which of these independent variables, or which weighted combination of these variables, best accounts for the topographical data (analogous to the dependent variable). Single-dipole fitting involves choosing one or a small number of locations, and distributed source imaging involves estimating the contribution of each of hundreds or thousands of dipoles and orientations. (See figure 5.1 for an illustration of dipole orientations and their relation to the scalp-recorded activity.)

2010), although at the possible expense of a decrease in the spatial accuracy of the localization of that signal.

There are several methods to compute the forward solution that range in their complexity and anatomical realism, from single- or multisphere shell models that assume the head is a perfect sphere to subject-specific and MRI-defined realistic head models. Most M/EEG data analysis packages provide at least one method of computing the forward model. The OpenMEEG package (Gramfort et al. 2010) is implemented in several analysis programs (e.g., in the Matlab toolboxes fieldtrip and brainstorm) and is relatively fast and accurate.

24.2 The Inverse Problem

The inverse problem is the best answer to the question: Given my observed pattern of topographical activity, what are the most likely location(s), orientation(s), and magnitude(s) of brain source(s) that could have produced this topographical pattern of activity?

This problem can be difficult to answer for both theoretical and practical reasons. At a theoretical level, there is no unique solution. This means that there are many possible—and possibly very different—configurations of brain activity that can produce the same topographical map (see figures 24.1 and 24.3). At a practical level, all methods for estimating the inverse solution require several assumptions and/or involve parameter selections, and those parameter selections may impact the results. The rest of this chapter introduces you to some of the commonly used methods for estimating solutions to the inverse problem as well as to some of the assumptions and parameters involved in each method.

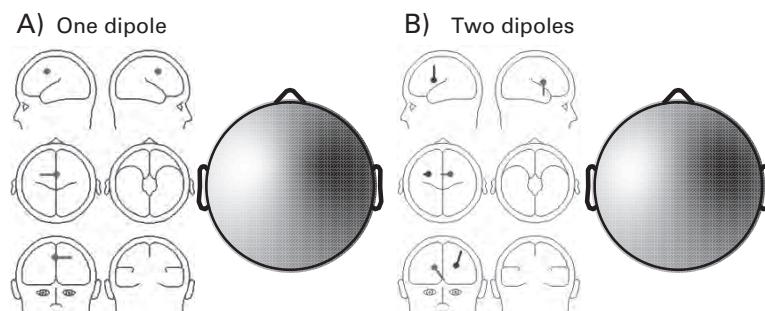


Figure 24.3

Results from the BESA dipole simulator (available for free download from the BESA website, www.besa.de), showing how different simulated dipoles can produce similar topographical results. Although a single- and two-dipole model is shown here, this figure illustrates a general feature of source-space imaging that applies to dipole as well as distributed-source imaging approaches.

24.3 Dipole Fitting

Dipole fitting involves estimating a single point or a small number of points in the brain that explain the maximal amount of topographical variance. Once the dipole location, orientation, and magnitude are estimated, a weight for each electrode can be determined such that the weighted sum of activity at all electrodes is the estimate of the dipole activity. Dipole fitting is typically done on time-domain data (ERPs) and is typically done on one time point or the average of a time window, although it is also possible to compute a dipole over a period of time (this is implemented, for example, in BESA: Scherg, Bast, and Berg 1999). Once weights are constructed, a dipole time series can be created, and analyses can be performed on this time series, including time-frequency decomposition. Realistic head and brain models for the forward solution are possible but not commonly used.

Single-dipole fitting can be used as a data reduction technique. Here the idea is to simplify the analyses by constructing one or a small number of dipole time series rather than dealing with dozens or hundreds of electrodes. In this case the goal of dipole fitting is not to localize the EEG activity per se but, rather, to use the framework of spatial filtering to combine activity from all electrodes in a data-driven way.

There is a practical limit to how many dipoles can be estimated because nonlinear search procedures for multiple dipoles may get “stuck” in local minima and therefore provide relatively poor solutions. But therein also lies the main disadvantage of fitting one or a small number of dipoles: interpreting dipole results rests on the assumption that that dipole is the only active source in the brain.

This may be a valid assumption for experiments that involve stimulation of primary sensory or motor cortices without concurrent cognitive or decision-making processes. However, many cognitive and perceptual processes rely on larger brain networks that can change dynamically over time and over frequency. The fact that a single dipole can sometimes account for over 80% or 90% of the topographical variance may result from fitting data that are highly temporally and spatially low-pass filtered at the single-subject and group levels. This does not invalidate the single-dipole-fitting approach, nor does it suggest that single-dipole results are uninterpretable. Rather, it is important to realize that if a dipole is fit to a very small part of the signal (one time point from the grand-averaged ERP over many subjects), that dipole will provide insights into a correspondingly small aspect of brain dynamics. For example, it might be difficult to find a single dipole that can account for over 90% of variance of single-trial time-frequency dynamics.

There are several software programs and Matlab toolboxes that allow you to perform dipole fitting with your data, including BESA, eeglab, and fieldtrip.

24.4 Nonadaptive Distributed-Source Imaging Methods

Imagine estimating activity at dipoles but instead of having a small number of dipoles whose locations, orientations, and magnitudes are estimated, you have thousands of dipoles all over the brain with fixed locations and orientations, and only their magnitudes are estimated. This is the idea of distributed-source imaging. With distributed-source imaging, the goal of the analysis is to define a set of electrode weights for each source location. That is, with 64 electrodes, 10,000 source locations, and three dipole orientations, you would have a $64 \times 3 \times 10,000$ weighting matrix (this would be $64 \times 10,000$ if dipoles are constrained to be perpendicular to the cortex). Multiplying the data from all electrodes by the weights for the 9876th element of that weighting matrix would give you the estimate of activity at the 9876th voxel in the brain.

Nonadaptive methods compute these electrode weights based on the electrode locations with respect to the brain. Thus, the weights are fixed and do not change over time or over frequency. LORETA and minimum-norm estimators are two commonly used nonadaptive inverse-source imaging techniques. Some algorithms are somewhere in between adaptive and nonadaptive. For example, some minimum-norm estimators use data variances to compute smoothing parameters when estimating weights.

These are several advantages of nonadaptive distributed source models. They can be computed quickly, can be applied to a single time point of data, and can produce brain maps that look like FMRI activation maps. There are relatively few parameters that you need to select that could affect the results. The main disadvantage relative to adaptive source-imaging approaches is that the weights for each electrode are not fine-tuned to the statistical properties of the data. Another disadvantage is the number of comparisons that can be computed and must be controlled for in statistical analyses. For example, if you estimate sources over time and frequency for 10,000 voxels and two experiment conditions, you could have over 100 million possible statistical comparisons. Controlling for multiple comparisons is further complicated by heterogeneous autocorrelations over space, time, and frequency (Barnes et al. 2004). This disadvantage also applies to adaptive distributed-source methods. Having clear hypotheses to constrain the analyses and results search space will help address the need for statistical control of multiple comparisons.

There are several software programs and Matlab toolboxes that allow you to perform nonadaptive distributed-source imaging with your data, including LORETA, MNE, brainstorm, spm8, fieldtrip, and nutmeg.

24.5 Adaptive Distributed-Source Imaging

Adaptive distributed-source imaging approaches differ from nonadaptive-source imaging approaches mainly in how the weights are computed. For adaptive distributed models, weights are computed not only according to physical electrode positions but also according to the data recorded from those electrodes. For many algorithms, weights are computed based on the covariance of the interelectrode activities. These covariance matrices can be computed in the same way as shown for PCA in the previous chapter, or they can be computed from cross-spectral densities (this is the frequency-domain analogue of the time-domain covariance matrix). Because the weights are computed from covariance matrices, they cannot be computed using a single time point but, rather, must be based on a time window. After the weights are computed, however, they can be applied to a single time point. Beamforming is the most common class of algorithms for adaptive distributed-source imaging that is used in EEG and MEG (Gross et al. 2001; Hillebrand et al. 2005; Van Veen et al. 1997).

The main advantage of adaptive spatial filters such as beamforming is, as the name suggests, that the weights are adapted to the data. This means that the weights can change over time, frequency, condition, and subject, according to changes in patterns of covariance in the data. This provides increased sensitivity for detecting subtle features of the results and provides a possible advantage over electrode-level recordings. For example, an effect with low amplitude might not be detected by electrode-level statistics, but if that low-amplitude activity is spatially coherent in a manner consistent with a certain brain source, an adaptive spatial filter may isolate this activity pattern and thus make it more visible in the data. Simulation studies suggest that beamforming provides results with higher accuracy compared to other source-imaging methods and that beamforming has the least amount of overestimation of the spatial extent of the activation (Dalal et al. 2008; Darvas et al. 2004; Kucukaltun-Yildirim, Pantazis, and Leahy 2006).

The main disadvantages of adaptive distributed source methods are the number of parameters that must be set and the consequences those parameters can have on the results. How much you have to consider these choices depends on what the goal of the analysis is, and on whether you write your own beamformers or use a toolbox or software package that uses predefined default values. Options include the algorithm to use, whether and which frequencies to analyze, whether to use time-domain covariances or frequency-domain cross-spectral densities, the length of the time window for computing the covariances and whether that window should change as a function of frequency, whether and how to regularize the covariance matrix, whether to compute weights based on all conditions or separately

for each condition, what to use as a normalization baseline, how many voxels to estimate and where they are located, what type of forward model to use (and, for EEG, what values to use for skull and scalp conductances), whether to fix dipole orientations with respect to the cortex or use three cardinal orientations, and, if using three orientations, how to estimate a voxel's activity from projections onto the three orientations, and so on. The number of parameters and choices should not dissuade you from applying adaptive spatial filters, but it is important to consider the parameters of the analyses and their possible influences on the results.

There are several software programs and Matlab toolboxes that allow you to perform adaptive distributed source imaging with your data, including spm8, fieldtrip, and nutmeg. A complete resource on adaptive spatial filters, including their mathematical bases, can be found in the book *Adaptive Spatial Filters for Electromagnetic Brain Imaging* (Sekihara and Nagarajan 2008).

24.6 Theoretical and Practical Limits of Spatial Precision and Resolution

Simulation studies suggest that impressively high spatial localization accuracy in source imaging—matching that of fMRI—can be obtained with very accurate electrode positions and forward models (Brookes et al. 2010; Murzin, Fuchs, and Kelso 2011). In practice, however, such high accuracy is not often obtained because of uncertainties regarding electrode positions, brain anatomy, electrode-to-MRI co-registration, head movement, and, for EEG, skull and scalp conductivity (Steinstrater et al. 2010). Thus, it is rare to see claims about fMRI-level anatomical accuracy of source localization results, and you should be cautious when hearing claims of functional-anatomical dissociations of less than a few centimeters based on results of source reconstruction. Such high spatial accuracy is possible but not common.

Striving for better and more realistic source reconstruction methods is an important and active research topic, although even standard head models can provide fairly accurate results (Fuchs et al. 2002). Source reconstruction voxels are typically 5–10 mm³, and spatial smoothing and normalization to a common brain space further reduces spatial precision. Although it is possible to have higher spatial precision, particularly in hypothesis-driven research, the “effective” spatial precision (taking smoothing into account) used in most source reconstruction studies is on the order of several centimeters (Barnes et al. 2004). On the other hand, retinotopic mapping of primary visual cortex and somatotopic mapping of primary sensory cortex are possible with beamforming (Brookes et al. 2010; Fisher et al. 2004).

It is often stated that MEG is superior to EEG for source localization accuracy. Simulation studies suggest that the localization accuracy of EEG can be as good as, or even superior to,

that of MEG (Murzin, Fuchs, and Kelso 2011; Steinstrater et al. 2010), in part because EEG is sensitive to radial and tangential dipoles, whereas MEG is maximally sensitive to tangential dipoles. However, improved accuracy for EEG source reconstruction requires many electrodes as well as precise electrode positions, head shapes, and brain-forward models (including accurate estimates of tissue conductivities). Such precision in measurement is not typically done for EEG, whereas it is more often done with MEG. Thus, in practice, brain localization accuracy of MEG is often better than brain localization accuracy of EEG. As EEG electrode technology improves and recording subject-specific electrode positions becomes more commonplace, the accuracy of EEG source reconstruction results will improve.

V Connectivity

25 Introduction to the Various Connectivity Analyses

How brain networks develop, function, and support cognition is a large and growing topic in many branches of neuroscience (Sporns 2011). Neural networks operate at multiple spatial and temporal scales (Varela et al. 2001), and considerable empirical research from multiple species, theories, and mathematical models over the past several decades points to oscillatory synchronization as being a key mechanism by which neural populations transmit information and form larger networks (Fries 2005; Salinas and Sejnowski 2001; Singer 1993). The purpose of this chapter is to provide an overview of the methods most commonly used to assess connectivity in cognitive electrophysiology and the issues involved in those analyses.

The term “connectivity” is used here to refer to any analysis for which more than one signal is considered at a time. This mostly refers to two signals from two different electrodes but can also refer to two signals from the same electrode or multiple signals from multiple electrodes. The term connectivity includes measures based on phase and on power using a variety of linear and nonlinear methods. These analyses often have disparate assumptions and utilize different aspects of the EEG signal but share the common goal of identifying brain connectivity; thus, the term connectivity is used to refer generally to all analyses that share this goal.

25.1 Why Only Two Sites (Bivariate Connectivity)?

Most but not all brain connectivity measures are bivariate, meaning that they involve interactions between only two brain regions/electrodes. Some brain connectivity measures may initially seem multivariate (one-to-all or all-to-all connectivity) but are in fact mass-bivariate measures because each step of the analysis involves connectivity between only a pair of electrodes.

Why are most connectivity measures bivariate? Perhaps this is related to our still-infantile view of brain interactions in which there are few detailed models of multinode networks

that are widely used in cognitive electrophysiology. This is possibly due to a paucity of approachable and intuitive mathematical/statistical analyses for quantifying larger and more complex networks. From a practical perspective, increasingly complex models of multinode brain interactions become increasingly difficult to conceptualize, and thus, it is easier to break them down into a set of simpler bivariate cases. For this reason, bivariate connectivity analyses are easier to implement, interpret, and test with established statistical procedures. Another possible reason for the abundance of bivariate connectivity methods is that the brain really works that way, and bivariate connections are the most relevant types of connections for many cognitive functions.

The focus in this book on bivariate connectivity methods is due to the practical reason listed above: bivariate methods are easier to understand, implement, visualize, and statistically quantify. This is in no way a rebuke of multivariate connectivity methods or an endorsement of the idea that only bivariate connections are relevant to brain function. The chapter on graph theory (chapter 31) provides an introduction to some multivariate network analyses.

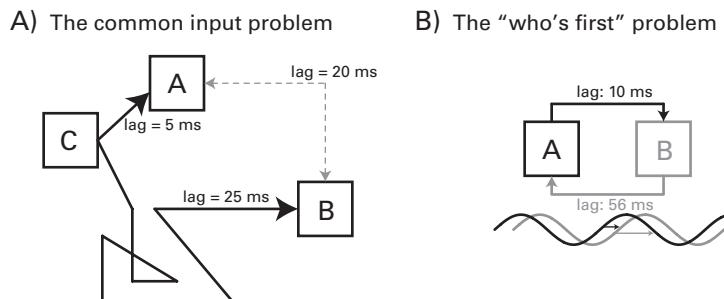
You should be aware that bivariate correlations can inflate or misrepresent estimates of relationships if the network structure is actually multivariate (an example of this is shown in figure 25.1A). This is particularly relevant for brain connectivity because the brain is a highly multivariate system. For task-related connectivity this potential inflation is mitigated somewhat by condition comparisons because inflated connectivity estimates should affect all conditions, and therefore, the inflation should subtract out during condition comparisons of connectivity.

25.2 Important Concepts Related to Bivariate Connectivity

The following five points should be kept in mind when you are interpreting results of bivariate connectivity analyses.

First, for many (though not all) types of connectivity analyses, the phase lag between the two electrodes is not taken into consideration. What matters is that the phase lag is consistent across time and/or trials. This means that connectivity between a pair of electrodes that are 0 ms, 10 ms, or 100 ms lagged from each other can be equally strongly synchronized. Most measures of connectivity provide information regarding the phase lag, although this can be difficult to interpret in some cases (see next paragraph and figure 25.1).

Second, a nonzero phase lag in connectivity does not necessarily imply a causal or directed relationship. Nonzero connectivity lag can be considered supporting evidence consistent with a causal or directed relationship, but this should be interpreted cautiously. For example,

**Figure 25.1**

Two scenarios to keep in mind when interpreting temporally lagged bivariate connectivity results.

if region A and region B each are entrained with region C, and if the $C \rightarrow A$ connection is faster than the $C \rightarrow B$ connection, there may appear to be phase-lagged connectivity between A and B without a causal or even a direct interaction between them (figure 25.1A). This example also highlights a danger of interpreting a bivariate correlation in a multivariate network. Another example: given a phase lag of 10 ms between regions A and B that are synchronous at 15 Hz, it may be difficult to determine whether A leads B by 10 ms, or whether B leads A by 56 ms (figure 25.1B). An additional complication of interpreting phase lags between electrodes is that phase (and thus, the phase lag between two electrodes) may be influenced by the relative dipole orientations of the sources measured by the electrodes.

Some measures such as Granger prediction (chapter 28) and the phase-slope-index (chapter 26) provide better evidence for directed connectivity compared to connectivity measures based on phase angle distributions (chapter 26) or power correlations (chapter 27). Nonetheless, because the brain is an enormously complex and dynamic system, because it is not possible to record from all components of this system at the same time, and because data contain noise, it is difficult to determine unambiguously whether a pattern of bivariate connectivity is truly causal. If claims about directionality or causality are important for your experiment and for interpreting your results, use directional methods such as Granger prediction and try to buttress your interpretation of causality or directionality with theory, known anatomical directional connectivity, previous relevant research, and, if possible, causal interference methods such as transcranial magnetic or electrical stimulation.

Third, phase-based and power-based measures of connectivity tend to reveal different patterns of results. This is not surprising from a mathematical perspective because phase and power are mostly independent measures. Phase and power likely reflect different

neurophysiological dynamics, with phase likely reflecting the timing of activity within a neural population and power likely reflecting the number of neurons or spatial extent of the neural population (chapter 21). However, the neurophysiological processes that contribute to phase-based versus power-based connectivity are not entirely clear, and too little cognitive electrophysiological research has been done to know when it is best to use phase-based versus power-based connectivity analyses, although phase-based connectivity is more commonly used in the literature. In general, however, phase-based connectivity analyses are useful for hypotheses concerning instantaneous connectivity (that is, at the same time, not necessary at zero phase lag), whereas power-based connectivity analyses are more robust to temporal offsets and jitters.

Fourth, a distinction can be made between functional and effective connectivity. Functional connectivity refers to linear or nonlinear covariation between fluctuations in activity recorded from distinct neural networks, and effective connectivity refers to a causal influence of activity in one neural network over activity in another neural network (Friston 1994). Thus, the distinction between functional and effective connectivity is analogous to a distinction between correlation and causation.

Fifth, many connectivity results can be confounded by volume conduction. Care must be taken to investigate and address this alternative hypothesis, and results must be interpreted cautiously in light of this potential confound and the ways to address it. This topic is discussed further in section 25.10.

25.3 Which Measure of Connectivity Should Be Used?

There are several classes of bivariate connectivity analyses (e.g., phase based vs. power based), several different analyses within each class, and several parameters and analysis options of each analysis. Each analysis has its advantages and limitations, and different measures are better suited for different purposes or assumptions about underlying neurocognitive processes. The following sections briefly describe the connectivity analyses discussed in this book and their advantages and limitations.

There is no correct or best connectivity method. Even when simulated data with known connectivity patterns are used, there may be no clear “winner” method that outperforms all other methods in all situations (Ansari-Asl et al. 2006; David, Cosmelli, and Friston 2004; Wendling et al. 2009). Some methods are more amenable to hypothesis testing, whereas others are more amenable to exploratory analyses. Some methods have a clear neurophysiological interpretation, whereas others are more based on computer science and engineering principles. Some methods are established and widely used, and other methods are novel and

more open to development and validation. Although this plethora of connectivity methods may seem to complicate cognitive electrophysiology research, it also provides increased flexibility for custom-tailoring analyses to specific research questions or goals.

If you would like to test for connectivity but do not know which measure you should use, first consider your hypotheses and expectations and which methods are best suited to address your research questions. You can also use the same connectivity methods used in publications that have a similar experiment design or similar kind of data. In general, it is a good idea to start with commonly used connectivity methods that are appropriate for your hypotheses and apply more sophisticated or less established connectivity measures only when your questions are not addressed by the less sophisticated methods or when the less sophisticated methods are difficult to interpret (e.g., if spurious connectivity due to artifacts cannot be ruled out).

25.4 Phase-Based Connectivity

Phase-based connectivity analyses (described in greater detail in chapter 26) rely on the distribution of phase angle differences between two electrodes, with the idea that when neural populations are functionally coupled, the timing of their oscillatory processes, as measured through phase, becomes synchronized. The mathematics of phase-based connectivity analyses is similar to that underlying ITPC (chapter 19).

There are several advantages to phase-based connectivity analyses. They are widely used in many experiments and across many species, and they have been used to examine network formation and network dynamics on many spatial and temporal scales. This is partly because phase-based connectivity analyses have a neurophysiological interpretation. These analyses are computationally fast, and thus, the results can be inspected quickly, and the analyses require few assumptions or parameter selections other than the parameters already selected for the time-frequency decomposition. Some phase-based analyses are also insensitive to time lag (others are sensitive to lag), meaning that as long as the temporal relationship between activity at two electrodes is consistent over time and/or trials, the phase lag will not affect the strength of the connectivity.

There are also a few disadvantages. Phase-based measures rely on precise temporal relationships, usually in the identical frequency band, and are therefore susceptible to temporal jitter or uncertainty in the precise timing of experiment events. These temporal uncertainties can have more significant effects at higher frequencies, as discussed in chapter 19. Second, phase-based measures do not provide compelling evidence for directionality for reasons outlined in section 25.2 and figure 25.1B.

25.5 Power-Based Connectivity

Power-based connectivity analyses (described in greater detail in chapter 27) involve correlating time-frequency power between two electrodes across time or over trials. These correlations can be computed between activity in the same or different frequencies and at the same or different time points.

Power-based connectivity measures provide ample opportunities for flexible analyses that can be custom-tailored toward testing specific hypotheses, and they can also be used for data-driven exploratory analyses. Power-based connectivity measures are arguably the most similar to connectivity measures often used in fMRI such as the psychophysiological interaction (which is based on correlating the BOLD time series between pairs of voxels), because the correlated fluctuations in activity are relatively slower, compared to phase-based connectivity measures. Power-based connectivity measures are also relatively insensitive to temporal jitter and uncertainty, as was shown in figure 19.9.

25.6 Granger Prediction

Granger prediction (also called Granger causality; described in greater detail in chapter 28) tests whether variance in one signal can be predicted from variance in another signal earlier in time. Granger prediction is similar to, and in some cases identical to, other autoregression-based estimates of directed connectivity, including the directed transfer function (Kaminski et al. 2001) and partial directed coherence.

The main advantages of Granger prediction are that it tests for and can dissociate directional connectivity, that is, $A \rightarrow B$ versus $B \rightarrow A$ connectivity. It can ignore simultaneous connectivity, which makes it less susceptible to volume conduction. There are several sophisticated analyses of multivariate networks that are based on Granger prediction, although usually in the literature (and in this book), the “basic” bivariate Granger prediction analyses are applied.

There are a few disadvantages of Granger prediction. It is sensitive to violations of stationarity, can be computationally time-consuming to perform, and doubles the number of results because each pair of electrodes contains two connectivity values (estimates of both $A \rightarrow B$ and $B \rightarrow A$ connectivity). If Granger prediction is used in an exploratory fashion, there will be twice the number of statistical comparisons that need to be controlled for, and thus, Granger prediction may become tedious for large-scale exploratory analyses.

25.7 Mutual Information

Mutual information is a simple but robust method of detecting shared information between two variables. It is computed based on the distributions of values within variables and the joint distribution of two (or more) variables (see it described in greater detail in chapter 29).

There are several advantages of mutual information analyses. First, mutual information can detect many kinds of relationships, including linear and nonlinear interactions that a correlation would fail to identify. For example, a circle has a correlation coefficient of zero but a mutual information value greater than zero. Second, mutual information has a long tradition of use and development in engineering and information technology. Finally, there are also several extensions for using mutual information and entropy to estimate system complexity or signal transmission integrity (e.g., channel-coding theorem).

There are also a few disadvantages of using mutual information for examining brain connectivity. First, mutual information does not provide information as to whether a relationship is linear or nonlinear, or positive or negative. Second, it is sensitive to the number of histogram bins. This is easy to control for but could be a significant confound if not addressed during analyses. Third, it can be computationally intensive, particularly if used for exploratory analyses. Finally, although it is a widely used signal-processing technique and may be particularly advantageous for quantifying nonlinear interactions, it does not have a clear neurophysiological interpretation.

25.8 Cross-Frequency Coupling

Cross-frequency coupling (described in greater detail in chapter 30) refers to a statistical relationship between activities in two different frequency bands. It can be used to infer local organization (when measured at a single electrode) and long-range connectivity (when activity from the two frequency bands is measured from different electrodes). Cross-frequency coupling has been observed in several species and has been linked to cognitive and perceptual processes (Canolty and Knight 2010) and disease states (Allen et al. 2011).

There are several advantages of cross-frequency coupling. It provides findings that can be linked across species and to computational models, and there are theories proposing a key role of cross-frequency coupling in information processing in the brain (e.g., Lisman 2005). Cross-frequency coupling might also help identify task-related high-frequency power, which may be difficult to identify with EEG in trial-averaging-based analyses (Nunez and Srinivasan 2010).

The main disadvantage (which can be an advantage if you enjoy exploratory data mining) is that there is a potentially huge search space ($\text{frequencies} \times \text{frequencies} \times \text{electrodes} \times \text{electrodes} \times \text{conditions} \times \text{time}$), which means that cross-frequency coupling analyses can be time-consuming and that there are many tests to control for during statistical evaluation. These can be minimized if you have hypotheses to help constrain the analyses.

25.9 Graph Theory

Graph theory (described in greater detail in chapter 31) is a mathematical framework for characterizing networks that can be represented as graphs containing nodes and vertices (for EEG connectivity, nodes and vertices are, respectively, electrodes and connectivity strengths). There are many analyses that fall under the umbrella term graph theory, and they are generally useful for providing summary information regarding large-scale or multivariate network dynamics.

There are several advantages of graph-theory-based analyses. They provide useful and often easy-to-interpret characterizations of multivariate networks. Because graph theory provides a general mathematical framework for conceptualizing networks, the same analyses can be applied to very different kinds of data, and high-level summary variables can be directly compared across, for example, EEG connectivity, diffusion MRI connectivity, and interneuron spike co-timing. Thus, graph-theory-based approaches can facilitate cross-methods and cross-species comparisons. Graph theory is arguably an underutilized analysis framework in cognitive electrophysiology that may provide novel insights into the electrophysiological network-level mechanisms of cognitive processes.

The main disadvantage of graph-theory-based measures is that they are often (although not always) used in exploratory data-mining analyses that lack a theoretical framework within which to understand the findings and link the results to other known functional properties of the brain. The reason this can be a disadvantage is that there are many graph-theory-based metrics that are used and relatively few applications, and it can be difficult to compare findings across studies that use different methods and that do not test specific hypotheses.

25.10 Potential Confound of Volume Conduction

Volume conduction is a potential confound that can lead to spurious connectivity results. There are two related concerns. First, sources in the brain generate large electromagnetic fields that are measured by more than one EEG electrode or MEG sensor, thus introducing

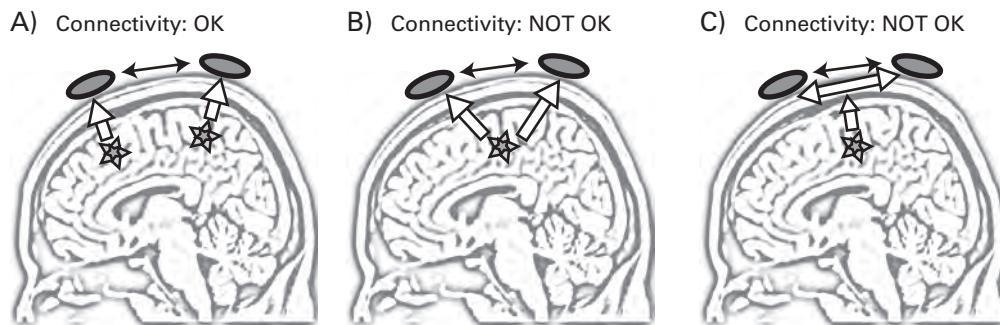


Figure 25.2

Illustration of the danger of volume conduction for interpreting interelectrode connectivity results. The black/gray rings represent electrodes, the black arrow between them illustrates measured connectivity, the stars represent neural sources in the brain, and the white arrows represent the path of electrical or magnetic activity from those sources. Ideally (panel A), each electrode measures only neural activity below the electrode, and thus, connectivity between two electrodes reflects connectivity between two physically distinct brain regions. Unfortunately, however, this situation cannot be assumed for EEG analyses: each electrode measures activity from overlapping brain regions (panel B), thus leading to the possibility that connectivity between two electrodes simply reflects those electrodes measuring activity from the same brain source. Furthermore, electrical fields can spread tangentially through the skull/scalp, causing further concern for EEG connectivity analyses (panel C).

spatial autocorrelation at the electrode level (figure 25.2B). This problem affects both EEG and MEG. The second issue is that electrical fields spread “laterally” through head tissues (skull, skin, etc.) and thus spread to neighboring electrodes (figure 22.1A and figure 25.2C). This problem affects EEG only; magnetic fields pass through these tissues undisturbed.

Volume conduction precludes an easy interpretation of brain localization based on electrode data, and it presents potential confounds for many but not all connectivity analyses. The confound is that connectivity between two electrodes could reflect true connectivity between different brain regions or could be due to those two electrodes measuring activity from the same brain sources (figure 25.2).

There are several options for addressing these potential confounds for connectivity analyses. No single option is optimal in all situations; instead, how you address this confound depends on the type of analysis performed and on the experimental design. It also depends on how hypothesis-driven versus exploratory your analyses are. Hypothesis-driven analyses typically involve a small number of tests. Thus, you might want to use analysis methods that have maximal sensitivity to detect true brain connectivity and then examine each effect for potential confounds. On the other hand, exploratory analyses typically involve a very large

number of tests, and it is impractical to examine whether each finding may be contaminated by volume conduction. Thus, you might want to use methods that are insensitive to volume conduction even though those methods may have decreased sensitivity to detect true brain connectivity.

If your connectivity finding is an artifact of volume conduction, you can expect the following pattern of results:

1. *Zero or π phase lag* Because volume-conducted activity is recorded instantaneously at multiple electrodes (within measurement capabilities), spurious connectivity due to volume conduction will have zero phase lag (or π phase lag if the electrodes are on opposite sides of the dipole). However, this is complicated by the fact that there is true zero-phase-lag connectivity in the brain (Chawla, Friston, and Lumer 2001; Roelfsema et al. 1997; Viriyopase et al. 2012). Thus, zero-lag or π -lag connectivity can reflect volume conduction, or it can reflect true zero-phase-lag brain connectivity.
2. *Very strong connectivity at neighboring electrodes and a decrease of connectivity strength with increasing interelectrode distance* The relationship between connectivity and interelectrode distance is somewhat complicated by cortical anatomy and dipole orientation, but in general, spurious connectivity due to volume conduction will be stronger with electrodes that are closer to each other, particularly for EEG. An example of this is shown in figures 22.7 and 22.8.
3. *Positive correlations* In the frequency and time-frequency domains, spurious connectivity due to volume conduction can only cause positive correlations. Time-domain connectivity would show negative correlations if the two electrodes are on opposite sides of the dipole.
4. *Positive correlations between connectivity and power in the same frequency band* If volume conduction is driving the connectivity, changes in power should correlate with changes in connectivity.

If your connectivity results are consistent with these four predictions, you should be concerned that those connectivity results are artifacts of volume conduction. On the other hand, if your results fail to conform to these predictions, it is unlikely that your connectivity results are due to volume conduction.

There are at least 10 approaches to addressing the potential contamination of volume conduction. Some of these approaches help minimize but do not necessarily completely eliminate volume conduction; thus, you may need to combine several of the following strategies.

1. Apply a spatial filter prior to computing connectivity, such as the surface Laplacian or source imaging. Most spatial filters will attenuate effects of volume conduction and therefore

render the data more appropriate for connectivity analyses. The surface Laplacian is a good spatial filter for electrode-level analyses, and distributed adaptive source solutions such as beamforming are good spatial filters for source-space analyses. However, spatial filtering does not guarantee that volume conduction is completely eliminated (particularly for neighboring electrodes or neighboring voxels), so you should still be cautious of connectivity results after application of a spatial filter, particularly for electrodes that are physically close to each other.

2. Examine only negative correlations in the frequency or time-frequency domains. Negative correlations in power at the same frequency band cannot be due to volume conduction. This option is not always a feasible approach because whether negative correlations can be expected depends on your task and on your hypotheses.
3. Test for temporally lagged connectivity rather than zero-phase connectivity. Because volume conduction is instantaneous, temporally lagged connectivity is less affected by volume conduction. Keep in mind, however, that temporally lagged connectivity does not necessarily eliminate volume conduction for all analyses because of temporal autocorrelation. Imagine that you have a signal comprising random numbers that you filter at 5 Hz, and then you correlate that signal with a 10-ms-lagged version of itself. You would still see a strong correlation between the “two” signals because at 5 Hz, the activity at one point in time is strongly correlated with activity 10 ms later due to temporal autocorrelation (see figure 25.3). The strength of temporal autocorrelation depends on the time-frequency decomposition characteristics and on the frequency (lower frequencies have stronger temporal autocorrelation). Thus, temporally lagged connectivity measures help minimize volume conduction, but they do not eliminate it, particularly when the signals are first bandpass filtered.
4. Test for condition differences in connectivity rather than single-condition effects. Some types of biases that are introduced by connectivity analyses will affect all conditions equally. Thus, subtracting connectivity between conditions (or, in some cases, between electrode pairs) will attenuate biases and thus may also attenuate the effect of volume conduction. One example of this is shown in figure 25.3. This figure shows that spurious connectivity resulting from bandpass filtering a signal made from random numbers is attenuated when connectivity results are compared across “conditions” (in this case, conditions are simulated simply as two different signals).
5. Test for a cross-frequency correlation (e.g., whether 6-Hz activity in one electrode correlates with 20-Hz activity in another electrode). If the 6-Hz and 20-Hz power activities are not correlated within each electrode, the correlation across electrodes cannot be due to volume conduction. Correlations across frequency bands should be interpreted cautiously if activity at those two frequency bands is correlated within one or both electrodes individually. For

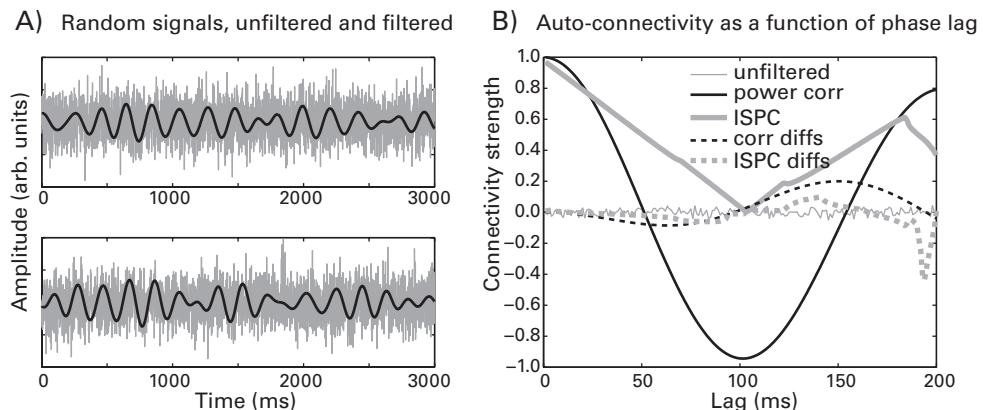


Figure 25.3

Illustration of how autocorrelation can be induced from bandpass filtering and how condition subtractions can attenuate the autocorrelation inflations. Panel A shows two signals generated by random numbers (gray lines) and the result of bandpass filtering each of those signals at 5 Hz (black lines; this was achieved by taking the real part of a convolution between the signals and a 5-Hz wavelet). Panel B shows autoconnectivity as a function of lag. Connectivity was computed between one signal and a lagged version of itself for 200 ms of lags (this corresponds to one cycle at 5 Hz). The *y*-axis refers to Pearson correlation ("power corr") or a measure of phase-based connectivity that is discussed in chapter 26 ("ISPC"). The thin gray line shows that before bandpass filtering, there is no autoconnectivity, whereas autoconnectivity is introduced by bandpass filtering. Subtracting the autoconnectivity values between signals ("diffs"; analogous to condition differences) attenuates the spurious connectivity. Some residual spurious connectivity after condition differences would be further attenuated when averaging is done over many trials.

example, if one electrode exhibits a correlated gamma power increase and an alpha power decrease, then negative correlations between alpha at that electrode and gamma at a different electrode could reflect volume conduction of the combined alpha/gamma effect.

6. Test for a statistical or qualitative dissociation between connectivity and power. For example, if connectivity between electrodes A and B increases but power simultaneously decreases, the connectivity cannot be due to volume conduction. A distinction between dynamics in power and dynamics in connectivity could be examined in several ways other than trial-averaged results. For example, you could test whether trial-to-trial fluctuations in connectivity and power covary with trial-to-trial fluctuations in behavior or stimulus properties. If connectivity correlates with behavior but power does not, the connectivity is thus decoupled from the power. Any dissociation between connectivity between electrodes and power at one or both electrodes provides evidence against volume conduction. These results should also be

interpreted with caution: a correlation between power and connectivity does not necessarily mean that the connectivity is due to volume conduction, but such a correspondence makes it more difficult to rule out the volume conduction alternative explanation.

7. Test whether the phase lag of connectivity between electrodes is significantly different from zero or π . Although zero-phase-lag connectivity can reflect true brain connectivity or volume-conducted activity, nonzero phase lag cannot be due to volume conduction. One limitation of this approach is that phase lags that are not zero but are close to zero may still be statistically indistinguishable from zero (section 26.10 provides a statistical test for phase angles).
8. For phase-based connectivity, you can use measures that are insensitive to volume conduction such as imaginary coherence, phase-lag index, weighted phase-lag index, or phase-slope-index.
9. For power-based connectivity, you can compute partial correlations between two electrodes holding constant a third electrode. This third electrode can be a neighbor of one of the electrodes. The idea is that the power time series of two neighboring electrodes are strongly correlated because of volume conduction; by computing partial correlations, shared variance that is mainly due to volume conduction with a neighboring electrode will be removed (section 27.4).
10. For power-based connectivity you can modify pairs of time series before calculating connectivity such that the coherent real parts (which include volume conduction effects) are removed, thereby removing any potentially volume-conducted signals (Hipp et al. 2012). This is complementary to using volume-conduction-independent measures because instead of ignoring potentially volume-conducted activity during the analysis, parts of the data that potentially contain volume-conducted signal are removed.

26 Phase-Based Connectivity

Many of the concepts and procedures for measuring phase-based connectivity are similar to those for ITPC, so understanding the material presented in chapter 19 will help you to understand the material in this chapter.

26.1 Terminology

There are several terms that are used in the literature to describe phase-based connectivity, including phase-locking value/statistic/factor, phase synchronization, and phase coherence. Arguably, these are suboptimal terms, partly because they are interpretations of results rather than descriptions of methods and partly because the same terms are used to indicate different methods (for example, the term phase-locking value can be used either for ITPC or for connectivity, and yet these two analyses have very different meanings and interpretations). “Phase correlation” is sometimes used, but this is a poor term because a correlation indicates a linear bivariate relationship that can take positive or negative values; phase-based connectivity measures are not linear, nor can they have negative values.

Here, the term intersite phase clustering (ISPC) is preferred. ISPC is a concise description of the method (clustering in polar space of phase angle differences between electrodes, voxels, or neurons) without any assumptions or interpretations regarding putative neurophysiological mechanisms being made. Other analysis-specific terms (e.g., phase-lag index) are introduced later in this chapter.

26.2 ISPC over Time

ISPC is similar to ITPC presented in chapter 19. Recall from equation 19.1 that ITPC is defined as the length of the average vector of many unit vectors whose phase angles are obtained by a point in complex space resulting from the convolution between a complex wavelet and the

data (or from applying the Hilbert transform to bandpass-filtered data). ISPC works in a similar fashion, but rather than taking the average of phase angles, you take the average of phase angle *differences* between electrodes over time (Lachaux et al. 2000; Mormann et al. 2000).

$$\text{ISPC}_f = \left| n^{-1} \sum_{t=1}^n e^{i(\phi_{xt} - \phi_{yt})} \right| \quad (26.1)$$

in which n is the number of time points, and ϕ_x and ϕ_y are phase angles from electrodes x and y at frequency f . The only difference between equation 26.1 and equation 19.1 is that equation 26.1 has a subtraction of phase angles from two electrodes rather than phase angles from one electrode. When implementing this equation in Matlab, remember to multiply the difference of the phase angles by the imaginary operator and not only the first phase angle. That is, write `exp(1i*(angles1-angles2))` and not `exp(1i*angles1-angles2)`.

To illustrate this method, it is useful to compute ISPC between two neighboring electrodes without applying any spatial filters. Here, electrodes Pz and P1 are used. Much of the apparent connectivity between these electrodes is due to volume conduction and therefore should not be interpreted in terms of connectivity between distinct brain regions. Nonetheless, using similar time series provides a visually intuitive learning exercise. Figure 26.1A shows bandpass-filtered time series from these two electrodes and their phase angles, and figure 26.1B shows the amplitude and phase angle differences between these two electrodes.

Figure 26.1C shows the individual phase angles from each electrode. You can see that the phase angles from the two electrodes are uniformly distributed throughout polar space. This is not surprising because there are many cycles at this frequency band in the time segment shown. However, when you subtract the phase angles from the two electrodes at each time point (figure 26.1D), you can see that the phase angle differences are nonuniformly distributed in polar space. In fact, they are strongly clustered around zero, which, in this case, reflects that much of the apparent connectivity is due to volume conduction.

Figure 26.2 shows that the actual phase lag between electrodes does not matter. What matters is that the phase lag is consistent across time. Thus, activity measured by pairs of electrodes that are 0 ms, 10 ms, or 100 ms lagged from each other can be equally strongly synchronized when measuring synchrony through ISPC.

ISPC (along with many other phase-based connectivity measures) is symmetric (that is, nondirectional). This means that ISPC $A \rightarrow B$ is the same as ISPC $B \rightarrow A$. You can confirm this yourself by swapping the order of the electrodes in the online Matlab code to see that it has no effect on ISPC.

So far, ISPC was computed over a window of time from one trial. It is likely that you will want to examine changes in ISPC over time. There are two options for examining task-related

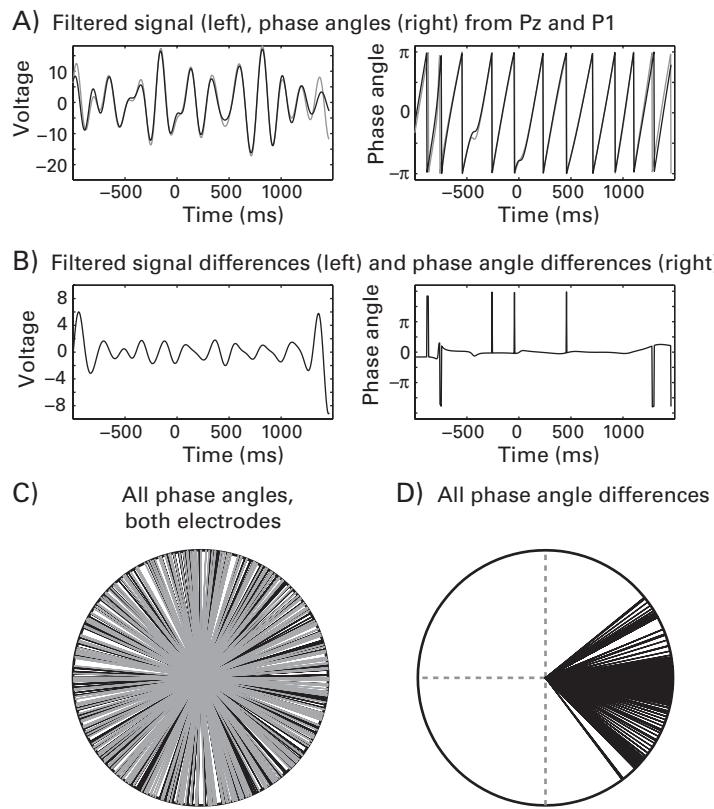
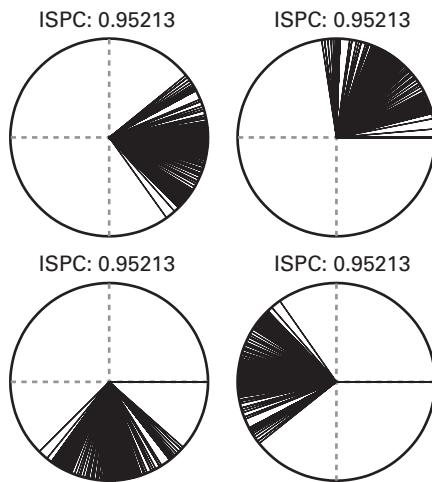
**Figure 26.1**

Illustration of phase angle differences between two electrodes over time. ISPC is defined as the length of the average vector of phase angle difference vectors (the vectors in panel D). Amplitude information, shown in the left plots of panels A and B, is not taken into consideration in this analysis. The apparent large jumps in the right-hand plot of panel B are due to small changes between -2π and $+2\pi$. This figure is actually the final frame of a movie that shows phase angle differences collecting over time. The online Matlab code will produce the movie.

**Figure 26.2**

ISPC does not depend on phase values themselves, only the clustering of phase angle differences.

or time-varying ISPC. One option is to compute ISPC in sliding time segments, analogous to the way the FFT was computed in sliding time segments in the short-time FFT method (chapter 15). After ISPC has been computed in sliding time segments within a trial, the ISPC would be computed for each trial, and then the results can be averaged across trials (or correlated with a trial-varying behavior or experiment parameter; more on this later). Hereafter, this approach is called *ISPC-time*.

The selection of time segment length is partly related to the frequency and partly related to the task design. There is a trade-off between signal-to-noise ratio and temporal precision: longer time segments include more oscillation cycles and therefore will provide more robust and higher signal-to-noise estimates of ISPC-time. However, this comes at the expense of decreased temporal precision for time-varying and task-related modulations of ISPC-time. A segment length of at least three cycles (i.e., 1.5 before and after each time point) is recommended. If you have a task with long events such as a working memory delay or long stimulus presentation times, you could use more cycles. Because higher-frequency activity tends to have lower signal-to-noise ratio, and because higher-frequency cycles are, by definition, faster, you can have the number of cycles increase with increasing frequency, similar to how the number of cycles increases when creating Morlet wavelets. This option will modify the balance between temporal precision and signal-to-noise as a function of frequency. Thus, for example, you could use time segments corresponding to three to seven cycles for frequencies ranging from 4 to 60 Hz. It is also possible to keep the segment length fixed (e.g., at

300 ms), but this results in a variable number of cycles over frequencies. Figure 26.3 (plate 15) illustrates the effects of window segment length on ISPC-time results.

The second option for examining task-related or time-varying ISPC is to compute ISPC over trials instead of over time, which is discussed below.

26.3 ISPC-Trials

ISPC over trials (hereafter called *ISPC-trials*; the term “ISPC” is used when discussing both ISPC-time and ISPC-trials) is a similar but alternative method for assessing task-related

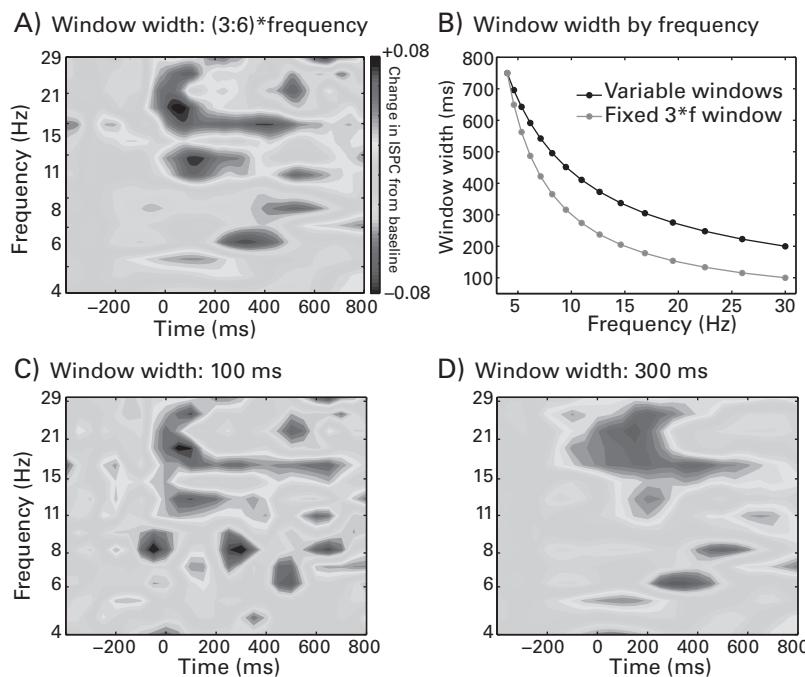


Figure 26.3 (plate 15)

ISPC-time between electrodes Fz and O1 using different time segment lengths. The results shown in panel A were obtained using a variable time segment length of three cycles at the lowest frequency (4 Hz) to six cycles at the highest frequency (30 Hz). The results in panels C and D were obtained using fixed time segment lengths for all frequencies. ISPC-time was computed over sliding time segments for each trial and then averaged across trials. All time-frequency plots have the same color scale (shown in panel A). ISPC strength from a pretrial baseline period of -400 to -200 ms was subtracted to highlight task-related effects. Panel B shows the length of the time segments used at each frequency when creating panel A (black line) and what the length of the time segments would be if a fixed ratio of three times the frequency ($3 \times \text{frequency}$ or $3*f$) had been used (gray line).

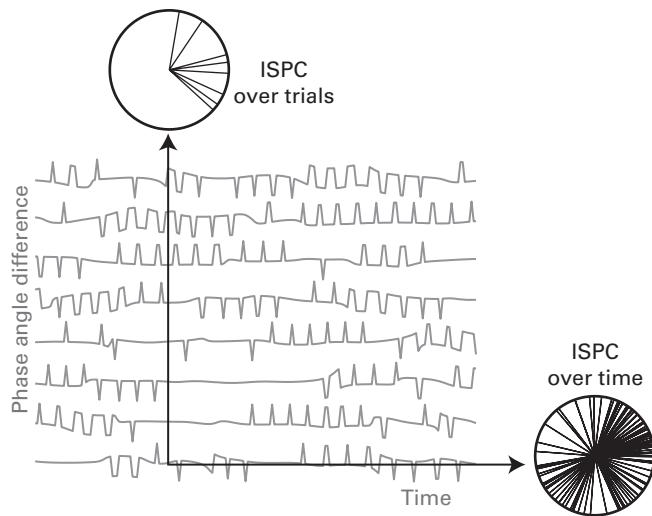


Figure 26.4

The conceptual difference between ISPC-time and ISPC-trials. Gray lines show time courses of phase angle differences over time for eight trials (each line is a trial). ISPC-time is computed from phase angle difference distributions over many time points for one trial, whereas ISPC-trials is computed from phase angle difference distributions at one time point over trials. Many connectivity analyses presented in this and subsequent chapters can be done over time or over trials.

phase-based connectivity. With ISPC-trials the assumption is slightly different than phase angle differences being clustered over time: the assumption is that connectivity produces a clustering of phase values at each time-frequency point relative to an experiment event over repeated trials (Lachaux et al. 1999). This is a subtle but important distinction. With ISPC-trials, the distribution of phase angle differences is generated at each time point over trials, whereas with ISPC-time, that distribution is computed at each trial over time points (figure 26.4). The distinction between computing connectivity over trials versus over time has implications both for analyses and for interpretations and will come up several times over the next few chapters; most connectivity analyses discussed in this book can be computed over time or over trials.

ISPC-trials add a stronger constraint, which is that phase angle differences must be consistent over trials. With ISPC-time, the preferred phase angle difference can be different on each trial as long as the amount of clustering is similar on each trial. With ISPC-trials, the preferred phase angle difference must be similar at each time-frequency point over trials. For example, if the four panels in figure 26.2 corresponded to phase angle differences over time

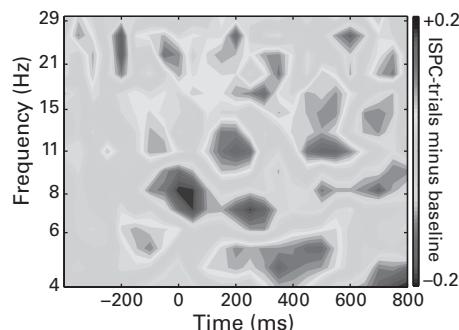


Figure 26.5 (plate 16)

Baseline-subtracted ISPC-trials using the same data as shown in figure 26.3 (plate 15).

for each of four different trials, ISPC-time would be very high, whereas ISPC-trials would be very low.

The equation for ISPC-trials is the same as equation 26.1, except that the t now refers to trial instead of time point, and n refers to the number of trials instead of the number of time points. An example result of ISPC-trials is shown in figure 26.5 (plate 16).

Should you compute ISPC-time or ISPC-trials? There are two advantages of ISPC-time. First, it is less sensitive to trial-to-trial jitter of experiment event timing (see figure 19.9) because even if the event timing is jittered, the timing of activity at different EEG electrodes will never be jittered relative to the timing of activity at other electrodes. Second, ISPC-time is insensitive to phase angle differences being different on different trials and can therefore be conceptualized as measuring total (non-phase-locked and phase-locked) rather than only phase-locked connectivity. For both of these reasons, ISPC-time is better suited for detecting high-frequency connectivity. The disadvantage of ISPC-time is that it requires longer within-trial time segments. This reduces temporal precision even more than was already done by time-frequency decomposition (as shown in figure 26.3B, plate 15). It may also be problematic if time segments extend into other trial events, the intertrial interval, or the trial buffer zones, which may contain edge artifacts. This problem becomes worse at lower frequencies, for which a window of three cycles could mean 1 s or longer. Thus, for lower frequencies, ISPC-time is better suited for experiments with long task events. In theory, it is possible to compute ISPC over both time and trials by concatenating all phase angle pairs over the time segment from all trials and computing one ISPC value. However, that may be a suboptimal strategy: it assumes that the preferred phase angle difference is the same over trials (which is a requirement of ISPC-trials but not ISPC-time) but still has the poor temporal precision of ISPC-time.

There are two advantages of ISPC-trials. First, it provides stronger evidence compared to ISPC-time for task-related modulations in connectivity because the connectivity must be in the same phase configuration on each trial. Second, ISPC-trials is computed at each time point individually, which means there is no loss of temporal precision beyond what was introduced by wavelet convolution or bandpass filtering. If you have hypotheses concerning the time course of connectivity over tens to hundreds of milliseconds, ISPC-trials should be preferred over ISPC-time. The two disadvantages of ISPC-trials are that it is sensitive to slight jitters in the uncertainty of the timing of experiment events, as shown in figure 19.9, and that it may fail to detect connectivity that occurs on each trial but with different phase values over trials. ISPC-trials cannot be performed on resting-state data.

ISPC-trials can be performed on results from any time-frequency decomposition method that provides phase values, including complex wavelet convolution, filter-Hilbert, short-time FFT, and multitaper. ISPC-time can be performed only on results from time-frequency decomposition methods that provide phase angle time series with the same temporal resolution as the original data. The two methods presented in this book that can be used for ISPC-time are complex wavelet convolution and filter-Hilbert.

26.4 ISPC and the Number of Trials

ISPC-trials, like ITPC, is sensitive to the number of trials used in the analyses. Figure 26.6 illustrates this point by showing ISPC-trials as a function of the number of trials included in the analysis. This figure is similar to figure 19.6A. More discussion of the relationship between the number of trials and ITPC was presented in section 19.3. Figure 26.6 also illustrates a strategy for determining whether you have enough trials for stable estimates of ISPC-trials within a condition. You can generate a figure like this by combining all trials from all conditions and noting the number of trials at which the lines appear to flatten for the frequencies you want to analyze. If the number of trials per condition exceeds this number, you probably have enough trials in that condition for a robust estimate of ISPC-trials. Based on the data used in figure 26.6, around 40 trials per condition should lead to stable estimates of ISPC-trials for most frequency bands. As with figure 19.6A, the results presented in figure 26.6 are based on one pair of electrodes and one subject and therefore should not be interpreted to indicate that 40 trials are always sufficient for ISPC-trials.

Three other ways of dealing with potentially low trial count are to compute ISPC_Z (Rayleigh's Z , see equation 19.2), to use other phase-based connectivity measures that are less sensitive to the number of data points, or to apply statistical corrections to account

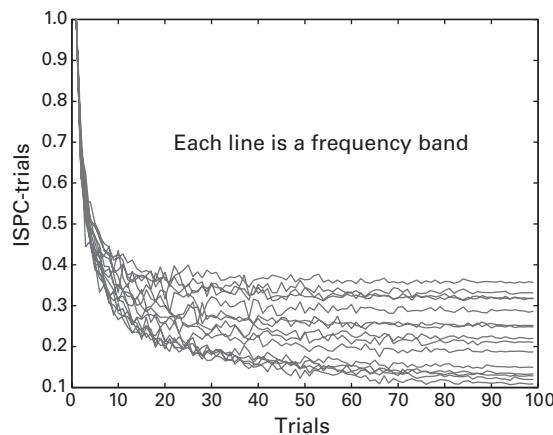


Figure 26.6

Relationship between ISPC-trials (at 300 ms poststimulus) and the number of trials used in the analysis. Results are averaged over 50 iterations of randomly selected trials.

for differences in trial count (Vinck et al. 2010). However, as discussed in chapters 7 and 19, the best strategy is to design the experiment such that you will have a sufficient number of trials in the experiment and a similar number of trials across the conditions you want to compare.

Note that ISPC-time is less sensitive to trial count than is ISPC-trials. This is because the inflation of ISPC-trials is due to having a small number of data points per se, not specifically a small number of trials. Thus, if you compute ISPC-time, you may have several hundred data points per ISPC-time value, even if there are only 25 trials. For this reason, if you are concerned that you have too few trials for ISPC-trials, it is safer to compute and interpret ISPC-time, even at the expense of reduced temporal precision.

26.5 Relation between ISPC and Power

As mentioned several times in this book, power and phase are mostly independent, except that with zero power, phase cannot be estimated. Section 19.6 showed that signal amplitude has minimal effect on ITPC except when power was zero (and this was not a big problem if power was only physically zero and the data were bandpass filtered). A related example is shown here. Of two electrodes between which ISPC-trials was computed, activity from one electrode was multiplied by 0.0001 to dampen its power. This had no appreciable effect on

the resulting estimate of ISPC, as seen in figure 26.7. Dampening activity from both electrodes or multiplying one or both time series by 10,000 also had no effect. Furthermore, the scatter plots in figure 26.7D show that in this dataset there is no clear relationship between ISPC-trials and power over time. This example suggests that the estimation of phase is not directly linked to power. As with ITPC in section 19.6 and figure 19.10, this illustration does not mean that you should be unconcerned about the possible effects of power on phase-based measures, but it does suggest that within reasonable ranges, and as long as power is not zero, phase-based measures of connectivity can be interpreted independent of power.

26.6 Weighted ISPC-Trials

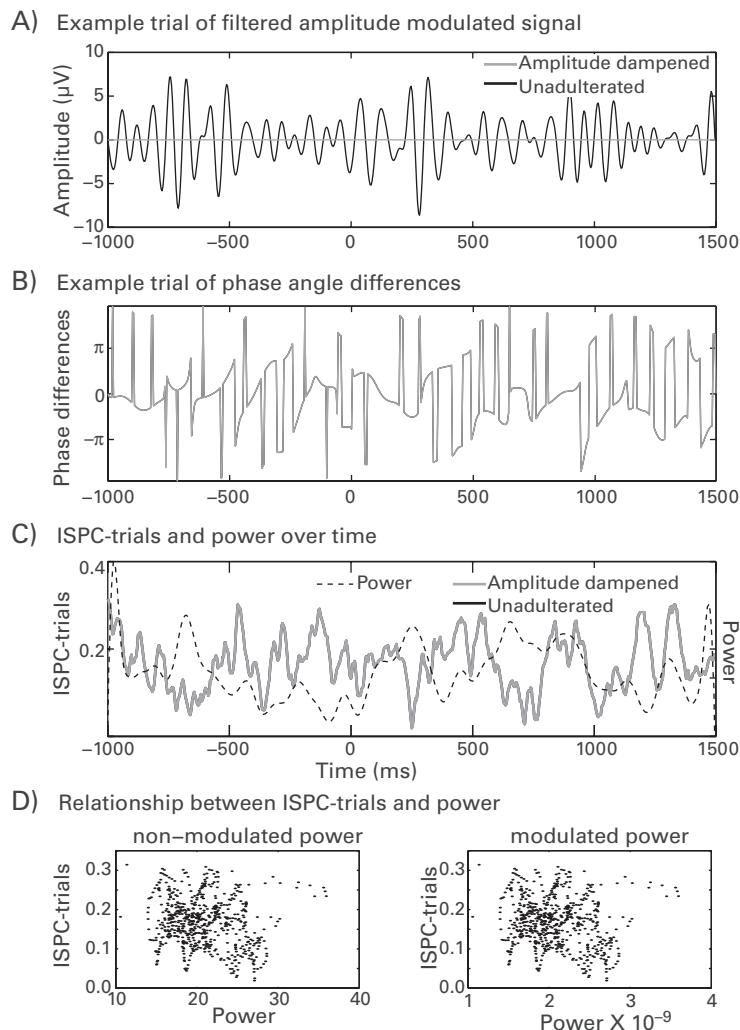
Weighted ISPC-trials will allow you to test for a statistical relationship between ISPC-trials and trial-varying behavior or experiment variables such as reaction time or stimulus features (Cohen and Cavanagh 2011). The procedures for wISPC-trials are identical to those for wITPC (section 19.7), except that phase angle differences between two electrodes are used instead of the phase angles from one electrode. The wISPC-trials can also help dissociate connectivity from volume conduction: if the wISPC-trials is significant, but there is no analogous correlation between power and that same trial-varying variable, the trial-varying phase angle differences are unlikely to be due to volume-conducted activity.

Correlating ISPC-time with a trial-varying variable does not require the same procedures as wITPC-trials because the connectivity value can already be computed on each trial. Thus, you can simply correlate (preferably using Spearman's correlation, because ISPC values are nonnormally distributed) the ISPC value at each time-frequency point over trials with the trial-varying variable.

26.7 Spectral Coherence (Magnitude-Squared Coherence)

Spectral coherence is similar to ISPC, but the phase values are weighted by power values. In many cases spectral coherence and ISPC will provide similar results. However, because spectral coherence also incorporates power information, results from spectral coherence are likely to be influenced by strong increases or decreases in power. For example, if connectivity increases but power simultaneously decreases, spectral coherence may provide biased results (Lachaux et al. 1999). Typically, you will see the equation for spectral coherence as follows:

$$Coher_{xy} = \left| \frac{S_{xy}}{S_{xx}S_{yy}} \right| \quad (26.2)$$

**Figure 26.7**

ISPC-trials was computed between electrodes Fz and O1, filtered from 10–20 Hz, with and without multiplying one of the time series by 0.0001 to dampen its amplitude (see panel A; the time course is not zero but too small to be visible on this scale). In the example phase angle differences from one trial in panel B and in the ISPC-trials in panel C, the time courses for the modulated and unmodulated data are overlapping and thus not distinguishable in the figure. The power time course from electrode Fz is overlaid, illustrating no relationship between power in one electrode and connectivity between electrodes (this lack of clear relationship holds for O1 power as well). The scatter plots in panel D show ISPC by power of one of the electrodes (each dot is a time point).

S_{xy} is the cross-spectral density between activities at electrodes X and Y, and S_{xx} and S_{yy} are the autospectral densities for electrodes X and Y (this is explained further below). Sometimes, the numerator is squared, which is called magnitude-squared coherence.

For consistency within this book, it is useful to rewrite and explain spectral coherence using a Euler-like format, which is equivalent to equation 26.2 but might be more intuitive in the context of the information you learned in chapters 11–13.

$$C_{xy} = \left| n^{-1} \sum_{t=1}^n |m_{tx}| |m_{ty}| e^{i\phi_{xy}} \right|^2 \quad (26.3)$$

m_x and m_y are the analytic signals X and Y (the vertical bars surrounding them indicate to take the magnitude of the analytic signals, e.g., via the Matlab function `abs`), ϕ_{xy} is the phase angle difference between electrodes X and Y, and t refers to trials or time points, depending on whether coherence is computed over time or over trials. This formula may look complicated, but if you break it up into chunks, you will see that it comprises concepts you have already learned. First consider the $e^{i\phi}$ part, which is Euler's formula. These are the phase angle differences between electrodes X and Y. Those phase angle differences are modulated (or weighted) by the signal magnitudes (similar to how wISPC was weighted by a trial-varying variable). The sum and n^{-1} indicates the average over magnitude-modulated phase values. Finally, the length of the complex result (that is, the length of the average vector) is taken. You can see that with the magnitude modulators dropped, equation 26.3 is similar to equation 26.1. You might notice that a problem with equation 26.3 is that the result scales with power, and power changes over frequency, time, task events, and so on. This is why spectral coherence is then normalized by power, by treating equation 26.3 as the numerator and the combined power as the denominator, as in equation 26.4.

$$Coher_{xy} = \frac{C_{xy}}{\left(n^{-1} \sum_{t=1}^n |m_{tx}|^2 \right) \left(n^{-1} \sum_{t=1}^n |m_{ty}|^2 \right)} \quad (26.4)$$

The numerator is equation 26.3. The denominator is simply the product of the average power values from electrodes X and Y (the averaging is done over trials or time points, depending on whether you are computing coherence over time or over trials). This normalization factor accounts for signal magnitudes and thus puts coherence on a scale from 0 to 1, with 1 being complete coherence and 0 being complete independence. These coherence values are thus interpreted in the same way you would interpret ISPC values. Note that although the equation is normalized by the total power, individual phase angle vectors are still weighted

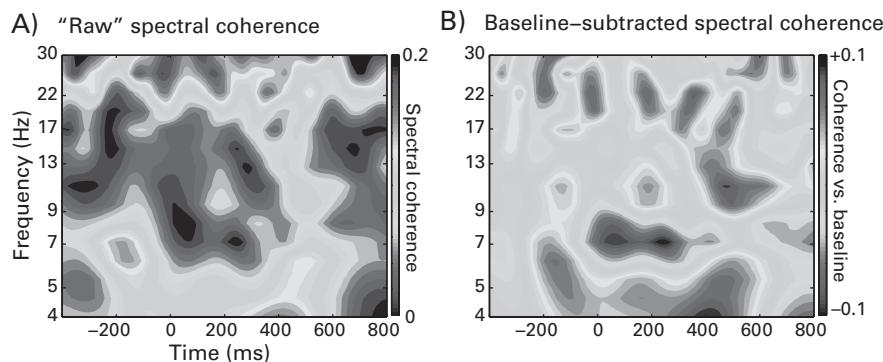


Figure 26.8 (plate 17)

Time-frequency spectral coherence over trials between Fz and O1, shown in “raw” units (no baseline subtraction; panel A) and after linear frequency band-specific baseline subtraction (panel B). These results can be compared with those presented in figure 26.5 (plate 16).

by power, and the mapping between power and phase angle can be nonrandom if particular regions of phase space are associated with relatively increased or decreased power. This may occur, for example, after stimulus onset, when there are both robust changes in power and simultaneous nonuniform phase angle distributions. This is why spectral coherence results can be influenced by power despite being normalized by the total power in the denominator. Figure 26.8 (plate 17) shows an example of spectral coherence computed over trials.

A brief aside on Matlab code: remember from section 13.5 that multiplying an analytic signal by its conjugate is the same as squaring its absolute value but is about twice as fast. Translated into Matlab code, this means that `sig1.*conj(sig1)` is the same thing as `abs(sig1).^2`, where `sig1` is an analytic signal (the result of complex wavelet convolution or filter-Hilbert). To compute the cross-spectral density, you can use `sig1.*conj(sig2)` (this is S_{xy} in equation 26.2). This is equivalent to and faster than, although perhaps less intuitive than, the Euler notation: `abs(sig1).*abs(sig2).*exp(1i*(angle(sig1)-angle(sig2)))`.

Equations 26.4 and 26.2 are for the magnitude of coherence, which quantifies the strength of the coherence. If you do not square and take the absolute value (magnitude) of the numerator, the resulting coherence is a complex number, and you can therefore also extract the phase value, which may be useful for testing the phase lag between the two electrodes or phase lag differences between two conditions. The online Matlab code shows you how to do this. More information on what to do with these phase values is presented in section 26.10.

26.8 Phase Lag-Based Measures

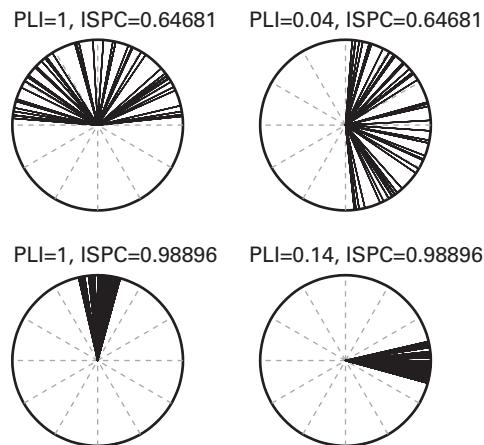
Because effects of volume conduction are instantaneous within measurement capabilities of M/EEG acquisition and within frequencies typically investigated in M/EEG research (Plonsey and Heppner 1967; Stinstra and Peters 1998), spurious connectivity results that are caused by two electrodes measuring activity from the same source will have phase lags of zero or π (π if the electrodes are on opposite sides of the dipole). (For the remainder of this chapter, “zero-phase” will be used instead of “zero- or π -phase” for convenience.) Thus, it seems sensible to avoid spurious connectivity results due to volume conduction by avoiding zero-phase-lag connectivity. There are several phase-based connectivity measures that ignore zero-phase-lag connectivity, including imaginary coherence (Nolte et al. 2004), phase-slope index (Nolte et al. 2008), phase-lag index (Stam, Nolte, and Daffertshofer 2007), and weighted phase-lag index (Vinck et al. 2011). These measures are insensitive to volume conduction, although in some cases they may still be susceptible to mixing sources (Peraza et al. 2012).

Imaginary coherence (Nolte et al. 2004) was developed as a way to apply spectral coherence without concern for spurious connectivity due to volume conduction. Computing imaginary coherence uses almost the same equation as that for spectral coherence, except the imaginary part of the spectral coherence is taken before the magnitude. The online Matlab code will show you how to compute imaginary coherence.

The phase-lag index (Stam, Nolte, and Daffertshofer 2007) measures the extent to which a distribution of phase angle differences is distributed toward positive or negative sides of the imaginary axis on the complex plane (that is, whether the vectors are consistently pointing “up” or “down” in polar space when the imaginary axis corresponds to a vertical line). The idea is that if spurious connectivity is due to volume conduction, the phase angle differences will be distributed around zero radians (as in figure 26.1D). In contrast, non-volume-conducted connectivity will produce a distribution of phase angles that is predominantly on the positive or on the negative side of the imaginary axis. Thus, with the phase-lag index, the vectors are not averaged, but instead, the sign of the imaginary part of the cross-spectral density is averaged. If all phase angle differences are on one side of the imaginary axis, the phase-lag index will be high. In contrast, if half of the phase angle differences are positive and half are negative with respect to the imaginary axis, the phase-lag index (PLI) will be zero.

$$PLI_{xy} = \left| n^{-1} \sum_{t=1}^n \text{sgn}(\text{imag}(S_{xyt})) \right| \quad (26.6)$$

in which $\text{imag}(S)$ indicates the imaginary part of the cross-spectral density at time point (or trial) t and is extracted from a complex number in Matlab using the function `imag`; sgn

**Figure 26.9**

Comparison of the phase-lag index (PLI) and ISPC under different phase angle distributions. ISPC depends only on the clustering of the distribution, regardless of its mean phase direction, whereas PLI is sensitive to the phase angle directions rather than clustering per se. Note that the cases on the right side of the figure may be difficult to interpret—zero phase lag can indicate either true zero-phase-lag connectivity or spurious connectivity due to volume conduction.

indicates the sign (-1 for negative values, $+1$ for positive values, and 0 for zero values). The phase-lag index is less sensitive to outliers, but it is also less sensitive to the amount of clustering in the distribution. That is, if the phase values are spread out in polar space but all on one side of the imaginary axis, phase-lag index will still be high. This is shown in figure 26.9.

The weighted phase-lag index (not to be confused with wISPC; weighted phase-lag index is not computed in relation to any trial-varying variable) is an extension of the phase-lag index in which angle differences are weighted according to their distance from the real axis (Vinck et al. 2011).

$$wPLI_{xy} = \frac{n^{-1} \sum_{t=1}^n |imag(S_{xyt})| \operatorname{sgn}(imag(S_{xyt}))}{n^{-1} \sum_{t=1}^n |imag(S_{xyt})|} \quad (26.7)$$

You can see equation 26.6 embedded in the numerator of equation 26.7, but equation 26.7 also scales the sign of the angles by the magnitude of the imaginary component (see also figure 1 in Vinck et al. 2011); thus, vectors further away from zero or π radians have a larger influence on the estimate of connectivity. As with spectral coherence, the weighting

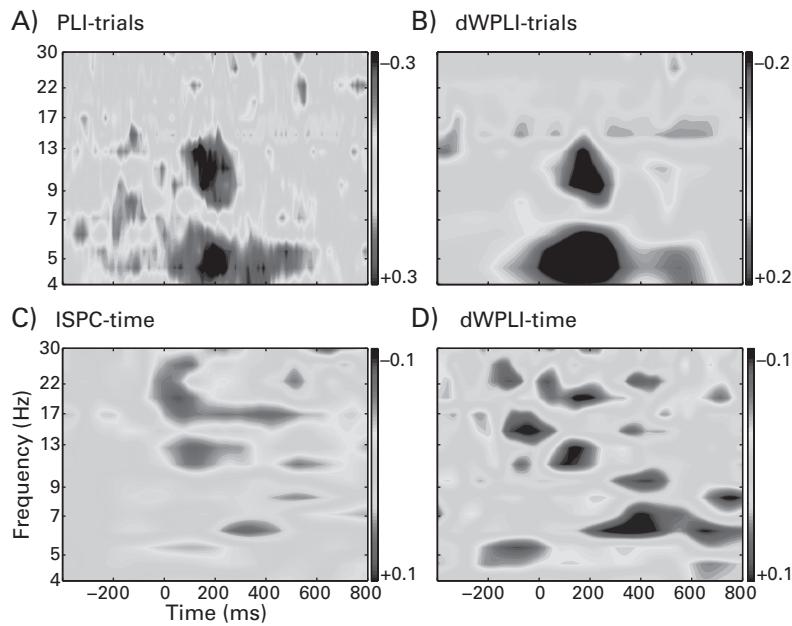
term would result in a scaling of wPLI values. Thus, the denominator unscales the final result. The weighted phase-lag index was further developed by introducing a debiasing term to correct for some inflation due to sample size (Vinck et al. 2011). This is shown in the online Matlab code.

The phase-slope index was developed to measure directed phase-based connectivity (Nolte et al. 2008). Consider that if there is a directed functional connection from area A to area B with a 10-ms phase lag, the spectral representation of the 10-ms phase lag (that is, the phase delay) will increase with increasing frequency. This is because a 10-ms delay corresponds to a larger phase lag at 10 Hz compared to 5 Hz (respectively, 0.628 and 0.314 radians, using equation 26.8 below). Thus, the phase-slope index measures whether the slope of the phase lag is consistently positive or negative over several adjacent frequency bins (computed from the Fourier transform). Furthermore, the sign of the slope indicates whether the net connectivity flows from region A to B or the reverse. You can specify which frequency bands to use, thus making the phase-slope index a frequency-band-specific phase-based measure of directed connectivity. The main limitation of the phase-slope index is that it measures only the overall asymmetry of directed connectivity; if there is equally strong bidirectional connectivity the phase-slope index will be close to zero. For this reason, it may be useful to compare results from the phase-slope index to results from ISPC or another non-directional phase-based connectivity measure. Because the phase-slope index is based on FFTs of time series data, it can be computed only over time, not over trials. The phase-slope index is not further discussed here, but the online Matlab code contains a function to compute the phase-slope index (`data2psiX.m`), which is a modified version of the Matlab code that accompanies the Nolte et al. (2008) paper.

Like other measures of connectivity discussed in this chapter, phase-lag-related measures can be computed over trials at each time point or over time at each trial. Figure 26.10 (plate 18) shows a comparison of results from ISPC, phase-lag, and weighted phase-lag indices over trials and over time.

In figure 26.10 (plate 18), you can see that the phase-lag index over trials and weighted phase-lag index over trials (panels A and B) look very similar to each other. The weighted phase-lag index over time (panel D) looks different from the weighted phase-lag index over trials but looks similar to ISPC-time (panel C; this is the same result as shown in figure 26.3A [plate 15] with different color scaling). This figure further highlights that connectivity measures can reveal different dynamics when assessed over time versus over trials.

There are two limitations of phase-lag-based measures. First, if the two electrodes have slightly different frequency concentrations, phase-lag indices can fluctuate rapidly as the

**Figure 26.10 (plate 18)**

Visual comparison of different measures of baseline-subtracted phase-based connectivity computed between Fz and O1. The color scaling differs across plots to facilitate qualitative comparisons.

phase angle differences spin around polar space. That is, although the analysis parameters are identical, because of some frequency smoothing from wavelet convolution or filtering, one electrode may have more energy at a slightly higher frequency than the other electrode. In this case the preferred phase difference angle will spin around polar space, creating large transient fluctuations in phase-lag indices as the preferred phase difference angle crosses zero or π radians. This can be seen in figure 26.11. Figure 26.11 intentionally shows a somewhat extreme example in which a short time segment length was chosen to illustrate this point. Nonetheless, this phenomenon can happen in task-related analyses if the time segment lengths are short or if there is strong phase clustering combined with rotating phase angle differences. Such an effect can also occur if the frequency of activity at one electrode changes slightly over time, which has been shown to occur in real data (Burgess 2012). This effect can be minimized by increasing the number of time points or trials in the analysis and by selecting time-frequency decomposition methods that increase frequency specificity.

