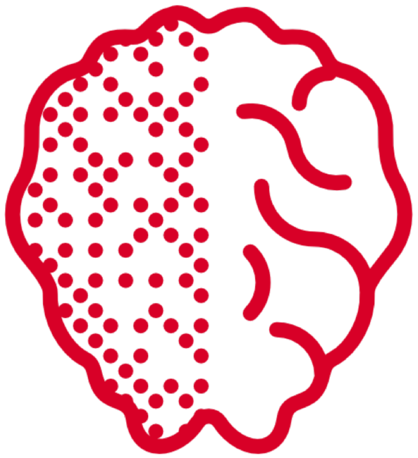
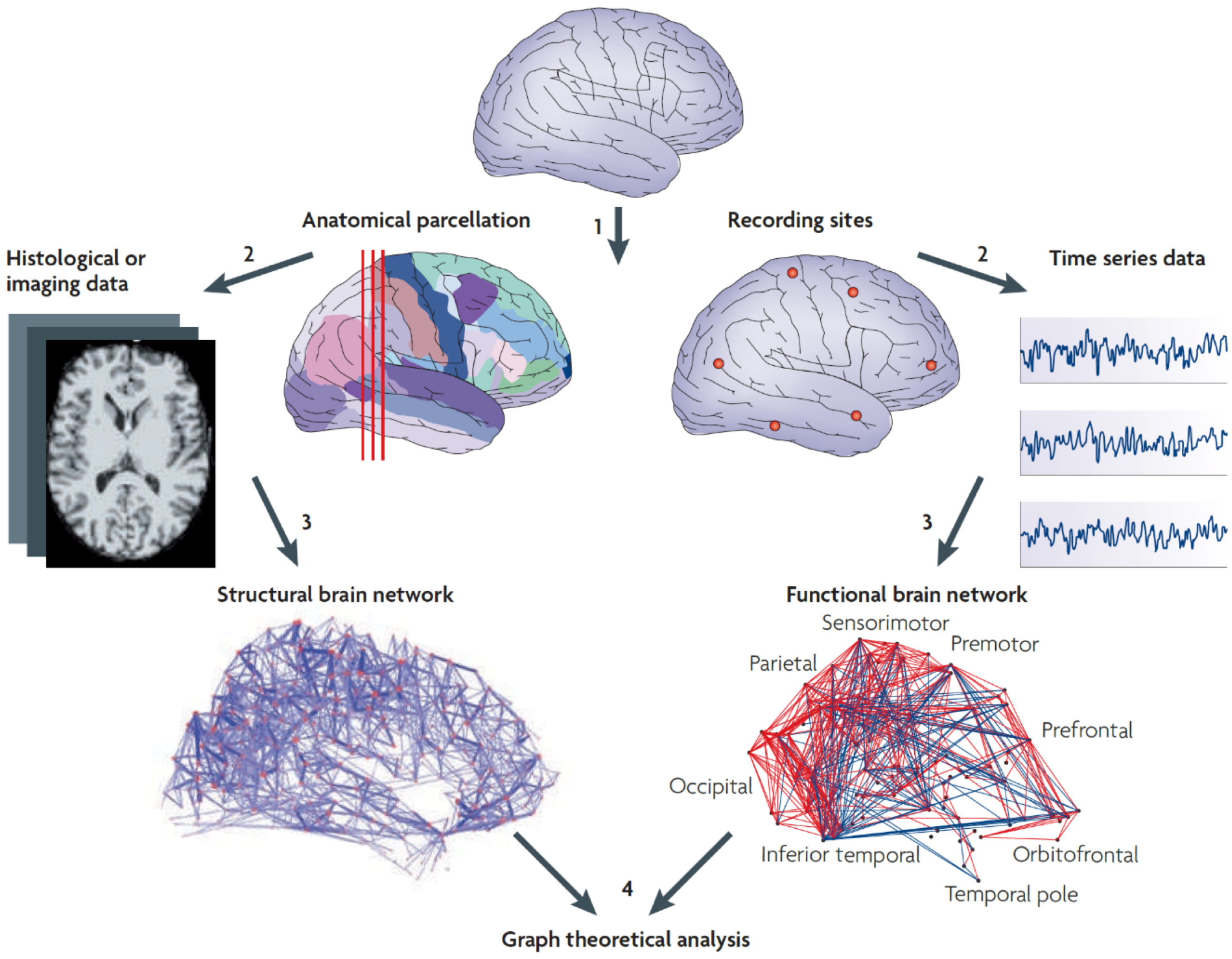
Fundamentals of Network Neuroscience



Connectivity & Electrophysiology

Applications to Human Memory

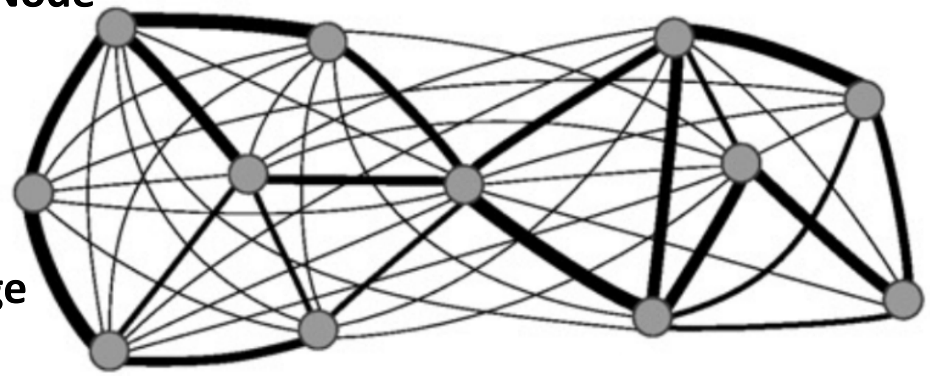
Network Neuroscience Fundamentals

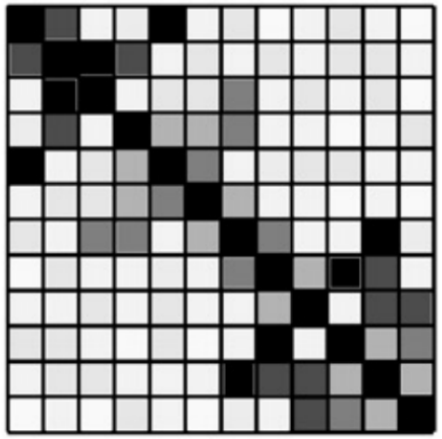


Bullmore and Sporns, 2009

weighted undirected networks

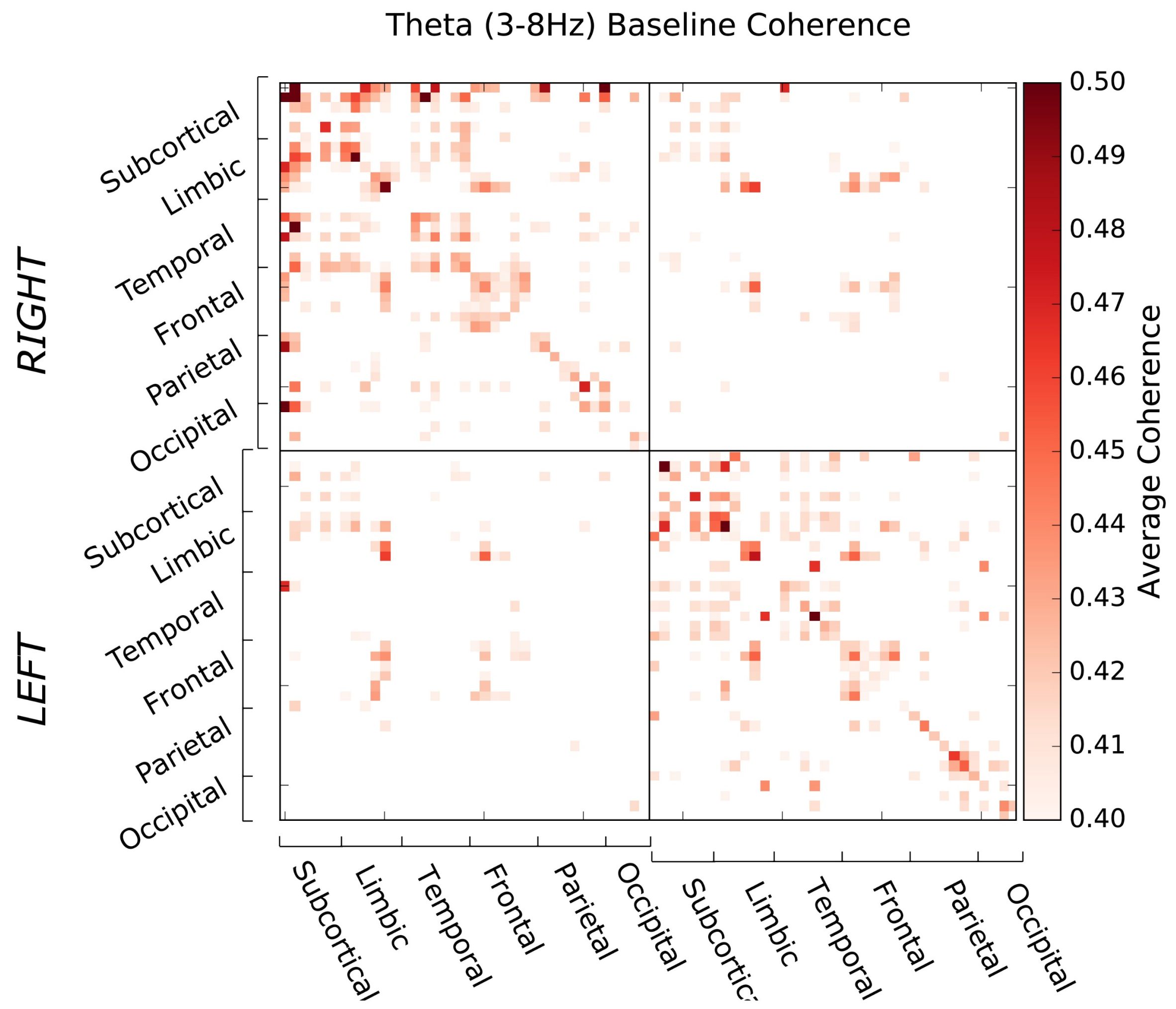
structural datasets: diffusion MRI, structural MRI
functional datasets: functional MRI, MEG, EEG



