

Methods in
Molecular Biology 1849

Springer Protocols

Robert G. Beiko
Will Hsiao
John Parkinson *Editors*

Microbiome Analysis

Methods and Protocols

EXTRAS ONLINE

 Humana Press

METHODS IN MOLECULAR BIOLOGY

Series Editor

John M. Walker

School of Life and Medical Sciences
University of Hertfordshire
Hatfield, Hertfordshire, AL10 9AB, UK

For further volumes:
<http://www.springer.com/series/7651>

Microbiome Analysis

Methods and Protocols

Edited by

Robert G. Beiko

Faculty of Computer Science, Dalhousie University, Halifax, NS, Canada

Will Hsiao

*Department of Pathology & Laboratory Medicine, University of British Columbia,
Vancouver, BC, Canada*

John Parkinson

*Program in Molecular Medicine, The Hospital for Sick Children, Toronto, ON, Canada;
Department of Biochemistry and Department of Molecular Genetics, University of Toronto,
Toronto, ON, Canada*



Editors

Robert G. Beiko
Faculty of Computer Science
Dalhousie University
Halifax, NS, Canada

Will Hsiao
Department of Pathology
& Laboratory Medicine
University of British Columbia
Vancouver, BC, Canada

John Parkinson
Program in Molecular Medicine
The Hospital for Sick Children
Toronto, ON, Canada

Department of Biochemistry
and Department of Molecular Genetics
University of Toronto
Toronto, ON, Canada

ISSN 1064-3745

ISSN 1940-6029 (electronic)

Methods in Molecular Biology

ISBN 978-1-4939-8726-9

ISBN 978-1-4939-8728-3 (eBook)

<https://doi.org/10.1007/978-1-4939-8728-3>

Library of Congress Control Number: 2018955280

© Springer Science+Business Media, LLC, part of Springer Nature 2018

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Humana Press imprint is published by the registered company Springer Science+Business Media, LLC, part of Springer Nature.

The registered company address is: 233 Spring Street, New York, NY 10013, U.S.A.

Preface

The microbiome, coined by Lederberg and McCray as “...the ecological community of commensal, symbiotic, and pathogenic microorganisms that literally share our body space” [1], draws together a remarkable number of disciplines with the overarching goal of understanding and ultimately harnessing the workings of microbial systems. True to the initial conception of the term, the human microbiome continues to be intensively studied, but microbial samples have been collected from nearly every imaginable habitat on Earth, from the upper atmosphere to the seabed subsurface, from hot springs to glacier ice, and from nematode guts to whale carcasses.

Microbiome analysis makes frequent use of a common set of tools to address many pertinent questions. A common workflow for microbiome analysis looks something like this: collect sample (e.g., soil, water, stool), extract DNA, perform DNA sequencing, and use bioinformatics tools to describe important properties of the microbiome. This pipeline has been applied to huge numbers of samples from a diverse array of environments. In particular, the targeted sequencing via polymerase chain reaction (PCR) amplification of “marker” genes that are seen as diagnostic of different types of microorganisms has seen widespread use. The workhorse of microbial diversity has thus far been the 16S ribosomal RNA gene, which has been the subject of intensive protocol development: see for example the Earth Microbiome Project protocols [2], and a detailed evaluation of 16S sequencing on the Illumina sequencing platforms [3]. However, while capturing the *taxonomic composition* of a microbial community, marker-gene sequencing is limited in its ability to reveal the diversity of *functions* present, requiring the application of alternative approaches.

Gaining an accurate and relevant picture of the microbiome requires careful experimental design, and the first part of our book focuses on the profiling of different habitats and elements of biodiversity. The procedures to collect representative and uncontaminated samples can be highly complex; one need look no further than Chapter 1 for an example of the challenges associated with collecting microbial samples from the deep subsurface. DNA sequencing might be seen as the foundation of microbial community analysis, even if arguably the first such analysis was done with RNA rather than DNA sequencing in the famous Octopus Spring study [4]. However, other “meta-omic” methods that consider messenger RNA transcripts, protein products, or metabolite levels can reveal a great deal more about microbial activities in a particular habitat. The combination of multiple such strategies can be especially powerful, as illustrated by the use of DNA sequencing to support targeted metaproteomics (Chapter 6).

The remainder of this volume addresses the computational challenges of microbiome analysis. An immense number of algorithms and software packages have been developed for the task, and even seemingly simple questions as “what is the biodiversity present in a given sample?” may not be straightforward, as exemplified by Chapter 10. At the same time, the rich information generated from these samples is driving the development of innovative tools and pipelines with the ability to generate novel data types and address new questions, such as the recovery of complete genomes from metagenomes (Chapter 14), and the use of network approaches to identify patterns of microbial association (Chapter 17).

Although no book on microbiome analysis can be exhaustive, in preparing our volume we have sought to convey what might be seen as standard practice in the field (to the extent

anything can be claimed as such!), while also highlighting techniques at the frontiers of the field that challenge standard practice. Reflecting the continued dominance of marker-gene approaches, the QIIME package [5] recently received its ten-thousandth citation: the recent release of the second version of this software is notable because it is developed in a completely different framework, and because it upends some of the tools and techniques that have heretofore served as its defaults (*see Chapter 8*).

By defining procedures in precise terms, the *Methods in Molecular Biology* series contributes to the reproducibility of experiments. However, reproducibility in bioinformatics is a big concern [6], with several moving parts including database versions, software updates, and parameter settings. Comparing new methods to existing ones demands that final results and all intermediate steps can be regenerated. The last few years have seen significant advances in reproducibility through means such as automated workflow tools including Galaxy, interactive code tools such as Jupyter Notebooks, and repositories with version control, the most notable example of which is currently Github. We are pleased that many of our authors have provided examples that make use of these tools, which will make it considerably easier for readers to perform analyses in a consistent manner.

It remains only for us to thank the individuals who have contributed their time and hard work to preparing a highly diverse and engaging set of chapters. We are also grateful to John Walker for the original invitation to prepare this book.

Halifax, NS, Canada
Vancouver, BC, Canada
Toronto, ON, Canada

Robert G. Beiko
Will Hsiao
John Parkinson

References

1. Lederberg J, McCray AT (2001) Ome SweetOmics—a genealogical treasury of words. *Scientist* 15;8
2. Earth Microbiome Project. Protocols and Standards. <http://www.earthmicrobiome.org/protocols-and-standards/>. Accessed 3 March 2018
3. Caporaso JG, Lauber CL, Walters WA, et al (2012) Ultra-high-throughput microbial community analysis on the Illumina HiSeq and MiSeq platforms. *ISME J* 6:1621
4. Stahl DA, Lane DJ, Olsen GJ, et al (1985) Characterization of a Yellowstone hot spring microbial community by 5S rRNA sequences. *Appl Environ Microbiol* 49:1379–1384
5. Caporaso JG, Kuczynski J, Stombaugh J et al (2010). QIIME allows analysis of high-throughput community sequencing data. *Nat Methods* 7:335
6. Sandve GK, Nekrutenko A, Taylor J et al (2013) Ten simple rules for reproducible computational research. *PLoS Comput Biol* 9:e1003285