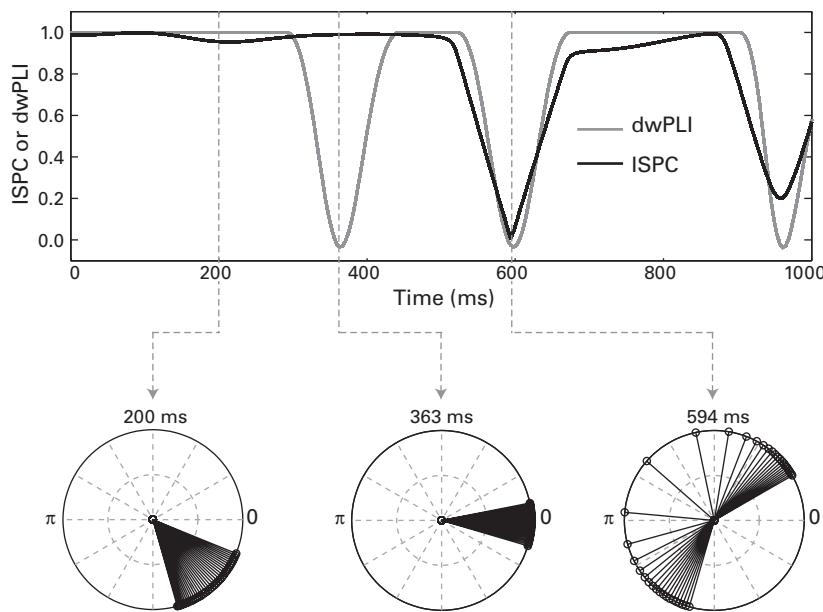


Figure 26.11

Illustration of one potential limitation of phase-lag-based measures. If the preferred phase difference angle is not stationary over time, phase-lag-based connectivity measures will transiently go toward zero as the distribution passes through 0 or π radians (this happens here at 363 ms). Polar plots show the phase angle differences over time corresponding to selected center time points. This figure is a screen shot of a movie, which you can view using the online Matlab code.

A second, related, limitation is that condition differences in phase-lag-based measures can reflect either differences in connectivity or differences in the preferred phase of the connectivity. For example, imagine two conditions with equal phase angle difference clustering, but condition A is associated with a phase angle difference of $\pi/2$ and condition B is associated with a phase angle difference of $\pi/5$. Because of measurement noise, there may be more individual phase angle difference values below zero (negative values on the imaginary plane) in condition B compared to condition A. This will cause phase-lag-based indices to be lower for condition B compared to condition A, although the strength of phase lag-based indices should be inspected, for example in combination with the gv-test (explained in section 26.10). If the preferred phase angles of the two conditions are not statistically significantly different, condition differences in phase-lag-based measures can be safely interpreted; if the preferred phase angles are statistically significantly different, the condition

difference in phase-lag-index could be due to differences in the preferred phase difference angles.

In other words, phase-lag-based measures assume that both the phase lag and the frequencies of activities of the two electrodes are stationary for the duration of time used in the analysis. If these two assumptions are not met, then it is possible that the same amount of phase coupling can cause spurious time-varying changes in the apparent strength of phase-lag-based measures.

26.9 Which Measure of Phase Connectivity Should You Use?

The phase-based measure of connectivity that is most appropriate depends in part on how hypothesis-driven versus exploratory and data-driven your analyses are. If you have a priori hypotheses about a small number of specific connectivity patterns, ISPC combined with tests against volume conduction might be a better option because ISPC is maximally sensitive to detecting connectivity, regardless of the phase angle differences.

On the other hand, if you have no or few hypotheses and will thus do a lot of data exploration by testing for connectivity over many electrode pairs, time points, and frequency bands, it might be better to use a connectivity measure that is insensitive to volume conduction, even at the risk of ignoring some potentially true connectivity patterns, because inspecting each of hundreds or thousands of results for possible contamination by volume conduction is impractical. With measures such as phase-lag index or imaginary coherence, you can be more confident about minimizing volume conduction contamination. On the other hand, because they are also sensitive to the mean phase angle of the phase angle distribution, phase-lag-based measures may be best suited for resting state or tasks in which connectivity strength is not compared across conditions.

Another consideration is whether to compute connectivity over trials or over time. This decision is less influenced by your orientation toward hypothesis testing versus data exploration but rather depends on your task design and expectations about the results. Differences between connectivity over trials versus over time are discussed in section 26.3, but briefly: connectivity over time is more sensitive to detecting high-frequency connectivity and for resting-state data or tasks that have long event durations (i.e., at least several hundred milliseconds) due to the poor temporal precision of connectivity over time; connectivity over trials has higher temporal precision and is therefore better able to identify the time course of changes in connectivity and is also better able to identify transient changes in connectivity.

26.10 Testing the Mean Phase Angle

Typically when ISPC is computed, the angle of the mean vector is ignored (as it is with ITPC from chapter 19). The angle of the mean vector (also called the “preferred angle”) can be extracted in Matlab by using `angle(mean(. . .))` instead of `abs(mean(. . .))`. The angle of connectivity may be useful for two reasons. First, it provides the phase lag or the time lag between the two electrodes. The phase lag has units of radians but can be converted to time in milliseconds using equation 26.8.

$$\text{lag(ms)} = \frac{1000\phi_d}{2\pi f} \quad (26.8)$$

in which ϕ_d is the phase angle difference between the two electrodes in radians (change the 1000 to 1 to convert to seconds). Time lag can be used as supportive evidence for directionality of connectivity but does not provide unambiguous evidence for directionality, nor does it imply causality (see section 25.2).

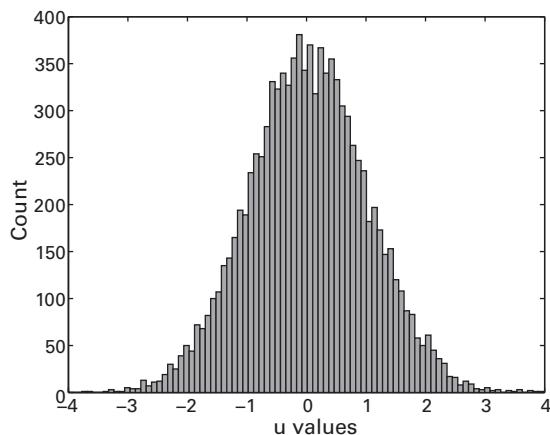
The second reason the phase angles from ISPC might be useful is to test whether a phase angle or a phase angle difference is statistically significantly different from some specified phase angle. This can be used either to test for possible contamination by volume conduction or to test for condition differences in the preferred phase angle of connectivity (e.g., if the phase lag is longer in one condition compared to another condition). This can be done with the “ v -test” (Durand and Greenwood 1958; Zar 1999).

$$u = nISPC \cos(\phi - \Phi) \sqrt{\frac{2}{n}} \quad (26.9)$$

in which n is the number of trials or time points, ϕ is the observed phase angle difference, and Φ is the hypothesized phase angle against which to test (an intermediary step folded into equation 26.9 is called v , hence the name). Remarkably, the quantity u is normally distributed under the null hypothesis (figure 26.12). This means that you can directly convert the u from equation 26.9 into a p -value, as you would for a normal Z-score. This can be done with the Matlab function `normcdf` (which is in the Matlab statistics toolbox), or you can look up key values in a p -value table (e.g., $u = 1.96$ corresponds to $p < 0.05$ two-tailed).

To run this test, specify Φ as a phase value against which to test (make sure Φ is radians and not milliseconds or angles in degrees). In the test for volume conduction you would hope to obtain a nonsignificant p -value (that is, the null hypothesis that the phase angle difference is not zero cannot be rejected).

Unfortunately, this test has four disadvantages when it comes to testing phase angles with typical EEG datasets. First, the v -test works well for a small number of data points but is less

**Figure 26.12**

A distribution of u values from equation 26.9 under the null hypothesis, generated by computing ISPC between 640 pairs of random vectors (10,000 tests were done to create this distribution). In this particular simulation, the mean was -0.015 , and the variance was 1.0003 .

useful when there are more than 20 data points. This can be seen in figure 26.13B, where $N = 300$ (a good number of trials or time points for an EEG study with a very high signal-to-noise ratio) leads to half of polar space being “significantly” equal to zero radians. Second, the v -test is symmetric, meaning that the p -value for phase angle φ is exactly 1 minus the p -value for $2\pi - \varphi$ (this results from the cosine component in equation 26.9, which gives a cyclic response). Third, it has narrow slopes, that is, p -values tend to be very close to 0 or very close to 1, particularly for large N . Fourth, it produces many false positives, particularly with many data points. In 10,000 simulations of 640 pairs of random phase angles, around 5–6% of phase couplings are statistically significant using an α of $p < 0.05$. Although it may seem unsurprising that random data are considered significant 5% of the time when using an α of 5%, a modification of the v -test presented below is more sensitive and therefore results in fewer Type-I errors (false positives).

Therefore, I propose a modification of the v -test, called the Gaussian v -test, or gv -test. The gv -test simply replaces the cosine component with a Gaussian component, as seen in equation 26.10. This minor change addresses several disadvantages of the v -test (see figure 26.13 for comparisons between the v -test and the gv -test): it is robust to a large number of data points because the width of the Gaussian scales with the number of points; it is asymmetric and thus tests precisely for one region of phase space rather than testing polar opposites of phase space; and it produces fewer false positives. In the same 10,000 simulations described above, there were around 0.3–0.5% false positives. Therefore, the gv -test is a

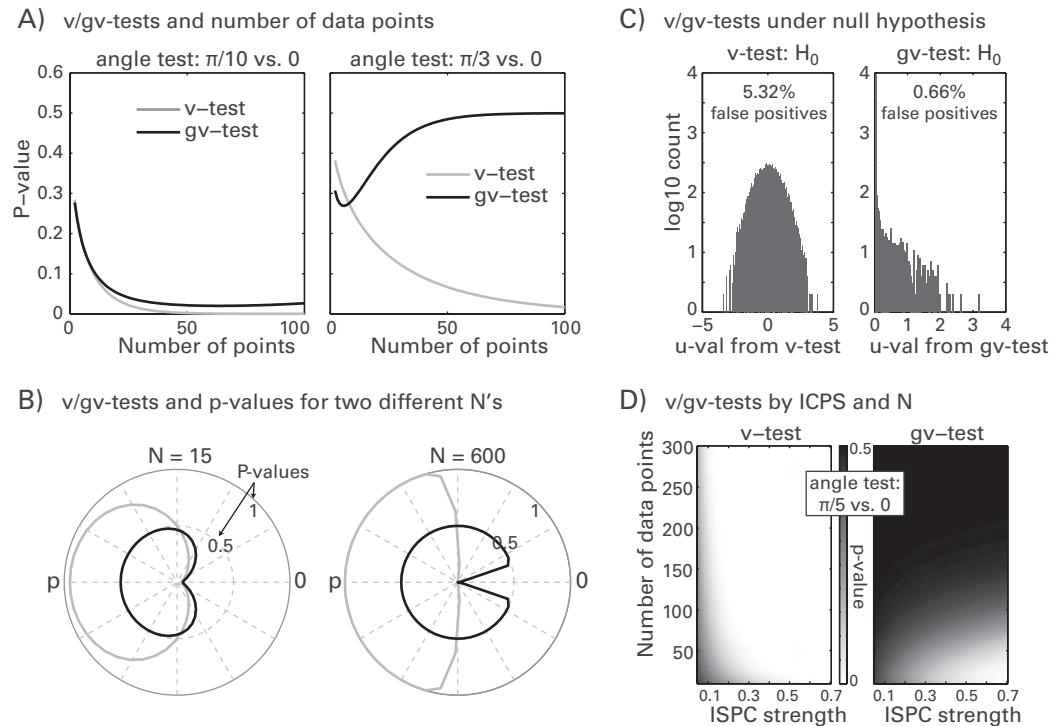


Figure 26.13

Comparison of *v*-test and *gv*-test. Panel A shows *p*-values (*y*-axis) as a function of the number of points (time points or trials) in the analysis (*x*-axis), for $\pi/10$ versus 0 radians (left) and $\pi/3$ versus 0 radians (right). Panel B shows *p*-values (radial axis) as a function of phase angles in radians in a test against zero radians. Panel C shows *u*-values from *v*-test and *gv*-test for random data. Panel D shows *p*-values (gray scale) as a function of the number of data points (*y*-axis) and ISPC strength (*x*-axis).

suitable alternative to the *v*-test in situations of large number of data points, as is typically utilized in EEG studies. For a small number of data points, the *gv*-test and the *v*-test will provide similar results.

$$u = nISPCe^{-\phi_d^2/4\pi/n} \sqrt{\frac{2}{n}} \quad (26.10)$$

in which ϕ_d is the difference between the observed phase value and the phase value against which to test [the same as $\varphi - \Phi$ in equation 26.9], and $4\pi/n$ is $2s^2$, where s is the square root of $2\pi/n$.

Thus, after ISPC has been computed, the gv-test is one possible analysis to test whether the result may have been due to volume conduction. If the gv-test provides evidence that the phase angle differences are 0 or π , the result could indicate zero-phase-lag connectivity or volume conduction (in this case, further inspection would be required, as described in section 25.10); if the gv-test provides evidence that the phase angle differences are not 0 or π , the result cannot be due to volume conduction.

One point to keep in mind about testing phase angles across subjects is that the precise phase value will depend on a variety of factors including cortical folding and dipole orientation, which may differ across subjects even if the neurocognitive process is the same.

26.11 Describing These Analyses in Your Methods Section

There are many possibilities for phase-based measures of connectivity. Thus, the two most important things to include in the Methods section are a justification for why you chose one measure over the other measures and a clear description of the analysis performed. The clear description is important because of inconsistent terminology in the literature and because of the number of possibilities for computing phase-based connectivity, including whether connectivity was computed over time or trials (this is often unclear in publications). Include the formulas, statistical procedures or transformations (e.g., z-normalization for wISPCz), the number of trials per condition, and, if relevant, the number of cycles from the wavelet convolution or the length of time if the phase-based connectivity measure was computed over time; these details provide information regarding the temporal precision of the results. If you make modifications to standard analysis approaches, you can also consider including your Matlab code as an appendix or in an online supplemental materials section.

26.12 Exercises

1. Select one seed electrode and one frequency band and compute phase-based connectivity between that seed electrode and every other electrode. Use two methods for phase-based connectivity that were presented in this chapter, one that is volume conduction independent (e.g., PLI) and one that could produce spurious connectivity due to volume conduction (e.g., ISPC). Do not apply a baseline subtraction. Make topographical plots of seeded connectivity in a time window of your choice (e.g., 300–350 ms). What are the similarities and differences between results from the two methods, and what might be the reasons for the similarities and differences?

2. Now apply a baseline subtraction to the results (you can choose the baseline time period). Are there any changes in the plots after baseline subtraction (note that the color scaling will be different after baseline subtraction), and how do results from the two analyses compare with each other after baseline subtraction?
3. From the results in exercise 1 above, pick one “target” electrode (any electrode other than the seed) and provide evidence, using additional data analyses if necessary, for or against that measure of phase-based connectivity between that electrode and the seed being driven by volume conduction.

27 Power-Based Connectivity

Functional connectivity analyses based on fluctuations in power provide a range of opportunities for examining connectivity over time and over frequency. You may have noticed in the previous chapter that phase-based connectivity analyses assume that the connectivity is instantaneous (although not necessarily with zero phase-lag), and at the same frequency. Power-based connectivity analyses do not have this constraint, which make them more flexible for hypothesis-driven as well as exploratory analyses.

27.1 Spearman versus Pearson Coefficient for Power Correlations

All of the power-based connectivity methods presented in this chapter rely on bivariate correlation coefficients. Thus, before learning about different methods of examining power-based connectivity, it is worth discussing whether the Pearson or the Spearman method of correlation is more appropriate for time-frequency power.

The Pearson correlation coefficient is a commonly used correlation measure and is defined as the covariance of two variables, scaled by the variance of each variable.

$$r = \frac{\sum_{t=1}^n (x_t - \bar{x})(y_t - \bar{y})}{\sqrt{\sum_{t=1}^n (x_t - \bar{x})^2 \sum_{t=1}^n (y_t - \bar{y})^2}} \quad (27.1)$$

This may look like a big equation, but the numerator is simply the sum of variables x times variables y at each time point (or trial) t , after subtracting the mean of each variable (the bar on top of the variable indicates the mean). The denominator is the variance of each electrode. Equation 27.2 shows an equivalent way of writing this equation using matrix algebra.

$$r = \frac{xy^T}{\sqrt{(xx^T)(yy^T)}} \quad (27.2)$$

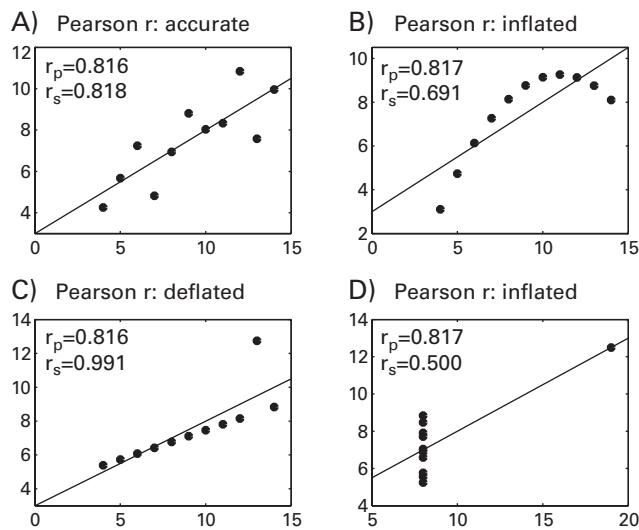
The superscript T indicates the matrix transpose. This formulation assumes that the data are electrodes-by-time; if you store the data as time-by-electrodes, you should transpose the first matrix instead of the second matrix. The online Matlab code demonstrates that equations 27.1 and 27.2 are equivalent.

It is insightful to compare equations 27.1/2, 26.2/4, and 23.1. You should see that the Pearson correlation coefficient is computed in a similar way to how spectral coherence is computed, which is similar to how covariances are computed. Thus, covariances scaled by variances offer a general approach for measuring bivariate relationships in many kinds of data.

Interpreting the Pearson correlation coefficient relies on the assumption that the data are normally distributed. Violations of this assumption will introduce a bias into the correlation coefficient. This is shown below.

The Spearman correlation (also sometimes called Spearman ρ) is similar to the Pearson correlation, but it is nonparametric. It is a rank correlation, which means that the data from each variable are ordered and then ranked, such that the numbers 0.1, 0.2, 0.21, 10,000.1, become 1, 2, 3, 4. Normally, 10,000.1 in this context would be an outlier, but ranking the variables eliminates the influence of this outlier without removing it from the data. After rank-transforming the data (each variable is separately rank-transformed), you can apply equation 27.1. There are faster algorithms to compute the Spearman correlation, which are shown later in this chapter. Conceptually, however, it is important to understand that the only difference between the Spearman and the Pearson correlation is that the Spearman correlation involves rank-transforming the data before applying equation 27.1 (or equation 27.2). The ranking can be done over time or over trials; which dimension is appropriate depends on which dimension you are using to compute the correlation coefficient. Ranking variables can be done with the Matlab function `tiedrank`. This function is included in the statistics toolbox; if you do not have the statistics toolbox, you can use the function `sort`, but you will need to develop a work-around for tied variables.

There are two reasons why within-subject power correlations should be done with the Spearman and not the Pearson correlation: power data are nonnormally distributed, and they contain outliers. To demonstrate how the Pearson correlation can misrepresent bivariate relationships that contain nonnormal distributions or outliers, consider “Anscombe’s quartet.” Anscombe’s quartet is a set of four pairs of vectors that have the same basic statistical properties and Pearson correlation coefficients but, as you can see in figure 27.1, very

**Figure 27.1**

Comparison of Pearson (r_p) and Spearman (r_s) correlation coefficients for Anscombe's quartet.

different distributions. (You can also create your own Anscombe-type data; see Chatterjee and Firat 2007.)

There are two interesting results to note from figure 27.1. One is that when the data are normally distributed as in panel A, Pearson and Spearman correlation coefficients are nearly identical. The second result is that when there are outliers or a nonnormal distribution, the Pearson correlation can provide an *inflated* (as in panels B and D) or a *deflated* (panel C) estimate of the monotonic relationship between the two variables. This is particularly troublesome when the outliers have large leverage, a measure of the influence of an outlier (outliers in the middle of the distribution have less leverage compared to outliers at the tails of the distribution). Thus, with nonnormal distributions or in the presence of outliers, the Spearman correlation is less biased than the Pearson correlation. And when the data are normally distributed and contain no outliers, Spearman and Pearson correlations provide nearly identical results (panel A).

In real EEG data, time-frequency power data are nonnormally distributed and contain outliers. The reasons for this were discussed in chapter 18 (in particular, section 18.8) and illustrated in figure 27.2. A logarithm-base-ten transform helps the distribution look more normal, although in this case, the distribution is still technically nonnormal, according to the Kolmogorov-Smirnov test. Furthermore, if you are correlating across frequency bands,

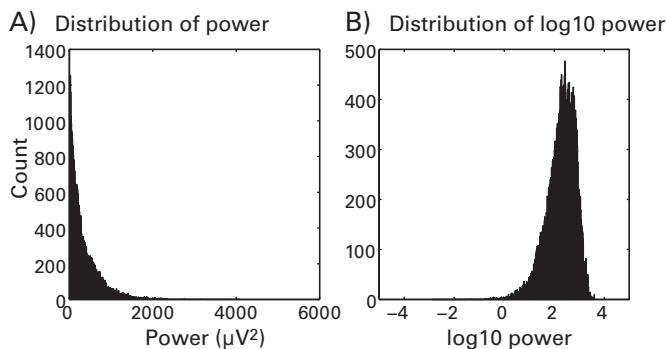


Figure 27.2

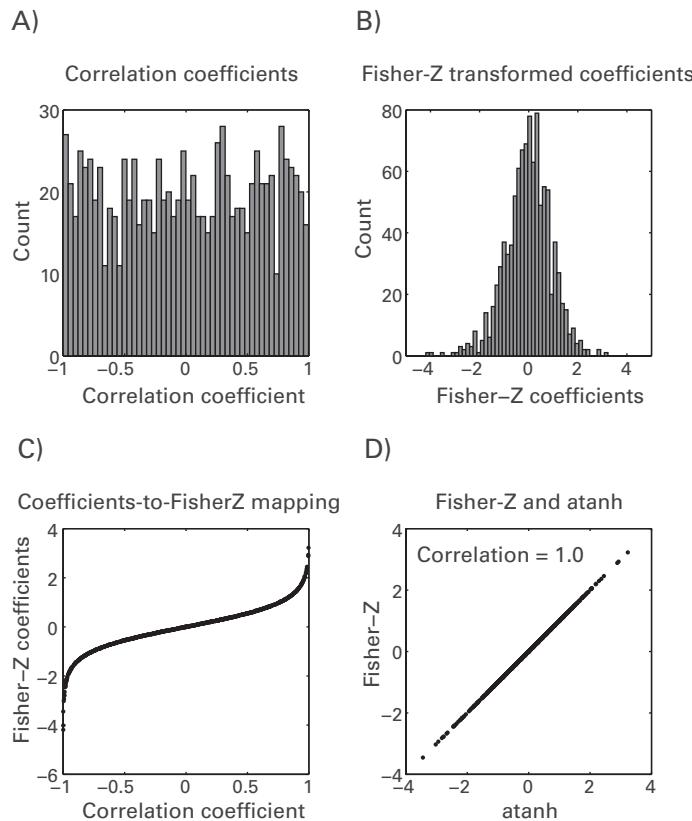
Distribution of raw power data shows a nonnormal distribution (panel A). Taking the logarithm-base-10 transform of the data helps, although the distribution is still not Gaussian, as assessed with the Kolmogorov-Smirnov test.

there will still be power differences over frequencies, even after taking the logarithm of power within each frequency band.

In summary, the lack of normal distribution and the possibility of outliers argue in favor of using the Spearman correlation instead of the Pearson correlation. In the best-case scenario in which the data are normally distributed, these two measures will provide the same result, and in the realistic case that the data are nonnormally distributed and contain outliers, the Spearman correlation will provide a less biased estimate of a bivariate relationships. If you want to use the Pearson correlation, you should inspect the data carefully for violations of normal distribution or outliers to make sure that the correlation will not be biased.

Note that trial-averaged baseline-corrected power (decibel or percentage change) is often normally distributed and therefore can be tested using Pearson correlations, for example when studying cross-subject correlations between EEG activity and task performance.

There is one final matter to discuss before going on to specific methods for power-based connectivity analyses. Correlation coefficients (whether computed by the Spearman or the Pearson method) are not drawn from a normal distribution; they have a bounded distribution between -1 and +1. Although in practice the observed correlation coefficients might appear normally distributed, they should be transformed before statistical evaluation. The most typical transform for correlation coefficients is the Fisher-Z transform, which effectively “stretches out” the numbers so that they have a broader and more normal-looking distribution (figure 27.3).

**Figure 27.3**

Correlation coefficients are bound between -1 and $+1$ but can approach a normal distribution by applying a Fisher-Z transform.

$$r_{fz} = 0.5 * \log\left(\frac{1+r}{1-r}\right) \quad (27.3)$$

where r is the correlation coefficient. The Fisher-Z transform is actually the inverse hyperbolic tangent (Matlab function `atanh`) (figure 27.3D). Using the Matlab function `atanh` is faster and more convenient than writing out equation 27.3.

27.2 Power Correlations over Time

Many connectivity measures can be done over time or over trials. See chapter 26 and figure 26.4 for an explanation of the difference between connectivity over time and connectivity

over trials. Computing power correlations over time involves simply correlating the power time series between two electrodes over a period of time. Power correlations over time can be used for task data or resting-state data. The power time series from the two electrodes can be in the same frequency band or in different frequency bands. If they are in different frequency bands, the correlation would reflect cross-frequency coupling (chapter 30) (Bruns et al. 2000; Hipp et al. 2012). Unless you have a hypothesis about the frequency bands used for cross-frequency coupling, it is best to use the same frequency bands for the two electrodes.

To perform this analysis, first pick two electrodes, compute the power time series (using any time-frequency decomposition method you prefer), and then compute the correlation coefficient between time-varying power of the two electrodes. In figure 27.4A you can see a plot of power over time at electrodes F5 and P4, and in figure 27.4B you can see power in one electrode plotted as a function of power in the other.

As with other connectivity measures computed over time, the main parameter to select is the length of the time segment used to compute the correlation coefficient. If the time segment is too long, transient changes in connectivity might not be detected, but if the time segment is too short, you may have too little data for robust correlation coefficient estimates. In the analysis shown in figure 27.4, the entire trial period was used, which means any correlation (or lack thereof) would reflect a mixture of tonic and task-related connectivity. The time segment should be at least one cycle of the frequency band. For task data, you could use time segments of at least two to four cycles, and this number could increase with higher frequencies and with longer task designs, as discussed for ISPC-time in the previous chapter. For resting-state data, you could segment the data into nonoverlapping chunks of a few seconds, compute a correlation coefficient on each segment, and then average the correlation coefficients together. This will help to increase the signal-to-noise ratio.

Power correlations over time are not limited to instantaneous correlations as was illustrated in figure 27.4. You can perform a cross-correlation analysis, which will reveal whether peak connectivity is observed when one time series is temporally shifted relative to the other. A cross-correlation works by repeatedly correlating two variables and time-shifting one variable with respect to the other at each iteration. In practice, the cross-correlation is computed in the frequency domain for computational efficiency (details are not presented here). You can use the Matlab function `xcov` with the ‘`coeff`’ option to return correlation coefficients. You might initially think that the function `xcorr` is better, but this function scales the result such that the maximum autocorrelation is 1.0, whereas the function `xcov` scales to the correlation coefficient. Note that the function `xcov` does not perform a Spearman correlation, so you will need to rank-transform your data first (Matlab function `tiedrank`).

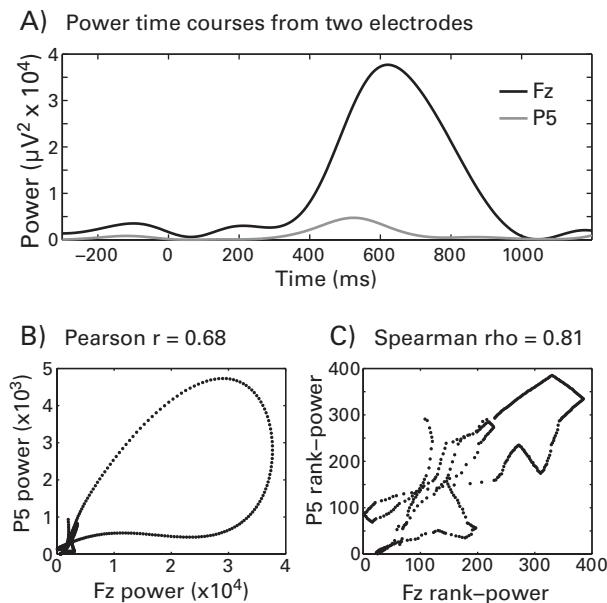


Figure 27.4

Power time series at 6 Hz from two electrodes during trial 10. This correlation was not robust in all trials. This can be seen in the online Matlab code by generating this figure for, for example, trial 2.

An example cross-correlation from one trial is shown in figure 27.5. You can see that there is an asymmetry, with a peak at around +15 ms. Cross-correlation results should be interpreted only if there is a peak in the cross-correlation function.

In a cross-correlation analysis, the theoretical maximum number of lags is the number of time points in the time series. But because these time series are rhythmic, you should use a number of lags equal to or less than one cycle. This can be specified as an input to the function `xcov`. The online Matlab code illustrates how to setup a cross-correlation analysis.

27.3 Power Correlations over Trials

There are three ways to perform power correlations over trials, depending on the goal of the analysis and on your hypotheses. The first method is the most hypothesis-driven because you specify the time-frequency-electrode windows to use a priori. To perform this analysis, select time-frequency windows for two electrodes (the time-frequency windows need not be the same for both electrodes), extract power from that window from each trial (averaging

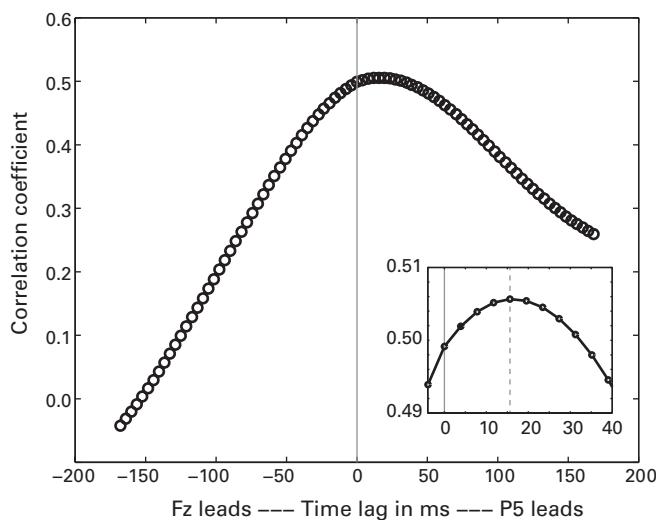
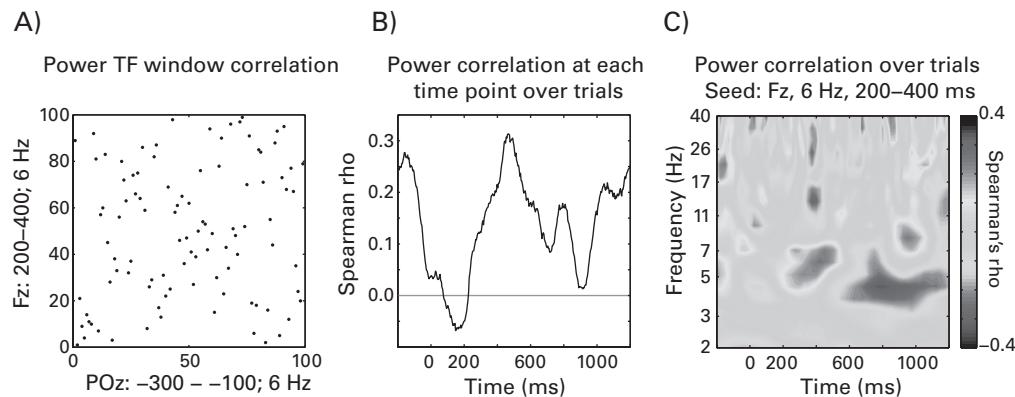


Figure 27.5

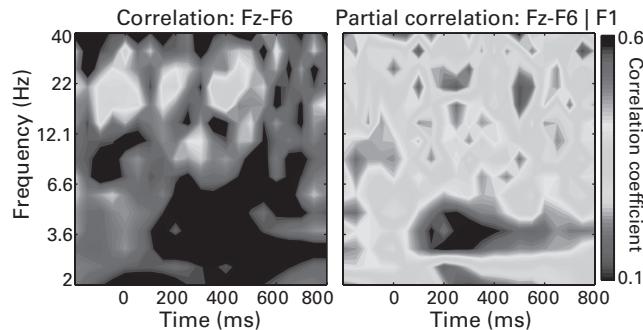
Cross-correlation plot for one trial, showing the correlation coefficient as a function of time lag between the two electrodes (the inset shows a zoom-in to the peripeak area). Without a peak in the cross-correlation plot, a cross-correlation result can be difficult to interpret. Similar to figure 27.4, this trial was selected for this illustration; this pattern of lagged correlations was not consistent over trials. You can explore the cross-correlation results from other trials using the online Matlab code.

over all time-frequency points within that window), and compute one correlation coefficient. In the example shown in figure 27.6A (plate 19), power from POz at 6 Hz from -300 to -100 ms was correlated with power from Fz at 6 Hz from 200 to 400 ms. Each dot in the scatter plot is a trial. This method has been used, for example, to show impaired connectivity between posterior alpha and frontal theta in children with ADHD compared to typically developing children (Mazaheri et al. 2010).

The second method is similar to how ISPC-trials is computed: correlate power at each time point over trials; this will produce a time series of correlation coefficients. A sample result is shown in figure 27.6B (plate 19). For this analysis, the correlation was computed using power from the same frequency band for both electrodes, but the frequency band may be different for the two electrodes. This method is useful when you have an a priori reason to select two electrodes and frequency band(s) but want the flexibility of assessing changes in connectivity over time. This method could be extended to correlate at many frequencies, thus creating a time-frequency map of correlation coefficients (a sample result from this analysis is shown in figure 27.7A [plate 20]).

**Figure 27.6 (plate 19)**

Results from three methods of computing power correlations over trials. Panel A shows the correlation using predetermined time-frequency windows (see *x*- and *y*-axis labels; each dot corresponds to one trial). In this case there is no relationship. Panel B shows the correlation computed at each time point separately (this is the power correlation analogue of ISPC-trials). Panel C shows exploratory power correlations over time and over frequency. The seed time-frequency region was 200–400 ms, 6 Hz power at Fz, and the target was time-frequency power from Oz.

**Figure 27.7 (plate 20)**

Time-frequency map of partial correlation coefficients between power at Fz and F6 (panel A), while holding F1 constant (panel B). Partial correlations can be used to test network-level hypotheses involving more than two electrodes, or they can be used to minimize the effects of volume conduction on power-based connectivity (as illustrated here) by removing the variance between two electrodes (in this case, Fz-F6) that is shared with—and likely volume-conducted from—a neighboring electrode (F1).

The final method for power correlations over trials is more open to exploratory analyses. This method involves selecting a time-frequency window from one “seed” electrode and correlating cross-trial power fluctuations in the seed time-frequency window with cross-trial power fluctuations in all other time-frequency points at one, some, or all electrodes. This will produce a time-frequency-electrode map of correlation coefficients, and the coefficient at each time-frequency point reflects the correlation between power at that time-frequency point and power in the seed time-frequency window. It is therefore advantageous over phase-based measures of connectivity because it does not require the connectivity to occur simultaneously or in the same frequency band.

In figure 27.6C (plate 19) you can see an example result of this analysis. The seed was electrode Fz from 200 to 400 ms at 6 Hz, and the time-frequency map shows correlations with Oz power. This result indicates, for example, that trials with more frontal theta power after stimulus onset also had less occipital theta from around 600 to 1200 ms.

27.4 Partial Correlations

Partial correlations allow you to measure the linear or monotonic (via Pearson or Spearman correlation) relationship between two variables (X and Y) while holding constant a third variable (Z). In the case of EEG power-based connectivity, partial correlations can be useful for two reasons. First, they can be used to test hypotheses about networks comprising more than two nodes. Second, partial correlations can be used to minimize volume conduction artifacts during power correlations. Consider that X and Y are two electrodes spaced many centimeters apart from each other, and that X and Z are physically neighboring electrodes. If the power correlation between X and Y is very similar to the correlation between Z and Y , this could be attributed to volume conduction between X and Z . Thus, computing the partial correlation between X and Y , holding Z constant, will remove shared variance between X and Z , much of which reflects volume conduction.

$$r_{xy \bullet z} = \frac{r_{xy} - r_{xz}r_{yz}}{\sqrt{1 - r_{xz}^2}\sqrt{1 - r_{yz}^2}} \quad (27.4)$$

Like other connectivity analyses, partial correlations can be computed over time or over trials. There are also extensions of partial correlations that allow you to hold constant more than one variable. Figure 27.7 (plate 20) shows time-frequency correlation maps between Fz and F6 before (panel A) and after (panel B) controlling for F1 (thus, F1 acts as the “Z” variable in the explanation above). Note that this correlation is different from the results presented in

figure 27.6 (plate 19) because the present correlation was done at each time-frequency point, not from a constant seed window.

27.5 Matlab Programming Tips

The first programming tip is that in many but not all cases you can downsample the results. This concept was introduced in section 18.14. Although the high temporal resolution of EEG is necessary to extract frequency information, after the frequency information has been extracted there is an increase in temporal autocorrelation and thus a decrease in temporal precision. The temporal resolution is now greater than the temporal precision, and thus, the resolution can be reduced to the level of the precision. Decreasing the temporal resolution is done by downsampling the results (note that you should not downsample the data before applying a time-frequency decomposition). Downsampling the results will decrease analysis time and file read/write times and will decrease disk space usage. This is particularly useful for correlations because they can be time-consuming to compute if you have many time points, electrodes, and conditions.

Downsampling results is not always a good idea, and the decision of whether and how much to downsample the results depends on your hypotheses, what you hope to find in the data, what kind of analysis you performed, the parameters that you used during time-frequency decomposition, and what frequencies you are interested in. In general, results from lower frequencies can be temporally downsampled more than results from higher frequencies, and time-frequency decomposition methods that increase temporal smoothing can be downsampling more than methods that involve less temporal smoothing. If you expect transient effects, then downsampling too much might be deleterious. In most situations, downsampling the results to 40 Hz or 50 Hz (thus, one data point every 20 or 25 ms) maintains the advantages of downsampling with minimal loss of information. An example of the loss of information at higher frequencies with postresults temporal downsampling is shown in figure 27.8. In the online Matlab code you can specify the downsampling rate to observe the effect of downsampling on information loss at any frequency you specify.

The second programming tip is that there are faster ways to compute correlations than using the Matlab functions `corr` or `corrcoef`. The Matlab function `corr` is written to be flexible and to handle many kinds of data, deal with exceptions and missing data, and compute *p*-values. It therefore contains many loops and command statements (`if`, `switch`, etc.) that, on the one hand, provide this function with flexibility to be used for a wide variety of inputs but, on the other hand, can be eliminated if you know that your data are OK (for

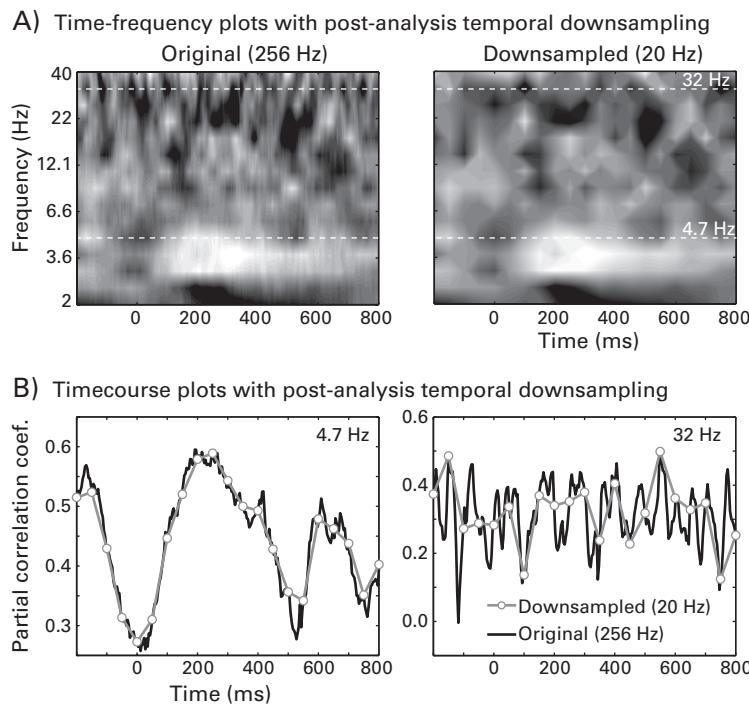
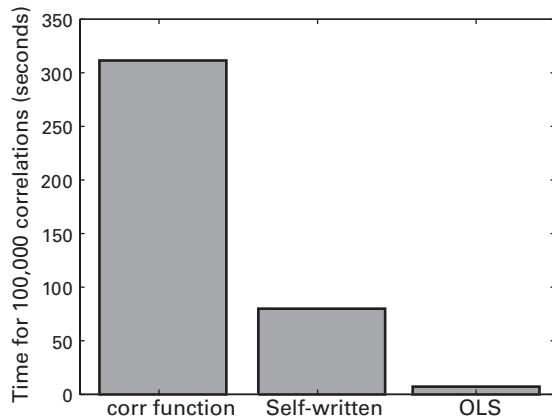


Figure 27.8

Examples of results before and after temporal downsampling. Note that the data were not downsampled; rather, the time-frequency decomposition was performed on the original sampling rate (in this case, 256 Hz), and then the results were temporally downsampled to 20 Hz (one data point every 50 ms; this is lower than the recommended downsampling of 40–50 Hz and was done to highlight the loss of information at higher frequencies). Panel A shows time-frequency plots before and after downsampling. Panel B shows time courses of connectivity at a lower (left) and a higher frequency (right) (see also white dashed lines in panel A) before and after downsampling. Because temporal autocorrelation decreases with increasing frequency, downsampling will result in more information loss at higher frequencies compared to lower frequencies. How much information is lost also depends on the time-frequency decomposition parameters—analyses associated with decreased temporal precision (for example, wavelets with many cycles) suffer less from postanalysis temporal downsampling.

**Figure 27.9**

Writing the Spearman correlation is considerably faster than the Matlab `corr` function, particularly with many tests (in this case, 100,000) and long sequences (in this case, 1000 data points per variable). Using ordinary least-squares (OLS) fitting is even faster.

example, the data contain no NaNs or complex numbers). One option for computing fast correlation coefficients is to write the Spearman correlation yourself in Matlab. It is much faster (figure 27.9) and gives you the same result as using the `corr` function. The short-cut formula for the Spearman correlation is presented in equation 27.5

$$1 - \frac{6 \sum_{t=1}^n (x_t - y_t)^2}{n(n^2 - 1)} \quad (27.5)$$

where x and y are the power time series (or trial series, if you are computing the analysis over trials) and n is the number of time points or trials. You must first rank-transform the data using the Matlab function `tiedrank`. Note that equation 27.5 assumes that there are no ties; if there are ties, the same formula for a Pearson correlation would be used. However, with power data this is not a major concern for two reasons. First, power data take so many possible values that there are typically no ties. Second, for a large number of data points, having ties makes no appreciable difference—with 1000 data points, for example, having a few ties will change the correlation coefficient between equation 27.5 and equation 27.1 on the order of 10^{-9} . Even with only 10 data points, the difference in correlation coefficient between applying equation 27.5 and equation 27.1 with a tie present in the data is around 0.005 (this is not a p -value, this is the difference of correlation coefficients).

If you are really in a hurry and do not have the patience to wait for equation 27.5, try performing a least-squares fit on the data. Particularly if you can work with unstandardized regression coefficients instead of scaled correlation coefficients (this is possible when used in combination with permutation testing, as discussed in chapters 33 and 34), least-squares fitting is by far the fastest method (around 50 times faster than the `corr` function and around 10 times faster than writing out the Spearman correlation; figure 27.9). Matlab code for computing an ordinary least squares fit is presented in the online Matlab code, and the equation and its use in single-trial regression are discussed more in chapter 34.

27.6 Describing This Analysis in Your Methods Section

Similar to phase-based connectivity methods, there are many possibilities for power-based measures of connectivity. Provide a justification for why you applied one method for computing power-based connectivity over other methods and include a clear description of the analysis performed. Describe the method clearly and state the frequencies, time segment lengths, and other relevant parameters. State whether you used Pearson or Spearman correlation, and, if you used Pearson correlation, provide a justification for not using the Spearman rank correlation.

27.7 Exercises

1. Perform a power correlation analysis over time. Pick two electrodes and use a sliding time segment of three cycles (1.5 cycles on either side of each center time point). Average the results over trials. Perform this analysis at three frequencies and plot the time series of correlation coefficients. Next, repeat the analysis twice, using fixed time-segment lengths of 150 ms and 900 ms. Do the results differ according to the time segment length and the frequency band, and how are they different? In what situations would it be beneficial to use each window width parameter, and in what situations might problems or limitations arise?
2. Select two “seed” time-frequency-electrode windows and perform an exploratory power correlation over trials at one selected “target” electrode, as in figure 27.6C (plate 19). Show the results in separate plots, and then show a time-frequency plot of correlation coefficient differences between the two seeds (Fisher-Z transform the coefficients before subtraction). Are there any striking qualitative differences between the two plots, and did plotting the difference map make the differences easier or more difficult to interpret?

28 Granger Prediction

First, a note on terminology. You probably have heard the term “Granger causality” or “Wiener-Granger causality” but perhaps not the term “Granger prediction.” Results from Granger causality analyses neither establish nor require causality. Granger causality results do not reveal causal interactions, although they can provide evidence in support of a hypothesis about causal interactions. This puts scientists in the awkward position of using the term Granger *causality* and then clarifying (typically, several times) that this method should not be interpreted as identifying causality. Therefore, the term “Granger prediction” is used here. Prediction is a brief description of the analysis and not an interpretation of the results (see sections 21.1 and 37.12 for more discussion on terminology).

Granger prediction is an established measure of directed functional connectivity in many fields including economics and neuroscience (Bressler and Seth 2011; Granger 1969) and answers the question: Can you predict the activity measured by electrode A *now* if you know the activity measured by electrode B *in the past* (and is this better than knowing only the past of A)? If there is statistical support for a “yes,” you can say that there is a Granger predictive effect of B on A. Unlike some other measures of connectivity discussed in this book, Granger prediction is based on multivariate autoregression rather than a frequency-domain transformation such as wavelet convolution or filter-Hilbert.

Extreme temporal downsampling can be detrimental to Granger prediction, and can potentially lead to spurious or even opposite-direction estimates (Florin et al. 2010; Seth, Chorley, and Barnett 2013). Thus, you should perform Granger prediction analyses on the original high-sampling-rate data and then optionally downsample the results to save disk space or reduce computation time (as discussed in section 27.5). On the other hand, if the sampling rate is higher than the temporal precision of the brain regional interactions, there will be more parameters to fit in the autoregression models (more on this point later). Thus, the temporal resolution of the data should match (or be slightly higher than) the temporal

precision of the expected brain connectivity dynamics. In general, sampling rates between 250 Hz and 1000 Hz will be appropriate.

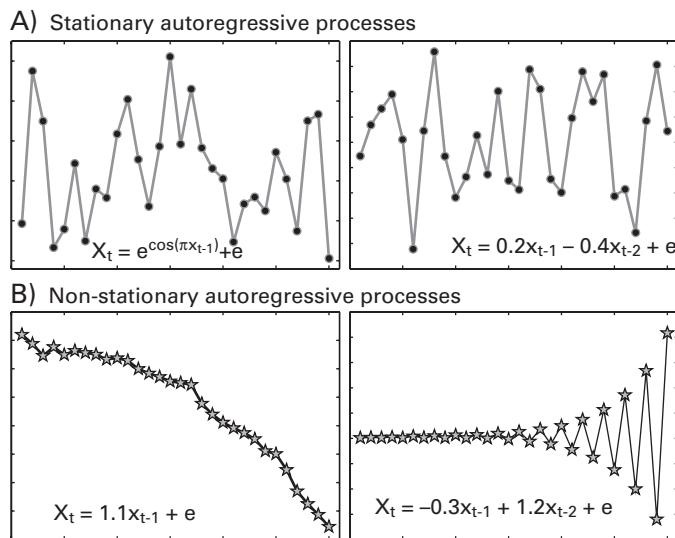
28.1 Univariate Autoregression

To learn about Granger prediction, it is useful to learn first about the basics of uni- and bivariate autoregression. The term “univariate autoregression” may be a mouthful, but by considering its linguistic components, you can guess that it involves a regression of one variable on itself. Consider, for example, the expression $X_t = 0.8X_{t-1}$. This expression describes a univariate autoregressive process because it contains one variable (X) that changes according to weighted previous values of itself. In this case, previous values are weighted by 0.8. Another example is the expression $X_t = 0.8X_{t-1} - 1.3X_{t-2}$. The difference between these two expressions is that the former changes as a function of one previous value, whereas the latter changes as a function of two previous values. Thus, the first expression is of order 1 and the second expression is of order 2; this is also written AR(1) and AR(2), where AR stands for autoregression. The order indicates how far back in time the model incorporates data and is measured in units of time steps. This means that if your data are sampled at 500 Hz, a model with an order parameter of 5 means the autoregression considers a weighted sum of the previous 10 ms. More generally, univariate autoregressive processes can be expressed using equation 28.1.

$$X_t = \sum_{n=1}^k a_n X_{t-n} + e_{xt} \quad (28.1)$$

In equation 28.1, X is the variable for which current values are predicted, t is time point, k is the order, a is the vector of the autoregression coefficients or the weighting terms, and e is the error or the residual that cannot be predicted from aX_{t-n} (the subscript x indicates that this error term is from the univariate model, which will help dissociate it from the error from the bivariate models, as shown in the next section). In the AR(2) expression in the previous paragraph, a_1 and a_2 would be 0.8 and -1.3, respectively.

Time series data can be stationary or nonstationary. As discussed in chapter 11, a signal is stationary if its statistical properties such as mean and variance do not change over time; in other words, if the time series data are “well behaved.” Autoregressive modeling assumes that the time series data are stationary, and thus, stationarity is important for Granger prediction. Figure 28.1 shows a few examples of stationary and nonstationary time series created from univariate autoregressive expressions.

**Figure 28.1**

Examples of time series generated by univariate autoregressive equations (units are arbitrary). Panel A shows stationary time series (the mean and variance do not change over time), and panel B shows non-stationary time series. These time series are nonstationary because their mean (left panel) and variance (right panel) change over time. As discussed later in this chapter, nonstationary time series should not be used with Granger prediction. e = random noise.

28.2 Bivariate Autoregression

A bivariate autoregression is similar to a univariate autoregression, except that there are two variables. Consider the following pair of expressions.

$$X_t = 0.8X_{t-1} - 0.3X_{t-2}$$

$$Y_t = -0.4Y_{t-1} + 1.3X_{t-1}$$

In this example, X is a function of previous values of X (a univariate autoregression), and Y is a function of previous values of Y and previous values of X . Thus, the first expression reflects a univariate autoregressive process, whereas the second expression reflects a bivariate autoregressive process. Bivariate autoregressive expressions can be characterized using the following equations, which are extensions of equation 28.1.

$$X_t = \sum_{n=1}^k a_n X_{t-n} + \sum_{n=1}^k b_n Y_{t-n} + e_{xyt} \quad (28.2)$$

$$Y_t = \sum_{n=1}^k c_n Y_{t-n} + \sum_{n=1}^k d_n X_{t-n} + e_{yxt} \quad (28.3)$$

You can see that equations 28.2 and 28.3 are similar to equation 28.1 except that they include additional terms to account for the influence of the other variable. Note that the error terms here are e_{xy} and e_{yx} . This indicates that the error terms are from a bivariate model in which (for equation 28.2) previous values of X are predicted from previous values of X and from previous values of Y . Equation 28.1 is sometimes referred to as the restricted model because the model is restricted to one variable, and equations 28.2 and 28.3 are sometimes referred to as the unrestricted model because the model contains two variables. Granger prediction, and autoregression functions more generally, are not limited to two variables, but this chapter discusses only bivariate Granger prediction.

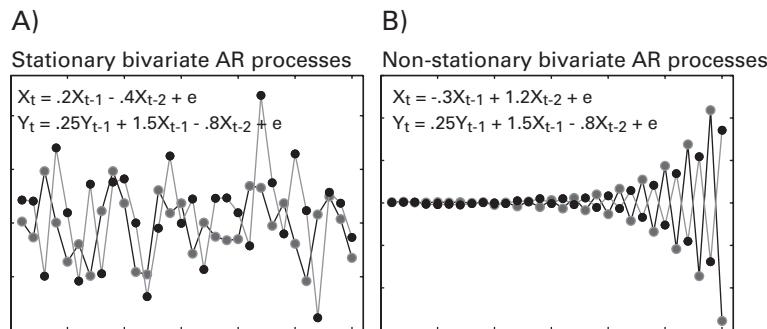
There are three features to note about equations 28.2 and 28.3 that will become relevant for Granger prediction. First, each equation now contains two weighting coefficient vectors: a and b , and c and d . These can be different from each other, which means that the influence of X on Y can be distinct from the influence of Y on X . In the expressions given at the beginning of this section, b would be zero, because there is no influence of previous values of Y on current values of X .

Second, X and Y have their own error terms, and there is an error term for each time point. This is important for evaluating Granger prediction.

Third, for expressions with an order greater than 1, the coefficients a , b , c , and d are vectors that contain as many coefficients as the order parameter. This will become relevant for spectral Granger prediction. Figure 28.2 shows a few examples of time series created through bivariate autoregressive expressions.

Now consider equations 28.2 and 28.3 in the context of directed connectivity. Imagine that a signal is propagated from brain region X to brain region Y . This would result in the d weightings being nonzero (whether they are positive or negative describes only the polarity of the influence and is not relevant for answering the question *whether* there is any influence). Imagine further that there is no signal propagating from Y to X . This would result in the b weightings in equation 28.2 being exactly zero, which would effectively remove Y from that equation. In this scenario there would be no difference between the bivariate and univariate autoregression models (equations 28.1 and 28.2) for variable X . This is an important concept because the measure of Granger prediction is related to how well the bivariate model fits the data compared to the univariate model.

In the examples in figures 28.1 and 28.2, the autoregression coefficients were specified. In real data, these coefficients are not known and must be estimated from data. There are

**Figure 28.2**

Examples of pairs of time series generated by bivariate autoregressive expressions. In both cases variable X is a function only of itself and random noise, whereas Y is a function of itself and previous values of X and noise. As with figure 28.1, panel B shows nonstationary data.

several algorithms for estimating autoregression coefficients. The online Matlab code uses the function `armorf.m`, which comes with the BSMART toolbox (Cui et al. 2008) and is provided in the online Matlab code. Other methods of estimating autoregression coefficients may provide slightly different results, although different methods should have little impact on Granger prediction estimates.

28.3 Autoregression Errors and Error Variances

In real data there is considerable unexplained variance, and current values of X cannot be fully predicted only by previous values of X and previous values of Y . Unexplained variance is taken into the e terms in equations 28.2 and 28.3. The error terms have an inverse relationship with the fit of the model: if the model is a good fit to the data, the errors will be relatively small. Thus, if there are several models that are fit to the same data, the model with the smallest errors can be said to fit the data better than the other models tested. This is an intuitive point, but it is useful to mention explicitly because the errors are crucial for computing Granger prediction.

When there is no influence of Y on X , the coefficients in b are all zero, which means that equation 28.2 is identical to equation 28.1. And this means that their errors will be the same. Now imagine that variable Y does indeed have some temporally lagged influence on variable X . The coefficients in b are nonzero, which means that the bivariate model (equation 28.2) will fit the data better than the univariate model (equation 28.1). Thus, the errors from the bivariate model will be smaller than the errors from the univariate model. The measure of

Granger prediction is the relative size of the errors from the two models—a univariate model in which current values of X are predicted only from past values of X , and a bivariate model in which current values of X are predicted both from past values of X and from past values of Y (Granger 1969).

Remember that there are errors for each time point. How can the error vectors between the univariate and bivariate autoregression models be compared? You cannot compare the average of the errors because errors are (ideally) random and have a mean of zero. Consider that if the model is a good fit to the data, the errors will be small and will therefore have relatively little variability, whereas if the model is a poor fit to the data, the errors will be large and will therefore have relatively high variability. Thus, the variance of the errors from the bivariate autoregression model is compared with the variance of the errors from the univariate autoregression model. This comparison is taken as the logarithm of the ratio of the error variances and is the mathematical definition of Granger prediction. The logarithm of variances is a convenient metric because, among other features, it is distributed as a chi-square function, which becomes useful for statistical evaluation (Seth, Chorley, and Barnett 2013).

$$\text{GrangerPrediction} = \ln\left(\frac{\text{var}(e_x)}{\text{var}(e_{xy})}\right) \quad (28.4)$$

The e_x comes from equation 28.1, and the e_{xy} comes from equation 28.2. This is the Granger prediction for $Y \rightarrow X$; Granger prediction for $X \rightarrow Y$ is the same, but the error variances are replaced by those from the uni- and bivariate autoregressions for Y (equation 28.3). From equation 28.4, it should be clear that if Y does not contribute to X , and thus the errors and error variances are the same for the uni- and bivariate models, the ratio of the error variances will be 1, the natural logarithm of 1 is 0, and thus the Granger prediction of $Y \rightarrow X$ will be zero. As the bivariate error term variance decreases (because previous values of Y help predict current values of X), the error variance ratio increases, and thus so does the estimate of Granger prediction. In practice, this ratio is generally always greater than zero simply because regression models with more predictors will fit data better than regression models with few predictors. If the Granger prediction estimate is less than zero, it is possible that the model was a poor fit to the data. If you observe negative Granger prediction values, you should closely inspect the data and the model results, in particular to check for violations of stationarity (how to deal with nonstationary time series is discussed in section 28.7).

28.4 Granger Prediction over Time

The previous section defined Granger prediction for one time segment. There are two reasons for computing Granger prediction repeatedly over successive time segments. First, if you

have a task-related design, you will likely want to know whether there are changes in connectivity over time as a function of task events. Second, computing Granger prediction in successive time segments helps ensure that the data are stationary within each time segment. Thus, in practice for EEG analyses, Granger prediction can be computed repeatedly in sliding segments of, say, 200 ms. At each step, Granger prediction is computed for that 200-ms time segment, and then Granger prediction is computed on the next (overlapping or nonoverlapping) time segment.

How long should the time segments be? There are advantages and disadvantages of both relatively long and relatively short segments. The advantage of long segments is that more data are used to fit the autoregression models, which means the parameter estimates (and thus, the error variances) will be more stable and less influenced by noise. The two disadvantages are that longer time segments are more likely to contain time series nonstationarities and that longer time segments have lower temporal precision for isolating task-related dynamics. For example, if the true directed connectivity lasts for 100 ms but the time segment used for Granger prediction analysis is 1000 ms, there is a danger that the true directed connectivity will not be detected because it is averaged together with 900 ms of no directed connectivity. The advantages and disadvantages of shorter time segments are the reverse of those for longer time segments: shorter time segments provide greater temporal precision of the results and therefore higher sensitivity to detect transient effects, and shorter time segments are more likely to contain stationary data. However, the autoregression model parameters can be overly influenced by noise or nonrepresentative data if too little data are used to estimate the parameters. In general, having many trials in your experiment will help improve the estimation of model parameters because the autoregression coefficients can be estimated from short time segments over many trials (via the function `armorf`).

Figure 28.3 shows an example of time-resolved Granger prediction between electrodes F5 and O1.

Published studies vary on the length of the time segments, ranging from as little as 50 ms (Ding et al. 2000) to 100 ms (Hesse et al. 2003) to 300–400 ms (Cohen and van Gaal 2012; Gaillard et al. 2009; Protopapa et al. 2011) up to 1 s (Nicolaou et al. 2012) or 2 s (Barrett et al. 2012). Because the length of the time segment will affect the estimate of Granger prediction, it is important that whatever time segment length you choose is used consistently for all conditions, electrode pairs, and subjects.

28.5 Model Order

There are advantages and disadvantages of small and large model orders. Models with a small order have fewer parameters to estimate, which means that the estimation of those

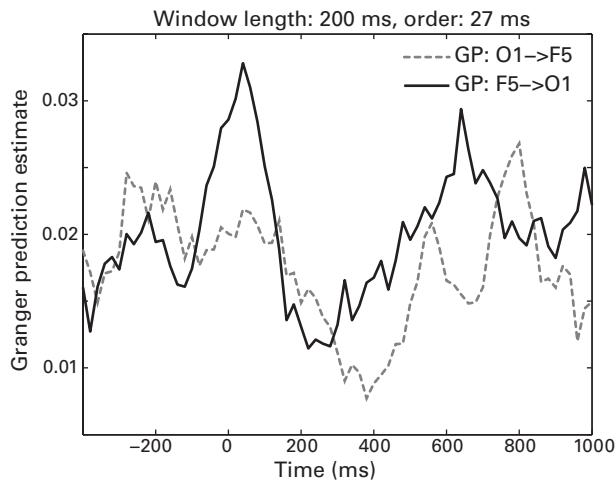


Figure 28.3

Granger prediction (GP) over time between electrodes F5 and O1. The ERP was subtracted from the single-trial data to minimize the possibility of violating stationarity around the time of the ERP (this strategy is discussed in section 28.7). However, the results were very similar with and without removing the ERP (not shown here, but you can try this yourself using the online Matlab code). This indicates that, for this result, the presence of the ERP would not have produced any spurious results.

parameters (e.g., via the Matlab function `armorf.m`) will be more robust to noise. On the other hand, models with a small order are insensitive to longer time lags and thus may fail to detect true interactions that have a long temporal lag. Models with a larger order are sensitive to longer time lags and will allow you to extract low-frequency interactions (discussed further in section 28.6) but have more parameters to estimate and thus require more data. Thus, with larger model orders, more trials and longer time segments are helpful. This is the main reason why, as mentioned at the beginning of this chapter, the temporal resolution of the data should be close to the temporal precision of the expected interactions. If you recorded the data at 2000 Hz, downsampling to 250 Hz or 500 Hz will reduce the model order (and thus improve the fit of the model) while preserving the time scale of the expected functional interactions (tens of milliseconds) (Barrett et al. 2012).

There are several statistical tests that will help guide an appropriate selection of model order. The most commonly used tests are the Bayes information criterion (BIC) and the Akaike information criterion. The BIC is recommended for datasets with a relatively large number of data points, which is often the case for EEG data (Seth 2010a). To determine the optimal model order, the autoregression model is computed repeatedly with a range of

orders, and the BIC is computed based on the error variances, the order, and the number of time points, according to the following equation:

$$BIC = \ln(\det(E)) + (2^2 m \ln n) n^{-1} \quad (28.5)$$

in which E is the error matrix that results from fitting the autoregression, \ln refers to the natural logarithm (in Matlab, the function `log`), \det is the matrix determinant (sometimes, the determinant is indicated using the same vertical bars as are used for the absolute value, or the magnitude of a complex number; in Matlab, the function is `det`), m is the model order, and n is the number of time points. The “2” in the equation refers to the number of electrodes in the model. For bivariate Granger prediction, this number will always be 2.

In practice, however, determining the optimal model order for task-related data is not straightforward for three reasons. First, it is possible that models with several different orders fit the data fairly equally well. Second, it is possible that different statistical criteria provide slightly different recommendations for the optimal model order. Finally, it is unlikely that one model order will be deemed optimal for all time segments, trials, electrode pairs, conditions, and subjects.

This is illustrated in figure 28.4. The BIC was evaluated on the data shown in figure 28.3. Figure 28.4A suggests that an order of seven (27 ms) is optimal for this dataset and this electrode pair because that point corresponded to the lowest BIC. However, it can also be

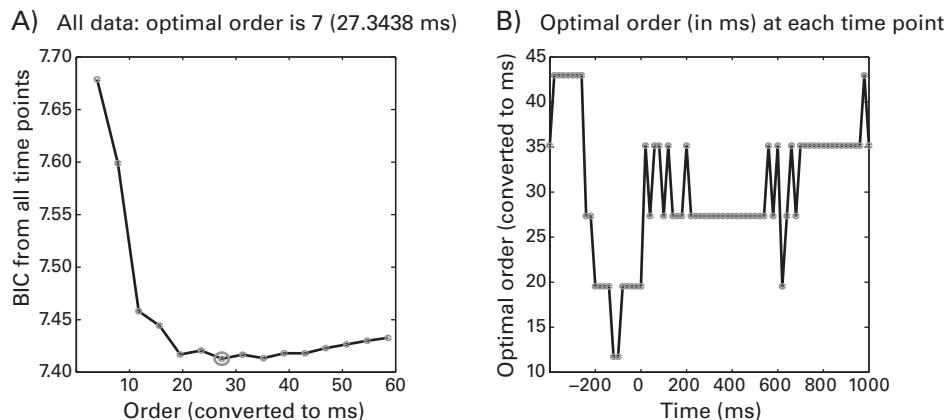


Figure 28.4

Optimal model order as suggested by the Bayes Information Criterion (BIC). Panel A shows BIC averaged over all time points, and panel B shows optimal order (defined as the order corresponding to the smallest BIC) for each time segment separately (200 ms surrounding each time point plotted).

seen that a range of values produced a fairly similarly small BIC, ranging from 19 to 60 ms. Furthermore, in time-varying Granger prediction, each time segment of the data is modeled separately, and thus, the model order can be estimated separately for each time segment. As shown in figure 28.4B, in this dataset with this electrode pair, the optimal order differed across time windows. Different electrode pairs gave slightly different results for the optimal order (you can try this yourself in the online Matlab code), although they were generally in the range of five to nine (19–43 ms).

In general, and as the results in figure 28.4 highlight, there is no optimal order for all electrode pairs, conditions, and time segments. Although the change in optimal model order might be an interesting phenomenon on its own (for example, if this can be linked to temporal integration of information or the amount of information that must be transferred from one brain region to another), using different orders for each time window can affect the estimate of Granger prediction. Thus, in practice, if it is possible to use BIC to determine the optimal model order for the entire dataset, do so—this would be the case when only a small amount of data are available. But with hundreds or thousands of trials, several conditions, and many subjects, statistical guidelines such as the BIC will help you select an appropriate order from within a range, but you should treat the “optimal” order as a statistically guided recommendation rather than an absolute rule. For example, from figure 28.4, it seems that order parameters between 20 ms and 45 ms are appropriate for this dataset. Because the order parameter may affect the results, when you select a model order, you should apply that order to all conditions, time segments, electrode pairs, and subjects.

When implementing Granger prediction in Matlab, keep in mind that although it may be useful to think about the model order in terms of time in milliseconds, the order parameter must be converted to time in samples (in the convenient case of 1000 Hz sampling rate, they are identical).

28.6 Frequency Domain Granger Prediction

If you have hypotheses about directed connectivity in specific frequency bands, you might think that the way to test them would simply be to bandpass filter the data and then apply Granger prediction to the filtered data. Unfortunately, this approach is inappropriate: Granger prediction is unaffected by filtering except that the autoregressive model fitting will be worse because the filtering will modify data smoothness and will result in a large increase in the required model order (Barnett and Seth 2011). This can lead to incorrect results (Florin et al. 2010). (Note that filtering the data to remove noise and artifacts such as electrical line noise is still a good idea and can facilitate the data becoming stationary.) Instead, localizing

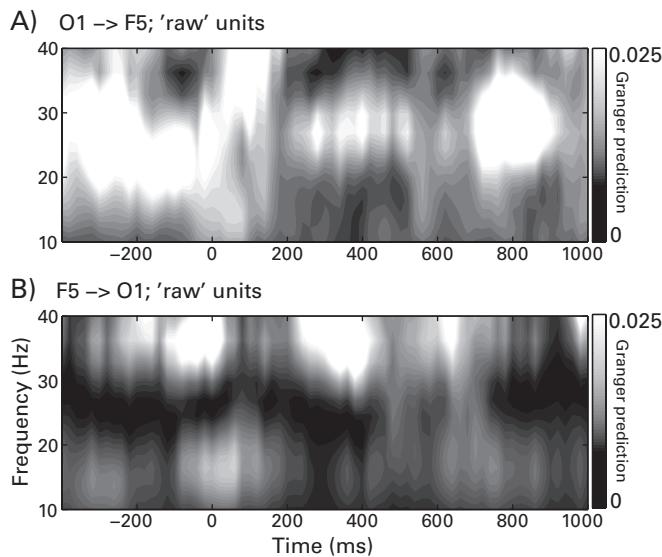


Figure 28.5

Time-frequency plots of spectral Granger prediction between F5 and O1. As with figure 28.4, Granger prediction here was computed using the data with the ERP subtracted. The results were similar with and without subtracting the ERP. A color version of these results can also be seen in figure 28.8 (plate 21).

Granger prediction in the frequency domain is done via a method proposed by Geweke (Dhamala, Rangarajan, and Ding 2008; Geweke 1982). This method involves computing dot products between the autoregression coefficients and complex sine waves, and then applying those results to the error variance via a transfer function. This is similar to a Fourier transform, although the frequencies can be specified by you rather than being determined by the number of sample points. The method is presented in the online Matlab code, which is adapted from the BSMART toolbox (Cui et al. 2008). Figure 28.5 shows the time-frequency Granger prediction results for the data shown in figure 28.3.

Although it is possible to specify any frequencies you want, in practice, the frequency resolution is determined by the model order. Imagine that the model order were 1. In this case, any frequency you wanted to extract would contain only one time point, in other words, far less than one full cycle. Having a higher order parameter means that there will be more cycles per frequency, which will improve the accuracy of the frequency-specific connectivity results (figure 28.6). If there is less than one full cycle, the accuracy of the frequency specificity is decreased, and the results should be interpreted cautiously, for example, “low-frequency Granger prediction” rather than “8.452-Hz Granger prediction.” This may seem

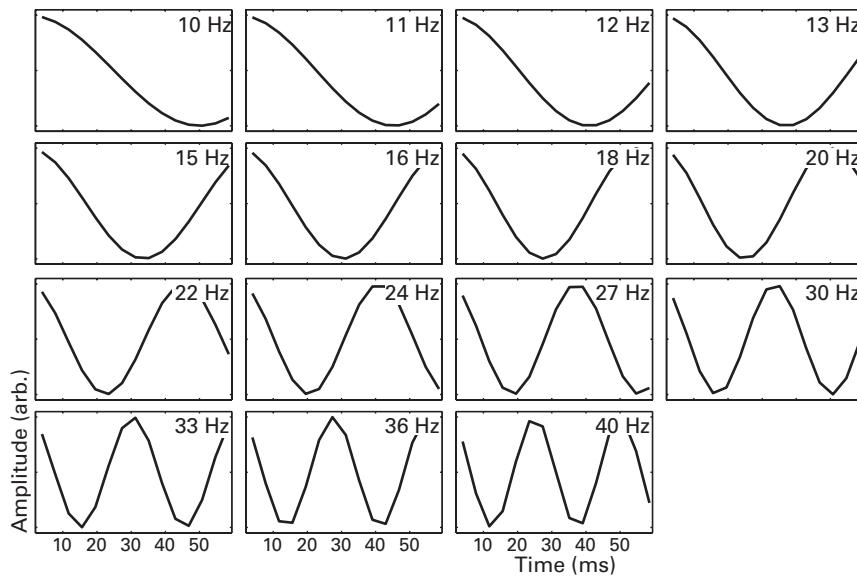


Figure 28.6

The order parameter for the autoregressive model in part constrains the frequency precision of spectral Granger prediction. Shown here are sine waves at the frequencies used in figure 28.5. These are the sine waves that are multiplied by the autoregression coefficients to extract frequency-band-specific Granger prediction results (in practice, complex sine waves are used to account for phase differences; real-valued sine waves are plotted here for convenience). Because the order parameter was 58.6 ms, just over one-half of one cycle can be extracted at 10 Hz. Thus, in this case, the accuracy of the frequency specificity of the Granger prediction results is low for frequencies below 20 Hz.

at odds with recommendations of previous chapters for using at least three cycles at each frequency. The difference is that with other methods of time series analysis, an arbitrary number of time points can be used, but with Granger prediction, having too many or too few time points will lead to poor autoregression coefficient estimates. Thus, there is a trade-off between an appropriate order for Granger prediction and the accuracy of low-frequency spectral Granger prediction.

Thus, for spectral Granger prediction, you should lean toward having a higher order parameter, although not so high as to make the model a poor fit to the data. In figure 28.5 an order of 15 was used (58.6 ms), which figure 28.4A suggests is a reasonable order for this dataset. This is one reason why downsampling the data before computing a spectral Granger prediction might be useful. That is, if the original data are sampled at 1000 Hz, a model with 50-ms order requires $50 \times N$ parameters to be estimated; in contrast, if the data are downsampled to

250 Hz, a model with 50-ms order requires $12 \times N$ parameters to be estimated. It is possible to use a different order parameter for time-domain and frequency-domain Granger predictions as long as the order for each analysis is used consistently for all analyses. In this case, results from time- and frequency-domain analyses are not directly compared against each other.

28.7 Time Series Covariance Stationarity

Stationarity is a statistical term that refers to a time series having the same statistical properties over time. For example, if a time series contains a slow trend such that values increase over time, or if the variance of the time series changes over time, the time series is nonstationary (see figures 28.1 and 28.2 for examples). Stationarity is an assumption of autoregression model estimation. This means that if your data are not stationary, the Granger prediction analysis might produce biased results. For example, in a poststimulus window of 100–400 ms, when there is an ERP, stationarity may be violated (Wang, Chen, and Ding 2008).

There are several simple methods to help make your data stationary. Detrending and z-normalization (subtract the mean and divide by the standard deviation) will help. Using shorter time segments will also increase the chances of having stationary data, particularly in combination with detrending and z-normalization. Subtracting the ERP from single trials (as was done to compute non-phase-locked time-frequency power, see chapter 20) may also improve stationarity (Ding et al. 2000), assuming that the ERP is present on each trial (Wang, Chen, and Ding 2008). Keep in mind that because the amplitude of the ERP is much smaller than the amplitude of single-trial EEG activity (see figure 9.1), subtracting the ERP might have only a modest effect on the single-trial dynamics. This is shown in figure 28.7. Another option is to apply Granger prediction to the derivative of the time series, that is, the difference between activity at each time point and the previous time point.

There are statistical methods to evaluate whether your data are nonstationary. The Granger Causal Connectivity Analysis toolbox (Seth 2010a) contains two such tests (function names are `cca_kpss` and `cca_check_cov_stat`), which can be computed on the portion of data used in each time segment prior to computing Granger prediction. If there is evidence for nonstationarity in many epochs, consider taking more stringent measures to ensure data stationarity such as using shorter time segments or taking the derivative of the signal. If a minority of time segments exhibit nonstationarity and you have many trials, the minor violations of stationarity are unlikely to have severe negative effects on the Granger prediction results. If many trials contain nonstationary data at a particular time window (which might occur, for example, during the ERP), you could avoid interpreting Granger prediction

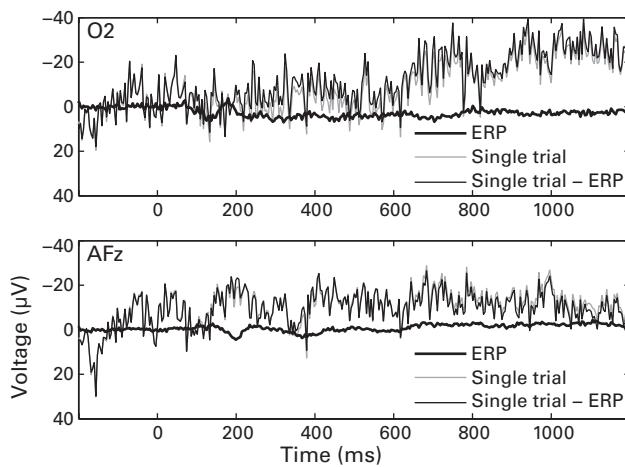


Figure 28.7

Example single trial data from two electrodes before and after removing the ERP. This figure illustrates two concepts. First, subtracting the ERP has a relatively small impact on the single-trial time series because the amplitude of the ERP tends to be around an order of magnitude smaller than the amplitude of single-trial EEG data. Second, whether subtracting the ERP has any appreciable effect on the single-trial data depends on the extent to which the single-trial data contain phase-locked activity. In this case, subtracting the ERP has a small but noticeable effect on the activity from one trial at electrode O2 and a smaller effect on the activity from one trial at electrode AFz (indeed, it is difficult to distinguish the gray and black lines in the lower plot). This result indicates that this trial contains little phase-locked activity, and it also suggests that subtracting the ERP will have little impact on measures of stationarity on this trial.

results from that time window. In general, it is safest to interpret Granger prediction results from periods of time in which the data will likely be stationary, such as time periods between experiment events or during long stimulus presentation times (after the first few hundred milliseconds when there is an ERP).

28.8 Baseline Normalization of Granger Prediction Results

Although Granger prediction results have an interpretable value that is unaffected by power-law scaling, it may be useful to perform a baseline normalization of Granger prediction results when comparing conditions, electrode pairs, or connectivity directions. The baseline normalization can be done by subtracting the Granger prediction results from the baseline time period or computing percentage change from the baseline period. Baseline normalization

will facilitate interpreting task-related changes in Granger prediction, particularly in situations in which there is a tonic or background level of directed connectivity that is asymmetric from region A to region B.