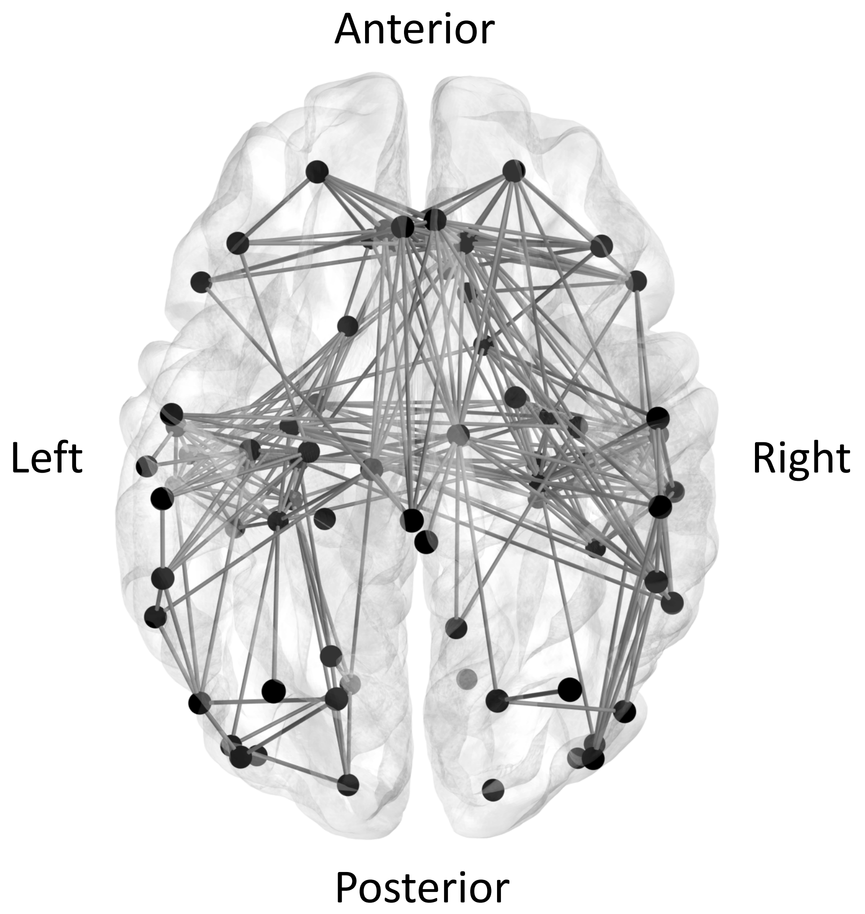


Rubinov and Sporns, 2010

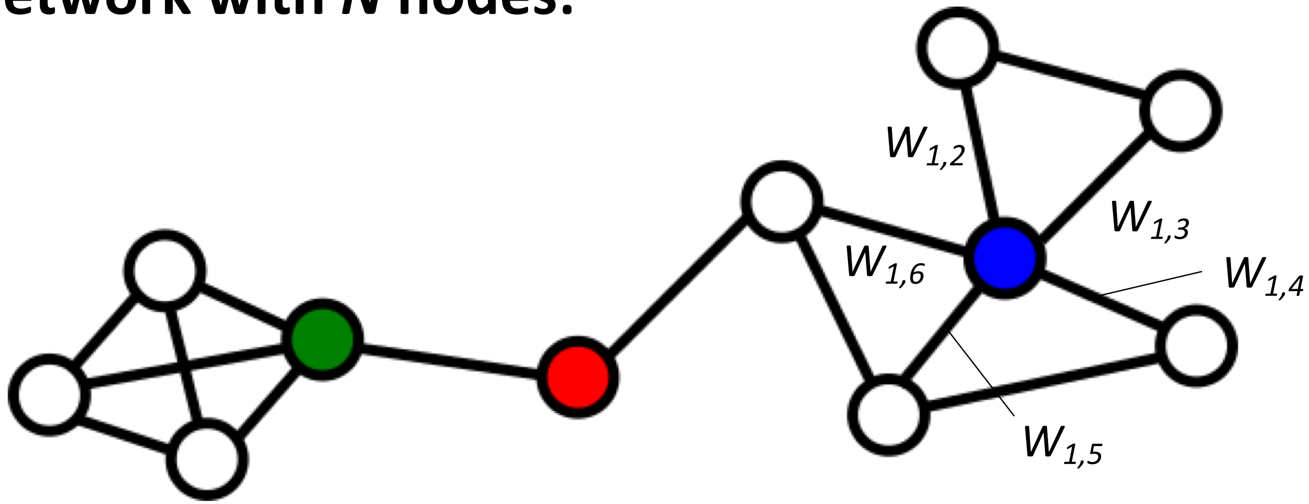
Examples from our data:



(Thresholded for visualization)



Network with *N* nodes:

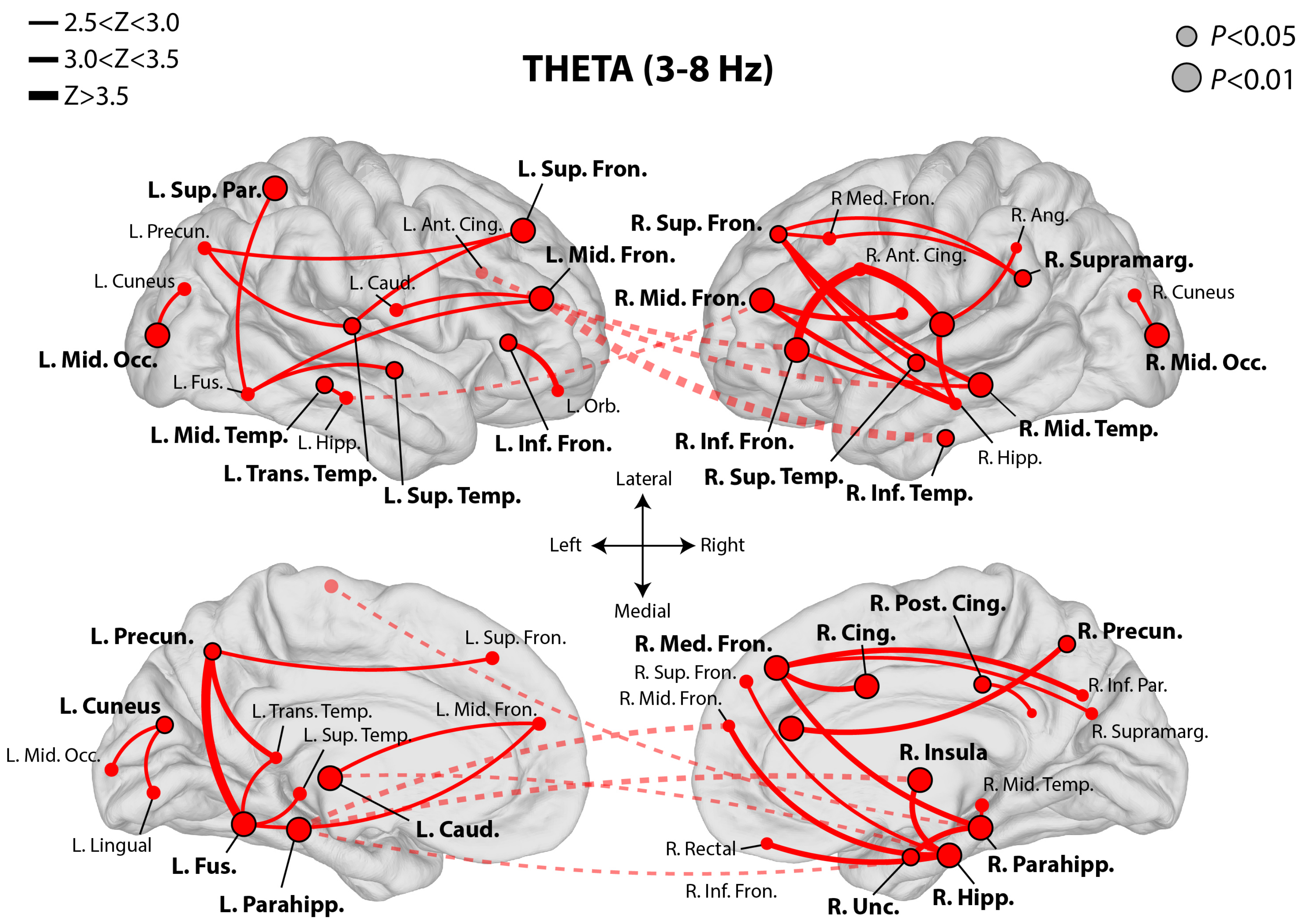


Node strength *k*:

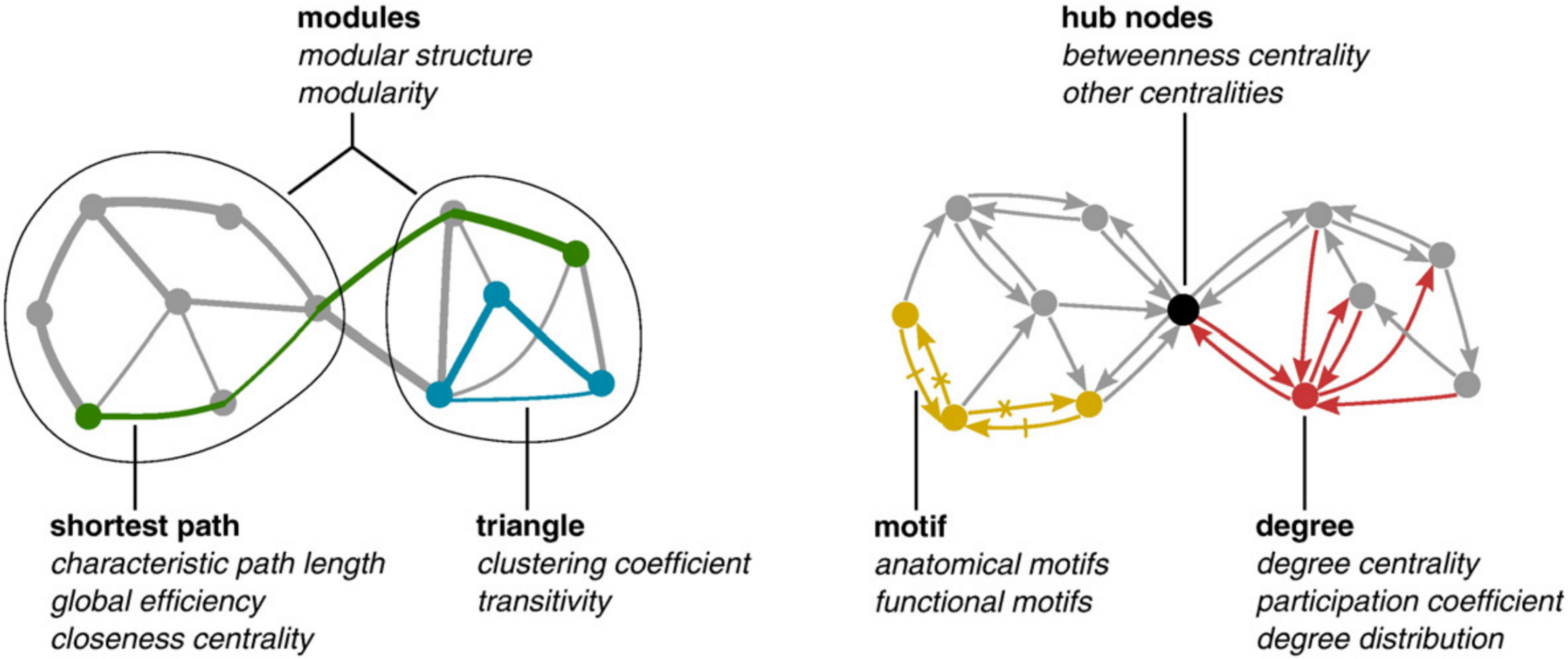
$$k_i^w = \sum_{j \in N} w_{ij}$$

(A measure of "hubness")