Contents

<i>Preface</i>	v
<i>Contributors</i>	ix
1 Characterizing the Deep Terrestrial Subsurface Microbiome <i>Rebecca A. Daly, Kelly C. Wrighton, and Michael J. Wilkins</i>	1
2 Freshwater Viromes: From Sampling to Evaluation <i>Catherine Putonti, Zoë Diener, and Siobhan C. Watkins</i>	17
3 Characterization of Eukaryotic Microbiome Using 18S Amplicon Sequencing <i>Ana Popovic and John Parkinson</i>	29
4 Culture and Molecular Profiling of the Respiratory Tract Microbiota <i>Fiona J. Whelan, Laura Rossi, Jennifer C. Stearns, and Michael G. Surette</i>	49
5 Methods and Strategies to Examine the Human Breastmilk Microbiome <i>Lauren LeMay-Nedjelski, Julia Copeland, Pauline W. Wang, James Butcher, Sharon Unger, Alain Stintzi, and Deborah L. O'Connor</i>	63
6 Quantification of Vitamin B₁₂-Related Proteins in Marine Microbial Systems Using Selected Reaction Monitoring Mass Spectrometry <i>Erin M. Bertrand</i>	87
7 Single-Cell Genomics of Microbial Dark Matter <i>Christian Rinke</i>	99
8 16S rRNA Gene Analysis with QIIME2 <i>Michael Hall and Robert G. Beiko</i>	113
9 Processing a 16S rRNA Sequencing Dataset with the Microbiome Helper Workflow <i>Gavin M. Douglas, André M. Comeau, and Morgan G. I. Langille</i>	131
10 Normalization of Microbiome Profiling Data <i>Paul J. McMurdie</i>	143
11 Predicting the Functional Potential of the Microbiome from Marker Genes Using PICRUSt <i>Gavin M. Douglas, Robert G. Beiko, and Morgan G. I. Langille</i>	169
12 Metagenome Assembly and Contig Assignment <i>Qingpeng Zhang</i>	179
13 From RNA-seq to Biological Inference: Using Compositional Data Analysis in Meta-Transcriptomics <i>Jean M. Macklaim and Gregory B. Gloor</i>	193
14 Subsampled Assemblies and Hybrid Nucleotide Composition/Differential Coverage Binning for Genome-Resolved Metagenomics <i>Laura A. Hug</i>	215

15	Transkingdom Networks: A Systems Biology Approach to Identify Causal Members of Host–Microbiota Interactions	227
	<i>Richard R. Rodrigues, Natalia Shulzhenko, and Andrey Morgun</i>	
16	Constructing and Analyzing Microbiome Networks in R	243
	<i>Mehdi Layeghifard, David M. Hwang, and David S. Guttman</i>	
17	Bayesian Inference of Microbial Community Structure from Metagenomic Data Using BioMiCo	267
	<i>Katherine A. Dunn, Katelyn Andrews, Rana O. Bashwib, and Joseph P. Bielawski</i>	
18	Analyzing Metabolic Pathways in Microbiomes	291
	<i>Mobolaji Adeolu, John Parkinson, and Xuejian Xiong</i>	
19	Sparse Treatment-Effect Model for Taxon Identification with High-Dimensional Metagenomic Data	309
	<i>Zhenqiu Liu and Shili Lin</i>	
	<i>Index</i>	319

Contributors

MOBOLAJI ADEOLU • *Program in Molecular Medicine, The Hospital for Sick Children, Toronto, ON, Canada*

KATELYN ANDREWS • *Department of Mathematics and Statistics, Dalhousie University, Halifax, NS, Canada*

RANA O. BASHWIH • *Department of Mathematics and Statistics, Dalhousie University, Halifax, NS, Canada*

ROBERT G. BEIKO • *Faculty of Computer Science, Dalhousie University, Halifax, NS, Canada*

ERIN M. BERTRAND • *Department of Biology, Dalhousie University, Halifax, NS, Canada*

JOSEPH P. BIELAWSKI • *Department of Mathematics and Statistics, Dalhousie University, Halifax, NS, Canada*

JAMES BUTCHER • *Ottawa Institute of Systems Biology, Ottawa, ON, Canada; Department of Microbiology and Immunology, University of Ottawa, Ottawa, ON, Canada*

ANDRÉ M. COMEAU • *Integrated Microbiome Resource, Dalhousie University, Halifax, NS, Canada*

JULIA COPELAND • *Centre for the Analysis of Genome Evolution and Function, University of Toronto, Toronto, ON, Canada*

REBECCA A. DALY • *Department of Microbiology, The Ohio State University, Columbus, OH, USA*

ZOË DIENER • *Department of Biology, New Mexico Institute for Mining and Technology, Socorro, NM, USA*

GAVIN M. DOUGLAS • *Department of Microbiology and Immunology, Dalhousie University, Halifax, NS, Canada*

KATHERINE A. DUNN • *Department of Mathematics and Statistics, Dalhousie University, Halifax, NS, Canada*

GREGORY B. GLOOR • *Department of Biochemistry, Schulich School of Medicine and Dentistry, The University of Western Ontario, London, ON, Canada; Canadian Centre for Human Microbiome and Probiotic Research, Lawson Health Sciences Centre, London, ON, Canada*

DAVID S. GUTTMAN • *Department of Cell and Systems Biology, University of Toronto, Toronto, ON, Canada; Centre for the Analysis of Genome Evolution and Function, University of Toronto, Toronto, ON, Canada*

MICHAEL HALL • *Dalhousie University, Halifax, NS, Canada*

LAURA A. HUG • *Department of Biology, University of Waterloo, Waterloo, ON, Canada*

DAVID M. HWANG • *Department of Pathology, University Health Network, Toronto, ON, Canada; Department of Laboratory Medicine and Pathobiology, University of Toronto, Toronto, ON, Canada*

MORGAN G. I. LANGILLE • *Department of Microbiology and Immunology, Dalhousie University, Halifax, NS, Canada; Integrated Microbiome Resource, Dalhousie University, Halifax, NS, Canada; Department of Pharmacology, Dalhousie University, Halifax, NS, Canada*

MEHDI LAYEGHIFARD • *Department of Cell and Systems Biology, University of Toronto, Toronto, ON, Canada*

LAUREN LE MAY-NEDJELSKI • *Faculty of Medicine, Department of Nutritional Sciences, University of Toronto, Toronto, ON, Canada; Translational Medicine, The Hospital for Sick Children, Toronto, ON, Canada*

SHILI LIN • *Department of Statistics, The Ohio State University, Columbus, OH, USA*

ZHENQIU LIU • *Samuel Oschin Comprehensive Cancer Institute, Cedars-Sinai Medical Center, Los Angeles, CA, USA*

JEAN M. MACKLAIM • *Department of Biochemistry, Schulich School of Medicine and Dentistry, The University of Western Ontario, London, ON, Canada; Canadian Centre for Human Microbiome and Probiotic Research, Lawson Health Sciences Centre, London, ON, Canada*

PAUL J. McMURDIE • *Whole Biome, Inc., San Francisco, CA, USA*

ANDREY MORGUN • *College of Pharmacy, Oregon State University, Corvallis, OR, USA*

DEBORAH L. O'CONNOR • *Faculty of Medicine, Department of Nutritional Sciences, University of Toronto, Toronto, ON, Canada; Translational Medicine, The Hospital for Sick Children, Toronto, ON, Canada; Department of Pediatrics, Mount Sinai Hospital, Toronto, ON, Canada*

JOHN PARKINSON • *Program in Molecular Medicine, The Hospital for Sick Children, Toronto, ON, Canada; Department of Biochemistry and Department of Molecular Genetics, University of Toronto, Toronto, ON, Canada*