The advantages and disadvantages of baseline normalization were discussed in chapter 18. Interestingly, in figure 28.8 (plate 21) you can see a situation in which performing a baseline normalization will affect the interpretation of the results. The “raw” Granger prediction results showed an increase in connectivity of O1 → F5 in the pretrial period (figure 28.8C1; this may have been due to stimulus anticipation and preparation, or it may have been due to a tonic level of directed connectivity when a task was not being performed), which decreased during the trial period between 100 and 700 ms. The baseline-normalized results (figure 28.8D1) show a mild task-related decrease in directed connectivity from around 200 to 700 ms. In contrast, computing the percentage change of the F5 → O1 directed connectivity (cf. figure 28.8C2,D2) revealed task-related increases in directed connectivity in the beta band (20–30 Hz). These results are present in the data shown in figure 28.8C2 but are more difficult to observe because of the color scaling. Thus, the percentage change transform facilitated an interpretation of the patterns of the connectivity that are specifically task related. Note also that, in this case, the percentage change results are visually more consistent with the time-domain results.

28.9 Statistics

There are two approaches to performing statistical analyses of Granger prediction results. The first approach is to determine whether the prediction from Y to X (that is, the result of equation 28.4) is significant in itself. As mentioned earlier, Granger prediction results generally follow a chi-square distribution, and the statistical significance can thus be evaluated with an *F*-statistic (Seth 2010a). Routines to compute this *F*-statistic can be found in Matlab Granger causality toolboxes.

Statistical significance can also be obtained via permutation testing. This approach can be applied to time-domain or to frequency-domain Granger prediction results. It is particularly useful for the frequency domain because the underlying distribution of spectral Granger prediction results is not fully known. As discussed in chapters 32 and 33, two advantages of nonparametric permutation testing that are relevant for Granger prediction are that nonparametric permutation testing does not require assumptions regarding data or parameter distributions and that it provides a convenient framework for correcting for multiple comparisons over time points or time-frequency points. For permutation testing of Granger prediction, you could permute the order of the time segments within trials. Note that permuting trial

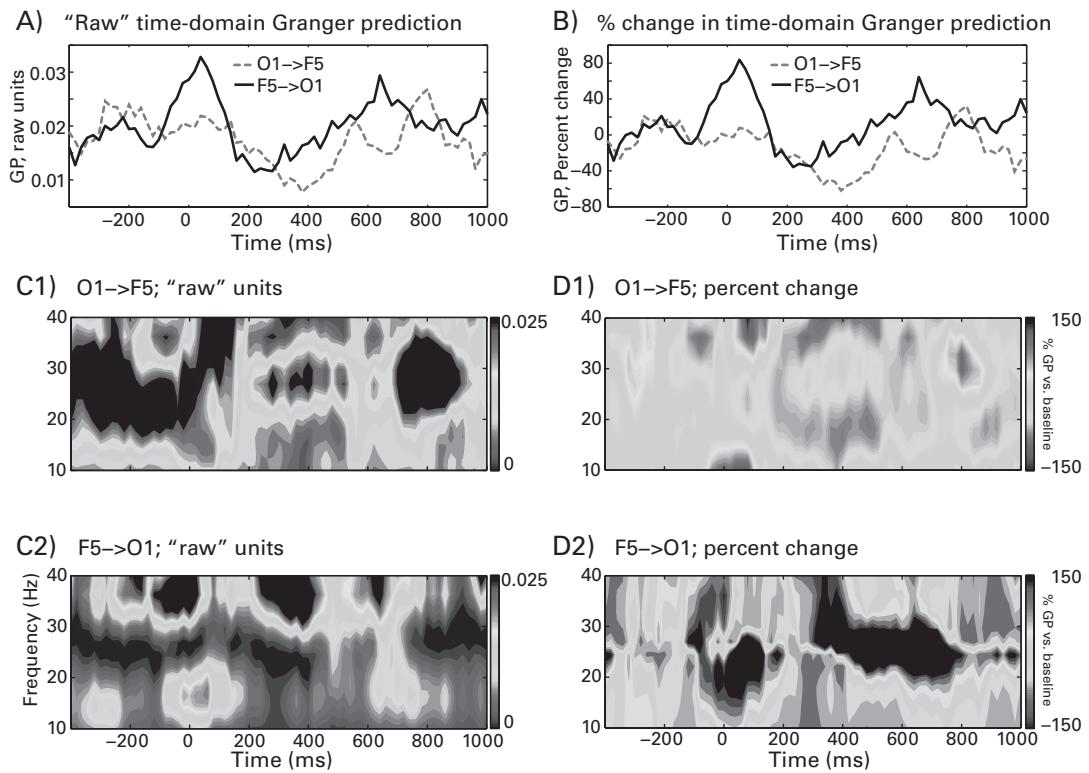


Figure 28.8 (plate 21)

Comparison of Granger prediction in "raw" units (log of error variance ratio, equation 28.4; panels A and C, replotted here from figures 28.3 and 28.5) and percentage change from baseline (panels B and D). Note that because of the pretrial increase in O1 → F5 directed connectivity in panel C, the task-related changes in connectivity appear as a relative decrease. This highlights the importance of a careful inspection of the results before and after baseline transformation so that the results are interpreted appropriately.

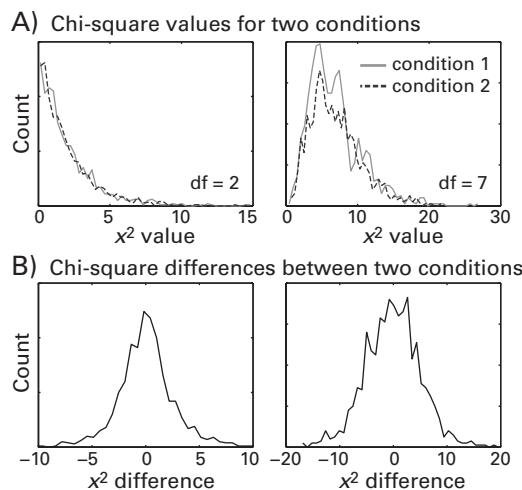


Figure 28.9

Granger prediction results generally follow a chi-square distribution (randomly generated numbers from chi-square distributions with different degrees of freedom are shown in the top row), and thus, condition differences in chi-square-distributed values can approach a normal distribution (bottom row).

order while leaving time segments intact might be suboptimal if the directed connectivity effect is strongly time-locked to the time = 0 event; in this case, the time-locked component will be similar regardless of trial order.

The second approach to statistical analyses of Granger prediction results is to focus on differences in Granger prediction results over time or across conditions or electrode pairs. Because Granger prediction results are chi-square distributed, condition difference values (that is, X → Y in condition A minus X → Y in condition B) are approximately normally distributed (see figure 28.9). Condition or direction differences in Granger prediction can also be assessed using nonparametric permutation testing by shuffling condition labels within or across subjects. Nonparametric approaches for evaluating time-frequency Granger prediction should be preferred over parametric approaches, particularly in exploratory analyses for which corrections for multiple comparisons are necessary.

28.10 Additional Applications of Granger Prediction

Presented in this chapter is what you need to know for the basic application of bivariate Granger prediction (between two electrodes) in the time and frequency domains. There is continual development of Granger prediction as an analysis framework in neuroscience, and

there are more advanced applications than those presented here. For example, multivariate Granger prediction allows assessment of larger networks (Chen, Bressler, and Ding 2006), including networks across individuals (Schippers et al. 2010) and tests of mediation and conditional interactions (e.g., whether the effect of brain region A on brain region B is mediated by brain region C). Granger prediction may also be used to evaluate “autonomy” and “emergence” of a system (Seth 2010b).

Two Matlab toolboxes that are designed for applying Granger prediction to EEG and LFP data are BSMART (Cui et al. 2008) and the Granger Causal Connectivity Analysis toolbox (Seth 2010). These toolboxes will perform basic Granger analyses like those presented in this chapter and can also be used for more advanced applications.

28.11 Exercises

1. How would you interpret panels A and C versus B and D in figure 28.8 (plate 21)? Discuss advantages and disadvantages of baseline-normalized Granger prediction results for a study investigating the role of visual attention in healthy university students and in a study comparing visual attention in children with ADHD versus typically developing controls.
2. Pick one electrode and one time segment and compute Granger prediction between that electrode (the “seed”) and all other electrodes in that time segment. Before selecting a time segment, examine the ERP from that electrode and choose a time window that, based on the ERP, is likely to contain stationary data. Justify your selection of time segment and model order. Show the results in a topographical map and comment on any striking or salient features you observe.
3. From the electrode and time segment used above, recompute Granger prediction separately for the first 40 trials and the last 40 trials. Make topographical maps of the early and late experiment effects and their difference. Are there any notable topographical features that can be seen in the difference topographical map? If so, how would you interpret these effects if they were statistically significant (there is no need to perform statistics; base your answer on qualitative visual inspection)?

29 Mutual Information

Mutual information and related concepts including entropy, joint entropy, and conditional entropy form a set of mathematical/statistical techniques that have many uses in science, engineering, and information communication. These techniques are less widely used in cognitive neuroscience, although they have some advantages for cognitive electrophysiology, particularly for connectivity analyses. Mutual information is a simple but robust framework for quantifying the amount of information that is shared between two variables (see figure 29.1). In the case of EEG analyses, the two variables can be signals from two different electrodes, two signals from the same electrode (e.g., power and phase, or activity in two different frequencies), or one signal from an electrode and a behavioral or experiment variable. Thus, mutual information is a flexible analysis framework that can be custom-tailored to identify patterns of connectivity regardless of the distributions of the data (e.g., linear, non-linear, circular).

29.1 Entropy

Entropy is the basic building block of mutual information and therefore is introduced first. Here, the term entropy is used to refer to Shannon entropy from information theory. It is the amount of information, or “surprise,” a variable has and should not be confused with the thermodynamic concept of entropy or the annoying fact that your apartment constantly gets more, not less, messy.

To compute entropy with continuous data such as an EEG, you first bin the data as you would to create a histogram. Binning continuous data can be done using the Matlab function `hist`. Next, compute the probability that a value of the data would fall into each bin. This is simply the bin count divided by the sum of all bin counts. Then, multiply the probability value by the logarithm-base-2 of that probability value. Finally, sum all the probability-log-probability values and multiply by -1.

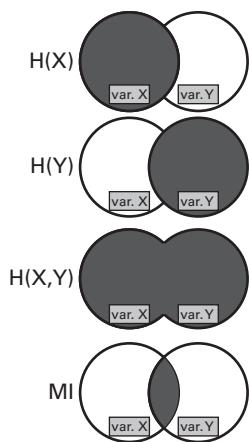


Figure 29.1

Venn diagrams illustrating entropy of variables X and Y (indicated by the letter H), joint entropy [$H(X,Y)$], and mutual information (MI).

$$H(X) = -\sum_{i=1}^n p(x_i) \log_2 p(x_i) \quad (29.1)$$

where H is the measure of entropy, p is the probability of observing the i^{th} value of the bin series data x , and n is the number of bins. Note that when applying equation 29.1 to continuous data, you no longer use the data themselves; rather, you use the counts of data values that fall into specific bins. For example, if your data time series is 10,000 points long and you use 50 bins, n is 50, not 10,000. This is illustrated in the online Matlab code.

The log of a number less than 1 is negative, and because probabilities are always less than one, the summation will always be negative. Multiplying by -1 therefore flips the sign of the result. This can also be achieved by subtracting instead of summing the probabilities. You can see that entropy is related only to probabilities and distributions, not to the original scale of the data. This means that entropy and mutual information are not subject to power-law scaling effects over frequency and thus do not require baseline normalizations (although, as with many connectivity analyses, baseline subtraction is still useful to remove tonic effects).

Entropy is also unrelated to the temporal structure of the data. That is, you could randomly shuffle around the time points, and the entropy would not change. This makes entropy a very different quantity than the Fourier transform, which could change drastically if the temporal structure of the data is altered.

If you use the logarithm-base-2 instead of logarithm-base-10 or the natural logarithm, then the unit of entropy is a bit. One bit is defined as the uncertainty of a binary variable that has equal probability of being zero or one (a fair coin toss, for example, has one bit of information). It is not necessary to use logarithm-base-2; you can use any logarithm base you want. The unit of equation 29.1 is bit for logarithm-base-2, nat for natural logarithm, and ban for logarithm-base-10. If you apply permutation testing for statistics (discussed in section 29.12), the units become normal-Z scores, and thus the original logarithm base does not matter.

It is possible that there are zero probability values for some bins. This is problematic because the logarithm of zero is undefined (-Inf in Matlab). Thus, in Matlab code, you should add a very small number inside the logarithm term to prevent taking the logarithm of zero. In Matlab you can use the built-in variable `eps` (the smallest numerical resolution). Thus, instead of writing `log2(p)`, you should write `log2(p+eps)`. This adds such a tiny number as to be insignificant to non-zero probabilities, and will prevent taking the logarithm of zero. It will generate very large negative values when the probability is zero (e.g., -52), but you also needn't worry about this because this value is then multiplied by zero [because of `p*log2(p+eps)`]. Another option would be to eliminate all bins with zero probabilities, but this may lead to complications in later parts of your code because the number of bins could change across variables.

Figure 29.2 shows a random signal and a sine wave and their distributions. You may initially be surprised to learn that the sine wave and the random signal have a similar amount of entropy (respectively, 5.3717 and 5.6394 bits). As mentioned earlier, the temporal sequence is not taken into consideration when computing entropy. Indeed, by inspecting the distributions of values in the sine wave and in the random signal (figure 29.2C), you can see that the two distributions are similar except that the sine wave has a larger clustering of data points around -1 and +1. By sorting the vectors (figure 29.2D), it is apparent that in a numerical-distributional sense (when ignoring the temporal structure), the two vectors contain a similar amount of information.

29.2 How Many Histogram Bins to Use

The second input to the Matlab function `hist` is the number of bins into which to discretize the data. How many bins should you use? You should not have too few bins; otherwise there will be little sensitivity to the distribution of the data (imagine an extreme situation of two bins, in which all values are classified as “low” or “high”). On the other hand, if you have too many bins, you lose generality and end up with only one value per bin and a flat

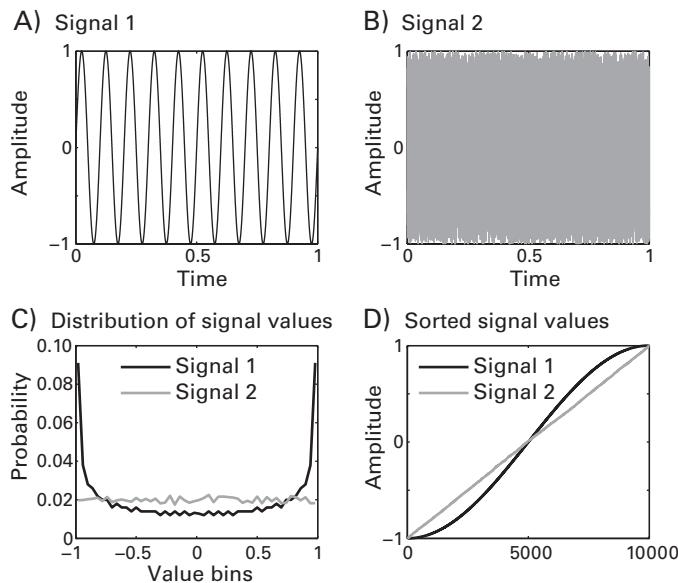


Figure 29.2

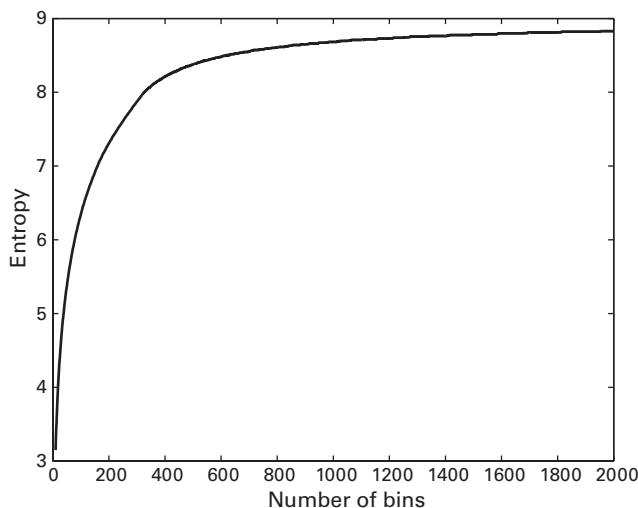
Examples of two different signals with similar amounts of entropy. Although a sine wave (panel A) and random noise (panel B) might seem like very different signals, it is their temporal structure, not their distributions of data values, that makes them so different. These two signals have similar entropy values (sine wave and noise, respectively, 5.3717 and 5.6394 bits), which shows that entropy is based on the “timeless” distribution and not on the temporal structure.

distribution (imagine the other extreme of having the number of bins equal to the number of data points). Unfortunately, the number of bins you use has implications for the estimate of entropy. This is shown in figure 29.3 for the sine wave used earlier. Entropy is plotted as a function of the number of bins, from 10 to 2000 (the signal is 10,001 points long). You can see that entropy increases with the number of bins, even up to 2000 bins.

Fortunately, there are statistical guidelines for determining an appropriate number of bins to use in a histogram. The guideline preferred here is called the Freedman-Diaconis rule (Freedman and Diaconis 1981), which states that the optimal number of histogram bins is related to the interquartile range and to the number of data points.

$$nbins = \left\lceil \frac{\max(x) - \min(x)}{2Q_s n^{-1/3}} \right\rceil \quad (29.2)$$

in which Q is the interquartile range of data distribution X (that is, the numerical distance between the 25th and the 75th percentiles of the data distribution), n is the total

**Figure 29.3**

Entropy (y-axis) of a sine wave as a function of the number of bins used to create the probability distribution (x-axis).

number of data points, and $\max(x)$ and $\min(x)$ indicate the maximum and minimum values in data X . The lopsided vertical bars indicate that the ceiling of this function should be taken.

There are other guidelines for selecting the number of bins. For example, there is Scott's rule (Scott 2010), which is similar to the Freedman-Diaconis rule except the $2Q$ term is replaced with $3.5s$, where s is the standard deviation. Scott's rule assumes that the data are normally distributed, whereas the Freedman-Diaconis rule does not. When the data are normally distributed, Scott's rule and the Freedman-Diaconis rule will provide the same or a similar number of bins. The absence of assumptions regarding data distributions means that the Freedman-Diaconis rule can be applied to any data distribution. This is useful for EEG time-frequency data, which can have a power distribution or a circular distribution. Another option is Sturges's rule (Sturges 1926), which states that the number of bins should be $1 + \log_2 n$. Sturges's rule can underestimate the appropriate number of bins when there are more than a few hundred data points, in part because it considers only the number of data points, not the range of variance of the distribution.

For the sine wave data, the optimal numbers of bins under the Freedman-Diaconis, Scott, and Sturges rules are, respectively, 16, 18, and 15. These may seem odd choices when considering only the plot of entropy as a function of number of bins (see figures 29.3 and 29.4A),

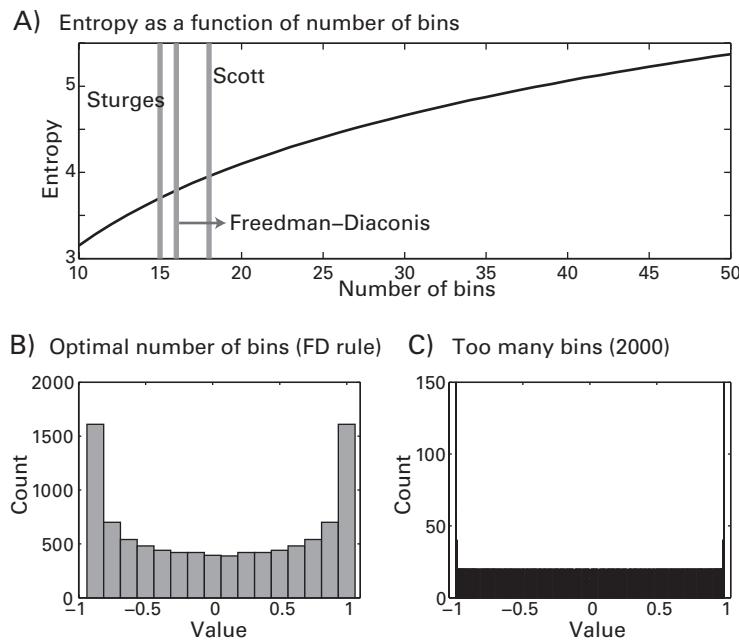


Figure 29.4

Panel A shows a zoom-in of the plot shown in figure 29.3 and the suggested optimal number of bins from different statistical guidelines (vertical lines). Panels B and C show histograms of the sine wave data using the number of bins recommended by the Freedman-Diaconis rule (FD; panel B), and 2000 bins (panel C). Clearly, a histogram with 2000 bins is a poor representation of the shape of the distribution.

but they are more sensible choices when considering histograms of the data with 16 versus 2000 bins (figure 29.4B,C).

The procedures for selecting an appropriate number of bins outlined above are for one variable. When you are computing mutual information (or computing entropy for multiple variables), choosing an appropriate number of bins becomes more difficult because the optimal number of bins might be different for each variable. Because the number of bins will influence the estimate of entropy, the same number of bins should be used for computing entropy in all conditions, electrodes, and subjects. This is demonstrated in section 29.9. Thus, a recommended strategy is to compute the optimal number of bins for each variable, take the ceiling (i.e., round up) of the average of the optimal number of bins, and then apply this to all variables. This procedure is used in the online Matlab code.

29.3 Enjoy the Entropy

Without going any further toward mutual information, entropy can itself be an informative measure of brain dynamics and complexity. Entropy is a measure of uncertainty. Higher entropy indicates that the system under investigation can take more states or configurations. Thus, decreased entropy indicates that the activity recorded by that electrode has a more restricted number of configurations in that time segment.

Entropy can be computed over time or over trials (see figure 26.4 for an illustration of analyses over time vs. over trials). However, unlike some other measures that are computed over time *or* over trials, entropy (and, therefore, mutual information) can also be computed over time *and* over trials simultaneously. This is because entropy is not based on the temporal structure of the data but, rather, on the probabilities that data points will take certain values. For example, if you use a time segment of 100 data points, and you have 100 trials, computing entropy over time and trials involves generating a distribution from 10,000 data points. This is a major advantage because it means that short time segments can be used for computing entropy (and mutual information), thus preserving temporal precision while retaining a sufficient amount of data for high-signal-to-noise estimates of entropy.

Figure 29.5 shows two examples of entropy computed in time segments. Figure 29.5A shows a topographical distribution of task-related entropy from 100 to 400 ms (entropy during a pretrial baseline period from -400 to -100 ms was subtracted). Figure 29.5B,C shows the time course of entropy at electrodes FCz and PO8 separately for the first 30 trials and the last 30 trials. Entropy was computed in overlapping time segments of 400 ms using 50-ms steps.

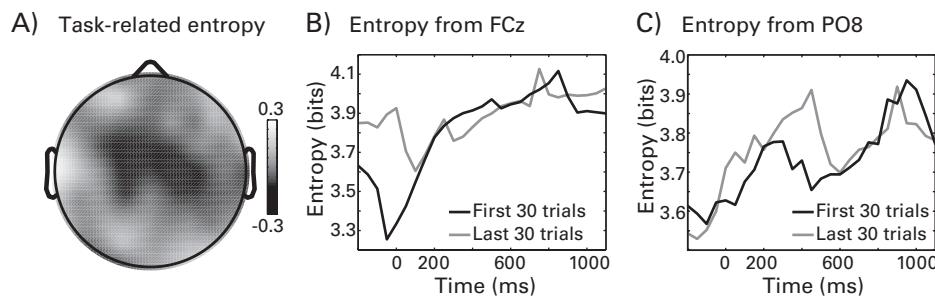


Figure 29.5

Topographical map of task-related entropy (panel A) and a time-course of entropy from electrodes FCz (panel B) and PO8 (panel C).

29.4 Joint Entropy

Joint entropy is the total entropy of a pair of variables. The formula for joint entropy follows.

$$H(X, Y) = -\sum_{j=1}^m \sum_{i=1}^n p(x_i, y_j) \log_2 p(x_i, y_j) \quad (29.3)$$

You can compare this equation to equation 29.1 and see that equation 29.3 has the same form but adds a new variable Y ; j and m are the corresponding counting indices.

In practice, joint entropy is computed slightly differently from how entropy is computed. With entropy you need to know the distributions of individual variables, but with joint entropy you need to know the codistributions of the two variables. This can be done in Matlab by creating two nested loops, one loop over bins of variable X and one loop over bins of variable Y . Inside the double loop, count the number of times that the two variables are in the same distribution bins. After this double loop, you have a joint probability matrix, with each element in the matrix reflecting the probability that a particular set of values occurred within one bin. Figure 29.6 shows some examples of joint probability matrices for different pairs of time series.

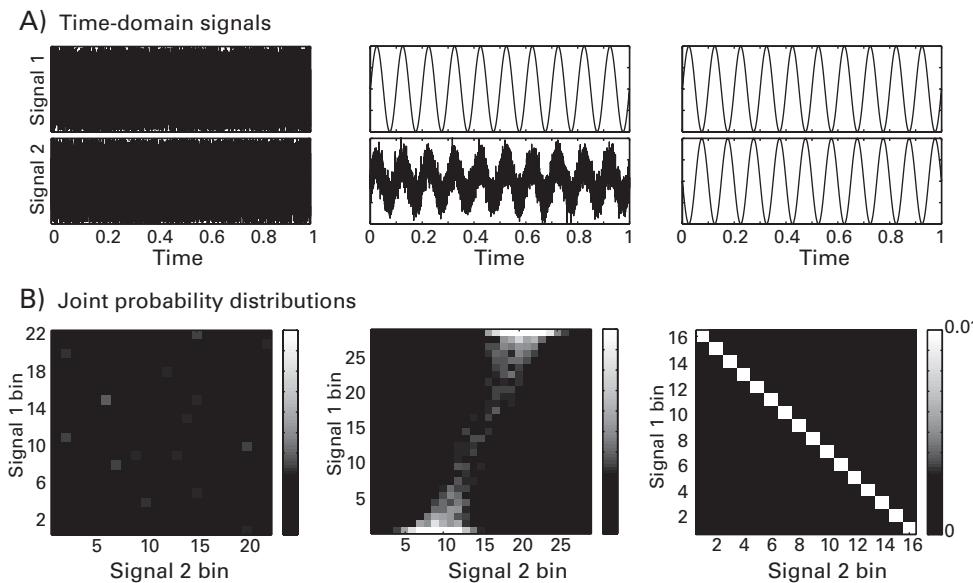
The probability matrices in Figure 19.6 are the $p(x,y)$ terms in equation 29.3. Thus, to compute joint entropy, you Shannonize this matrix by multiplying the probability matrix by the \log_2 of that matrix. As discussed above with univariate entropy, do not forget to add `eps` inside the \log_2 in case there are bins with no values. Note that joint entropy is not limited to the bivariate case; you can compute joint entropy over more variables than two [e.g., $p(X, Y, Z)$], although this is not further discussed here.

29.5 Mutual Information

Mutual information is the amount of shared information between two variables (figure 29.1). It can be computed by summing the individual entropies of the two variables and then subtracting their joint entropy (this expression should make intuitive sense from considering figure 29.1). There are alternative and equivalent ways of expressing the computation of mutual information, but equation 29.4 is presented here because it follows from equations 29.1 and 29.3.

$$MI(X, Y) = H(X) + H(Y) - H(X, Y) \quad (29.4)$$

In figure 29.7 you can see several bivariate functions and their mutual information. Spearman correlations are also shown for comparison. Perhaps the most striking advantage of

**Figure 29.6**

Example pairs of time series (panel A) and their joint probability matrices (panel B; grayscale intensity refers to probability and has the same scale for all panels).

mutual information in these examples is the circle—clearly, there is a relationship between X and Y (in fact, they are simply sine and inverted sine of same numbers from 0 to π), but their correlation coefficient is basically zero. Another striking feature about mutual information is that you cannot determine from the mutual information value itself whether a relationship is positive or negative.

The blindness of mutual information to linear versus nonlinear and positive versus negative relationships can be either an advantage or a disadvantage. It is an advantage if you want to test for a relationship between two brain signals and you do not care what the shape or sign of that relationship is (e.g., linear, monotonic, logarithmic, circular, negative, positive), and it is a disadvantage if you are interested in specifically discovering linear or nonlinear relationships or limiting yourself to positive or negative relationships. Mutual information is also advantageous when computing connectivity between signals that have different distributions, such as exponential and circular (for power and phase). Probably in most situations, you might not care about linear versus nonlinear relationships, but you might care about identifying positive versus negative relationships. Either way, it is important to keep this feature of mutual information in mind when interpreting the results.

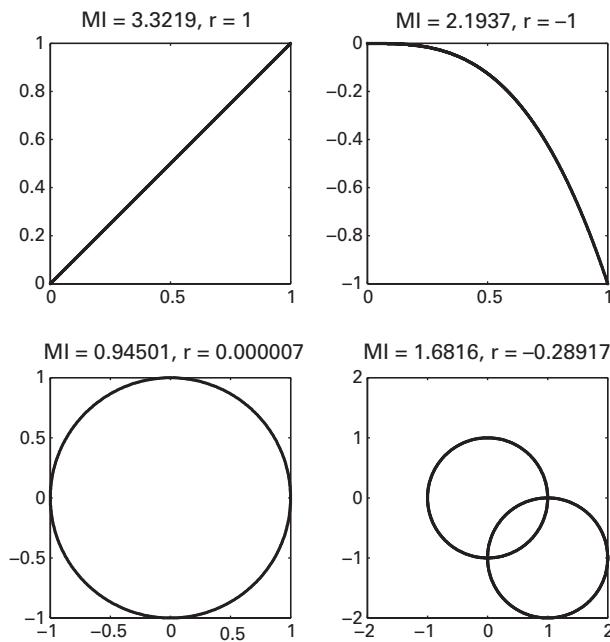


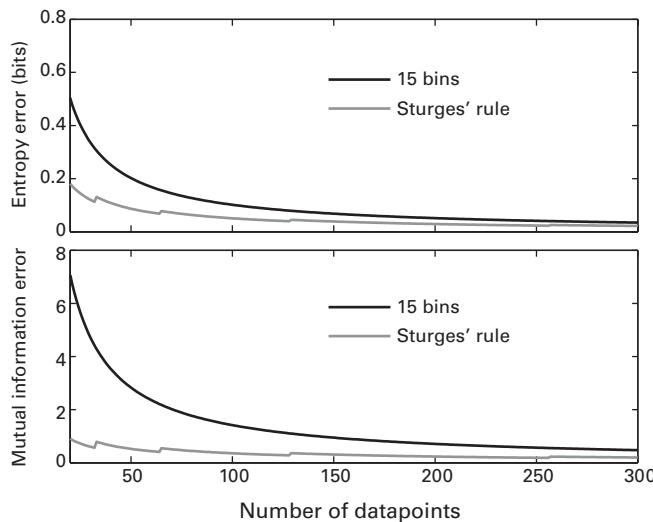
Figure 29.7

Comparison of mutual information (MI) and Spearman rho (r) for different bivariate distributions. Note that although mutual information is clearly superior to correlation for “unusual” distributions such as a circle, mutual information does not tell you the shape of the relationship between the two variables (linear, nonlinear, circular) or the sign of the relationship (negative or positive).

29.6 Mutual Information and Amount of Data

Having too little data can inflate estimates of entropy and mutual information (note that how much data you need is a separate issue from how many bins you use to discretize the data). This is because entropy cannot be negative, and thus having too few data points will increase, not decrease, estimates of entropy and mutual information. For resting state data this is not a concern because you likely have hundreds of thousands or perhaps millions of data points. But for task-related data, and in particular, when computing time-varying mutual information, this may be a concern. The inflation of entropy and mutual information due to sample size can be estimated (Herzel and Grosse 1995; Steuer et al. 2002) according to equations 29.5 and 29.6.

$$\Delta H = \frac{\text{bins} - 1}{2N \ln 2} \quad (29.5)$$

**Figure 29.8**

Estimate of the inflation in entropy (top panel) and mutual information (bottom panel) as a function of the number of data points, plotted separately for a fixed bin size and variable bin size.

$$\Delta MI = \frac{(bins - 1)^2}{2N \ln 2} \quad (29.6)$$

Here, *bins* is the number of bins in the histogram, *N* is the number of sample points, and $\ln 2$ is the natural logarithm of 2, which scales the result to units of bits. If you use a different number of bins for different variables (although this is not recommended for reasons illustrated in figure 29.4), the numerator in equation 29.5 becomes $(bins_{xy} - bins_x - bins_y + 1)$, where $bins_{xy}$ refers to the joint probability distribution.

Figure 29.8 shows the expected inflation of entropy and mutual information as a function of the number of sample points. Fortunately, having a variable bin size already helps reduce this inflation, as is shown in figure 29.8 for Sturges's rule (the other bin-size-optimizing rules are not shown because they require data to compute the variance, but they would be expected to show similar functions).

What should you do about the impact of a small number of data points? There are three options. The first is to compute and subtract equation 29.5 or 29.6 from the results (if computing mutual information, apply only equation 29.6, not equation 29.5 first to the entropies and then equation 29.6 to the mutual information). There are other corrections as well, for example, based on variance of the signal (Herzel and Grosse 1995) or based

on resampling techniques (Cogan and Poeppel 2011). The second option is to ignore the sample size bias: as long as this bias is similar across conditions, electrode pairs, time segments, and so on, the bias will affect all conditions equally and thus will not affect interpretations of condition differences or statistical comparisons. The third option is to evaluate entropy and mutual information with respect to a null hypothesis distribution obtained via permutation testing. Permutation testing for mutual information has additional advantages beyond correcting for limited sample size and is discussed in section 29.12.

If you have many trials, a small number of time points is less of a concern if you compute mutual information by concatenating all trials from that time bin into one distribution, as is done in figure 29.5.

29.7 Mutual Information with Noisy Data

Because the number of sample points helps stabilize estimations of entropy and mutual information, you might think it would be a good idea to have as high a sampling rate as possible to increase the number of data points. Although it is generally a good idea to have a fairly high sampling rate for most EEG data analyses, simply adding more samples will do little to improve mutual information in the presence of noise.

Figure 29.9 shows that when noise is added to one of two variables (in this case, the y -axis variable from the lower right panel of figure 29.7), estimates of mutual information decrease. This is not very surprising. What is perhaps initially more surprising is that increasing the number of sample points has little appreciable restorative effect. That is, if there is noise in the data, having more data will not necessarily offset the adverse effects of noise. In part, this is because mutual information is based on co-occurrences among two signals, and thus, adding the same amount of noise to an increasing amount of data does not eradicate spurious co-occurrences in data distributions. This is conceptually different from, for example, estimating a mean parameter, for which more data reduce the effects of noise because the noise averages out.

The take-home message from figure 29.9 is that if you plan to perform mutual information analyses, use clean data. Having more data and a high sampling rate is generally a good idea for EEG research but will not necessarily compensate for noise when you are computing mutual information. Furthermore, make sure the data are roughly equally noisy across conditions, electrode pairs, subject groups, or whatever your key comparison is. Condition differences in noise might spuriously cause condition differences in mutual information, even if there are no true differences in brain connectivity. If you are concerned about this

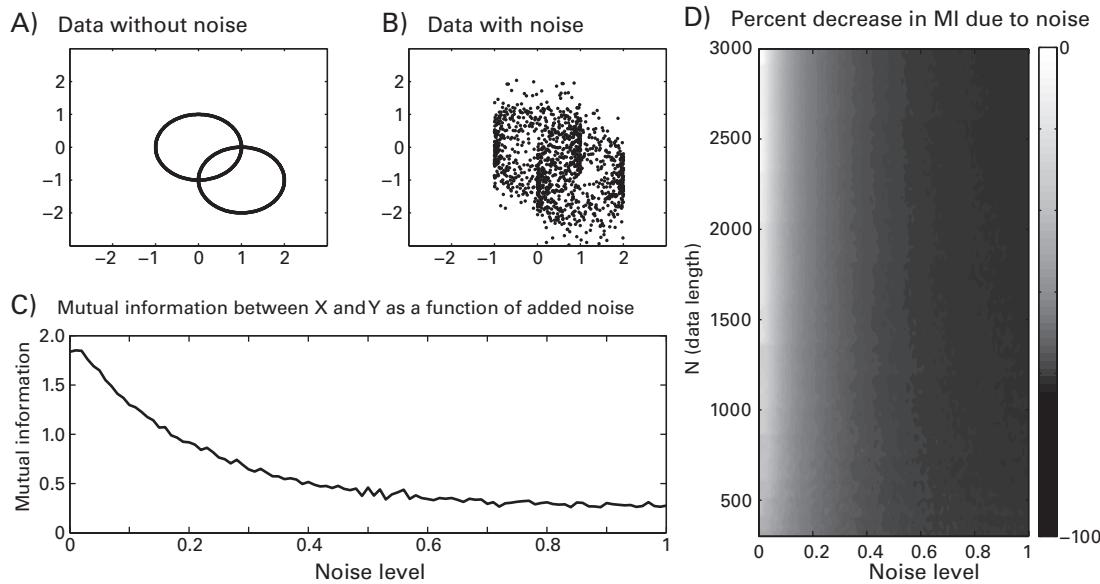


Figure 29.9

Effects of noise on estimates of mutual information. The double-circle function plotted in figure 29.7 was used, with an increasing amount of noise added on the y -dimension. An example of the function with and without noise is shown in panels A and B. Panel C shows the decrease in mutual information as a function of increasing noise levels. Panel D shows the relationship between the level of noise and the number of data points, defined here by increasing the sampling rate. Gray intensity indicates the percentage decrease in mutual information relative to the “best-case scenario,” which is 3000 data points and no noise added. You can see that noise decreases the mutual information and that this is unaffected by the number of data points.

potential alternative explanation to condition differences in mutual information, you could examine signal-to-noise ratios (see section 18.12) or any other measure of signal variability, and interpret differences in mutual information when the signal-to-noise ratios are similar across conditions, subjects, groups, or whatever is the independent variable.

The deleterious influence of noise on estimated mutual information is another argument for frequency-band-specific mutual information because the signal-to-noise ratio of frequency-band-specific power tends to be higher than the signal-to-noise ratio of broadband EEG (section 18.12). If you will only compute mutual information of time-domain data, it is a good idea to low-pass filter the EEG data, for example at 40 Hz, to remove high-frequency fluctuations and noise.

29.8 Mutual Information over Time or over Trials

As with other connectivity measures mutual information can be computed over time or over trials at each time point (figure 26.4). And as with other measures of connectivity, although the math and programming of the analysis are nearly identical for over time versus over trials (the difference is in how the distributions are built), the interpretation is different: mutual information over trials has better temporal precision and is well suited for detecting phase-locked changes in connectivity, whereas mutual information over time has poorer temporal precision but is more sensitive to identifying non-phase-locked connectivity.

As was explained in section 29.3, mutual information can also be performed over time and over trials simultaneously. This will help increase the robustness of the mutual information estimates because there is likely to be a sufficient amount of data in each probability bin.

29.9 Mutual Information on Real Data

Figure 29.10 shows mutual information over time and trials between electrodes Fz and O1. This was computed by taking the data from these two electrodes in sliding 400-ms segments (step size 100 ms) over trials, thus using 10,197 data points per electrode per time window. By inspecting figure 29.10A it appears that mutual information peaks prior to trial onset and then declines steadily. However, in this analysis, the number of bins was set separately for each time segment. Figure 29.10B shows the danger of directly comparing results using different bin sizes: the mutual information is strongly related to the number of bins (as shown in figure 29.3 for entropy; in this example, the correlation coefficient is 0.916). Keeping the number of bins constant over time reveals a different and more accurate picture: mutual information between Fz and O1 decreases around stimulus onset relative to the pretrial baseline and then increases, peaking at around 500 ms. Keep in mind when interpreting the time course of the mutual information that with a time segment of 400 ms, the temporal precision is decreased, and thus, early posttrial changes in mutual information may be measured by time segments centered in the pretrial time period.

29.10 Mutual Information on Frequency-Band-Specific Data

Mutual information on frequency-band-specific data can be computed based on phase or on power. Figure 29.11A,B (plate 22) shows an example of mutual information over trials applied to power and phase data between electrodes Fz and O1. In this example, mutual

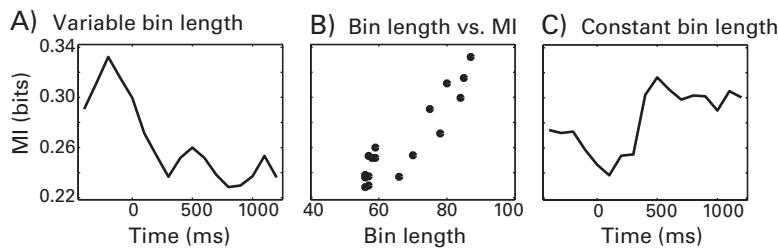


Figure 29.10

Mutual information between Fz and O1 over time. Allowing the number of histogram bins to change at each time segment suggests that mutual information peaks prior to trial onset (panel A). However, panel B shows a strong correlation between mutual information and histogram bin size. The results of keeping the histogram bin size constant over time suggest that mutual information first decreases immediately after stimulus onset and then increases, peaking at around 500 ms. The y-axis limits are the same for all panels.

information based on power and phase show different dynamics, which is not very surprising, considering that power and phase are independent measures. Mutual information of power shows sustained connectivity in the theta band, whereas mutual information of phase shows more transient connectivity in the theta/delta ranges. For comparison, figure 29.11C,D (plate 22) show connectivity based on cross-trial power correlations over trials (chapter 28) and ISPC-trials (chapter 27). The ISPC-trials and mutual information of phase information show similar patterns of results (note that both are nonlinear connectivity measures).

Comparing the power correlations and mutual information of power highlights an interesting feature of mutual information. You might initially look at these two plots and conclude that the mutual information reveals different dynamics than the power correlations. However, closer inspection reveals that they show similar dynamics. Because mutual information is insensitive to the sign of a monotonic relationship, mutual information will increase regardless of the direction of the effect (this was shown in figure 29.7). For example, imagine plotting in panel C the absolute value of the correlation coefficients rather than the signed value of the correlations; then the power correlations and mutual information results would look more similar.

In figure 29.11 (plate 22) mutual information was computed between power values or between phase values at the same frequency. It is also possible to compute mutual information between power and phase and between signals at different frequencies. This is easily seen in the grayscale version of figure 29.11. This would be one method of computing cross-frequency coupling.

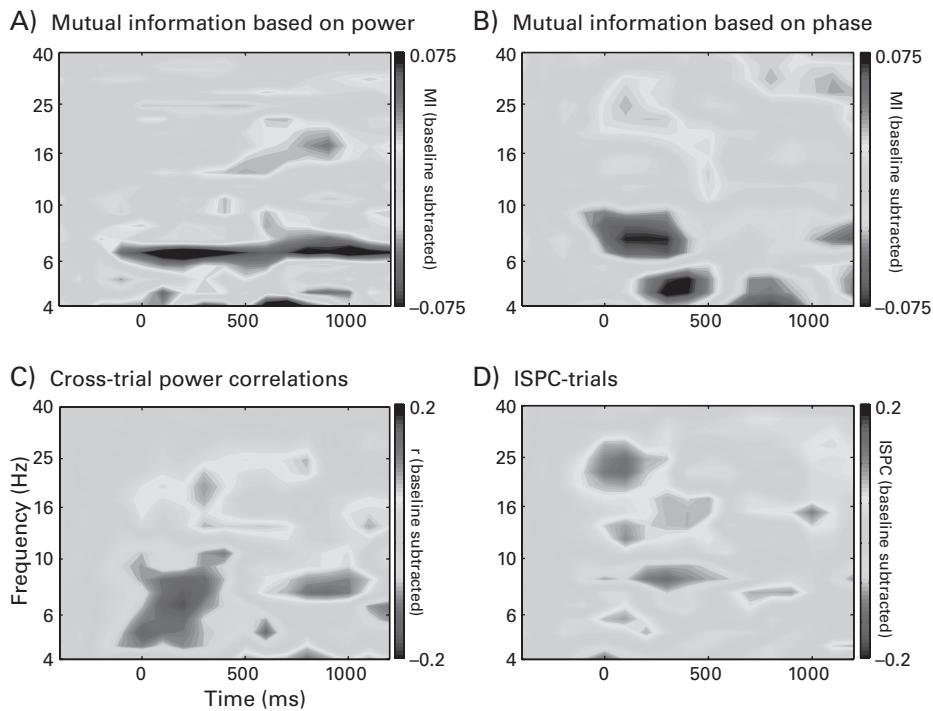


Figure 29.11 (plate 22)

Mutual information over trials between Fz and O1 based on frequency-band-specific power (panel A) and phase (panel B). For comparison, cross-trial power correlations are shown in panel C (r indicates the Spearman correlation coefficient), and ISPC-trials is shown in panel D. In all panels data were baseline-subtracted to focus on task-related effects.

29.11 Lagged Mutual Information

Mutual information as described thus far is bidirectional, meaning it is not possible to determine whether the connectivity goes $A \rightarrow B$ or $B \rightarrow A$ (or both). However, mutual information can be adapted to infer information about directionality. One adaptation is lagged mutual information, which involves time-shifting one signal with respect to another and computing mutual information repeatedly for multiple time lags (Fraser and Swinney 1986). This method has been applied to time-frequency-phase data (Wilmer, de Lussanet, and Lappe 2012). Figure 29.12A shows lagged mutual information on simulated data. The data here were a 10-Hz sine wave and an inverted sine wave, which is why there are sharp rhythmic peaks every half-phase cycle.

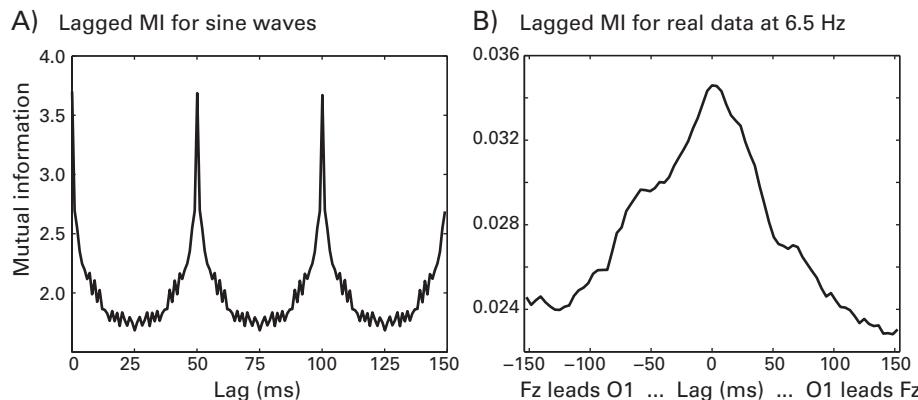


Figure 29.12

Lagged mutual information between a sine and an inverted sine wave at 10 Hz (panel A) and between power at 6 Hz from electrodes Fz and O1.

Figure 29.12B shows lagged mutual information for real data. As with the lagged power correlations in chapter 28, when frequency-band-specific data are used for mutual information, the maximum number of lags should be one cycle of the frequency.

The main limitation of lagged mutual information is that specific directional information is not separated from common inputs and common history of the two variables. Transfer entropy is an extension of mutual information (Schreiber 2000) that takes into account temporally recent states when estimating the mutual information between two variables.

29.12 Statistics

Parametric statistics are not appropriate for mutual information because the values are non-normally distributed. Although there are analytic solutions to derive a statistical threshold for mutual information given a statistical significance p -value, the number of time points, and the number of histogram bins (Dawy et al. 2006), a useful alternative is to perform permutation testing to transform the mutual information result from bits to a standard statistical Z -value. Permutation testing of mutual information controls for sample size bias and provides a useful framework for addressing multiple-comparisons corrections. Permutation testing is detailed in more detail chapter 32 and is discussed briefly here.

For mutual information, the null hypothesis is that there is no relationship between the distributions of the two variables. This null hypothesis implies that temporally shifting one time series with respect to the other has no bearing on the measure of mutual information.

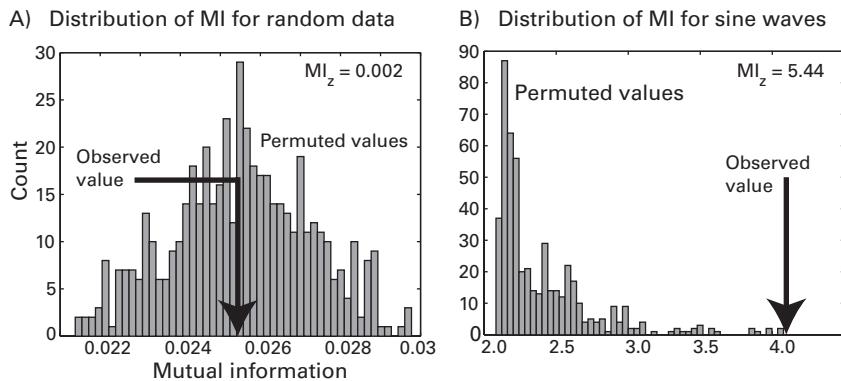


Figure 29.13

Examples of distributions of mutual information generated under the null hypothesis of no relationship between the two variables. The null hypothesis was created by randomly shuffling the timing of one time series relative to the other. The black arrows indicate the mutual information value of the original (unshuffled) time series. Panel A shows results for the mutual information between a sine wave and random data, and panel B shows results for mutual information between two sine waves with differing phases.

Thus, at each iteration of permutation testing, one time series is temporally shifted by some random amount without changing the other time series. Mutual information is then computed. This is repeated hundreds or thousands of times, thus generating a distribution of mutual information values expected under the null hypothesis. Note that the entropies of the individual variables do not change, only their joint entropy. After permutation testing, the resulting standardized measure of mutual information, MI_z , can be statistically evaluated on its own or used in parametric statistical tests at the group level. Figure 29.13 illustrates two distributions of mutual information values computed under the null hypothesis for a sine wave and random data and for two sine waves with different phases. Not surprisingly, the MI_z of the sine wave with random data (panel A) is close to zero, whereas the MI_z of two sine waves is very large, corresponding to a p -value of less than 0.0001 (panel B).

29.13 More Information

Entropy and mutual information are major topics in engineering, physics, and information technology. Presented in this chapter are the basics of what you need to know to apply entropy and mutual information to the study of EEG connectivity dynamics, but there are more advanced uses and developments beyond what is presented here, including computationally

efficient ways to compute mutual information. If you are interested in reading more about this topic, the following resources might be useful (Moddemeijer 1989; Paninski 2003; Shannon and Weaver 1949).

29.14 Describing This Analysis in Your Methods Section

Mutual information does not involve a large number of complex equations, so it is a good idea to include the formulas in your paper. More important than the formulas, however, are the parameters and data features. Be clear about how many bins were used and justify your choice of that number. State the number of trials and data points used to compute entropy and whether the number of data points differed across conditions, electrode pairs, and subjects. If there is reason to be concerned about noise and condition differences possibly being driven by differences in noise, state whether signal-to-noise ratio or some other criteria were examined to rule out this alternative explanation.

29.15 Exercises

1. Pick two electrodes and two frequencies (one frequency per electrode) and compute mutual information over time and trials between power from the first electrode and phase from the second electrode. Justify your choice of bin size. Next, recompute mutual information using phase from the first electrode and power from the second electrode. Make sure you use the same bin size you used in the previous analysis, so the results are directly comparable. Plot the time courses of the mutual information from these two analyses. Do you notice any striking features in the results, and are there differences between using phase and power from the two different electrodes?
2. Pick one electrode and two frequency bands. Compute mutual information between phase in that electrode and power in all other electrodes, separately in those two frequency bands. Run this analysis for a prestimulus period (e.g., -500 to -100 ms) and a task-related period (e.g., +100 to +500 ms). Show topographical maps for each frequency band, the two time periods, and additionally for mutual information during the task period minus the baseline period. Comment on any topographically salient features and any differences between baseline and task periods.

30 Cross-Frequency Coupling

Cross-frequency coupling refers to a statistical relationship between activities in two different frequency bands, has been observed in many species and in many brain regions, and has been linked to several cognitive processes and disease states (Canolty and Knight 2010). Cross-frequency coupling analyses require both the high temporal resolution and the high temporal precision of electrophysiological measurements. Therefore, you should perform cross-frequency coupling analyses on high-sampling-rate data. You should also carefully select your time-frequency decomposition analysis parameters: try to select parameters that increase temporal precision even though this entails some reduction of frequency precision. Although this will have implications for interpreting the results—for example, it may be difficult to determine whether theta phase is maximally coupled with 40-Hz or 55-Hz power—having a high temporal precision will provide greater sensitivity to detect true cross-frequency coupling.

There are several different manifestations of cross-frequency coupling and several ways to test for cross-frequency coupling; this chapter presents a few of the more commonly used approaches. The mathematical principles underlying several of the analysis techniques in this chapter are based on Euler's formula and on the use of Euler's formula for averaging phase values. Thus, before reading this chapter, make sure you are familiar with the material presented in chapters 13 and 19 (in particular, section 19.7).

30.1 Visual Inspection of Cross-Frequency Coupling

Even without performing any analyses you may see patterns in your data that suggest the presence of cross-frequency coupling. If you look at the time course of band-specific power, particularly above 20 Hz, you will probably see that power is not constant over time but rather appears in bursts that sometimes seem to occur at regular intervals. Figure 30.1 shows an example of 25-Hz power from the first trial at electrode O1. It seems from visual inspection

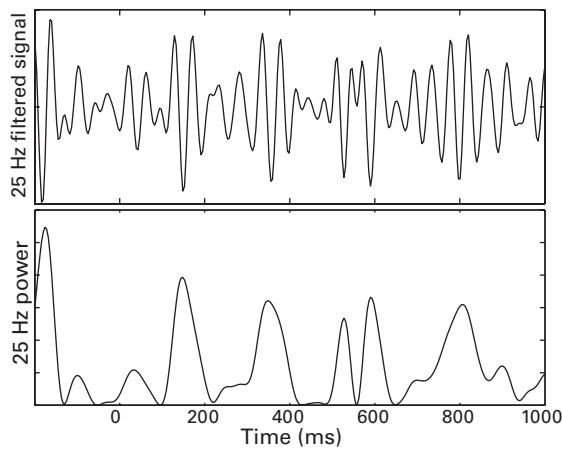


Figure 30.1

From visual inspection of one trial of 25-Hz power, it seems that there may be cross-frequency coupling between 25-Hz power and 5-Hz phase—note the approximate 200-ms rhythmicity in power bursts. This pattern of results was not robust over trials.

that 25-Hz power occurs in bursts at intervals of around 200 ms, suggesting cross-frequency coupling between 25-Hz power and 5-Hz phase. Of course this is an anecdotal observation of a single trial that requires quantitative statistical confirmation before being interpreted, but visual inspection often provides an informative and encouraging first step. In fact, this relationship is not robust over trials, which is demonstrated in the online Matlab code. This is therefore a useful illustration of how a small amount of “representative data” can be enticing and encouraging but should not be overly trusted.

30.2 Power-Power Correlations

There are two methods for computing power-power cross-frequency coupling; both were introduced in chapter 27. One method is to correlate two power time series over time, where the two time series are taken from different frequency bands (at the same or different electrodes). The second method for computing power-power cross-frequency coupling is cross-trial time-frequency power correlations, which was shown in figure 27.6C (plate 19).

30.3 A Priori Phase-Amplitude Coupling

Phase-amplitude coupling is perhaps the most commonly used method of computing cross-frequency coupling (Canolty et al. 2006) and has been confirmed in physiology studies (Bragin

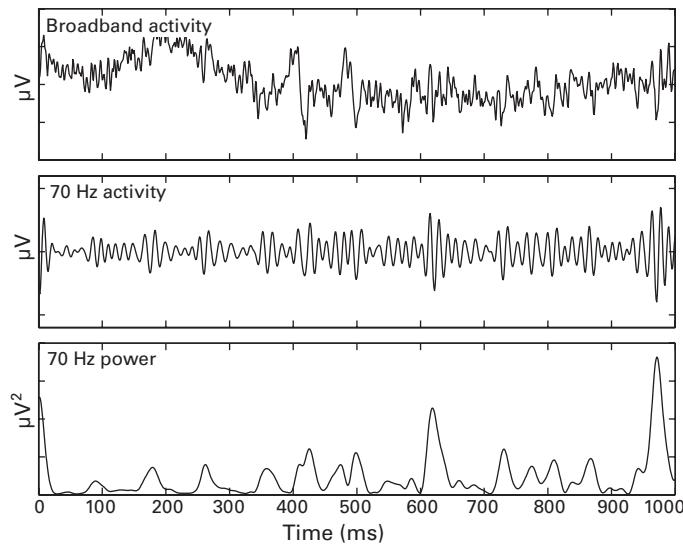
et al. 1995; Jensen and Colgin 2007; Lakatos et al. 2005) and computational and theoretical simulations (Lisman 2005; Zhang et al. 2012). Phase-amplitude coupling involves testing for a relationship between the phase of one frequency band and the power of another, typically relatively higher, frequency band. Thus, within phase-amplitude coupling, you refer to the “frequency for phase” and the “frequency for power.”

A distinction here is made among “*a priori* phase-amplitude coupling,” “mixed *a priori*/exploratory phase-amplitude coupling,” and “exploratory phase-amplitude coupling.” The differences are that with *a priori* phase-amplitude coupling you specify both frequency bands between which to assess cross-frequency coupling (e.g., based on *a priori* hypotheses); with mixed *a priori*/exploratory phase-amplitude coupling you specify one of the two frequency bands while using exploratory analyses for the other frequency band; and with exploratory phase-amplitude coupling you do not specify either frequency band but, rather, search through a frequency-frequency space and evaluate the strength of the phase-amplitude coupling at each frequency pair.

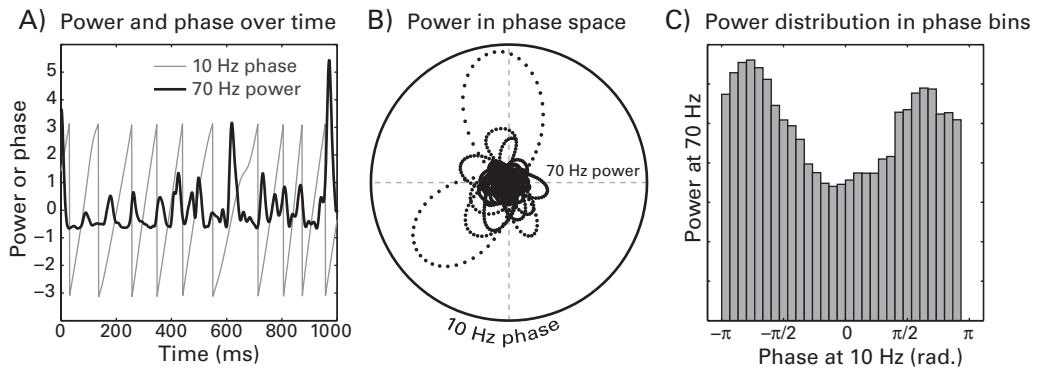
Some of the analyses in this chapter will use sample data recorded from the human nucleus accumbens (available with the online Matlab code) because they show strong alpha-gamma coupling (Cohen et al. 2009). One second of this 8-s snippet of data is shown in figure 30.2, along with the bandpass-filtered signal at 70 Hz (the real part of the result of convolution between the data and a 70-Hz wavelet) and power at 70 Hz. As with figure 30.1, visual inspection suggests that 70-Hz power contains bursts that occur around every 90 ms (you can count 12–13 peaks in this 1-s period).

Before learning about the mathematical quantification of phase-amplitude coupling, consider visual representations of the relationship between the 70-Hz power and 10-Hz phase. Figure 30.2 shows this relationship in three different ways: plotted simultaneously over time (figure 30.3A), 70-Hz power as a function of 10-Hz phase in polar space (figure 30.3B), and a histogram of 70-Hz power values over 10-Hz phase values (Figure 30.3C). The null hypothesis—that 70-Hz power is unrelated to 10-Hz phase—would produce uniformly distributed black dots in panel B (that is, a perfect circle) and a flat histogram in panel C. The lack of uniform distribution in panels B and C suggests that relative increases in 70-Hz power occur preferentially at certain regions of the 10-Hz phase distribution.

The quantification of phase-amplitude coupling is based on Euler’s formula. Recall that Euler’s formula (e^{ik} ; section 13.4) can be used to represent a phase angle as a unit-length vector in a circle, and it can also be used to weight phase values by some trial-varying variable to test for a relationship between that variable and phase (wITPC; section 19.7). Phase-amplitude coupling is similar to wITPC except that instead of using a trial-varying weighting such as reaction time, phase-amplitude coupling uses the time-varying power time series from a higher frequency band. In other words each vector in polar space (one vector per

**Figure 30.2**

Broadband, 70-Hz-filtered, and 70-Hz power time series data from 1 s of recordings from the human nucleus accumbens. The data were sampled at 1000 Hz.

**Figure 30.3**

Different ways of showing power and phase data from different frequency bands. Data from only the first second of 8 s are shown in panel A for visibility. The “loops” in panel B result from the power increasing and decreasing over time (each dot is a time point).

time-frequency point) is defined as the angle from the 10-Hz phase angle time series and the length from the 70-Hz power time series. This is easy to see in figure 30.3B by imagining that each dot is the endpoint of a vector to the origin. The length of the average vector is the measure of phase-amplitude coupling (PAC) (Canolty et al. 2006):

$$PAC = \left| n^{-1} \sum_{t=1}^n a_t e^{i\phi_t} \right| \quad (30.1)$$

in which t is time point, a is the power at 70 Hz at time point t , i is the imaginary operator, ϕ is the phase angle (in radians) at 10 Hz at time point t , and n is the total number of time points. You should use the raw, untransformed power time series because this will ensure that all vectors have positive length.

There are three confounds of assessing phase-amplitude coupling via equation 30.1, but fortunately, there is one solution that solves them all. The first confound is related to the scale of the result of this equation. Similar to wITPC, PAC values from equation 30.1 are not bound by 1.0 but instead can be arbitrarily large. In fact, simply multiplying the power values by 10 will increase PAC, even though the relationship between relative fluctuations in power and phase has not changed (figure 30.4B). This is not good: a measure of cross-frequency coupling should reflect the relationship between power and simultaneous phase, and it should not be arbitrarily influenced the magnitude of the power values. Although it

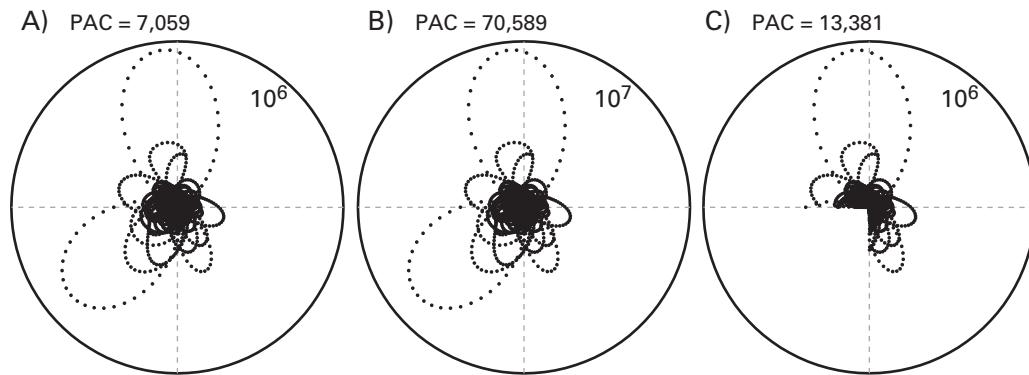


Figure 30.4

Three potential confounds of measuring phase-amplitude coupling via equation 30.1. First, the PAC value scales arbitrarily with power values (panel B; in this case, the power data were multiplied by the number 10). Second, nonuniform phase angle distributions will artificially increase phase-amplitude coupling strength (panel C). Third, outliers or large power fluctuations may increase PAC strength. Numbers near the outer circle indicate 70-Hz power (arbitrary units).

is possible to address this issue either by scaling the PAC measure by the power (Ozkurt and Schnitzler 2011) or by converting power values to rank and applying Moore's modification of the Rayleigh test (Zar 1999, p. 641), there are other potential concerns that favor an alternative option.

The second confound afflicting equation 30.1 is that a nonuniform distribution of phase angles will bias the result of equation 30.1. This can be seen in figure 30.4C. Removing phase angles selectively from one region of phase space nearly doubled the measure of phase-amplitude coupling strength defined by equation 30.1, although the relationship between power and phase at existing phase points has not changed. This may seem like an extreme and unrealistic illustration, but nonuniform phase angle distributions often occur after experiment events (see figure 19.2 for an even more extreme example using real data). Even if the distribution is full but not uniform, the violation of a von Mises distribution (the circular equivalent of a normal distribution) will bias the result of equation 30.1. In theory, you could statistically test each phase angle distribution to determine whether it is von Mises distributed, but this is not practical when testing for phase-amplitude coupling over time, frequencies, conditions, electrodes, and subjects.

The final confound associated with equation 30.1 is that very large power fluctuations that may be outliers can unfairly influence PAC. Figure 30.3 illustrates this concern. There are two bursts that can be seen in the power time series data. These bursts might reflect noise, or they might be driven by true (that is, of neural origin) brief large-amplitude bursts of power, but either way, they have a disproportionate effect on phase-amplitude coupling as defined by equation 30.1. Situations like this will often arise in power data, and it is generally not feasible to inspect each phase-amplitude coupling result to determine whether that result may have been unduly influenced by brief large-amplitude fluctuations that happened to occur at a specific region of phase space. Thus, a useful measure of cross-frequency coupling must be robust to large fluctuations in power that are outliers or otherwise nonrepresentative of the power time series.

Fortunately, there is a single solution to all three potential confounds listed here. Furthermore, this solution has the added benefit of making phase-amplitude coupling strength more amenable to statistical evaluation and condition comparisons, both at the single-subject level and across subjects at the group level. The solution is to apply nonparametric permutation testing (discussed in more detail in chapter 33) to determine how the PAC value compares to a distribution of phase-amplitude coupling values expected under the null hypothesis. The null hypothesis, that there is no temporal relationship between phase and power, implies that if the two time series (power and phase) are shuffled with respect to each other, this will not have an effect on the phase-amplitude coupling value shown

in equation 30.1. Permutation testing for phase-amplitude coupling therefore involves temporally shifting the power time series by a random temporal offset without changing the phase angle time series. The phase-amplitude coupling value is then computed according to equation 30.1, producing one phase-amplitude coupling value under the null hypothesis. This procedure is then repeated hundreds or thousands of times, generating a distribution of phase-amplitude coupling values expected under the null hypothesis. The observed PAC (that is, the result of equation 30.1 without altering the data) is compared to the distribution of PAC values under the null hypothesis by subtracting the mean and dividing by the standard deviation (figure 30.5). This creates a standardized Z-value of PAC, or PAC_z .

There are several advantages to interpreting the PAC_z instead of the “raw” PAC value: normal-Z values are independent of the scale of the original data (this addresses the first

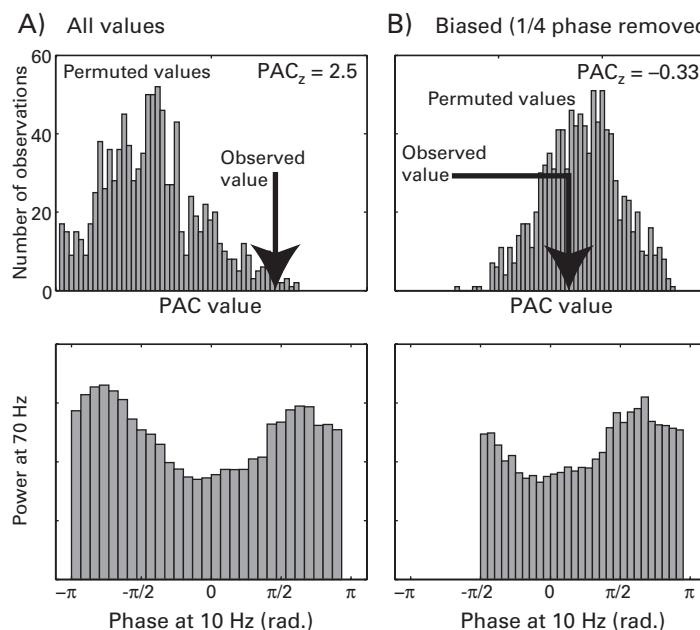


Figure 30.5

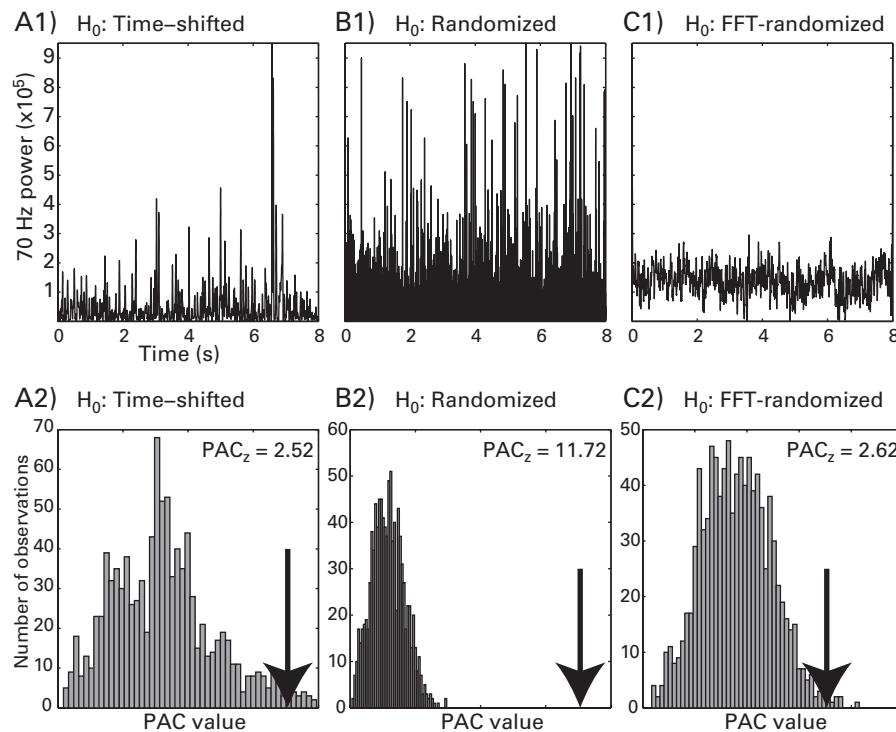
Illustration of PAC_z . Repeatedly assessing PAC with a randomly permuted power time series yields a distribution of PAC values expected under the null hypothesis (panels A and B, top row). The PAC value observed without shuffling the time series (see black arrow) is normalized with respect to this distribution. The normalized PAC, or PAC_z , is in standard deviation units and is unrelated to the scale of the power data. The bottom rows show the distribution of 70-Hz power values over 10-Hz phase bins from the observed data.

confound associated with equation 30.1), they are based on the observed phase distribution and thus are not influenced by violations of a von Mises distribution (this addresses the second confound), and the result is not influenced by large power fluctuations because those fluctuations will be paired with different phase values in different permutations (this addresses the third confound). PAC_z values have additional advantages: they have a straightforward statistical interpretation (standard deviation units), are easily evaluated in a statistical sense within the context of permutation testing (normal-Z values can be converted to p -values), and are potentially suitable for group-level parametric statistical tests (because they have an average value of zero and variance of 1 under the null hypothesis).

There are several ways to shuffle the power time series. One is to cut the power time series at a random point and put the postcut time series before the precut time series. A second way is to completely randomize the time points, which destroys the temporal characteristics of the original power time series. A third method is to use a frequency domain randomization procedure, which involves taking the FFT of the power time series, shuffling the phases of the FFT, and taking the inverse FFT. This method generally provides similar results as the first method and often removes nonrepresentative large-amplitude peaks, although it changes the values from the original power time series. The random shuffling method is the least preferred because it eliminates the temporal structure of the power time series, and thus may inflate the statistical significance in some cases. The online Matlab code shows you how to implement each of these methods, and their results are compared in figure 30.6. In all of these examples the power time series was shuffled while the phase angle time series was left intact, but you could also shuffle the phase angle time series while leaving the power time series intact. In theory, you could also shuffle both the power time series and the phase angle time series, although this will not provide any better test of the null hypothesis than shuffling only one of the time series.

So far, cross-frequency coupling has been computed in one time segment. Assessing changes in cross-frequency coupling over time can be done by computing PAC_z over successive time segments to create a time series of PAC_z values. An example result is shown in figure 30.7.

How much data do you need for reliable estimates of phase-amplitude coupling? You need at least one full cycle of the lower frequency in order to build a distribution of higher-frequency power values; otherwise you will create a situation like that presented in figure 30.4C. But using data from only one cycle will not provide a robust estimate of phase-amplitude coupling. This is in part because of the relatively lower signal-to-noise ratio of high-frequency activity, particularly in noninvasive recordings. On the other hand, if the cross-frequency coupling is transient, using too many cycles in the analysis may

**Figure 30.6**

Different methods for constructing the null hypothesis power time series when computing phase-amplitude coupling and the resulting estimates of PAC_z . The top row shows examples of one (out of 1000) permuted power time series, and the bottom row shows the distribution of PAC values under the null hypothesis (gray bars) and the observed PAC value (black arrow). The observed PAC value is identical in all situations, but the distributions of null-hypothesis PAC values are different for the different ways of constructing the null hypothesis.

decrease temporal precision to the point of losing sensitivity for detecting transient effects. Fortunately, having many trials will help increase the signal-to-noise ratio because you can concatenate data from a small number of cycles over many trials. For example, if you use 5 Hz as the frequency for phase, three cycles (corresponding to 600 ms) may provide too little data for a robust estimate of cross-frequency coupling, but if you concatenate three cycles at the same time segment (e.g., 200–800 ms poststimulus) over 100 trials, the cross-frequency coupling analysis will be performed using 300 cycles, which should be sufficient to detect true task-related phase-amplitude coupling in the presence of noise (Tort et al. 2010). Pooling the data from small time segments over many trials is also advantageous

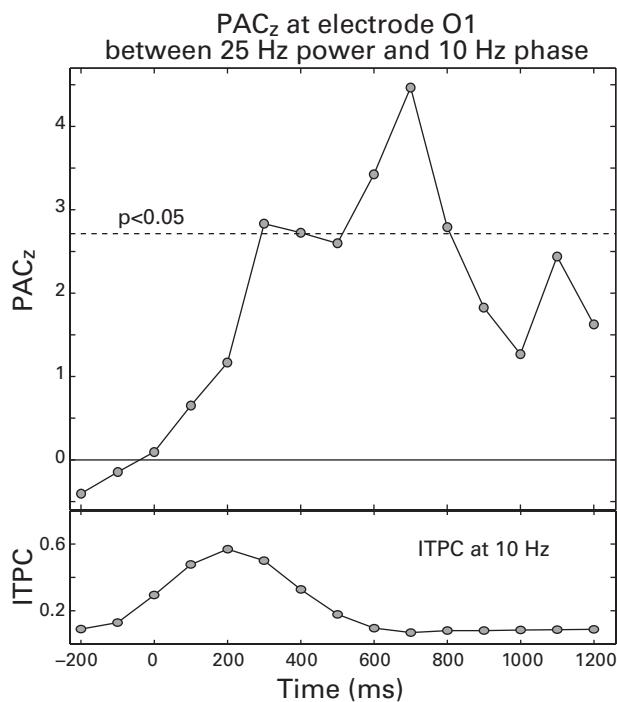


Figure 30.7

PAC_z over time, computed between 10-Hz phase and 25-Hz power from electrode O1 of the sample scalp EEG data. The dotted horizontal line corresponds to $p < 0.05$ Bonferroni-corrected for multiple comparisons across time bins. The lower plot shows the time course of ITPC, which demonstrates that the PAC_z does not temporally co-peak with stimulus-locked ITPC (discussed in section 30.4).

for highlighting task-related changes in cross-frequency coupling because only the phase-amplitude coupling results that are consistent over trials will be identified.

30.4 Separating Task-Related Phase and Power Coactivations from Phase-Amplitude Coupling

Imagine a burst of gamma power that is time-locked to a stimulus and also an increase in alpha ITPC time-locked to that same stimulus. The likely PAC result is that gamma power will be time-locked to alpha phase. This could reflect true phase-amplitude coupling, but it could also be a spurious result driven by gamma power and alpha phase being independently time-locked to the stimulus onset.

There are several ways to avoid misinterpreting this possible spurious result. First, you can plot ITPC from the lower frequency to show that PAC_z and ITPC do not temporally co-occur (as is shown in figure 30.7). Second, you can simply avoid analyzing cross-frequency coupling during time periods with strong ITPC in the lower-frequency phase or mention explicitly the time periods in which phase-amplitude coupling is difficult to interpret because of ITPC. This would be appropriate if the PAC_z shown in figure 30.7 were statistically significant around 100–400 ms. Third, you can subtract the ERP from the single-trial EEG data before computing phase-amplitude coupling, as was done to compute the non-phase-locked power in chapter 20. This will eliminate ITPC while leaving the upper frequency-power time series mostly unchanged (the power time series will remain mostly unchanged because activity from frequencies above around 20 Hz generally do not contribute to ERPs, particularly after the first 200 ms of the ERP).

30.5 Mixed A Priori/Exploratory Phase-Amplitude Coupling

So far in this chapter, frequency bands for phase and for power were selected, and PAC_z was computed only between those two bands. This is a useful approach if you can select frequency bands based on hypotheses or previous studies or based on an analysis of task-related power and phase. The advantages of this a priori approach are that it is hypothesis driven, fast (because it involves only one pair of frequency bands), and maximally sensitive to detecting an effect at the hypothesized frequency bands. The increased sensitivity is due to having only one statistical test (or a small number of statistical tests), so a *p*-value of 0.05 can be used without correcting for multiple comparisons over dozens or hundreds of frequency-frequency pairs. The main two limitations are that you cannot determine whether the effect is selective to those frequency bands and that you may not have precise hypotheses about which frequency bands to use for power or phase.

The mixed a priori/exploratory phase-amplitude coupling method is useful when you have an a priori reason to select one frequency band but would like the flexibility to find the other frequency band using data-driven techniques. There are two options—you select the lower frequency for phase and then use an exploratory approach to find the best-fitting frequency for power, or you select the higher frequency for power and then use an exploratory approach to find the best-fitting frequency for phase.