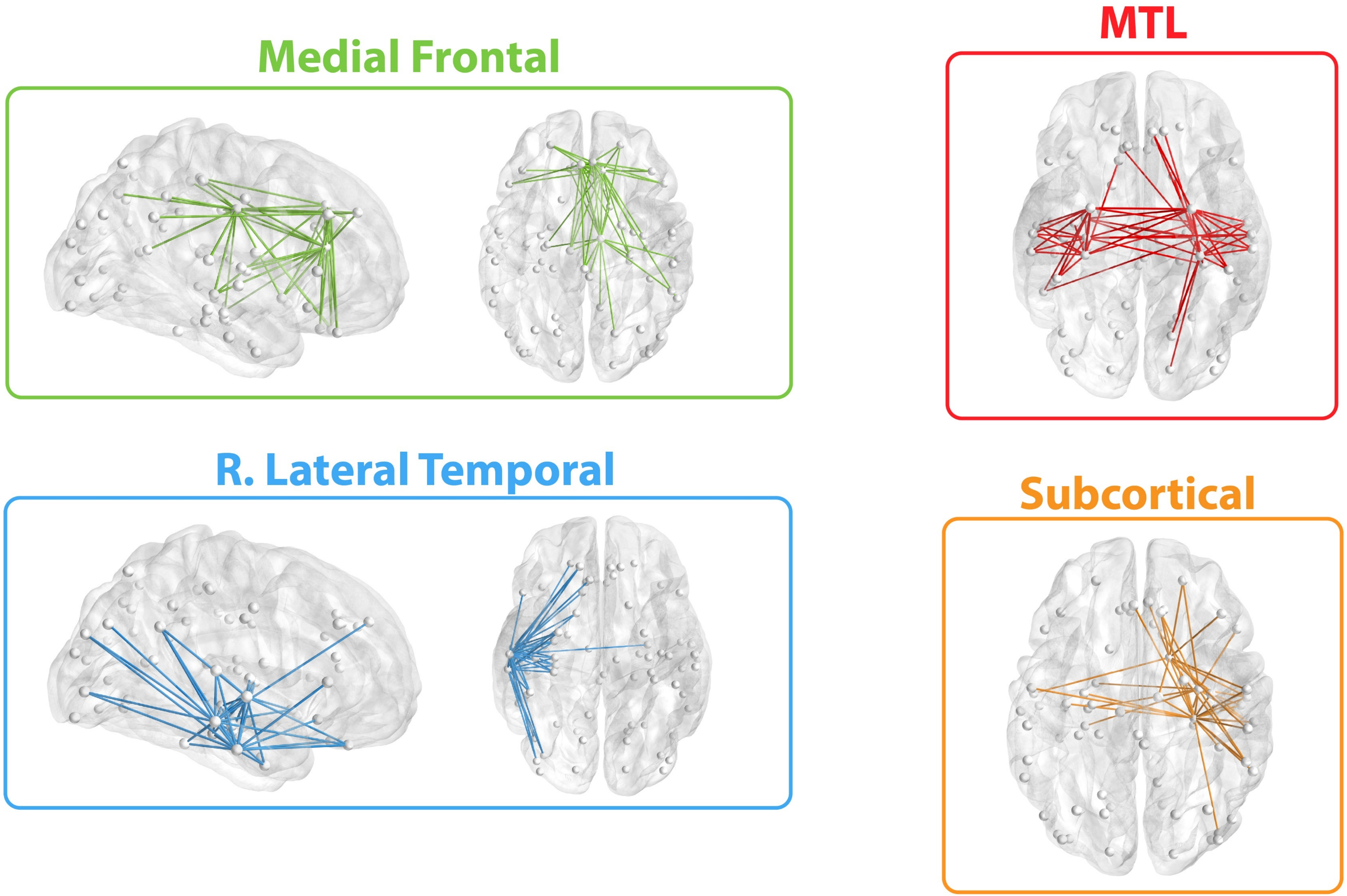
How to assess significance?

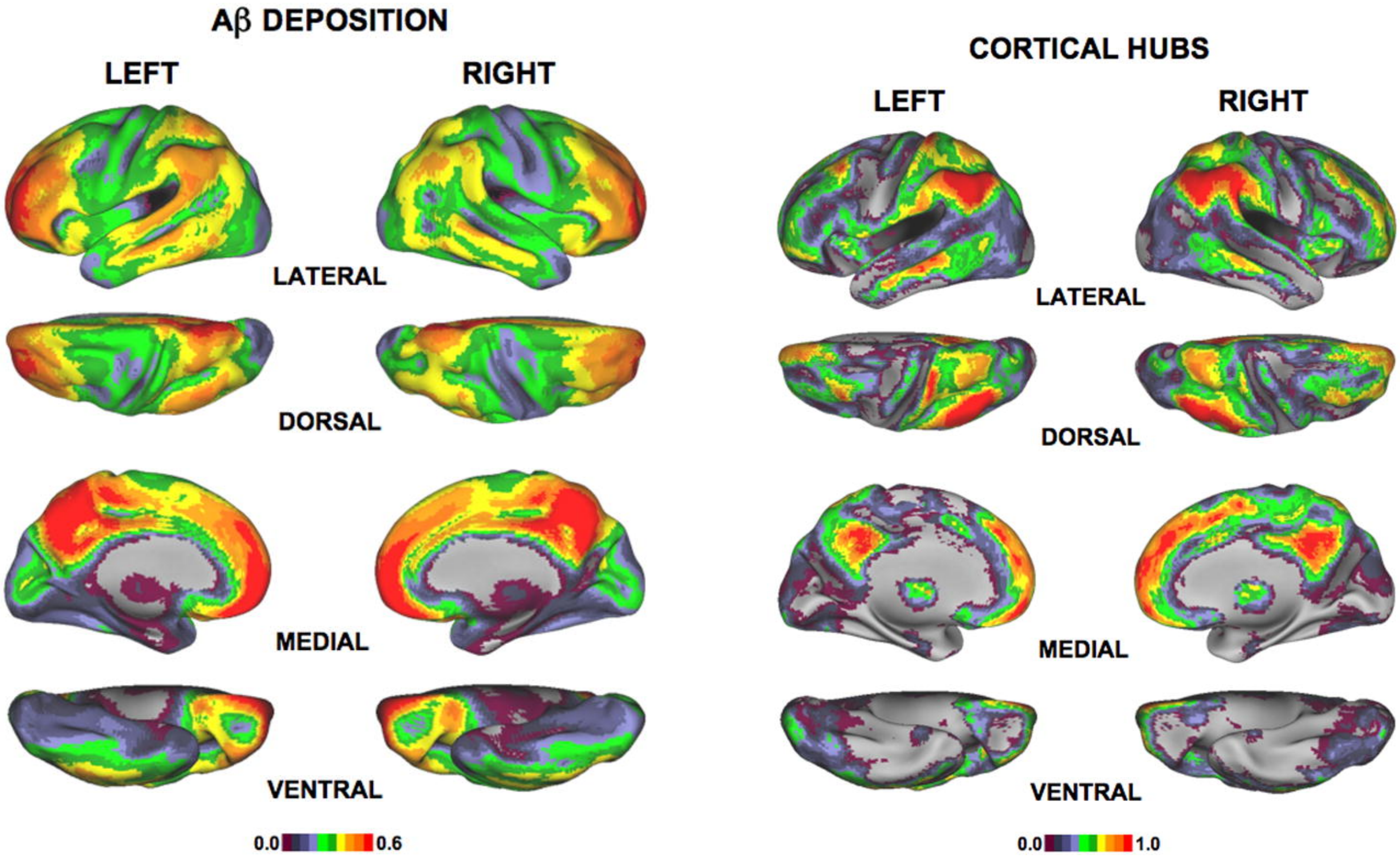


Solomon, et al. 2017



Rubinov and Sporns, 2010


Medial Frontal
MTL
R. Lateral Temporal
Subcortical



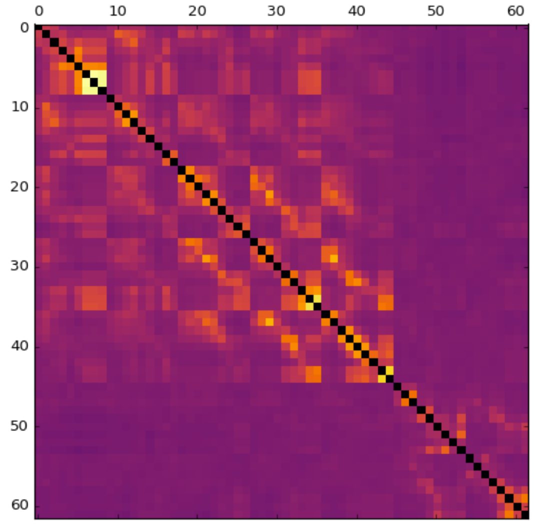
Bucker, et al. (2010)

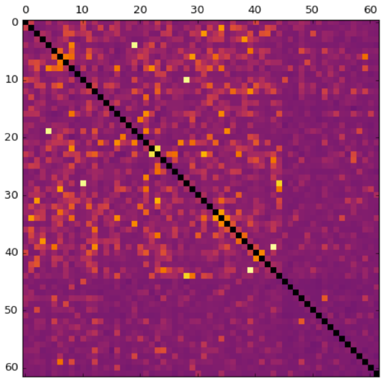
Picking a good null model

1. Randomize your true network
    - Algorithms will "rewire" your network to give a null distribution of network statistics.

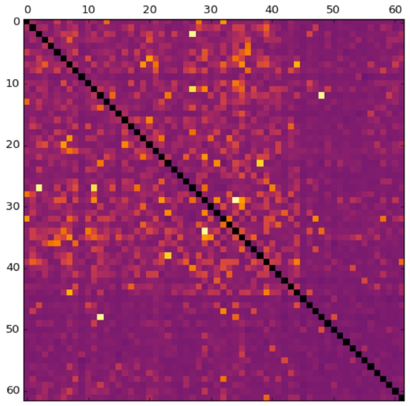
2. Randomize your **original data** used to create network.
    - Avoids possibility that network null is not strong enough.