ANA POPOVIC • *Program in Molecular Medicine, The Hospital for Sick Children, Toronto, ON, Canada; Department of Biochemistry, University of Toronto, Toronto, ON, Canada*

CATHERINE PUTONTI • *Department of Biology, Loyola University Chicago, Chicago, IL, USA; Department of Computer Science, Loyola University Chicago, Chicago, IL, USA; Bioinformatics Program, Loyola University Chicago, Chicago, IL, USA; Department of Microbiology and Immunology, Stritch School of Medicine, Loyola University Chicago, Maywood, IL, USA*

CHRISTIAN RINKE • *Australian Centre for Ecogenomics, University of Queensland, Brisbane, QLD, Australia*

RICHARD R. RODRIGUES • *College of Pharmacy, Oregon State University, Corvallis, OR, USA*

LAURA ROSSI • *Department of Biochemistry and Biomedical Sciences, McMaster University, Hamilton, ON, Canada*

NATALIA SHULZHENKO • *College of Veterinary Medicine, Oregon State University, Corvallis, OR, USA*

JENNIFER C. STEARNS • *Department of Medicine, McMaster University, Hamilton, ON, Canada*

ALAIN STINTZI • *Ottawa Institute of Systems Biology, Ottawa, ON, Canada; Department of Microbiology and Immunology, University of Ottawa, Ottawa, ON, Canada*

MICHAEL G. SURETTE • *Department of Medicine, McMaster University, Hamilton, ON, Canada*

SHARON UNGER • *Translational Medicine, The Hospital for Sick Children, Toronto, ON, Canada; Faculty of Medicine, Department of Pediatrics, University of Toronto, Toronto, ON, Canada; Department of Pediatrics, Mount Sinai Hospital, Toronto, ON, Canada*

PAULINE W. WANG • *Centre for the Analysis of Genome Evolution and Function, University of Toronto, Toronto, ON, Canada*

SIOBHAN C. WATKINS • *Department of Biology, New Mexico Institute for Mining and Technology, Socorro, NM, USA*

FIONA J. WHELAN • *Department of Biochemistry and Biomedical Sciences, McMaster University, Hamilton, ON, Canada*

MICHAEL J. WILKINS • *Department of Microbiology, The Ohio State University, Columbus, OH, USA*

KELLY C. WRIGHTON • *Department of Microbiology, The Ohio State University, Columbus, OH, USA*

XUEJIAN XIONG • *Program in Molecular Medicine, The Hospital for Sick Children, Toronto, ON, Canada*

QINGPENG ZHANG • *Department of Energy, Joint Genome Institute, Walnut Creek, CA, USA*



Chapter 1

Characterizing the Deep Terrestrial Subsurface Microbiome

Rebecca A. Daly, Kelly C. Wrighton, and Michael J. Wilkins

Abstract

A large portion of the earth's biomass resides in the subsurface and recent studies have expanded our knowledge of indigenous microbial life. Advances in the field of metagenomics now allow analysis of microbial communities from low-biomass samples such as deep (>2.5 km) shale core samples. Here we present protocols for the best practices in contamination control, handling core material, extraction of nucleic acids, and low-input library preparation for subsequent metagenomic sequencing.

Key words Deep life, Deep biosphere, Terrestrial subsurface, Contamination, Shale, Metagenomics, Low biomass

1 Introduction

While it is estimated that the terrestrial subsurface is the largest reservoir of life on Earth, hosting between 40% and 60% of all bacterial cells, densities are typically low in deep subsurface ecosystems [1, 2]. Research into the deep terrestrial biosphere stems from interest in biogeochemical cycling and the discovery of novel biodiversity and metabolisms [3]. However, due to the difficulty in obtaining core samples from thousands of meters below the Earth's surface and characteristic low biomass, it is critical that samples are collected and preserved in a manner to limit contamination. Advances in metagenomic sequencing technology which permits library construction from picogram quantities of DNA can now be utilized to examine not only the presence of single-gene markers (i.e., 16S rRNA genes), but allow for reconstruction of entire microbial genomes, providing insight into functional potential within the deep terrestrial subsurface microbiome.

There is a large body of published work detailing how to deal with contamination of low-biomass samples during drilling and coring, and the sources of contamination [3–6]. Small amounts of contaminating bacteria can mask the signal from indigenous

microorganisms and compromise future analyses. Methods often include the use of tracers during the drilling process, including microbial tracers (e.g., live cells) [7], chemical tracers such as perfluorocarbons [8], and visual tracers such as fluorescein [9] and fluorescent microspheres [4, 10–13]. The collection of “blanks” at multiple points of sample processing provides the ability to distinguish contaminating microorganisms from indigenous life. Once contamination controls have been implemented, the question still remains of how to extract DNA from low-biomass rock matrices.

DNA is commonly extracted by using a combination of chemical and physical lysis [14–18] in order to recover and subsequently purify DNA from lysed cells, usable for downstream molecular assays. There are a wide range of DNA-extraction protocols, with certain methods optimized for particular sample types, including commercial kits. For challenging and unique samples, there may be no established methods. There is no “one-size-fits-all” DNA-extraction protocol, and it is recommended that multiple methods are tested and compared on sample material physically and chemically similar to the targeted samples. Once DNA has been successfully extracted and purified from the matrix, commercial kits are available for low-input library preparation for metagenomic sequencing [19, 20].

This chapter outlines procedures for contamination control in the field, the use of fluorescent microsphere tracers during drilling, contamination control in the lab, sample decontamination/cleaning, sample grinding, DNA extraction and library preparation for metagenomic sequencing for core samples obtained from the Marcellus shale formation in West Virginia, USA.

2 Materials

2.1 Collection and DNA Extraction of Drilling Muds and Other Fluids

1. Nalgene 1 L wide-mouth HDPE bottles, autoclaved.
2. 0.2 µm polyethersulfone (PES) vacuum filter device.
3. MoBio PowerMax Soil DNA isolation kit (MoBio Laboratories, Carlsbad, CA, USA).
4. Standard PCR reagents and primers for 16S rRNA gene amplification.

2.2 Fluorescent Microspheres

1. Fluoresbrite YG carboxylate microspheres 0.50 µm (Polysciences, Warrington, PA, USA).
2. Sterile carboy or other similar container filled with deionized water with a sensitivity of 18 MΩ cm at 25 °C.

2.3 Core Collection

1. Whirl-Pak bags (Nasco, Fort Atkinson, WI, USA) of the appropriate size to hold a single core.
2. Anaeropak 7.0 L rectangular jars (Mitsubishi, Tokyo, Japan).
3. AnaeroPouch System-Anaero Anaerobic Gas Generator sachets (Mitsubishi, Tokyo, Japan).

2.4 Quantification of Microspheres

1. Inverted, fluorescent microscope with a 10 \times objective (for 100 \times total magnification). Ensure that the microscope has enough clearance to allow the core sample to be placed on the stage. We used an Eclipse Ti inverted microscope (Nikon, Tokyo, Japan).
2. Software for obtaining color micrographs.

2.5 Core Sample Decontamination

1. 1.5 M NaCl solution: 87.66 g of NaCl. Add DNA-free water to a volume of 1 L. Mix and autoclave.
2. Autoclaved aluminum dish with fluted sides, ~200 mL (volume dependent on the size of core samples).
3. Autoclaved high quality, fine (size 00) steel wool. Autoclaved steel wool wrapped in loose aluminum foil. After autoclaving dry in a 50 °C oven to prevent rusting.