In the first example the frequency for power was selected, and an exploratory method was applied to find the best frequency for phase (that is, the frequency for phase at which the strongest coupling with the frequency for power can be observed). This exploratory method involves looping through several frequency bands (here, from 2 to 20 Hz) and repeatedly

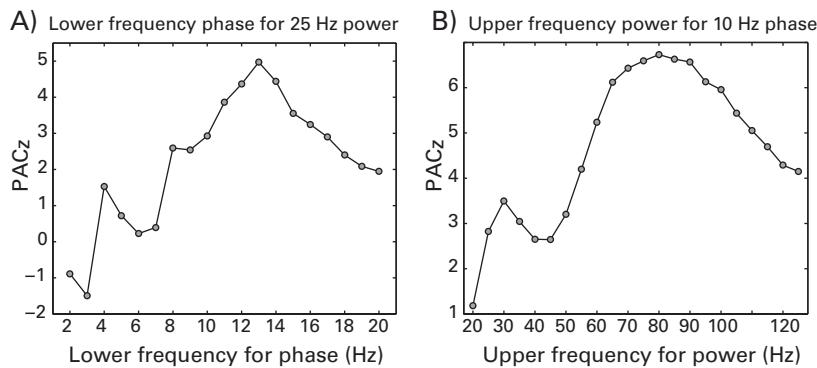


Figure 30.8

Results from mixed a priori/exploratory phase-amplitude coupling. Panel A illustrates the situation in which the frequency for power is selected a priori (in this case, 25 Hz), and an exploratory procedure is taken to identify the frequency for phase that has the strongest PAC_z with the power time series. Panel B illustrates the complementary situation, in which the frequency for phase is selected a priori (in this case, 10 Hz), and an exploratory procedure is applied to higher-frequency power.

computing PAC_z for each frequency band (figure 30.8A). It can be seen in the figure that the phase at 13 Hz was most robustly coupled with power at 25 Hz.

In the second example the frequency for phase was selected a priori, and an exploratory method was applied to determine the best frequency for power. Results are shown in figure 30.8B. Power at 80 Hz was most robustly coupled with the phase at 10 Hz. Note that because of the randomization involved in permutation testing (discussed more in section 33.4), it is possible that the frequencies with maximum PAC_z will differ slightly when this analysis is repeatedly performed. The differences, however, should be minor. For example, in different repetitions the analyses for figure 30.8B produced 75, 80, and 85 Hz as the peak frequency. Arguably, slight differences in peak frequency are not a major concern because the time-frequency decomposition parameters should be focused on temporal precision at the expense of frequency precision. Thus, there is likely little unique information between power at 75 Hz and power at 85 Hz. Using data with a higher sampling rate should improve the signal-to-noise ratio at higher frequencies.

The results of a mixed a priori/exploratory phase-amplitude coupling analysis should be treated appropriately to avoid making circular inferences (also sometimes called “double-dipping”; this is discussed more in section 35.1). For example, based on the results presented in figure 30.8A, it is inappropriate to select 13 Hz for phase and then statistically test for the coupling between 13 Hz phase and 25 Hz power using a statistical significance threshold of $p < 0.05$. This is inappropriate because the frequency band was selected based on that

frequency pair showing a strong effect. There are two ways to approach the mixed a priori/exploratory phase-amplitude coupling that avoids circular inference. The first approach is to test for statistical significance of the 13 Hz phase result while appropriately correcting for multiple comparisons across 19 frequency bands that were examined (see chapter 32 for a discussion on strategies for correcting for multiple comparisons). The second approach is to use the exploratory results as a means of selecting data for a subsequent orthogonal condition comparison. For example, imagine that figure 30.8A was generated based on all trials from all conditions in the experiment. This result shows that, across all conditions, 13-Hz phase is most strongly coupled with 25-Hz power. Rather than assessing whether that PAC_z is significant per se, you use this as a selection procedure for condition differences. That is, based on this result, you can compute phase-amplitude coupling strength between 13-Hz phase and 25-Hz power separately for condition A and condition B and then test whether the phase-amplitude coupling strength is significantly different between those two conditions. In this scenario the statistical comparison of PAC_z strengths across conditions is not biased by the procedure used to identify the two frequency bands for PAC_z .

Studies on phase-amplitude coupling sometimes show time-frequency plots of power time-locked to the lower frequency phase (e.g., Figure 1B in Canolty et al. 2006). To create these plots, perform a time-frequency decomposition to create a time-frequency power plot as you would to create a normal task-related time-frequency power plot. However, instead of using a stimulus or other experiment event as time = 0, you use the troughs (or peaks) of the lower-frequency phase angle time series. Thus, the time-frequency power plot is time-locked to an internal event (a particular phase of a slower oscillation) rather than being time-locked to an external event (a stimulus onset). To normalize the power over frequencies so that a wide range of frequencies can be viewed on the same plot, power from each frequency can be normalized to the pretrial power (as is done for typical time-frequency power plots), or it can be normalized by taking the Z-transform of each time series surrounding each trough. The latter option is a sensible approach because for phase-amplitude coupling you are not interested in the overall power of high-frequency activity relative to the power during a baseline period; rather, you are interested in the local fluctuations in power that surround lower-frequency phase dynamics. An example of a time-frequency plot is shown in figure 30.9 (color plate 23).

30.6 Exploratory Phase-Amplitude Coupling

This final method of computing phase-amplitude coupling is useful when you have no a priori reason to select either frequency band or if you want to confirm the frequency-frequency selectivity of a phase-amplitude coupling result that was defined based on a

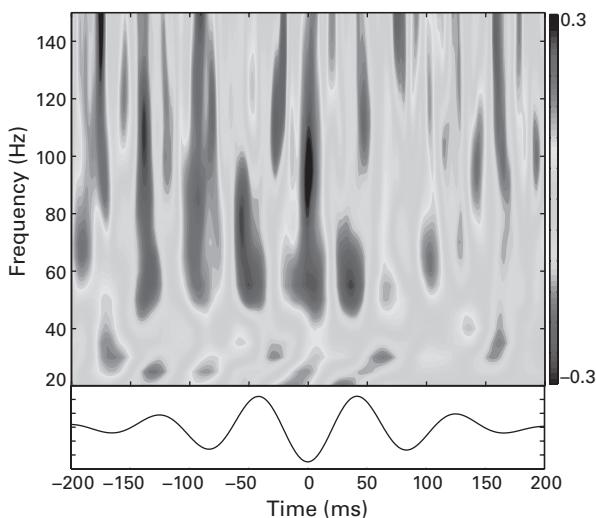


Figure 30.9 (plate 23)

Time-frequency power plot showing Z -normalized high-frequency power time-locked to alpha-band-filtered troughs. These data are from the nucleus accumbens recording.

priori-specified frequency bands. The advantage of this exploratory method is that it offers the flexibility to identify phase-amplitude coupling patterns in any pairs of frequency bands. The disadvantage is that because of the large number of tests performed, which could number in the hundreds or thousands, an appropriately conservative statistical threshold must be applied to address the multiple comparisons.

In light of the previous sections it should be sensible that, to compute exploratory phase-amplitude coupling, you combine the exploratory approaches for phase and power. This produces a two-dimensional (2-D) matrix, and the value at each point in this matrix is the PAC_z between the corresponding phase and power frequencies. An example result is shown in figure 30.10 for the nucleus accumbens recordings. The most robust feature in this frequency-frequency space is the coupling between 10- and 13-Hz phase and 55- to 105-Hz power. Thus, this 8-s segment of data replicates the effects observed over many dozens of trials and patients (Cohen et al. 2009).

30.7 Notes about Phase-Amplitude Coupling

Here are six notes and hints about computing phase-amplitude coupling. First, phase-amplitude coupling is best done using wavelet convolution or filter-Hilbert because these methods return phase angle time series (that is, estimates of instantaneous phase angles).

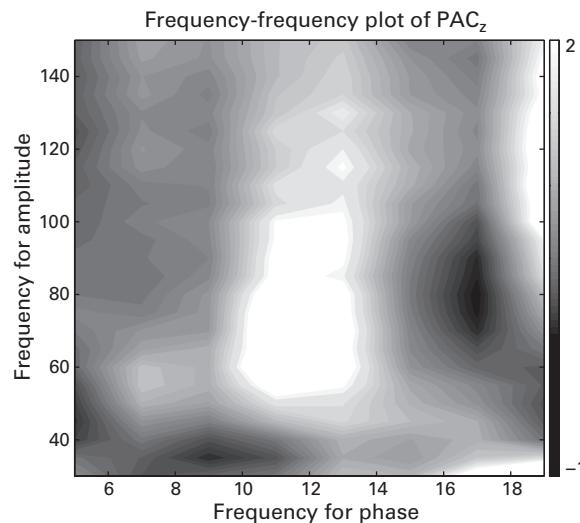


Figure 30.10

Exploratory PAC_z, which is useful (1) when you have no a priori hypotheses about which frequency band pairs to use for phase-amplitude coupling, or (2) to confirm the specificity of frequency band pairs that were selected based on a priori hypotheses.

Second, because the timing of the high frequency power is critical for the sensitivity of these analyses, you should use time-frequency decomposition parameters that highlight temporal precision over frequency precision. That is, use wavelets with a small number of cycles (three to five) or filter-Hilbert with relatively wide frequency bands.

Third, when implementing equation 30.1 in Matlab, make sure you write `abs(mean(...))` and not `mean(abs(...))` (this was also discussed in chapter 19).

Fourth, keep in mind that there is reduced signal-to-noise ratio at higher frequencies, particularly as those frequencies approach the Nyquist frequency. You can be more confident about your results if you have at least five samples of power per cycle of phase. For this reason if you plan on analyzing cross-frequency coupling, try to record the data with a relatively high sampling rate, for example, 1000 Hz. Figure 30.11 illustrates the number of time points of power per cycle of phase using a sampling rate of 256 Hz. The plot shows the logarithm of counts to facilitate comparison. If you have fewer than five power time points per cycle of phase, try to offset the balance by using more data to estimate PAC_z.

Fifth, edge artifacts from time-frequency decomposition can cause spurious phase-amplitude coupling (Kramer, Tort, and Kopell 2008). Avoid including potential edge artifact periods in the analyses and try to use clean data with no noise spikes or brief amplifier

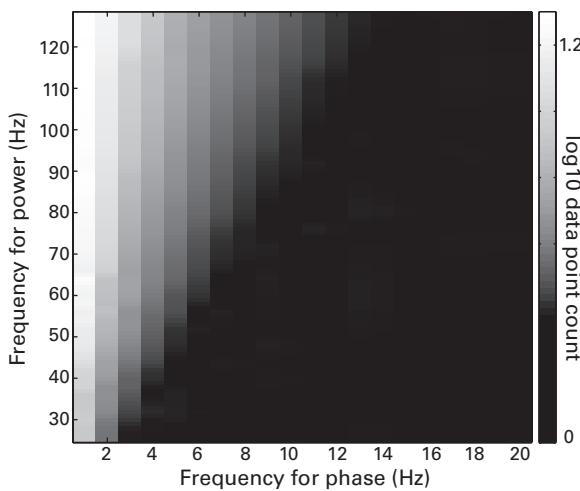


Figure 30.11

Illustration of the number of data points per cycle of frequency for power that fit into one cycle at each frequency for phase, as a function of frequencies. The numbers were scaled by logarithm-base-10 to facilitate visual inspection; the brightest white color corresponds to 20 data points. These results are based on a sampling rate of 256 Hz, so there would be about four times as many data points per phase cycle using a sampling rate of 1000 Hz. Note also that this is points per phase cycle, and normally you would have dozens or hundreds of phase cycles per analysis.

saturations that might cause edge artifacts. Permutation testing will also help to minimize the negative impact of noise spikes because those noise spikes will be paired with different phase values in different iterations during permutation testing.

Finally, phase-amplitude coupling analyses typically involve a very large search space with commensurately large multiple-comparisons concerns. One general strategy for minimizing multiple comparisons while preserving the freedom to use exploratory data-driven approaches is the following (a similar procedure was also described in section 30.5): compute exploratory phase-amplitude coupling for broad frequency ranges (e.g., 2–15 Hz for phase and 20–100 Hz for power), collapsing over all conditions. Find a region in this phase-power space that shows significant phase-amplitude coupling across all conditions, appropriately correcting for multiple comparisons (e.g., using strategies presented in chapter 33). Next, compute phase-amplitude coupling only for this frequency pair, separately for each condition. Condition differences can now be tested without multiple comparisons across all phase and power-frequency combinations. Depending on your specific design and analysis, it is possible that you will still need to correct for multiple comparisons across electrodes or

conditions or time points, but at least the multiple comparisons problem has been reduced from hundreds of comparisons to a few comparisons.

30.8 Phase-Phase Coupling

So far, power-power coupling and phase-amplitude coupling have been discussed. You can probably guess that the third type of cross-frequency coupling analysis is based on phase-phase coupling.

Consider that the upper frequency power time series itself can be conceptualized as an oscillatory signal from which phase values can be extracted. Then, phase synchronization can be computed between the phase of the upper-frequency power time series and the phase of the lower frequency (Mormann et al. 2005). This method is frequently used in the literature (Cohen 2008; Penny et al. 2008; Vanhatalo et al. 2004; Voytek et al. 2010). A graphical overview of phase-phase coupling is shown in figure 30.12. To isolate the lower-frequency component in the upper-frequency power time series, you can filter the upper-frequency power time series using the same filter characteristics as were applied to the lower frequency. This can be done by applying the filter-Hilbert method to the power

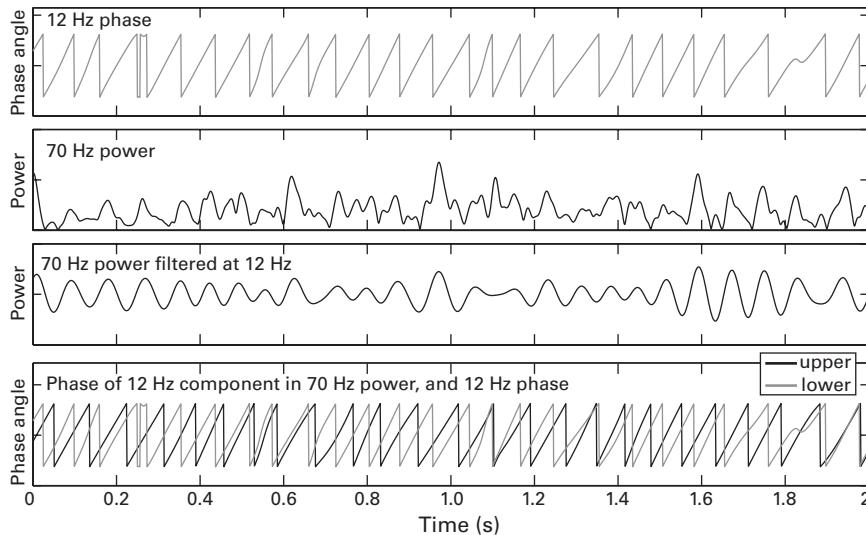


Figure 30.12

Overview of phase-phase cross-frequency coupling method. The first 2 s of 8 s of data are shown. This figure is partly modeled after figure 1 in Penny et al. (2008).

time series or convolving the upper-frequency power time series with the same wavelet used for the lower-frequency phase angle time series (in this case, the 12-Hz wavelet; this is the approach used in the online Matlab code and in figure 30.12). In the data shown in figure 30.12 the phase-phase coupling between the phase angle time series of 12 Hz and the phase of the 70-Hz power time series was 0.275, which is statistically significant considering that 8000 time points were used for the statistical measure (chapter 34 shows methods for computing statistics on mean vectors of phase values).

30.9 Other Methods for Quantifying Cross-Frequency Coupling

This chapter focuses on phase-amplitude coupling via an extension of Euler's formula, which reflects that this approach is the most commonly used in the literature. But there are several other approaches for assessing cross-frequency coupling. For example, the general linear model can be used to test whether variance in phase can account for variance in power (Penny et al. 2008). You can also compute the nonuniformity of a power distribution over phase bins (Tort et al. 2010), which is particularly useful if there are multiple power peaks in phase space. Another method of phase-phase coupling is called "n:m" phase coupling (Tass et al. 1998), which involves multiplying the phase angles from two different frequency bands by integer multiples, such as 1:2, 1:3, 1:4, etc. There are decomposition methods that might be useful in situations of low signal-to-noise ratio (Nikulin, Nolte, and Curio 2012). There has also been a recent extension of phase-amplitude coupling for multivariate networks (Canolty et al. 2012).

30.10 Cross-Frequency Coupling over Time or over Trials

The analyses discussed in this chapter are focused on cross-frequency coupling over time. This is sensible because cross-frequency coupling relies on precise timing between signals at two frequency bands, and thus, even small jitters in cross-trial timing or a lack of phase-locking to a stimulus suggests that cross-frequency coupling may be better measured using methods that can detect non-phase-locked activity. However, as with most connectivity measures, phase-amplitude coupling can also be performed at each time point over trials (Voytek et al. 2013).

30.11 Describing This Analysis in Your Methods Section

Cross-frequency coupling analyses often involve many steps and are infrequently used in the literature. Therefore, be clear about your motivations for performing cross-frequency

coupling. Make sure the description of the methods is clear because cross-frequency coupling analyses typically involve several steps. If you had a priori motivations for selecting particular frequencies, justify those choices as being based on hypotheses, previous experiments, or orthogonal data selection from within your results. A clear description of the statistical analyses and the control of multiple comparisons is particularly important because cross-frequency coupling analyses often involve a large number of tests.

30.12 Exercises

1. Online Matlab code is not provided for figure 30.9 (color plate 23). Recreate this plot using the sample data recorded from the nucleus accumbens. Here is a Matlab tip to help you identify local minima of a time series vector `data` that contains a bandpass-filtered signal (change the “>0” to “<0” to identify local maxima):

```
troughs=find(diff(sign(diff(data)))>0)+1;
```

2. Recreate figure 30.10 (exploratory PAC_z) for two electrodes from the sample scalp EEG dataset, one electrode from the front of the head and one electrode from the back of the head. Are there any noticeable features in the PAC_z from the two electrodes, and are there any frequency pairs that seem to show cross-frequency coupling, and are there any visually striking differences between the two electrodes?

31 Graph Theory

Graph theory is a mathematical framework dating back to the eighteenth century (Biggs and Lloyd 1999) that continues to be widely applied and developed in mathematics and engineering. It has been used to describe such seemingly diverse networks as forest fires, electrical grids, online social websites, and the slug nervous system. There are many analysis possibilities contained within the world of graph theory, and many books have been written about this topic (e.g., Sporns 2011). Discussed in this chapter are a few of the graph-theory-based analyses commonly used in cognitive electrophysiology that lend themselves to a physiological interpretation; this does not imply that other graph theory metrics are less applicable or less interpretable.

In this chapter all of the connectivity matrices were generated using phase-based connectivity measures (in most cases, ISPC-trials). However, you can apply the procedures and analyses in this chapter to matrices formed from any measure of connectivity that is suitable for your data (e.g., phase-based, power-based, mutual information).

31.1 Networks as Matrices and Graphs

In graph theory, networks can be represented as matrices or as graphs (figure 31.1). Matrices and graphs have vertices (also sometimes called nodes), and they have edges that represent connections among vertices. In terms of EEG connectivity, vertices correspond to electrodes, and edges correspond to some measure of connectivity between pairs of electrodes. Matrices (figure 31.1A) and graphs (figure 31.1B) contain identical information regarding the vertices and edges of the network but show the information in different ways. Matrices are useful for visualizing large networks, although they generally contain no meaningful information about relative spatial positions among vertices. Graphs can contain information regarding the topology and relative locations of the vertices (EEG topographical maps, for example,

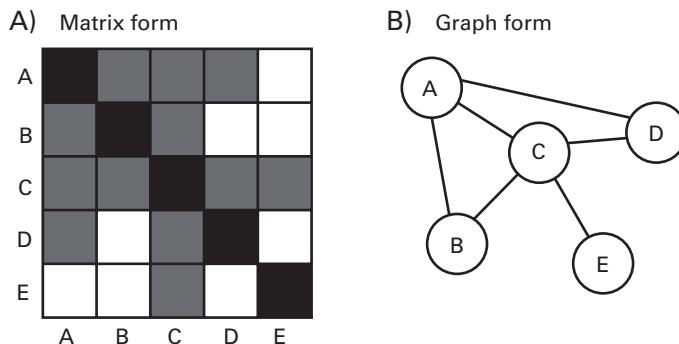


Figure 31.1

Networks and network connectivity can be represented using matrix form (panel A) or graph form (panel B). In this connectivity matrix, gray boxes indicate an edge between vertices (i.e., connectivity between electrodes), and black boxes indicate unity (each edge is connected to itself). Note that there is no necessarily inherently interpretable relationship between relative spatial positions of edges in the matrix and their position in the connectivity matrix. That is, the rows/columns of the connectivity matrix could be reordered as A, C, B, E, D, and the information contained in the matrix would not change. The matrix does not show, for example, that vertices A and C are physically closer to each other than are A and B. The graph form in panel B, on the other hand, can show meaningful topological and relative spatial relationships among vertices (as in EEG topographical maps) but may become difficult to interpret for large matrices.

can be conceptualized as graphs), but they are generally easy to interpret visually only for relatively small networks.

Figure 31.2 shows a connectivity matrix for real EEG data. The *x*- and *y*-axes both reflect electrodes, and the grayscale intensity at each pixel corresponds to the strength of phase-based connectivity between each pair of electrodes. The pixels on the diagonal are all 1.0 because each electrode has perfect phase connectivity with itself. The diagonal could also be different from 1.0 if the connectivity metric were not standardized. For example, if this were a covariance matrix, the diagonal would reflect each electrode's variance.

In an EEG connectivity matrix each electrode pair (X,Y) has two positions on the matrix: X (row) and Y (column), and Y (row) and X (column). The connectivity matrix is symmetric if the information in X,Y is identical to the information in Y,X, as if you folded the matrix along the diagonal and mirrored the lower-left triangle in the upper-right triangle. The connectivity matrix is asymmetric if the data in the lower and upper triangles are different. Asymmetric matrices can be used either to represent directional information (e.g., the lower triangle shows connectivity from X → Y, whereas the upper triangle shows connectivity from Y → X), or to show different connectivity measures (e.g., the lower triangle

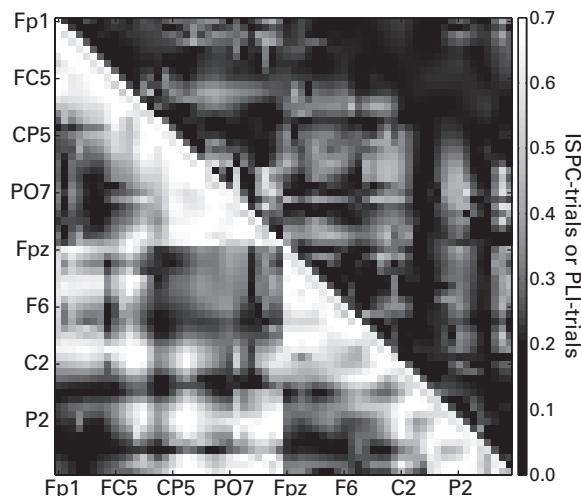


Figure 31.2

EEG all-to-all connectivity matrix. Representing connectivity in this format is a first step in many graph-theory-based analyses. The upper-right triangle of the matrix shows connectivity computed using the phase-lag index, whereas the lower-left triangle of the matrix shows connectivity computed using ISPC-trials.

shows phase-based connectivity while the upper triangle shows power-based connectivity). For example, the asymmetric connectivity matrix in figure 31.2 shows connectivity computed using the phase-lag index (upper-right triangle of the matrix) and using intersite phase clustering over trials (ISPC-trials; lower-left triangle of the matrix).

Connectivity can be computed at many time points, frequencies, and conditions, and thus, you can imagine that the 2-D connectivity matrix shown in figure 31.2 represents connectivity at one time-frequency point. In other words, there is an all-to-all connectivity hypercube through which figure 31.2 shows one slice (see section 3.2 for a discussion of data cubes, and see also figure 31.3). As you can imagine, these connectivity hypercubes can become very large.

Figure 31.1A contains pixels that take binary values (connections are present or absent), whereas figure 31.2 contains pixels that can take a range of values. In EEG data you cannot say definitively that there is or is not connectivity between any pair of electrodes; rather, you can quantify the strength of the connectivity. Although graded measures of connectivity more accurately reflect the meso- and macroscopic scale of networks measured with EEG, many graph-theory-based analyses can be simplified by using binarized matrices. For this reason it is often useful, although not always necessary, to threshold and binarize EEG

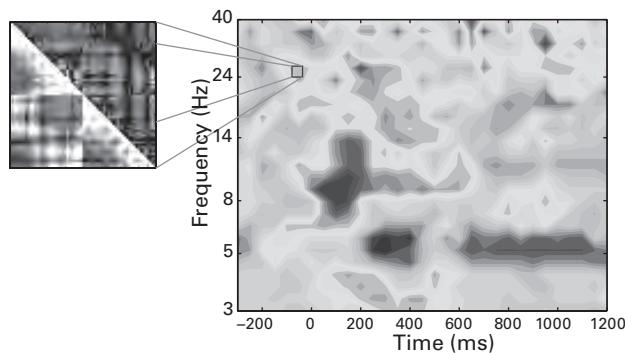


Figure 31.3

Illustration of how the electrodes-by-electrodes connectivity matrix illustrated in figure 31.2 comes from a single time-frequency point. In other words, an all-to-all connectivity matrix is one slice of a 5-D hypercube (electrodes-electrodes-time-frequency-condition).

connectivity matrices. In many cases using binarized versus graded connectivity matrices should give similar results.

31.2 Thresholding Connectivity Matrices

The connectivity matrix shown in figure 31.2 is a full matrix, meaning that each point in the two-dimensional space has a nonzero value. Connectivity matrices can also have “holes” in them if the value at pixels representing subthreshold connectivity are set to zero or NaN. These are called thresholded connectivity matrices and are useful for computing several graph theory metrics including connectivity degree, clustering coefficient, and small-world networks.

There are several ways to threshold a connectivity matrix, and the thresholding procedure that is most appropriate depends on the purpose of the thresholding. Often with graph theory analyses, the goal of thresholding is not to evaluate the statistical significance of each connection but, rather, to set up the data for subsequent graph theory-based analyses. In these cases a liberal threshold based on data distributions is appropriate. For example, you could select a threshold of one standard deviation above the median connectivity value.

Because the threshold value will change depending on the distribution of the data, it is important to choose this threshold in a manner that is independent of the condition comparisons you plan on testing. For example, you can pool together connectivity values from all conditions, then determine an appropriate threshold, and then apply that threshold to

the data to each condition separately. Pooling connectivity values from multiple conditions to determine the threshold has the added benefit that it increases the signal-to-noise ratio, thereby providing a threshold that is more robust to nonrepresentative data. Each frequency band should have its own threshold because average connectivity values are likely to change across frequency bands (for example, the strength of phase-based connectivity is likely to be lower for higher frequencies, particularly for connectivity computed over trials instead of over time).

Another option for thresholding is to specify the number of connections (sometimes referred to as k) and then keep the k strongest connections, setting the rest to zero. This approach may be useful for analyses involving topographical distributions of connectivity but may also make condition comparisons more difficult because the relative strengths of connectivity may differ across conditions. That is, if condition A has stronger overall connectivity than condition B, but the topographical regions that exhibit the connectivity are similar, fixing the k number of connections as a threshold may suggest that there are no condition differences.

For the rest of this chapter, a threshold of one standard deviation above the median connectivity value is used. This threshold seems to work well in the EEG datasets in which I have applied graph-theory-based analyses, but this threshold is based more on qualitative inspection and intuition than quantitative evidence. Nonetheless, the methods presented in this chapter could be used with any reasonable threshold. You will also see in this chapter how some graph theory results are affected by the choice of threshold.

After thresholding, connectivity matrices can be binarized by changing suprathreshold connectivity values to 1 and the subthreshold values to 0 (see figure 31.4). As mentioned earlier, binarizing thresholded connectivity matrices results in some loss of sensitivity but simplifies many analyses. For the rest of this chapter, connectivity matrices are binarized.

31.3 Connectivity Degree

Connectivity degree (also sometimes simply called degree) is a simple but informative measure of the extent to which a vertex acts as a “hub” or intersection for many connections (think of the difference in flight connections between London Heathrow Airport and Knuffingen Airport). Computing connectivity degree is straightforward: for each electrode, count the number of suprathreshold connections, regardless of the location of those connections (figure 31.5). Do not count each vertex in its own degree value. These “raw” summed connectivity degree values can be scaled to percentages or fractions of the total number of electrodes minus one (minus one because you do not count autoconnectivity). If there is a

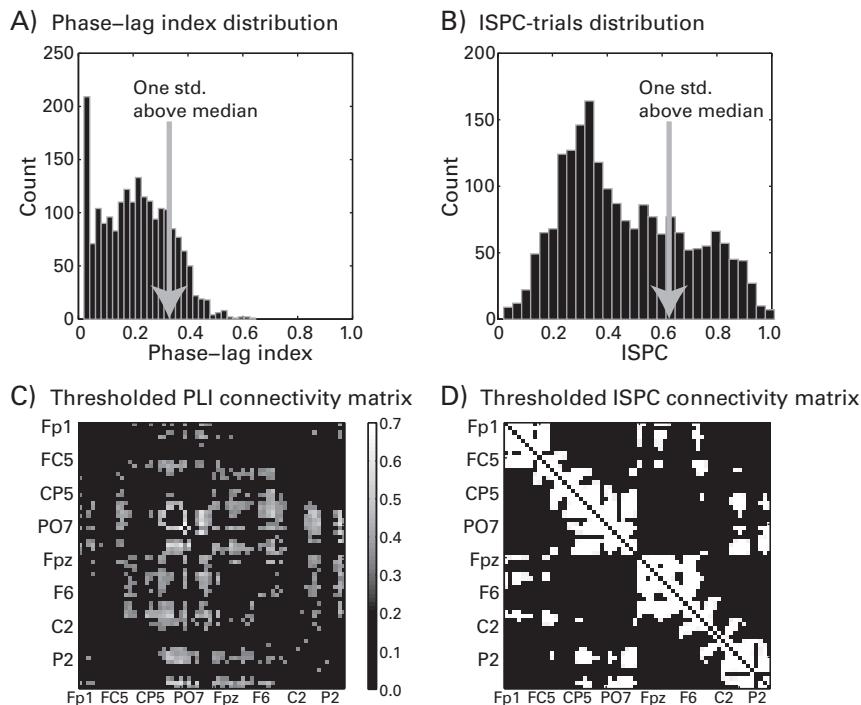


Figure 31.4

Thresholded connectivity matrices for phase-lag index (panels A and C) and ISPC-trials (panels B and D), using a threshold of one standard deviation above the median connectivity value (panels A and B). Panel C shows a graded thresholded connectivity matrix, and panel D shows a binarized connectivity matrix; binarized thresholded matrices are used for the rest of the analyses in this chapter.

constant number of electrodes for all conditions and subjects, it is not necessary to convert the raw degree values.

Once you have counted the number of suprathreshold connections for each electrode, you no longer have an electrodes-by-electrodes matrix but rather, a 1-by-electrodes vector of numbers. Because topographical differences in connectivity degree are interpretable and may be relevant for your hypotheses, connectivity degree can be shown in a topographical map. Figure 31.6 shows topographical maps of the connectivity degree computed from the connectivity matrices shown in figure 31.4.

The disadvantage of topographical maps of connectivity degree is that they do not show to which other electrodes there is connectivity; they show only which electrodes generally have more suprathreshold connections. If you want to know what the topographical pattern

A) Connect. degree in matrix

	A	B	C	D	E	sum
A	3	2	4	2	1	3
B	2	3	2	1	2	2
C	4	2	3	2	1	10
D	2	1	2	3	2	8
E	1	2	1	2	3	7

B) Connect. degree in graph

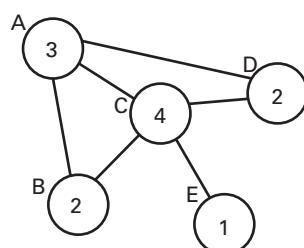


Figure 31.5

Illustration of connectivity degree represented in matrix (panel A; see light gray boxes in rightmost column) and graph (panel B; see numbers inside each circle) forms. Connectivity degree is the number of edges connected to each vertex (in EEG terms, the number of suprathreshold connectivity values for each electrode). Autoconnections are typically not counted. Degree values can be converted to proportion by dividing by the number of electrodes minus one, but this is not necessary if all subjects have the same number of electrodes.

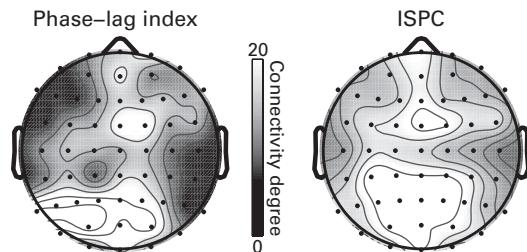


Figure 31.6

Connectivity degree plotted for phase-lag index and ISPC-trials based on the thresholded connectivity matrices shown in figure 31.4.

of connectivity looks like, pick one electrode as the “seed” (for example, the electrode that has the highest connectivity degree, or perhaps an electrode selected based on an orthogonal analysis of task-related power), and then make a topographical map of the connectivity from that seed electrode to all other electrodes.

As illustrated in figure 31.3, each all-to-all connectivity matrix is actually one slice of a larger hypercube of results. Because it is reasonable to expect that patterns of large-scale connectivity change over time, frequency, and condition, you can compute and plot the connectivity degree at different frequency bands and at different time points. Figure 31.7, for

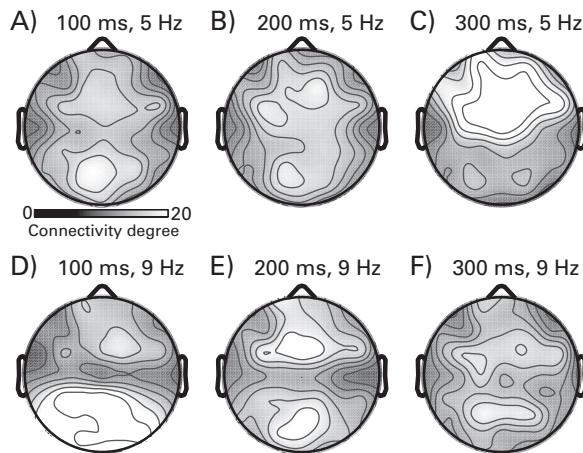


Figure 31.7

Topographical maps of connectivity degree (the number of suprathreshold connections seeded from each electrode) over time for two frequencies. All topographical maps have the same gray scale (see panel A). These results can be interpreted to indicate that different topographical regions act as “hubs” in different time-frequency windows (posterior regions for alpha-band connectivity and midfrontal regions for theta-band connectivity). Hubs can also be formally defined according to a criterion such as having greater than the median connectivity degree.

example, shows topographical maps of connectivity degree at three time points (100, 200, and 300 ms) and at two frequency bands (5 Hz and 9 Hz).

Figures 31.6 and 31.7 show the “topographical slice” of connectivity degree at select time-frequency windows. You can also focus on one electrode and examine how large-scale connectivity at that electrode unfolds over time and frequency. Figure 31.8 (plate 24) shows time-frequency plots of task-related connectivity degree changes at a frontal and an occipital electrode. This was done by computing connectivity degree at each time-frequency point, and then subtracting the average pretrial connectivity degree from each frequency to focus on task-related changes in connectivity. Note that connectivity degree is unrelated to power-law scaling over frequencies, and thus, linear baseline subtractions are appropriate (percent-age change or decibels would also be appropriate).

31.3 Clustering Coefficient

An intuitive way to conceptualize clustering coefficient is to think about social friendships. How many of your friends are friends with each other? If they are all friends with each other,

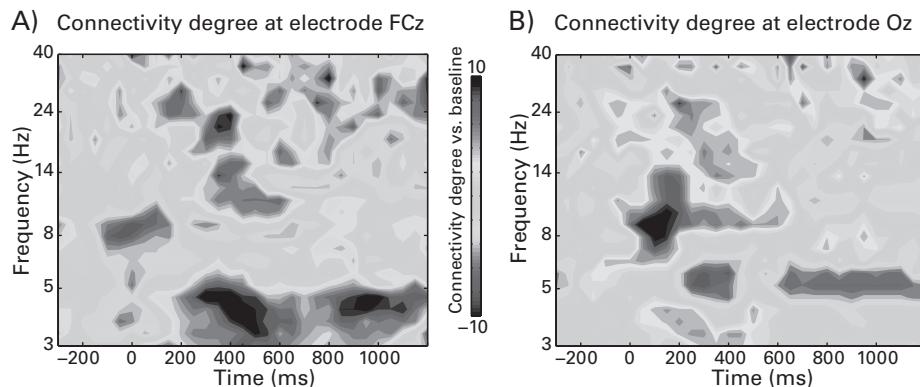


Figure 31.8 (plate 24)

Changes in connectivity degree over time and frequency, separately for frontal electrode FCz and for occipital electrode Oz. The connectivity degree from a pretrial baseline period of -300 to -100 ms was subtracted for each frequency.

you have a high clustering coefficient; if they are scattered throughout the world and do not know each other, you have a low clustering coefficient. In EEG connectivity terms, the clustering coefficient of electrode A is the proportion of electrodes that have suprathreshold connectivity with electrode A that also have suprathreshold connectivity with each other (figure 31.9). Note that here, clustering coefficient is based on suprathreshold connections among pairs of electrodes, regardless of whether those electrodes are physical neighbors. The formula to compute clustering coefficient follows:

$$clustcoef_i = \frac{2 \sum_{j=1}^n v_{ij}}{n_i(n_i - 1)} \quad (31.1)$$

in which i refers to each electrode, v_{ij} is 1 if there is suprathreshold connectivity between electrodes i and j and 0 otherwise, and n is the number of electrodes with suprathreshold connections to electrode i (note that this is not the total number of electrodes; it is the connectivity degree of electrode i). If you have directional graphs, drop the "2" in equation 31.1 and evaluate the equation separately for each direction. There are other ways to compute clustering coefficient, for example, by conceptualizing the network as a series of triangles instead of individual nodes and vertices.

In practice, an efficient way to compute the clustering coefficient is to create a "local network," comprising only electrodes with suprathreshold connections to electrode i , and

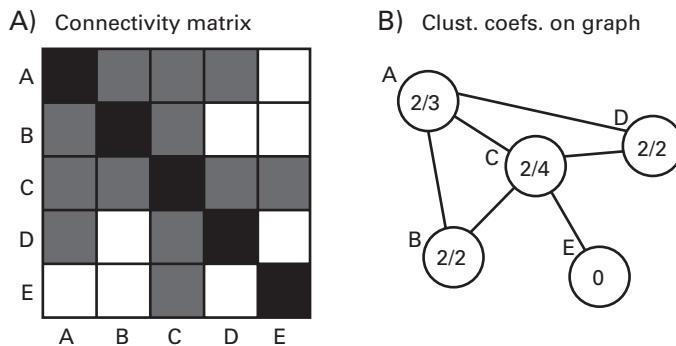


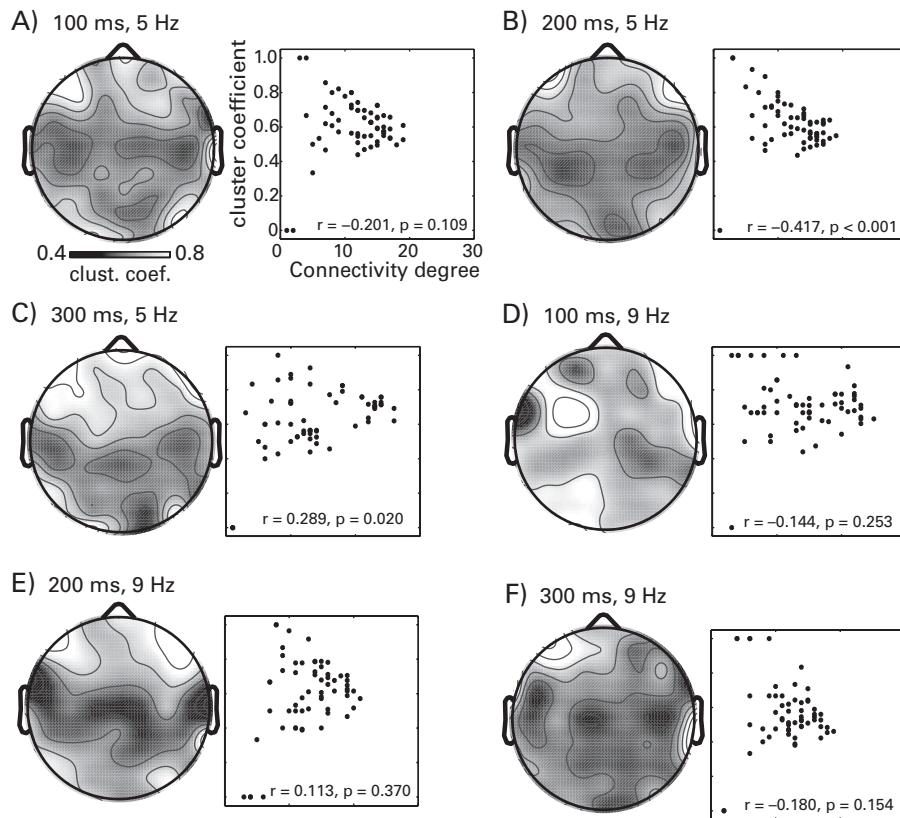
Figure 31.9

Clustering coefficient is represented as a fraction: the denominator is the number of vertices to which there are edges (i.e., connectivity degree), and the numerator is the number of those connected vertices that are also connected to each other. For example, vertex A is connected to B, C, and D (thus, the denominator is 3); B and C are connected to each other, and so are C and D, but B and D are not connected to each other. Thus, the numerator of A is 2. Edges with only one connection have a clustering coefficient of zero.

then summing and scaling the interconnections. This is about an order of magnitude faster than exhaustively looping through each connected electrode separately, and the relative increase in speed is higher when there are more electrodes. This procedure is shown in the online Matlab code.

You can average the clustering coefficients for all electrodes to obtain an aggregate measure of the local clustering coefficient across the topography, or you can show topographical maps of the clustering coefficient at each electrode. The latter is a sensible approach because clustering coefficient is a measure of network interactions that may have regional topographical specificity. Topographical maps of clustering coefficients are shown in figure 31.10, using the same time-frequency windows that were used in figure 31.7.

You may have noticed from equation 31.1 and figure 31.9 that the denominator of the clustering coefficient is the same as the connectivity degree, except in the case that there is only one connection (as was the case for electrode E in figure 31.9B), in which case the clustering coefficient is zero. This may lead you to wonder whether these two measures are independent of each other. In fact, clustering coefficient and connectivity degree are mathematically and conceptually independent of each other—they are mathematically independent because of the numerator, and they are conceptually different because connectivity degree reflects long-range connectivity whereas clustering coefficient reflects the integrity and interconnectedness of a smaller network. The scatterplots in figure 31.10 demonstrate the independence of these measures.

**Figure 31.10**

Topographical maps of clustering coefficient over selected time-frequency points and correlations between clustering coefficients and connectivity degrees. All six maps have the same gray scale range (see panel A for gray scale). Scatterplots show the relationship between clustering coefficient (y-axis) and connectivity degree (x-axis); each dot is an electrode. The r and p values are taken from a Spearman correlation. The topographical maps can also be visually compared with the connectivity degree maps in figure 31.7 because the same time-frequency points were used for both figures. In this dataset, clustering coefficient and degree are often nonsignificantly correlated, demonstrating that these two metrics provide independent information regarding network organization. The x-axis and y-axis limits on all scatterplots are the same as that in panel A.

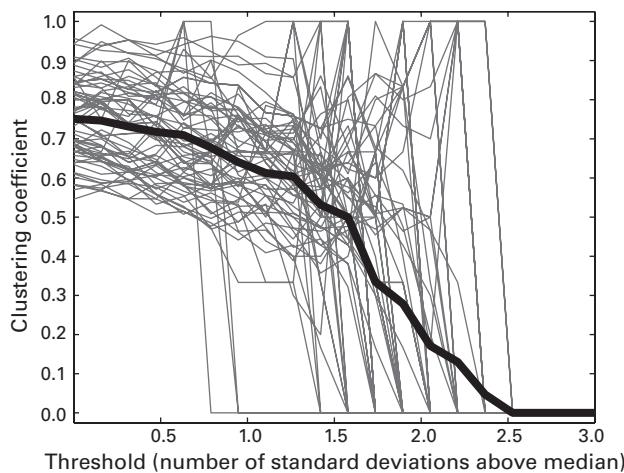


Figure 31.11

Relationship between clustering coefficient (y -axis) and the threshold used to binarize the connectivity matrix (x -axis). The threshold here is the number of standard deviations above the median connectivity value. Each gray line corresponds to one electrode, and the thick black line corresponds to the average across all electrodes. This graph shows results for one time-frequency point (300 ms, 5 Hz). Although the precise relationship between the threshold and clustering coefficient will differ across time points and frequency bands, the clustering coefficient will generally always decrease with increasing binarizing threshold.

The binarizing threshold will affect the estimate of clustering coefficient (figure 31.11). As the binarizing threshold increases, there will be fewer suprathreshold connections, and thus, the clustering coefficient will decrease. This illustrates why it is important to keep the binarizing threshold constant across all conditions.

31.4 Path Length

Path length is the average distance from each vertex (electrode) to any other vertex. It can be computed by looping through all connections from each vertex and counting how many way-points separate that vertex from all other vertices, and then averaging over all vertices (figure 31.12). There are also more efficient methods that involve using power-law increases (Taylor and Higham 2009); this is implemented in the online Matlab code.

For noninvasive EEG recordings, path length has a limited physiological interpretation because there are many nodes in the brain through which information can transfer, and many of those are not measurable with EEG (e.g., thalamic nuclei). However, path

A) Table: shortest path lengths

		TO					
		A	B	C	D	E	Average
FROM	A	-	1	1	1	2	1.25
	B	1	-	1	2	2	1.50
	C	1	1	-	1	1	1.00
	D	1	2	1	-	2	1.50
	E	2	2	1	2	-	1.75

B) Path lengths on graph

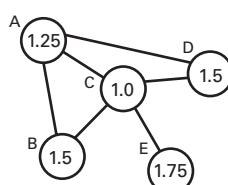


Figure 31.12

Illustration of how path length is computed. From each vertex, the minimum number of steps from that vertex to every other vertex can be computed. The path length of each vertex is the average of these minima.

length is one of the two key metrics for assessing whether a network has “small-world” characteristics.

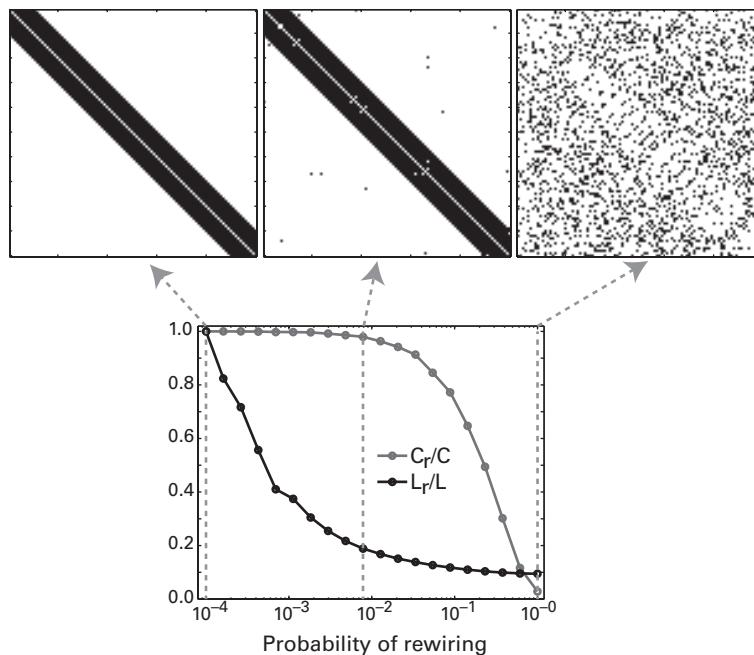
31.5 Small-World Networks

Small-world networks are characterized by higher clustering coefficient and lower path length than would be expected by chance. Many efficient networks, ranging from airline connections to electrical power grids to neural systems, seem to be organized as a small-world network (Buchanan 2003).

A good starting point for learning about how to test for small-world networks is to replicate the simulated results in Watts and Strogatz (1998) and then apply that method to EEG connectivity data. Watts and Strogatz showed that in the transition between a “regular,” or highly organized, network in which neighbors are connected to each other, to a “random” network in which connections among vertices are randomly assigned, path length drops quickly with a small increase in rewiring, whereas clustering coefficient drops more slowly. Small-world networks are characterized by properties in this transition. The transition is achieved by randomly rewiring connectivity targets from each vertex with increasing probability. Figure 31.13 shows the matrix representation of example networks, and path lengths and clustering coefficients as a function of rewiring probability.

After replicating the simulated results, we can apply this method to EEG connectivity data. An example result is presented in figure 31.14.

Is the EEG connectivity network shown in figure 31.14 a small-world network? Some aspects of the results in figure 31.14 appear similar to the simulated results in figure 31.13. For example, path length declines faster than clustering coefficient at small probabilities,

**Figure 31.13**

Simulation of the basic small-world-network topology. Top row shows connectivity matrix representations of 3 out of 20 simulated networks. Bottom row shows the clustering coefficient and path length, scaled by those values from the highly organized network, as a function of rewiring probability (this is a replication of figure 2 in Watts and Strogatz 1998). An organized network is one in which each vertex is connected to its nearest neighbors. Following Watts and Strogatz, 1000 vertices, each with 10 edges, were used for this simulation, and results are averaged over 10 simulated networks. The matrices shown in the top row contain only 100 vertices to facilitate visual inspection of the effect of rewiring. Dotted vertical lines show where in the rewiring probabilities the matrices are drawn from (note that the x-axis has a logarithmic scale). C and L indicate clustering coefficient and path length from the organized network, and the subscript r indicates networks with probabilistic rewirings.

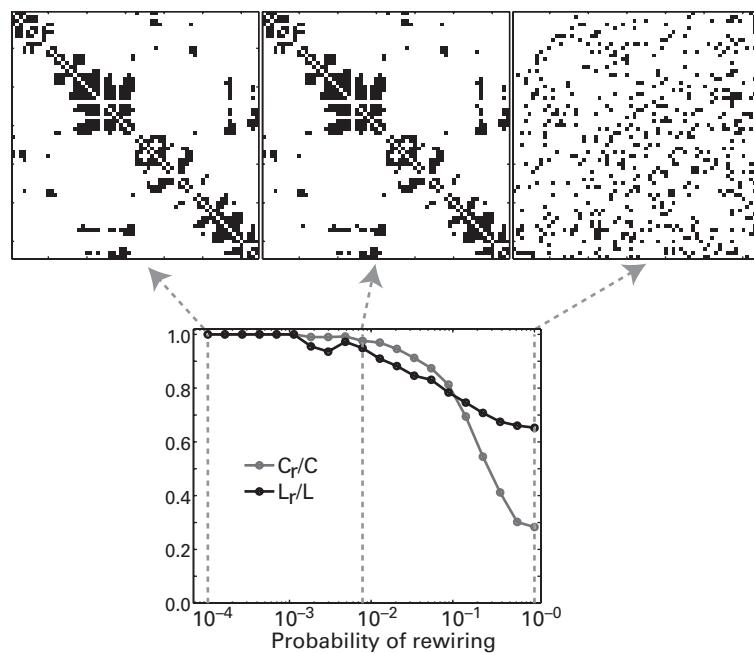


Figure 31.14

The analyses applied to simulated networks shown in figure 31.13 were also applied to EEG connectivity data. The connectivity matrix used here is from 6 Hz and 300 ms poststimulus, and the binarizing threshold was one standard deviation above the median connectivity value.

and then clustering coefficient becomes lower than path length for very high rewiring probabilities. A formal test for evaluating whether a network conforms to a small-world network (Humphries and Gurney 2008) involves a double ratio: the numerator is the ratio of the observed average clustering coefficient to that of a random network, and the denominator is the ratio of the observed average path length to that of a random network.

$$swn = \frac{C/C_r}{L/L_r} \quad (31.2)$$

in which C refers to clustering coefficient, L refers to path length, and the subscript r refers to a random network. If the swn value is greater than 1, the network can be said to exhibit properties consistent with a small-world network. You can see that middle-range values of figure 31.13 would result in large swn values because the scaled clustering coefficient is larger than the scaled path lengths. Note that for equation 31.2, random networks should be defined by

Erdős-Rényi random graphs (Erdős and Rényi 1960) and not the $p = 1$ rewired graphs used in figures 31.13 and 31.14; the latter graphs produce smaller clustering coefficients than path lengths (see figure 31.13 and also figure 2 in Watts and Strogatz, 1998). Erdős-Rényi graphs are constructed by fixing the number of vertices and edges while setting the connections randomly. Note that equation 31.2 is not the only formal evaluation of a small-world network (e.g., Telesford et al. 2011), but it has been used for neural networks and therefore is highlighted here (Humphries and Gurney 2008).

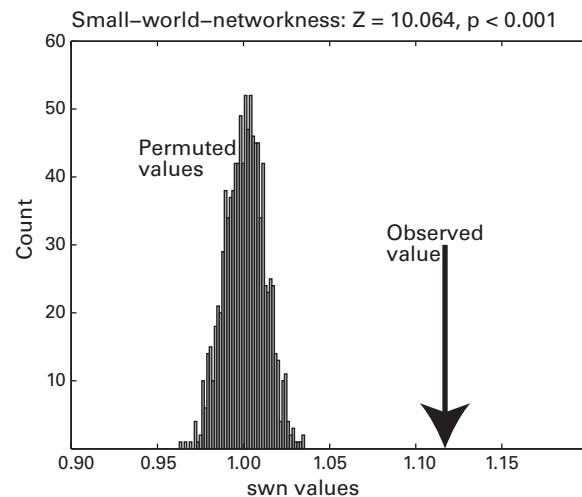
The statistical significance of the swn value in equation 31.2 can be evaluated via permutation testing. At each iteration during the permutation testing, a random network is created by a network with the same number of connections as the observed network, but those connections are randomly rewired. Next, the swn value is recomputed, replacing the C and L terms in equation 31.2 with other random networks. After equation 31.2 has been evaluated repeatedly over many random networks, a distribution of swn values is created under the null hypothesis that the networks have a random organization. Next, the swn value observed in the unadulterated connectivity data can be compared against this null hypothesis distribution to obtain a statistical value in normal Z standard deviation units. (This is done by subtracting the mean of the distribution and dividing by the standard deviation.) This procedure is shown in the online Matlab code, and the results of one permutation test are illustrated in figure 31.15.

Whether a network can be considered significantly small-world-like is influenced by several factors, including the connectivity method used, network size (small-world-ness increases as a function of the number of nodes; Humphries and Gurney 2008), and other parameters (Telesford et al. 2011). Small-world-ness is also sensitive to the binarizing threshold. This is shown in figure 31.16. Here, small-world-ness was computed via permutation testing as described above for a range of binarizing thresholds. As the threshold increased, which decreased the number of edges, the network became decreasingly identifiable as a small-world network (also see Humphries and Gurney 2008 for more examples).

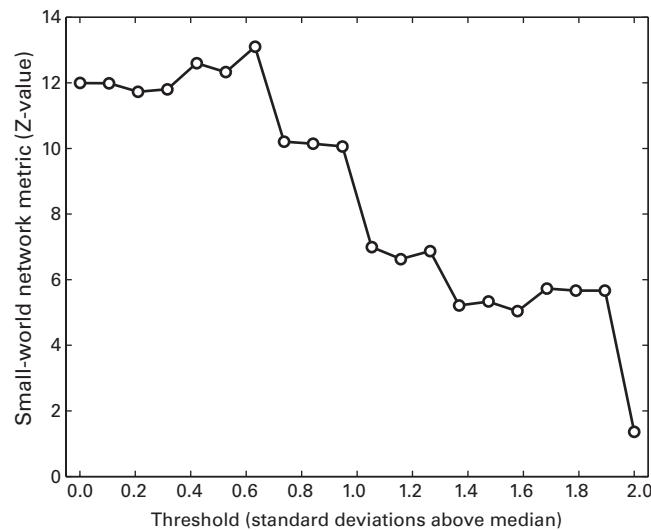
Because statistical measures of small-world-ness depend on parameter selection, it is important to select parameters in a manner that is independent of your hypotheses and condition differences, and it is important to apply the same parameters to all time points and conditions. However, it may be useful to allow some parameters to vary across frequencies; otherwise, low-frequency connectivity may dominate the results.

31.6 Statistics

Many graph-theory-based metrics are not appropriate for parametric statistical analyses because they are not normally distributed. This includes connectivity degree, clustering

**Figure 31.15**

Permutation testing of small-world-ness. The normalized distance from the observed *swn* value (obtained via equation 31.2) to the distribution of null hypothesis *swn* values can be interpreted as a normal-Z value and evaluated for its statistical significance or used in parametric group-level analyses.

**Figure 31.16**

As the binarizing threshold increases, the statistical significance of the small-world-ness of the network decreases.

coefficient, and path length. However, condition differences, or changes from a baseline period, can be negative, have a null hypothesis value of zero, and may be normally distributed and used in group-level parametric statistical analyses. As with many other measures of EEG dynamics, nonparametric permutation testing is an appropriate statistical approach for the graph theory-based analyses presented in this chapter, particularly if the graph-theory-based analyses are used for data-driven exploratory analyses. As discussed further in chapters 33 and 34, permutation testing is appropriate because no assumptions are made regarding the distribution of the data and because permutation testing is amenable to correction for multiple comparisons over many time-frequency points.

31.7 How to Describe These Analyses in Your Paper

Although many graph-theory-based analyses are not very complicated, they often involve many analysis steps. Describe each step clearly and include the parameters used at each step. Furthermore, if you tested a range of parameters (such as the binarizing threshold), report that range and whether the results of interest changed with different parameter values. Consider including a figure that outlines each step of the analyses.

31.8 Exercises

1. Pick two frequency bands and two time segments (e.g., pretrial and poststimulus or early poststimulus and late poststimulus). Compute connectivity degree and clustering coefficient separately for the first 40 and the last 40 trials from the two frequency bands and the two time segments, and show the results in topographical plots. Are there visually salient differences in network properties between the first and the last 40 trials? How would you interpret the results if they were statistically significant?
2. Pick two frequency bands and test whether small-world-ness changes over time in those frequency bands. Make sure the binarizing threshold is constant over time within each frequency, but it can be different for the two frequency bands.

VI Statistical Analyses

32 Advantages and Limitations of Different Statistical Procedures

Throughout most of this book, statistics have been largely ignored aside from the frequent but brief references to chapters 32–36. Instead, results were interpreted qualitatively based on visual inspection. The next three chapters introduce you to the statistical approaches that are commonly used in cognitive electrophysiology. This does not imply that these are the only or the best statistical approaches, but they are the statistical procedures you are most likely to encounter in the literature and perform on your cognitive electrophysiology data. For more general resources on statistics in biology and psychology, there are several excellent books, including Zar (1999), Cohen and Cohen (no relation to the author; 1983), and Maxwell and Delaney (2004). This chapter provides an introduction to statistics in cognitive electrophysiology and descriptions of the statistical methods covered in detail in the next several chapters.

32.1 Are Statistics Necessary?

Qualitative visual inspection should not be discouraged, and its importance in science should not be underestimated. But qualitative visual inspection should also not be overly trusted, and strong claims based on visual inspection should be buttressed by statistical support. One example from this book was presented in chapter 30, when there appeared to be cross-frequency coupling based on visual inspection of one trial, but this was not a robust finding over many trials (figure 30.1).

On the other hand, there is also a danger of relying too much on statistics and statistical thresholding. Blindly applying a statistical threshold without inspecting the results may lead to a poor representation of the underlying landscape of time-frequency-space dynamics (also see a related discussion in section 36.3). Thus, results should be inspected for qualitative patterns, not only for whether the p -value is lower than the statistical threshold. Consider the condition difference bar plot in figure 32.1. Although this bar clearly shows that there is a

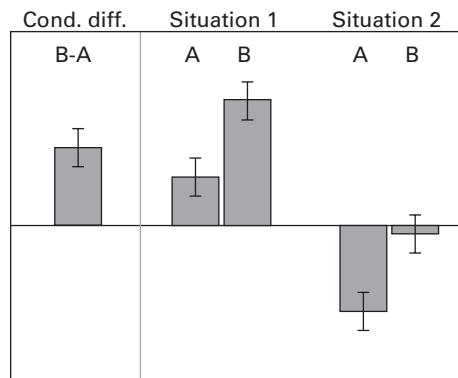


Figure 32.1

The same condition difference effect (left panel) could be driven by different patterns of within-condition activity. Qualitative inspection of the data (right panel) will facilitate an appropriate interpretation.

condition difference, this result could be equally likely driven by two different situations that would be interpreted in different ways; only qualitative inspection of the data would reveal which of two interpretations is appropriate. Thus, statistics are important, but they provide an incomplete understanding of the nature of the findings without a qualitative inspection of the results.

Ultimately, statistics are just tools in a toolbox. On their own, they cannot answer research questions, they cannot link results to theory or help develop new and better theories, and they cannot design new and better experiments. Only you, the scientist, and the people who hear about and read about your work can answer research questions, bridge findings across theories and experiments, and build better theories. The purpose of statistics is to help guide you to know which results should be interpreted and how much confidence you should have in those interpretations.

32.2 At What Level Should Statistics Be Performed?

“Level” here refers to what is considered the unit of data for statistical analysis. Within-subjects statistics (also called level-1) consider the trial to be the unit for analysis; group-level statistics (level-2) consider the subject to be the unit for analysis.

Within-subjects statistics are not often performed in cognitive electrophysiology but can be necessary or useful in some situations. For example, evaluating the statistical significance of within-subjects effects is useful for studies that have a small number of subjects,

including studies involving rare patient groups, or studies involving many trials from a small number of human or animal subjects. In these cases group-level statistics might prove difficult because of the small sample size, whereas within-subjects statistics can take advantage of the larger number of trials. As a general rule of thumb, if you have fewer than eight subjects, you should consider performing within-subjects analyses. With few subjects, group-level analyses might lack the sensitivity to detect even modestly sized effects. For example, a *t*-test against zero on the numbers 2.3, 1.4, 1.5, 0.8, 1, and -1 yields a *t*-statistic of 2.21 with a *p*-value of 0.0781. Thus, the data are not statistically significantly different from zero, although only one data point is below zero. In this case a group-level analysis with six subjects does not reflect the group-level characteristics.

Within-subjects statistics are also useful for supporting claims about the robustness of an effect. That is, an effect can be qualitatively considered more robust if it is statistically significant within 13 out of 15 subjects, compared to an effect that is not statistically significant within any subject individually but is statistically significant at the group level. In this latter scenario the within-subject variance may be too large to detect a true, although subtle, effect.

Finally, within-subjects statistics may be necessary for some analyses in which the “raw” analysis values are either uninterpretable or uncomparable across subjects. One example (among several presented in this book) is phase-amplitude cross-frequency coupling (chapter 30), for which raw values must be transformed to standard Z values at the within-subject level before interpretation and comparisons. In these situations the goal of performing within-subjects statistics is not necessarily to evaluate the statistical significance of an effect within each subject but, rather, to convert the within-subject value to a metric that is amenable to group-level comparisons and statistical analyses.

The more commonly used approach in cognitive electrophysiology is to average data from all trials within each subject and then perform group-level analyses on the trial-average data across subjects. The assumption underlying this approach is that single trials are noisy, and when many trials are averaged together, noise is attenuated while signal is left intact. Group-level analyses may have only 10–20 data points per test (one data point per subject), compared to hundreds of data points per test for within-subject analyses (one data point per trial), but trial-averaged data points have a higher signal-to-noise ratio than do individual trials.

There are statistical methods such as hierarchical linear modeling that include both within- and across-subject variance, but these methods are rarely used in the electrophysiology literature and are not discussed further here.

It is important to realize that within-subjects and group-level analyses have different goals and different interpretations. Within-subjects analyses provide information regarding the

cross-trial variability of an effect relative to the magnitude of the effect; they provide no information regarding the generalizability of the effect to other subjects. Group-level analyses, on the other hand, provide information regarding the consistency of the direction of the effect across the group of subjects, and provide little information regarding the within-subject variability.

For this reason, you should not confuse the two and think that within-subjects and group-level analysis results should necessarily go hand-in-hand. For example, imagine that you have 20 subjects in your experiment, and each subject individually shows a statistically significant difference between conditions A and B. However, one half of the subjects show greater activity in A compared to B and the other half show greater activity in B compared to A. The group-level analysis will be non-statistically significant, even though each subject has a statistically significant effect. Now imagine that you have few trials in your task and thus high cross-trial variability. It is possible that each subject individually shows the effect of A greater than B, but that the effect is not statistically significant in any individual subject. However, because all subjects show the effect in the same direction, the group-level analysis will reveal a highly statistically significant effect of A greater than B.

32.3 What *p*-Value Should Be Used, and Should Multiple-Comparisons Corrections Be Applied?

The *p*-value thresholds are arbitrary. There is little difference in terms of effect sizes between $p = 0.051$ and $p = 0.049$. In practice, however, a result associated with $p = 0.051$ is considered a trend or marginally significant, whereas a result associated with $p = 0.049$ is considered statistically significant. This can be frustrating and may lead to temptations to test more subjects or change analysis parameters until the *p*-value drops by what may seem a mathematically trivial amount (needless to say, any manipulations to data or analyses that critical observers would find questionable should not be done). Nonetheless, $p = 0.05$ is the accepted significance threshold, and any other statistical threshold would be equally arbitrary. *p*-Value thresholds smaller than 0.05 are also accepted, such as $p = 0.01$ or $p = 0.001$. *p*-Value thresholds that do not end with 5 or 1, such as $p = 0.0323$, are infrequently used and may generate confusion and suspicion.

The appropriateness of your threshold depends in part on the goal of the analysis. Hypothesis-driven research generally involves a small number of statistical comparisons that are motivated by theories or previous results, and thus, $p = 0.05$ can be an acceptable threshold. Exploratory data-driven analyses, in contrast, often involve thousands or tens of thousands of statistical comparisons that may have a loose or no relationship to hypotheses

and theories, and thus, the *p*-value threshold should be more stringent or used in combination with multiple-comparisons corrections.

Correcting for multiple comparisons is often necessary with electrophysiology data because of the large number of tests (over electrodes, time points, frequency bands, and so on). There are several ways to correct for multiple comparisons, and the method that is most appropriate depends on the analysis being performed, the goal of the statistical thresholding, whether the data are autocorrelated (which influences whether the multiple tests are independent), and whether the analyses are exploratory or hypothesis driven. Perhaps the most common way to correct for multiple comparisons in psychology is the Bonferroni correction, which involves dividing the *p*-value by the number of statistical comparisons. Although the Bonferroni correction is appropriate for cognitive electrophysiology data in some cases, it is inappropriate in many other cases. In part this is because Bonferroni correction assumes that the tests are independent of each other (time-frequency results are usually autocorrelated), and in part because the Bonferroni method corrects for the number of tests, not the amount of information available in those tests. In these cases nonparametric permutation testing provides an appropriate control for multiple comparisons. This is discussed in greater detail in section 33.5.

It is not necessary to use only one significance threshold across all analyses, although the same threshold should be used for different tests within each kind of analysis. That is, if you have a priori hypotheses to test but you also wish to perform subsequent exploratory analyses, you can apply one statistical threshold for all of the hypothesis-driven tests and a different, perhaps more conservative, threshold for all of the exploratory analyses. Make sure this is made clear to your audience, so there is no confusion about what statistical thresholds were used for each analysis.

32.4 Are *p*-Values the Only Statistical Metric?

No. There is arguably an overreliance on *p*-values for interpreting the significance of results. This is, on the one hand, sensible: a *p*-value is a standardized and widely used metric, and it is easy to interpret, evaluate, and compare across findings and publications. But *p*-values have disadvantages. Perhaps the main disadvantage is that the *p*-value (when derived through parametric statistical procedures) is based on the number of data points, which means that very small effects can have very small *p*-values if there are enough data points, and it also means that large effects can have large *p*-values if there are too few data points.

p-Values should not be dispensed with, in part because any replacement would have its own set of disadvantages and arbitrary cutoffs. But one can make the case that *p*-values

should, when appropriate, be supplemented by other statistical metrics, such as measures of effect size, signal-to-noise ratio, odds ratio, sensitivity, receiver-operating-characteristic, or η -squared (this is not an exhaustive list). Again, the argument is not that *p*-values should not be used but, rather, that you should not assume that the *p*-value is the only or even the best measure of the statistical robustness for all analyses. Therefore, you should report not only whether a *p*-value was below or above a specific threshold but also other relevant information including test statistic values (e.g., *t*-values or correlation coefficients), condition means and standard errors, and so on.

32.5 Statistical Significance versus Practical Significance

A finding is statistically significant if its test statistic value exceeds a predetermined threshold. A computer can determine whether a finding is statistically significant without any user intervention or interpretation.

Practical significance, in contrast, is more subjective and open to interpretation. A finding is practically significant if it has implications for a hypothesis or theory. The practical significance of a finding does not necessarily coincide with its statistical significance. This is part of the reason why a finding with a *p*-value of 0.045 may be cited hundreds of times and may inspire new research and new experiments, whereas a finding with a *p*-value of 0.000000001 may never be cited or influence any research programs. Furthermore, two different scientists cannot disagree about whether a finding is statistically significant, but they can (and often do) debate whether a finding is practically significant, particularly if that finding has implications for the utility or the veracity of one of those scientists' theories.

Statistical significance may also be trivial and obvious and therefore have no practical significance. For example, imagine you are testing the statistical significance of the voltage deflection of an ERP recorded from an occipital electrode 200 ms after a visual stimulus appeared on the screen. The voltage value at 200 ms is likely to be highly statistically significantly different from zero. But this finding has no practical significance because it is a trivial result. However, if the time point at which the deflection first becomes significantly different from zero depends on whether subjects receive a reward associated with that stimulus or if the amplitude of the ERP is statistically significantly correlated with schizophrenia symptoms, that finding may have practical significance.

Whether a finding has practical significance also has implications for how strongly that finding can be interpreted. A finding with a *p*-value of 0.051 might be appropriate to interpret (albeit cautiously) if it is predicted by a theory or motivated by previous research, whereas that same *p*-value might be inappropriate to interpret in context of a hypothesis-free exploratory analysis for which there are no prior expectations regarding that finding.

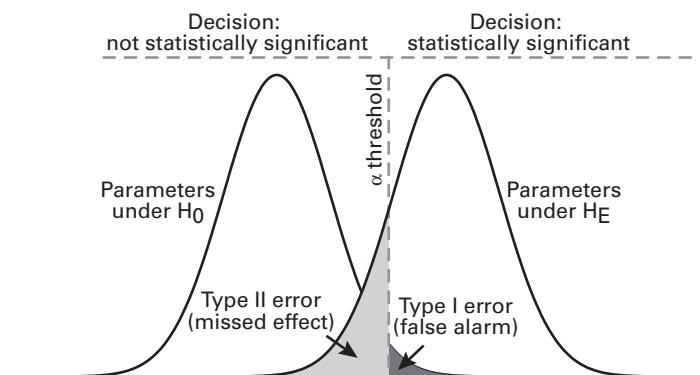


Figure 32.2

Illustration of Type I and Type II errors. H_N and H_E refer to the null hypothesis and the effect (or alternative) hypothesis.

32.6 Type I and Type II Errors

Type I errors refer to reporting effects as if they are true effects although they are not true effects. These are also called false alarms (figure 32.2). False alarms can result from sampling error, bias, or having too little or too much data (because p -values are related to the number of data points in the test). Using a statistical threshold that is too lenient, or not appropriately correcting for multiple comparisons, will increase the risk of Type I errors.

Type II errors refer to failing to identify true effects. This can occur if the statistical threshold is too high, if the effects of interest are relatively subtle, or if the variability in the data is too high.

These two types of errors are correlated: when the statistical threshold is stringent (see dotted gray line in figure 32.2), the risk of Type I errors decreases while the risk of Type II errors increases. Conversely, when the statistical threshold is too lenient, the risk of Type I errors increases while the risk of Type II errors decreases. Where to set the balance between Type I and Type II errors depends in part on the goal of the study. For exploratory studies, it is arguably better to avoid interpreting false alarms, even if that means failing to identify a true effect (in part because it might be difficult to interpret effects without relevant hypotheses or theories). For hypothesis-driven studies, in contrast, it is arguably better to have the sensitivity to identify true effects, even though some results might be incorrectly labeled as statistically significant. Indeed, for hypothesis-driven studies, unexpected but statistically significant findings are sometimes reported but not interpreted. These unexpected findings might be false alarms, or, if they are consistently observed in many studies, might reflect true

findings that challenge theories to be advanced to incorporate the unpredicted but consistently observed effects.

32.7 What Kinds of Statistics Should Be Applied?

The main difference between parametric and nonparametric statistics concerns the assumption regarding the distribution from which the data and their parameters were drawn. With parametric statistics the assumption is that data, and the parameters that characterize those data, were drawn from a known distribution (for many statistical procedures, a Gaussian distribution), and thus, statistical parameters such as *p*-values are computed based on a theoretical distribution that is analytically computed—that is, once a test statistic is obtained, the original data are no longer required to compute the *p*-value. With nonparametric statistics there is no assumption made about the population distribution from which the data were drawn. Nonparametric statistics are often evaluated using permutation testing, which involves computing statistical parameters based on the observed distribution of data points rather than on theoretical distributions (more details on this distinction are made in section 33.1). Thus, *p*-values are derived not from theoretical distributions but, rather, from distributions that are created from the data, by creating situations within the dataset that could arise if the null hypothesis were true.

Should you use parametric or nonparametric statistical approaches? In some situations, the choice is clear: if you have nonnormally distributed data, nonparametric tests are more appropriate; if you are performing exploratory data-driven analyses and need methods for correcting for multiple comparisons at the pixel or cluster level, nonparametric tests might prove beneficial; if you have a mixed fixed- and random-effects design that produces a 3×4 ANOVA for which you have hypotheses about main effects and interactions, parametric statistics will better suit your statistical needs. If parametric statistics are necessary or appropriate but the data are nonnormally distributed (thus violating one of the assumptions of parametric statistics), there are often transformations that can make the data have a normal distribution, such as baseline correction for time-frequency power. For some data that are inherently nonnormally distributed, such as phase-amplitude coupling, applying nonparametric permutation testing can be used to convert nonnormally distributed data (in this case, PAC from equation 30.1) to normally distributed data that are appropriate for parametric statistics (in this case, PAC_z).

Parametric statistics have several advantages. They are widely used in a large range of disciplines, including psychology, economics, and other fields that often interface with cognitive electrophysiology. There are more options for custom-tailoring analyses with

parametric statistics, in part because considerably more statistical and theoretical development goes into parametric statistics compared to nonparametric statistics (Kiebel, Tallon-Baudry, and Friston 2005). Finally, even some basic analyses such as factorial designs or interactions among fixed and random factors are difficult to perform with nonparametric permutation testing. The approach advocated here is to use standard parametric statistics (when possible and when the data appropriately meet assumptions of parametric tests) for hypothesis testing and to use nonparametric permutation testing when there are no hypotheses constraining time-frequency windows or when parametric statistics are not appropriate. One practical disadvantage of parametric statistics is that there are few routines in Matlab to perform the kinds of statistics often performed in psychology, such as ANOVAs. Thus, to perform parametric statistics it might be necessary to export data from Matlab to R, SPSS, SAS, or some other statistical software package. This is another motivation for performing parametric statistics on only a small number of hypothesis-driven tests.

Bayesian statistics are rarely applied in cognitive electrophysiology. The idea of Bayesian statistics is to describe findings in terms of your belief in the finding, based on prior information as well as the observed results. In many situations results from Bayesian statistics will lead you to the same conclusion as results from “frequentist” statistics (those based on p -values). In some cases Bayesian statistics may be advantageous, for example, if you want to estimate the distribution of likely parameters rather than the value of the most likely parameter. If you would like to apply Bayesian statistics to cognitive electrophysiology data, SPM8 has statistical routines based on Bayesian methods, and there are several books about Bayesian statistics, as well as Matlab third-party toolboxes that will help you implement Bayesian methods.

32.8 How to Combine Data across Subjects

In many analyses data are simply averaged across subjects, time point by time point, frequency by frequency, and electrode by electrode. The assumption is that, for example, the activity recorded from electrode POz at 13 Hz and 190 ms reflects the neural process from the same functional brain source in every subject. Is this a valid assumption and an appropriate way to pool data across individuals? Obviously, it is a successful approach, because most studies use it. It is successful in part because of the smoothing properties of time-frequency decomposition methods and the spatial smoothing resulting from volume conduction. Thus, if one subject’s alpha peak frequency is 9 Hz and another subject’s alpha peak frequency is 11.4 Hz, spectral smoothing from wavelet convolution or band-pass filtering will help ensure that the data can be compared across the two subjects. This is why having some smoothing

in the time-frequency decomposition is beneficial and is not necessarily a limitation of time-frequency analyses. This is analogous to how spatial smoothing of fMRI activation maps helps account for individual differences in functional and anatomical organization.

However, pooling data across subjects in this fashion might not always be an appropriate strategy. If there is a lot of intersubject variability in the frequency band or in the time course of the activation, it is possible that group-level results would not be statistically significant even if the effect is present and robust in each subject individually. Fortunately, it is also possible and appropriate to define time-frequency windows for each subject, based on subject-specific time-frequency characteristics. This approach allows increased sensitivity to identify findings that may be more subtle or more dependent on individual factors. More is presented on this topic in section 35.4.

33 Nonparametric Permutation Testing

Permutation testing was mentioned in several previous chapters, and brief explanations were given in chapters 19 (wITPC), 30 (phase-amplitude coupling), and 31 (small-world networks). This chapter covers the details of nonparametric permutation testing, including in-depth discussions of parameter settings, options, and methods to use the permutation-testing framework for statistical evaluation and for correction for multiple comparisons. This chapter contains only theory and explanation of how nonparametric permutation testing works. However, all of the methods described in this chapter are applied to real data in the next chapter. Thus, Matlab code for implementing the ideas in this chapter can be found in the Matlab code for chapter 34. The two most relevant publications from which this chapter draws are Maris and Oostenveld (2007) (nonparametric permutation testing applied to M/EEG data), and Nichols and Holmes (2002) (nonparametric permutation testing applied to MRI data).