True Network

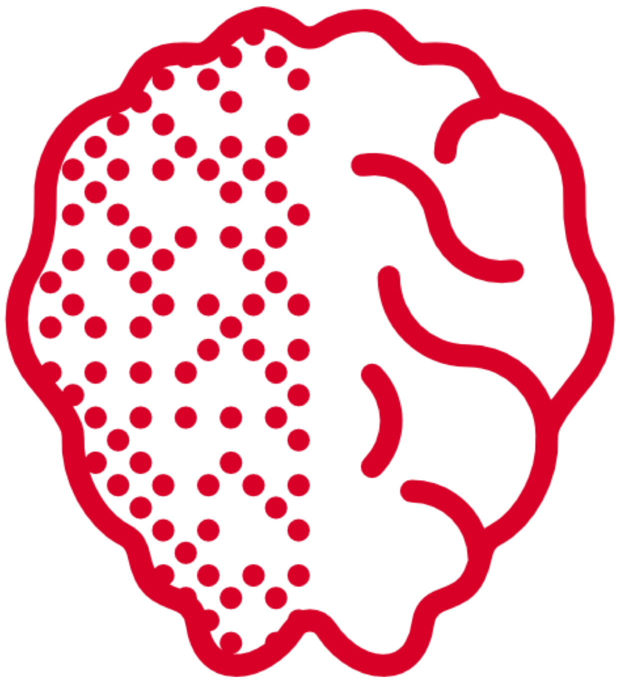




Random graphs w/ preserved degree distribution



Connectivity and Electrophysiology



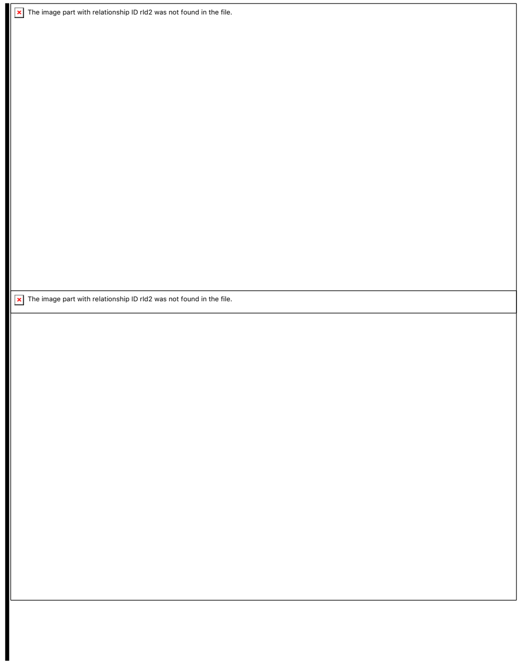
- Functional connectivity between electrodes reflects **correlated timeseries activity.**

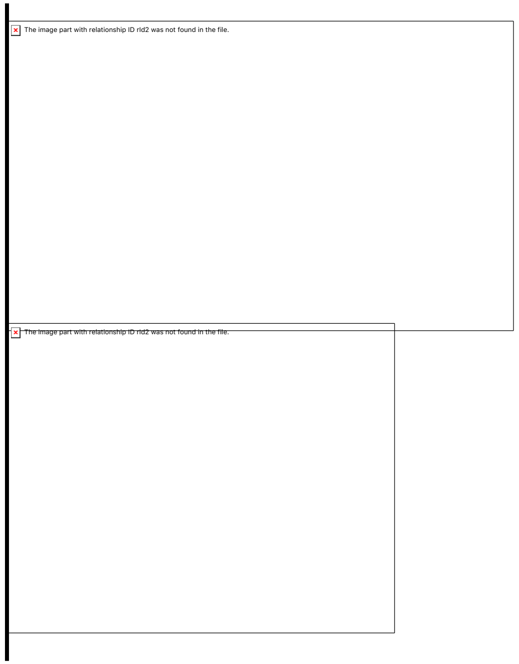
Distinction #1

Consistent in time

Consistent but lagged

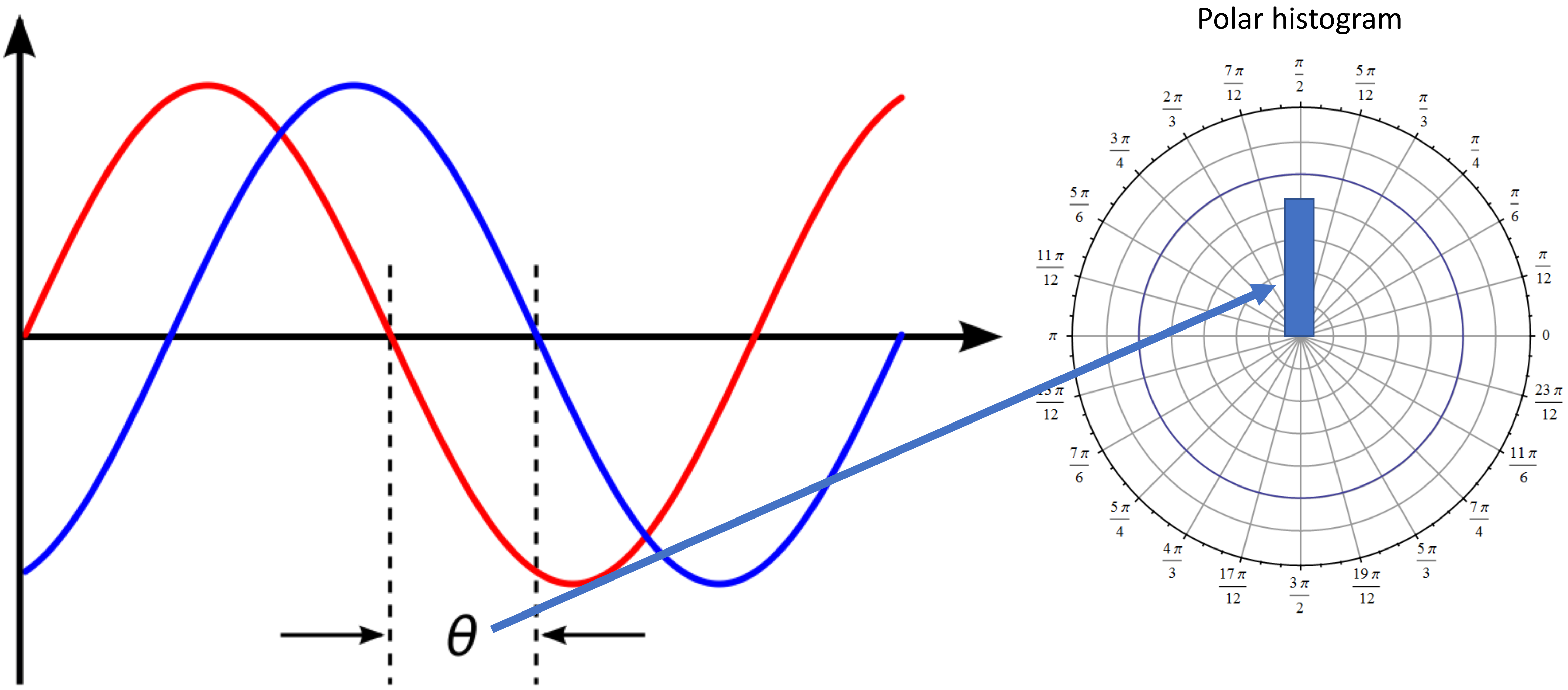
e.g., Pearson correlation



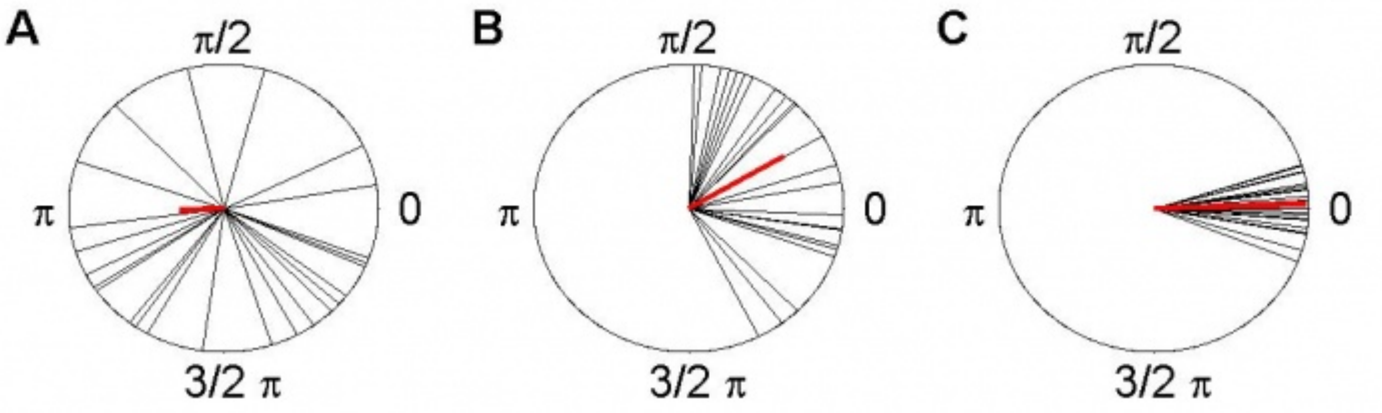


e.g., PLV, Coherence, PLI, Causality, …

Phase differences are key!

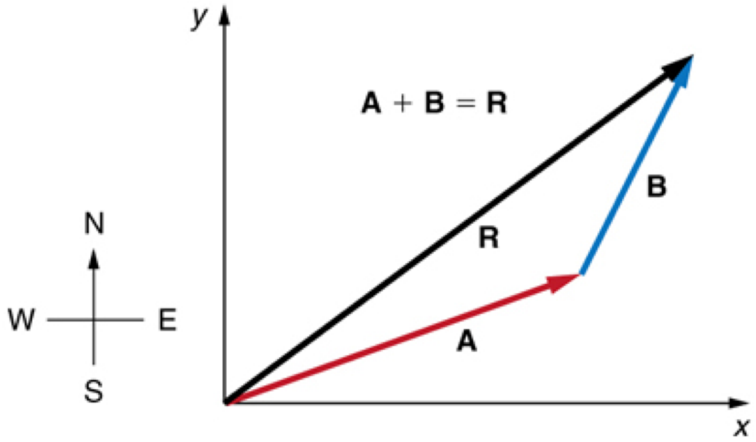


Low phase locking



High phase locking

(Phase Locking Value; PLV)



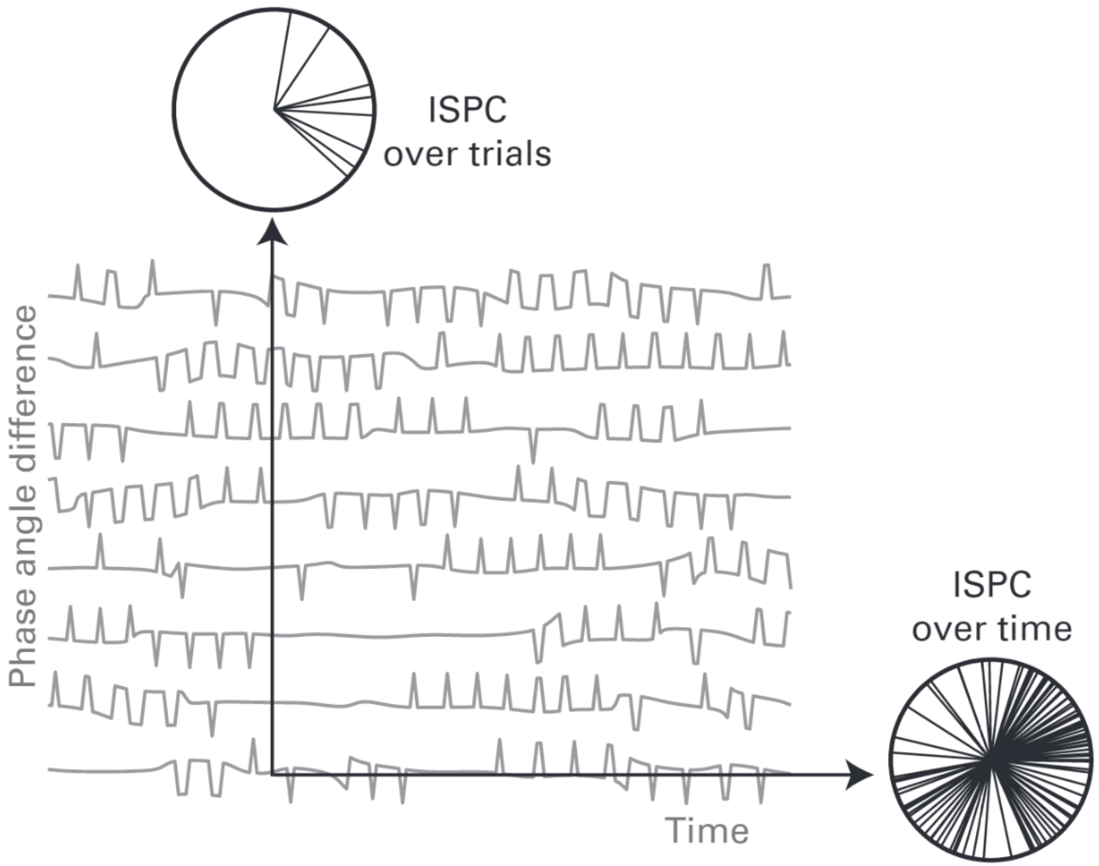
Formally: The mean **resultant vector length** of the phase difference distribution