2.6 Grinding of Core Samples

1. Bunsen burner.
2. Metal tongs for holding core samples.
3. Autoclaved 12 \times 12" sheets of aluminum foil.
4. Cleaned, autoclaved ceramic mortars and pestles.
5. Brass mesh sieves with SS wire, including sieve cover and bottom (e.g., stackable 3" sieves in #10, #18, and #35 mesh (Humboldt Manufacturing Company, Elgin, IL, USA)).
6. Plattner's hardened steel mortars and pestles (Humboldt Manufacturing Company, Elgin, IL, USA).
7. Cold chisel, ½" cut width.
8. Sledge/drilling hammer with compact handle (~3 lb. hammer).

2.7 DNA Extraction from Ground Core

1. DNAZap PCR DNA Degradation solutions (ThermoFisher Scientific, Waltham, MA, USA).
2. Lysis Buffer I, pH 10, 250 mL: Add 17.5 mL of 1.0 M, pH 7.5 Tris-HCl; 15.0 mL of 0.5 M, pH 8.0 EDTA, 25.0 mL of 8.0 M guanidine hydrochloride, and 1.25 mL of Triton X-100 to a sterile flask. Add DNA-free water to a volume of 250 mL. Mix and adjust pH to 10.0 with NaOH. Filter sterilize through a 250 mL, 0.1 μ m, PES vacuum filter unit.
3. DNA LoBind 1.5 mL tubes (Eppendorf, Hauppauge, NY, USA).

4. Ultra-high-speed 50 mL centrifuge tubes (VWR International, Radnor, PA, USA).
5. Phenol:chloroform:isoamyl alcohol, 25:24:1, pH 8.0.
6. Chloroform:isoamyl alcohol, 24:1.
7. 100% ethanol.
8. 70% ethanol, prepared with DNA-free water.
9. EB buffer (Qiagen, Valencia, CA, USA).
10. Linear acrylamide, 5 mg/mL (ThermoFisher Scientific, Waltham, MA, USA).

2.8 DNA Quantification

1. Qubit fluorometer (ThermoFisher Scientific, Waltham, MA, USA).
2. Qubit dsDNA HS (high sensitivity) assay kit (ThermoFisher Scientific, Waltham, MA, USA).
3. Qubit assay tubes, 0.5 mL (ThermoFisher Scientific, Waltham, MA, USA).

2.9 Library Preparation for Metagenomic Sequencing and Optional MDA Amplification

1. REPLI-g Single Cell WGA kit (Qiagen, Valencia, CA, USA).
2. Nextera XT DNA library preparation kit (Illumina, Inc., San Diego, CA, USA).
3. Nextera XT Index kit (Illumina, Inc., San Diego, CA, USA).
4. Agencourt AMPure XP magnetic beads (Beckman Coulter Life Sciences, Indianapolis, IN, USA).
5. Magnetic stand for 1.5 mL microcentrifuge tubes.
6. Bioanalyzer Instrument (Agilent Technologies, Santa Clara, CA, USA).
7. Bioanalyzer High Sensitivity DNA kit (Agilent Technologies, Santa Clara, CA, USA).

3 Methods

3.1 Contamination Control in the Field

Recovering core material from the subsurface requires drilling technologies which can introduce contamination from several sources, including drilling mud additives, surface water mixed with drilling muds, contamination from mud tanks, pumping equipment, and contamination from overlying formations and groundwater. In order to obtain reliable information about indigenous microorganisms, it is extremely important to sample all sources of potential contamination, extract DNA, and sequence these samples for subtractive analysis. Chemical and particle tracers are commonly used to assess sample integrity. Here we describe use of fluorescent microspheres as a visual tracer, and DNA extraction of the fluids used to clean core samples as a molecular tracer, to ensure sample integrity.

3.1.1 Sampling Drilling Muds and Other Fluids

1. Sample all fluids that can potentially go down-well, before and during the drilling process starts, including freshwater added to drilling muds, and the drilling muds (*see Note 1*).
2. Ensure that all sampling equipment is sterile, and wear disposable gloves at all times. It is useful to have large ladles and buckets on hand that are easily sterilized with a bleach solution or 70% ethanol, for retrieving samples at the well pad.
3. Filter freshwater sources through a sterile, large-volume filter apparatus with a 0.2 µm polyethersulfone (PES) membrane, filtering a minimum of 3 L of each sample for future DNA extraction and analysis. If the samples cannot be filtered in the field, collect samples in sterile 1 L Nalgene containers, or larger carboys, filled to the brim to minimize headspace and thus alteration of the microbial community and transport on ice. If the samples can be filtered in the field, have dry ice on hand to rapidly freeze the filters during transport to the laboratory.
4. Collect drilling muds in 1 L sterile Nalgene containers.
5. DNA from freshwater source filters (~5 g of filter material) and aliquots of drilling muds (~5 mL) can be extracted using the manufacturer's recommended protocol using the PowerMax Soil DNA isolation kit (*see Note 2*).
6. Test for PCR amplification using "universal" 16S rRNA gene primers to ensure that samples are suitable for downstream analyses (metagenomic library preparation).

3.1.2 Addition of Fluorescent Microspheres to Drilling Muds

During the drilling process, drilling muds and other fluids can penetrate core samples. We used Fluoressbrite carboxylate yellow-green fluorescent polystyrene microspheres 0.50 µm in diameter as a proxy for contaminating bacterial cells, and quantified microsphere penetration and removal during the decontamination process by fluorescence microscopy.

1. In order to calculate the amount of microspheres to add to the drilling mud, it is necessary to determine the volume of the drilling mud in the tank (*see Note 3*). Drilling muds are contained in large mixing tanks, where the liquid is constantly being mixed to maintain homogeneity and allow for the addition of drilling mud chemicals as required by the operator.
2. Calculate the volume of fluorescent microspheres needed to obtain a concentration of $\sim 5 \times 10^5$ microspheres/mL in the drilling mud tank.
3. In order to ensure that the microspheres are dispersed evenly throughout the drilling mud, first dilute the required volume of microspheres in a carboy or other similar sterile container containing deionized water with a sensitivity of 18 MΩ cm at 25 °C.

4. Mix well and add to the mud tank while the drilling mud is being mixed.
5. After mixing, sample drill muds for microsphere quantification and DNA extraction (*see Note 4*).

3.1.3 Core Collection

Cores from black shale formations contain high amounts of hydrocarbons and extremely low biomass. In order to obtain samples adequate for metagenomic analyses, rigorous sterile technique, proper storage/transport conditions, and minimal time lag before processing samples are critical.

1. Be on site during the actual collection of core samples. Arrange with the operator on-site to have access to cores immediately.
2. Label Whirl-Pak bags with pertinent core information, including the well, depth, and time of sampling.
3. Photograph each core before inserting in a Whirl-Pak bag (dimensions dependent on core size), as some cores are recovered intact, while others are recovered in small pieces.
4. If the reservoir is pressurized, it may be necessary to pierce each Whirl-Pak bag with a thin (25G) needle so that sample bags do not burst due to de-gassing of the cores.
5. Place Whirl-Pak bags with cores in Anaeropak 7.0 L rectangular jars, with 3, O₂-consuming sachets in each jar.
6. Transport cores in anaerobic boxes on ice, back to the laboratory as quickly as possible.