33.1 Advantages of Nonparametric Permutation Testing

Permutation testing is a useful framework for assessing the statistical significance of EEG results, particularly time-frequency results (Maris and Oostenveld 2007). The two main advantages of permutation testing are that it does not rely on assumptions about the distribution of the data or their parameters and that appropriate corrections for multiple comparisons can easily be incorporated into the analysis (Theiler et al. 1992). This can be contrasted with parametric statistics, which rely on assumptions about the data and the statistical parameters that characterize those data, typically that they are drawn from a normal (Gaussian) distribution (figure 33.1), and for which corrections for multiple comparisons that are often applied may be inappropriate for EEG data, for reasons discussed in section 33.5.

For parametric statistical testing, the test statistic (e.g., the t -value or χ^2 -value or correlation coefficient) is compared against a theoretical distribution of test statistics expected

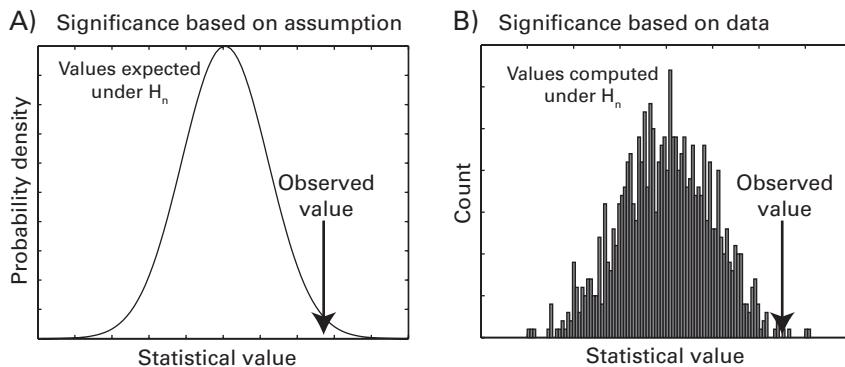


Figure 33.1

Illustration of distributions of test statistics under the null hypothesis (H_n) for permutation testing (panel A) and nonparametric permutation testing (panel B). The observed value is the value of the test statistic computed from the data without any adulterations or condition relabeling.

under the null hypothesis, and the probability (*p*-value) of obtaining a statistic under the null hypothesis at least as large as the observed statistic is computed (figure 33.1A). Different statistical tests make different assumptions about the distribution of the statistical value under the null hypothesis, such as a Gaussian, chi-square, or *F* distribution.

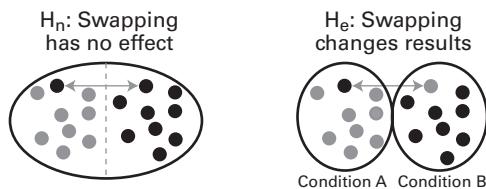
For nonparametric permutation testing no assumptions are made about the theoretical underlying distribution of test statistics under the null hypothesis. Instead, the distribution is created from the data you have (figure 33.1B) by observing what the test statistic would be if the null hypothesis were true. This is done, for example, by iteratively shuffling the condition labels over trials (for within-subject analyses) or over subjects (for group-level analyses) and recomputing the test statistic. (Procedures for creating a null hypothesis distribution are discussed in the next section.) In situations in which the data and the parameters that describe those data are normally distributed, nonparametric permutation testing and parametric statistics are likely to provide very similar results. But if the data are nonnormally distributed, nonparametric permutation tests are more appropriate.

Note that permutation testing is different from bootstrapping. With permutation testing the goal is to determine the probability that the observed test statistic could have been obtained if the null hypothesis were true (in other words, whether a finding is statistically significant). With bootstrapping the goal is to define confidence intervals on data characteristics (e.g., mean or variance) or parameter estimates, based on subsampling the data. Bootstrapping is not further discussed here.

33.2 Creating a Null-Hypothesis Distribution

To create a distribution of test statistic values under the null hypothesis, first consider how to create one null-hypothesis test statistic. Imagine an EEG dataset with two conditions, condition A and condition B. You hypothesize that a measure of EEG activity (for example, parietal alpha power), will be greater in condition A compared to condition B. You evaluate this hypothesis by performing a *t*-test of alpha power between the two conditions. The null hypothesis is that there is no difference in parietal alpha power between the two conditions. This null hypothesis can be interpreted to mean that if trials were randomly labeled as condition A or condition B, the test statistic would be as large as the test statistic before the random relabeling. In other words, under this null hypothesis, a trial from condition A could be labeled as condition B, and vice-versa for a trial from condition B (figure 33.2A). After condition labels from many trials have been swapped, a *t*-test across “conditions” can be computed. Because the condition labels have been randomly swapped across trials, you would expect the test statistic value to be zero. If the test statistic is not zero, you would

A) Null and effect hypotheses: discrete tests



B) Null and effect hypotheses: continuous tests

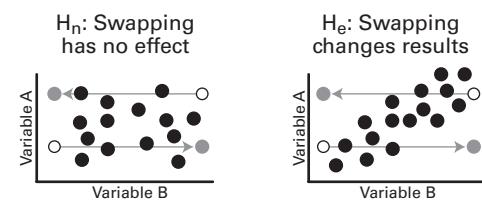


Figure 33.2

Illustration of the null and effect hypotheses for discrete tests (typically the case for comparing conditions; panel A) and for continuous tests (typically the case for correlating two continuous variables; panel B). In both of these cases condition labels are swapped between trials. If there is no condition difference, swapping labels has no effect on the test statistic. The effect hypothesis is often called the alternative hypothesis, but generally the hypothesis is that there is an effect, not that there is an alternative.

attribute this to sampling error or to the presence of outliers that happened to fall into one condition.

The preceding paragraph describes situations with discrete conditions, but the same principle applies to testing a relationship between two continuous variables (for example, when testing the statistical significance of a correlation coefficient). In this case, creating a test statistic value under the null hypothesis involves swapping data points in terms of their value on variable B. Because the independent variable has been randomly shuffled, a correlation coefficient of zero would be expected, and any nonzero correlation coefficient would be attributed to sampling error or outliers (figure 33.2B).

Note that in both of these situations (for discrete and continuous variables), the data themselves are not altered; instead, the mapping between the independent and dependent variables is shuffled.

The procedures described above create one null-hypothesis test statistic value. When those procedures are repeated hundreds or thousands of times, the null hypothesis values over all iterations create a distribution of test statistic values observed under the null hypothesis.

Statistical evaluation works by comparing the observed test statistic value (that is, the test statistic associated with nonshuffled data) against a distribution of null-hypothesis test statistic values. If the observed test statistic is well within the boundaries of the distribution of null-hypothesis test statistic values, the null hypothesis cannot be rejected. This means that the observed mapping between condition labels and trials could have been randomly generated; in other words, the effect is not statistically significant. In contrast, if the observed test statistic is “far enough” away from the null-hypothesis distribution (determining significance is discussed in section 33.4), the null hypothesis can be rejected, and the effect is unlikely to have occurred due to random condition labeling; in other words, the effect is statistically significant.

It should be clear now how permutation testing differs from parametric statistical testing: there are no assumptions required regarding the distribution of the test statistic under the null hypothesis because the null-hypothesis test statistic values are generated from the observed data rather than from a theoretical distribution. This is particularly useful if there are outliers, data points with nonrepresentative values, or unusual distributions, because all data points are shuffled into all conditions during the permutations.

33.3 How Many Iterations Are Necessary for the Null-Hypothesis Distribution?

Obviously, you do not want to compare the observed test statistic to only one null-hypothesis test statistic based on one reshuffling of the condition labels; because of sampling

error, small N , or outliers, one reshuffling may produce an unusually large or an unusually small test statistic by chance.

In theory, for N trials and two conditions, the multinomial coefficient states that there are $N!/(N_A!N_B!)$ possible permutations [N_A and N_B are the number of trials in conditions A and B, and N is the total number of trials; the ! here indicates the factorial, or $N(N - 1)(N - 2) \dots (1)$]. Unless you have an extremely small N , testing every possible permutation is impractical: with 40 trials in condition A and 60 trials in condition B, there are over 10^{28} possible ways to permute the condition labels.

In practice, condition- or variable-label swapping is done randomly for several hundred or a few thousand iterations. Performing more iterations is advantageous because the estimates of the null hypothesis distribution will be more robust, and therefore the significance of the result will be more reliable. However, more iterations also means longer computation times. There is no magic number of iterations or mathematical formula to indicate how many iterations are necessary or sufficient. For most applications, 1000 iterations are sufficient to build a high-signal-to-noise-ratio distribution. However, this may be time-consuming; if the permutation testing must be done at each trial, time point, and frequency (e.g., for some types of cross-frequency coupling or weighted phase-based analyses), 200–500 iterations may be sufficient. In general, if you have noisy data, few trials, or asymmetries in trial counts between conditions, it is better to have more iterations. In fact, it is always better to have too many iterations at the expense of longer analysis times than to have too few iterations at the expense of uncertainty in the significance of the result. The issue of how many iterations are necessary is further discussed in the text concerning figure 33.5.

33.4 Determining Statistical Significance

Once you have obtained a distribution of statistical test values expected under the null hypothesis, the next step is to compute a p -value associated with the observed test statistic. There are two methods to compute the p -value. First, you can count the number of null-hypothesis statistical test values that are more extreme than the observed statistical test value. More extreme means further to the right when testing the positive tail of the distribution, and it means further to the left when testing the negative tail of the distribution. That count divided by the total number of tests is the p -value. For example, in figure 33.1B, there are four null-hypothesis statistical test values that are greater than the observed value, giving a p -value of $4/1000 = 0.004$. With this approach it is possible to have a p -value of exactly zero if there are no null hypothesis test statistical values more extreme than the

observed test statistic. This is referred to as p_N , because the p -value is based on the number of suprathreshold tests.

A second method of obtaining a p -value is to compare the observed test statistic to the statistical properties of the null-hypothesis distribution. This is done by converting the observed test statistic value to standard deviation units of the null-hypothesis distribution, which can then be converted to a p -value. The standard deviation unit is created by subtracting the mean of the null-hypothesis distribution from the observed statistic value and then dividing by the standard deviation of the null-hypothesis distribution. Probably you already recognize this as a standard Z value.

$$Z = \frac{v_e - \bar{V}_n}{std(V_n)} \quad (33.1)$$

where v_e is the observed-effect test statistic, V_n is a vector of null-hypothesis test statistics, the horizontal bar on top of the V_n indicates the mean, and std indicates the standard deviation. This Z value can then be converted to a p -value by evaluating its position on a Gaussian probability density (Matlab function `normcdf`, which is included in the statistics toolbox). The p -value cannot be exactly zero, and it is not based on the number of null-hypothesis iterations performed. This is referred to as p_Z , because the p -value is based on a Z distribution of null-hypothesis test statistics. p_N and p_Z will generally provide similar results, particularly if the null-hypothesis distribution is approximately Gaussian (see figure 33.3). Applying equation 33.1 is valid if the distribution of null-hypothesis test statistic values is approximately Gaussian. If the distribution is not approximately Gaussian, you may be able to interpret Z -values and p_Z from one tail of the distribution.

Figure 33.3 illustrates a null-hypothesis distribution for one test, but with EEG data there are many tests to perform over time points, frequency bands, and electrodes. Thus, each time-frequency point can have its own distribution of null-hypothesis values (figure 33.4), and, thus, at each pixel, a p -value or Z -value can be computed. This is advantageous because data distribution characteristics at each time-frequency-electrode point are incorporated into the null-hypothesis distributions and therefore are incorporated into the statistical thresholding. Some methods for multiple-comparisons corrections do not require the full distribution to be stored at each pixel.

It is important to realize that, because the data are randomly reshuffled, the p -value will change each time you recompute the null-hypothesis distribution. If you have a sufficient number of iterations, the fluctuations in the p -values should be small and inconsequential (e.g., $p = 0.2333$ vs. $p = 0.2317$) and should not affect the interpretation of the results. However, if there are too few iterations, and the effect is perithreshold, it is possible to

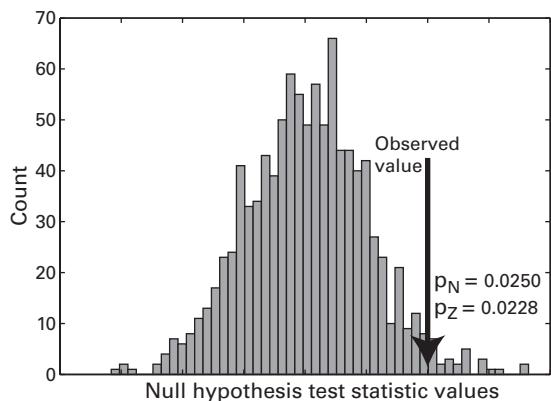
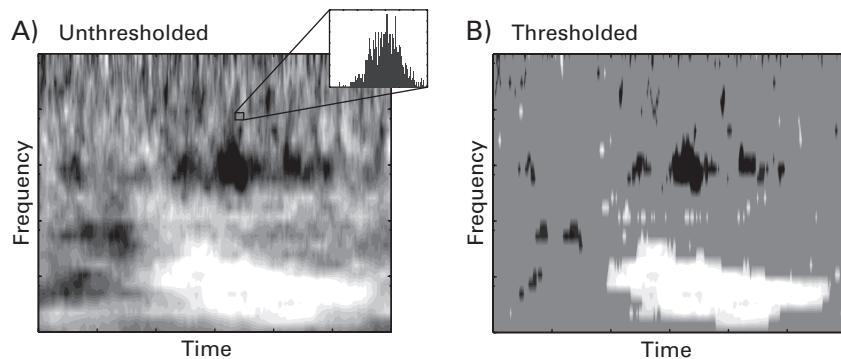
**Figure 33.3**

Illustration of two methods for computing the p -value from a null-hypothesis distribution. p_N is computed by counting the number of null-hypothesis values above the observed value and dividing by the number of null-hypothesis iterations; p_z is computed by converting the observed value to standard deviation units and evaluating the probability of that standard deviation under a Gaussian distribution. p_N does not rely on assumptions about the shape of the null-hypothesis distribution, whereas p_z is valid if the distribution is approximately Gaussian. In this example the two methods provide similar values.

**Figure 33.4**

Panel A shows a time-frequency map from one electrode, illustrating how each pixel has a corresponding distribution of null-hypothesis test statistic values. Panel B shows this map with the values at subthreshold pixels set to zero. Although the threshold at each pixel is set to the same p -value, the precise test statistic value (e.g., t -test or correlation coefficient) will change slightly over pixels because each pixel has its own null-hypothesis distribution.

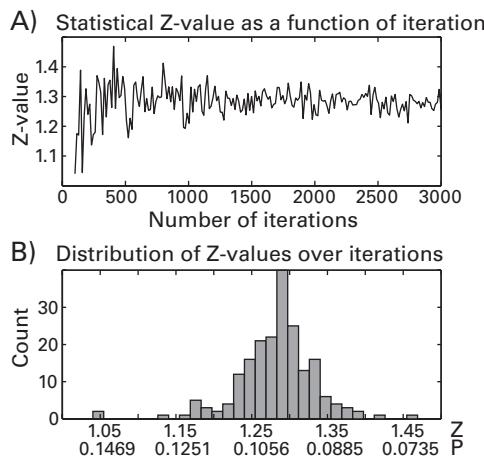


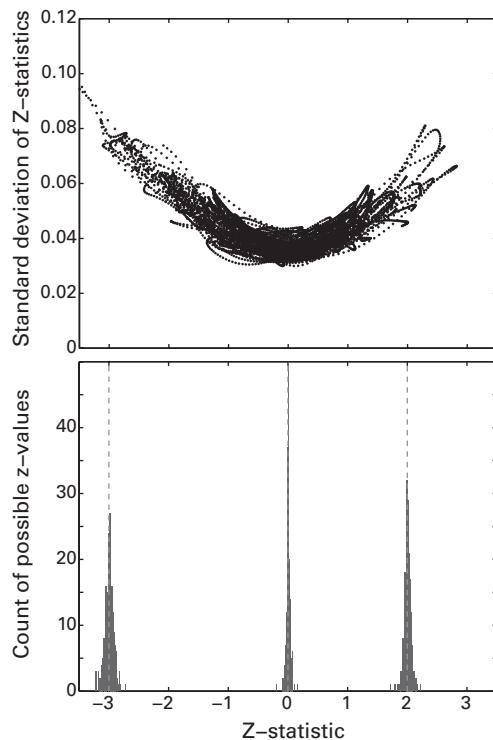
Figure 33.5

Panel A shows a test statistic value (*y*-axis, these are standard *Z* values from equation 33.1) as a function of the number of iterations used to generate the null hypothesis distribution (*x*-axis). Panel B shows a histogram of these test statistic values. The top row of *x*-axis labels shows *Z*-values, and the bottom row of *x*-axis labels shows the corresponding *p*-values.

get $p = 0.052$ and $p = 0.045$ on different calculations of the null-hypothesis distribution. Having more iterations should help stabilize the *p*-value (in theory, as the number of iterations tends toward infinity and with a lot of data, the histogram distribution begins to look like a probability density, and a *p*-value can be more accurately obtained). Figure 33.5 shows an example of how the *p*-value will produce slightly different results during permutation testing, even when several thousand iterations are used. Figure 33.5A shows that the statistical *Z* value (equation 33.1) fluctuates around a mean of 1.28 with repeated permutation tests. Although the variability decreases at around 1500 iterations, there continues to be some variability up to 3000 iterations. Figure 33.5B shows a histogram of the *Z*-values and associated *p*-values over all iterations.

How much variability you can expect in your data depends in part on how clean the data are and in part on how significant the result is—results toward the tails of the null-hypothesis distribution are likely to have relatively more variability, whereas results toward the center of the distribution are likely to have relatively less variability. This can be seen in figure 33.6. This figure was generated from real data.

A real-data example of this phenomenon is shown in figure 34.4, for which a correlation between single-trial power and reaction time is supra- or subthreshold in different permutation tests. If you are concerned about this variability, for example if your effect fluctuates

**Figure 33.6**

The variability of the permutation test depends in part on the statistical significance of the result. Permutation testing was applied to time-frequency data from one electrode, using 1000 permutations at each of 200 iterations (that is, permutation testing was performed on the same data 200 times). Plotted in the top panel is the standard deviation (y-axis) of permutation tests across iterations as a function of the average Z-statistic value (x-axis) over all 200 iterations. Each dot is a time-frequency point. This plot shows that results that are toward the tails of the null hypothesis distribution have more variability in the resulting p -value over iterations. The bottom panel illustrates three examples of distributions of Z-values obtained from the 200 iterations (each distribution corresponds to one time-frequency pixel). You can see, for example, that the distribution of Z-statistic values around -3 is wider than that around 0 . These three distributions were selected on the basis of their average Z-value, not on the widths of their distributions.

between *p*-values of 0.045 and 0.053, you can run the permutation test 20 times (using >1000 iterations each time) and average the *p*-values together. This “meta-permutation test” will provide a more stable estimate of the true *p*-value. For example, in figure 33.5, the test statistic converges on a mean of $Z = 1.28$ even though individual tests ranged from 1.05 to 1.45.

33.5 Multiple Comparisons and Their Corrections

One commonly used approach in statistics to correct for multiple comparisons is to divide the *p*-value by the number of tests that will be evaluated. For example, if you have three statistical tests, the *p*-value significance threshold per test is 0.016667. This is called Bonferroni correction. Bonferroni correction is appropriate in some situations for EEG analyses, namely hypothesis-driven analyses in which you test a small number of a priori defined regions of time-frequency-electrode space. For example, if your hypothesis predicts differences in prestimulus alpha-band activity between two conditions, and you will perform statistical tests only in one time-frequency window (e.g., -500 to -100 ms prestimulus and 8–12 Hz), the Bonferroni correction would be appropriate for testing the effect at three different electrodes. Bonferroni correction can also be appropriate if there are not too many tests and if you expect robust effects. For example, if you are testing effects at each of 32 electrodes, a Bonferroni correction of $0.05/32$ is 0.0015625; robust effects are likely to remain significant at this threshold.

However, Bonferroni correction is inappropriate when you are performing many tests over time points, frequency bands, and electrodes. This would be the case if you are performing exploratory data-driven analyses or if you have general expectations concerning, for example, the frequency band of the effects but not the time point or electrode. Bonferroni correction is inappropriate in these situations for three reasons.

First, Bonferroni correction assumes that the tests are independent, which is not the case for many EEG results: neighboring time points, frequency bands, and electrodes are correlated (in other words, there is strong spatial-temporal-frequency autocorrelation). Furthermore, the autocorrelation changes as a function of time, space, and frequency.

Second, Bonferroni correction is inappropriate because the *p*-value will be reduced so severely that even true effects are unlikely to be identified as statistically significant. For example, an experiment with two conditions, 64 electrodes, 100 time points, and 30 frequency bands will produce 192,000 possible between-condition tests. But this is actually a huge underestimate of the number of possible tests. You might want to compare the condition difference across electrodes, time points, and frequencies (for example, does the condition difference emerge earlier in the alpha band over parietal areas compared to the

condition difference in the theta band over frontal electrodes?). Thus, the true number of possible tests in this case is extremely large and, for practical purposes, not countable. Even taking the “simple” case of 192,000 tests, a Bonferroni-corrected *p*-value of 0.05 would result in each test being evaluated at $p < 0.00000026$. Even true condition differences with fairly large effect sizes are unlikely to be deemed statistically significant at this threshold.

Third, Bonferroni correction is based solely on the number of tests rather than on the information available in those tests. This may be appropriate in situations with discrete variables such as conditions, electrodes, and groups of subjects, but it is awkward for time-frequency analyses because the number of tests (that is, the number of time-frequency points) can be arbitrarily large or small. Consider that if you extract 20 frequencies between 2 Hz and 30 Hz, the Bonferroni correction would be half as stringent as if you were to extract 40 frequencies between 2 Hz and 30 Hz. It is unlikely that the additional 20 frequencies would provide any unique information, and yet the statistical significance threshold suffers. The same is true for temporal downsampling of the results. An appropriate correction for multiple comparisons should be based on the amount of information in the results, not purely on the number of tests that could be performed.

Fortunately, the permutation-testing framework is amenable to incorporating corrections for multiple comparisons that (1) corrects for the information present in the results rather than the number of tests while (2) providing corrected *p*-value thresholds that remain sensitive enough to detect effects in correlated multidimensional data.

There are two methods for correcting for multiple comparisons within the framework of nonparametric permutation testing. One method is based on correcting for multiple comparisons by considering the pixel to be the unit for determining a threshold, and the other method is based on correcting for multiple comparisons by considering the cluster to be the unit for determining a threshold (a cluster is a group of contiguous suprathreshold pixels).

In general, solutions to the multiple-comparisons problem work by considering null-hypothesis distributions at the map level (e.g., a time-frequency map, or a frequency-electrode map) rather than at the pixel level. That is, rather than having a null-hypothesis distribution at each pixel as shown in figure 33.4, there is a null-hypothesis distribution over null-hypothesis maps. And whereas the elements in the null-hypothesis distribution at each pixel contain the statistical values from only that time-frequency pixel, with corrections for multiple comparisons the elements in the distribution are not related to any single time-frequency point or electrode but instead reflect the information from the entire time-frequency-electrode space at which permutations are performed.

Permutation testing is not the only framework for correcting for multiple comparisons. Other approaches include the false discovery rate correction (discussed below in section

33.8), Gaussian random field theory (not discussed here but implemented in SPM8; Kilner, Kiebel, and Friston 2005), and using a stringent uncorrected statistical threshold.

33.6 Correction for Multiple Comparisons Using Pixel-Based Statistics

In general, correcting for multiple comparisons using pixel-based statistics involves creating a distribution that contains the pixel from each iteration of permutation testing with the most extreme statistical value. Correcting for multiple comparisons using pixel-based statistics tends to be (but is not necessarily always) more stringent than correcting for multiple comparisons using cluster-based statistics.

To perform pixel-based multiple-comparisons correction, perform permutation testing using the procedures previously outlined for generating test statistic values under the null hypothesis. However, at each iteration during permutation testing, find and store in a matrix the value of the one or two pixels with the most extreme null hypothesis test statistic values. Whether you store one or two values depends on whether you are using a one-tailed or two-tailed test. For a two-tailed test, find the largest positive value and the largest negative value.

After all iterations have completed you will have two distributions of the largest positive and largest negative pixel values. You can then define the statistical threshold to be the value corresponding to the 2.5th percentile of the smallest values and the value corresponding to the 97.5th percentile of the largest values. These are the lower and upper bounds of the threshold at $p = 0.05$, correcting for multiple comparisons over all time-frequency points (figure 33.7). If you are testing only positive or only negative effects, you will need to store only one extreme test statistic from each iteration and can thus take the fifth or the ninety-fifth percentile of the distribution (for negative and positive tails, respectively). Any pixel that has a value greater than the upper threshold or smaller than the lower threshold can be considered statistically significant, correcting for multiple comparisons using pixel-based thresholding.

Notice that you do not save a null-hypothesis distribution at each pixel, as was illustrated in figure 33.4. Instead, you save a summary of the most extreme null hypothesis statistical values across all pixels at each iteration of permutation testing. Thus, the multiple-comparisons problem is addressed by performing map-level thresholding instead of pixel-level thresholding.

Because the distribution is based on map-level and not pixel-level information, this multiple-comparisons correction is based on the amount of information in the results, regardless of how many tests were performed. That is, if you have the frequencies 5.1 Hz and 5.2 Hz in your data, the corrected threshold is unlikely to change because those two frequency

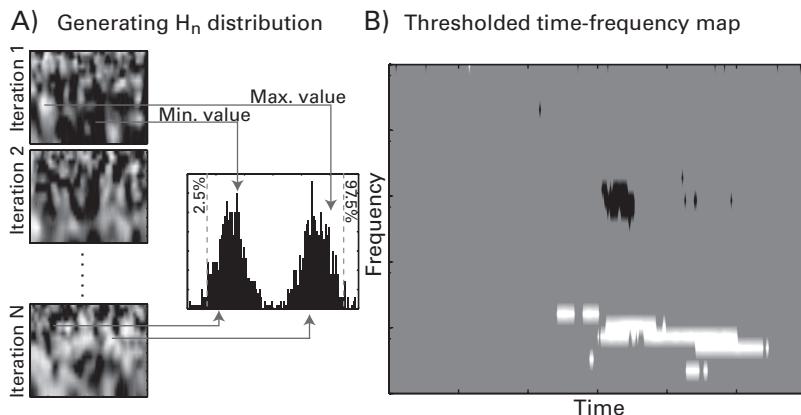


Figure 33.7

Illustration of multiple comparisons correction using pixel-based statistics. Panel A illustrates that at each iteration in the permutation testing, the maximum and minimum pixel values are stored, generating two distributions of extreme values (both are shown in the same plot for visual convenience). The values at 2.5% and 97.5% of these distributions (vertical dashed gray lines) are taken as the threshold, which is then applied to the observed statistic values, shown in Panel B.

bands provide little unique information (similar situation for the number of time points and temporally downsampling the results). This is good because it means that you do not have to be concerned about the effect of using too many or too few frequency bands or the extent to which you temporally downsample after analyses on the statistical thresholding.

With this approach, it is possible that a single pixel can be statistically significant without any contiguously significant neighbors. Whether a result from one significant pixel can be interpreted depends on your experiment design and on the size of your time-frequency pixels. For example, if you have very large time-frequency pixels comprising several hundreds of milliseconds, one pixel might be appropriate to interpret.

33.7 Corrections for Multiple Comparisons Using Cluster-Based Statistics

A cluster is a group of contiguously significant points in time-frequency-electrode space. Clusters can be seen after a threshold has been applied and any pixel that has a value below that threshold has been set to zero. The idea of cluster-based correction is that, because of autocorrelation in the data, a finding is significant if it is “big enough,” that is, if enough neighboring pixels also have suprathreshold values. Individual pixels that are significant are therefore considered false alarms. How big is “big enough”? This depends on the resolution

of the results (that is, how many frequencies were extracted and whether the results were temporally downsampled after time-frequency decomposition had been performed). For example, if each time point reflects activity from 1 ms, a single significant time-frequency point may be considered a false alarm. However, if there was considerable temporal and frequency downsampling and averaging, such that a single pixel represents activity from several hertz and several hundreds of milliseconds, a single significant time-frequency pixel may be considered a valid result.

One simple way to perform cluster correction is to remove clusters that have fewer than some predefined number of time and frequency points. For example, you might remove any clusters less than 200 ms long and 3 Hz wide. The main limitation of this method is that there is a subjective and non-data-driven quality to it. Although this method may in some cases result in a more stringent threshold than data-driven approaches, it should be a less preferred method. The method described below, in contrast, is data-driven and involves user intervention only in selecting *p*-value thresholds.

Cluster-based correction works slightly differently than pixel-based correction, but it is based on the same principle of obtaining a maximum statistic from the entire map at each iteration, thus controlling for false alarms at the map level rather than at the single-pixel level. The difference between cluster-based correction and pixel-based correction is that with cluster-based correction, statistical values are based on clusters rather than on pixels.

As with pixel-based correction, you start by performing permutation testing as described previously in this chapter. However, at each iteration of null-hypothesis test statistic generation during permutation testing, a threshold is applied to the time-frequency map at an uncorrected level such as $p < 0.05$ (this is referred to as the precluster threshold). This null-hypothesis iteration map can be thresholded in one of two ways. First, it can be thresholded using parametric statistics. For example, you can use the *p*-value that results from a parametric *t*-test or correlation coefficient. This might be an appropriate thresholding strategy if the data are approximately normally distributed, as is the case for baseline-corrected power in group-level analyses. Other data might have clear departures from a normal distribution, in which case a *p*-value from a parametric test might be inappropriate. Thus, you can use the second method for thresholding the null-hypothesis maps of statistical values, which is to loop through iterations twice: once to build null-hypothesis distributions at each pixel, as in figure 33.4, and then a second time to threshold them using the nonparametric pixel-based significance thresholding described in section 33.4.

Thresholding the null-hypothesis map at each iteration will result in some individual pixels (which can be thought of as clusters of size one) and some clusters. Because these statistical maps were generated under the null hypothesis, any suprathreshold clusters can be

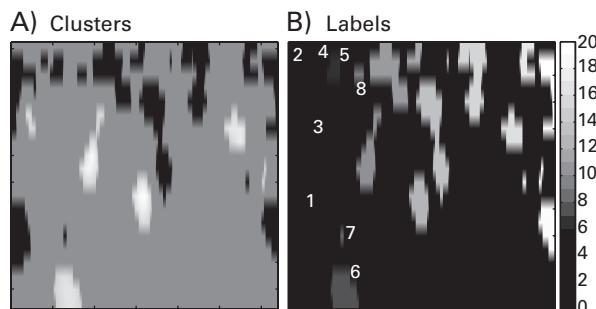


Figure 33.8

Illustration of cluster identification procedure in Matlab. From the thresholded map shown in panel A, one of several Matlab functions can be applied that find and label each contiguous cluster with a number, as shown in panel B. In this example there are 21 clusters, the first 8 of which are labeled inside the plot in panel B.

considered false alarms. The next step is to collect data from the biggest cluster in the map. Before this is discussed further, a short digression on identifying clusters in Matlab will be useful.

There are several Matlab functions that will find and label contiguous clusters in matrices. The two most useful functions are `bwlabeled` and `bwconncomp`, which are both included in the image-processing toolbox. These functions return the number, size, and locations of contiguous nonzero values (figure 33.8). The online Matlab code for this chapter provides some instruction on how to use these functions and what to do with their outputs.

After clusters have been identified using `bwlabeled` or `bwconncomp` or another appropriate function, the next step is to identify the cluster with the most extreme value on one property and store the value of that property from that cluster in a matrix. There are several properties that can be used. The two most commonly used properties are the count of suprathreshold pixels and the sum of the test statistic values in each cluster (e.g., the sum of the t -values over all within-cluster pixels; if the effect can be negative, sum the absolute value of the test statistic values). The advantage of using the sum of test statistic values within the cluster is that the threshold is sensitive both to small clusters with large test statistic values and to large clusters with modest test statistic values. There are other cluster properties that can be extracted, but these two are the most commonly used and have been argued to be the most sensitive based on one dataset (Maris and Oostenveld 2007).

After all iterations have been completed, you have a distribution of the largest suprathreshold clusters that can be expected under the null hypothesis. The final step of cluster-based correction is to threshold the map of observed statistical values (that is, the unadulterated

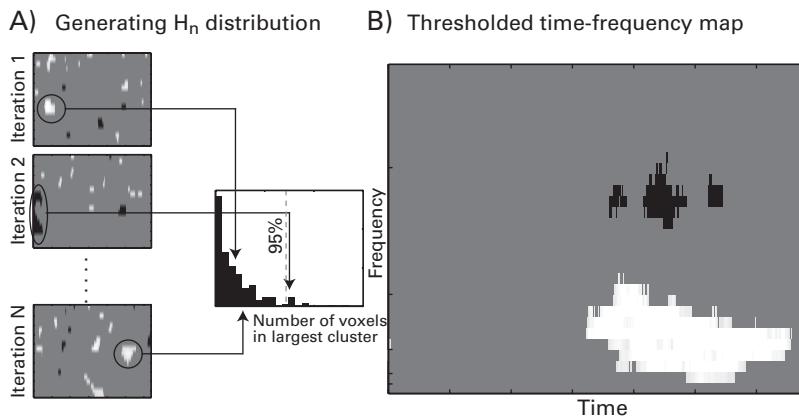


Figure 33.9

Illustration of multiple-comparisons correction using cluster-based statistics. Panel A illustrates that at each iteration in the permutation testing, the permuted time-frequency map of statistical values is thresholded, and the number of pixels in the largest cluster is stored. (Other features of the cluster can also be stored, such as the sum of test statistic values from the cluster with the maximum test statistic value sum.) This generates a histogram of cluster sizes expected under the null hypothesis. The value at 95% of this distribution (vertical dashed gray lines) is taken as the cluster threshold, which is then used to remove clusters from the observed test statistic values that contain fewer than the threshold number of pixels, shown in panel B.

data) using an uncorrected p -value, identify clusters in that thresholded map, and remove any clusters that are less than the ninety-fifth percentile of the distribution of largest clusters expected under the null hypothesis (figure 33.9).

As with pixel-based multiple-comparisons corrections, the cluster-based method described here addresses the multiple-comparisons problem by performing map-level thresholding instead of pixel-level thresholding. Although having more time-frequency points will produce larger clusters (e.g., extracting 40 frequency steps instead of 20 frequency steps for the same frequency range), the increase in cluster size is the same for the null-hypothesis distribution and for the observed statistical map. Thus, similar to the pixel-based multiple-comparisons correction, cluster correction is based on the information available in the results rather than on the number of tests performed.

One feature of cluster-based correction to be aware of is that the precluster threshold affects the cluster correction threshold. This should make sense: at an extreme threshold of $p < 0.001$, very few clusters will survive, and the surviving clusters will be very small; at the other extreme, a very lenient threshold of $p < 0.20$ will result in many large clusters. Make

sure you use a reasonable precluster threshold and apply the same (or more stringent, but not more lenient) threshold to the data as you applied to obtain the clusters during null-hypothesis generation permutations.

The main disadvantage of cluster-based correction is that it is maximally sensitive to large clusters, whereas true effects that are more localized in time-frequency space may fail to be identified as statistically significant. This is further complicated by the fact that spatial-temporal-frequency smoothing is heterogeneous—for example, there is more smoothing (and, thus, larger clusters) at lower frequencies compared to higher frequencies. This heterogeneity in autocorrelation, which affects cluster sizes, is not taken into account during cluster correction. However, adjusting the balance between temporal and frequency precision as a function of frequency during time-frequency decomposition will help minimize the heterogeneity of autocorrelation over frequencies.

33.8 False Discovery Rate for Multiple-Comparisons Correction

False discovery rate (FDR) is an alternative method for correcting for multiple comparisons. It is commonly used in fMRI research (Genovese, Lazar, and Nichols 2002) and is also included in several M/EEG analysis packages. FDR works by controlling for the probability of type I errors within a distribution of p -values (which can be obtained either via nonparametric permutation testing or via parametric statistics). You can find a Matlab script to implement the FDR correction in the online Matlab code, which was downloaded from Dr. Thomas Nichols's website. One limitation of FDR with respect to time-frequency results is that the critical p -value is based not only on the distribution of p -values but also on the number of tests performed. More precisely, the FDR-defined critical p -value becomes exponentially smaller with linear increases in the number of p -values above 0.05, whereas the critical p -value increases only very slightly with linear increases in the number of p -values below 0.05 (figure 33.10) (this phenomenon does not depend on the range of the p -values, only their magnitude relative to the desired threshold). This means that the FDR-defined critical p -value will become more stringent if you include both 5.1 Hz and 5.2 Hz, even though the activities at those frequencies may contain little unique information.

33.9 What Should Be Permuted?

So far in this chapter permutation testing has been described as shuffling the mapping between condition labels and trials. More generally, permutation testing involves shuffling the mapping between the independent and dependent variables. Thus, the feature of the

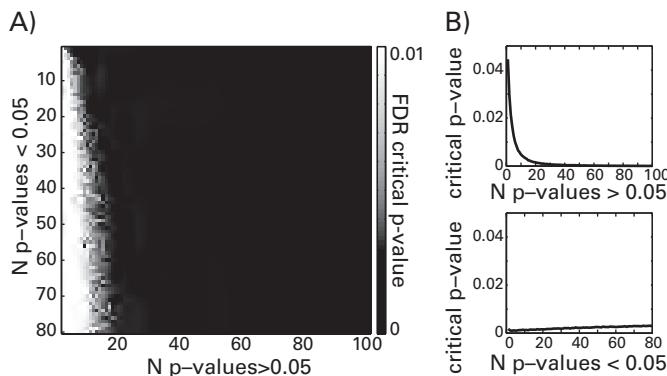


Figure 33.10

Critical p -values obtained from the false discovery rate (FDR). In this simulation populations of p -values were created with a variable number of “significant” ($p < 0.05$) and “nonsignificant” ($p > 0.05$) values (see x - and y -axes) (panel A). The critical p -value decreased exponentially with linear increases in the number of nonsignificant p -values (panel B, upper plot), whereas the critical p -value increased only very slightly with linear increases in the number of significant p -values (panel B, lower plot). Results were averaged over 20 simulations to smooth the plots.

data that should be shuffled depends on the focus of the analysis and on the hypothesis. When two conditions are compared, condition labels could be shuffled; when correlations between time-frequency power and reaction time over trials are computed, the mapping of reaction time to trials should be shuffled; when connectivity between two electrodes is computed, the ordering of time segments within each trial could be shuffled.

If you are unsure which dimension should be shuffled, think about your hypothesis. If the hypothesis concerns effects across trials, then shuffle trials (and not time). If your hypothesis concerns effects across subjects, shuffle subjects (and not trials). You should shuffle only the property of the data that you want to investigate while leaving the rest of the data as untouched as possible (Theiler et al. 1992). Some methods of permutation testing necessarily involve altering properties of the data, such as the cut-and-shift procedure used for phase-amplitude coupling. If data alterations are necessary, try to keep them to a minimum.

In some cases there is more than one option for what can be shuffled. For example, many connectivity measures can be computed over time or over trials; whether you shuffle the trial sequence or the temporal structure within trials has implications for what kinds of dynamics can be statistically identified. If the connectivity is phase-locked to the time = 0 event, then that pattern of connectivity will be present on each trial, and thus, shuffling trials will produce a nonsignificant result even if the connectivity is very strong. If you are

testing for directional connectivity effects, you can also permute the direction of time (Haufe et al. 2013).

33.10 Nonparametric Permutation Testing beyond Simple Bivariate Cases

You may have noticed that all of the explanations of nonparametric permutation testing in this chapter involved simple bivariate cases such as comparing two conditions or correlating two variables. This may seem limiting, particularly to those with a psychology background who are used to more complex statistical designs such as factorial ANOVAs. Although there are methods available to adapt nonparametric permutation-based statistics to factorial designs such as ANOVAs (Anderson and Ter Braak 2003; Zhou and Wong 2011), these are rarely used in cognitive electrophysiology and are not implemented in many M/EEG analysis packages. If you have a factorial design, it is best to take a hypothesis-testing approach by extracting data from a priori specified time-frequency-electrode windows, exporting the data to a text file, and performing the appropriate analyses in dedicated statistical software packages such as SPSS, SAS, or R. Chapter 35 discusses strategies for hypothesis testing of time-frequency results. If you have a factorial design, it is not a good idea to test many simple effects because of increased risk of type I errors.

33.11 Describing This Analysis in Your Methods Section

First, you should justify your decision to use nonparametric permutation testing over parametric statistics. Often, permutation testing is used because the analyses were exploratory rather than hypothesis driven or because the data violated assumptions of parametric statistics. Although permutation testing is fairly common in cognitive electrophysiology, there are many options involved in its implementation. Be clear about what variables and dimensions were shuffled, how many iterations were performed, how the p -values were created, and how correction for multiple comparisons was done (if at all). For multiple-comparisons corrections based on clusters, report the p -value threshold used to obtain null-hypothesis clusters.

34 Within-Subject Statistical Analyses

This chapter illustrates several approaches for performing statistical analyses over trials within a single subject. There are two different motivations for performing within-subject statistics. First, within-subject statistics can be used to evaluate the statistical significance of single-subject results (as discussed in the previous chapter, this is useful in situations with few subjects or to test the robustness of the effect for each subject). Second, within-subject statistics can be used to facilitate group-level analyses. In this latter scenario, permutation testing is not used to evaluate statistical significance per se, but, rather, is used as a means to “normalize” the data by transforming them into values that are more amenable to cross-subjects comparisons and parametric statistics. This may be useful for, among other analyses, phase-amplitude coupling, wITPC, and mutual information.

This chapter does not cover all of the possible single-subject analyses but, rather, the ones that are most likely to be relevant for typical cognitive electrophysiology studies. Most of the analyses of time-frequency power presented in this chapter rely on nonparametric permutation testing, as described in the previous chapter. Therefore, this chapter should be read after you understand the procedures for nonparametric permutation testing presented in chapter 33.

In all analyses presented here statistics are evaluated at each of 436 time points (data points were removed from the original 640 time points to cut out edge artifacts). However, as discussed in section 27.5, the results after time-frequency decomposition can often be downsampled with little loss of information. Temporal downsampling will decrease computation time and disk storage requirements but will not affect multiple-comparisons corrections that are based on nonparametric permutation testing.

34.1 Changes in Task-Related Power Compared to Baseline

Before testing for possible differences in activity between conditions or testing for correlations between brain and behavior, you may want to test whether a time-frequency power feature is statistically significant on its own.

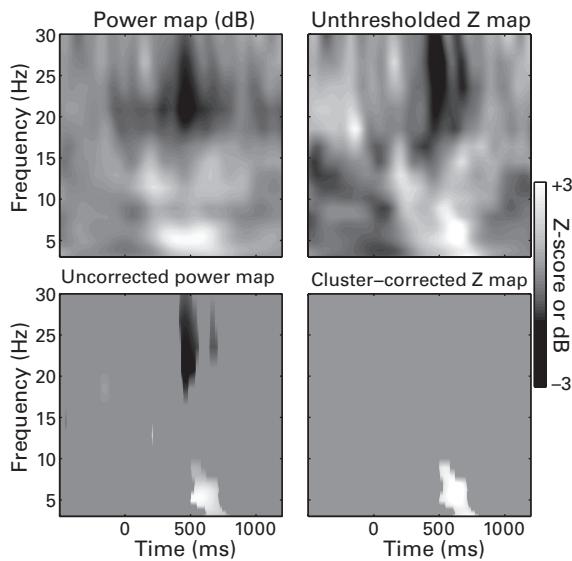


Figure 34.1

Statistical significance testing for changes in time-frequency power in one condition relative to a pre-stimulus baseline period (-500 to -100 ms), based on nonparametric permutation testing and correction for multiple comparisons at the cluster level. The p -values applied were $p = 0.01$ for the uncorrected map and $p = 0.05$ for the cluster correction. In the plots on the bottom row, the value of nonsignificant pixels was set to 0 (light gray).

Testing power values against a null hypothesis of zero power is inappropriate because power values cannot be negative, and their distance away from zero depends in part on the frequency band (due to power-law scaling). Thus, an appropriate null hypothesis is that there is no change in power relative to power during some other time period. Typically, that null hypothesis time period is the pretrial baseline.

There are several ways to implement a statistical test of a change in power relative to the baseline time period. The method presented here and shown in figure 34.1 is to compute the decibel change from baseline and, at each iteration during permutation testing, temporally shift the time series data by a random offset. This is similar to how the power time series was shifted in phase-amplitude coupling analyses and is illustrated in figure 34.2A. Statistical thresholding and corrections for multiple comparisons were done using the procedures described in chapter 33.

There are several other ways to test for changes in power with respect to a prestimulus baseline period. For example, rather than randomly shifting the power-time series, you

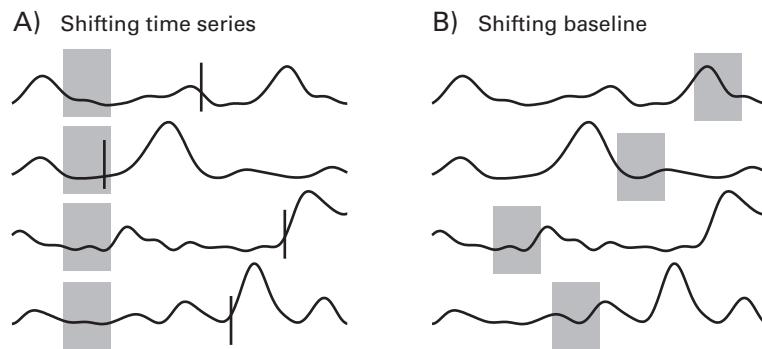


Figure 34.2

Illustration of two different methods of permutation testing for changes in time-frequency power relative to the pretrial baseline period. Each time series corresponds to a trial, the gray boxes indicate the time period used as the baseline, and the vertical black lines correspond to locations in which the time series was cut. In these cases the time series after the vertical line was moved to before the time series before the vertical line. These two methods will generally produce similar results; the main difference is that the temporal structure of the signal is not changed in the shifting-baseline approach.

could shift the time period of the baseline from the entire trial period. Thus, different trials would have different randomly selected baseline periods, some of which would occur during the trial itself. This procedure is illustrated in figure 34.2B. This approach is well suited for transient changes in power but has decreased sensitivity for detecting tonic changes in power: if the epoch is 2 s long, and power is increased relative to baseline for 1.5 s, the extended increase might not be statistically significant because the random baseline will often include periods of increased power. Another approach, used in eeglab, is to shuffle the prestimulus baseline period time points at each trial and use the largest and smallest N values as the cutoff threshold, where N is related to the number of permutations and to the p -value.

34.2 Discrete Condition Differences in Power

Testing for differences in power between two conditions involves comparing power values across trials between the two conditions at each time-frequency point (figure 33.2A). It can be achieved by applying permutation-based t -tests. There are no conditions in the online sample EEG dataset, so “conditions” were defined as the first half and the second half of trials. At each iteration during permutation testing, the condition labels were shuffled such that trials could be randomly labeled as being in the first half or in the second half of trials.

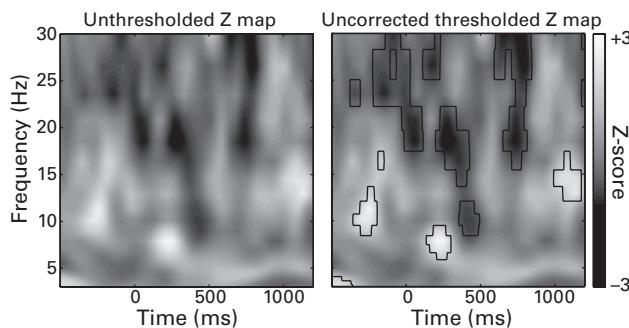


Figure 34.3

Time-frequency plots of power at electrode FCz showing differences between the first half and the second half of trials ($p < 0.05$ uncorrected; contour lines indicate regions of suprathreshold pixel clusters). No results survived corrections for multiple comparisons using pixel-based or cluster-based statistics.

In this analysis clusters were obtained by thresholding null-hypothesis maps assuming a parametric distribution of t -values (as discussed in section 33.7).

The uncorrected ($p < 0.05$) time-frequency plot from electrode FCz suggested that the first half of trials, compared to the second half of trials, was associated with increased pre- and poststimulus alpha power and decreased beta-band power (figure 34.3). However, none of these differences survived correction for multiple comparisons via either pixel-based or cluster-based thresholds (results not shown here because the plots are empty). This is not surprising, considering that these trials were drawn from the same condition and differ only according to the time in the experiment in which those trials occurred (the experiment lasted about an hour).

34.3 Continuous Relationship with Power: Single-Trial Correlations

In many experiments there are not only discrete conditions but also variables that can have a unique value on each trial. Reaction time is one example. Reaction time is a composite behavioral measure that includes stimulus processing, decision making, and response selection and execution. It is one of the most widely used behavioral measures in experimental psychology and cognitive science, although it is rarely integrated with brain data in cognitive electrophysiology studies. Examples of other trial-varying variables that might be relevant to link to EEG data include stimulus properties such as visual luminance or sound frequency, environmental events, or other subject responses such as pupil diameter or post-trial confidence ratings.

How should you relate a continuous trial-varying variable to the trial- and time-varying EEG data? One approach is to discretize the data. That is, in the case of reaction time, group trials into, for example, five bins based on quintiles of the reaction time distribution and test for changes in brain activity over those five discrete bins. Although this approach is often applied in cognitive neuroscience, it is a suboptimal procedure for several reasons. Discretizing continuous data reduces sensitivity and statistical power, can misrepresent bivariate relationships (particularly if those relationships are nonlinear), can introduce confounds or biases that were not previously present in the data, and can increase the likelihood of false positives. The risk of these negative effects increases when fewer bins are used, such as two or three discretizations. There are several statistics papers arguing why continuous variables should not be discretized (Fedorov, Mannino, and Zhang 2009; Naggara et al. 2011; Royston, Altman, and Sauerbrei 2006). There is rarely a justification for discretizing (in particular, dichotomizing) continuous variables, and it should be avoided whenever possible.

Fortunately, appropriate statistical procedures for linking two continuous variables are straightforward and amenable to both parametric and nonparametric statistical evaluation. In fact, the appropriate statistical procedure to relate two continuous variables is a correlation coefficient. For single-trial power data, Spearman correlations should be used because power data are nonnormally distributed (see section 27.1 for further justification of using Spearman over Pearson correlation coefficients for within-subject power data).

Figure 34.4 (plate 25) shows results of correlating reaction time with power over time-frequency points at electrode FCz. This analysis was done using nonparametric permutation testing. At each iteration during the permutation testing, the assignment of reaction time value to trial was randomized. Note that there is one reaction time value per trial, but that same reaction time value is correlated with power at each time-frequency point.

To threshold the permuted map to obtain clusters (that is, the precluster threshold), a nonparametric approach was taken that differed from the approach taken in the analysis in section 34.2. Rather than obtaining statistical significance p -values from the correlations assuming a parametric distribution of correlation coefficients, two loops were performed. In the first loop 1000 time-frequency correlation coefficient maps were generated under the null hypothesis by shuffling trial mappings between reaction time and EEG data. In the second loop each permuted map was converted to a statistical Z-map relative to the other 999 time-frequency permuted correlation maps, and then subthreshold pixels were set to zero and the remaining cluster properties were obtained. In this approach, no assumptions were made about data or parameter distributions.

In chapter 33, it was discussed that because permutation testing involves randomizing trial mappings, the results may be slightly different each time you rerun the statistical test,

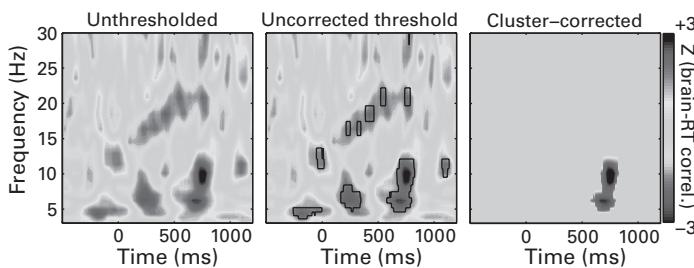


Figure 34.4 (plate 25)

Single-trial correlations (estimated via ordinary least squares) between reaction time and time-frequency power. The left panel shows the Z-values (standardized units from each correlation away from the distribution of null-hypothesis correlation coefficients); the middle panel shows the Z-values with black contours indicating regions that are significant using an uncorrected pixel-level threshold of $p < 0.01$; and the right panel shows the same result with cluster thresholding at $p < 0.05$. No pixels survived thresholding at a corrected pixel-based thresholding of $p < 0.05$. This result indicates that trials with increased power from 5–12 Hz and 500–800 ms tend to have longer reaction times. The middle and left panels show two different ways of illustrating statistically significant results, either by showing all results and outlining the significant regions or by setting nonsignificant pixels to 0 (advantages of different methods of showing results are discussed in chapter 36).

even though the data have not changed. For findings that are highly statistically significant or highly not statistically significant, this is unlikely to make a difference. However, figure 34.4 (plate 25) shows a situation in real data in which a finding is perithreshold and can be sub- or suprathreshold on different tests. If you run the online Matlab code that generates figure 34.4 many times, you will always find the positive correlation at around 600 ms to be statistically significant; however, the low-frequency negative correlation at around 300 ms may or may not be statistically significant (it will more often not be statistically significant). In this case it would be appropriate to perform the entire permutation testing procedure several times (e.g., 20 times) and take the average. This will help stabilize the statistical significance more than increasing the number of null-hypothesis iterations (1000 were used to generate figure 34.4).

One final tip here concerns computing correlation coefficients. If you do not care about the correlation coefficient value itself but rather the direction and statistical significance of the effect, you can use a least-squares fit without scaling the result to a correlation coefficient. This does not change the interpretation of the result or its statistical significance. The equation is discussed in the next section, and figure 27.9 shows that it is much faster than computing a correlation coefficient.

34.4 Continuous Relationships with Power: Single-Trial Multiple Regression

Perhaps you have more than one independent variable that you want to correlate with brain activity, such as reaction time and a posttrial confidence rating. If so, you should not perform several separate correlations but, rather, a multiple regression. Particularly if the independent variables are correlated, applying multiple simple correlations can inflate or misrepresent multivariate relationships and may increase the possibility of false alarms (however, if the independent variables are strongly correlated, this introduces another problem, called multicollinearity). Therefore, multiple regression is a preferred strategy for linking several independent variables to EEG data.

Multiple regression involves creating a design matrix and fitting that matrix to the data using ordinary least squares. There are other methods for computing the fit of a design matrix to data, but ordinary least squares is a simple, robust, and widely used method. The design matrix contains one row for each trial and as many columns as you have independent variables. In the example design matrix shown in figure 34.5A, there are two regressors:

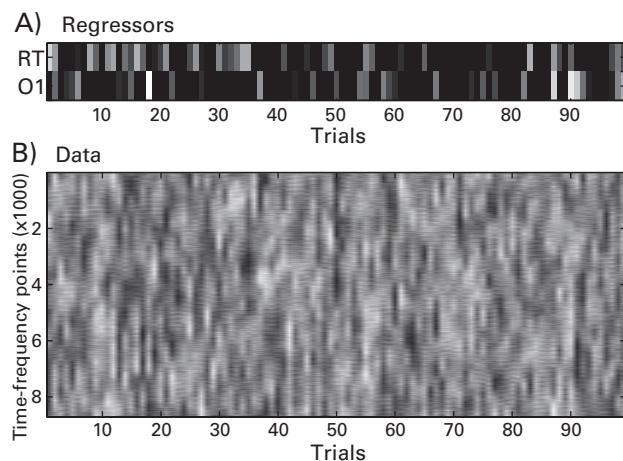


Figure 34.5

Design (panel A) and data (panel B) matrices. The regressors and data are normalized (mean centered and standard-deviation scaled); if they have a mean offset, an intercept (extra regressor of all ones) should be added. “RT” refers to reaction time, and “O1” refers to alpha-band power at electrode O1. The data matrix in panel B contains trials in columns and time-frequency points in rows (time points in the buffer zone were removed, thus leaving 436 time points and 20 frequencies, or 8720 points in total). The least-squares fit of the design matrix on the data produces a 1-by-8720 vector, which can be reshaped back to a time-by-frequency matrix.

reaction time and alpha-band power at electrode O1 between 0 and 250 ms post-stimulus-onset. Note that the power regressor is similar to the approach of power-based connectivity shown in figure 27.6C.

The goal of the regression analysis is to obtain regression coefficients (also called beta weights) for each regressor that express the mapping of the regressors to the data. To this end, the data should be reshaped to a 2-D matrix of trials by time and frequency (figure 34.5B). The solution to ordinary least squares can be obtained using the following equation.

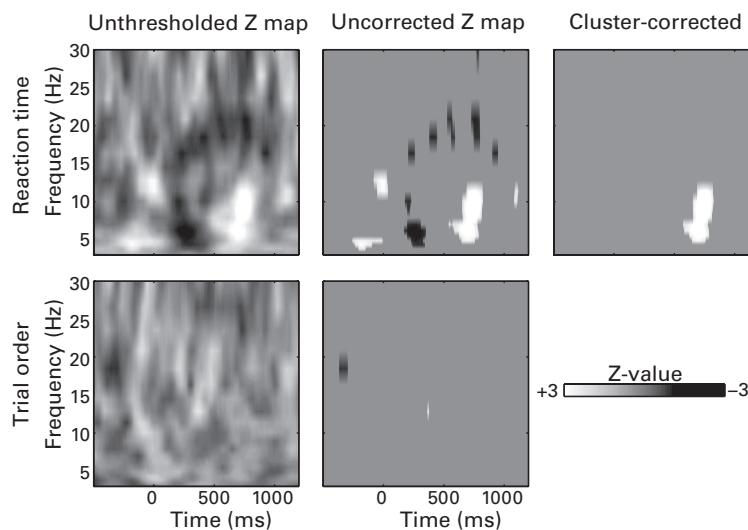
$$\beta = (XX^T)^{-1}XY^T \quad (34.1)$$

in which β is the vector of regression coefficients, X is the design matrix (figure 34.5A), the superscript T indicates the transpose, and Y is the data matrix (figure 34.5B). Parts of this equation should look similar from chapters 24 (principal components analysis) and 27 (power correlations). In particular, the XX^T term is the covariance of the regressors (the design matrix), and the XY^T term is the variance between the regressors and the data. The format of equation 34.1 assumes that the X matrix is regressors-by-trials and that the data matrix is time-frequency-by-trials (in other words, variables-by-observations). This format is used here to be consistent with the equations presented in chapters 24 and 27. If your matrix dimensions are the other way around (e.g., trials-by-regressors and trials-by-data), you would write $(X^TX)^{-1}X^TY$.

Statistical significance of the β coefficients can be obtained through parametric or nonparametric approaches. The nonparametric approach involves repeatedly applying equation 34.1 while shuffling the trial order of the regressors on each iteration, thereby creating a distribution of β coefficients expected under the null hypothesis.

Results of this analysis are shown in figure 34.6. After cluster correction for multiple comparisons, one cluster remained significant for the reaction time regressor (this is the same effect as was observed in figure 34.4 [plate 25]), and no clusters were significant for the O1 alpha power regressor.

Multiple regression is a major topic in statistics and psychology. In fact, multiple regression is one example of a general linear model framework within which there are many possibilities for analyses, including using partial least squares, robust regression, and stepwise regression. Furthermore, you are not limited to using continuous trial-varying variables but can also include discrete conditions. The SPM8 toolbox implements general linear modeling approaches for within-subject M/EEG data (Litvak et al. 2011).

**Figure 34.6**

Results of a multiple regression of reaction time and post-stimulus alpha-band power from electrode O1 (see design matrix in figure 34.4A [plate 25]) on time-frequency power from electrode FCz. No results survived cluster-level correction for multiple comparison for the O1 power results.

34.5 Determining Statistical Significance of Phase-Based Data

There are at least three approaches to evaluating the statistical significance of ITPC (chapter 19) and ISPC (chapter 26), depending on the goal of the analysis. For the rest of this chapter the term ITPC is used, but the methods can also be applied to ISPC values.

One approach is to test the difference in ITPC between conditions or between a posttrial and a pretrial time period. Here, the question you are testing is not whether ITPC per se is significant but, rather, whether ITPC differs between conditions or between time segments. Although ITPC values are bound between 0 and 1, their difference values are bound between -1 and $+1$, similar to correlation coefficients. Thus, the Fisher-Z transform of condition differences in ITPC will be approximately normally distributed under the null hypothesis (figure 34.7). These difference values can then be used in group-level parametric analyses such as ANOVA or regression. ITPC differences can also be tested using nonparametric approaches as described below.

The second approach is to test whether an ITPC value is statistically significant on its own, not in relation to ITPC during a baseline time period or other condition. Here you have two

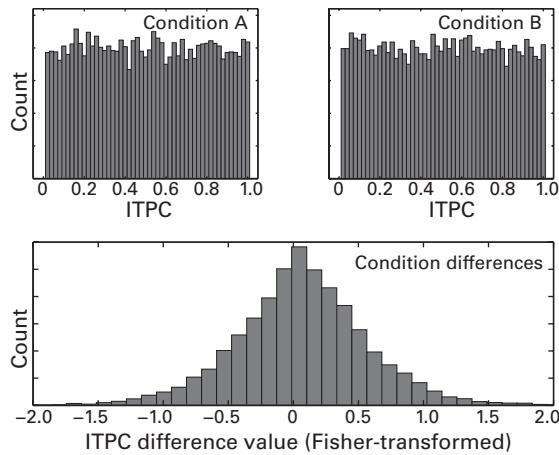


Figure 34.7

ITPC values are uniformly distributed (see top panel), but ITPC difference values between conditions, when Fisher-Z transformed, approach a normal distribution (bottom panel).

complementary options. One is to compute the p -value of the observed ITPC, and the second is to compare the observed ITPC to a “critical ITPC,” given a specific p -value. The following formula will allow you to compute the p -value of ITPC (Zar 1999).

$$p = e^{\sqrt{1+4N+4(N^2-(N*ITPC)^2)}-(1+2N)} \quad (34.2)$$

N is the number of trials (this would be time points if you are testing ISPC-time), and the asterisk indicates multiplication. It turns out that the above equation can be approximated using Rayleigh’s Z ($n*ITPC^2$, equation 19.2).

$$p = e^{-N*ITPC^2} \quad (34.3)$$

Some have cautioned that this approximation is invalid for a small number of data points, but in practice this is not a concern except at an extremely small number of data points, which is rare in EEG analyses. This is illustrated in figure 34.8.

The complementary approach to determine the statistical significance of an ITPC value is to compute the critical ITPC given a p -value and consider this value to be a statistical threshold. This critical value was plotted as a gray line in figures 19.5 and 19.6. The formula for computing $ITPC_{crit}$ is presented in equation 34.4.

$$ITPC_{crit} = \sqrt{\frac{-\ln(p)}{n}} \quad (34.4)$$

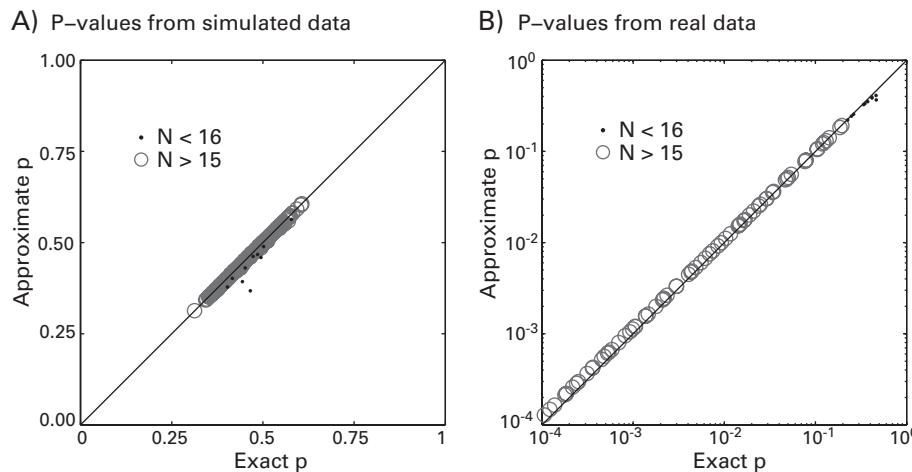


Figure 34.8

Comparison between exact p -values of ITPC (equation 34.2) and their approximation (equation 34.3). Black dots show the relationship for fewer than 16 trials; gray circles show the relationship for more than 15 trials. The diagonal black line depicts a hypothetical correlation of 1.0. This shows that even for a small number of data points, equation 34.3 is a close approximation to the exact p -value for ITPC. The x - and y -axes in panel B are shown in a log-log plot to facilitate visual inspection, because many p -values were close to zero.

In this equation, p is the desired p -value (e.g., 0.05), \ln is the natural logarithm, and n is the number of trials (or time points for ITPC-time). Any ITPC value that is greater than $ITPC_{crit}$ can be considered statistically significant at the p -value specified in equation 34.4.

Note that equations 34.2, 34.3, and 34.4 rest on the assumption that the phase angles under the null hypothesis are drawn from a von Mises distribution, which is the circular analog to a normal distribution (Stephens 1969).

The previous options are useful when you are testing a relatively small number of ITPC points, which is likely to be the case if your analyses are hypothesis driven. However, those methods are not well suited for controlling for multiple comparisons over many time-frequency points, which will be the case in exploratory analyses. Thus, the third approach for statistical testing of ITPC values is to use nonparametric permutation testing.

ITPC is driven by the consistency of the timing of frequency-band-specific activity over trials. Thus, the null hypothesis is that shifting the time course of the phase angle time series by a random amount would not affect the final ITPC strength. The null hypothesis can be generated by shifting the time series of each trial by a random amount, as shown in figure 34.2A. Figure 34.9 shows results from the single-trial shuffling procedure.

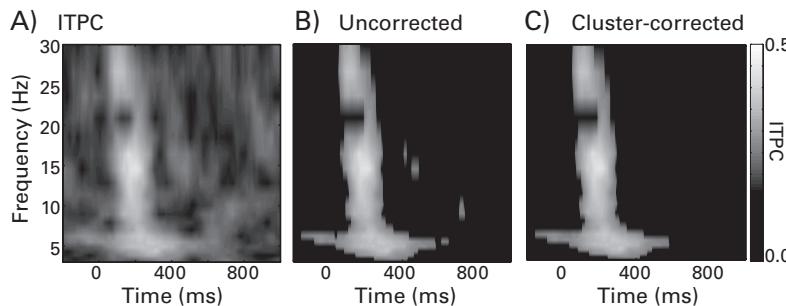


Figure 34.9

ITPC from electrode P1, showing all ITPC values (panel A), ITPC thresholded at an uncorrected p -value of 0.05 (panel B), and thresholded again after correction for multiple comparisons using cluster-based statistics (panel C). No pixels survived correction for multiple comparisons using pixel-based correction, which is why that map is not shown.

34.6 Testing Preferred Phase Angle across Conditions

In most studies using ITPC, the ITPC value itself (i.e., the length of the average vector) is evaluated in statistical comparisons. However, it is possible that the length of the vector is similar across conditions but the mean phase angle differs. That is, the strength of phase clustering is similar, but the phase angle at which maximal clustering occurs differs. In this case the preferred phase angles can be tested using the *gv*-test described in chapter 26.10. Another approach is phase bifurcation (Busch, Dubois, and VanRullen 2009), which tests whether phase distributions are different between two conditions and a baseline period.

34.7 Testing the Statistical Significance of Correlation Coefficients

The following equation allows you to test the statistical significance of a correlation coefficient:

$$t = r \sqrt{\frac{n-2}{1-r^2}} \quad (34.5)$$

in which r is the correlation coefficient, and n is the number of data points that were used when computing the correlation coefficient. This would be the number of trials per condition if you computed the correlation coefficient over trials, or it would be the number of subjects if you computed the correlation coefficient over subjects; t is a t -statistic that can be evaluated with $n - 2$ degrees of freedom. The Matlab function `tcdf` will return a p -value associated

with this t -value. The easiest way to use this function is to write `1-tcdf(abs(t),n-2)`. The reason for the “1-” and the absolute value is that the function returns the probability of a value being smaller than the t -value. Thus, the p -value will be close to 0 for large negative t -values but close to 1 for large positive t -values. The “1-” and absolute value will therefore ensure that the resulting p -value always approaches 0 as the t -statistic becomes increasingly extreme, regardless of its sign. Also note that the resulting p -value is for a two-tailed test.

For task data, you might compute correlation coefficients for different conditions, such as cross-trial correlations between EEG activity and reaction time for condition A and condition B. Equation 34.6 will allow you to test whether these two correlation coefficients are significantly different from each other (Cohen and Cohen 1983).

$$Z_{diff} = \frac{r_1 - r_2}{\sqrt{(n_1 - 3)^{-1} + (n_2 - 3)^{-1}}} \quad (34.6)$$

In this equation r_1 and r_2 are Fisher-Z-transformed correlation coefficients, and n_1 and n_2 are the numbers of data points that went into the two correlations. Z_{diff} can be converted to a p -value assuming a normal distribution. Note that equation 34.6 is the appropriate way to compare the strengths of correlation coefficients; if you separately evaluate the statistical significance of r_1 and r_2 , and find that r_1 is statistically significant but r_2 is not, this cannot be interpreted as indicating the r_1 and r_2 are different from each other.

35 Group-Level Analyses

Group-level statistical analyses can be performed using either nonparametric permutation-based approaches or parametric approaches such as ANOVAs. Nonparametric permutation testing on group-level analyses involves following the procedures outlined in chapters 33 and 34 except that trial-averaged data over subjects are shuffled instead of single trials within one subject. The permutation testing approach is useful for nonnormally distributed data or if the analyses are exploratory, and therefore corrections for multiple comparisons are required. The parametric approach, on the other hand, allows for complex designs beyond bivariate comparisons, including the types of analyses that psychology-oriented scientists may be more familiar with, such as repeated-measures mixed-design ANOVAs and factor analyses.

This chapter is focused on general strategies for performing group-level statistics rather than on their mathematical and Matlab-implementational details. Three strategies for performing group-level statistics are presented. Because each strategy has its advantages and limitations (discussed below in each section), it may be fruitful to apply several analysis strategies to gain a more complete understanding of the nature of the results.

35.1 Avoid Circular Inferences

Also colloquially called “double-dipping,” circular inference refers to inappropriate analysis strategies that lead to biased statistical analyses and can also lead to misinterpretations of the results. (Note that circular inference should not be confused with circular analyses; the latter refers to analysis techniques based on treating phase angles as points on a circle.) Most circular inferences result from selecting data that show a particular characteristic and then testing whether that characteristic is statistically significant within the data selected. In other words, circular inference occurs when the data selection method is correlated with the statistical analysis.

Here is an example of a circular inference: imagine that your hypothesis is that there is more gamma power in condition A than in condition B. You examine a topographical map of gamma power condition differences and observe that the condition difference is largest at electrode POz. Then, you select the data from electrode POz and apply a *t*-test to gamma power between conditions using a *p*-value threshold of 0.05. This analysis strategy is biased because the procedure used for selecting data (a condition difference in gamma activity) is correlated with the statistical evaluation (a condition difference in gamma activity). In this example, two possible appropriate analysis strategies (these are not the only possible appropriate analysis strategies) would be (1) to select an electrode based on an increase in gamma power over all conditions averaged together and then test for possible condition differences in gamma power within that electrode, or (2) to create a topographical map of condition differences as described above but apply an appropriate correction for multiple comparisons across all electrodes.

Avoiding circular inferences is generally not difficult if you have had some statistics education and can think critically about how the data are selected and presented. In some cases, analysis strategies can be appropriate or inappropriate, depending on how they are used and interpreted. Tips on avoiding circular inference are presented throughout this and the next chapters, and see Kriegeskorte et al. (2009, 2010) for more discussion on the dangers of circular inference and advice for how to avoid them.

35.2 Group-Level Analysis Strategy 1: Test Each Pixel and Apply a Mapwise Threshold

Group-level analysis strategy 1 is appropriate for analyses in which you have no strong a priori hypotheses concerning the time-frequency-electrode window within which to perform statistical analyses. In general, this approach is less susceptible to circular inference because, although contrasts are formed based on expectations of the results (that is, activity during condition A is compared against activity during condition B), corrections for multiple comparisons are applied.

The advantages and limitations of group-level analysis strategy 1 were discussed in chapters 33 and 34. Briefly, this strategy allows the discovery of effects without being overly constrained by theory and hypotheses. This approach is therefore particularly useful in novel experiments or studies for which the electrophysiological characteristics are not well known. This approach can also be used to confirm the temporal and frequency specificity of hypothesis-driven results. For example, if your hypothesis concerns the relationship between beta-band power suppression over motor areas and severity of Parkinson disease, you may perform statistical analyses only on 15- to 25-Hz power using group-level analysis strategy 2a

(discussed in the next section). However, if you take only this statistical approach, you will not know whether beta-band power is the only time-frequency dynamic that is related to the task. Thus, after performing the hypothesis-driven analysis, you can apply group-level analysis strategy 1 to investigate whether beta-band activity is the only relevant time-frequency feature or whether other time-frequency features (for which there were no hypotheses) also show a similar effect.

There are three main limitations of group-level analysis strategy 1. First, theoretically relevant but subtle findings might be missed because the statistical threshold must be appropriately stringent for all of the comparisons that are performed. Second, nonparametric permutation testing has limited options for factorial designs and other statistical tests that are more complex than bivariate comparisons. This limitation will likely affect researchers leaning toward the psychology end of the cognitive electrophysiology spectrum more than it will affect researchers leaning toward the neuroscience end of the spectrum (see figure 1.1). Third, this strategy has minimal flexibility for investigating individual differences because it assumes that each time-frequency point is the same for all subjects. That is, pixel-based analyses assume that the activity at 10.4 Hz and 625 ms at electrode F4 reflects the same neural process for all subjects.

35.3 Group-Level Analysis Strategy 2a: Time-Frequency Windows for Hypothesis-Driven Analyses

The next two analysis strategies can be used when testing hypotheses that specify the time-frequency windows within which to test for effects. For example, if you are performing a study on the influence of motivation on response conflict, you can hypothesize based on previous research that the effects of interest will be in the time interval between stimulus and response, in the theta frequency band, and at electrode FCz. In this case the specificity of the hypothesis allows you to maximize statistical sensitivity by analyzing a specific time-frequency-electrode window. It is not necessary to test for an effect in, for example, the beta-band over occipital electrodes; doing so makes little sense given the hypothesis and will needlessly require more stringent statistical thresholding to correct for multiple comparisons.

To follow group-level analysis strategy 2a, first select a time-frequency-electrode window. Using the example from the previous paragraph, you might select a time-frequency window of 4–8 Hz from −300 to +100 ms around each trial’s button press at electrode FCz, because this time-frequency window was used in a previous study on this topic. The window can also be guided by a combination of previous findings and the characteristics of the results. Thus, you would expect the window to be around the time of the response, around the theta

band, and around electrode FCz, but you want the flexibility to define the precise time and frequency boundaries based on the data. Defining the boundaries of these windows can be done in one of two ways (discussed after the following paragraph), but it is important that the windows be defined in a manner that is orthogonal to the condition comparisons.

This is an important point. If the boundaries of the time-frequency window are based on the results, they cannot be based on the condition differences. For example, if your study involves comparing conditions A and B, the time-frequency window selection must be made on the time-frequency results of $(A + B)/2$, not on A-B. This latter method is biased because you will select data to be analyzed based on those data showing the effect you plan on testing. The selection of the window could also be based on an orthogonal condition effect such as the condition difference C-D. As long as conditions C and D are independent of conditions A and B, the selection of the window for the test A-B is not biased.

Once you have determined the appropriate results contrast from which to define the window, there are two methods to define the precise boundaries of that window. The first method is to draw a box in time-frequency space based on visual inspection of the results. This is done by examining the condition-average results (or any other hypothesis-independent contrast) and drawing a box around the time-frequency feature or features that are most relevant to the hypotheses (figure 35.1A). The other method of defining the boundaries of the window is to select pixels that are statistically significant based on a contrast that is orthogonal to the hypothesis (figure 35.1B). This is analogous to contrast-masking procedures done in fMRI, whereby a mask is created based on statistically significant voxels in one contrast (such as all conditions versus baseline) and then activity at those voxels is tested in an orthogonal contrast (such as conditions A-B). You should pick one of these methods for defining boundaries and use it consistently for all analyses within a study.

Regardless of how you define the boundaries of the window, it is a good idea to use windows rather than a single time-frequency point. Averaging points within a window will increase signal-to-noise ratio and minimize the possibility of the results being influenced by one pixel that happens to contain noisy or nonrepresentative data.

After precise boundaries have been defined, average together the data from all pixels in that time-frequency-electrode window, separately for each subject and each condition, and store the results in a matrix. This matrix can then be exported to a text file and then imported into a statistics program such as SAS, SPSS, or R. If you need a refresher on exporting formatted data from Matlab, see the code accompanying chapter 4 (script “c”). From here, you can perform any number of analyses that are appropriate for your experiment design and hypotheses, ranging from repeated-measures ANOVA to structural equation modeling. Figure 35.1 shows what one of these exported data files might look like. Each row contains

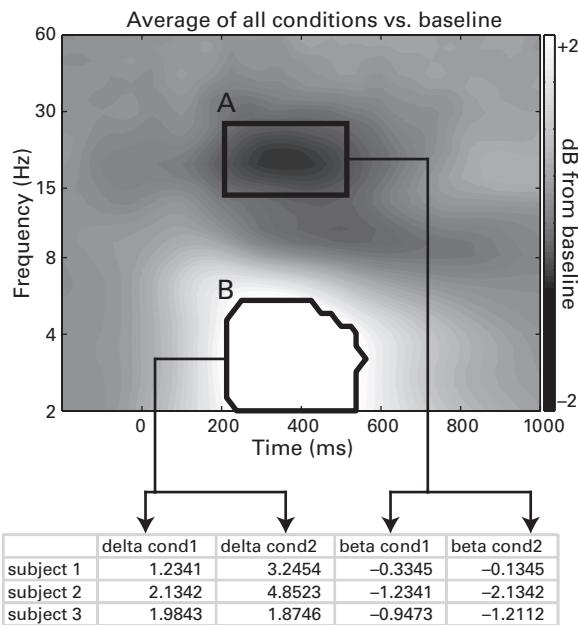
**Figure 35.1**

Illustration of extracting data for group analysis strategy 2a. Based on either visual inspection (box A) or a statistical contrast orthogonal to the hypotheses (area B), time-frequency regions are defined. In practice, either but not both of these methods should be used within a study; both are shown here for illustration. The activity from all pixels in each region is averaged separately for each subject and each condition and then placed into a matrix that can be exported to a statistics program. Although this figure illustrates only two conditions (cond1 and cond2), this approach is well suited for factorial designs.

data from one subject, and columns are variables. Depending on your design and hypotheses, it is possible that you will need to correct for multiple comparisons over multiple time-frequency windows or electrodes. In this case you could use a Bonferroni correction.