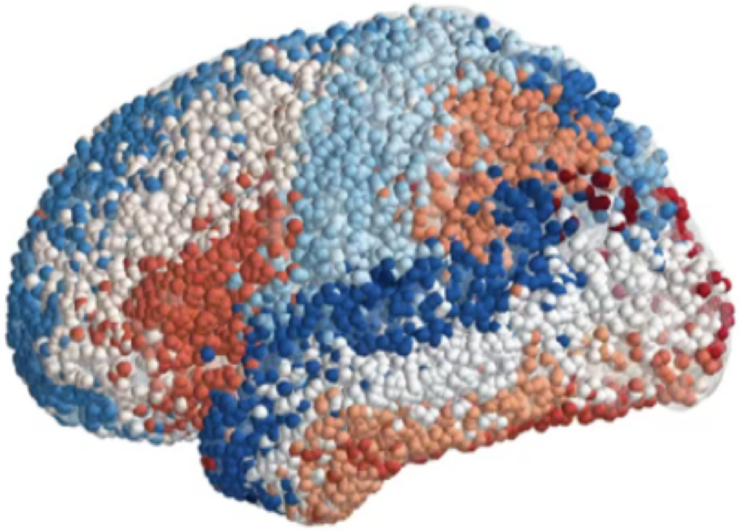
Consistent

Distinction #2

Over Time

Over Trials

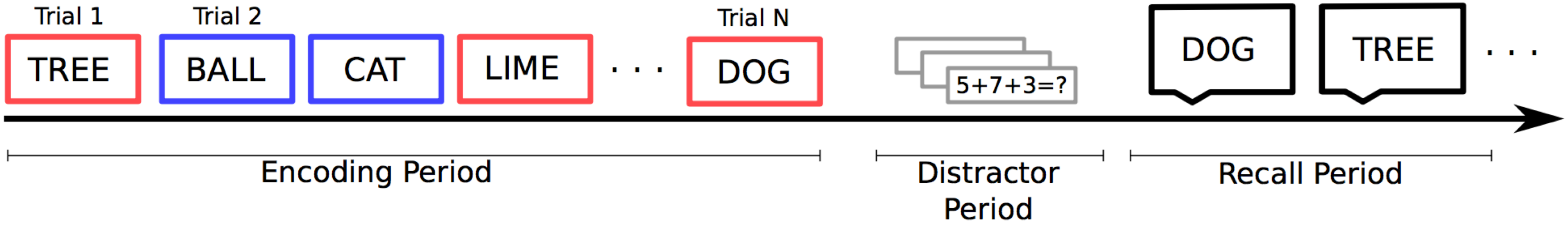




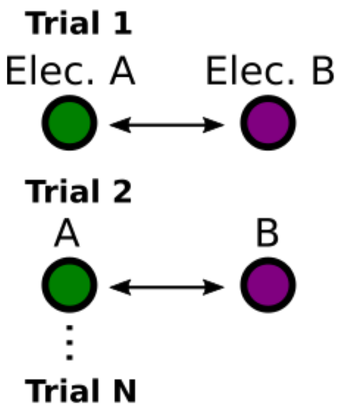
300 neurosurgical patients
w/ indwelling electrodes

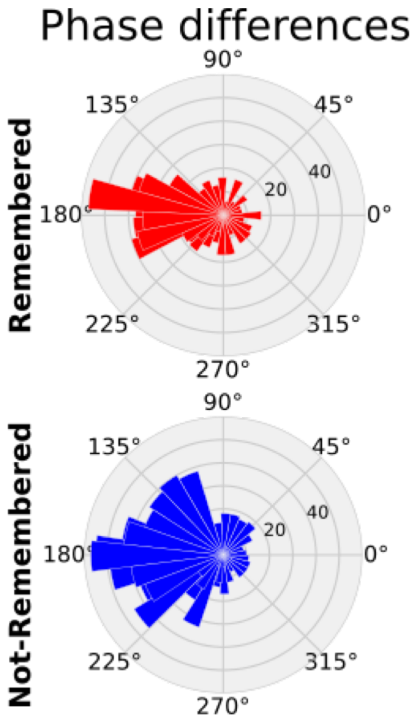
30,000+ electrodes

Verbal free-recall memory task:

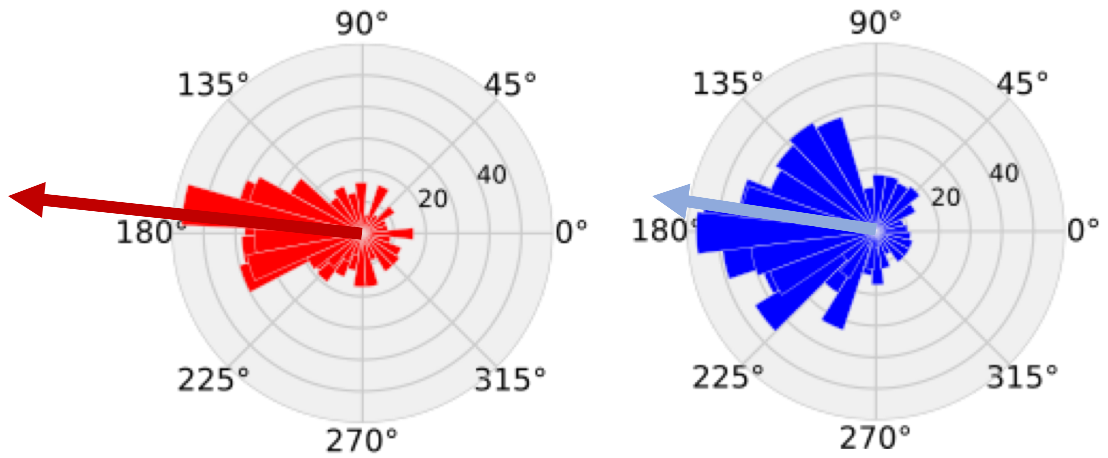


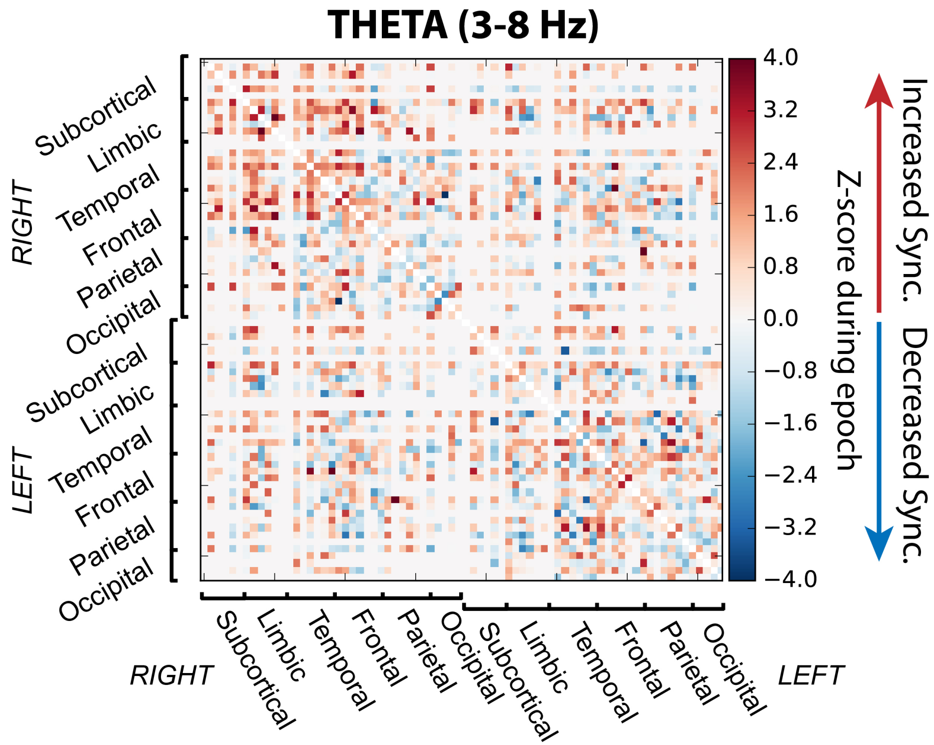
For a given epoch/frequency:
Compute **phase differences** between
All pairs of electrodes at each trial





For a given A-B pair:
Sort by remembered,
not-remembered





Length of vector reflects degree of **phase-locking**, called **phase-locking value.**

Non-parametric permutation test for significant difference in phase-locking values between conditions.