3.2 Contamination Control in the Lab

3.2.1 Quantification of Microspheres Prior to Decontamination

Minimize time at room-temp and freeze-thaw cycles. Process samples as quickly as possible, storing at 4 °C during daily processing steps and store ground samples at -80 °C for long-term storage.

1. Carefully wipe the microscope stage with 70% ethanol.
2. Drill mud samples with microspheres can be quantified by putting 10–20 µL on a glass slide covered by a glass coverslip.
3. Place the core sample on the microscope stage, being careful to ensure no drilling mud or core particles contact the objectives. Microspheres on core samples can be enumerated by taking at least four images, one on each end of the core, and two images on each side.
4. Images can be saved using the appropriate software for the microscope, and counted at a later date, to minimize the time the sample is exposed to the air.
5. For each image, count the number of microspheres in each field of view (or multiple, smaller windows from each image).

6. Clean the microscope stage with 70% ethanol between each sample.
7. Placing autoclaved aluminum foil on the stage with an opening for the objective helps keep the microscope stage clean and prevents cross-contamination.

3.2.2 Core Sample Decontamination

There are many described methods for decontamination of core samples, including paring via circular saws with hydraulic crushing, and hammer and/or chisel. Here we determined that the best results were obtained from three successive NaCl washes while scrubbing with fine steel wool. Due to the relatively soft nature of the shale samples from our studies, this resulted in removal of a thin layer of the exterior, past the point of penetration of microspheres, and thus microbial contamination.

1. Perform all work in a laminar flow hood dedicated to core samples.
2. Wipe all surfaces, equipment, and pipettes with 70% ethanol.
3. Put all reagents, supplies, and pipettes in the laminar flow hood, UV sterilize for 30 min. Rotate pipettes and supplies, UV sterilize for an additional 30 min.
4. Place three weigh dishes in the laminar flow hood and aliquot ~50 mL of 1.5 M NaCl solution in each dish.
5. Place a small piece (~2" × 2") of steel wool in each dish.
6. Place the core sample in the first dish, and while wetting the steel wool, scrub the exterior of the core sample with small circular motions. You may notice sloughed off core particles in the NaCl solution.
7. Remove the core from the first wash, and place it in the second dish. Repeat scrubbing the exterior of the core sample with the fresh NaCl solution and new steel wool. Repeat the process in the third wash.
8. Photograph the core sample after the decontamination/cleaning process for documentation purposes.
9. Place the core in a new, labeled, Whirl-Pak bag.

3.2.3 Quantification of Microspheres Prior to Decontamination

Repeat the process in Subheading [3.2.1](#), taking images of the cleaned core surface. It is likely that no microspheres will be detected after the decontamination procedure. If microspheres are observed, repeat the core sample decontamination procedure in Subheading [3.2.2](#) and reimagine the core until no microspheres are detected.

3.2.4 Grinding of Core Samples

1. Perform all work in a laminar flow hood dedicated to core samples.
2. Wipe all surfaces, equipment, and pipettes with 70% ethanol.
3. Put all reagents, supplies, and pipettes in the laminar flow hood, UV sterilize for 30 min. Rotate pipettes and supplies, UV sterilize for an additional 30 min.
4. Spread a sheet of sterile aluminum foil on the working surface.
5. Light a Bunsen burner near the opening to the laminar flow hood. Holding the core sample in metal tongs, quickly pass the core sample through the flame twice while rotating the core slightly, and place on the aluminum foil sheet in the laminar flow hood. Allow the core exterior to cool for 1 min.
6. Using a sledge/drilling hammer and cold chisel, break the core material into smaller pieces, until they will fit in the Plattner's mortar sleeve. Place the pestle in the mortar and strike the pestle with the sledge/drilling hammer. Repeat until the sample is broken into smaller pieces (*see Note 5*).
7. Once the sample is broken into small pieces (*see Note 6*), transfer the material to the series of stacked sieves, with the largest mesh at the top. Gently shake with a side-to-side motion.
8. Remove the material from the base of the stacked sieves and place in a sterile glass or plastic container for storage.
9. Remove the material in the larger sieves that did not pass through to the base and place in the Plattner's mortar and pestle or a standard ceramic mortar and pestle to further reduce the size. Pass through the stacked sieves until all sample is ground to the desired size.
10. Store ground core samples at -80 °C until DNA extraction.

3.3 DNA Extraction and Quantification from Ground Core

It is essential to use a laminar flow hood dedicated to core samples during processing. Thoroughly clean the hood before and after any procedure. The use of extra-long nitrile gloves to cover the exposed wrist region and disposable Tyvek lab coats is recommended. It is critical to treat every "blank" exactly as the actual sample and even if the "blanks" result in non-detectable DNA, to carry the "blanks" through to library preparation and subsequent sequencing. Low-level contamination has been well documented even in commercial nucleic acid extraction and purification and amplification kits [21, 22]. DNA sorption onto mineral surfaces is an additional problem with low-biomass samples. The use of blocking agents and carrier molecules have been shown to help overcome this difficulty [23, 24], and is extensively presented in [16]. However our tests showed that with these shale core samples, blocking agent and

carrier molecules had deleterious effects on DNA recovery and additionally contained high levels of microbial contamination.

3.3.1 DNA Extraction from Ground Core

1. Perform all work in a laminar flow hood dedicated to core samples.
2. Wipe all surfaces, equipment, and pipettes with 70% ethanol.
3. Put all reagents, supplies, and pipettes in the laminar flow hood, UV sterilize for 30 min. Rotate pipettes and supplies, UV sterilize for an additional 30 min.
4. Turn on the oven to 50 °C.
5. Label all tubes, including the same number of tubes for blanks as for each sample (e.g., if you are extracting from a total 20 g of ground sample, you will need 40 1.5 mL tubes containing 0.5 g of sample each and 40 1.5 mL tubes for the extraction blank).
6. Spray all surfaces with (first) DNAzap solution 1, then DNAzap solution 2, wipe with sterile paper towels, then with sterile water.
7. Put 0.5 g ground shale into each 1.5 mL tube (nothing for blanks). The total number of 0.5 g aliquots is dependent on the total mass of sample to extract from (*see Note 7*).
8. Add 1.0 mL of lysis buffer I solution to each tube, vortex to mix and freeze at –80 °C.
9. Thaw samples, vortex, and incubate for 1 h at 50 °C, vortexing every 10 mins.
10. Centrifuge samples at 10k × g for 3 min at room temperature, transfer supernatants from replicate extractions to a 50 mL conical tube.
11. Put 50 mL tube on ice and extract again from each tube containing ground shale, and extraction blank tubes, by adding 1.0 mL of lysis buffer I solution, vortex to mix.
12. Centrifuge samples at 10k × g for 3 min at room temperature, transfer supernatants from replicate extractions to a 50 mL conical tube.
13. Add one volume of phenol:chloroform:isoamyl alcohol to the combined samples, mix by inversion, centrifuge at 10k × g for 10 min at room temperature, transfer supernatant to new 50 mL tube.
14. Add one volume of chloroform:isoamyl alcohol to the samples, mix by inversion, centrifuge at 10k × g for 5 min at room temperature, transfer supernatant to a new 50 mL tube.
15. Repeat chloroform:isoamyl purification in **step 14**.