Why should you export the data from Matlab to another software program? Unfortunately, Matlab lacks a comprehensive toolbox for the kinds of behavioral statistics often performed in psychology research. For example, there are no functions available in Matlab (as of late 2012) for performing an unbalanced repeated-measures ANOVA with Greenhouse-Geisser correction for sphericity violations, and yet within psychology research this is a commonly used analysis. Perhaps in the future there will be a behavioral statistics toolbox for Matlab, but until that time, you may need to export the data to another program. There are also routines for allowing Matlab and R to communicate with each other and share data. If you are comfortable with R, this might be an approach worth looking into.

There are several advantages of group-level analysis strategy 2a. The main advantage is that it is designed for hypothesis testing. Hypothesis testing promotes the development of theories and models and is a useful way to compare results across species, experiments, and methodologies. Hypothesis testing also maximizes statistical sensitivity because you test only a few comparisons rather than hundreds or thousands of comparisons that require appropriately stringent statistical thresholds.

There are also several limitations of group-level analysis strategy 2a. This strategy, like strategy 1, rests on the assumption that the window defined for the analysis reflects the same time-frequency feature for all subjects; individual differences in temporal, spatial, and frequency characteristics are ignored. A second limitation is that if the hypotheses are misguided, you might miss the most important features of the data simply because you specified the wrong time-frequency window.

Finally, in some cases, defining a window based on the condition average may provide misleading or null results. For example, if condition A increases power and condition B decreases power, the condition average would show no change in power relative to baseline. Another example is that the condition differences might not be in the same time-frequency regions where the main condition effect is. An example can be seen in figure 35.2: based on the condition average power effect, a time-frequency window was selected in the alpha band (panel A); however, the condition-difference time-frequency plot revealed that the main condition effect was in a slightly lower frequency band (panel B). In this particular case there are two appropriate strategies. First, keep this time-frequency window; although it might not

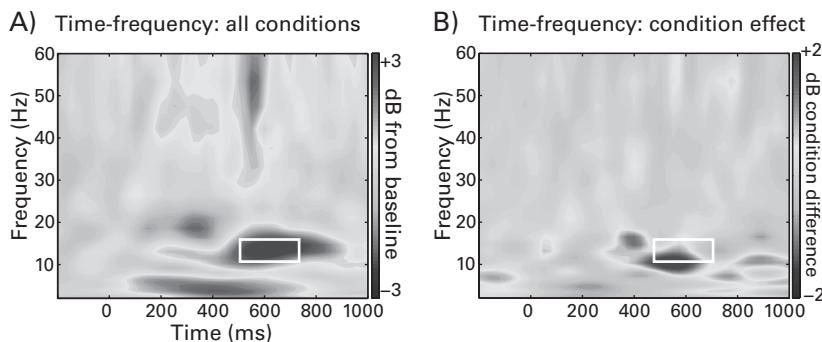


Figure 35.2

Illustration of one example of the danger of selecting a time-frequency window based on a condition-averaged effect. In this case a time-frequency window was selected based on the condition-averaged effect (shown in panel A), but this window might not be ideal for the condition difference (panel B). See also figure 36.1 (plate 26) for color versions of these plots.

be an ideal window for condition differences, it was defined in a statistically appropriate and bias-free manner. Second, use group-level analysis strategy 1 and apply appropriate corrections for multiple comparisons over all time-frequency pixels (in this case, -200 ms to 1000 ms and 2 Hz to 60 Hz). It would be inappropriate to reselect a time-frequency window (in panel A) based on visual inspection of the condition difference (in panel B); this would be circular inference.

35.4 Group-Level Analysis Strategy 2b: Subject-Specific Time-Frequency Windows for Hypothesis-Driven Analyses

This strategy is similar to strategy 2a in that time-frequency windows are defined, data are averaged from all time-frequency points within that window, and then those data are submitted to subsequent statistical analyses. The difference from strategy 2a is that the windows can be defined individually for each subject. Strategy 2b thus increases sensitivity to individual differences in peak times and frequencies of time-frequency features. This may be relevant because individual variability in frequency peaks has been linked to, for example, genetics (Landolt 2011; Linkenkaer-Hansen et al. 2007; Smit et al. 2012), GABA concentration (Muthukumaraswamy et al. 2009, 2012), and memory task performance (Cohen 2011a; Moran et al. 2010). Strategies 1 and 2a ignore these individual differences because those strategies make the implicit assumption that there is no meaningful cross-subject variability in the temporal and frequency characteristics of the results.

Similar to strategy 2a, you start by defining a time-frequency-electrode window based on the group results. However, the windows for strategy 2b should be larger than those you would use for strategy 2a. This large window can be seen as a group window within which smaller windows will be defined for each subject. Thus, the group window should be large enough to account for what you expect will be a reasonable amount of variability across subjects. For example, if your hypothesis concerns the theta band, you might define the group window to span 2 Hz to 10 Hz, thus allowing for some subjects who may have relatively low or high theta-band spectral peaks. After this group window has been defined, smaller windows can be identified for each subject (subject window) based on each subject's time-frequency peak within that group window (see figure 35.3). It is still a good idea to have a window around each subject's peak to increase the signal-to-noise ratio.

As with strategy 2a, it is critical in strategy 2b to define windows at both the group and subject levels in a manner that is appropriate and avoids circular inference. Thus, the windows could be based on a condition-average time-frequency map or based on a condition

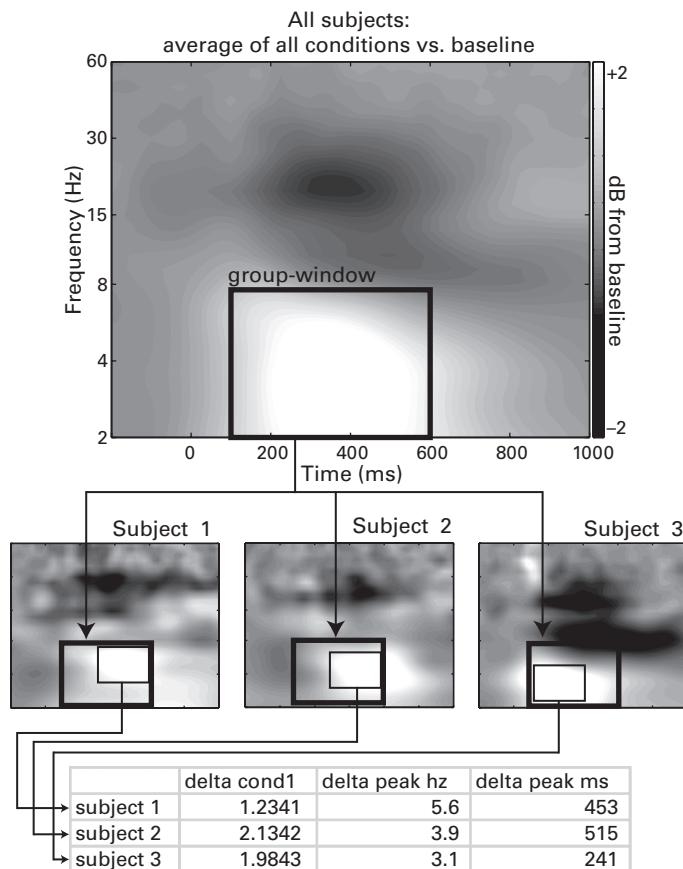


Figure 35.3

Illustration of group-level analysis strategy 2b. A relatively large window is drawn based on the average of all conditions and all subjects (the group window). From that window a smaller window within the group window is selected for each subject individually, based on the subject-specific time-frequency peak. This method allows you to extract not only the average activity for each condition and subject but also peak time and frequency points.

comparison that is uncorrelated with the condition comparison that you will test (the C–D example in section 35.3).

The advantages of strategy 2b are the same as those of strategy 2a, and strategy 2b has the additional advantage that it facilitates investigation of individual differences in time-frequency peaks. The limitations of strategy 2b are also the same as those of strategy 2a. Strategy 2b has an additional limitation compared to strategy 2a, which is that the increased sensitivity to individual differences also means increased sensitivity to noise and nonrepresentative data. Selecting peaks based on the condition-average activity will help increase the signal-to-noise ratio and thus compensate to some extent for nonrepresentative peak values. If you are concerned that the single-subject windows might become driven by nonrepresentative data, you should use strategy 2a.

35.5 Determining How Many Subjects You Need for Group-Level Analyses

There is no magic number of subjects you need to interpret group-level analyses with confidence. The number of subjects you need depends on several factors, including what kinds of effects you are focusing on, how many trials were performed by each subject, how clean the data are, whether you will examine individual differences, and so on. Large-size effects require fewer subjects, whereas subtle effects require more subjects. The more trials each subject performs, the cleaner the trial-average single-subject data will be, and thus, the fewer subjects you will need. If you are testing for a large-size effect that will be present in all subjects, eight subjects might be sufficient for group-level analyses of this robust effect. But if the focus of the study is on individual differences, eight subjects are unlikely to provide enough data for correlation analyses. If you are trying to demonstrate a methodological point or develop an analysis method, only a few subjects (perhaps even only one subject) are necessary as a proof-of-principle demonstration. If you are trying to demonstrate the workings of a social psychological construct that depends on cultural background, perhaps 50 subjects might be an appropriate minimum.

The number of subjects you include also depends on practical factors such as budget and feasibility. For example, studies on long-term cognitive training and studies with brain-damaged patients might use fewer subjects because of the difficult and time-consuming nature of acquiring the data. In general, experiment designs that involve within-subjects analyses provide higher statistical power and therefore require fewer subjects compared to those that involve between-subjects analyses.

As you can see, there is no specific number of subjects that will be appropriate for all studies. There are statistical guidelines to determine how many subjects you need. These

guidelines are usually based on statistical power, which is a metric that concerns how much confidence you can have in rejecting the null hypothesis. The computation of statistical power depends on the analysis you want to perform and on the assumptions you make about the distribution of the analysis parameters under the null hypothesis. Thus, within one dataset, an appropriate number of subjects for some analyses may be an insufficient number of subjects for other analyses.

For most cognitive electrophysiology studies, 20 subjects should be sufficient to identify both large and subtle effects and to perform individual differences correlation analyses. However, this is a recommendation, not a rule. If you are unsure how many subjects you will need for your study, find publications that applied similar methods, analyses, and cognitive constructs, and use their number of subjects as a guideline.

When planning how many subjects to test, keep in mind that it is typical to exclude a few subjects from group analyses due to excessive EEG or cognitive artifacts, poor task performance, or technical problems during data acquisition. In most cases around 10% or fewer datasets need to be excluded. Thus, if you want to have a sample size of 20 subjects for group-level analyses, consider acquiring data from 23 subjects.

You should not collect data one subject at a time, perform group-level analyses, and decide whether to test another subject based on the *p*-value of the group-level result. In other words do not keep acquiring data until the *p*-value of the hypothesized effect is significant. *p*-Values can fluctuate with the number of subjects, particularly when the number of subjects is relatively small (Simmons, Nelson, and Simonsohn 2011). You should decide in advance how many subjects you will test. If you decide to test more subjects after looking at the group-level results, decide to collect a specified number of subjects (e.g., another 10 subjects) before looking at the group-level results again.

36 Recommendations for Reporting Results in Figures, Tables, and Text

Communicating your results and ideas to the scientific community is almost as important as analyzing the data. There are many reasons why you should try to disseminate your findings in a clear and concise way. There are personal motivations—peer-reviewed publications are the currency of science and facilitate career advancement and grant funding. There are scientific motivations—showing the scientific community your results opens opportunities for constructive criticisms and collaborations and therefore will help you become a better scientist. And there are communal motivations—if you want to contribute to science and increase the corpus of human knowledge, you need to communicate your results in a manner that will disseminate your knowledge into the world and inspire others to improve their research based on your findings. Regardless of your motivations, making clear figures that are easy to interpret is a crucial part of communicating your results and ideas to the scientific community.

Probably you, like many scientists, are busy and, sadly, do not have enough time to read thoroughly all of the papers that you would like to (and probably should). Probably you have had the experience of downloading the pdf of a publication and deciding whether to read it based on the title, the abstract, and the figures. For those readers who teach or give lectures often, you may also know the experience of deciding which scientific findings to highlight in your lecture based on which publications have clear figures that can be interpreted with minimal effort.

Making excellent scientific figures is an art. I do not claim to be an artist in this respect, and I do not always make excellent or easily interpretable figures. But I have tried to think about what differentiates good figures from poor figures. This chapter contains recommendations and points of consideration for making figures. You may disagree with some of the suggestions in this chapter. Even if you agree with all of them, there is no guarantee that following them will produce excellent figures. Furthermore, there are no rules to follow to make excellent figures; some guidelines may be more or less useful depending on what

message you are trying to convey in the figure. The best way to make good figures is to think carefully and critically about them. Therefore, this chapter is not a cookbook for making figures but, rather, is a list of points to consider that I hope will help you make excellent figures. Additional advice about how to show scientific results can be found in these references (Allen, Erhardt, and Calhoun 2012; Franzblau and Chung 2012), any of Tufte's books on the topic (e.g., Tufte 1983), and of course, the humorous and informative book *How to Lie with Statistics* (Huff 1991).

36.1 Recommendation 1: One Figure, One Idea

Each figure should have only one “take-home message.” Expressing too many ideas within one figure makes the figure confusing and difficult to interpret. Similarly, redundantly expressing one idea over two or more figures might lead readers to wonder whether they misunderstood the idea of one or both figures. This does not mean that a figure should contain only one result—many results might be related to one idea.

If a figure has one take-home message, you should be able to write a one-sentence summary that expresses the main idea of that figure. For example, “This figure shows that stimulus-contralateral alpha suppression during an attention task is enhanced after eating a grapefruit.” This figure might show several results, perhaps from several experiments and follow-up control tests, but all of the results support the one idea of that figure—eating a grapefruit boosts the attention-modulation of alpha suppression. This summary sentence could be the first sentence in the figure legend. This will help you decide whether the figure contains too much or too little information and it will also help readers quickly interpret the figure.

36.2 Recommendation 2: Show Data

One of the many advantages of EEG is that you can show data that have been only lightly processed. Showing data in figures is less common in fMRI research and in many psychology studies; instead, parameter estimates from a statistical model that reflect the fit of that model to the data are shown. Time-frequency power plots, for example, simply reflect the raw data after temporal filtering and baseline normalization. Therefore, you can and should show data in figures.

How much data you should show depends on whether the study was driven by hypothesis testing or by data exploration. If the study was strictly hypothesis driven, only the

most important results need to be shown, and this might be best done with bar plots or scatterplots. For hypothesis-driven tests, showing too much data can be confusing. For example, if your study concerns lateralized visual evoked potentials, showing time-frequency plots of cross-frequency coupling at anterior electrode AF3 will only confuse readers.

On the other hand, if your study is primarily exploratory and data driven, there are a lot of results to show, and it might not be known which results will become the most relevant. Therefore, there should be many figures illustrating results using several different slices of the results hypercube (figure 3.2). The Matlab utility `tfviewervx` (available in the online Matlab code) should help you search through the results hypercube to know which slices show the results effectively.

The following is a general suggestion for how to show task-related time-frequency results. First, show one plot (e.g., a time-frequency plot and/or a topographical map) that illustrates the overall task-related effects, measured as the average activity of all conditions relative to baseline (“activity” is whatever measure is most relevant, such as power, ITPC, or connectivity). This is useful because it illustrates the electrophysiological landscape of task-related activity. Next, show plots of condition differences to highlight which of the dynamics are most relevant for the cognitive process under investigation. Use the same time-frequency-electrode characteristics that were used in the first plot to facilitate comparison. An example is shown in figure 36.1 (plate 26).

One limitation of difference plots is that it may be difficult to infer the direction of the effect for each condition. Thus, it may be useful to supplement condition difference plots with bar plots that illustrate the effect for each condition separately. An example is shown in figure 36.2. The two sets of bar plots illustrated in panel B are equally likely given the difference map in panel A, but these two patterns of results would elicit different interpretations. Note that illustrating bar plots based on condition differences is not circular inference when done appropriately: the data were selected for the bar plots because of a condition difference, and thus, the difference between the bars is biased toward being statistically significant. It would be inappropriate to perform a statistical test on the data shown in the bar plots because those data were selected in a biased manner. However, showing the bar plot to facilitate a qualitative interpretation of the condition difference is not circular inference. If you are concerned that readers might get confused about the appropriateness of how you show the data, you could mention explicitly that the data in the bar plot are selected based on their statistical significance and are thus shown to facilitate interpretation rather than to be further statistically analyzed.

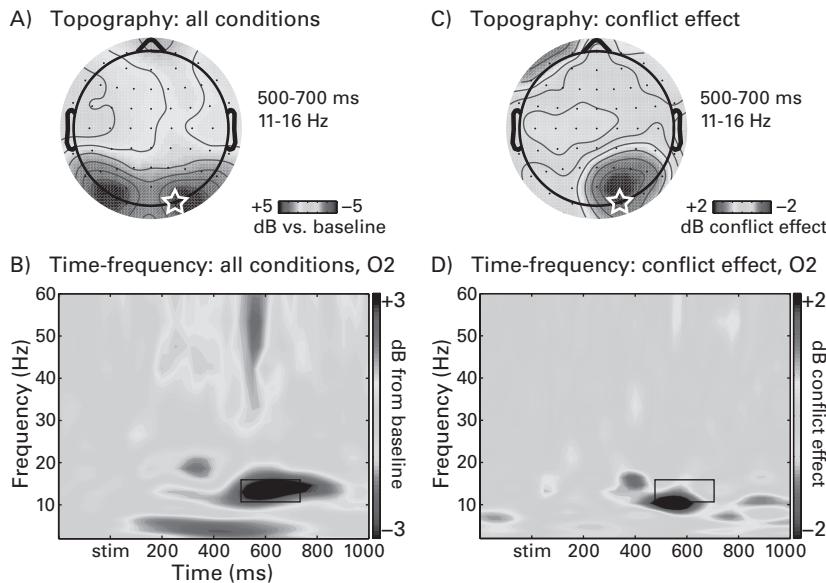


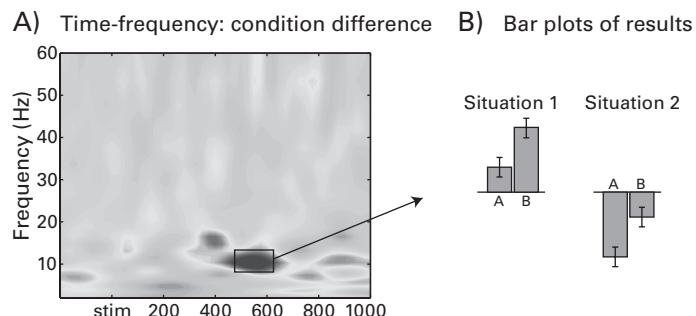
Figure 36.1 (plate 26)

Recommendation for how to show results over time, frequency, and space. Panels A and B show results from all conditions averaged together, relative to the baseline activity. These plots are useful because they show the basic task-related responses. The white star over the right posterior electrode O2 illustrates that the time-frequency plot was taken from this electrode, and the black box in the time-frequency plot illustrates the window selected for the topographical map. Panels C and D show the condition effect (in this case it was a subtraction of trials that contained response conflict from trials that contained no response conflict).

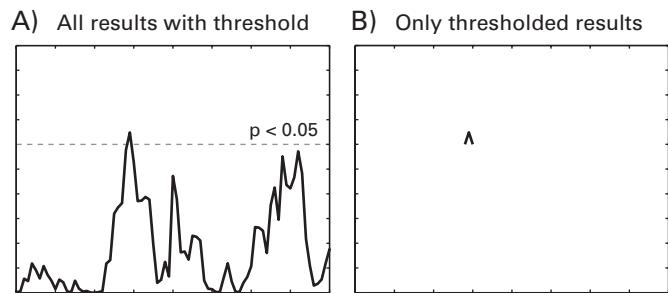
36.3 Recommendation 3: Highlight Significant Effects Instead of Removing Nonsignificant Effects

Consider figure 36.3. Panels A and B both illustrate that there is a suprathreshold effect at around the third tick on the *x*-axis. Panel A, however, reveals other features of the results that are large in magnitude and nearly statistically significant. These features of the results might be theoretically relevant or might be of interest to readers who can link those results to other literatures.

For time-frequency plots, you can show all of the data and use contour lines to illustrate statistically significant regions. An example using real data is shown in figure 36.4 (plate 27), and the online Matlab code from chapter 33 will show you how to overlay significant results

**Figure 36.2**

Although a condition difference plot highlights regions in time-frequency space where there are condition differences (panel A), they do not show the direction of the effect for each condition. The bar plots in panel B highlight two situations that would produce an identical condition difference map and test statistic but would have different interpretations. Because the bar plots were selected from a time-frequency region that shows a condition difference, it would be inappropriate to perform statistical tests on the data in the bar plots. However, showing the data for visual inspection is appropriate and will facilitate a deeper appreciation of the results.

**Figure 36.3**

An information-rich and complex landscape (panel A) may be poorly represented by removing all subthreshold results (panel B).

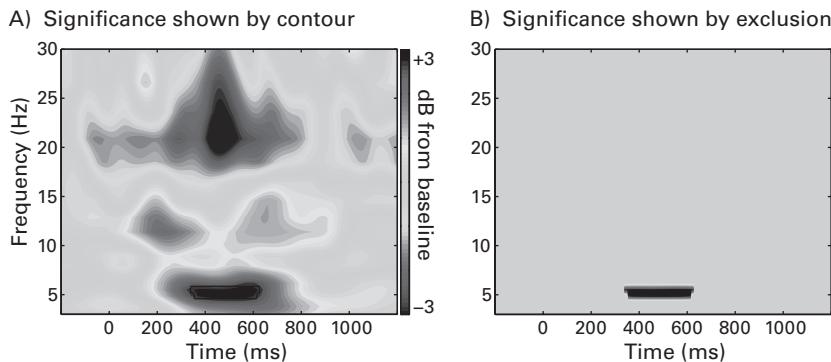


Figure 36.4 (plate 27)

Two methods of showing statistical significance in time-frequency plots. Panel A shows all results and highlights the statistically significant time-frequency regions. This is a useful representation because readers can see a broad range of dynamics and might be able to link the results to other literatures and findings. It is also useful because it shows what other effects might be present in the data that are perhaps relatively subtle, and it shows whether the effects are specific to a certain time-frequency range (see also figure 36.3). In contrast to panel A, panel B contains an alternative method of showing significant results. Here, all nonsignificant pixels are set to zero. Although this makes the significant effect more visually salient, it also obscures a considerable amount of information in the time-frequency dynamics from this electrode. Other examples of highlighting significant regions versus setting nonsignificant pixels to zero were shown in chapter 34.

using single contours. For topographical maps, you can indicate electrodes with statistically significant results by putting circles or stars on those electrodes.

The advantage of showing all results is that it allows you and readers to determine whether the statistically significant effect is an “island” or merely the tip of an iceberg. This is a non-trivial point: the interpretation of a result might be different if a significant region appears to be brief in time and localized in frequency, compared to a temporally extended and broadband effect for which one particular region happened to have relatively less variance and therefore to exceed a significance threshold.

36.4 Recommendation 4: Show Specificity (or Lack Thereof) in Frequency, Time, and Space

Imagine you are conducting an experiment about the role of motivation on conflict monitoring. Based on previous findings, you hypothesize that there will be modulations in theta-band power between the stimulus and response at electrode FCz. Technically, to test

this hypothesis, you need to use only one wavelet centered at 6 Hz and analyze data only from electrode FCz.

In practice, however, it is useful to analyze activity in a range of frequencies (in this case, perhaps 2 to 40 Hz) and many electrodes. This will allow you to show time-frequency plots and topographical maps. This does not necessarily mean that you need to evaluate the statistical significance of activity in every electrode, frequency band, and time point, and it does not mean that you need to correct for multiple comparisons across all possible results that could be tested. You can still restrict your statistical analyses to the window for which you had a priori hypotheses. There are two reasons to show results from a broader time-frequency range than what your hypotheses specify.

First, it will allow you to determine whether the effect of interest is specific to the time-frequency-electrode window for which you had a hypothesis. As mentioned at the end of section 36.3, the interpretation of your finding might be different if the effect spans several frequency bands or hundreds of milliseconds rather than being isolated in one frequency band for a fairly brief period of time. Second, showing a broader range of results will allow you and readers to visually inspect a richer portion of the landscape of EEG activity. You might discover additional task-related dynamics that were not specified by your hypotheses. Post hoc data exploration should not be discouraged as long as it is labeled as such, evaluated with appropriate statistical corrections, and interpreted cautiously. Post hoc analyses and qualitative data inspection may help contextualize the findings and may provide novel links between your results and other literatures. This is particularly important when the time-frequency characteristics of the cognitive process under investigation are relatively unknown or understudied.

36.5 Recommendation 5: Use Color

Color provides an additional dimension for encoding information in figures and should be used when possible. The advantages of color for interpreting time-frequency plots can be seen in many figures in this book by comparing the grayscale figures in the main text with the color figures in the pages in the center of the book.

Choose your colors carefully. Some color combinations may look good on your computer screen but may be difficult to differentiate on someone else's computer screen or when printed on paper. The best colors to use in figures are those that are very different from each other. For example, you should use green and purple rather than sky blue and electric blue. If you are unsure whether your color combination is good, print out the figure rather than looking at it on your computer screen. It is also a good idea to print the figure in grayscale

because your figures should ideally be readable to people who use black-and-white printers. In some cases color figures will not be interpretable when converted to grayscale because, for example, deep blue and hot red colors are both converted to black.

Try to make your figures color-coordinated. If you show activity for conditions A and B using the colors green and black, use those same colors for the same conditions in all figures. This will help readers compare results across analyses.

If you cannot use colors, try to use different shades of gray and change the color thickness and type (solid, dashed, or dotted). If you need to plot many lines on the same grayscale figure, use different shades of gray, such as 100%, 66%, and 33% black saturation rather than 90%, 80%, and 70%. The same advice holds for line thickness: use 0.5- and 2.0-point lines rather than 0.75- and 1.0-point lines (figure 36.5).

Most people use the color map “jet” for their analyses. This is the color map that goes from blue for large negative values to red for large positive values and green for zero values. Because this color map is so widely used and easily interpreted, you should use it as well, particularly for results that can take positive or negative values. If you are plotting a variable that cannot be negative, such as ITPC or nontransformed Granger results, you should avoid using “jet” or any other bipolar color scale. The reason is that most people expect blue colors in figures to correspond to something negative, and thus, seeing ITPC

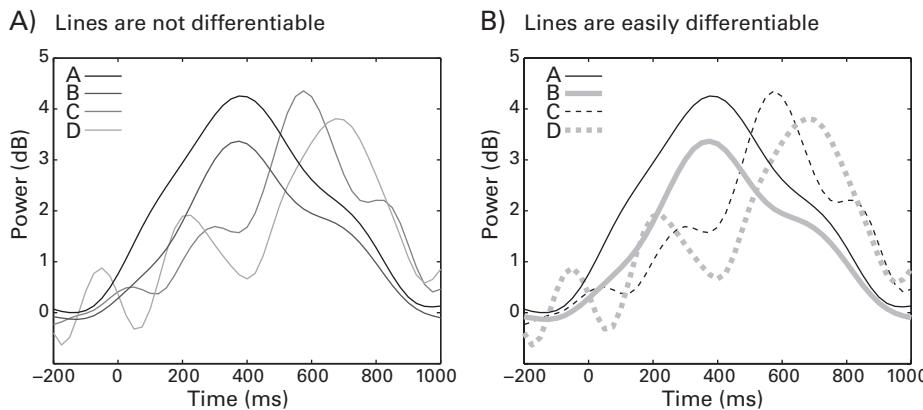


Figure 36.5

Poor use of grayscale and line styles makes it difficult to identify which time courses belong to which conditions. In panel A, grayscale saturation levels of 100%, 80%, 60%, and 40% were used for conditions A, B, C, and D. Panel B illustrates a better method for differentiating lines. In this case the four conditions are differentiated based on line thickness, solid versus dashed lines, and grayscale saturation. Even readers with poor-quality printers will see which lines correspond to which conditions.

corresponding to “negative colors” can be confusing (it would make sense, for example, in a figure with ITPC condition differences). There are several unipolar color scales, including “hot,” “bone,” and “grayscale.” You can also define your own color scales in Matlab.

Some journals will publish color figures for free in the online version and use grayscale for the print version. Very few people walk to the university library to look at physically printed and bound journal articles, so having grayscale figures in the print version of a publication should not be a major drawback.

36.6 Recommendation 6: Use Informative Figure Labels and Captions

Some figures are surprisingly difficult to interpret. To be sure, the majority of figures are at least reasonably well documented, but it is also not rare to see a publication with figures for which you cannot determine from which electrode activity is being plotted, or what conditions, time points, or frequencies.

Ideally, the information presented in the figure should be self-evident from the figure, legend, and captions. This is particularly the case if you want to attract potential readers to your paper. If the figure legend simply reads “Figure 3. See text,” many people will not follow that terse instruction but will doom the paper to sit forever in the “to-read” pile (everyone has one of those). The first line of the figure legend should be a succinct description of the main message of that figure (as discussed in section 36.1), and ideally, this should be understandable even to people who have not read the paper. For example: “Figure 3. Connectivity degree, a measure of large-scale network connectivity, increased over parietal areas after food-deprived subjects were shown pictures of grapefruits.” If you use abbreviations in the figures, define them in the legend even if they are also defined in the main text, unless they are commonly used abbreviations such as ms, Hz, cm, or dB.

Label axes with meaningful information, including the scale of the values [e.g., dB, percentage change, time (ms), frequencies (Hz)]. Show a color bar and label both ends of the color bar with numbers. “Min” and “max” colorbar labels are self-evident and meaningless and should not be used. If the units are arbitrary, write “arbitrary units” or “arb. units.” Figure 36.6 (plate 28) illustrates how useful informative labeling can be.

Figures published in papers are almost always smaller than you think they will be while you are making the figures on your computer. Therefore, you should use a large font size. The text should be as large as it reasonably can be without losing spatial selectivity or becoming bigger than the images. If the font size seems a bit small, it is definitely too small; if the font size seems right, it’s too small; and if the font size seems too big, it’s just right (see figure 36.7). Also consider that if someone else is thinking about using your figure in their talk or

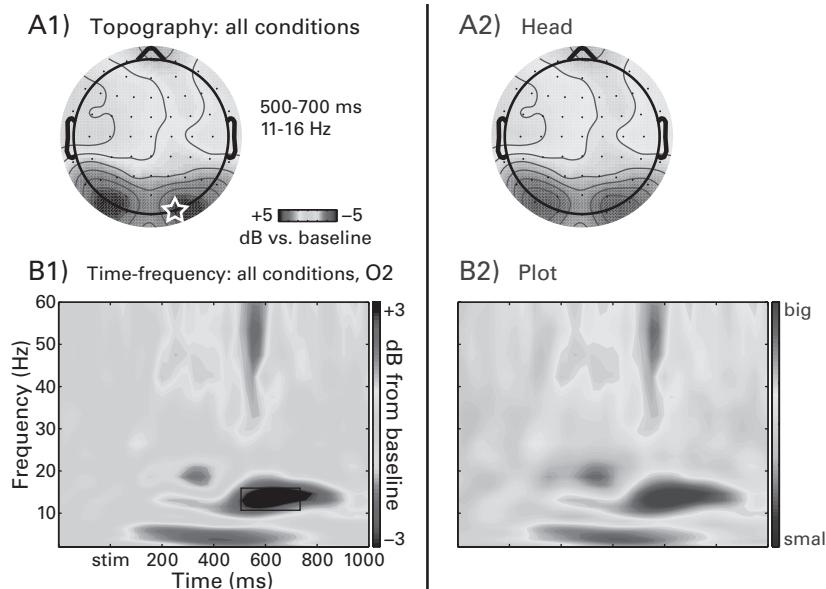


Figure 36.6 (plate 28)

Time-frequency and topographical plots that have informative (panels A1 and B1) and uninformative (panels A2 and B2) captions accompanying the plots. It is difficult or impossible to extract information from the plots in panels A2 and B2.

lecture, your figure will be pixilated and expanded to many times its original size. If the font is barely visible in the paper, it will render simply as a gray box in someone's PowerPoint slide show.

36.7 Recommendation 7: Avoid Showing “Representative” Data

Sometimes you see a figure showing what the authors claim to be representative data—a small amount of data, typically one or a few trials from one subject, one electrode, or one neuron.

There are two reasons why you should not show “representative data” unless you are illustrating how an analysis works. First, critical readers will be suspicious of representative data no matter what those data show. If the representative data look similar to the group result, readers will suspect that you hand-picked that example. If the representative data look different from the group result, readers will wonder why you picked that example, and they may question the robustness and validity of the group result. Second, the reason you collect

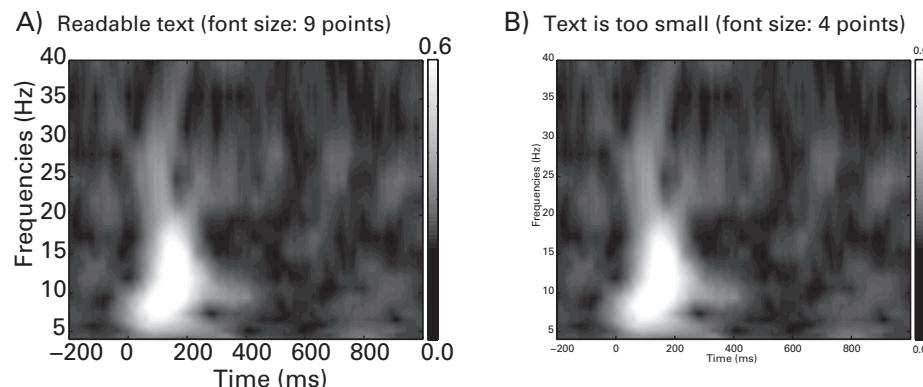


Figure 36.7

Text inside figures can easily become too small to be legible. When zoomed in to 800% on my 27-inch desktop monitor, the text labels were easily readable in both panels A and B. However, you should not assume that your readers will have the same comfortable viewing environment. When this figure is printed the text may become unreadable, and when a screen shot of this figure is taken and projected on a wall, the text in panel B will likely seem like small gray dots. Some journals make explicit recommendations for minimum font sizes, but in general, avoid using font sizes smaller than 8 points when possible.

data from many trials and from many subjects is that single trials and single subjects are not representative of the population response. In other words, critical readers will be cautious about making any inferences based on the results in a figure that shows representative data.

This does not mean that you should avoid showing single-subject data. Showing single-subject data can make a compelling case for the robustness of an effect, or it can highlight the individual variability in the results. If there are few subjects it is arguably better to show single-subject data than group-averaged results. For example, in intracranial EEG studies, each of five patients in a dataset might have electrodes placed in different brain regions. In this case you can and should show data from each subject. Furthermore, if you want to illustrate that you have clean data and can observe the effect in single subjects, show the single-subject results from all subjects (in a supplemental materials section if necessary). The argument here is that you should not show data from one subject when you collected data from 20 subjects.

The exception to this recommendation—the situation in which you should show “representative” data—is in a figure in which you are illustrating how a method works (which was done throughout this book, for example) or showing the temporal structure of the experiment. If this is the goal, do not pick data randomly; hand-pick a nice-looking

piece of data that you think clearly illustrates the point. Make sure you state clearly in the figure legend that those data were selected for illustration purposes. That way, readers know that they should not interpret the result in the figure but, rather, understand the method that those data highlight. For example, data were shown in the figures in this book to illustrate how different methods work; you should not draw firm conclusions about fundamental brain mechanisms based on the results here because all of the figures are based on one subject (and in many cases, one trial).

36.8 A Checklist for Making Figures

Following is a list of questions that you can ask yourself to make sure your figures will be interpretable. Perhaps you have additional questions to add to this list, or perhaps some of these questions are not relevant for your figures. However, asking yourself these questions will help you make clear and readable figures.

- What is the main message of the figure?
- Does the figure convey one idea, several related ideas, or several different ideas?
- Can you easily discern which lines/bars/plots correspond to which conditions/groups/analyses?
- Can you easily distinguish the significant results from the nonsignificant results?
- Is the color helpful or annoying?
- Will this be readable when printed on a grayscale printer?
- Can the figure be interpreted on its own, or do you need to read the text to understand it?

36.9 Tables

Tables of activation are common in fMRI research. The tables typically list spatial XYZ coordinates for peak activations, a measure of effect size or statistical significance such as a *t*-value or a *p*-value, and a description of the effect that revealed the activations (e.g., condition A > condition B). Tables in fMRI papers are so common that automated programs can search journal websites and read data directly from tables of activation to use for meta-analyses (Yarkoni et al. 2011).

Tables of activation are uncommon in cognitive electrophysiology, particularly for time-frequency results. This likely reflects that most cognitive electrophysiology research is done in one of two ways: the research is hypothesis driven, and only the analyses that were specified *a priori* are tested and reported in the Results section; or the research is data driven and

exploratory, and suprathreshold results are shown in figures and discussed in terms of their qualitative patterns. Either way, researchers rarely organize the information into easily readable tables.

The lack of tables is understandable—time-frequency-based analyses create a large search space, and it is impractical to search through the entire space for significant results to put in a table. For example, multiple-comparisons correction is only sometimes done over time points, frequencies, and electrodes simultaneously; more often, one or a few electrodes are selected, and then pixels in the time-frequency map are tested using multiple-comparisons correction, or one time window is selected and all electrodes are tested with correction for multiple comparisons.

However, tables of activation are useful for organizing results and will allow readers to inspect other findings that may have been statistically significant but not focused on in the paper. Furthermore, tables of activation will facilitate future meta-analyses. Tables should be included in cognitive electrophysiology publications. At a minimum, the following columns should be included:

1. The type of analysis (ERP, time-frequency power, ITPC, phase-amplitude coupling, etc.)
2. The statistical contrast (all conditions vs. baseline, conditions A–B, adolescents–adults, etc.)
3. The electrode label or XYZ coordinate (can be the center electrode if several electrodes were pooled)
4. The time point at which the effect was maximal or the time window used in the analysis
5. The frequency band at which the effect was maximal or the frequency band used in the analysis
6. A measure of effect size (dB, percentage change, $F/t/z$ -statistic, etc.)

36.10 Reporting Results in the Results Section

Because time-frequency-based results can be complicated, and because it is common to perform many analyses (e.g., power, phase, connectivity), try to write the Results section as if you are guiding the reader along a path through the landscape of the results. This is particularly important if your study concerns a topic for which time-frequency-based analyses are infrequently applied, and thus readers may be unfamiliar with interpreting time-frequency-based results.

This is no trivial matter. On the one hand, you want to be open and forthcoming about the results without hiding data, or seeming to readers as if you are hiding data. On the other hand, there are simply too many results to show everything without drowning the reader in

a confusing ocean of numbers, colors, and contour lines. Having clear and specific hypotheses will help you decide which results should be shown and discussed, and they will help you justify your decisions of showing some regions of the data hypercube and not other regions.

Organize the presentation of the results in a sensible way. The results could be organized according to hypotheses, or according to the time period during the task (e.g., stimulus, delay, response, feedback), or according to the type of analysis (e.g., ERP, time-frequency power, phase-based connectivity). While working on the manuscript, do not be afraid to reorganize the Results section if you think a different organization scheme might be more sensible.

The organization of the Results section also depends in part on how hypothesis driven versus exploratory the study is. The Results section of hypothesis-driven research is likely to contain statistical values, degrees of freedom, *p*-values, and statements concerning main effects and interactions. The Results section of an exploratory study, in contrast, will contain few statistical values and instead contain qualitative descriptions of the patterns of the results that exceeded a statistical significance threshold.

Related to the organization and as mentioned above in section 36.2, it is informative for readers if you describe the overall characteristics of the results first, before moving on to the condition-specific effects and hypothesis-driven analyses. These descriptions can be qualitative and need not be accompanied by statistical tests unless that is relevant. For example: “When we examined the main task effect (all conditions averaged together compared to baseline), we observed increases in theta-band power over midfrontal electrodes from around 200–800 ms post-stimulus-onset, alpha-band suppression over occipitoparietal areas around 500–2000 ms, and beta-band suppression over motor areas around 1000–1700 ms.” These general descriptions will help readers form a framework of cortical dynamics within which to place the condition-specific results.

Introduce each set of analyses to help the reader understand what results they are about to read. This is particularly important for papers with many complex sets of analyses. For example, you can start each paragraph or subsection of the Results with “the purpose of the next set of analyses was to examine whether...”

Relatedly, having a closing sentence will also help readers consolidate what they just read. For example, “Taken together, these statistical tests generally confirm our hypothesis that food-deprived subjects exhibit enhanced local connectivity when viewing pictures of grapefruits, although, in contrast to our expectations, the results were observed primarily over parietal areas instead of occipital areas.”

Avoid interpreting the results or speculating about the meaning of the results when reporting condition comparisons and statistical effects. This is the purpose of the Discussion

section; the Results section should ideally be constrained to an objective description of the significant and, when relevant, nonsignificant results. In some cases it is necessary to interpret results, point out that a (perhaps unexpected) finding is similar to a finding published previously, or motivate a post hoc analysis that is based on an interpretation of a result. But statements about interpretations of what the results might mean should be used sparingly in the Results section.

A final matter discussed here with regard to the Results section is reporting *p*-values. You might think this is an easy issue: report *p*-values associated with statistical tests. For hypothesis testing, this is good advice: you should report the *p*-value associated with each statistical test. Furthermore, you should report the *p*-value itself rather than simply noting whether the *p*-value was less than 0.05. This is because, for example, the *p*-values 0.048 and 0.0000048 are both statistically significant using a threshold of 0.05 but might be interpreted differently with respect to how strongly the results support the hypotheses.

On the other hand, if your study involves exploratory data analyses, there may be no theoretical framework within which to compare qualitatively findings associated with *p*-values of 0.048 and 0.0000048. Therefore, you could report no *p*-values at all but instead show the results in figures such that suprathreshold pixels or electrodes are outlined or otherwise demarcated and then describe qualitatively the patterns of significant results. Indeed, if mapwise results are thresholded at a specific *p*-value, reporting *p*-values associated with the results would be biased because the pixels would be selected based on being statistically significant. In the case of selecting data based on a significant result, it would be impossible to report a *p*-value that you do not already know exceeds the significance threshold, and therefore, reporting *p*-values might be circular inference.

VII Conclusions and Future Directions

37 Recurring Themes in This Book and Some Personal Advice

There are several themes in this book that permeate the text, figures, and analyses. Some of these themes were repeatedly made explicit; others were implicitly suggested. This chapter reiterates and expands on these themes. I also give some personal advice about analyzing cognitive electrophysiology data. The advice is based on my personal experiences, and you are free to disagree.

37.1 Theme: Myriad Possible Analyses

There seems to be a limitless parameter/analysis space within which to analyze data. This can be intimidating to newcomers in the field because they do not know which analyses should be used in which circumstances. And this may be fodder to skeptics unfamiliar with the analyses, who will claim that any finding desired can be obtained by “torturing the data” enough (although I doubt this is true).

If you are unsure which analyses to perform, follow the analysis protocol of a published paper on a similar topic. If there are no relevant publications, or if you think the published analysis protocols are not optimal for your study, start with the simple analyses and apply more complex analyses only when necessary.

Analyses are like tools in a toolbox: you should pick the right one(s) for the job. Even your favorite method might not always be appropriate, much as a sledgehammer might not make the best fly swatter. Whether an analysis is appropriate depends almost entirely on the research question. Data analyses should be as sophisticated as they need to be to answer the question, and you should avoid using complex methods when simpler methods will answer the question equally well. If your preference is to pick a method and then find a problem to which to apply that method, you should try to find an appropriate research question for which that method is particularly well suited.

37.2 Advice: Avoid the Paralysis of Analysis

Analyzing data is fun and satisfying (for some people). It provides both instant and long-term gratification and gives a sense of achievement and closure when you solve a data analysis problem. But for many cognitive electrophysiologists, particularly those toward the “cognitive” end of the spectrum of figure 1.1, analyzing data should be the means to the end, not the end in itself. The purpose of analyzing EEG data should be to extract meaningful patterns that are embedded in the EEG signal but are difficult to observe in the raw signal, and the result of data analyses should be empirical support for or against a statement concerning some aspect of brain function.

The potential problem with data analyses is that the search space of electrophysiology data is so huge that there are limitless possibilities for new or different ways to analyze the same data. Thus, there is a risk of ceaselessly analyzing and reanalyzing your data without getting any new information from the data. If your head is in the data-analysis clouds, try to keep your feet on the purpose-of-the-study ground.

On the one hand, it is good to be cautious and check your analyses, and it is good to confirm a result using alternative analysis methods. But on the other hand, at some point new analyses will take more of your time and resources without giving you anything back in return. At some point, you need to accept what you’ve done with that dataset and move on.

37.3 Theme: You Don’t Have to Program Your Own Analyses, but You Should Know How Analyses Work

Programming your own analyses increases your freedom and flexibility. It allows you to have more control over what analyses you can perform, and it allows you to custom-tailor analyses to the idiosyncrasies of your experiment design and your data, and to select the most appropriate analysis parameters for your hypotheses. It also helps you interpret the results more appropriately because you understand at a mathematical level what happened to the data.

There is nothing wrong with using existing and user-friendly analysis packages or running Matlab scripts that someone else wrote. It is important to understand at a conceptual and mathematical level what happens to your data between pressing the “record” button during data collection and making publication-quality figures. Without an understanding of at least the basics of how the analyses work, there is a danger of using inappropriate analysis parameters, misinterpreting results, or doing something wrong without even knowing it.

You should try to understand at least conceptually what happens to your data, the advantages and disadvantages of each analysis method, the key parameters and what effects those parameters have on the results, when it is appropriate and inappropriate to use that analysis, and how the results can be interpreted. This level of understanding is not the same as—and is more important than—being able to write out all the equations on a blank piece of paper or being able to write Matlab code from scratch to do the analyses.

37.4 Advice: If It Feels Wrong, It Probably Is

Another major benefit of understanding how analyses work is that you develop an intuition for when a result doesn't "feel right." This intuition builds with experience as well as mathematical knowledge, but by understanding the mathematical bases of analyses and how they are implemented, you will develop a kind of "sixth sense" about when something went wrong or when there are artifacts in the data. This is no metaphysical awareness but rather a set of mental templates for what analysis results should look like. Furthermore, when something goes awry, by understanding how analyses work, you will have some good guesses about what the problem might be and how to solve it.

37.5 Advice: When in Doubt, Plot It Out

A picture is worth a thousand words. If you are not sure what a particular Matlab function does, plot the data before and after applying that function to the data. This will help you develop an intuitive feeling of what the function does to the data. The Matlab commands `plot`, `hist`, and `imagesc` are the most useful functions for understanding what happens to your data. Furthermore, if you suspect that something is wrong with your results or with the analyses, plot the data at each step of the analysis until you see a problem. NaNs and Infs will produce empty spaces in figures. Plotting and looking at data are the best ways to understand how Matlab functions work, how your data are transformed, what mathematical equations mean, and what may have gone wrong with the data or with the analyses.

One advantage of software programs such as Excel and SPSS is that the data are all on the screen in front of you. You cannot avoid seeing the raw data. Thus, a disadvantage of software programs like Matlab and R is that the data are hidden, in that you won't see them unless you make a specific effort to look at them. This distinction between being visually confronted with your data and having to make an effort to see the data is akin to a distinction between bottom-up and top-down processing. If there is an outlier in, for example, reaction time data, you are likely to notice this outlier via bottom-up visual capture if your

data are in Excel, but you will need to exert top-down control to notice this outlier in Matlab (e.g., by plotting the reaction times of all trials). Most or perhaps all EEG analysis packages have visualization routines that allow you to look at and scroll through your raw data as well as results of the analyses. The importance of looking at raw and processed data cannot be understated, both in terms of making sure the data are clean and in terms of understanding the final results. Thus, try to employ top-down data inspection strategies because the bottom-up processes might be insufficient.

37.6 Advice: Know These Three Formulas like the Back of Your Hand

There are three equations that form the backbone of the majority of EEG time-frequency-based analyses. If you memorize these equations, you will be able to understand most time-frequency analyses. You should be so familiar with these equations that you can recite them at any time of day, in any mood you happen to be in, or in nearly any state of consciousness.

Sine wave: $A \sin(2\pi ft + \theta)$

Euler's formula: $Me^{ik} = M[\cos(k) + i \sin(k)]$

Gaussian: $e^{-t^2/2s^2}$

If you become very familiar with these equations, you will be able to see them embedded in seemingly complex equations that otherwise might be too difficult to understand. Here are a few examples: the discrete time Fourier transform is a combination of a sine wave and Euler's formula; a complex Morlet wavelet is a combination of all three formulas; wITPC and phase-amplitude coupling are Euler's formula. Obviously, these are not the only three equations that govern all EEG time-frequency-based analyses, but these are the main ones.

37.7 Theme: Connectivity over Trials or over Time

Many connectivity analyses can be done over time or over trials. Although the math is often very similar, the interpretations and the kinds of results you might obtain can be different. Connectivity over time increases sensitivity to simultaneous coupling and is better for high-frequency connectivity that might not be phase-locked to an experiment event. On the other hand, it has increased susceptibility to volume conduction, has relatively poor temporal precision, and may reflect tonic connectivity in addition to task-related connectivity. Connectivity over trials provides a better link to the timing of task events and has higher temporal precision. On the other hand, it can be used only with task data, is more affected by

having a small number of trials, and is less sensitive to detecting high-frequency connectivity that might not be phase-locked to the time = 0 event. Unfortunately, many publications do not specify whether their measure of connectivity was performed over time or over trials. Try to be clear about this when describing your analyses.

37.8 Theme: Most Analysis Parameters Introduce Bias

Nearly all analysis parameters introduce some kind of systematic bias into the results. Some of these biases are related to the trade-off between frequency and temporal precisions, and some biases are related to the kinds of findings you will observe or be blind to (e.g., phase-lag-based vs. phase-clustering-based connectivity measures or phase-locked vs. non-phase-locked activity). How concerned you need to be about these biases depends on the kind of bias and on the effects you are testing. In some cases, in comparing results across conditions, electrodes, time segments, and so on, the effects of biases are diminished or eliminated because the biases affect all results and are therefore minimized during subtraction or statistical comparison. This is why the same set of parameters should be applied to all data. For resting-state data or task data in which only one condition is examined, you should be more careful about interpreting the results in light of any biases that may have been introduced by analysis parameters. If you are using a new analysis or are unsure which parameter settings to use, you could perform the analysis several times using different parameter settings. If you do this, report in the publication what ranges of parameters you tried and whether that affected the patterns of results. This will not only help readers evaluate your results, it will also help other researchers, who are in a similar position as you are, to know which analysis parameters to use.

37.9 Theme: Write a Clear Methods Section so Others Can Replicate Your Analyses

This should be an obvious point, but it is worth stressing: write your Methods section clearly so that other scientists will be able to replicate your analyses. That does not necessarily mean that you need to write every minor detail, such as the formula for a Fourier transform. But there should be enough detail that someone with a reasonable amount of knowledge about analyzing EEG data will be able to reproduce your analysis protocol. If you use toolboxes, state which functions or methods were used and note any nondefault parameters. If you wrote your own code for the analyses, consider making the code publically available on a website, as an appendix or supplemental online section in the publication, or as an extension to an analysis toolbox such as eeglab or fieldtrip. If the analysis protocol was complex,

write a step-by-step list of what was done. When in doubt, have a colleague read the Methods section and decide whether he or she feels the method could be replicated. It is better to include too much than too little information, particularly for journals that either do not have strict word limits or that allow supplemental online material. If the method is novel, difficult, or involves many steps, consider making a figure to illustrate the procedure for how the analysis was conducted. There are recently published suggestions for what details to report in Methods sections of studies using MEG (Gross et al. 2013). That paper was written specifically with MEG studies in mind, but some of the suggestions are also applicable to EEG and LFP studies.

Be open and honest about the limitations of your methods and analyses that are relevant to the results. No method is perfect, and limitations are nearly always balanced with advantages. For example, the surface Laplacian increases spatial precision and allows connectivity results to be better interpreted at the electrode level, although this comes at the expense of filtering out some low-spatial-frequency features of the data. Not all limitations of a method are relevant to your study. For example, if your study concerns parietal alpha oscillations and attention, the spatial precision of EEG might not be a relevant limitation. However, if your experiment design is modeled after an fMRI study that suggests an important role for the thalamus, the spatial precision of EEG might relevant to mention. Another example is that time-frequency decomposition methods decrease temporal precision, but this may not be relevant unless your hypotheses involve testing for minor differences in timing between conditions.

37.10 Theme: Use Descriptive and Appropriate Analysis Terms

I thought carefully about the analysis terms used in this book and decided to use terms that are brief descriptions of the mathematical aspects of the method, rather than terms that are interpretations of the results. Some terms were avoided because they are ambiguous (for example, “spectral perturbation” can mean a change in power, phase, connectivity, band-specific graph properties, or other spectral features).

One could argue that it doesn’t matter what the method is called as long as the formulas are written and the procedure is clearly described. This rose-by-any-other-name argument is understandable, and it certainly is the case that the results and what they mean for brain function are important, not the short name given to a set of equations. But from a practical standpoint, using different terms in different papers to refer to the same analysis hinders efficient evaluation of results and also hinders cross-study comparisons. Ideally, when someone says or writes the name of a method, you should know exactly what method they

mean. In other words, there should ideally be a one-to-one mapping between the name of a method and the math behind that method. This is often the case in other branches of statistics. Consider the following terms: correlation, ANOVA, factor analysis, structural equation model. There is no ambiguity about which sets of equations are employed when someone says that he performed an ANOVA. Cognitive electrophysiology should be this exact, but it is not. If someone says that there was an increase in alpha synchronization, you don't know whether that person means an increase in power or phase-based connectivity, and these two analyses have very different interpretations, putative neurophysiological origins, theoretical implications, and methodological concerns.

The point is not to come up with perfect terms; at the end of the day, terms are merely words, and their connotations are influenced by semantic, cultural, linguistic features and personal preferences. I think Lewis Carroll best expressed the idea that there is an arbitrary relationship between words and their real-world referents (specifically through the character Humpty Dumpty in *Through the Looking Glass*). My point here is not that we should struggle to find the most linguistically perfect expressions for sets of mathematical equations that are applied to data, but rather, that there should be widespread agreement that a particular term implies a particular analysis. For example, you might think that the term "receiver operating characteristic" is a poor description of that analysis, but when you use that term (or its abbreviation ROC) to an audience with a statistics, psychology, or engineering background, it is very likely that they will know to which analysis you refer.

In my opinion—and others may disagree—the terms used to describe analyses should refer to the mathematical procedures that underlie those analyses, not to an interpretation of what the results of that analysis might mean in a neurophysiological sense. Interpretations of results are subjective, incite collegial disagreement, change over time as theories develop and new data are acquired, and can be incorrect or misguided. Mathematical procedures, on the other hand, are simply equations and data transformations and will remain the same regardless of what contemporary scientists believe the results of those analyses imply about brain function.

37.11 Advice: Interpret Null Results Cautiously

Null results can be difficult to interpret in any branch of science. There are many reasons why you might obtain a null result, including poor experiment design, low-quality data, too few trials, inappropriate analyses, and others. In this sense, null results from EEG analyses can arise for the same reasons that null results can arise in, for example, a psychology questionnaire study.

However, there are additional considerations for null results with EEG data, in part because EEG measures only a fraction of all brain activity and in part because, despite measuring only a fraction of brain activity, the EEG signal is multidimensional. You can obtain a null result simply because you looked at the wrong feature of the data, even if the experiment, data collection, and analyses were properly conducted.

For example, if a time-frequency plot of baseline-corrected power is all green (values near zero) in the theta band, it is inappropriate to conclude, “There was no theta activity in the brain.” There could have been neurons firing in the theta band, there could have been theta activity that was spatially asynchronous so as not to generate an electrical field large enough to be measured by the electrode, there could have been deep theta generators that were filtered out during surface Laplacian, there could have been task-related theta activity in phase but not in power, and so on.

The appropriate interpretation of null results should be close to the analysis and should be anchored with a significant result if possible, which suggests that there was sufficient statistical power to detect a result if it were present. For example, “Theta power was not different between conditions A and B, whereas it was significantly greater in condition B compared to condition C.”

It is also important to consider the situations in which a null effect would be observed with each particular analysis. For example, an absence of a change in connectivity as measured by phase-lag-based measures can indicate that there was no connectivity or that the connectivity had a near-zero- or $-\pi$ phase lag. If an ERP is nonsignificantly different between two conditions, this may indicate that the same neurocognitive process was employed in those two conditions, or it could be that the neurocognitive processes that differed between the two conditions was non-phase-locked and therefore not visible in the ERP.

You can be more confident about interpreting null results if you can demonstrate (1) that the data are of sufficient quality and that there is sufficient statistical power to detect a result if it were present, and (2) that the analyses applied are appropriate to address the research question. Finally, if the interpretation of the null result is important for interpreting other results, show that null result in a figure in addition to reporting statistical *p*-values. That way, readers can judge whether the null result is going in a particular direction and could be significant if there were more or cleaner data, or whether the result is a true null effect and unlikely to be significant even if more and higher-signal-to-noise data were obtained.

37.12 Advice: Try Simulations but Also Trust Real Data

The obvious advantage of simulations is that there is a ground truth that you define, and you know whether the analysis worked because you can compare the results to the simulated

data. Simulations are also useful to illustrate the effects of extreme data on analyses because extreme data might be difficult to find in real data.

However, simulated data often do not look or behave like real data, particularly if the simulations do not contain realistic amounts of noise with realistic time-frequency characteristics. Therefore, if you are trying to understand how a method works, or if you are extending existing methods, you should use both simulated data and real data. If possible, use data from an often-used task that produces replicable results so this pattern can be considered an empirical ground truth.

37.13 Advice: Trust Replications

No dataset is perfect, and all datasets have idiosyncrasies and quirks. These may be due to subject sampling error or to experiment design features. One can make a distinction between a statistical false alarm that arose because of, for example, a lenient statistical threshold, and a result that is true in one sample but may not generalize to another sample. Replications are important because they demonstrate the generalizability of an effect beyond what evidence can be obtained from a statistical measure in one dataset. Of course, this does not mean that findings reported in only one study should be ignored or not trusted. Rather, findings should be taken more seriously after they have been replicated in several different experiments, ideally using different experiment designs and analysis approaches, and by different research groups.

This is another reason why using clear analysis terms and publishing tables of activation will facilitate identifying replicable effects across different research groups and experiments.

37.14 Theme: Analyses Are Not Right or Wrong; They Are Appropriate or Inappropriate

This theme was made explicit in most chapters, with such wording as *should you <insert question>? It depends on the goal of your study. . . .* If you have nothing better to do on a Saturday night, you can probably think of some data analyses that are completely wrong in all situations. But within the realm of reasonable analyses that a scientist would consider applying to his or her data, whether a particular analysis method should or should not be used depends on the goals of the study, the experiment design, the quality of the data, and the other analyses planned. Analyses that are appropriate in some situations might be inappropriate in other situations. For example, if you have a hypothesis about condition differences in phase-based connectivity between one frontal electrode and one parietal electrode, ISPC is a more appropriate measure than phase-lag-based measures because ISPC is more sensitive to detecting connectivity and condition differences, and the result can be inspected for possible

contamination of volume conduction. On the other hand, if you are examining whether individuals with autism have increases in short-range connectivity compared to matched controls, phase-lag-based measures are more appropriate than ISPC because some of the ISPC results may be contaminated by volume conduction, and it is impractical to examine each of hundreds or thousands of connectivity results for potential artifacts.

This is also why clear writing in the Methods section is important. It is possible that your analysis protocol is justifiable and appropriate, but because the description of your analysis protocol was not clear enough, reviewers and readers suspect that it was not done appropriately or, worse, that you intentionally biased the data selection procedure to obtain a specific result. For example, consider the following passage of an imaginary Methods section:

In the first step of the analysis, we averaged time-frequency power from all conditions and selected a time-frequency window in the gamma band, based on a statistical comparison between time-frequency power over all conditions versus the prestimulus baseline. Note that because this selection procedure was based on the condition average, it is orthogonal to any possible condition differences. In the second step, we extracted power from this time-frequency region separately for each condition and each subject. These values were then entered into a 2 (visibility: low vs. high) \times 3 (feedback condition: punishment, neutral, reward) ANOVA.

Now consider how this same procedure could lead readers to become suspicious because of unclear writing: “Condition differences in gamma power were tested where there were significant gamma power effects.”

There are useful guidelines for knowing when certain methods are appropriate or inappropriate, which were (I hope) made clear in this book when each method was introduced and discussed. Ultimately, however, it is up to you to make that decision and justify it. If you are unsure whether a particular method is appropriate in your situation, ask colleagues or search for how that method was used in peer-reviewed publications. Keep in mind, however, that the published application of a particular method does not necessarily mean it was appropriately applied.

37.15 Advice: Hypothesis Testing Is Good/Bad, and So Is Data-Driven Exploration

Strict hypothesis testing is good: it is theory driven, has implications for theories and models, will likely help you design better experiments, and increases statistical sensitivity because it minimizes multiple-comparisons problems. Hypothesis-driven analyses also tend not to be very time-consuming because only a small number of analyses need to be performed. On the other hand, strict hypothesis testing is constraining and analogous to horses that wear blinders so they don’t get distracted by their peripheral vision. Perhaps the most interesting

and most insightful results in the data are not the ones predicted by the limiting and probably incorrect theory (all theories are wrong . . .).

Data-driven exploration is also good: it offers freedom to discover patterns of results, provides an avenue for the more number-crunching-oriented scientists to expand and develop new methods, and it facilitates new discoveries unconstrained by the blinders of theories. Data-driven exploratory studies tend to be time-consuming, in part because there is so much multidimensional data to inspect and in part because many analyses are often performed as the researcher thinks “perhaps the best result is still hidden and will be revealed by a fancier analysis.” On the other hand, data-driven exploration can be like fishing in a polluted lake (you might catch a fish or you might catch a plastic bag), requires appropriate correction for multiple comparisons that might push the most interesting and important results below the statistical threshold, and may provide results that are uninterpretable and are consigned to the basement of science because the finding is not relevant to any theory, model, or other research finding.

There is no correct position to take on the issue of hypothesis testing versus data exploration, and there is no optimal balance between them. The appropriate levels of hypothesis testing and data exploration should be based on your preferences, data analysis skills, patience, and orientation toward theory versus data. They are also based on the field in which you are working, including how much research has already been done in that area, what previous findings have shown, and whether there are theories detailed enough to make relevant predictions that can be confirmed or disconfirmed. In many cases simply too little is known about the electrophysiological dynamics of brain function and its relation to behavior, brain structure, and the body to perform only hypothesis-testing analyses.

These two approaches are not mutually incompatible. You can design an experiment to test specific hypotheses and then perform additional data-driven analyses to explore what else might be lurking in the data, waiting to be discovered. The other way around is possible as well, though less straightforward. If you design an experiment for data exploration, it is inappropriate to test specific hypotheses on the data after seeing the results, but it is appropriate (and good science) to design a new experiment to confirm the findings you observed in the exploratory analyses.

37.16 Advice: Find Something That Drives You and Study It

Science is hard. It requires time, energy, patience, perseverance, and self-discipline. For many scientists, science is not just a job, not just something you do during the day to earn money. Instead, it is a career, a passion, and a lifestyle. Whether you love or hate your life as a

scientist will affect your happiness and life satisfaction. Pick a research topic that fascinates you and a method that draws you. And keep long-term goals in mind. You won't love every single minute of being a scientist, but very often, the parts you don't like will help get you to the parts you do like. This advice is not just about preserving your sanity: it is also about becoming a better scientist. If you are uninterested in your research topic, you will probably put in the minimum required effort to turn it into a publication or conference poster presentation that others may not read past the abstract. If you are fascinated by your research topic, you will be motivated to do high-quality, creative, and progressive scientific research.

If you need some inspiration for scientists persevering—and ultimately triumphing—in the face of repeated rejections, setbacks, and constraints, consider reading the first chapter of *Roving Mars: Spirit, Opportunity, and the Exploration of the Red Planet*, by Steve Squyres (2005), who was the principal investigator of the mission to put mechanical rovers on the planet Mars.

37.17 Cognitive Electrophysiology: The Art of Finding Anthills on Mountains

Perhaps in an ideal world (at least, the cognitive electrophysiologist's ideal world), this book would not be necessary. Data analysis toolboxes and software programs would also not be necessary. You could simply put a few electrodes on one subject's head, test a few trials per condition, look at the raw EEG data with no processing or analyses, and understand the complex spatial-temporal-spectral landscape of cortical electrophysiological dynamics. There would be no noise, confounds, or alternative explanations to rule out, and there would be no sophisticated analyses, statistics, or probabilities to compute and base speculative inferences on. You would just look at the data and understand how the system works.

Regardless of whether your ideal world includes performing EEG research and looking at data, this is not the world we have. EEG signals are noisy, findings are sometimes hidden in dimensions that are difficult to visualize without filters or other data transformations, results can be infected with artifacts or may be statistical false alarms, and interpretations are usually based on probabilities, inferences, and speculations.

Good science—in addition to being based on theory, previous research, careful experiment design, and appropriate data analyses and statistics—is also about discovering and appreciating subtlety. And the appreciation of subtlety becomes more important with time as basic findings become established fact. This is good; it reflects progress. For example, you can no longer publish a paper in which you demonstrate that the human occipital lobe is involved in vision. Instead, you have to find more subtle and interesting features of the computations performed in visual processing areas and the neural implementations of those

computations. Often, the biggest and most obvious results are not the most interesting and insightful results. That doesn't mean you should ignore the obvious results—they are often important for data quality demonstrations and for characterizing the overall landscape of cortical electrical dynamics. But in many situations the big obvious effects are not the ones that provide novel insights into neurocognitive function.

Discovering a subtlety in the brain seems as though it should be easy, but it is not. Subtleties can be hidden in noise or hidden under a much more robust but less interesting result. You might have almost discovered a subtlety but had the wrong condition comparison, frequency band, or time window. Discovering subtleties involves considering theories, previous research, and openness to letting the data "speak for themselves." Science, or at least cognitive electrophysiology, can sometimes be more of an art than a science.

38 The Future of Cognitive Electrophysiology

Scientists occasionally lose the forest for the trees. This is an occupational hazard—good science is done by delving deep into the details of each project, and sometimes it's hard to remember what the purpose of that project was in the first place. It is useful to take a step back and try to put your study in perspective, not only in the context of the wider literature but also in the context of the development of cognitive electrophysiology as a broad discipline. The purpose of this chapter is to offer some suggestions for what could be important developments within the field of cognitive electrophysiology. There is no single unifying theme; rather, this chapter lists a set of future developments that I think would help to propel cognitive electrophysiology in the twenty-first century.

38.1 Developments in Analysis Methods

Without methods development, we would still be counting ink zero-crossings on polygraph paper. Certainly there will be improvements over existing analysis methods, and there will likely be novel analysis methods that are more sophisticated and neurophysiologically interpretable than our current analysis methods. However, development of data analysis methods for their own sake may not incite real progress in understanding brain function. The reason is that, as discussed in section 37.1, the search space of the brain is so huge that there are a limitless number of ways to analyze EEG data. Furthermore, the present repertoire of existing analysis techniques is already quite large, arguably larger than the need to apply those methods. Although there are many publications each year detailing new data analysis techniques, most new methods are published as proof of principle; whether those new methods are better than existing methods is often not demonstrated.

This is not to say that methods development is not important or should not be pursued. One area of methods development that is underutilized but could provide a boost to progress in cognitive electrophysiology is multivariate analyses. It is naive to think that each cognitive

process is subserved by one brain region and one frequency band. It is easily demonstrated that there is no one-to-one mapping between frequency band and cognitive process—for one thing, there are around six frequency bands (delta, theta, alpha, beta, gamma, high gamma) but more than six cognitive processes. The brain must use complex and multidimensional information-processing schemes. And yet, most cognitive electrophysiology studies do not consider how the neural code is expressed through multidimensional interactions; instead, most cognitive electrophysiology studies are based on a mass-univariate statistical approach, which means that each time-frequency-electrode point is treated as a separate (though not necessarily independent) statistical test. Mass-univariate approaches may be overly simplistic (Cohen 2011b), but they are useful and powerful. There has been and will continue to be a wealth of knowledge gained about brain functional organization from mass-univariate analyses. But a better understanding of the neurophysiological mechanisms of cognition will require a more nuanced approach to conceptualizing information processing over a multi-dimensional (i.e., multivariate) space. Multivariate analyses are increasingly applied to fMRI datasets with much success. There is a slowly growing number of publications on multivariate analyses of EEG and MEG data, and there is much room for continued development.

At a practical level, multivariate analyses involve weighted combinations of activity over frequency, time, and/or space. In some sense basic multivariate analyses are already commonly used: temporal and spatial filters involve taking weighted sums of activity over time or over electrodes, and the result is a feature of the signal that may be difficult to observe without filtering. These “multivariate” analyses have revealed novel insights into brain function beyond the “temporal univariate” technique, event-related potentials. Better multivariate techniques should take into account features over time, space, and frequency (and possibly other dimensions such as power vs. phase, connectivity, or cross-frequency coupling), and weights would be set not only according to mathematical templates but also according to patterns in the data.

The difficult part of multivariate analyses, particularly in their early development, is knowing which dimensions and combinations of dimensions are most relevant. This aspect of multivariate analyses is easier with fMRI because fMRI-based multivariate analyses typically use only one dimension—space. Data-driven exploration seems an appealing approach for finding which combinations of which EEG data dimensions are most relevant, but there are too many possibilities for combining dimensions and usually too little data, and the appropriate corrections for multiple comparisons might be so stringent as to limit discovery of subtle but important patterns in the data. And without better theories to reduce the dimension-combinatorial explosion of possibilities, you might not even know the right pattern when you see it.

38.2 Developments in Understanding the Neurophysiology of EEG

What does it mean, in terms of neurophysiological mechanisms, if a condition effect is reflected in power or phase? How should power-based connectivity results be interpreted compared to phase-based connectivity results? There are speculative answers to these and other related questions about how to interpret time-frequency features, but relatively little is known about the precise neurophysiological and neurobiological events that produce the kinds of dynamics measurable with noninvasive EEG electrodes and MEG sensors. The mathematical development of time-frequency-based analyses and their applications to studying cognition have advanced beyond the understanding of the physiological events that might underlie the results of those analyses.

Imagine that the following (hypothetical) conjectures were established facts about the neurophysiological events that are measured by scalp-level recordings: alpha power reflects rhythmic inputs from the thalamus to cortical layer IV, theta phase reflects the temporal dynamics of GABAergic interneurons that are driven by long-range cortical inputs, and alpha-gamma coupling reflects the regulation of layer II cells by layer V cells. If it were possible to make inferences like this with confidence, the understanding of the neural mechanisms underlying human cognition and the ability to make detailed hypotheses linking cognition to physiological events would increase greatly.

Development of this level of knowledge relies on neurobiologists and neuroscientists performing *in vitro* electrophysiology. Hopefully, with an increase in interest in M/EEG and LFPs, these mechanisms will be increasingly well understood, and cognitive electrophysiologists will be able to make increasingly neurobiologically precise hypotheses. One challenge of bridging this gap is that EEG measures meso- and macroscopic-level activity, whereas the *in vitro* and *in vivo* physiological studies are typically done at the microscopic level. The difference in spatial scale spans several orders of magnitude, and it is not clear to what extent mesoscopic-level observations can be interpreted in terms of microscopic processes.

Aside from improved understanding of physiological processes, cognitive theories and models could be developed to make predictions not only regarding cognitive processes and whether a brain region is “active” (which is an ambiguous term; Singh 2012) but to make more precise predictions regarding frequency-band-specific dynamics, connectivity, and other physiological processes. Many biophysical models that make predictions regarding physiological dynamics do not exhibit cognitive behaviors, and most cognitive models that can account for some behaviors do not make predictions regarding neurophysiological dynamics (other than “activation” in some brain region). Some integration between these

levels of modeling has been done, and ideally these two literatures will continue to merge to the mutual benefit of both the neurophysiologists and the cognitive psychologists.

38.3 Developments in Experiment Design

Cognitive electrophysiology lacks a rich repertoire of experiment paradigms that have even a mediocre amount of external validity. Most of the basic tasks that are used in cognitive electrophysiology were developed decades or even a hundred years ago. On the one hand, the reliance on standard and widely used but unrealistic experiment designs is understandable—the old tasks are still used because they produce replicable results, and they provide solid foundations for new methods and analyses and for investigations of individual and group differences (e.g., disease or genotype). Any researcher will be able to defend his or her choice of task and identify situations in the real world that that task is designed to capture (often, these situations involve decoding objects in the rain while driving, crossing the street in London, or remembering where you left your car keys). But these tasks are rarely “like” their claimed real-world referents; they lack the context and experiential qualities. Volunteer undergraduate research subjects don’t imagine they are in a big parking lot looking for their car; they realize they are in a dark room, looking at abstract pictures on a computer screen, sitting in an uncomfortable chair with wires attached to their head and face, and remembering not to look around, smile, talk, swallow, or blink except at prespecified times.

Cognitive electrophysiologists can hardly be blamed or criticized for using such abstract experiment designs that have little basis in or relevance to real-world behavior (at least I hope not—I am among those who use decades-old experiment paradigms that are being criticized here). EEG data are so complex that even simple tasks with few conditions often lead to overwhelmingly complicated datasets. Furthermore, researchers are understandably hesitant to introduce potential sources of noise and artifact, not to mention the myriad uncontrolled variables present outside the safe and sterile confines of the lab. But those uncontrolled variables form the context within which the environment is interpreted, within which information is processed and used, and within which emotions, social interactions, and all of the other important and meaningful events in life occur. At some point we may all need to come to terms with the gap between what we want to understand about how the brain works and what we actually elicit from subjects in cold, dark, sound-attenuated EEG recording booths.

Fortunately, some of the artifacts that are present in the real world become less of a concern with advances in signal processing. Consider, for example, the demonstration of cognitive ERPs recorded from a subject on a treadmill (Gramann et al. 2011). Data analysis

and processing capabilities are one important aspect to performing more real-world relevant studies. The other important aspect will be figuring out how to deal with the myriad uncontrolled (but measurable) variables that are dynamic over time, both within and across subjects. This is no small feat, of course, but will likely become increasingly important in cognitive electrophysiology and neuroscience in general.

38.4 Developments in Measurement Technology

Noninvasive recording technology has improved considerably in the past few decades. Consider that in the 1980s EEG typically involved a few to 16 electrodes, and subjects might leave the lab with scabs on their scalp from all the scrubbing that was necessary to lower impedances. Now, use of up to 256 electrodes is becoming increasingly common, as are active electrodes (electrodes with a small built-in preamplifier), high-impedance gel-less sponge-net systems, MRI-compatible EEG, and mobile wireless EEG. Along with high-density EEG comes more complete coverage of the head, including electrodes on the neck and face, that improve the precision of source localization and the surface Laplacian. Perhaps future developments in measurement technology, in addition to improvements on present-day technology, will involve affordable and portable magnetic sensors or the ability to monitor simultaneous cortical hemodynamic (e.g., via optical imaging) and electrical activity without an MRI scanner.

Another area for improvement is precise localization of subject-specific electrode positions and head shape. In most EEG studies electrode caps are placed on each subject's head individually. Although the net is positioned in such a way that the electrodes are in approximately the same location in all subjects, there inevitably is variability across subjects on the scale of a few centimeters. Precise electrode positions are rarely measured, and thus, variability in electrode placement increases error in cross-subject comparisons. Several companies sell electrode localization equipment, but these systems are generally not practical, either due to the high cost or because they involve touching a sonic pen to each electrode individually, which is not feasible for more than 50 electrodes and certainly not for 256 electrodes. If the precise electrode positions were known for each subject, data could be more accurately averaged across subjects, for example via spline interpolation to common head points. Having precise electrode positions will improve the accuracy of spatial filters, topographical mapping, and source localization accuracy.

Simultaneous peripheral measurement is also an area of technology that should be improved. As discussed in the next section, the brain does not exist in an isolated vat but rather in the dynamic milieu of the body. The EEG cap itself should be merely one component

of the cognitive electrophysiology setup. Simultaneous with EEG should be eye recordings (X and Y positions and pupil diameter), facial expression recordings (to identify microexpressions), heart rate, breathing rate, skin conductance, muscle activity from the hands used to indicate responses, continuous cortisol sampling, and perhaps other sensors that measure gut activity and body temperature (preferably through noninvasive sensors). The technology will have to improve such that these extra sensors are neither time-consuming to set up nor distracting to the subject. All of these sensors would be measured with the EEG as separate data channels. Thus, the EEG data file might contain 256 electrodes and 15 peripheral sensors. Although some of these peripheral sensors are already technologically possible and used in some labs, they are often difficult or cumbersome to operate and therefore are used only when they are key to the study design. Ideally, additional peripheral data would be recorded “for free,” which will prompt researchers to collect the data and, offline and initially in an exploratory manner, determine, for example, the influence of heart rate variability on gamma-band synchrony during visual conscious awareness or the influence of a microexpression of anger on the magnitude of the error-related negativity.

Having 270 electrodes and other sensors recorded at 1 kHz will produce an enormous amount of data (around one gigabyte of compressed data every 12–15 min). Fortunately, there are parallel continual improvements in computing technology, not only in terms of personal computers but also servers and grid and cloud computing. In the future, having 10 gigabytes of raw data will be no problem because 256-bit computers with a steady-state-drive of several terabytes and hundreds of gigabytes of RAM will be standard. You will be able to view the results of the analyses remotely, via a 3-D projection system built into your watch.

38.5 The Role of the Body in Brain Function

The brain does not exist on its own, independent of the body. The brain receives a large array of signals from the body concerning its internal milieu, including heart rate, breathing, digestion, hunger, and the urgency of expelling biowaste (Badr et al. 1982; Tuk, Trampe, and Warlop 2011). Many states in the body’s milieu affect brain function (Cryan and Dinan 2012), and the visceral systems probably have a larger role in shaping perception, cognition, emotion, and action than we appreciate.

The opposite is true as well: the brain does not only generate signals for itself; it generates signals that are used to modulate processes in the body. By measuring from the body, it will be possible to determine which aspects of brain activity are used for internal processing and which aspects of brain activity are used to modulate ongoing visceral processes.