Solomon, et al. (2017)

“Phase Lag” metrics:

Phase lag index (PLI)
Weighted PLI (wPLI)

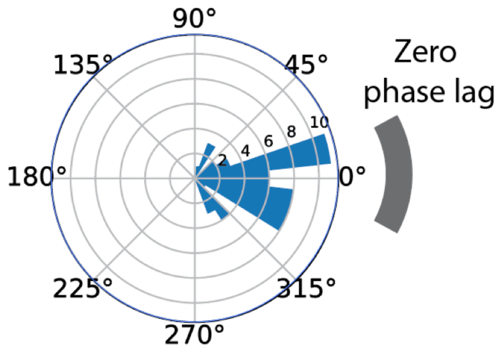
Addresses
Volume conduction

“Mixed” metrics

Coherence

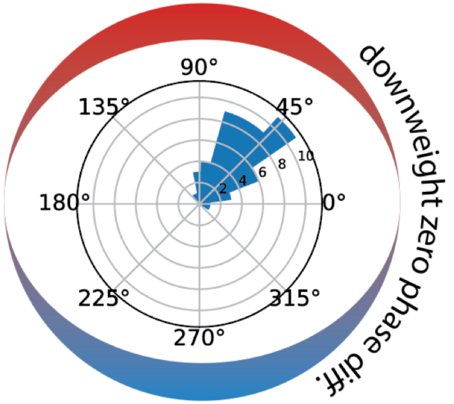
Combines power
and phase

Phase-locking value (PLV)



Possible volume-conduction confound

Weighted Phase Lag Index (wPLI)



Essentially:

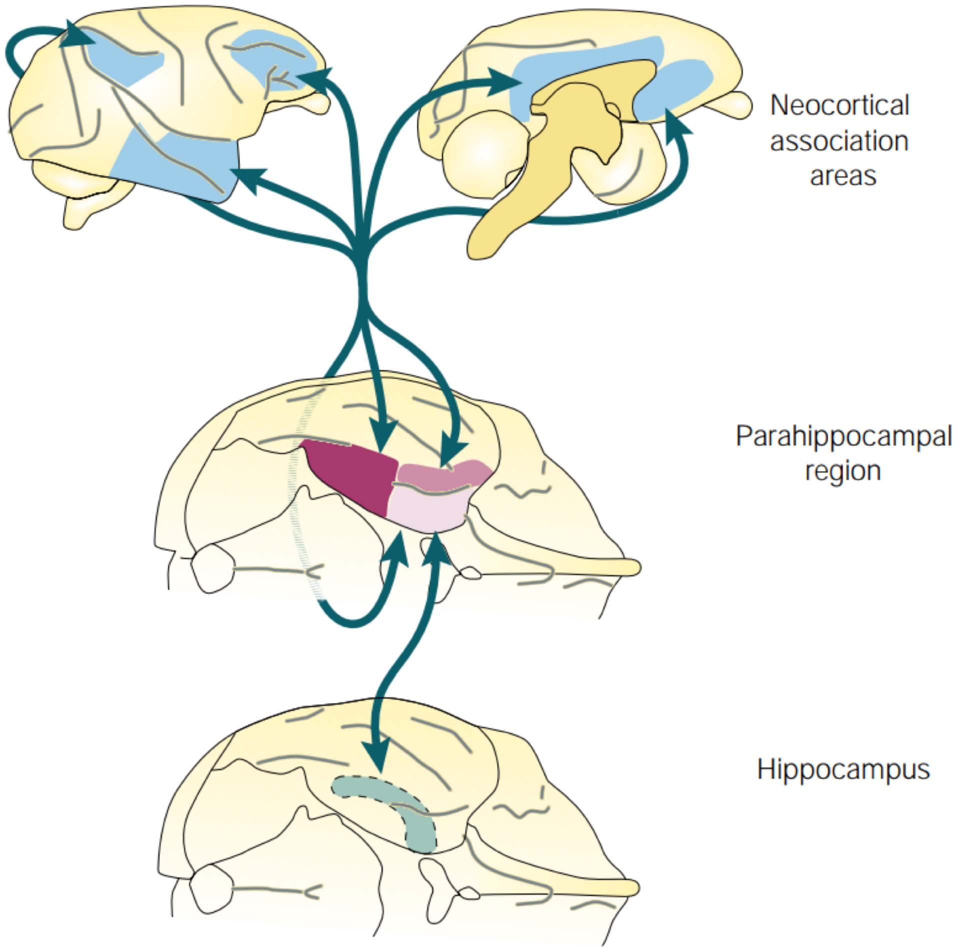
- The PLV, but weighted by power at the frequency of interest
- Frequently used *over time* and not trials
- Not suitable for very short time periods or if need for precise timing.

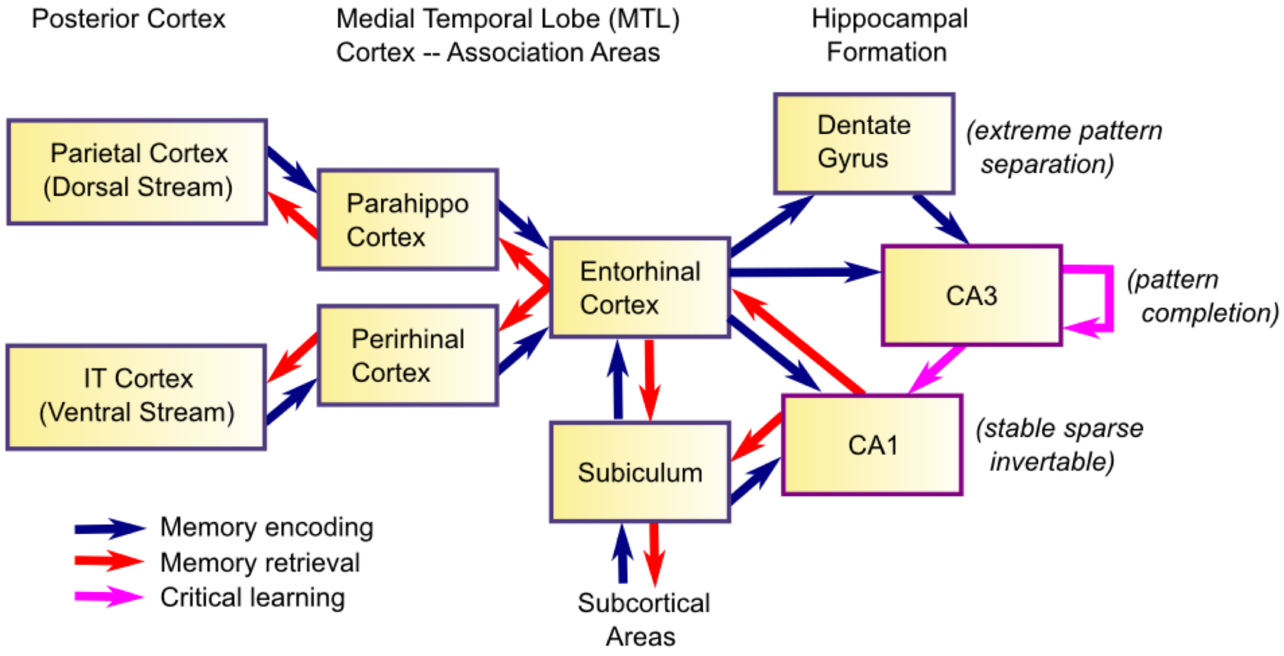
Applications to Human Memory

Discussing Today:

- Overview of connectivity in human memory.

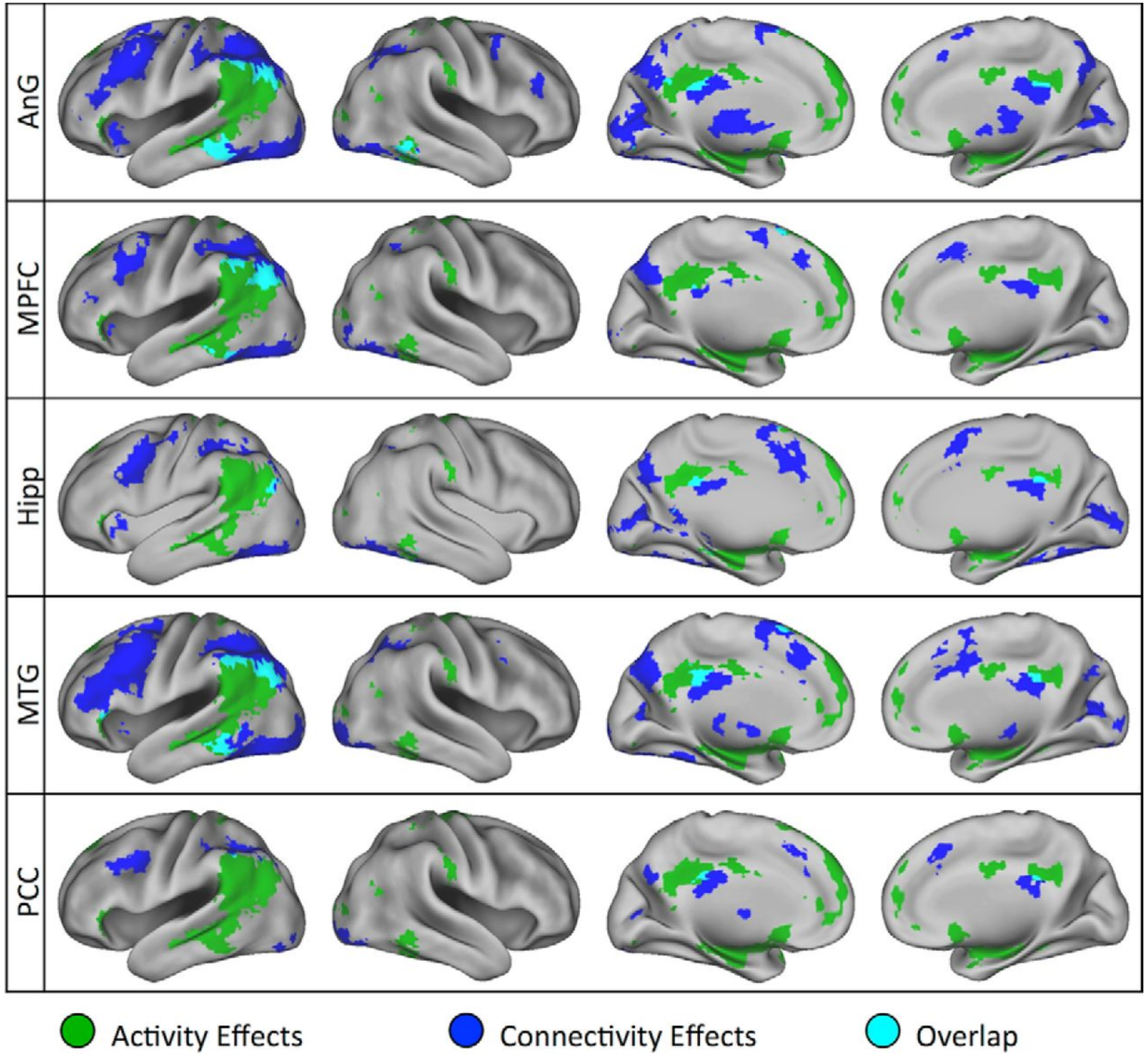
- Different connectivity metrics: Principles and applications
  - PLV
  - Coherence
  - PLI
- Example phase-locking code/demo.