16. Precipitate by adding 30 μ L of linear acrylamide and 0.2 volumes of 5 M NaCl solution (*see Note 8*). Calculate the new total volume, then add 2.5 volumes of 100% ethanol. The precipitation may need to be divided between multiple 50 mL conical tubes.
17. Incubate for 2 h at room temperature or overnight at 4 °C.
18. Centrifuge samples at 12k \times g for 30 min at room temperature.
19. Wash 2 \times with 70% ethanol, centrifuge at max speed for 5–15 min, dry pellet in the laminar flow hood by inverting tube on sterile aluminum foil.
20. Resuspend pellets in 50–100 μ L in warm (50 °C) EB buffer. Use the minimum volume possible to resuspend pellets.
21. Store extracted DNA at –80 °C or quantify immediately.

3.3.2 Quantification of DNA

It is important to use as small a volume of sample as possible during quantification. The Qubit dsDNA HS 2.0 fluorometer protocol allows for up to 20 μ L of sample to be quantified. However the sample is not recoverable after measurement. It is not always necessary to quantify the amount of DNA recovered for extremely low-biomass samples, as future library preparation for metagenomic sequencing can tolerate a wide variety of template concentrations. The Qubit dsDNA HS fluorometer 2.0 assay has a detection limit of 10 pg/ μ L when using 20 μ L of sample. If DNA concentrations are below the Qubit dsDNA HS detection limit, 10 pg/ μ L can be used as an upper limit to estimate template input for library preparation. Other fluorescence-based assays, such as the Quant-iT PicoGreen dsDNA assay, can also be used but the use of a nanodrop is not recommended, as some co-extracted contaminants will cause interference and will inflate the detected concentrations.

1. Wipe all surfaces, equipment, and pipettes with 70% ethanol.
2. Put all reagents, supplies, and pipettes (except for samples and Qubit standards) in the laminar flow hood, UV sterilize for 30 min. Rotate pipettes and supplies, UV sterilize for an additional 30 min.
3. Thaw DNA on ice.
4. Prepare a master mix according to the manufacturer's instructions, appropriate for the number of samples to be quantified.
5. Aliquot samples and standards.
6. Measure using the Qubit fluorometer and record detected DNA concentrations.
7. Store extracted DNA at –80 °C or proceed immediately to metagenomic library preparation.

3.3.3 Library Preparation and Optional MDA Amplification

Typical metagenomic library preparations require input DNA concentrations ranging from several nanograms up to a microgram. Yet DNA extractions from samples with low cell density yield DNA masses in the femtogram to picogram range (~1 fg DNA per cell). Methods have been developed to circumvent low DNA yields, such as MDA amplification which produces millions of copies of template DNA and is used for single-cell genomic amplification; commercial kits such as the Nextera XT DNA library preparation kit (recommended input 1 ng DNA), and the Accel-NGS 1S Plus library kit (inputs as low as 10 pg) which was developed for ancient, degraded and/or ssDNA. Despite the fact that MDA has been shown to have inherent biases during amplification, it may be necessary for certain samples when library preparation fails with non-detectable input DNA; alternatively, the extracted DNA can be split and libraries prepared with and without MDA amplification.

It is essential to use dedicated laminar flow hoods. Thoroughly clean the hood before and after any procedure. The use of extra-long nitrile gloves to cover the exposed wrist region and disposable Tyvek lab coats is recommended. It is critical to treat every “blank” exactly as the actual sample and even if the “blanks” result in non-detectable DNA, to carry the “blanks” through all analyses. Extreme diligence is required in order to prevent contamination of samples, and inadvertent amplification of contaminating DNA. Contamination of a single cell or DNA fragment will distort downstream analyses.

1. Carefully read the manufacturer protocols and recommendations to ensure libraries are representative of the original sample.
2. Perform all work in a laminar flow hood dedicated to core samples.
3. Wipe all surfaces, equipment, and pipettes with 70% ethanol.
4. Put all reagents, supplies, and pipettes in the laminar flow hood, UV sterilize for 30 min. Rotate pipettes and supplies, UV sterilize for an additional 30 min.
5. Spray all surfaces with (first) DNAzap solution 1, then DNAzap solution 2, wipe with sterile paper towels, then with sterile water.
6. For optional MDA amplification, follow the manufacturer protocol with these recommendations:
 - (a) If performing MDA amplification, be sure to include a no-template control.
 - (b) Use the Qiagen “Whole genome amplification from genomic DNA using the REPLI-g Single Cell Kit with increased sample volumes.” This increases the volume of template to add from 2.5 to 15 µL.

- (c) An incubation time of less than the recommended 8 h may yield better results. It may be useful to set up reactions that incubate for 4, 6, and 8 h, and carry all through to sequencing.
 - (d) Quantify MDA-amplified DNA using the Qubit protocol in Subheading 3.3.2. Amplified DNA may need to be diluted 1:100 or 1:1000 to be in the quantification range.
7. For the Nextera XT DNA library preparation kit, follow the manufacturer protocol with these recommendations:
- (a) Do not overdry the AMPure XP magnetic beads or the DNA will not be recoverable.
 - (b) Check the quality and size of library preparation on a Bioanalyzer High Sensitivity DNA chip.
 - (c) Contact your sequencing facility to determine if you should proceed through the Normalize Libraries step. Many sequencing facilities prefer to perform this step.

4 Notes

1. The drilling process can take days or weeks. It is advantageous to take samples continuously during the drilling process, especially when new fluids and chemicals are added to the drilling muds.
2. Extraction of DNA from drilling muds using the method described for core samples can result in viscous, unusable DNA, due to coextraction of drilling mud components. We obtained the best results using the MoBio PowerMax Soil DNA isolation kit for drilling muds. Regardless of the DNA extraction method used, nucleic acids are often still contaminated with substances that inhibit PCR reactions or metagenomic sequencing due to the complex nature of the samples. We have found that a secondary cleanup of the nucleic acids is usually required. The Genomic DNA Clean & Concentrator-10 kit from Zymo Research gives good results, although many other commercial kits are available. Ethanol precipitation can have adverse effects, resulting in the concentration of nucleic acids as well as inhibitory substances.
3. Communication with the operator on-site is critical to obtain representative samples. Well in advance of sampling, meet with the operators to determine the amount of site access you'll have, and whom to contact during each phase of the operation. Provide a list of samples needed and instructions for sterile technique if the operators will take the samples.

4. Bentonite, a clay added to drilling muds in order to protect the formation from invasion of drilling fluids, has autofluorescent properties. Be sure to test visualization of microspheres with a sample of drill mud, before addition. Too much bentonite coating cores can interfere with visualization of fluorescent microspheres.
5. Be careful not to strike the Plattner's pestle when the sleeve is crooked on the base, or core particles are between the sleeve and the base. Test by placing the pestle in the sleeve and first rotating the sleeve on the base. Next rotate the pestle in the sleeve. If the sleeve and pestle rotate smoothly, the pestle is seated properly and can be struck with the sledge/drilling hammer. Failure to properly seat the pestle can result in sample loss and a bent sleeve, making the Plattner's pestle unusable.
6. The desired size of core material from the Plattner's mortar and pestle depends on the hardness and porosity of each sample type. For some sample types, it may be easier to first reduce the size of the core material in the Plattner's mortar and pestle, then transfer the material to a standard ceramic mortar and pestle, using a circular motion to reduce the size further before sieving. The mesh size of each sieve can be modified based on the sample type, resulting in larger or smaller particles as desired.
7. The amount of sample to extract from is dependent on the sample matrix and the amount of biomass. It may be necessary to extract from 50–100 g of sample to obtain enough DNA for sequencing purposes.
8. The final concentration of NaCl during the precipitation process is lower than recommended for most protocols, including the protocol here that we adapted from [16]. We optimized the concentration of NaCl for our particular samples, as shale contains high amounts of salts due to the fact that they are usually originally deposited as marine sediments. Tests of precipitations using recommended NaCl concentrations resulted in recovery of a large salt pellet with little to no quantifiable DNA recovered. This illustrates the importance of testing all protocols before performing them on actual samples.