A better understanding of the role of the body in brain function will be facilitated by improvements in recording technology, as mentioned above. When body signals are simply embedded in the EEG data file with no or minimal extra hassle, scientists will begin to examine brain-body interactions and will incorporate them into their research programs and theory developments. This will also facilitate linking experiment findings to real-world behaviors, as discussed in section 38.3.

38.6 Determining Causality

The vast majority of experiments in cognitive electrophysiology are correlational in that they involve passively measuring brain activity during a task or resting period with no manipulations of brain activity or brain chemistry. Correlational studies are critical to scientific progress, particularly early in a research topic when too little is known to make specific predictions for causal interventions. With so many dimensions in the brain and so many potential links to behavior, correlational studies provide the necessary groundwork for isolating the cognition-relevant brain dynamics and disentangling them from the background and non-task-related brain dynamics.

Causal interventions, however, are necessary to determine which features of those brain dynamics reflect core mechanisms of brain computation. There are several methods for assessing causality in human brain research. Arguably the best method for assessing the causal involvement of brain oscillations in cognition in humans is transcranial alternating-current stimulation (TACS; also discussed in section 5.5). TACS involves placing two electrodes, an anode and a cathode, on the scalp and running a current between them. The alternating current can be specified from between 0.01 Hz and 100 Hz. This provides the opportunity to stimulate a brain region in a highly frequency-band-specific manner. The disadvantages are that despite its incredible temporal precision, its temporal resolution is rather poor. This is due to the long up-ramp and down-ramp times (10–30 s), which are necessary to avoid physical discomfort. The spatial resolution and precision of TACS are fairly poor at this time, although there will likely be future improvements in the technology to increase spatial precision, perhaps to a few cubic centimeters.

Other methods currently available for studying causality in human neuroscience are not well suited for testing hypotheses about brain oscillations. Transcranial magnetic stimulation, for example, can induce band-specific power increases, but this depends on the brain region of stimulation (Rosanova et al. 2009) and seems to have transient effects (Thut et al. 2011). Pharmacological agents can affect many chemical systems in many brain regions. Brain lesions do not generate specific patterns of oscillations and are often associated with

anatomical and functional reorganization. These are not criticisms of the methods per se or of their use in cognitive neuroscience; rather, if the goal of the research is to investigate the causal role of frequency-band-specific activity, TACS has clear advantages over other methods.

For studying oscillations in mice or rats, the best current technology is optogenetics, where specific oscillations can be induced at the cellular level by light-activated neurons. Unfortunately for researchers of the human brain, optogenetics in humans is unlikely to be approved by ethics committees in the near future.

38.7 Inferring Cognitive States from EEG Signatures: Inverse Inference

Young scientists are taught to avoid making inverse inferences and to shun others for making such inferences. In many cases this advice is appropriate: it is demonstrably incorrect to infer that a person experienced fear because there was a change in blood flow in his or her amygdala. Similarly, one cannot infer that a subject experienced response conflict simply because the power of theta-band activity in prefrontal cortex increased. These considerations lead to the conclusion that inverse inference should be avoided altogether.

But there is a difference between inverse inference being limited by our infantile understanding of brain function and inverse inference being absolutely impossible. There may exist one-to-one mappings between patterns of brain activity and mental states or cognitive processes, but our current thinking and analytic approaches are too simplistic to discover these one-to-one mappings. This is why inverse inference is often inappropriate. Perhaps in the future accurate decoding of at least some mental states or basic cognitive processes based on patterns of EEG activity will be uncontroversial.

Inverse inference is a major goal of research on brain-computer interfaces. Here the idea is to allow people to control external devices such as robotic arms or computer software based on patterns of EEG activity. It would be great to decode scalp EEG data to allow someone to move a robotic arm and precision-grasp a pen in order to sign an informed consent document in a research study. How far could brain-state decoding go? It already seems feasible to decode enough information from the EEG signal to spin a wheel on a computer screen to pick letters (Pires, Nunes, and Castelo-Branco 2011; Schalk and Leuthardt 2011) or to move a cursor to four cardinal directions (Bahramisharif et al. 2010). Perhaps it might also be possible to decode basic emotional reactions such as pain or happiness. How about language or dreams? Decoding such intricate and complex phenomena seems out of reach, mainly because the individual representations of words and images are complex and likely to overlap in space, time, and frequency. However, visual decoding from fMRI signals is

improving (Tong and Pratte 2012), so perhaps inverse inference of some internal states will be possible in the future. In addition to potential clinical applications, development of brain-computer interfaces will also facilitate a basic understanding of neurocognitive mechanisms by identifying the features of brain activity that are most task-relevant and under direct conscious control (Jensen et al. 2011).

Multivariate analyses may facilitate development of inverse inference and brain-computer interfaces by considering that not only particular EEG signals but also relationships among the dimensions of EEG signals are relevant. To some extent, this is already done in some studies by using the ratio of, for example, alpha power to theta power rather than the absolute level of alpha power. More sophisticated methods that might increase the accuracy of classification for brain-computer interfaces might employ algorithms that determine optimal weights for each frequency. One major limitation with this approach, as mentioned earlier and as is a common limitation with many other data-driven EEG analyses, is that the search space is larger than the typical amount of data, and the analysis possibilities typically exceed the theory-driven guidance of those analyses.

38.8 Tables of Activation

Tables of activations have been standard in MRI-based research for over a decade but are rarely used in cognitive electrophysiology. This is unfortunate because tables of time-frequency results would facilitate both qualitative and quantitative cross-study comparisons. There is no reason not to include tables of activation in cognitive electrophysiology studies, both for hypothesis-driven and for exploratory data-driven studies. Suggestions for what information to include in such tables were presented in section 36.9.

38.9 Disease Diagnosis and Predicting Treatment Course and Success

It would be great to be able to record an EEG from a patient and determine the disease and most appropriate course of treatment for that patient (that is, whether that patient will best respond to some types of medication or therapies). Given the complexity of brain diseases in terms of their genetic and biological development, environmental influences, subtypes, and variability of treatment options and their success rates, a set of EEG signatures unique for each disease-treatment combination is unlikely. However, it is plausible that EEG measurements could help determine subtypes or suggest likely treatment options.

It is already established that characteristics of the EEG signal are modulated by disease states. For example, ERPs such as the P50 (Potter et al. 2006) and gamma-band phase

synchronization (Uhlhaas and Singer 2011) are linked to schizophrenia; the error-related negativity is linked to obsessive-compulsive disorder, anxiety disorders, and depression (Endrass et al. 2010; Olvet, Klein, and Hajcak 2010; Weinberg, Klein, and Hajcak 2012); visual evoked responses are linked to autism (Milne 2011); and resting-state connectivity is linked to Alzheimer (Stam 2010), Parkinson (Stoffers, Bosboom, and Deijen 2008), and other diseases. This is certainly not an exhaustive list of the number of EEG characteristics that have been linked to disease states.

Predicting the success of treatment options should, in theory, be straightforward. It would require large-scale studies rather than studies with few subjects, and it would benefit from a data-driven approach based on statistical prediction rather than a “top-down” approach of starting with a theory about a cognitive process. If there were a database with hundreds or thousands of patients who had EEG before, during, and after different treatments, the data could be mined to find the combination of EEG signatures (including frequency-band-specific activity, connectivity, and others) that best predicts individual treatment outcomes. No hypotheses about specific ERP components, tasks, or brain regions would be necessary because the goal of this research is to find and confirm patterns in the data that are most clinically relevant, regardless of whether and how those patterns could be interpreted with respect to cognitive tasks or psychological theories. This is an example of how the multidimensional nature of EEG can be optimally utilized, provided that there is sufficient statistical power and replicability to reject false positives.

The outcome of this research would be an empirically guided facilitation of a diagnosis or course of treatment that is likely to be successful based on characteristics of the EEG data. The clinician would use this information as a valuable piece of evidence for decision making, along with other pieces of evidence such as a questionnaire or observation of walking gait. If the statistical prediction were high enough, the analyses could be automated such that the practitioners interacting with the patient would not need to be experts on EEG data analyses because a software program could perform the appropriate analyses and return empirically guided diagnosis or treatment recommendations with appropriate estimates of uncertainty in the results. This would be much more practically and clinically significant than showing a modulation of a single ERP component that may be difficult to interpret at the single-patient level because of high intersubject variability (Luck et al. 2011).

38.10 Clinical Relevance Is Not Necessary for the Advancement of Science

Increasingly in recent years, grant proposals are required to contain a section on clinical or societal relevance. The grant writer must come up with a story (sometimes, a stretched and unbelievable one) about how a better understanding of the electrophysiological mechanisms

of process X will lay important groundwork for developing treatments of whatever disease or brain disorder that funding agency lists on their website as being part of their core mission statement. Certainly, in some cases, this is true: scientists request funding to do research that will directly benefit the health of individuals suffering from some disease. But most of the time, this is not the case. Scientists request funding to do research so that we will have a better understanding of the fundamental mechanisms of the human brain and how it produces cognition and behavior. This is a worthy goal in itself, and there is no reason why all of neuroscience research should be directed at improving the quality of life for a minority of individuals who suffer from a disease.

The pursuit of information and the expansion of the corpus of human knowledge should be a valid justification for funding scientific research. Of course, studies that have direct or indirect benefit on the lives of a few people with diseases is also a noble goal, but clinical relevance should not be a necessary condition for funding high-quality scientific research on the workings of the human brain. I hope this trend toward forcing scientists to provide clinical justification for their research will reverse course in the future.

38.11 Replications

If you see an exciting finding in one paper but have not seen it replicated in other papers, it is difficult to know why. It could be that other people have tried and failed to replicate the finding but did not publish it (the “file drawer phenomenon”), or it could simply be that no one has tried to replicate it. Although replication is important for science and for scientists, the publication process and grant funding agencies generally favor novelty.

The focus on novelty over replication is understandable: EEG research is time-consuming, and you have to think carefully and strategically about what experiments to do. If it takes 18 months to bring a study from inception to journal acceptance, and you are under pressure to finish your PhD within 4 years and be competitive for a postdoc or faculty position or a grant that has a 15% success rate, no one would blame you for wanting to spend your time working on novel experiments rather than replicating existing findings. Furthermore, many of the most exciting findings in cognitive electrophysiology that should be replicated involve complicated and novel analyses. Replicating those analyses can often be performed only by scientists with sufficient expertise in math and programming rather than by students who may be highly motivated but who lack the necessary expertise (perhaps this book will help with that limitation).

The seemingly low number of replication studies cannot be blamed entirely on stressed-out PhD students; it also results from the attitudes of editors and reviewers. Many journals, not only top-tier journals, will reject manuscripts that do not contain enough novelty, and a

lack of novelty is always an easy criticism to mention when reviewing a manuscript or grant proposal (I admit that I have also been guilty of this below-the-belt criticism). Thus, replication studies can be harder to publish than novel studies, even if they are of equal or greater scientific rigor and importance.

Replication studies are important because experiments are imperfect. All experiments have some suboptimal design feature or have a few nonrepresentative subjects or suffer from some other limitation. In some cases these limitations could have been avoided by more careful pilot testing and critical thinking before the experiment began. But in most cases, suboptimal experiment features were necessary to optimize other experiment design features that were judged to be more important. A finding that is observed in one study might have resulted from an awkward feature of the experiment design, or it may be a statistical false alarm. However, if the same pattern of results is replicated in several different studies using different experiment designs and different analysis techniques, the likelihood of that finding reflecting a true brain process increases, and the chance that that finding was a false alarm or was due to a confound in the experiment decreases.

Attempts to replicate a result need not be driven by a lack of trust in the original authors. Furthermore, failures to replicate do not imply that the original authors were lying or that they manipulated their data. Of course, we all hope that our findings will generalize and replicate in other studies, but it is inevitable that some results reported in the literature were false alarms, and it is important for the development of cognitive electrophysiology (and science more generally) to determine which findings were overinterpreted. Indeed, even true effects may fail to be replicated if the sample sizes are too small or if the variability is too high. Replicating results in independent samples is better than increasing the number of subjects in a single study because p -values are related to the number of data points, and thus, having too many subjects may increase, not decrease, the risk of Type I errors (split-halves replication would be a good solution if there are too many subjects).

What would it take for the field to put more emphasis on replications? Some psychology journals are now explicitly publishing replication studies, with the idea that the experiment proposals are peer-reviewed before the data are collected, and the final paper is guaranteed to be accepted for publication so long as the procedures were followed appropriately and the authors made an earnest attempt to replicate the original experiment design and analysis methods.

Fortunately, because EEG data are so rich, replicating an EEG study does not need to be as dry as replicating a questionnaire study or other behavior-only study. That is, the replication itself could be simply the first half of the Results section. If the original findings were replicated, the authors could perform additional, perhaps exploratory, analyses to help

contextualize the original findings or link the findings to other areas of the literature. And if the original findings were not replicated, the authors could perform additional analyses to investigate why the original results were not replicated, for example by selecting a subsample of subjects who exhibited a particular behavioral effect.

Because of the multidimensional nature of EEG it may be difficult to evaluate how close a replication is to the original finding. For example, imagine that one study shows that the condition effect is maximal at 400 ms and 15 Hz at electrode Pz, and a replication study finds the maximal condition effect at 380 ms and 18 Hz at electrode P1. Probably you would say that the original finding was replicated within an acceptable margin of error. However, if the replication study found the condition effect at 5 Hz and 800 ms at electrode PO7, you might not consider that to be a replication. Exactly where the line is drawn between a replication and a failure to replicate is not clear and so would probably need to be evaluated on a case-by-case basis.

38.12 Double-Blind Review for Scientific Publications

The review process is a good but imperfect system. In my opinion, its biggest flaw—which is also easy to fix—is that there are personal biases from reviewers and from editors. I'll give most reviewers and most editors the benefit of the doubt that any biases they have are unconscious. That is, most reviewers and editors probably do not intentionally try to reject manuscripts from authors they do not like or consider competitors, nor do they intentionally try to accept manuscripts that are poor but from authors they like or consider colleagues. However, there are unavoidable biases that influence many of our decisions. You want your friends to do well, and maybe you have an unconscious bias against that guy with the really annoying laugh. Personally, I try to be as objective as possible when reviewing manuscripts and grant proposals, but humans can be objective only up to a point, and I'm sure that at an unconscious level, my evaluations have been and continue to be biased by my knowledge of the author list and by my opinion of their previous studies.

Other biases might be less subconscious. If a reviewer sees a manuscript from a well-known research group, that reviewer might be inclined to give them the benefit of the doubt when the manuscript is not clearly written or if the interpretations and conclusions of the manuscript do not appear warranted based on the results. In contrast, if that reviewer sees a manuscript with authors who are unfamiliar or who are new to a research topic, that reviewer might be overly critical and biased toward rejecting what otherwise might be an acceptable manuscript. These kinds of biases can never be completely eliminated from the peer-review process, but they can be minimized by having a double-blind review process.

Currently, the review process at most journals is single-blind, such that the authors do not know the identity of the reviewers, but the reviewers know the identities of the authors. Some journals, such as the *Frontiers* journals, publish the names of reviewers along with the paper to make the review process more open. I believe the opposite should be the case: The review process will be more fair and objective if the identities of the authors are concealed from the reviewers and from the editors, thus making the review process double-blind. A double-blind review process would mean that neither the editors nor the reviewers would have access to the author list until the paper has been accepted. Only a journal staff member or a computer program not involved in the decision-making process would have access to the identities of the authors.

There is no justification for the reviewers and the editors to know the identities of the authors. A scientific manuscript submitted to a peer-review journal should be evaluated on its own merits regardless of the previous work of that research group or their university affiliation. Of course, in some cases the reviewers will guess the authors or the research group based on the topic, methods, or writing style, but this will not be any worse than the current system of single-blind reviews. The review system can only become more meritocratic with double-blind reviews.

38.13 ?

I am not a soothsayer, and I cannot predict the future. (This is evidenced by my failure to pick winning lottery tickets and stocks.) I do not know what cognitive electrophysiology will be like 20 years from now, but I am excited to find out.

References

- Acunzo, D. J., G. Mackenzie, and M. C. van Rossum. 2012. Systematic Biases in Early ERP and ERF Components as a Result of High-Pass Filtering. *Journal of Neuroscience Methods* 209 (1):212–218.
- Akam, T. E., and D. M. Kullmann. 2012. Efficient “Communication through Coherence” Requires Oscillations Structured to Minimize Interference between Signals. *PLoS Computational Biology* 8 (11): e1002760.
- Allen, E. A., E. B. Erhardt, and V. D. Calhoun. 2012. Data Visualization in the Neurosciences: Overcoming the Curse of Dimensionality. *Neuron* 74 (4):603–608.
- Allen, E. A., J. Liu, K. A. Kiehl, J. Gelernter, G. D. Pearlson, N. I. Perrone-Bizzozero, and V. D. Calhoun. 2011. Components of Cross-Frequency Modulation in Health and Disease. *Frontiers in Systems Neuroscience* 5:59.
- Allen, J. J., and M. X. Cohen. 2010. Deconstructing the “Resting” State: Exploring the Temporal Dynamics of Frontal Alpha Asymmetry as an Endophenotype for Depression. *Frontiers in Human Neuroscience* 4:232.
- Anastassiou, C. A., R. Perin, H. Markram, and C. Koch. 2011. Ephaptic Coupling of Cortical Neurons. *Nature Neuroscience* 14 (2):217–223.
- Anderson, M. J., and C. J. F. Ter Braak. 2003. Permutation Tests for Multi-Factorial Analysis of Variance. *Journal of Statistical Computation and Simulation* 73 (2):85–113.
- Ansari-Asl, K., L. Senhadji, J. J. Bellanger, and F. Wendling. 2006. Quantitative Evaluation of Linear and Nonlinear Methods Characterizing Interdependencies between Brain Signals. *Physical Review E Stochastic, Nonlinear and Soft Matter Physics* 74 (3), Pt. 1:031916.
- Appelbaum, L. G., C. N. Boehler, R. Won, L. Davis, and M. G. Woldorff. 2012. Strategic Allocation of Attention Reduces Temporally Predictable Stimulus Conflict. *Journal of Cognitive Neuroscience* 24 (9):1834–1848.
- Axmacher, N., F. Mormann, G. Fernandez, C. E. Elger, and J. Fell. 2006. Memory Formation by Neuronal Synchronization. *Brain Research. Brain Research Reviews* 52 (1):170–182.

- Badr, G., C. A. Carlsson, M. Fall, S. Friberg, L. Lindstrom, and B. Ohlsson. 1982. Cortical Evoked Potentials Following Stimulation of the Urinary Bladder in Man. *Electroencephalography and Clinical Neurophysiology* 54 (5):494–498.
- Bahramisharif, A., M. van Gerven, T. Heskes, and O. Jensen. 2010. Covert Attention Allows for Continuous Control of Brain-Computer Interfaces. *European Journal of Neuroscience* 31 (8):1501–1508.
- Barnes, G. R., A. Hillebrand, I. P. Fawcett, and K. D. Singh. 2004. Realistic Spatial Sampling for MEG Beamformer Images. *Human Brain Mapping* 23 (2):120–127.
- Barnett, L., and A. K. Seth. 2011. Behaviour of Granger Causality under Filtering: Theoretical Invariance and Practical Application. *Journal of Neuroscience Methods* 201 (2):404–419.
- Barrett, A. B., M. Murphy, M. A. Bruno, Q. Noirhomme, M. Boly, S. Laureys, and A. K. Seth. 2012. Granger Causality Analysis of Steady-State Electroencephalographic Signals during Propofol-Induced Anaesthesia. *PLoS One* 7 (1):e29072.
- Benar, C. G., T. Papadopoulo, B. Torresani, and M. Clerc. 2009. Consensus Matching Pursuit for Multi-Trial EEG Signals. *Journal of Neuroscience Methods* 180 (1):161–170.
- Berman, B. D., S. G. Horovitz, B. Morel, and M. Hallett. 2012. Neural Correlates of Blink Suppression and the Buildup of a Natural Bodily Urge. *NeuroImage* 59 (2):1441–1450.
- Biggs, N. L., and E. K. Lloyd. 1999. *Graph Theory 1736–1936*. New York: Oxford University Press.
- Bokil, H., N. Laaris, K. Blinder, M. Ennis, and A. Keller. 2001. Ephaptic Interactions in the Mammalian Olfactory System. *Journal of Neuroscience* 21 (20):RC173.
- Bragin, A., G. Jando, Z. Nádasdy, J. Hetke, K. Wise, and G. Buzsaki. 1995. Gamma (40–100 Hz) Oscillation in the Hippocampus of the Behaving Rat. *Journal of Neuroscience* 15 (1 Pt. 1):47–60.
- Bressler, S. L., and A. K. Seth. 2011. Wiener-Granger Causality: A Well Established Methodology. *NeuroImage* 58 (2):323–329.
- Britz, J., and C. M. Michel. 2010. Errors Can Be Related to Pre-Stimulus Differences in ERP Topography and Their Concomitant Sources. *NeuroImage* 49 (3):2774–2782.
- Britz, J., and C. M. Michel. 2011. State-Dependent Visual Processing. *Frontiers in Psychology* 2:370.
- Bro, R., E. Acar, and T. Kolda. 2007. Resolving the Sign Ambiguity in the Singular Value Decomposition. *Journal of Chemometrics* 22 (2):135–140.
- Brookes, M. J., J. M. Zumer, C. M. Stevenson, J. R. Hale, G. R. Barnes, J. Vrba, and P. G. Morris. 2010. Investigating Spatial Specificity and Data Averaging in MEG. *NeuroImage* 49 (1):525–538.
- Brunet, D., M. M. Murray, and C. M. Michel. 2011. Spatiotemporal Analysis of Multichannel EEG: Cartool. *Computational Intelligence and Neuroscience* 2011:813–870.
- Bruns, A. 2004. Fourier-, Hilbert- and Wavelet-Based Signal Analysis: Are They Really Different Approaches? *Journal of Neuroscience Methods* 137 (2):321–332.

- Bruns, A., R. Eckhorn, H. Jokeit, and A. Ebner. 2000. Amplitude Envelope Correlation Detects Coupling among Incoherent Brain Signals. *Neuroreport* 11 (7):1509–1514.
- Buchanan, M. 2003. *Nexus: Small Worlds and the Groundbreaking Theory of Networks*. New York: W. W. Norton.
- Burgess, A. P. 2012. Towards a Unified Understanding of Event-Related Changes in the EEG: The Firefly Model of Synchronization through Cross-Frequency Phase Modulation. *PLoS One* 7 (9):e45630.
- Busch, N. A., J. Dubois, and R. VanRullen. 2009. The Phase of Ongoing EEG Oscillations Predicts Visual Perception. *Journal of Neuroscience* 29 (24):7869–7876.
- Buzsaki, G. 2006. *Rhythms of the Brain*. New York: Oxford University Press.
- Buzsaki, G., C. A. Anastassiou, and C. Koch. 2012. The Origin of Extracellular Fields and Currents—EEG, ECoG, LFP and Spikes. *Nature Reviews Neuroscience* 13 (6):407–420.
- Buzsaki, G., and A. Draguhn. 2004. Neuronal Oscillations in Cortical Networks. *Science* 304 (5679):1926–1929.
- Buzsaki, G., and X. J. Wang. 2012. Mechanisms of Gamma Oscillations. *Annual Review of Neuroscience* 35:203–225.
- Cadusch, P. J., W. Breckon, and R. B. Silberstein. 1992. Spherical Splines and the Interpolation, Deblurring and Transformation of Topographic EEG Data. *Brain Topography* 5:59.
- Canolty, R. T., C. F. Cadieu, K. Koepsell, R. T. Knight, and J. M. Carmena. 2012. Multivariate Phase-Amplitude Cross-Frequency Coupling in Neurophysiological Signals. *IEEE Transactions on Bio-Medical Engineering* 59 (1):8–11.
- Canolty, R. T., E. Edwards, S. S. Dalal, M. Soltani, S. S. Nagarajan, H. E. Kirsch, M. S. Berger, et al. 2006. High Gamma Power Is Phase-Locked to Theta Oscillations in Human Neocortex. *Science* 313 (5793):1626–1628.
- Canolty, R. T., and R. T. Knight. 2010. The Functional Role of Cross-Frequency Coupling. *Trends in Cognitive Sciences* 14 (11):506–515.
- Capilla, A., P. Pazo-Alvarez, A. Darriba, P. Campo, and J. Gross. 2011. Steady-State Visual Evoked Potentials Can Be Explained by Temporal Superposition of Transient Event-Related Responses. *PLoS One* 6 (1):e14543.
- Caplan, J. B., J. R. Madsen, S. Raghavachari, and M. J. Kahana. 2001. Distinct Patterns of Brain Oscillations Underlie Two Basic Parameters of Human Maze Learning. *Journal of Neurophysiology* 86 (1):368–380.
- Carp, J. 2012. On the Plurality of (Methodological) Worlds: Estimating the Analytic Flexibility of fMRI Experiments. *Frontiers in Neuroscience* 6:149.
- Cavanagh, J. F., L. Zambrano-Vazquez, and J. J. Allen. 2012. Theta Lingua Franca: A Common Mid-Frontal Substrate for Action Monitoring Processes. *Psychophysiology* 49 (2):220–238.

- Chatfield, C. 2004. *The Analysis of Time Series: An Introduction*. New York: Taylor & Francis Group.
- Chatterjee, S., and A. Firat. 2007. Generating Data with Identical Statistics but Dissimilar Graphics: A Follow up to the Anscombe Dataset. *American Statistician* 61 (3):248–254.
- Chawla, D., K. J. Friston, and E. D. Lumer. 2001. Zero-Lag Synchronous Dynamics in Triplets of Inter-connected Cortical Areas. *Neural Networks* 14 (6–7):727–735.
- Chen, Y., S. L. Bressler, and M. Ding. 2006. Frequency Decomposition of Conditional Granger Causality and Application to Multivariate Neural Field Potential Data. *Journal of Neuroscience Methods* 150 (2):228–237.
- Clayson, P. E., S. A. Baldwin, and M. J. Larson. 2012. How Does Noise Affect Amplitude and Latency Measurement of Event-Related Potentials (ERPs)? A Methodological Critique and Simulation Study. *Psychophysiology* 5 (2):174–186.
- Cogan, G. B., and D. Poeppel. 2011. A Mutual Information Analysis of Neural Coding of Speech by Low-Frequency MEG Phase Information. *Journal of Neurophysiology* 106 (2):554–563.
- Cohen, J., and P. Cohen. 1983. *Applied Multiple Regression: Correlation Analysis for the Behavioral Sciences*. New York: Taylor & Francis Group.
- Cohen, M. X. 2008. Assessing Transient Cross-Frequency Coupling in EEG Data. *Journal of Neuroscience Methods* 168 (2):494–499.
- Cohen, M. X. 2011a. Hippocampal-Prefrontal Connectivity Predicts Midfrontal Oscillations and Long-Term Memory Performance. *Current Biology* 21 (22):1900–1905.
- Cohen, M. X. 2011b. It's About Time. *Frontiers in Human Neuroscience* 5:2.
- Cohen, M. X., N. Axmacher, D. Lenartz, C. E. Elger, V. Sturm, and T. E. Schlaepfer. 2009. Nucleus Accumbens Phase Synchrony Predicts Decision-Making Reversals Following Negative Feedback. *Journal of Neuroscience* 29 (23):7591–7598.
- Cohen, M. X., and J. F. Cavanagh. 2011. Single-Trial Regression Elucidates the Role of Prefrontal Theta Oscillations in Response Conflict. *Frontiers in Psychology* 2:30.
- Cohen, M. X., and K. R. R. Ridderinkhof. 2013. EEG Source Reconstruction Reveals Frontal-Parietal Dynamics of Spatial Conflict Processing. *PLoS One* 8 (2):e57293.
- Cohen, M. X., and S. van Gaal. 2012. Dynamic Interactions between Large-Scale Brain Networks Predict Behavioral Adaptation after Perceptual Errors. *Cerebral Cortex* 23 (5):1061–1072.
- Cohen, M. X., and B. Voytek. in press. Linking Nonlinear Neural Dynamics to Single-Trial Human Behavior. In *Multiscale Analysis and Nonlinear Dynamics: From Molecules to the Brain*, ed. M. Z. Pesenson. New York: Wiley-VCH.
- Coles, M. G., G. Gratton, T. R. Bashore, C. W. Eriksen, and E. Donchin. 1985. A Psychophysiological Investigation of the Continuous Flow Model of Human Information Processing. *Journal of Experimental Psychology. Human Perception and Performance* 11 (5):529–553.

- Coles, M. G., M. K. Scheffers, and L. Fournier. 1995. Where Did You Go Wrong? Errors, Partial Errors, and the Nature of Human Information Processing. *Acta Psychologica* 90 (1–3):129–144.
- Coombes, S. 2010. Large-Scale Neural Dynamics: Simple and Complex. *NeuroImage* 52 (3):731–739.
- Cryan, J. F., and T. G. Dinan. 2012. Mind-Altering Microorganisms: The Impact of the Gut Microbiota on Brain and Behaviour. *Nature Reviews Neuroscience* 13 (10):701–712.
- Cui, J., L. Xu, S. L. Bressler, M. Ding, and H. Liang. 2008. Bsmart: A Matlab/C Toolbox for Analysis of Multichannel Neural Time Series. *Neural Networks* 21 (8):1094–1104.
- Dalal, S. S., A. G. Guggisberg, E. Edwards, K. Sekihara, A. M. Findlay, R. T. Canolty, M. S. Berger, et al. 2008. Five-Dimensional Neuroimaging: Localization of the Time-Frequency Dynamics of Cortical Activity. *NeuroImage* 40 (4):1686–1700.
- Darvas, F., D. Pantazis, E. Kucukaltun-Yildirim, and R. M. Leahy. 2004. Mapping Human Brain Function with MEG and EEG: Methods and Validation. *NeuroImage* 23 (Suppl 1):S289–S299.
- David, O., D. Cosmelli, and K. J. Friston. 2004. Evaluation of Different Measures of Functional Connectivity Using a Neural Mass Model. *NeuroImage* 21 (2):659–673.
- David, O., L. Harrison, and K. J. Friston. 2005. Modelling Event-Related Responses in the Brain. *NeuroImage* 25 (3):756–770.
- David, O., J. M. Kilner, and K. J. Friston. 2006. Mechanisms of Evoked and Induced Responses in MEG/EEG. *NeuroImage* 31 (4):1580–1591.
- Dawy, Z., B. Goebel, J. Hagenauer, C. Andreoli, T. Meitinger, and J. C. Mueller. 2006. Gene Mapping and Marker Clustering Using Shannon's Mutual Information. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 3 (1):47–56.
- Deco, G., V. K. Jirsa, P. A. Robinson, M. Breakspear, and K. Friston. 2008. The Dynamic Brain: From Spiking Neurons to Neural Masses and Cortical Fields. *PLoS Computational Biology* 4 (8):e1000092.
- Delorme, A., and S. Makeig. 2004. EEGLAB: An Open Source Toolbox for Analysis of Single-Trial EEG Dynamics. *Journal of Neuroscience Methods* 134:9–21.
- de Munck, J. C., and F. Bijma. 2010. How Are Evoked Responses Generated? The Need for a Unified Mathematical Framework. *Clinical Neurophysiology* 121 (2):127–129.
- Deng, S., W. R. Winter, S. Thorpe, and R. Srinivasan. 2011. EEG Surface Laplacian Using Realistic Head Geometry. *International Journal of Bioelectromagnetism* 13 (4):173–177.
- Deng, S., W. Winter, S. Thorpe, and R. Srinivasan. 2012. Improved Surface Laplacian Estimates of Cortical Potential Using Realistic Models of Head Geometry. *IEEE Transactions on Bio-Medical Engineering* 59 (11):2979–2985.
- Dhamala, M., G. Rangarajan, and M. Ding. 2008. Estimating Granger Causality from Fourier and Wavelet Transforms of Time Series Data. *Physical Review Letters* 100 (1):018701.

- Dien, J. 1998. Issues in the Application of the Average Reference: Review, Critiques, and Recommendations. *Behavior Research Methods, Instruments, & Computers* 30 (1):34–43.
- Dien, J. 2010. The ERP PCA Toolkit: An Open Source Program for Advanced Statistical Analysis of Event-Related Potential Data. *Journal of Neuroscience Methods* 187 (1):138–145.
- Dien, J. 2012. Applying Principal Components Analysis to Event-Related Potentials: A Tutorial. *Developmental Neuropsychology* 37 (6):497–517.
- Dien, J., D. J. Beal, and P. Berg. 2005. Optimizing Principal Components Analysis of Event-Related Potentials: Matrix Type, Factor Loading Weighting, Extraction, and Rotations. *Clinical Neurophysiology* 116 (8):1808–1825.
- Ding, J., G. Sperling, and R. Srinivasan. 2006. Attentional Modulation of SSVEP Power Depends on the Network Tagged by the Flicker Frequency. *Cerebral Cortex* 16 (7):1016–1029.
- Ding, M., S. L. Bressler, W. Yang, and H. Liang. 2000. Short-Window Spectral Analysis of Cortical Event-Related Potentials by Adaptive Multivariate Autoregressive Modeling: Data Preprocessing, Model Validation, and Variability Assessment. *Biological Cybernetics* 83 (1):35–45.
- Donner, T. H., D. Sagi, Y. S. Bonneh, and D. J. Heeger. 2008. Opposite Neural Signatures of Motion-Induced Blindness in Human Dorsal and Ventral Visual Cortex. *Journal of Neuroscience* 28 (41):10298–10310.
- Donner, T. H., and M. Siegel. 2011. A framework for Local Cortical Oscillation Patterns. *Trends in Cognitive Neuroscience* 15:191–199.
- Drewes, J., and R. VanRullen. 2011. This Is the Rhythm of Your Eyes: The Phase of Ongoing Electroencephalogram Oscillations Modulates Saccadic Reaction Time. *Journal of Neuroscience* 31 (12):4698–4708.
- Durand, D., and J. A. Greenwood. 1958. Modifications of the Rayleigh Test for Uniformity in Analysis of Two-Dimensional Orientation Data. *Journal of Geology* 66 (3):229–238.
- Egner, T., S. Ely, and J. Grinband. 2010. Going, Going, Gone: Characterizing the Time-Course of Congruency Sequence Effects. *Frontiers in Psychology* 1:154.
- Endrass, T., B. Schuermann, C. Kaufmann, R. Spielberg, R. Kniesche, and N. Kathmann. 2010. Performance Monitoring and Error Significance in Patients with Obsessive-Compulsive Disorder. *Biological Psychology* 84 (2):257–263.
- Engel, A. K., P. Fries, and W. Singer. 2001. Dynamic Predictions: Oscillations and Synchrony in Top-Down Processing. *Nature Reviews. Neuroscience* 2 (10):704–716.
- Erdős, P., and A. Rényi. 1960. On the Evolution of Random Graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Sciences* 5:17–61.
- Fedorov, V., F. Mannino, and R. Zhang. 2009. Consequences of Dichotomization. *Pharmaceutical Statistics* 8 (1):50–61.

- Fell, J., T. Dietl, T. Grunwald, M. Kurthen, P. Klaver, P. Trautner, C. Schaller, et al. 2004. Neural Bases of Cognitive ERPs: More Than Phase Reset. *Journal of Cognitive Neuroscience* 16 (9):1595–1604.
- Feurra, M., G. Bianco, E. Santaruccio, M. Del Testa, A. Rossi, and S. Rossi. 2011. Frequency-Dependent Tuning of the Human Motor System Induced by Transcranial Oscillatory Potentials. *Journal of Neuroscience* 31 (34):12165–12170.
- Fisher, A. E., G. R. Barnes, A. Hillebrand, C. Burrow, P. L. Furlong, and I. E. Holliday. 2004. Can You Tell Your Clunis from Your Cubitus? A Benchmark for Functional Imaging. *BMJ (Clinical Research Edition)* 329 (7480):1492–1493.
- Florian, G., and G. Pfurtscheller. 1995. Dynamic Spectral Analysis of Event-Related EEG Data. *Electroencephalography and Clinical Neurophysiology* 95 (5):393–396.
- Florin, E., J. Gross, J. Pfeifer, G. R. Fink, and L. Timmermann. 2010. The Effect of Filtering on Granger Causality Based Multivariate Causality Measures. *NeuroImage* 50 (2):577–588.
- Franzblau, L. E., and K. C. Chung. 2012. Graphs, Tables, and Figures in Scientific Publications: The Good, the Bad, and How Not to Be the Latter. *Journal of Hand Surgery* 37 (3):591–596.
- Fraser, A. M., and H. L. Swinney. 1986. Independent Coordinates for Strange Attractors from Mutual Information. *Physical Review A*. 33 (2):1134–1140.
- Freedman, D., and P. Diaconis. 1981. On the Histogram as a Density Estimator: L2 Theory. *Probability Theory and Related Fields* 57 (4):453–476.
- Freeman, W. J. 2004. Origin, Structure, and Role of Background EEG Activity. Part 1. Analytic Amplitude. *Clinical Neurophysiology* 115 (9):2077–2088.
- Freeman, W. J. 2006. Origin, Structure, and Role of Background EEG Activity. Part 4: Neural Frame Simulation. *Clinical Neurophysiology* 117 (3):572–589.
- Fries, P. 2005. A Mechanism for Cognitive Dynamics: Neuronal Communication through Neuronal Coherence. *Trends in Cognitive Sciences* 9 (10):474–480.
- Fries, P., R. Scheeringa, and R. Oostenveld. 2008. Finding Gamma. *Neuron* 58 (3):303–305.
- Friston, K. 1994. Functional and Effective Connectivity in Neuroimaging: A Synthesis. *Human Brain Mapping* 2:56–78.
- Fuchs, M., J. Kastner, M. Wagner, S. Hawes, and J. S. Ebersole. 2002. A Standardized Boundary Element Method Volume Conductor Model. *Clinical Neurophysiology* 113 (5):702–712.
- Gaiplard, R., S. Dehaene, C. Adam, S. Clemenceau, D. Hasboun, M. Baulac, L. Cohen, et al. 2009. Converging Intracranial Markers of Conscious Access. *PLoS Biology* 7 (3):e61.
- Genovese, C. R., N. A. Lazar, and T. Nichols. 2002. Thresholding of Statistical Maps in Functional Neuroimaging Using the False Discovery Rate. *NeuroImage* 15 (4):870–878.

- Geurts, H. M., R. P. Grasman, S. Verte, J. Oosterlaan, H. Roeyers, S. M. van Kammen, and J. A. Sergeant. 2008. Intra-Individual Variability in ADHD, Autism Spectrum Disorders and Tourette's Syndrome. *Neuropsychologia* 46 (13):3030–3041.
- Geweke, J. 1982. Measurement of Linear Dependence and Feedback between Multiple Time Series. *Journal of the American Statistical Association* 77:304–313.
- Gonsalvez, C. L., and J. Polich. 2002. P300 Amplitude Is Determined by Target-to-Target Interval. *Psychophysiology* 39 (3):388–396.
- Gramann, K., J. T. Gwin, D. P. Ferris, K. Oie, T. P. Jung, C. T. Lin, L. D. Liao, et al. 2011. Cognition in Action: Imaging Brain/Body Dynamics in Mobile Humans. *Reviews in the Neurosciences* 22 (6):593–608.
- Gramfort, A., T. Papadopoulo, E. Olivi, and M. Clerc. 2010. Openmeeg: Opensource Software for Quasi-static Bioelectromagnetics. *Biomedical Engineering Online* 9:45.
- Grandchamp, R., and A. Delorme. 2011. Single-Trial Normalization for Event-Related Spectral Decomposition Reduces Sensitivity to Noisy Trials. *Frontiers in Psychology* 2:236.
- Granger, C. W. J. 1969. Investigating Causal Relations by Econometric Models and Cross-Spectral Methods. *Econometrica* 37 (3):424–438.
- Granholm, E., and S. R. Steinhauer. 2004. Pupillometric Measures of Cognitive and Emotional Processes. *International Journal of Psychophysiology* 52 (1):1–6.
- Gratton, G., M. G. Coles, and E. Donchin. 1983. A New Method for Off-Line Removal of Ocular Artifact. *Electroencephalography and Clinical Neurophysiology* 55 (4):468–484.
- Gray C. M., and W. Singer. 1989. Stimulus-specific neuronal oscillations in orientation columns of cat visual cortex. *Proceedings of the National Academy of Science* 86:1698–1702.
- Gross, J., S. Baillet, G. R. Barnes, R. N. Henson, A. Hillebrand, O. Jensen, K. Jerbi, et al. 2013. Good Practice for Conducting and Reporting MEG Research. *NeuroImage* 65:349–363.
- Gross, J., J. Kujala, M. Hamalainen, L. Timmermann, A. Schnitzler, and R. Salmelin. 2001. Dynamic Imaging of Coherent Sources: Studying Neural Interactions in the Human Brain. *Proceedings of the National Academy of Sciences of the United States of America* 98 (2):694–699.
- Hallez, H., B. Vanrumste, R. Grech, J. Muscat, W. De Clercq, A. Vergult, Y. D'Asseler, et al. 2007. Review on Solving the Forward Problem in EEG Source Analysis. *Journal of Neuroengineering and Rehabilitation* 4:46.
- Handelsman, D. J. 2002. Optimal Power Transformations for Analysis of Sperm Concentration and Other Semen Variables. *Journal of Andrology* 23 (5):629–634.
- Handy, T. C., ed. 2004. *Event-Related Potentials: A Methods Handbook*. Cambridge, MA: MIT Press.
- Hannula, D. E., R. R. Althoff, D. E. Warren, L. Riggs, N. J. Cohen, and J. D. Ryan. 2010. Worth a Glance: Using Eye Movements to Investigate the Cognitive Neuroscience of Memory. *Frontiers in Human Neuroscience* 4:166.

- Hassler, U., N. T. Barreto, and T. Gruber. 2011. Induced Gamma Band Responses in Human EEG after the Control of Miniature Saccadic Artifacts. *NeuroImage* 57 (4):1411–1421.
- Haufe, S., V. V. Nikulin, K. R. Muller, and G. Nolte. 2013. A Critical Assessment of Connectivity Measures for EEG Data: A Simulation Study. *NeuroImage* 64:120–133.
- Herrmann, C. S. 2001. Human EEG Responses to 1–100 Hz Flicker: Resonance Phenomena in Visual Cortex and Their Potential Correlation to Cognitive Phenomena. *Experimental Brain Research* 137 (3–4):346–353.
- Herrmann, C. S., I. Frund, and D. Lenz. 2010. Human Gamma-Band Activity: A Review on Cognitive and Behavioral Correlates and Network Models. *Neuroscience and Biobehavioral Reviews* 34 (7):981–992.
- Herzel, H., and I. Grosse. 1995. Measuring Correlations in Symbol Sequences. *Physica A: Statistical Mechanics and Its Applications* 216 (4):518–542.
- Hesse, W., E. Moller, M. Arnold, and B. Schack. 2003. The Use of Time-Variant EEG Granger Causality for Inspecting Directed Interdependencies of Neural Assemblies. *Journal of Neuroscience Methods* 124 (1):27–44.
- Hillebrand, A., and G. R. Barnes. 2011. Practical Constraints on Estimation of Source Extent with MEG Beamformers. *NeuroImage* 54 (4):2732–2740.
- Hillebrand, A., K. D. Singh, I. E. Holliday, P. L. Furlong, and G. R. Barnes. 2005. A New Approach to Neuroimaging with Magnetoencephalography. *Human Brain Mapping* 25 (2):199–211.
- Hipp, J. F., D. J. Hawellek, M. Corbetta, M. Siegel, and A. K. Engel. 2012. Large-Scale Cortical Correlation Structure of Spontaneous Oscillatory Activity. *Nature Neuroscience* 15 (6):884–890.
- Hjorth, B. 1975. An On-Line Transformation of EEG Scalp Potentials into Orthogonal Source Derivations. *Electroencephalography and Clinical Neurophysiology* 39:526–530.
- Hoffmann, S., and M. Falkenstein. 2008. The Correction of Eye Blink Artefacts in the EEG: A Comparison of Two Prominent Methods. *PLoS One* 3 (8):e3004.
- Hoozemans, M. J., and J. H. van Dieen. 2005. Prediction of Handgrip Forces Using Surface EMG of Forearm Muscles. *Journal of Electromyography and Kinesiology* 15 (4):358–366.
- Huang, N. E., Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N. C. Yen, et al. 1998. The Empirical Mode Decomposition and the Hilbert Spectrum for Nonlinear and Nonstationary Time Series Analysis. *Proceedings of the Royal Society of London* 454 (1971):903–995.
- Huff, D. 1991. *How to Lie with Statistics*. New York: Penguin Books.
- Humphries, M. D., and K. Gurney. 2008. Network “Small-World-Ness”: A Quantitative Method for Determining Canonical Network Equivalence. *PLoS One* 3 (4):e0002051.
- Jensen, O., A. Bahramisharif, R. Oostenveld, S. Klanke, A. Hadjipapas, Y. O. Okazaki, and M. A. van Gerven. 2011. Using Brain-Computer Interfaces and Brain-State Dependent Stimulation as Tools in Cognitive Neuroscience. *Frontiers in Psychology* 2:100.

- Jensen, O., M. Bonnefond, and R. VanRullen. 2012. An Oscillatory Mechanism for Prioritizing Salient Unattended Stimuli. *Trends in Cognitive Sciences* 16 (4):200–206.
- Jensen, O., and L. L. Colgin. 2007. Cross-Frequency Coupling between Neuronal Oscillations. *Trends in Cognitive Sciences* 11 (7):267–269.
- Jensen, O., H. van Dijk, and A. Mazaheri. 2010. Amplitude Asymmetry as a Mechanism for the Generation of Slow Evoked Responses. *Clinical Neurophysiology* 121 (7):1148–1149; author reply 1149–1150.
- Jeong, J., J. C. Gore, and B. S. Peterson. 2002. A Method for Determinism in Short Time Series, and Its Application to Stationary EEG. *IEEE Transactions on Bio-Medical Engineering* 49 (11):1374–1379.
- Jmail, N., M. Gavaret, F. Wendling, A. Kachouri, G. Hamadi, J. M. Badier, and C. G. Benar. 2011. A Comparison of Methods for Separation of Transient and Oscillatory Signals in EEG. *Journal of Neuroscience Methods* 199 (2):273–289.
- Jones, K. A., B. Porjesz, D. Chorlian, M. Rangaswamy, C. Kamarajan, A. Padmanabhapillai, A. Stimus, et al. 2006. S-Transform Time-Frequency Analysis of P300 Reveals Deficits in Individuals Diagnosed with Alcoholism. *Clinical Neurophysiology* 117 (10):2128–2143.
- Jung, T. P., S. Makeig, C. Humphries, T. W. Lee, M. J. McKeown, V. Iragui, and T. J. Sejnowski. 2000. Removing Electroencephalographic Artifacts by Blind Source Separation. *Psychophysiology* 37 (2):163–178.
- Junghofer, M., T. Elbert, D. M. Tucker, and C. Braun. 1999. The Polar Average Reference Effect: A Bias in Estimating the Head Surface Integral in EEG Recording. *Clinical Neurophysiology* 110 (6):1149–1155.
- Kaminski, M., M. Ding, W. A. Truccolo, and S. L. Bressler. 2001. Evaluating Causal Relations in Neural Systems: Granger Causality, Directed Transfer Function and Statistical Assessment of Significance. *Biological Cybernetics* 85 (2):145–157.
- Kaplan, A. Y., A. A. Fingelkurts, A. A. Fingelkurts, S. V. Borisov, and B. S. Darkhovsky. 2005. Nonstationary Nature of the Brain Activity as Revealed by EEG/MEG: Methodological, Practical, and Conceptual Challenges. *Signal Processing* 85 (11):2190–2212.
- Keren, A. S., S. Yuval-Greenberg, and L. Y. Deouell. 2010. Saccadic Spike Potentials in Gamma-Band EEG: Characterization, Detection and Suppression. *NeuroImage* 49 (3):2248–2263.
- Kiebel, S. J., C. Tallon-Baudry, and K. J. Friston. 2005. Parametric Analysis of Oscillatory Activity as Measured with EEG/MEG. *Human Brain Mapping* 26 (3):170–177.
- Kilner, J. M., S. J. Kiebel, and K. J. Friston. 2005. Applications of Random Field Theory to Electrophysiology. *Neuroscience Letters* 374 (3):174–178.
- Kistler, W. M., J. L. van Hemmen, and C. I. De Zeeuw. 2000. Time Window Control: A Model for Cerebellar Function Based on Synchronization, Reverberation, and Time Slicing. *Progress in Brain Research* 124:275–297.

- Klimesch, W., R. Freunberger, P. Sauseng, and W. Gruber. 2008. A Short Review of Slow Phase Synchronization and Memory: Evidence for Control Processes in Different Memory Systems? *Brain Research* 1235:31–44.
- Klimesch, W., H. Russegger, M. Doppelmayr, and T. Pachinger. 1998. A Method for the Calculation of Induced Band Power: Implications for the Significance of Brain Oscillations. *Electroencephalography and Clinical Neurophysiology* 108 (2):123–130.
- Klimesch, W., P. Sauseng, and S. Hanslmayr. 2007. EEG Alpha Oscillations: The Inhibition-Timing Hypothesis. *Brain Research. Brain Research Reviews* 53 (1):63–88.
- Kopell, N., M. A. Kramer, P. Malerba, and M. A. Whittington. 2010. Are Different Rhythms Good for Different Functions? *Frontiers in Human Neuroscience* 4:187.
- Kral, A., R. Hartmann, J. Tillein, S. Heid, and R. Klinke. 2000. Congenital Auditory Deprivation Reduces Synaptic Activity within the Auditory Cortex in a Layer-Specific Manner. *Cerebral Cortex* 10 (7):714–726.
- Kramer, M. A., A. B. Tort, and N. J. Kopell. 2008. Sharp Edge Artifacts and Spurious Coupling in EEG Frequency Comodulation Measures. *Journal of Neuroscience Methods* 170 (2):352–357.
- Kravitz, A. V., and A. C. Kreitzer. 2011. Optogenetic Manipulation of Neural Circuitry in Vivo. *Current Opinion in Neurobiology* 21 (3):433–439.
- Krieg, J., A. Trebuchon-Da Fonseca, E. Martinez-Montes, P. Marquis, C. Liegeois-Chauvel, and C. G. Benar. 2011. A Comparison of Methods for Assessing Alpha Phase Resetting in Electrophysiology, with Application to Intracerebral EEG in Visual Areas. *NeuroImage* 55 (1):67–86.
- Kriegeskorte, N., M. A. Lindquist, T. E. Nichols, R. A. Poldrack, and E. Vul. 2010. Everything You Never Wanted to Know about Circular Analysis but Were Afraid to Ask. *Journal of Cerebral Blood Flow and Metabolism* 30 (9):1551–1557.
- Kriegeskorte, N., W. K. Simmons, P. S. Bellgowan, and C. I. Baker. 2009. Circular Analysis in Systems Neuroscience: The Dangers of Double Dipping. *Nature Neuroscience* 12 (5):535–540.
- Kucukaltun-Yildirim, E., D. Pantazis, and R. M. Leahy. 2006. Task-Based Comparison of Inverse Methods in Magnetoencephalography. *IEEE Transactions on Bio-Medical Engineering* 53 (9):1783–1793.
- Lachaux, J., E. Rodriguez, M. Le Van Quyen, A. Lutz, J. Martinerie, and F. Varela. 2000. Studying Single-Trials of Phase-Synchronous Activity in the Brain. *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering* 10 (10):2429–2439.
- Lachaux, J. P., E. Rodriguez, J. Martinerie, and F. J. Varela. 1999. Measuring Phase Synchrony in Brain Signals. *Human Brain Mapping* 8 (4):194–208.
- Lakatos, P., A. S. Shah, K. H. Knuth, I. Ulbert, G. Karmos, and C. E. Schroeder. 2005. An Oscillatory Hierarchy Controlling Neuronal Excitability and Stimulus Processing in the Auditory Cortex. *Journal of Neurophysiology* 94 (3):1904–1911.

- LaLumiere, R. T. 2011. A New Technique for Controlling the Brain: Optogenetics and Its Potential for Use in Research and the Clinic. *Brain Stimulation* 4 (1):1–6.
- Landolt, H. P. 2011. Genetic Determination of Sleep EEG Profiles in Healthy Humans. *Progress in Brain Research* 193:51–61.
- Larson, M. J., S. A. Baldwin, D. A. Good, and J. E. Fair. 2010. Temporal Stability of the Error-Related Negativity (ERN) and Post-Error Positivity (PE): The Role of Number of Trials. *Psychophysiology* 47 (6):1167–1171.
- Lattari, E., B. Velasques, F. Paes, M. Cunha, H. Budde, L. Basile, M. Cagy, et al. 2010. Corticomuscular Coherence Behavior in Fine Motor Control of Force: A Critical Review. *Revista de Neurologia* 51 (10):610–623.
- Le, J., and A. Gevins. 1993. Method to Reduce Blur Distortion from EEGs Using a Realistic Head Model. *IEEE Transactions on Bio-Medical Engineering* 40 (6):517–528.
- Lehmann, D. 1971. Multichannel Topography of Human Alpha EEG Fields. *Electroencephalography and Clinical Neurophysiology* 31 (5):439–449.
- Lehmann, D., P. L. Faber, S. Galderisi, W. M. Herrmann, T. Kinoshita, M. Koukkou, A. Mucci, et al. 2005. EEG Microstate Duration and Syntax in Acute, Medication-Naive, First-Episode Schizophrenia: A Multi-Center Study. *Psychiatry Research* 138 (2):141–156.
- Lehmann, D., C. M. Michel, I. Pal, and R. D. Pascual-Marqui. 1994. Event-Related Potential Maps Depend on Prestimulus Brain Electric Microstate Map. *International Journal of Neuroscience* 74 (1–4):239–248.
- Le Van Quyen, M., J. Foucher, J. Lachaux, E. Rodriguez, A. Lutz, J. Martinerie, and F. J. Varela. 2001. Comparison of Hilbert Transform and Wavelet Methods for the Analysis of Neuronal Synchrony. *Journal of Neuroscience Methods* 111 (2):83–98.
- Lin, C. F., and J. D. Zhu. 2012. Hilbert-Huang Transformation-Based Time-Frequency Analysis Methods in Biomedical Signal Applications. *Proceedings of the Institution of Mechanical Engineers. Part H, Journal of Engineering in Medicine* 226 (3):208–216.
- Linkenkaer-Hansen, K., D. J. Smit, A. Barkil, T. E. van Beijsterveldt, A. B. Brussaard, D. I. Boomsma, A. van Ooyen, et al. 2007. Genetic Contributions to Long-Range Temporal Correlations in Ongoing Oscillations. *Journal of Neuroscience* 27 (50):13882–13889.
- Lisman, J. 2005. The Theta/Gamma Discrete Phase Code Occuring During the Hippocampal Phase Precession May Be a More General Brain Coding Scheme. *Hippocampus* 15 (7):913–922.
- Lisman, J. E., and N. A. Otmakhova. 2001. Storage, Recall, and Novelty Detection of Sequences by the Hippocampus: Elaborating on the Socratic Model to Account for Normal and Aberrant Effects of Dopamine. *Hippocampus* 11 (5):551–568.
- Litvak, V., J. Mattout, S. Kiebel, C. Phillips, R. Henson, J. Kilner, G. Barnes, et al. 2011. EEG and MEG Data Analysis in SPM8. *Computational Intelligence and Neuroscience* 2011:852961.

- Luck, S. J. 2005. *An Introduction to the Event-Related Potential Technique*. Cambridge, MA: MIT Press.
- Luck, S. J., D. H. Mathalon, B. F. O'Donnell, M. S. Hamalainen, K. M. Spencer, D. C. Javitt, and P. J. Uhlhaas. 2011. A Roadmap for the Development and Validation of Event-Related Potential Biomarkers in Schizophrenia Research. *Biological Psychiatry* 70 (1):28–34.
- Makeig, S. 1993. Auditory Event-Related Dynamics of the EEG Spectrum and Effects of Exposure to Tones. *Electroencephalography and Clinical Neurophysiology* 86 (4):283–293.
- Makeig, S., S. Debener, J. Onton, and A. Delorme. 2004. Mining Event-Related Brain Dynamics. *Trends in Cognitive Sciences* 8 (5):204–210.
- Makeig, S., M. Westerfield, T. P. Jung, S. Enghoff, J. Townsend, E. Courchesne, and T. J. Sejnowski. 2002. Dynamic Brain Sources of Visual Evoked Responses. *Science* 295 (5555):690–694.
- Makinen, V., H. Tiitinen, and P. May. 2005. Auditory Event-Related Responses Are Generated Independently of Ongoing Brain Activity. *NeuroImage* 24 (4):961–968.
- Mallat, S. G., and Z. Zhang. 1993. Matching Pursuits with Time-Frequency Dictionaries. *IEEE Transactions on Signal Processing* 41 (12): 3397–3415.
- Maris, E., and R. Oostenveld. 2007. Nonparametric Statistical Testing of EEG- and MEG-Data. *Journal of Neuroscience Methods* 164 (1):177–190.
- Martinez-Montes, E., E. R. Cuspinera-Bravo, W. El-Deredy, J. M. Sanchez-Bornot, A. Lage-Castellanos, and P. A. Valdes-Sosa. 2008. Exploring Event-Related Brain Dynamics with Tests on Complex Valued Time-Frequency Representations. *Statistics in Medicine* 27 (15):2922–2947.
- Mathewson, K. E., C. Prudhomme, M. Fabiani, D. M. Beck, A. Lleras, and G. Gratton. 2012. Making Waves in the Stream of Consciousness: Entrainment Oscillations in EEG Alpha and Fluctuations in Visual Awareness with Rhythmic Visual Stimulation. *Journal of Cognitive Neuroscience* 24 (12):2321–2333.
- Maxwell, S. E., and H. D. Delaney. 2004. *Designing Experiments and Analyzing Data: A Model Comparison Perspective*, 2nd ed. New York: Psychology Press, Taylor & Francis Group.
- Mazaheri, A., S. Coffey-Corina, G. R. Mangun, E. M. Bekker, A. S. Berry, and B. A. Corbett. 2010. Functional Disconnection of Frontal Cortex and Visual Cortex in Attention-Deficit/Hyperactivity Disorder. *Biological Psychiatry* 67 (7):617–623.
- Mazaheri, A., and O. Jensen. 2006. Posterior Alpha Activity Is Not Phase-Reset by Visual Stimuli. *Proceedings of the National Academy of Sciences of the United States of America* 103 (8):2948–2952.
- Mazaheri, A., and O. Jensen. 2008. Asymmetric Amplitude Modulations of Brain Oscillations Generate Slow Evoked Responses. *Journal of Neuroscience* 28 (31):7781–7787.
- McBain, C. J., and J. A. Kauer. 2009. Presynaptic Plasticity: Targeted Control of Inhibitory Networks. *Current Opinion in Neurobiology* 19 (3):254–262.
- Michel, C. M., M. M. Murray, G. Lantz, S. Gonzalez, L. Spinelli, and R. Grave de Peralta. 2004. EEG Source Imaging. *Clinical Neurophysiology* 115 (10):2195–2222.

- Miller, K. J., S. Zanos, E. E. Fetz, M. den Nijs, and J. G. Ojemann. 2009. Decoupling the Cortical Power Spectrum Reveals Real-Time Representation of Individual Finger Movements in Humans. *Journal of Neuroscience* 29 (10):3132–3137.
- Milne, E. 2011. Increased Intra-Participant Variability in Children with Autistic Spectrum Disorders: Evidence from Single-Trial Analysis of Evoked EEG. *Frontiers in Psychology* 2:51.
- Mitra, P. P., and H. Bokil. 2007. *Observed Brain Dynamics*. Oxford: Oxford University Press.
- Mitra, P. P., and B. Pesaran. 1999. Analysis of Dynamic Brain Imaging Data. *Biophysical Journal* 76 (2):691–708.
- Moddemeijer, R. 1989. On Estimation of Entropy and Mutual Information of Continuous Distributions. *Signal Processing* 16:233–246.
- Montez, T., S. S. Poil, B. F. Jones, I. Manshanden, J. P. Verbunt, B. W. van Dijk, A. B. Brussaard, et al. 2009. Altered Temporal Correlations in Parietal Alpha and Prefrontal Theta Oscillations in Early-Stage Alzheimer Disease. *Proceedings of the National Academy of Sciences of the United States of America* 106 (5):1614–1619.
- Moran, R. J., P. Campo, F. Maestu, R. B. Reilly, R. J. Dolan, and B. A. Strange. 2010. Peak Frequency in the Theta and Alpha Bands Correlates with Human Working Memory Capacity. *Frontiers in Human Neuroscience* 4:200.
- Moratti, S., B. A. Clementz, Y. Gao, T. Ortiz, and A. Keil. 2007. Neural Mechanisms of Evoked Oscillations: Stability and Interaction with Transient Events. *Human Brain Mapping* 28 (12):1318–1333.
- Mormann, F., J. Fell, N. Axmacher, B. Weber, K. Lehnertz, C. E. Elger, and G. Fernandez. 2005. Phase/Amplitude Reset and Theta-Gamma Interaction in the Human Medial Temporal Lobe during a Continuous Word Recognition Memory Task. *Hippocampus* 15 (7):890–900.
- Mormann, F., K. Lehnartz, P. David, and C. E. Elger. 2000. Mean phase coherence as a measure for phase synchronization and its application to the EEG of epilepsy patients. *Physica D* 144:358–369.
- Muller, T. J., T. Koenig, J. Wackermann, P. Kalus, A. Fallgatter, W. Strik, and D. Lehmann. 2005. Subsecond Changes of Global Brain State in Illusory Multistable Motion Perception. *Journal of Neural Transmission* 112 (4):565–576.
- Murakami, S. and Y. Okada. 2006. Contributions of Principle Neocortical Neurons to Magnetoencephalography and Electroencephalography Signals. *Journal of Neurophysiology* 575 (3):925–936.
- Murray, M. M., D. Brunet, and C. M. Michel. 2008. Topographic ERP Analyses: A Step-by-Step Tutorial Review. *Brain Topography* 20 (4):249–264.
- Murzin, V., A. Fuchs, and J. A. Kelso. 2011. Anatomically Constrained Minimum Variance Beamforming Applied to EEG. *Experimental Brain Research* 214 (4):515–528.
- Muthukumaraswamy, S. D., R. A. Edden, D. K. Jones, J. B. Swettenham, and K. D. Singh. 2009. Resting GABA Concentration Predicts Peak Gamma Frequency and fMRI Amplitude in Response to Visual Stim-

- ulation in Humans. *Proceedings of the National Academy of Sciences of the United States of America* 106 (20):8356–8361.
- Muthukumaraswamy, S. D., C. J. Evans, R. A. Edden, R. G. Wise, and K. D. Singh. 2012. Individual Variability in the Shape and Amplitude of the BOLD-HRF Correlates with Endogenous GABAergic Inhibition. *Human Brain Mapping* 33 (2):455–465.
- Muthukumaraswamy, S. D., and K. D. Singh. 2011. A Cautionary Note on the Interpretation of Phase-Locking Estimates with Concurrent Changes in Power. *Clinical Neurophysiology* 122 (11):2324–2325.
- Naggara, O., J. Raymond, F. Guilbert, D. Roy, A. Weill, and D. G. Altman. 2011. Analysis by Categorizing or Dichotomizing Continuous Variables Is Inadvisable: An Example from the Natural History of Unruptured Aneurysms. *AJNR. American Journal of Neuroradiology* 32 (3):437–440.
- Neuper, C., and G. Pfurtscheller. 2001. Event-Related Dynamics of Cortical Rhythms: Frequency-Specific Features and Functional Correlates. *International Journal of Psychophysiology* 43 (1):41–58.
- Neuper, C., M. Wortz, and G. Pfurtscheller. 2006. ERD/ERS Patterns Reflecting Sensorimotor Activation and Deactivation. *Progress in Brain Research* 159:211–222.
- Nichols, T. E., and A. P. Holmes. 2002. Nonparametric Permutation Tests for Functional Neuroimaging: A Primer with Examples. *Human Brain Mapping* 15 (1):1–25.
- Nicol, R. M., S. C. Chapman, P. E. Vertes, P. J. Nathan, M. L. Smith, Y. Shtyrov, and E. T. Bullmore. 2012. Fast Reconfiguration of High-Frequency Brain Networks in Response to Surprising Changes in Auditory Input. *Journal of Neurophysiology* 107 (5):1421–1430.
- Nicolaou, N., S. Hourris, P. Alexandrou, and J. Georgiou. 2012. EEG-Based Automatic Classification of “Awake” Versus “Anesthetized” State in General Anesthesia Using Granger Causality. *PLoS One* 7 (3):e33869.
- Nikulin, V. V., K. Linkenkaer-Hansen, G. Nolte, and G. Curio. 2010. Non-Zero Mean and Asymmetry of Neuronal Oscillations Have Different Implications for Evoked Responses. *Clinical Neurophysiology* 121 (2):186–193.
- Nikulin, V. V., K. Linkenkaer-Hansen, G. Nolte, S. Lemm, K. R. Muller, R. J. Ilmoniemi, and G. Curio. 2007. A Novel Mechanism for Evoked Responses in the Human Brain. *European Journal of Neuroscience* 25 (10):3146–3154.
- Nikulin, V. V., G. Nolte, and G. Curio. 2012. Cross-Frequency Decomposition: A Novel Technique for Studying Interactions between Neuronal Oscillations with Different Frequencies. *Clinical Neurophysiology* 123 (7):1353–1360.
- Nolte, G., O. Bai, L. Wheaton, Z. Mari, S. Vorbach, and M. Hallett. 2004. Identifying True Brain Interaction from EEG Data Using the Imaginary Part of Coherency. *Clinical Neurophysiology* 115 (10):2292–2307.

- Nolte, G., A. Ziehe, V. V. Nikulin, A. Schlogl, N. Kramer, T. Brismar, and K. R. Muller. 2008. Robustly Estimating the Flow Direction of Information in Complex Physical Systems. *Physical Review Letters* 100 (23):234101.
- Nottage, J. F. 2010. Uncovering Gamma in Visual Tasks. *Brain Topography* 23 (1):58–71.
- Nunez, P. L., and R. Srinivasan. 2006. *Electric Fields of the Brain: The Neurophysics of EEG*. Oxford University Press.
- Nunez, P. L., and R. Srinivasan. 2010. Scale and Frequency Chauvinism in Brain Dynamics: Too Much Emphasis on Gamma Band Oscillations. *Brain Structure & Function* 215 (2):67–71.
- Olkkonen, H., P. Pesola, and J. T. Olkkonen. 2010. Computation of Hilbert Transform via Discrete Cosine Transform. *Journal of Signal and Information Processing* 1:18–23.
- Olvet, D. M., D. N. Klein, and G. Hajcak. 2010. Depression Symptom Severity and Error-Related Brain Activity. *Psychiatry Research* 179 (1):30–37.
- Oostenveld, R., P. Fries, E. Maris, and J. M. Schoffelen. 2011. Fieldtrip: Open Source Software for Advanced Analysis of MEG, EEG, and Invasive Electrophysiological Data. *Computational Intelligence and Neuroscience* 2011:156869.
- Oweis, R. J., and E. W. Abdulhay. 2011. Seizure Classification in EEG Signals Utilizing Hilbert-Huang Transform. *Biomedical Engineering Online* 10:38.
- Ozkurt, T. E., and A. Schnitzler. 2011. A Critical Note on the Definition of Phase-Amplitude Cross-Frequency Coupling. *Journal of Neuroscience Methods* 201 (2):438–443.
- Paninski, L. 2003. Estimation of Entropy and Mutual Information. *Neural Computation* 15:1191–1253.
- Panoulas, K. I., L. J. Hadjileontiadis, and S. M. Panas. 2008. Hilbert-Huang Spectrum as a New Field for the Identification of EEG Event Related De-/Synchronization for BCI Applications. *Conference Proceedings, Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society*, 3832–3835.
- Penny, W. D., E. Duzel, K. J. Miller, and J. G. Ojemann. 2008. Testing for Nested Oscillation. *Journal of Neuroscience Methods* 174 (1):50–61.
- Penny, W. D., S. J. Kiebel, J. M. Kilner, and M. D. Rugg. 2002. Event-Related Brain Dynamics. *Trends in Neurosciences* 25 (8):387–389.
- Peraza, L. R., A. U. Asghar, G. Green, and D. M. Halliday. 2012. Volume Conduction Effects in Brain Network Inference from Electroencephalographic Recordings Using Phase Lag Index. *Journal of Neuroscience Methods* 207 (2):189–199.
- Percival, D. B., and W. T. Walden. 1993. *Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques*. Cambridge, UK: Cambridge University Press.
- Perrin, F., J. Pernier, O. Bertrand, and J. F. Echallier. 1989a. Corrigenda EEG 02274. *Electroencephalography and Clinical Neurophysiology* 76:565.

- Perrin, F., J. Pernier, O. Bertrand, and J. F. Echallier. 1989b. Spherical Splines for Scalp Potential and Current Density Mapping. *Electroencephalography and Clinical Neurophysiology* 72 (2):184–187.
- Perrin, F., J. Pernier, O. Bertrand, M. H. Giard, and J. F. Echallier. 1987. Mapping of Scalp Potentials by Surface Spline Interpolation. *Electroencephalography and Clinical Neurophysiology* 66 (1):75–81.
- Pesaran, B., M. J. Nelson, and R. A. Andersen. 2008. Free Choice Activates a Decision Circuit between Frontal and Parietal Cortex. *Nature* 453 (7193):406–409.
- Pfurtscheller, G., and F. H. Lopes da Silva. 1999. Event-Related EEG/Meg Synchronization and Desynchronization: Basic Principles. *Clinical Neurophysiology* 110 (11):1842–1857.
- Pigorini, A., A. G. Casali, S. Casarotto, F. Ferrarelli, G. Baselli, M. Mariotti, M. Massimini, et al. 2011. Time-Frequency Spectral Analysis of TMS-Evoked EEG Oscillations by Means of Hilbert-Huang Transform. *Journal of Neuroscience Methods* 198 (2):236–245.
- Pinnegar, C. R., H. Khosravani, and P. Federico. 2009. Time-Frequency Phase Analysis of Ictal EEG Recordings with the S-Transform. *IEEE Transactions on Bio-Medical Engineering* 56 (11):2583–2593.
- Pires, G., U. Nunes, and M. Castelo-Branco. 2011. Gibbs Block Speller: Toward a Gaze-Independent P300-Based BCI. *Conference Proceedings, Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society*. 2011:6360–6364.
- Pitts, M. A., and J. Britz. 2011. Insights from Intermittent Binocular Rivalry and EEG. *Frontiers in Human Neuroscience* 5:107.
- Plochl, M., J. P. Ossandon, and P. Konig. 2012. Combining EEG and Eye Tracking: Identification, Characterization, and Correction of Eye Movement Artifacts in Electroencephalographic Data. *Frontiers in Human Neuroscience* 6:278.
- Plonsey, R., and D. B. Heppner. 1967. Considerations of Quasi-Stationarity in Electrophysiological Systems. *Bulletin of Mathematical Biophysics*. 29 (4):657–664.
- Polania, R., M. A. Nitsche, C. Korman, G. Batsikadze, and W. Paulus. 2012. The Importance of Timing in Segregated Theta Phase-Coupling for Cognitive Performance. *Current Biology* 22 (14):1314–1318.
- Potter, D., A. Summerfelt, J. Gold, and R. W. Buchanan. 2006. Review of Clinical Correlates of P50 Sensory Gating Abnormalities in Patients with Schizophrenia. *Schizophrenia Bulletin* 32 (4):692–700.
- Protopapa, F., D. Mylonas, K. Spiliotis, C. Siettos, N. Smyrnis, and I. Evdokimidis. 2011. Dynamic Analysis of EEG Signals during Spatial Working Memory Used for Either Perception Discrimination or Planning of Action. *Conference Proceedings, Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society*. 5896–5899.
- Ramsay, J., G. Hooker, and S. Graves. 2009. *Functional Data Analysis with R and Matlab*. New York: Springer.
- Ray, S., and J. H. R. Maunsell. 2010. Differences in Gamma Frequencies Across Visual Cortex Restrict Their Possible Use in Computation. *Neuron* 67:885–896.

- Regan, D. 1977. Steady-State Evoked Potentials. *Journal of the Optical Society of America* 67 (11):1475–1489.
- Regan, D. 1989. *Human Brain Electrophysiology: Evoked Potentials and Evoked Magnetic Fields in Science and Medicine*. Norwalk, CT: Appleton & Lange.
- Roelfsema, P. R., A. K. Engel, P. Konig, and W. Singer. 1997. Visuomotor Integration Is Associated with Zero Time-Lag Synchronization among Cortical Areas. *Nature* 385 (6612):157–161.
- Rohenkohl, G., and A. C. Nobre. 2011. Alpha Oscillations Related to Anticipatory Attention Follow Temporal Expectations. *Journal of Neuroscience* 31 (40):14076–14084.
- Rosanova, M., A. Casali, V. Bellina, F. Resta, M. Mariotti, and M. Massimini. 2009. Natural Frequencies of Human Corticothalamic Circuits. *Journal of Neuroscience* 29 (24):7679–7685.
- Rosenbaum, D. A. 2007. *Matlab for Behavioral Scientists*. New York: Taylor & Francis Group.
- Rousselet, G. A. 2012. Does Filtering Preclude Us from Studying ERP Time-Courses? *Frontiers in Psychology* 3:131.
- Royston, P., D. G. Altman, and W. Sauerbrei. 2006. Dichotomizing Continuous Predictors in Multiple Regression: A Bad Idea. *Statistics in Medicine* 25 (1):127–141.
- Salinas, E., and T. J. Sejnowski. 2001. Correlated Neuronal Activity and the Flow of Neural Information. *Nature Reviews. Neuroscience* 2 (8):539–550.
- Sauseng, P., B. Griesmayr, R. Freunberger, and W. Klimesch. 2010. Control Mechanisms in Working Memory: A Possible Function of EEG Theta Oscillations. *Neuroscience and Biobehavioral Reviews* 34 (7):1015–1022.
- Sauseng, P., W. Klimesch, W. R. Gruber, S. Hanslmayr, R. Freunberger, and M. Doppelmayr. 2007. Are Event-Related Potential Components Generated by Phase Resetting of Brain Oscillations? A Critical Discussion. *Neuroscience* 146 (4):1435–1444.
- Schalk, G., and E. C. Leuthardt. 2011. Brain-Computer Interfaces Using Electrocorticographic Signals. *IEEE Reviews of Biomedical Engineering* 4:140–154.
- Scheffer-Teixeira, R., H. Belchior, F. V. Caixeta, B. C. Souza, S. Ribeiro, and A. B. Tort. 2012. Theta Phase Modulates Multiple Layer-Specific Oscillations in the CA1 Region. *Cerebral Cortex* 22 (10):2404–2414.
- Scherg, M. 1990. Fundamentals of Dipole Source Potential Analysis. In *Advances in Audiology, Auditory Evoked Magnetic Fields and Electric Potentials*, ed. F. Grandori, M. Hoke, and G. L. Romani, 40–69. Basel: Karger.
- Scherg, M., T. Bast, and P. Berg. 1999. Multiple Source Analysis of Interictal Spikes: Goals, Requirements, and Clinical Value. *Journal of Clinical Neurophysiology* 16 (3):214–224.
- Schiff, S. J. 2005. Dangerous Phase. *Neuroinformatics* 3 (4):315–318.

- Schippers, M. B., A. Roebroeck, R. Renken, L. Nanetti, and C. Keysers. 2010. Mapping the Information Flow from One Brain to Another During Gestural Communication. *Proceedings of the National Academy of Sciences of the United States of America* 107 (20):9388–9393.
- Schreiber, T. 2000. Measuring Information Transfer. *Physical Review Letters* 85 (2):461–464.
- Scott, David W. 2010. Scott's Rule. *Wiley Interdisciplinary Reviews: Computational Statistics* 2 (4):497–502.
- Sekihara, K., and S. S. Nagarajan. 2008. *Adaptive Spatial Filters for Electromagnetic Brain Imaging*. London: Springer.
- Sekihara, K., S. S. Nagarajan, D. Poeppel, and A. Marantz. 2004. Asymptotic SNR of Scalar and Vector Minimum-Variance Beamformers for Neuromagnetic Source Reconstruction. *IEEE Transactions on Bio-Medical Engineering* 51 (10):1726–1734.
- Seth, A. K. 2010a. A Matlab Toolbox for Granger Causal Connectivity Analysis. *Journal of Neuroscience Methods* 186 (2):262–273.
- Seth, A. K. 2010b. Measuring Autonomy and Emergence via Granger Causality. *Artificial Life* 16 (2):179–196.
- Seth, A. K., P. Chorley, and L. C. Barnett. 2013. Granger Causality Analysis of fMRI BOLD Signals Is Invariant to Hemodynamic Convolution but Not Downsampling. *NeuroImage* 65:540–555.
- Shah, A. S., S. L. Bressler, K. H. Knuth, M. Ding, A. D. Mehta, I. Ulbert, and C. E. Schroeder. 2004. Neural Dynamics and the Fundamental Mechanisms of Event-Related Brain Potentials. *Cerebral Cortex* 14 (5):476–483.
- Shannon, C. E., and W. Weaver. 1949. *The Mathematical Theory of Communication*. Urbana, IL: University of Illinois Press.
- Shepard, D. 1968. A Two-Dimensional Interpolation Function for Irregularly-Spaced Data. Paper presented at the ACM National Conference, New York.
- Siegel, M., T. H. Donner, R. Oostenveld, P. Fries, and A. K. Engel. 2008. Neuronal Synchronization along the Dorsal Visual Pathway Reflects the Focus of Spatial Attention. *Neuron* 60 (4):709–719.
- Simmons, J. P., L. D. Nelson, and U. Simonsohn. 2011. False-Positive Psychology: Undisclosed Flexibility in Data Collection and Analysis Allows Presenting Anything as Significant. *Psychological Science* 22 (11):1359–1366.
- Singer, W. 1993. Synchronization of Cortical Activity and Its Putative Role in Information Processing and Learning. *Annual Review of Physiology* 55:349–374.
- Singh, K. D. 2012. Which “Neural Activity” Do You Mean? fMRI, MEG, Oscillations and Neurotransmitters. *NeuroImage* 62 (2):1121–1130.
- Sinkkonen, J., H. Tiitinen, and R. Naatanen. 1995. Gabor Filters: An Informative Way for Analysing Event-Related Brain Activity. *Journal of Neuroscience Methods* 56 (1):99–104.