Eichenbaum, 2000

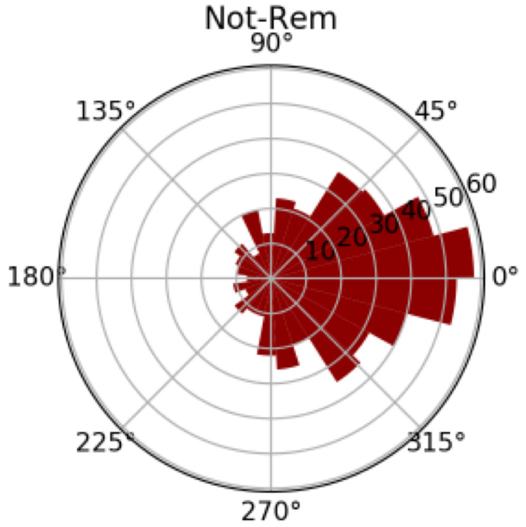
O'Reilly and Munakata, 2012



King, et al. 2015

Phase-locking value (PLV)

- Computationally efficient
- Phase-only
- Inter-trial or over-time
- Biased by # samples
- Sensitive to zero phase lag



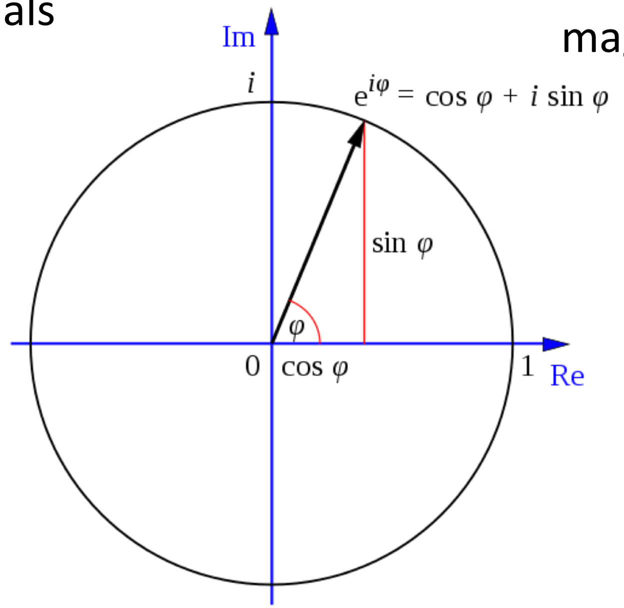
Sum over trials or time

Exponential representation of vector (Euler's)

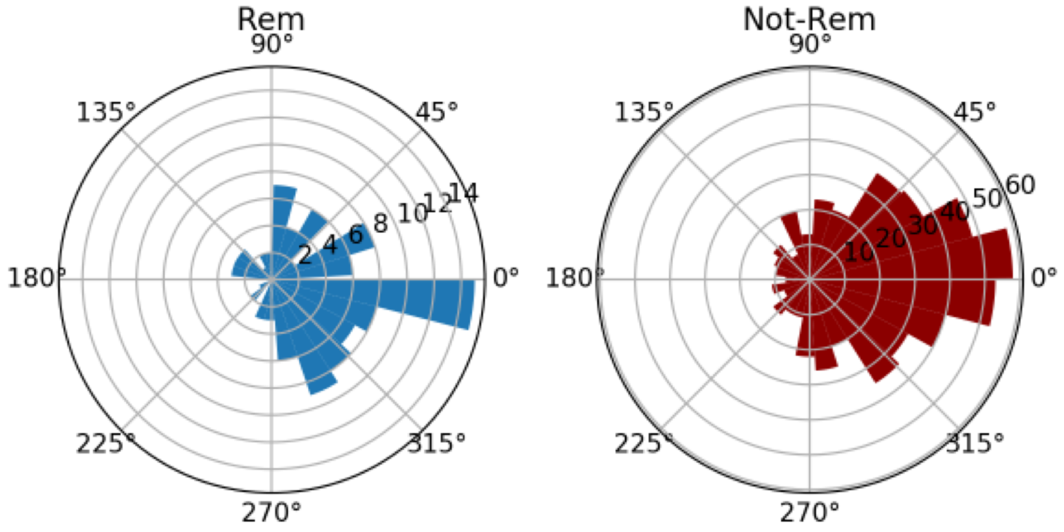
$$ISPC_{tf} = \left| n^{-1} \sum_{r=1}^{n} e^{i(\phi_{xtfr} - \phi_{ytfr})} \right|$$

Taking the mean over time/trials

Just want vector magnitude

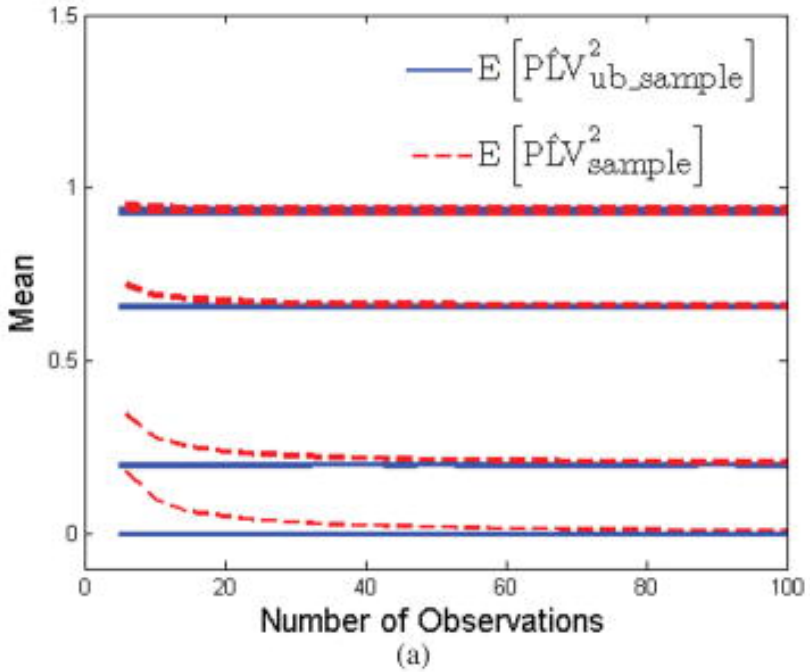


❗ Bias Problem!



Far fewer "Remembered" vs. "Not-Remembered" trials!

Values cannot be directly compared between conditions



Aydore, et al. (2013)

Bias problem solutions:

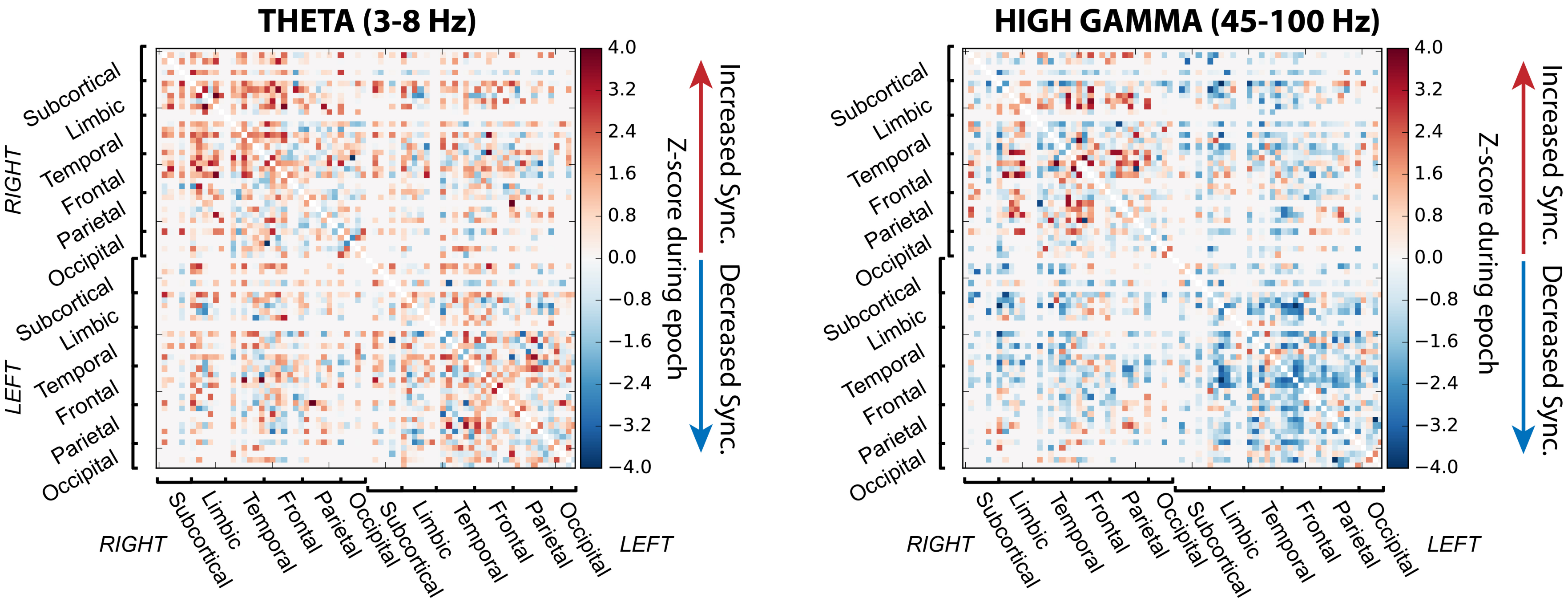
1. Match sample size between conditions (e.g. random draws)

2. Nonparametric permutation (e.g. shuffle trial labels) to create null distribution of test statistic

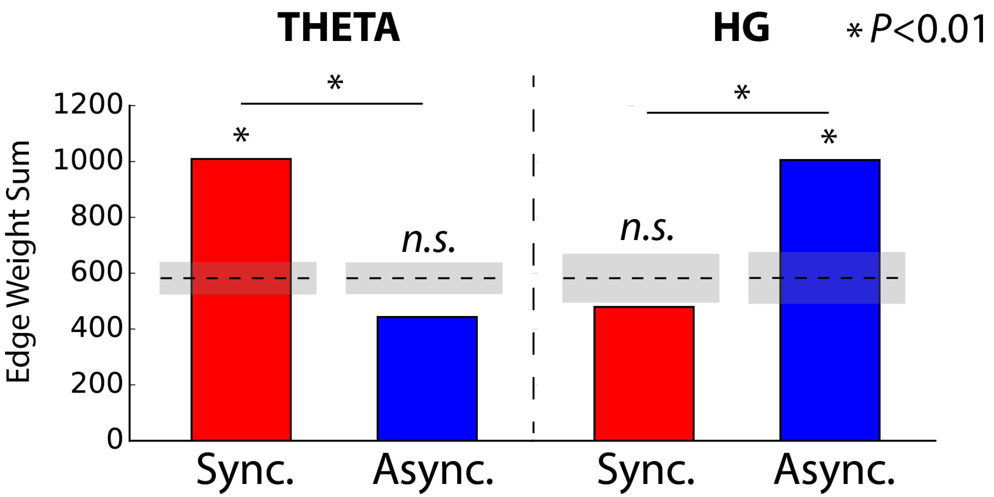
3. Use an unbiased/debiased estimator (e.g. PPC, debiased wPLI, etc.)
    - See MNE package

4. Use matched baseline period for each event's PLV (see assignment)

PLV correlates with memory!