Acknowledgments

Rebecca Daly, Kelly Wrighton and Michael Wilkins were partially supported by funding from the National Sciences Foundation Dimensions of Biodiversity (Award No. 1342701) and by the Marcellus Shale Energy and Environment Laboratory (MSEEL) funded by Department of Energy's National Energy Technology laboratory (DOE-NETL) grant DE#FE0024297.

References

1. McMahon S, Parnell J (2013) Weighing the deep continental biosphere. *FEMS Microbiol Ecol* 87(1):113–120
2. Whitman WB, Coleman DC, Wiebe WJ (1998) Prokaryotes: the unseen majority. *Proc Natl Acad Sci U S A* 95(12):6578–6583
3. Wilkins MJ, Daly R, Mouser PJ, Trexler R (2014) Trends and future challenges in sampling the deep terrestrial biosphere. *Front Microbiol* 5:481. <https://doi.org/10.3389/fmicb.2014.00481>
4. Kallmeyer J, Mangelsdorf K, Cragg B, Horsfield B (2006) Techniques for contamination assessment during drilling for terrestrial subsurface sediments. *Geomicrobiol J* 23 (3–4):227–239
5. Kieft TL, Onstott TC, Ahonen L, Aloisi V, Colwell FS, Engelen B, Fendrihan S, Gaidos E, Harms U, Head I, Kallmeyer J, Kiel Reese B, Lin LH, Long PE, Moser DP, Mills H, Sar P, Schulze-Makuch D, Stann-Lotter H, Wagner D, Wang PL, Westall F, Wilkins MJ (2015) Workshop to develop deep-life continental scientific drilling projects. *Sci Drill* 19:43–53
6. Tsesmetzis N, Maguire MJ, Head IM, Lomans BP (2016) Protocols for investigating the microbial communities of oil and gas reservoirs. In: McGenity TJ, Timmis KN, Nogales Fernandez B (eds) *Hydrocarbon and lipid microbiology protocols*. Humana Press, Totowa, NJ
7. Zhang G, Dong H, Xu Z, Zhao D, Zhang C (2005) Microbial diversity in ultra-high-pressure rocks and fluids from the Chinese continental scientific drilling project in China. *Appl Environ Microbiol* 71(6):3213–3227
8. Santelli CM, Banerjee N, Bach W, Edwards KJ (2010) Tapping the subsurface ocean crust biosphere: low biomass and drilling-related contamination calls for improved quality controls. *Geomicrobiol J* 27(2):158–116
9. Wandrey M, Morozova D, Zettlitzer M, Würdemann H, Group CS (2010) Assessing drilling mud and technical fluid contamination in rock core and brine samples intended for microbiological monitoring at the CO₂ storage site in Ketzin using fluorescent dye tracers. *Int J Greenhouse Gas Control* 4(6):972–980
10. Cardace D, Hoehler T, McCollom T, Schrenk M, Carnevale D, Kubo M, Twing K (2013) Establishment of the coast range ophiolite microbial observatory (CROMO): drilling objectives and preliminary outcomes. *Sci Drill* 16:45–55
11. Kieft TL, Phelps TJ, Fredrickson JK (2007) Drilling, coring, and sampling subsurface environments. In: Hurst C, Crawford R, Garland J, Lipson D, Mills A, Stetzenbach L (eds) *Manual of environmental microbiology*, Third Edition. ASM Press, Washington, DC, pp 799–817
12. Pfiffner SM, Onstott TC, Ruskeeniemi T, Talikka M, Bakermans C, McGown D, Chan E, Johnson A, Phelps TJ, Puil ML, Difurio SA, Pratt LM, Stotler R, Frappe S, Telling J, Lollar BS, Neill I, Zerbin B (2008) Challenges for coring deep permafrost on earth and Mars. *Astrobiology* 8(3):623–638
13. Yanagawa K, Nunoura T, McAllister S (2013) The first microbiological contamination assessment by deep-sea drilling and coring by the D/V Chikyu at the Iheya North hydrothermal field in the Mid-Okinawa Trough (IODP Expedition 331). *Front Microbiol* 4:327. <https://doi.org/10.3389/fmicb.2013.00327>
14. Griffiths RI, Whiteley AS, O'Donnell AG, Bailey MJ (2000) Rapid method for Coextraction of DNA and RNA from natural environments for analysis of ribosomal DNA- and rRNA-based microbial community composition. *Appl Environ Microbiol* 66(12):5488–5491
15. Hurt RA, Qiu X, Wu L, Roh Y, Palumbo AV, Tiedje JM, Zhou J (2001) Simultaneous recovery of RNA and DNA from soils and sediments. *Appl Environ Microbiol* 67(10):4495–4503
16. Lever MA, Torti A, Eickenbusch P, Michaud AB, Šantl-Temkiv T, Jørgensen BB (2015) A modular method for the extraction of DNA and RNA, and the separation of DNA pools from diverse environmental sample types. *Front Microbiol* 6(327):1281
17. Morono Y, Terada T, Hoshino T, Inagaki F (2014) A hot-alkaline DNA extraction method for deep subseafloor archaeal communities. *Appl Environ Microbiol* 80(6):1985–1994
18. Zhou J, Bruns MA, Tiedje JM (1996) DNA recovery from soils of diverse composition. *Appl Environ Microbiol* 62(2):316–322
19. Bowers RM, Clum A, Tice H, Lim J, Singh K, Ciobanu D, Ngan CY, Cheng J-F, Tringe SG, Woyke T (2015) Impact of library preparation protocols and template quantity on the metagenomic reconstruction of a mock microbial community. *BMC Genomics* 16:856. <https://doi.org/10.1186/s12864-015-2063-6>
20. Chafee M, Maignien L, Simmons SL (2014) The effects of variable sample biomass on comparative metagenomics. *Environ Microbiol* 17 (7):2239–2253

21. Salter SJ, Cox MJ, Turek EM, Calus ST, Cookson WO, Moffatt MF, Turner P, Parkhill J, Loman NJ, Walker AW (2014) Reagent and laboratory contamination can critically impact sequence-based microbiome analyses. *BMC Biol* 12(1):87
22. Woyke T, Sczyrba A, Lee J, Rinke C, Tighe D, Clingenpeel S, Malmstrom R, Stepanauskas R, Cheng J-F (2011) Decontamination of MDA reagents for single cell whole genome amplification. *PLoS One* 6(10):e26161
23. Barton HA, Taylor NM, Lubbers BR, Pemberton AC (2006) DNA extraction from low-biomass carbonate rock: an improved method with reduced contamination and the low-biomass contaminant database. *J Microbiol Methods* 66(1):21–31
24. Direito SOL, Marees A, Röling WFM (2012) Sensitive life detection strategies for low-biomass environments: optimizing extraction of nucleic acids adsorbing to terrestrial and Mars analogue minerals. *FEMS Microbiol Ecol* 81(1):111–123



Chapter 2

Freshwater Viromes: From Sampling to Evaluation

Catherine Putonti, Zoë Diener, and Siobhan C. Watkins

Abstract

There are a number of options available to researchers who wish to collect and analyze viral metagenomes (viromes) from environmental samples. Here we describe a laboratory procedure for generation of viromes from freshwater samples, specifically targeting dsDNA bacteriophages. We also discuss methods for bioinformatic analysis of the resulting data.