- Slepian, D. 1978. Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty—V: The Discrete Case. *Bell System Technical Journal* 57:1371–1430.
- Smit, D. J., D. I. Boomsma, H. G. Schnack, H. E. Hulshoff Pol, and E. J. de Geus. 2012. Individual Differences in EEG Spectral Power Reflect Genetic Variance in Gray and White Matter Volumes. *Twin Research and Human Genetics* 15 (3):384–392.
- Smith, J. O. 2007. *Mathematics of the Discrete Fourier Transform (DFT) with Audio Applications*. San Diego, CA: W3K Publishing.
- Sohal, V. S., F. Zhang, O. Yizhar, and K. Deisseroth. 2009. Parvalbumin Neurons and Gamma Rhythms Enhance Cortical Circuit Performance. *Nature* 459 (7247):698–702.
- Sporns, O. 2011. *Networks of the Brain*. Cambridge, MA: MIT Press.
- Squyres, S. 2005. *Roving Mars: Spirit, Opportunity, and the Exploration of the Red Planet*. New York: Hyperion Books.
- Srinivasan, R., F. A. Bibi, and P. L. Nunez. 2006. Steady-State Visual Evoked Potentials: Distributed Local Sources and Wave-Like Dynamics Are Sensitive to Flicker Frequency. *Brain Topography* 18 (3):167–187.
- Srinivasan, R., E. Fornari, M. G. Knyazeva, R. Meuli, and P. Maeder. 2007a. fMRI Responses in Medial Frontal Cortex That Depend on the Temporal Frequency of Visual Input. *Experimental Brain Research* 180 (4):677–691.
- Srinivasan, R., W. R. Winter, J. Ding, and P. L. Nunez. 2007b. EEG and MEG Coherence: Measures of Functional Connectivity at Distinct Spatial Scales of Neocortical Dynamics. *Journal of Neuroscience Methods* 166 (1):41–52.
- Srinivasan, R., W. R. Winter, and P. L. Nunez. 2006. Source Analysis of EEG Oscillations Using High-Resolution EEG and MEG. *Progress in Brain Research* 159:29–42.
- Stam, C. J. 2010. Use of Magnetoencephalography (MEG) to Study Functional Brain Networks in Neurodegenerative Disorders. *Journal of the Neurological Sciences* 289 (1–2):128–134.
- Stam, C. J., G. Nolte, and A. Daffertshofer. 2007. Phase Lag Index: Assessment of Functional Connectivity from Multichannel EEG and MEG with Diminished Bias from Common Sources. *Human Brain Mapping* 28 (11):1178–1193.
- Steinstrater, O., S. Sillekens, M. Junghofer, M. Burger, and C. H. Wolters. 2010. Sensitivity of Beam-former Source Analysis to Deficiencies in Forward Modeling. *Human Brain Mapping* 31 (12):1907–1927.
- Stemmer, B., M. Vihla, and R. Salmelin. 2004. Activation of the Human Sensorimotor Cortex during Error-Related Processing: A Magnetoencephalography Study. *Neuroscience Letters* 362 (1):44–47.
- Stephens, M. A. 1969. Tests for the Von Mises Distribution. *Biometrika* 56 (1):149–160.
- Steriade, M. 2005. Cellular Substrates of Brain Rhythms. In *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*, ed. E. Niedermeyer and F. H. L. Da Silva. Philadelphia: Lippincott Williams & Wilkins.

- Steriade, M. 2006. Grouping of Brain Rhythms in Corticothalamic Systems. *Neuroscience* 137 (4):1087–1106.
- Steuer, R., J. Kurths, C. O. Daub, J. Weise, and J. Selbig. 2002. The Mutual Information: Detecting and Evaluating Dependencies between Variables. *Bioinformatics (Oxford, England)* 18 (Suppl 2):S231–S240.
- Stinstra, J. G., and M. J. Peters. 1998. The Volume Conductor May Act as a Temporal Filter on the ECG and EEG. *Medical and Biological Engineering and Computing* 36 (6):711–716.
- Stockwell, R. G., L. Mansinha, and R. P. Lowe. 1996. Localization of the Complex Spectrum: The S Transform. *IEEE Transactions on Signal Processing* 44 (4):998–1001.
- Stoffers, D., J. L. Bosboom, J. B. Deijen, E. C. Wolters, C. J. Stam, and H. W. Berendse. 2008. Increased Cortico-Cortical Functional Connectivity in Early-Stage Parkinson's Disease: A MEG Study. *NeuroImage* 41 (2):212–222.
- Stone, J. L., M. Calderon-Arnulphi, K. S. Watson, K. Patel, N. S. Mander, N. Suss, J. Fino, and J. R. Hughes. 2009. Brainstem Auditory Evoked Potentials—a Review and Modified Studies in Healthy Subjects. *Journal of Clinical Neurophysiology* 26 (3):167–175.
- Sturges, H. 1926. The Choice of a Class-Interval. *Journal of the American Statistical Association* 21:65–66.
- Sun, W., and Y. Dan. 2009. Layer-Specific Network Oscillation and Spatiotemporal Receptive Field in the Visual Cortex. *Proceedings of the National Academy of Sciences of the United States of America* 106 (42):17986–17991.
- Sweeney-Reed, C. M., and S. J. Nasuto. 2007. A Novel Approach to the Detection of Synchronization in EEG Based on Empirical Model Decomposition. *Journal of Computational Neuroscience* 23:79–111.
- Tallon-Baudry, C., and O. Bertrand. 1999. Oscillatory Gamma Activity in Humans and Its Role in Object Representation. *Trends in Cognitive Sciences* 3 (4):151–162.
- Tass, P., M. G. Rosenblum, J. Weule, J. Kurths, A. Pikovsky, J. Volkmann, A. Schnitzler, et al. 1998. Detection of N:M Phase Locking from Noisy Data: Application to Magnetoencephalography. *Physical Review Letters* 81 (5):3291–3294.
- Taylor, A., and D. J. Higham. 2009. Contest: A Controllable Test Matrix Toolbox for Matlab. [TOMS] *ACM Transactions on Mathematical Software* 35 (4):1–16.
- Telesford, Q. K., K. E. Joyce, S. Hayasaka, J. H. Burdette, and P. J. Laurienti. 2011. The Ubiquity of Small-World Networks. *Brain Connect* 1 (5):367–375.
- Tenke, C. E., and J. Kayser. 2005. Reference-Free Quantification of EEG Spectra: Combining Current Source Density (CSD) and Frequency Principal Components Analysis (FPCA). *Clinical Neurophysiology* 116 (12):2826–2846.
- Theiler, J., S. Eubank, A. Longtin, B. Galdrikian, and J. D. Farmer. 1992. Testing for Nonlinearity in Time Series: The Method of Surrogate Data. *Physica D* 58 (1):74–94.
- Thomson, D. J. 1982. Spectrum Estimation and Harmonic Analysis. *Proceedings of the IEEE* 70:1055–1096.

- Thut, G., P. G. Schyns, and J. Gross. 2011. Entrainment of Perceptually Relevant Brain Oscillations by Non-invasive Rhythmic Stimulation of the Human Brain. *Frontiers in Psychology* 2:170.
- Thut, G., D. Veniero, V. Romei, C. Miniussi, P. Schyns, and J. Gross. 2011. Rhythmic TMS Causes Local Entrainment of Natural Oscillatory Signatures. *Current Biology* 21 (14):1176–1185.
- Tong, F., and M. S. Pratte. 2012. Decoding Patterns of Human Brain Activity. *Annual Review of Psychology* 63:483–509.
- Tort, A. B., R. Komorowski, H. Eichenbaum, and N. Kopell. 2010. Measuring Phase-Amplitude Coupling between Neuronal Oscillations of Different Frequencies. *Journal of Neurophysiology* 104 (2):1195–1210.
- Truccolo, W. A., M. Ding, K. H. Knuth, R. Nakamura, and S. L. Bressler. 2002. Trial-to-Trial Variability of Cortical Evoked Responses: Implications for the Analysis of Functional Connectivity. *Clinical Neurophysiology* 113 (2):206–226.
- Tufte, E. R. 1983. *The Visual Display of Quantitative Information*. Cheshire, CT: Graphics Press.
- Tuk, M. A., D. Trampe, and L. Warlop. 2011. Inhibitory Spillover: Increased Urination Urgency Facilitates Impulse Control in Unrelated Domains. *Psychological Science* 22 (5):627–633.
- Uhlhaas, P. J., and W. Singer. 2011. The Development of Neural Synchrony and Large-Scale Cortical Networks during Adolescence: Relevance for the Pathophysiology of Schizophrenia and Neurodevelopmental Hypothesis. *Schizophrenia Bulletin* 37 (3):514–523.
- van Drongelen, W. 2006. *Signal Processing for Neuroscientists: An Introduction to the Analysis of Physiological Signals*. Burlington, MA: Academic Press.
- Vanhatalo, S., J. M. Palva, M. D. Holmes, J. W. Miller, J. Voipio, and K. Kaila. 2004. Infraslow Oscillations Modulate Excitability and Interictal Epileptic Activity in the Human Cortex During Sleep. *Proceedings of the National Academy of Sciences of the United States of America* 101 (14):5053–5057.
- Van Veen, B. D., W. van Drongelen, M. Yuchtman, and A. Suzuki. 1997. Localization of Brain Electrical Activity via Linearly Constrained Minimum Variance Spatial Filtering. *IEEE Transactions on Bio-Medical Engineering* 44 (9):867–880.
- van Vugt, M. K., P. B. Sederberg, and M. J. Kahana. 2007. Comparison of Spectral Analysis Methods for Characterizing Brain Oscillations. *Journal of Neuroscience Methods* 162 (1–2):49–63.
- Varela, F., J. P. Lachaux, E. Rodriguez, and J. Martinerie. 2001. The Brainweb: Phase Synchronization and Large-Scale Integration. *Nature Reviews. Neuroscience* 2 (4):229–239.
- Vialatte, F. B., M. Maurice, J. Dauwels, and A. Cichocki. 2010. Steady-State Visually Evoked Potentials: Focus on Essential Paradigms and Future Perspectives. *Progress in Neurobiology* 90 (4):418–438.
- Vinck, M., R. Oostenveld, M. van Wingerden, F. Battaglia, and C. M. Pennartz. 2011. An Improved Index of Phase-Synchronization for Electrophysiological Data in the Presence of Volume-Conduction, Noise and Sample-Size Bias. *NeuroImage* 55 (4):1548–1565.

- Vinck, M., M. van Wingerden, T. Womelsdorf, P. Fries, and C. M. Pennartz. 2010. The Pairwise Phase Consistency: A Bias-Free Measure of Rhythmic Neuronal Synchronization. *NeuroImage* 51 (1): 112–122.
- Viriyopase, A., I. Bojak, M. Zeitler, and S. Gielen. 2012. When Long-Range Zero-Lag Synchronization Is Feasible in Cortical Networks. *Front Comput Neurosci* 6:49.
- Von Stein, A., and J. Sarnthein. 2000. Different Frequencies for Different Scales of Cortical Integration: From Local Gamma to Long Range Alpha/Theta Synchronization. *International Journal of Psychophysiology* 38 (3):301–313.
- Voytek, B., R. T. Canolty, A. Shestyuk, N. E. Crone, J. Parvizi, and R. T. Knight. 2010. Shifts in Gamma Phase-Amplitude Coupling Frequency from Theta to Alpha over Posterior Cortex during Visual Tasks. *Frontiers in Human Neuroscience* 4:191.
- Voytek, B., M. D'Esposito, N. Crone, and R. T. Knight. 2013. A Method for Event-Related Phase/Amplitude Coupling. *NeuroImage* 64:416–424.
- Vreugdenhil, M., E. Bracci, and J. G. Jefferys. 2005. Layer-Specific Pyramidal Cell Oscillations Evoked by Tetanic Stimulation in the Rat Hippocampal Area CA1 in Vitro and in Vivo. *Journal of Physiology* 562 (1):149–164.
- Vyssotski, A. L., G. Dell'Osso, G. Dell'Ariccia, A. N. Abramchuk, A. N. Serkov, A. V. Latanov, A. Loizzo, et al. 2009. EEG Responses to Visual Landmarks in Flying Pigeons. *Current Biology* 19 (14):1159–1166.
- Wallisch, P. 2009. *Matlab for Neuroscientists: An Introduction to Scientific Computing in Matlab*. Burlington, MA: Elsevier/Academic Press.
- Wang, C., I. Ulbert, D. L. Schomer, K. Marinkovic, and E. Halgren. 2005. Responses of Human Anterior Cingulate Cortex Microdomains to Error Detection, Conflict Monitoring, Stimulus-Response Mapping, Familiarity, and Orienting. *Journal of Neuroscience*. 25 (3):604–613.
- Wang, X. J. 2010. Neurophysiological and Computational Principles of Cortical Rhythms in Cognition. *Physiological Reviews* 90 (3):1195–1268.
- Wang, X., Y. Chen, and M. Ding. 2008. Estimating Granger Causality after Stimulus Onset: A Cautionary Note. *NeuroImage* 41 (3):767–776.
- Watts, D. J., and S. H. Strogatz. 1998. Collective Dynamics of “Small-World” Networks. *Nature* 393 (6684):440–442.
- Weinberg, A., D. N. Klein, and G. Hajcak. 2012. Increased Error-Related Brain Activity Distinguishes Generalized Anxiety Disorder with and without Comorbid Major Depressive Disorder. *Journal of Abnormal Psychology* 121 (4):885–896.
- Welch, P. D. 1967. The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging over Short, Modified Periodograms. *IEEE Transactions on Audio and Electroacoustics* 15 (2):70–73.

- Wendling, F., K. Ansari-Asl, F. Bartolomei, and L. Senhadji. 2009. From EEG Signals to Brain Connectivity: A Model-Based Evaluation of Interdependence Measures. *Journal of Neuroscience Methods* 183 (1):9–18.
- Whittingstall, K., and N. K. Logothetis. 2009. Frequency-Band Coupling in Surface EEG Reflects Spiking Activity in Monkey Visual Cortex. *Neuron* 64 (2):281–289.
- Wierda, S. M., H. van Rijn, N. A. Taatgen, and S. Martens. 2012. Pupil Dilation Deconvolution Reveals the Dynamics of Attention at High Temporal Resolution. *Proceedings of the National Academy of Sciences of the United States of America* 109 (22):8456–8460.
- Williams, J., Ramaswamy, D., and Oulhaj, A. 2006. 10 Hz Flicker Improves Recognition Memory in Older People. *BMC Neuroscience* 7 (21):7-21.
- Wilmer, A., M. de Lussanet, and M. Lappe. 2012. Time-Delayed Mutual Information of the Phase as a Measure of Functional Connectivity. *PLoS One* 7 (9):e44633.
- Wimber, M., A. Maass, T. Staudigl, A. Richardson-Klavehn, and S. Hanslmayr. 2012. Rapid Memory Reactivation Revealed by Oscillatory Entrainment. *Current Biology* 22 (16):1482–1486.
- Winter, W. R., P. L. Nunez, J. Ding, and R. Srinivasan. 2007. Comparison of the Effect of Volume Conduction on EEG Coherence with the Effect of Field Spread on MEG Coherence. *Statistics in Medicine* 26 (21):3946–3957.
- Yamaguchi, Y., N. Sato, H. Wagatsuma, Z. Wu, C. Molter, and Y. Aota. 2007. A Unified View of Theta-Phase Coding in the Entorhinal-Hippocampal System. *Current Opinion in Neurobiology* 17 (2):197–204.
- Yarkoni, T., R. A. Poldrack, T. E. Nichols, D. C. Van Essen, and T. D. Wager. 2011. Large-Scale Automated Synthesis of Human Functional Neuroimaging Data. *Nature Methods* 8 (8):665–670.
- Yeung, N., R. Bogacz, C. B. Holroyd, S. Nieuwenhuis, and J. D. Cohen. 2007. Theta Phase Resetting and the Error-Related Negativity. *Psychophysiology* 44 (1):39–49.
- Yuval-Greenberg, S., and L. Y. Deouell. 2009. The Broadband-Transient Induced Gamma-Band Response in Scalp EEG Reflects the Execution of Saccades. *Brain Topography* 22 (1):3–6.
- Yuval-Greenberg, S., O. Tomer, A. S. Keren, I. Nelken, and L. Y. Deouell. 2008. Transient Induced Gamma-Band Response in EEG as a Manifestation of Miniature Saccades. *Neuron* 58 (3):429–441.
- Zaeble, T., S. Rach, and C. S. Herrmann. 2010. Transcranial Alternating Current Stimulation Enhances Individual Alpha Activity in Human EEG. *PLoS One* 5 (11):e13766.
- Zar, J. H. 1999. *Biostatistical Analysis*. Upper Saddle River, NJ: Prentice Hall.
- Zhang, X., K. M. Kendrick, H. Zhou, Y. Zhan, and J. Feng. 2012. A Computational Study on Altered Theta-Gamma Coupling during Learning and Phase Coding. *PLoS One* 7 (6):e36472.
- Zhou, B., and W. H. Wong. 2011. A Bootstrap-Based Non-parametric ANOVA Method with Applications to Factorial Microarray Data. *Statistica Sinica* 21:495–514.

Index

- 1/f scaling, 218
Additive, 56
Advantages and limitations, 20–22
Akaike information criterion, 378
Amplitude asymmetry, 57
Analysis terms, 10, 243, 265, 333, 371, 526–527
Analytic signal, 175
Anscombe's quartet, 358
Asking colleagues, 48
Autocorrelation, 330, 468
Autoregression, 211, 372
Autoregression model order, 372, 378–379
Background activity, 34, 55, 218, 220
Bad electrodes, 84
Baseline time period, 221, 231–232, 384
Bayes information criterion, 378–379
Beamforming, 313
Beta-band, 92
Binarizing threshold, 440
Biophysics, 51
Bivariate connectivity, 319
Bivariate statistical analyses, 477
Blinks, 89–90
Bonferroni correction, 453, 468–469
Bootstrapping, 460
Brain-behavior correlations, 482
Brain localization, 27
Buffer zones, 76, 190, 198, 301
Butterworth filter, 189
Cartesian plane, 112
Causal evidence, 57–59, 320–321, 541
Center frequency, 144, 180
Chi-square, 376, 385
Circular inferences, 420–421, 493, 505, 517
Clean data, 85
Clear thinking, 44
Cluster-based corrections, 472
Clustering coefficient, 436–437
Cognitive artifacts, 87, 93
Cognitive electrophysiology, 3, 15–16, 532
Color figures, 509
Comfortable chair, 69
Complex dot product, 160
Complexity, 300
Complex Morlet wavelet, 151, 157, 266
Complex number, 151
Computation time test, 139
Condition-average baseline, 233, 248
Condition differences, 481, 505
Conjugate, 162, 345
Connectivity, 286, 319
Connectivity degree, 433
Connectivity over time vs. trials, 338, 351, 362, 402, 524
Control condition, 233
Convolution kernel, 113

- Convolution signal, 113
- Convolution theorem, 135, 180
- Corkscrew, 151
- Correlation significance, 490
- Cosine wave, 155
- Costs of EEG, 29
- Covariance matrix, 293, 313
- Critical ITPC, 488
- Cross-correlation, 118, 362
- Cross-covariance, 118
- Cross-frequency coupling, 403
- Cross-spectral density, 344
- Current source density, 276
- DC component. *See* Zero frequency
- Deblurring, 279
- Decibel, 220
- Deep brain sources, 52
- Degree. *See* Connectivity degree
- Dipole, 53, 308, 311
- Directed connectivity, 321
- Discrete time Fourier transform, 124–125
- Discretizing continuous data, 483
- Distributed source imaging, 312–314
- Dot product, 111, 125, 149, 159
- Double-dipping. *See* Circular inferences
- Downsampling, 239, 367, 479
- Edge artifacts, 76, 183, 190, 197
- Edges in graphs, 429
- Electrode localization, 68
- Electrooculogram, 91
- EMG (electromyography), 68
- Empirical mode decomposition, 212
- Entropy, 389
- Epochs, 75
- Erdos-Renyi graphs, 444
- erpviewerx, 38
- Error variance, 376
- Euler’s formula, 125, 155, 242, 244, 411
- Event markers, 62
- Event-related potentials, 18, 97, 293
- Excitatory cells, 54
- Exercises, 11
- Experiment design, 538
- Exploratory analyses
 - as an approach to science, 531
 - connectivity analyses, 324, 351, 366
 - cross-frequency coupling, 411
 - multiple comparisons, 270
 - statistics, 452, 515
- Eye movements, 91
- Eye tracker, 68
- False discovery rate, 475
- Fast Fourier transform, 130, 416
- Femtotesla, 17
- Figure labeling, 511
- Filtering, 80, 98–99, 147, 178–179, 188
- Filter kernel construction, 181
- Filter order, 181
- Finite impulse response filter, 179
- fir1, 181, 184, 186
- firls, 181–182, 184
- Fisher-Z transform, 360
- Five-step plan, 38
- Flicker effect. *See* Frequency tagging
- Fly swatter, 521
- fMRI, 17, 60
- Force grips, 68
- Forward solution, 307
- Fourier representation, 121
- Frequency, 31, 126
- Frequency-domain filter, 135
- Frequency precision
 - definition, 24–25
 - filtering, 183
- Fourier transform, 134, 199
- wavelets, 142, 145, 169, 266
- Frequency resolution, 126, 199, 381
- Frequency smoothing, 145, 165, 206
- Frequency tagging, 101–102
- Full width at half maximum, 172
- Functional localization, 16

- Gabor wavelet, 141
G and H matrices, 279
Gamma, 15, 33, 207, 214
Gaussian, 144, 157, 197
Global field power, 100
Granger causality, 371, 376
Group-level statistics, 450–451
Gv-test, 353–354, 490
- Hamming window, 184, 197
Hann window, 197
Hertz (Hz), 31, 121
High-dimensional vectors, 113
Hilbert transform, 176, 266
Histogram bins, 392–394
Hubs, 433–434
Hypercube, 35, 431, 505, 516
Hypothesis-driven analyses
 analysis strategies, 495, 498
 as an approach to science, 530
 connectivity analyses, 323, 351,
 363
 cross-frequency coupling, 411
 multiple comparisons, 270
 statistics, 452, 514
- Ideal filter, 181–182, 186
Image, 167
Imaginary coherence, 346
Imaginary numbers, 153, 176
Independent components analysis, 87,
 304
Individual differences, 499
Induced power, 259
Infinite impulse response filter. *See* Finite impulse
 response filter
Inhibiting blinks, 90
Inhibitory cells, 54
Interpolation, 83–84
Interpreting time-frequency results, 272
Inter-site phase clustering, 333
Interstimulus interval, 63–64
- Intertrial interval, 64, 301
Intertrial phase clustering, 242, 266, 268, 333,
 419, 487
Inverse Fourier transform, 129
Inverse hyperbolic transform. *See* Fisher-Z
Inverse inference, 542
Inverse problem, 310
- Joint entropy, 396
- Keyboard, 69
- Laplacian. *See* Surface Laplacian
Least-squares fit, 237, 370, 484–485
Legendre polynomial, 279
Level-1 and level-2 statistics, 450
Leverage, 359
Limitations, 269. *See also* Advantages and
 limitations
Long blinks, 90
LORETA, 312
- Magnitude, 154, 161
Magnitude squared coherence. *See* Spectral
 coherence
Matching pursuit, 213
Matching trial count, 79, 248
Matlab, 7–8. *See also* Programming tips
Matrices and graphs, 429
Maximum frequency, 165
Mean, 227
Median, 227–229, 236
Meta-permutation test, 468
Microsaccades, 91
Microstates, 104
Microvolts, 17, 278
Minimum norm estimate, 312
Morelet wavelet, 141–142
Multidimensional, 16
Multiple comparisons, 453, 468
Multiple regression, 485
Multitaper, 203, 266

- Multivariate analyses, 320, 388, 536
 Mutual information, 396
- N-dimensional space, 113
 Negative frequencies, 128, 176
 Neural oscillations, 33, 41, 54
 Noise, 297
 Nonparametric statistics, 456
 Non-phase-locked
 computing, 259, 383, 419
 connectivity, 339
 covariance matrix, 294
 definition, 34, 55, 97
 interpreting null results, 528
 Non-uniformly distributed, 242
 Normally distributed, 219, 358–359, 456
 Null hypothesis, 460, 461
 Number of subjects, 501
 Number of trials. *See* Trial count
 Nyquist theorem, 66–67, 126, 145, 176, 181
- Oculomotor activity, 90–91
 Online Matlab code, 9
 Optogenetics, 59
 Ordinary least squares. *See* Least-squares fit
 Oscillations, 270, 272
 Outlier, 228, 359
- Parametric statistics, 456, 493
 Partial correlations, 331, 366
 Partial errors, 94–95
 Path length, 440
 Peak frequency, 144, 158, 161, 170, 180
 Pearson correlation, 357
 P-episode, 213
 Percent change, 223
 Peripheral sensors, 540
 Permutation testing
 cross-frequency coupling, 415
 Granger prediction, 385
 group-level analyses, 493
 justifications and mechanisms, 459–477
- mutual information, 405
 PCA, 289
 small-world networks, 444
 wITPC, 254
 Phase, 31, 121, 200, 204, 242
 Phase-amplitude coupling, 413
 Phase clustering, 242
 Phase lag, 149, 320, 329, 334, 346, 352
 Phase lag index, 346
 Phase-locked, 34, 55, 259, 294, 339
 Phase-phase coupling, 425
 Phase quadrature, 176
 Phase reset, 57
 Phase slope index, 348
 Pilot testing, 61
 Pixel-based corrections, 470
 Polar notation, 153, 159
 Positive frequencies, 128, 176
 Power, 31
 Power–power correlations, 365, 410
 Practical significance, 454
 Preferred phase angle, 245
 Preprocessing, 73
 Principal components analysis, 291
 Programming tips
 clustering, 473
 clustering coefficient, 438
 convolution, 138, 173
 correlations, 368–369, 484
 cross-frequency coupling, 423, 427
 downsampling, 367
 Fisher-Z, 361
 general advice, 48
 hilbert, 177
 imagesc, 167
 ITPC, 245
 mutual information, 391
 scalp Laplacian, 283
 spectral power, 162
 t-statistics, 491
 P-values, 453, 463–464
 Pyramidal cells, 54

- Radial dipoles, 53, 277
Rank, 299
Raw power, 218
Rayleigh's Z, 488
Reaction times, 79, 482
Rectangular notation, 153
Referencing data, 82
Reflection, 77
Rejecting data, 80–81
Replications, 529, 545
Resting state, 301
- Sample data, 9
Sampling rate, 66
Short-time FFT, 196, 204, 266
Signal processing toolbox, 189
Signal-to-noise ratio, 234–235, 336, 362
Signal-to-noise trade-off, 74
Sine wave, 121–122, 155
Single-trial baseline, 230
Slepian tapers, 205
Small-world networks, 441–442
Sources of EEG, 53
Spatial accuracy. *See* Spatial resolution
Spatial filters, 81, 275
Spatial localization, 314
Spatial precision. *See* Spatial resolution
Spatial resolution, 26–27, 280, 288, 314, 270
Spatial scales, 27
Spearman correlation, 237, 357, 398, 404, 483
Specificity, 508
Spectral coherence, 342
Spike-field coherence, 257
Stationarity
 empirical mode decomposition, 212
ERPs, 263, 383
Fourier transform, 132, 195
Granger prediction, 372
phase estimation, 253, 350
wavelets, 171
Statistical significance, 454, 463
- Steady-state evoked potential. *See* Frequency tagging
S-transform, 214
Stripes, 40
Sum of squared errors, 186
Surface Laplacian, 276, 329
Suspicious results, 40, 523
Synchronization, 265, 268
- Tables, 514, 543
Take-home-message, 504
Tangential dipoles, 277
Tapers, 196, 205
Task-related activity, 34
Temporal accuracy. *See* temporal resolution
Temporal autocorrelation, 367
Temporal jitter, 249, 322, 323
Temporal precision
 connectivity results, 336
 cross-frequency coupling, 409
 definition, 24–25
 downsampling, 239
 filtering, 183
 Fourier transform, 199
 Granger prediction, 371
 multitaper, 207
 wavelets, 142, 145, 169, 266
Temporal resolution, 24–25, 239, 367, 371
Temporal smoothing, 206, 232
Terminology. *See* Analysis terms
tfviewervx, 37–38, 505
Theta, 15, 33, 238
Threshold for connectivity, 432
Time segments, 199, 301, 336, 362, 377, 395, 416
Time shifting, 480
Time = 0, 64, 75, 206
Time-domain convolution, 113
Time-frequency representation, 141
Time-frequency trade-off, 34, 141, 199, 206
Time-frequency window, 496
Toolboxes, 6
Topographical localization, 27, 277, 283

Topographical maps, 102–103
Total power, 259
Trade-offs. *See* Time-frequency trade-off
Transcranial alternating current stimulation, 59, 541
Transient activity, 170
Trial averaging, 222
Trial concatenation, 173, 190
Trial count, 65, 79, 237, 246, 248, 340, 501
Type I and Type II errors, 455

Uncertainties, 34, 250

Variable names, 45
Vertices (nodes), 429
Violation of stationarity, 132, 141
Visual inspection, 410, 449
Volume conduction, 275, 286, 326, 346
von Mises distribution, 414, 489
V-test, 352

Wavelet family, 145
Wavelet parameters, 164–172, 269
Weighted phase clustering (wITPC), 253, 342, 411
Weighted phase lag index, 347
Within-subjects statistics, 450–451, 479

Zero frequency, 125, 134
Z-transform, 224

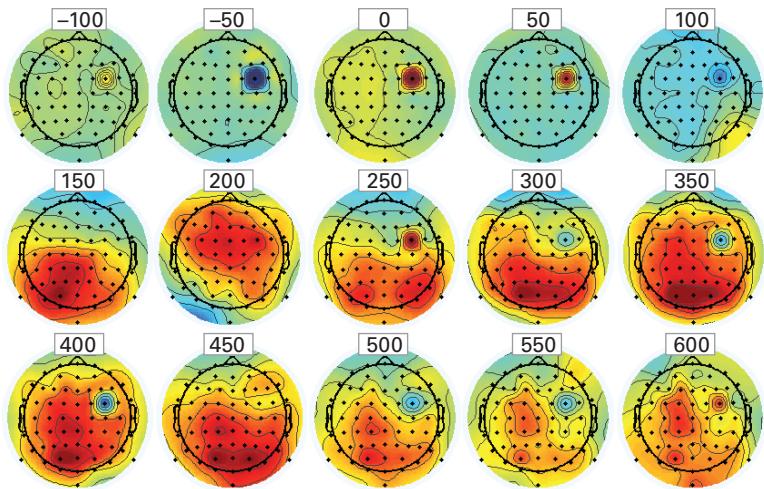


Plate 1 (figure 9.5)

Plotting topographical maps over time facilitates rapid data quality inspection. The numbers in white boxes indicate the latency at which the topographical data are plotted (in milliseconds) with respect to trial onset. These plots show, among other things, that there is one bad electrode. In this case the bad electrode was generated by replacing the true EEG activity at electrode FC4 with randomly generated numbers.

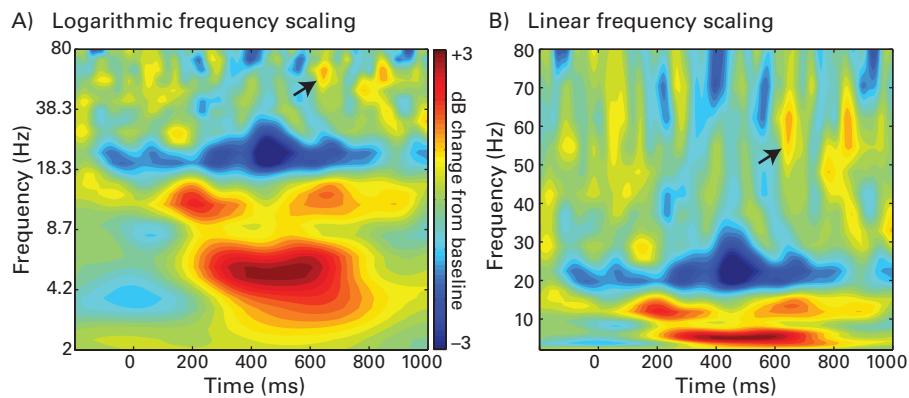


Plate 2 (figure 13.11)

Time-frequency results of the same analyses applied to the same data can look different depending on whether the frequencies in the y -axis are scaled logarithmically (panel A) or linearly (panel B). The black arrows, for example, show the same gamma power burst in both plots.

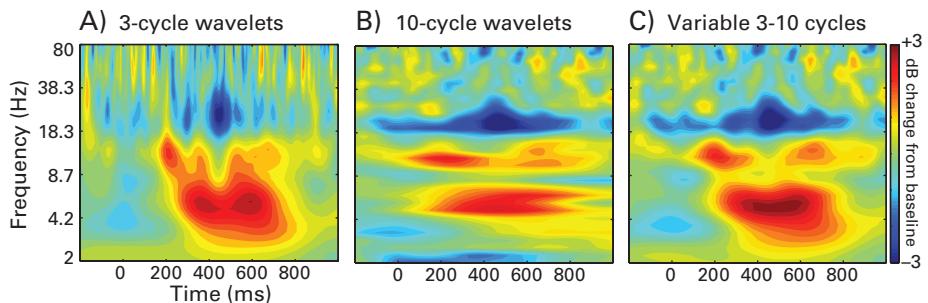


Plate 3 (figure 13.14)

The width of the Gaussian that is used to create the wavelets affects the features of the results that will be obtained from complex Morlet wavelet convolution. Different numbers of cycles can be used to highlight temporal precision (panel A) or frequency precision (panel B). Panel C shows that the balance between temporal and frequency precisions can change as a function of frequency; this increases temporal precision at lower frequencies and increases frequency precision at higher frequencies. The results were transformed to decibel (dB) change relative to a pretrial baseline. Baseline transformations are discussed in depth in chapter 18.

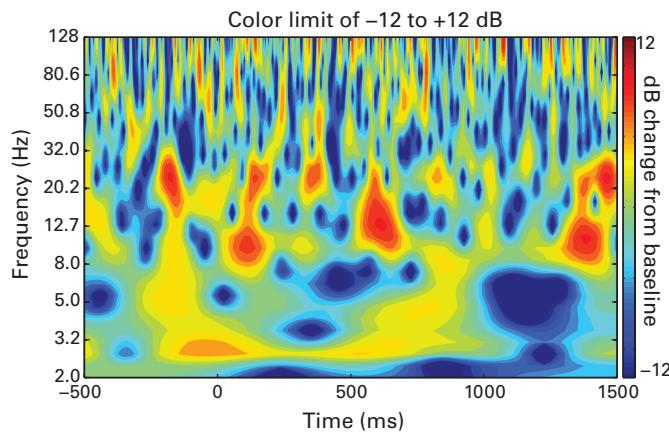


Plate 4 (figure 18.3)

Same time-frequency results that were shown in figure 18.2 but now decibel-transformed relative to a pretrial baseline period of -500 to -200 ms. Note that power across the entire range of frequencies can be compared.

Plate 6 (figure 18.7) (opposite)

Results from the same data and same analysis, but with different baseline normalization methods applied. As with previous figures in this chapter, the color scaling is larger here than what you would normally expect because data from one trial were used.

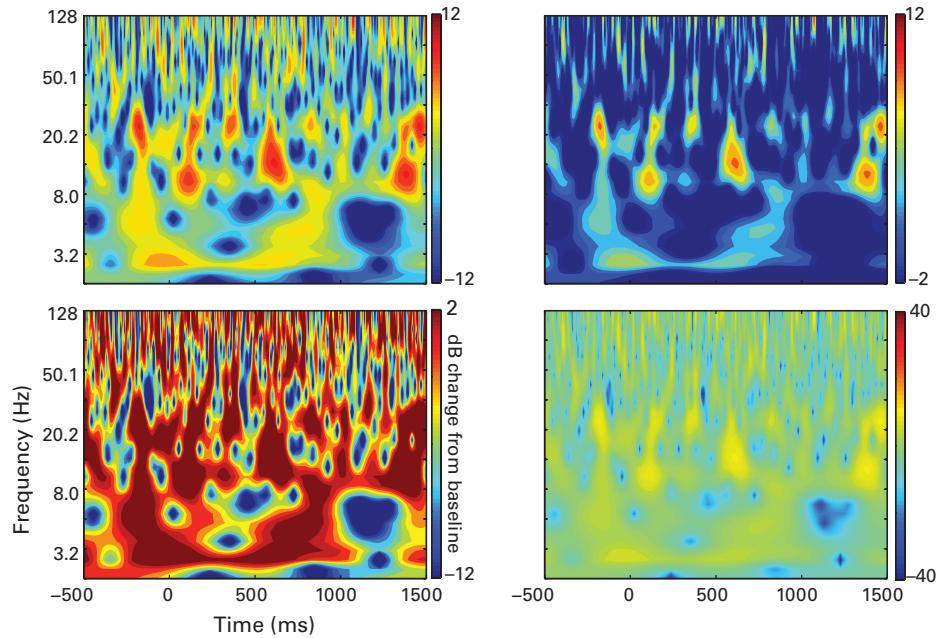
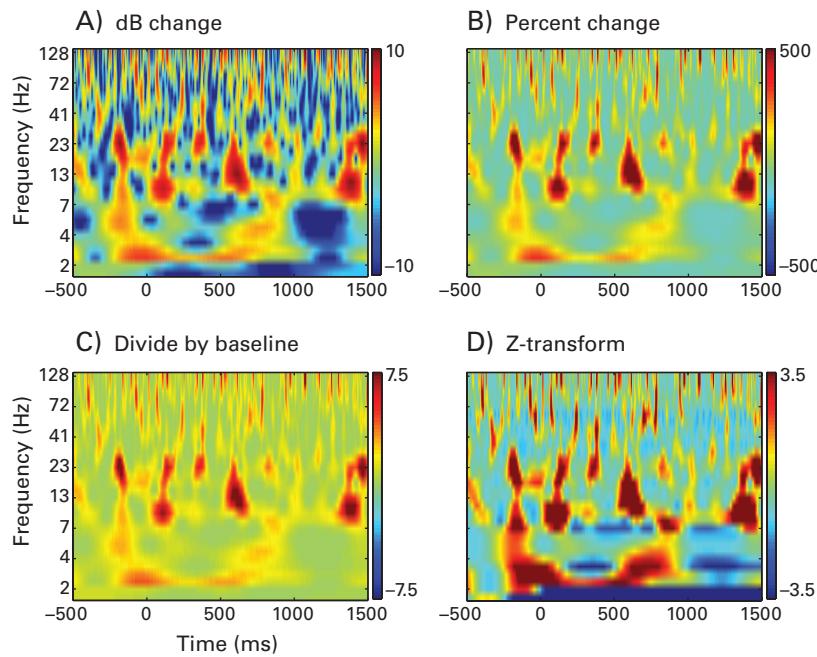


Plate 5 (figure 18.5)

Different color scaling applied to the same time-frequency power results. Symmetric color scaling should be preferred in most situations. Asymmetric color scaling can highlight or obscure different features of the data. Values in the color bars correspond to decibel change from a pretrial baseline of -500 to -200 ms. These data are taken from one trial and one electrode.



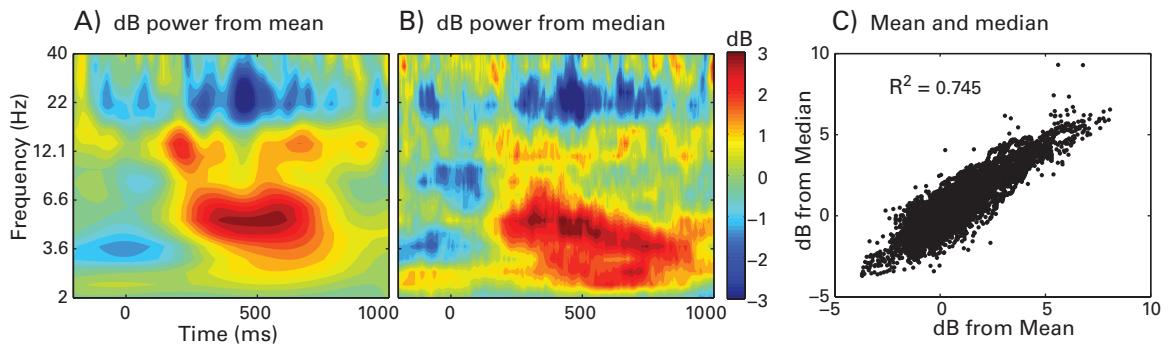


Plate 7 (figure 18.8)

Time-frequency power plots of the same data using mean (panel A) or median (panel B) to combine data across trials. Panel C shows the correlation between the results using mean and using median.

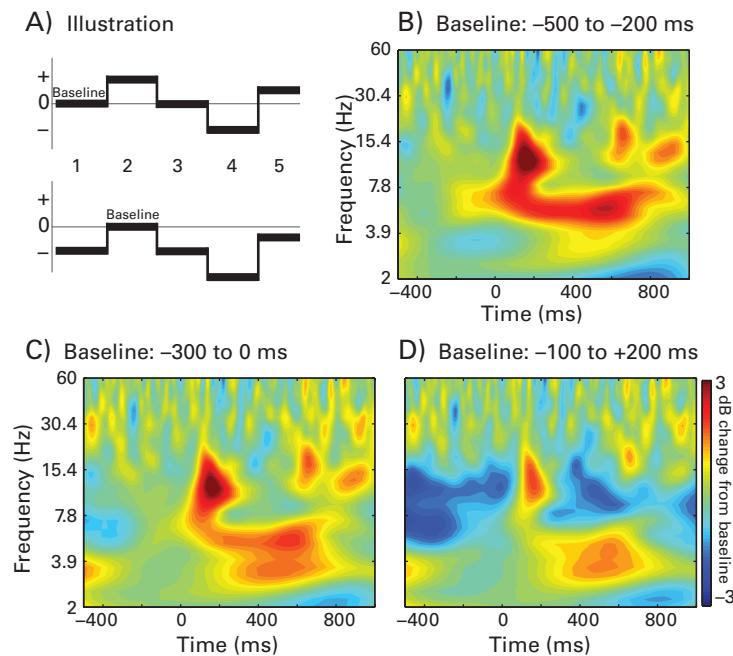


Plate 8 (figure 18.11)

The choice of baseline time period used for power normalization has implications for interpreting the results. Panel A illustrates conceptually that the same activity can be interpreted differently depending on what is used as a baseline period. Panels B-D illustrate this effect in real data. Panel D shows an extreme and unrealistic choice of baseline because it includes poststimulus time but demonstrates that activity present in the baseline period will appear as the inverse of that activity at other time points.

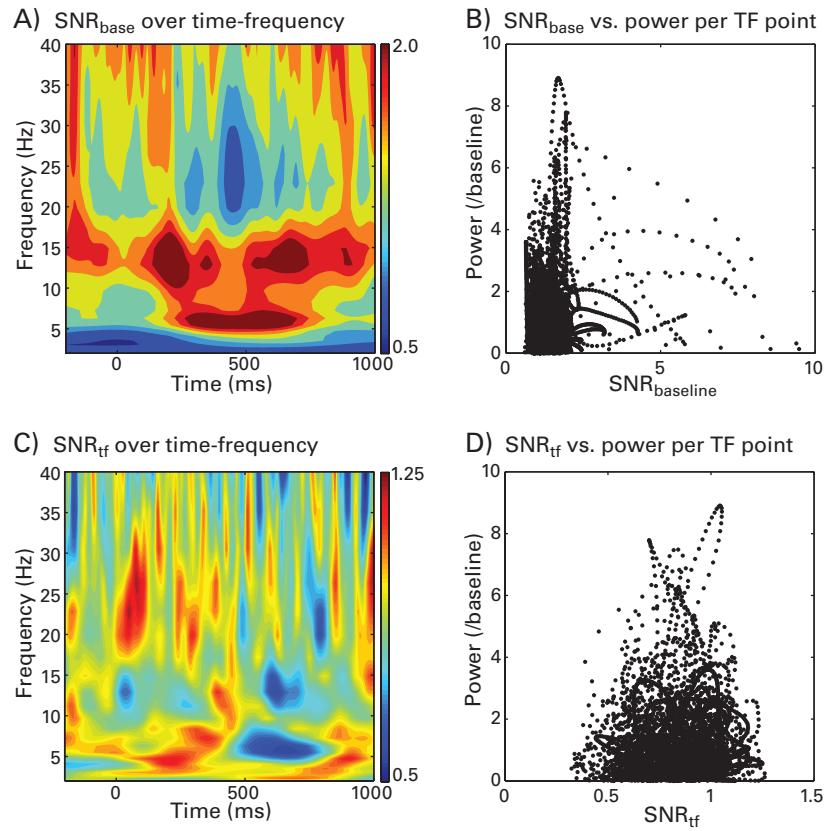


Plate 9 (figure 18.12)

Time-frequency SNR results, using the same time-frequency data that were shown in figure 18.11. Panels A and B show SNR_{base} (SNR relative to the baseline time period), and panels C and D show SNR_{tf} (SNR across trials at each time-frequency point).

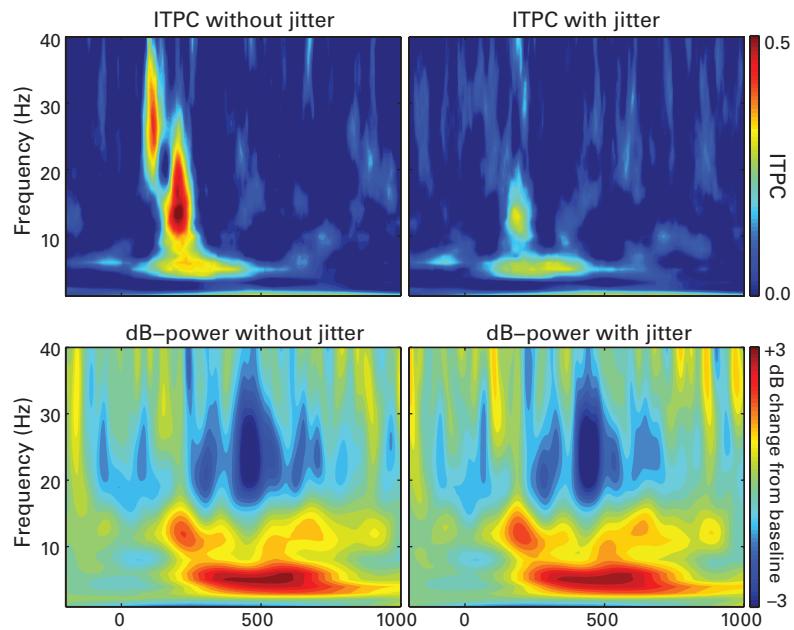


Plate 10 (figure 19.9)

Temporal jitters of less than 40 ms can have deleterious effects on ITPC (top row), particularly at frequencies above 10 Hz. In contrast, temporal jitters have little noticeable effects on power.

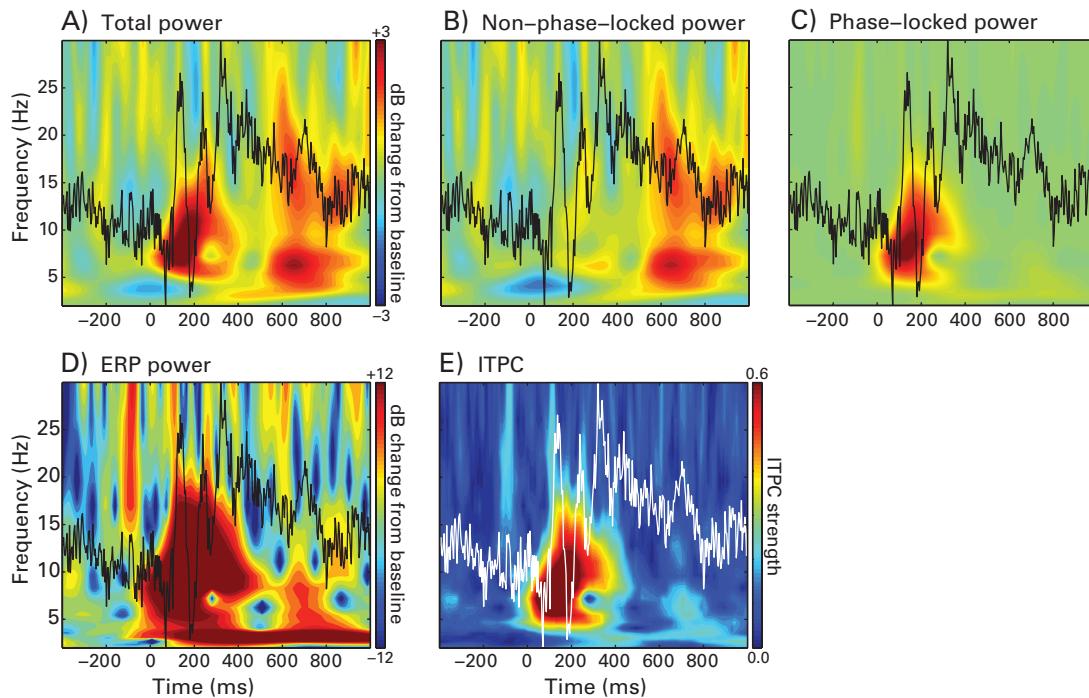


Plate 11 (figure 20.1)

Example results from different methods of computing time-frequency power (panels A-D) and ITPC (panel E) at electrode O1. The ERP is overlaid on the time-frequency plots to facilitate comparison between the ERP and the time-frequency features (the phase-locked ERP is shown in panel B rather than the non-phase-locked ERP because the latter is zero). The ERP is colored white in panel E for increased visibility in the grayscale version of this figure. The color scaling in panels B and C is the same as that in panel A.

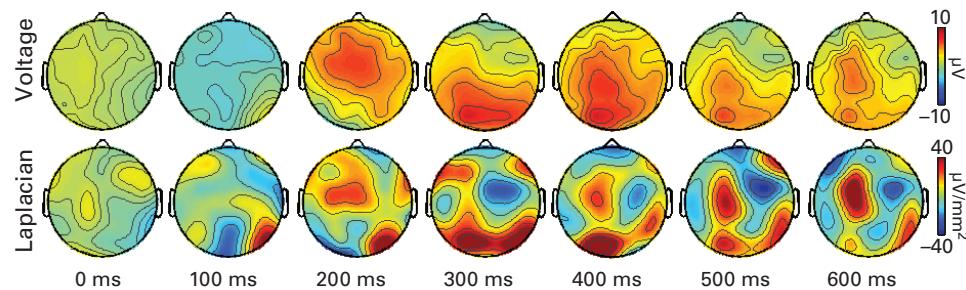


Plate 12 (figure 22.6)

Comparison of spatially unfiltered voltage topographies (top row) and surface Laplacians (bottom row) of the same data. The color scaling is the same for all time points within each row. Note, for example, the topographical maps at 200 ms, for which the clearly visible right lateralized occipital component seen in the surface Laplacian is difficult to observe in the unfiltered data.

A) Before Laplacian B) After Laplacian

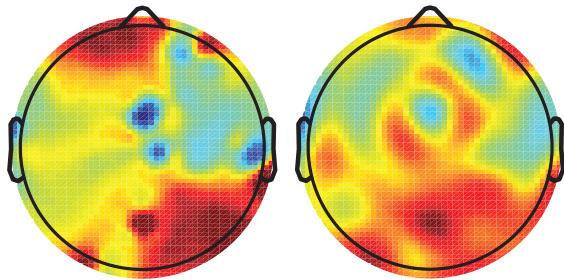
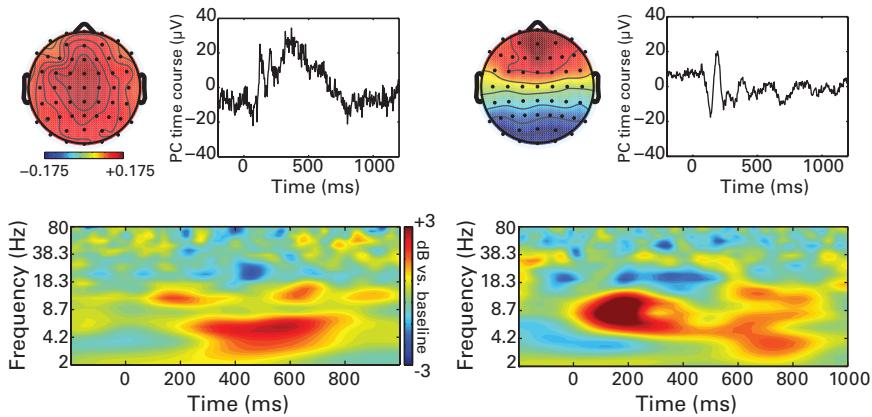


Plate 13 (figure 22.9)

The surface Laplacian attenuates high-spatial-frequency features in addition to low-spatial-frequency features. One time point of activity from a 256-electrode net is shown before (panel A) and after (panel B) application of the surface Laplacian. Because the single-electrode fluctuations (which are probably due to noise) have a very high spatial frequency representation, they are attenuated due to the bandpass filtering of the Laplacian.

A) Results from component 1



B) Results from component 2

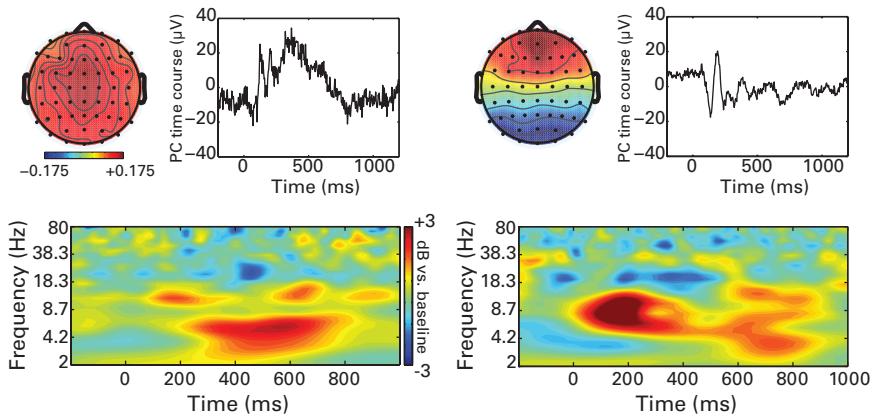


Plate 14 (figure 23.4)

Activities from the first two PCs. The plots show the topographical map, trial-averaged component time course (this is the ERP of that component), and the time-frequency power (both panels use the same color scaling for the topographical maps and for the time-frequency power plots). These two components reflect large-scale spatial processing modes and seem to have dissociated early sensory processing from later cognitive processing, although they were both computed from the same data. Note that these two components are, by definition, orthogonal to each other; whether the neurocognitive dynamics they capture are orthogonal to each other is a separate issue.

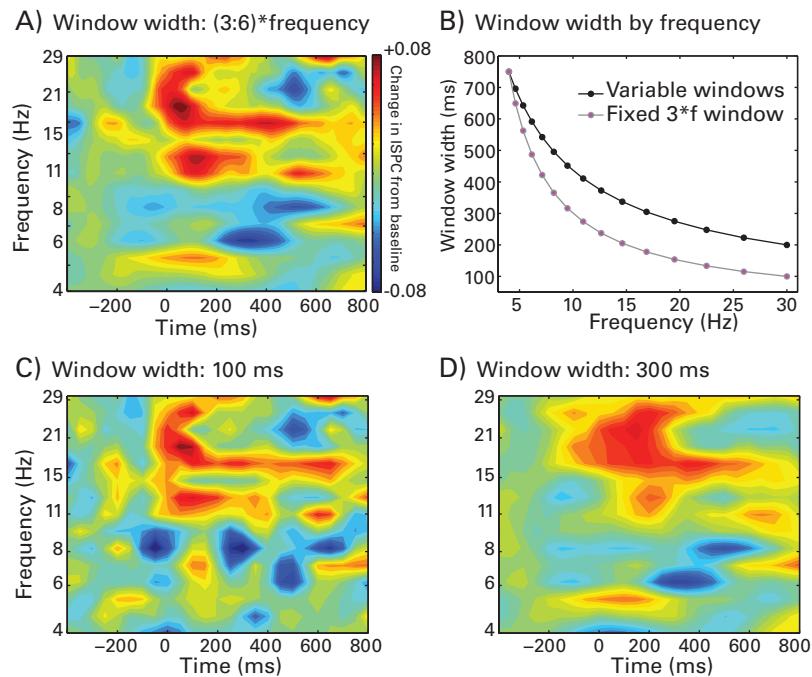


Plate 15 (figure 26.3)

ISPC-time between electrodes Fz and O1 using different time segment lengths. The results shown in panel A were obtained using a variable time segment length of three cycles at the lowest frequency (4 Hz) to six cycles at the highest frequency (30 Hz). The results in panels C and D were obtained using fixed time segment lengths for all frequencies. ISPC-time was computed over sliding time segments for each trial and then averaged across trials. All time-frequency plots have the same color scale (shown in panel A). ISPC strength from a pretrial baseline period of -400 to -200 ms was subtracted to highlight task-related effects. Panel B shows the length of the time segments used at each frequency when creating panel A (black line) and what the length of the time segments would be if a fixed ratio of three times the frequency ($3*f$) had been used (gray line).

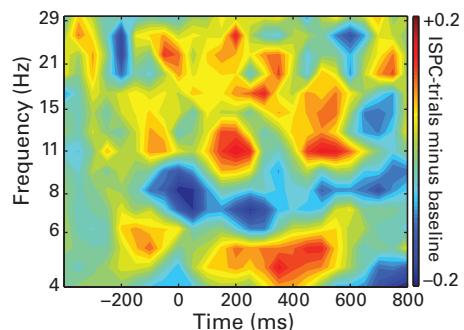


Plate 16 (figure 26.5)

Baseline-subtracted ISPC-trials using the same data as shown in figure 26.3 (plate 15).

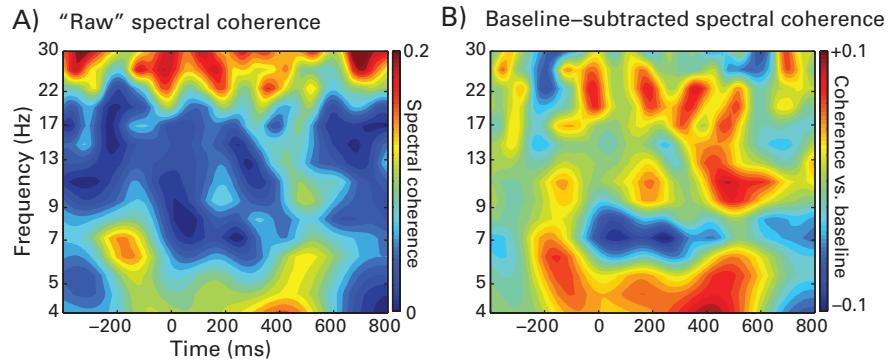


Plate 17 (figure 26.8)

Time-frequency spectral coherence over trials between Fz and O1, shown in “raw” units (no baseline subtraction; panel A) and after linear frequency band-specific baseline subtraction (panel B). These results can be compared with those presented in figure 26.5 (plate 16).

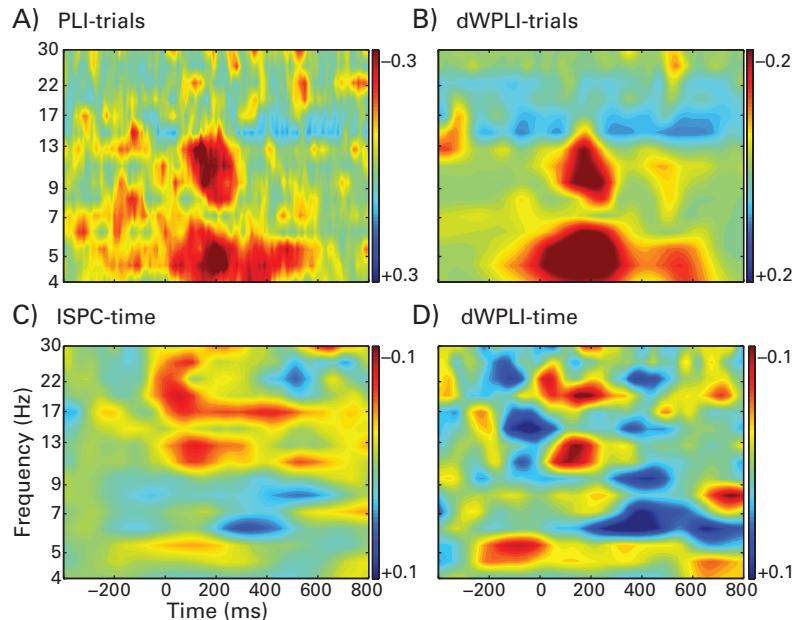


Plate 18 (figure 26.10)

Visual comparison of different measures of baseline-subtracted phase-based connectivity computed between Fz and O1. The color scaling differs across plots to facilitate qualitative comparisons.

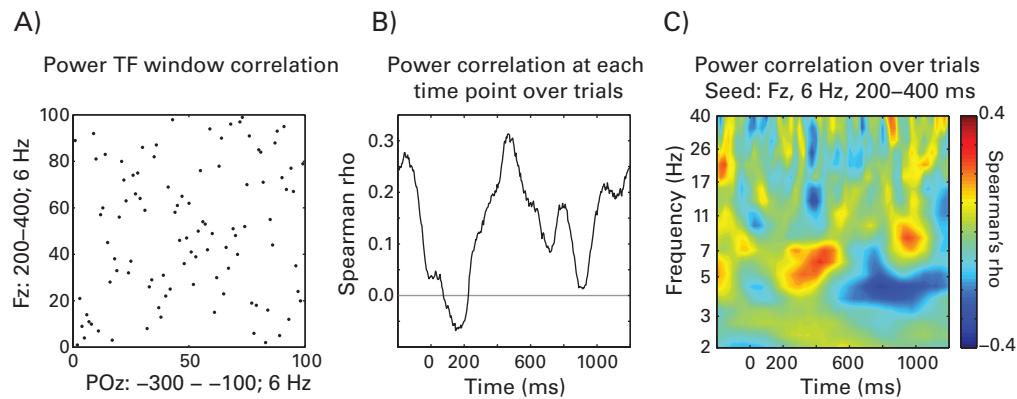


Plate 19 (figure 27.6)

Results from three methods of computing power correlations over trials. Panel A shows the correlation using predetermined time-frequency windows (see x- and y-axis labels; each dot corresponds to one trial). In this case there is no relationship. Panel B shows the correlation computed at each time point separately (this is the power correlation analogue of ISPC-trials). Panel C shows exploratory power correlations over time and over frequency. The seed time-frequency region was 200–400 ms, 6 Hz power at Fz, and the target was time-frequency power from Oz.

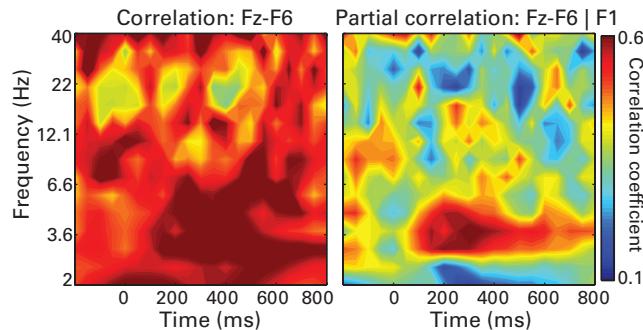


Plate 20 (figure 27.7)

Time-frequency map of partial correlation coefficients between power at Fz and F6 (panel A), while holding F1 constant (panel B). Partial correlations can be used to test network-level hypotheses involving more than two electrodes, or they can be used to minimize the effects of volume conduction on power-based connectivity (as illustrated here) by removing the variance between two electrodes (in this case, Fz–F6) that is shared with—and likely volume-conducted from—a neighboring electrode (F1).

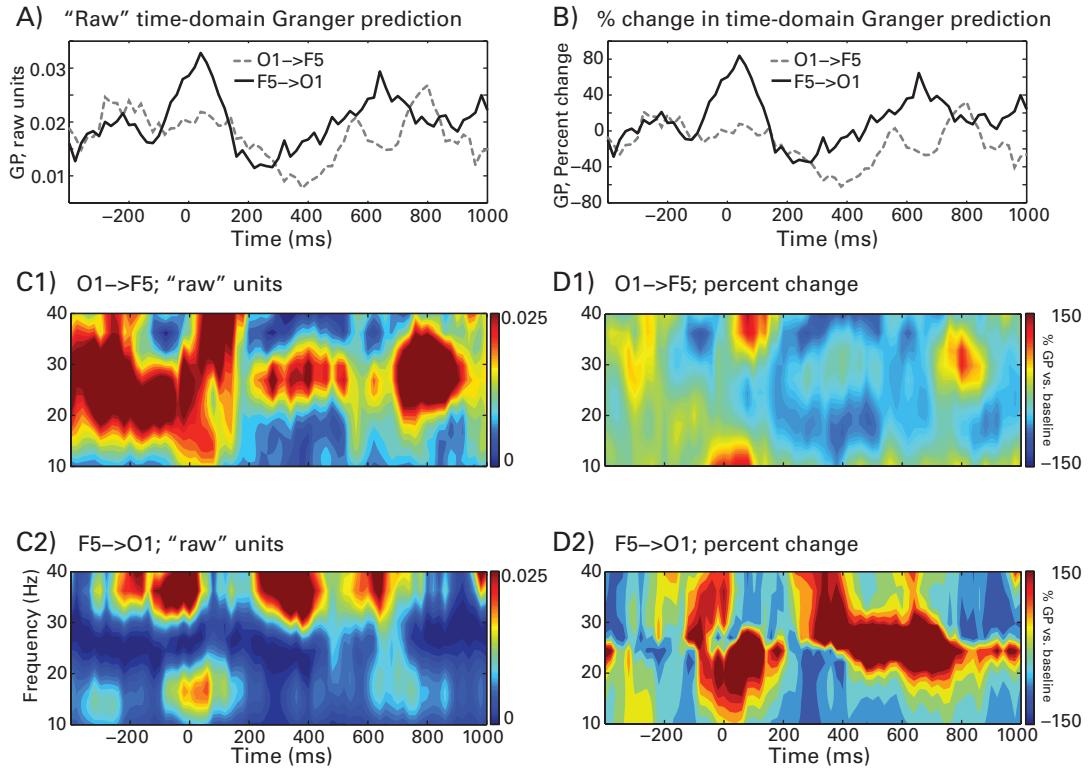


Plate 21 (figure 28.8)

Comparison of Granger prediction in “raw” units (log of error variance ratio, equation 28.4; panels A and C, replotted here from figures 28.3 and 28.5) and percentage change from baseline (panels B and D). Note that because of the pretrial increase in O1 → F5 directed connectivity in panel C, the task-related changes in connectivity appear as a relative decrease. This highlights the importance of a careful inspection of the results before and after baseline transformation so that the results are interpreted appropriately.

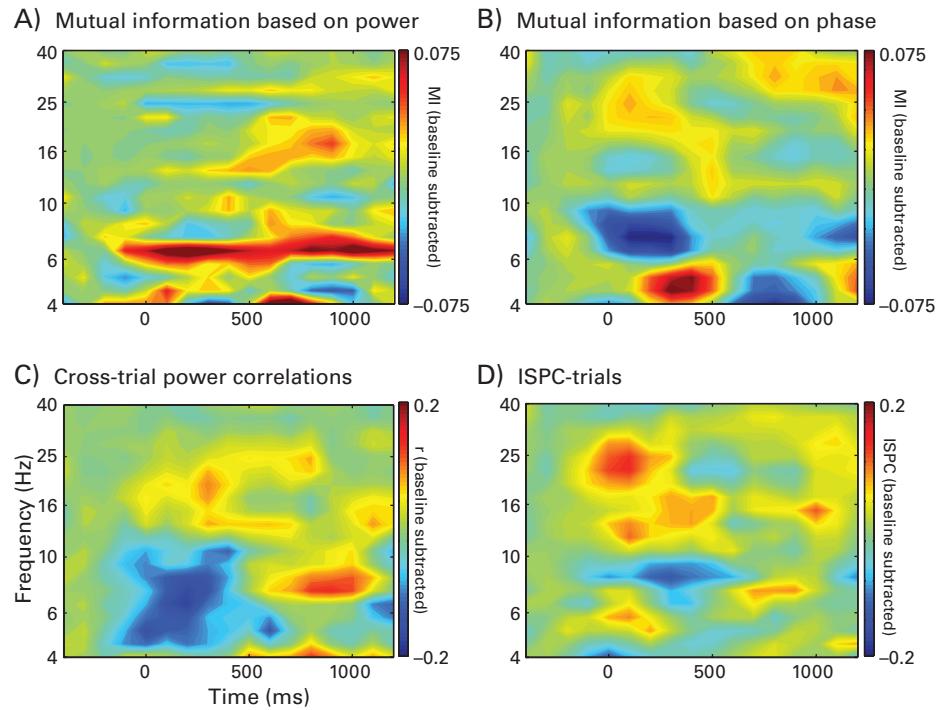


Plate 22 (figure 29.11)

Mutual information over trials between Fz and O1 based on frequency-band-specific power (panel A) and phase (panel B). For comparison, cross-trial power correlations are shown in panel C (r indicates the Spearman correlation coefficient), and ISPC-trials is shown in panel D. In all panels data were baseline-subtracted to focus on task-related effects.

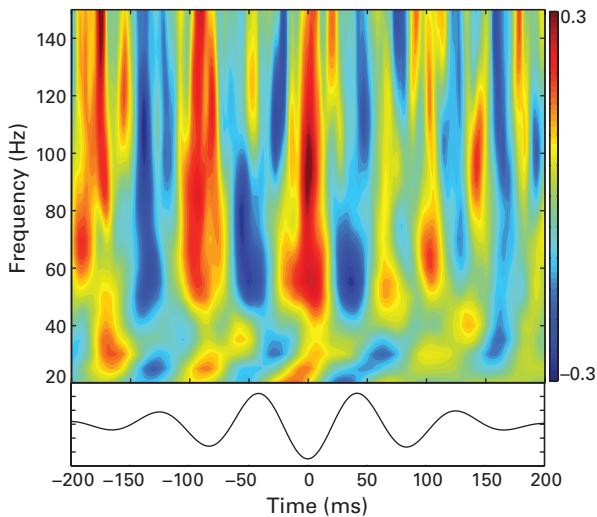


Plate 23 (figure 30.9)

Time-frequency power plot showing Z -normalized high-frequency power time-locked to alpha-band-filtered troughs. These data are from the nucleus accumbens recording.

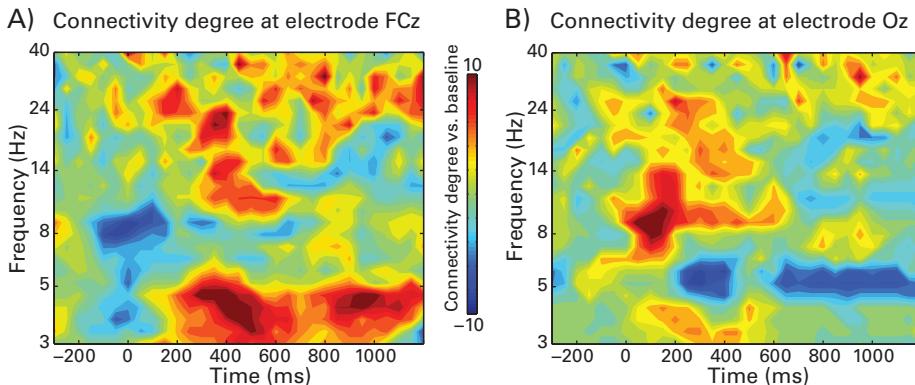


Plate 24 (figure 31.8)

Changes in connectivity degree over time and frequency, separately for frontal electrode FCz and for occipital electrode Oz. The connectivity degree from a pretrial baseline period of -300 to -100 ms was subtracted for each frequency.

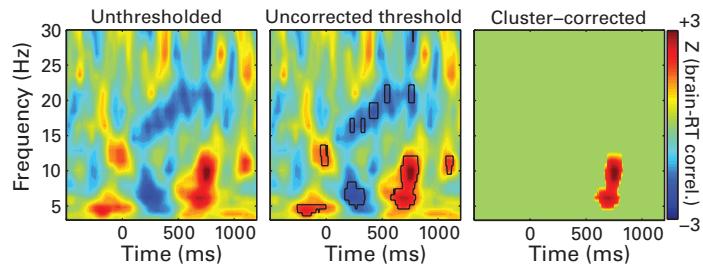


Plate 25 (figure 34.4)

Single-trial correlations (estimated via ordinary least squares) between reaction time and time-frequency power. The left panel shows the Z-values (standardized units from each correlation away from the distribution of null-hypothesis correlation coefficients); the middle panel shows the Z-values with black contours indicating regions that are significant using an uncorrected pixel-level threshold of $p < 0.01$; and the right panel shows the same result with cluster thresholding at $p < 0.05$. No pixels survived thresholding at a corrected pixel-based thresholding of $p < 0.05$. This result indicates that trials with increased power from 5–12 Hz and 500–800 ms tend to have longer reaction times. The middle and left panels show two different ways of illustrating statistically significant results, either by showing all results and outlining the significant regions or by setting nonsignificant pixels to 0 (advantages of different methods of showing results are discussed in chapter 36).

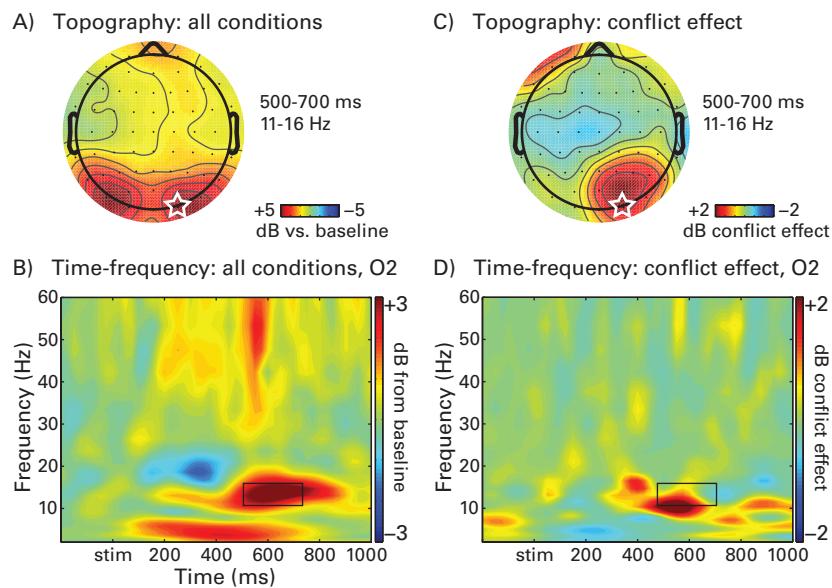


Plate 26 (figure 36.1)

Recommendation for how to show results over time, frequency, and space. Panels A and B show results from all conditions averaged together, relative to the baseline activity. These plots are useful because they show the basic task-related responses. The white star over the right posterior electrode O2 illustrates that the time-frequency plot was taken from this electrode, and the black box in the time-frequency plot illustrates the window selected for the topographical map. Panels C and D show the condition effect (in this case it was a subtraction of trials that contained response conflict from trials that contained no response conflict).

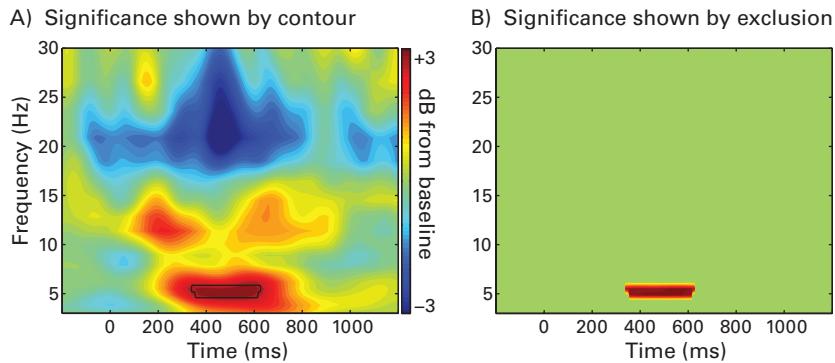


Plate 27 (figure 36.4)

Two methods of showing statistical significance in time-frequency plots. Panel A shows all results and highlights the statistically significant time-frequency regions. This is a useful representation because readers can see a broad range of dynamics and might be able to link the results to other literatures and findings. It is also useful because it shows what other effects might be present in the data that are perhaps relatively subtle, and it shows whether the effects are specific to a certain time-frequency range (see also figure 36.3). In contrast to panel A, panel B contains an alternative method of showing significant results. Here, all nonsignificant pixels are set to zero. Although this makes the significant effect more visually salient, it also obscures a considerable amount of information in the time-frequency dynamics from this electrode. Other examples of highlighting significant regions versus setting nonsignificant pixels to zero were shown in chapter 34.

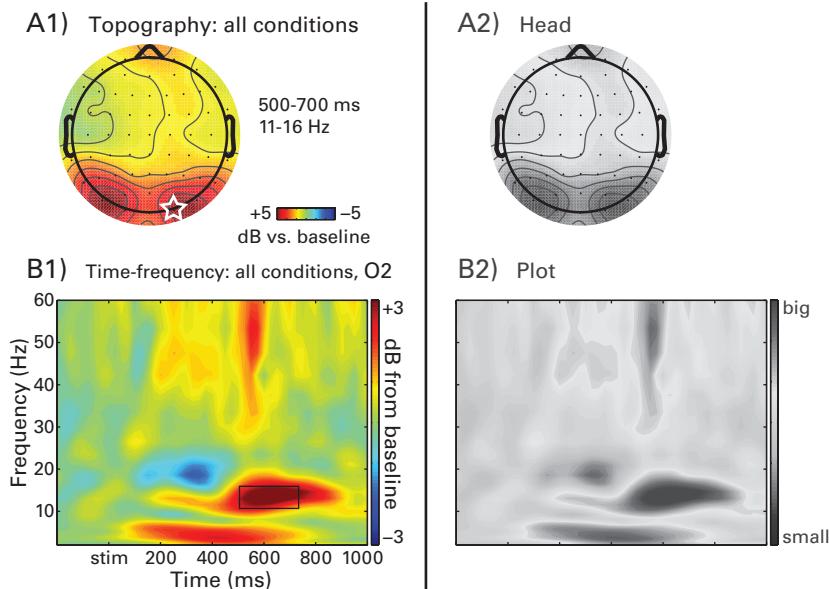


Plate 28 (figure 36.6)

Time-frequency and topographical plots that have informative (panels A1 and B1) and uninformative (panels A2 and B2) captions accompanying the plots. It is difficult or impossible to extract information from the plots in panels A2 and B2.