Band-specific effect



Spectral coherence

- Normalized cross-spectral density
  - Phase/power combo
- Commonly over-time (3 cycles minimum)

Cross-spectral density

$$Coher_{xy} = \left| \frac{S_{xy}}{S_{xx} S_{yy}} \right|$$

Auto-spectral densities

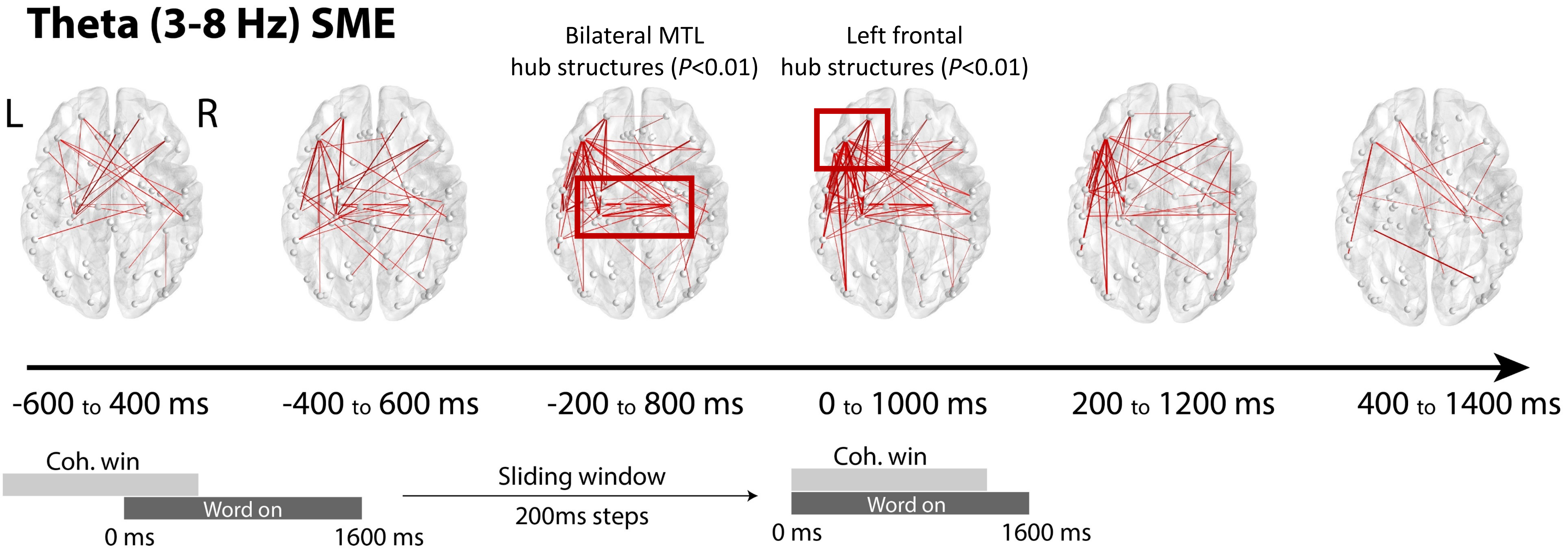
$$Coher_{xy} = \frac{C_{xy}}{\left(n^{-1}\sum_{t=1}^{n}|m_{tx}|^2\right)\left(n^{-1}\sum_{t=1}^{n}|m_{ty}|^2\right)}$$

Normalized cross-spectral density

Power at electrodes x, y

$$C_{xy} = \left| n^{-1}\sum_{t=1}^{n}|m_{tx}||m_{ty}|e^{i\phi_{txy}} \right|^2$$

Vector sum of phase differences (same as PLV!)


Theta (3-8 Hz) SME
Bilateral MTL
hub structures (P<0.01)
Left frontal
hub structures (P<0.01)
L
R
-600 to 400 ms
-400 to 600 ms
-200 to 800 ms
0 to 1000 ms
200 to 1200 ms
400 to 1400 ms
Coh. win
Word on
0 ms
1600 ms
Sliding window
200ms steps
Coh. win
Word on
0 ms
1600 ms

Phase lag index (PLI)

- Similar to PLV
- Measures asymmetry of circular distribution
- Reduces influence of zero-lag differences

Weighted PLI for near-zero jitters

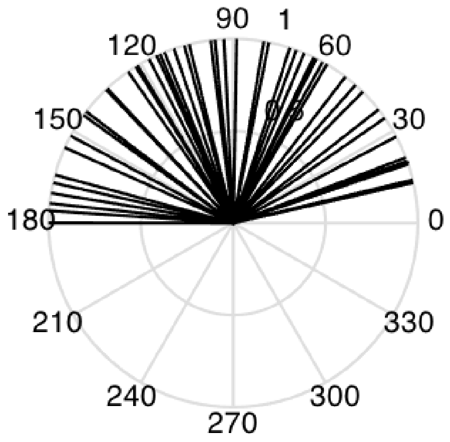
Same as PLV; taking the mean

Imaginary component of cross-spectrum = Phase differences

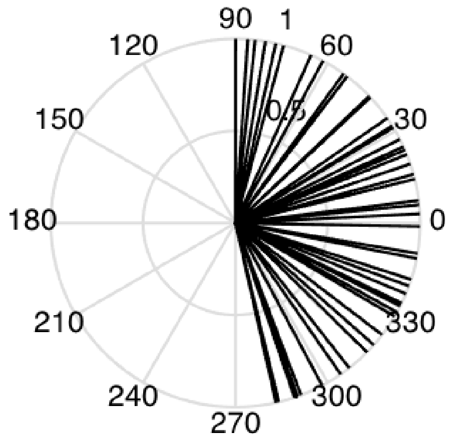
$$PLI_{xy} = \left| n^{-1} \sum_{t=1}^{n} sgn(imag(S_{xyt})) \right|$$

Only interested in whether above or below zero axis

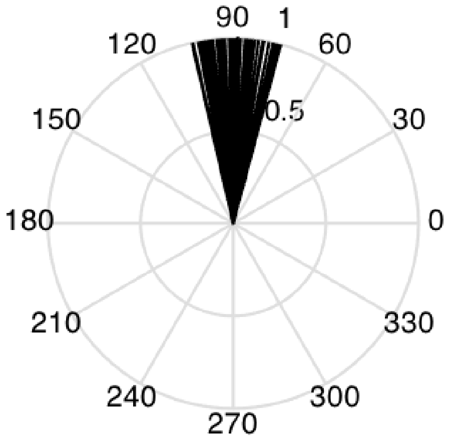
High PLI;
Low PLV



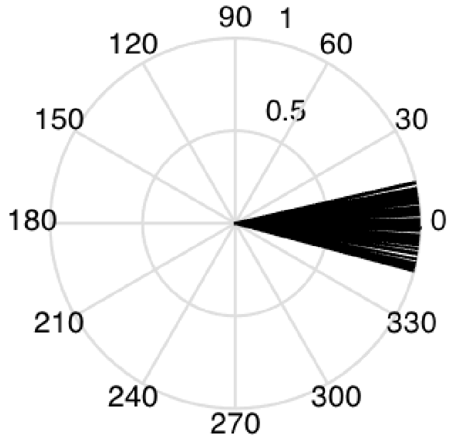
Low PLI;
Low PLV



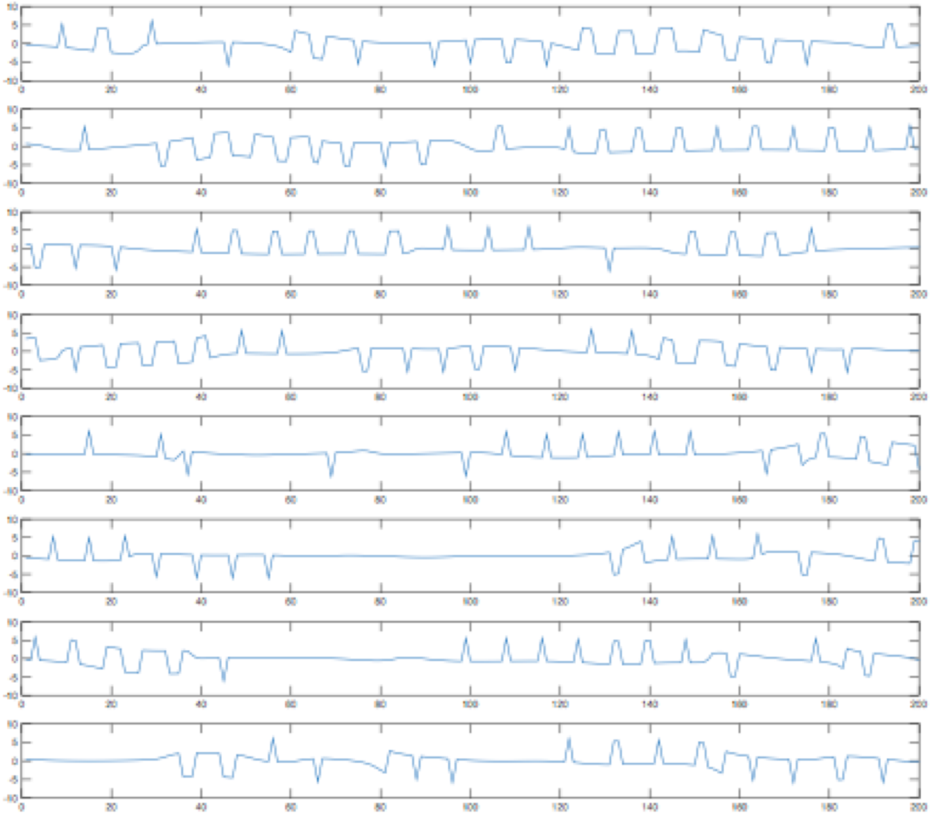
High PLI;
High PLV

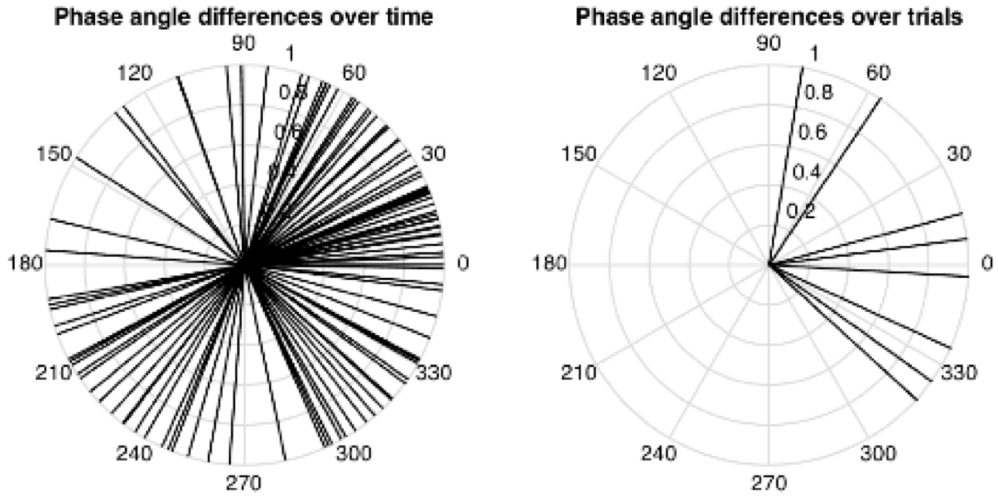


Low PLI;
High PLV



Time vs. Trials





$$PLV_t = \frac{1}{N}\left|\sum_{n=1}^{N} \exp\left(j\theta(t, n)\right)\right|$$

```
def resultant_vector_length(alpha, axis=None):
    #Returns the phase locking value (also called rbar) for an array of phase differences

    #Get PLV over a given axis, or the unraveled data
    if axis is None:
        axis = 0
        alpha = alpha.ravel()

    #Compute the mean resultant vector (Euler's formula)
    cmean = ( (np.exp(1j * alpha)).sum(axis=axis) / alpha.shape[axis] )

    #Not interested in sign -- take absolute value of mean
    r = np.abs(cmean)

    return r
```