Key words Viromes, Metagenomics, Freshwater viruses, Bacteriophage, Viral bioinformatics

1 Introduction

Viruses evolve rapidly, demonstrate very high levels of genomic plasticity, and are an understudied group in comparison to prokaryotes. While metagenomics approaches have filled in the “missing branches” of the prokaryotic tree of life [1], sequencing of complex viral communities has to date only sampled a fraction of the diversity likely present (e.g., [2]). Many of the sequences present in a given viral metagenome (henceforth referred to as a virome) dataset will have no known homologs in the existing sequence databases. In some environments, the novel fraction can be upward of 90% [3–5]. In addition, phylogeny is poorly traceable owing to the lack of a universal gene marker equivalent to the 16S rRNA gene. However, some viruses, particularly bacterial viruses (bacteriophages), are well suited to metagenomics-based analysis, due to their accessibility and abundance in certain sample types.

Preparation of viruses for virome analysis may vary with sample, or by the question being asked, but for water, all observe a general workflow: filtration and concentration of the viral fraction, preparation and sequencing of DNA, and DNA sequence analysis. Concentration may be achieved physically, via tangential flow filtration (TFF) or ultracentrifugation, or via chemical flocculation (e.g., FeCl₃). However, purification of adequate quantities of DNA using these methods may still not be possible. In such cases,

amplification protocols can be used to bolster the DNA found in the sample—however, these are associated with inherent and consistent biases [6]. Current methodology for assessment of viromic datasets relies heavily on homology-based comparisons to sequenced and characterized viral genomes. Only a small fraction of the extant viral diversity—estimated less than 1%—has been characterized [3]. Of the sequenced viruses currently available, there is also a clear bias: viruses that infect humans or can be easily propagated within the lab are overrepresented [5]. Bioinformatic analysis is further complicated by the inherent challenges posed by genetic mobility [7].

This method outlines a basic protocol for preparation and analysis of a virome generated from freshwater samples, targeting dsDNA bacteriophages. While RNA and ssDNA phages are important members of viral communities [8], these viral species necessitate refined methods for sampling, genomic extraction, and amplification [5]. As such, the diversity of RNA and ssDNA phages is severely undersampled, although a few freshwater viral communities have been surveyed (e.g., [9–12]). Here, we also describe bioinformatic approaches for data assessment. This is particularly important for freshwaters and samples of a similar microbiological complexity, which are likely to contain a large fraction of previously undiscovered genetic diversity.

2 Materials

2.1 Sample Prep, Nucleic Acid Extraction, Sequencing Prep

1. Sterile container (e.g., bottle, carboy) to collect water sample.
See Note 1 with respect to volume for sampling.
2. 100 mL 0.45 µm MicroFunnel™ Disposable Filter Funnels (cat. No. 4804; Pall Laboratory, Port Washington, NY) [Qia- gen DNeasy® PowerWater® Kit recommendation] or 500 mL sterile bottle-top filters, 0.45 µm cellulose acetate membrane filters (cat. No. 430514, Corning Inc., Corning, NY).
3. 100 mL 0.22 µm MicroFunnel™ Disposable Filter Funnels (cat. No. 4803; Pall Laboratory, Port Washington, NY) [Qia- gen DNeasy® PowerWater® Kit recommendation] or 500 mL sterile bottle-top filters, 0.22 µm cellulose acetate membrane filters (cat. No. 430513, Corning Inc., Corning, NY).
4. Chloroform, biotechnology grade (cat. No. VWRV0757-500ML, VWR, Radnor, PA).
5. OPTIZYME™ DNase I (RNase-free) (cat. No. BP81071, Fisher Bioreagents, Pittsburg, PA).
6. RNase One™ (cat. No. M4261, Promega, Fitchburg, WI).
7. Labscale tangential flow filtration (TFF) system (cat. No. XX52LSS11, EMD Millipore Corp, Billerica, MA).

8. Pellicon XL Filter Module Durapore 0.1 µm 50 cm² polypropylene filter (cat. No. PXVVPPC50, EMD Millipore Corp, Billerica, MA).
9. Sterile 15 mL centrifuge tube.
10. Qiagen DNeasy® PowerWater® Kit (cat. No. 14900-50-NF, Qiagen, Germantown, MD).

2.2 Sequencing with Illumina MiSeq

1. Qubit™ 4 Fluorometer (cat. No. Q33226, Invitrogen, Carlsbad, CA).
2. Qubit™ dsDNA BR Assay Kit (cat. No. Q32850, Invitrogen, Carlsbad, CA).
3. Qubit Assay Tubes (cat. No. Q32856, Invitrogen, Carlsbad, CA).
4. Nextera XT DNA Library Preparation Kit (cat. No. FC-131-1024, Illumina, San Diego, CA).
5. MiSeq Reagent Kit v2 (500-cycles) (cat. No. MS-102-2003, Illumina, San Diego, CA).

2.3 Sequencing Data Quality Control and Assembly

1. Computer with a UNIX-based operating system and at least 8 GB of RAM.
2. Software: Sickle [13] version 1.33 or later.
3. Software: FastQC [14] version 0.11.7 or later.
4. Software: SPAdes [15] version 3.9.0 or later.

2.4 Virome Classification Tools

1. Computer with a UNIX-based operating system (if performing gene predictions, otherwise any operating system can be used) and at least 4 GB of RAM.
2. Blast+ Executable [16].
3. GeneMarkS [17] v. 4.30 or later.
4. Informativity Analysis Tool [18] v. 1.0.

3 Methods

3.1 Sample Prep, Nucleic Acid Extraction, Sequencing Prep

1. Collect viral particles from freshwater samples by sequential filtration and concentration. Pass water through sterile 0.45 µm and 0.22 µm filters to remove eukaryotic cells and debris and prokaryotic cells, respectively.
2. Filter and concentrate using a 0.10 µm filter attached to a TFF system, collecting the final filtrate in a sterile 15 mL centrifuge tube. Follow the filtration process outlined in the Labscale TFF protocol. This final concentration (3–10 mL in final volume) contains the viral fraction of the sample, from which DNA will be extracted (*see Note 2*).

3. Treat the concentrated sample with chloroform (1 μ L/mL), DNase (3–5 U/mL), and/or RNase (1 U/mL) (*see Note 3*).
4. Extract viral DNA using the Qiagen DNeasy® PowerWater® Kit per the manufacturer’s instructions, including the following modification step: add a heat treatment at 70 °C for at least 10 min before initial vortexing (*see Note 4*).

3.2 Sequencing with Illumina MiSeq

What follows is a general description of the methods we employed during the analysis of nine freshwater viromes [19]. More information can be found in the cited paper. While a greater throughput (number of reads) can be generated via the Illumina HiSeq platform and longer reads (at a higher error rate) can be generated via the PacBio or Nanopore platforms, the MiSeq platform provides a quality and read length that increases the length of contigs assembled. It is for these reasons that the MiSeq platform has been used for many recent freshwater viromes including, e.g., [20], and a single run has the capacity to provide sufficient coverage for genome reconstruction [21].

1. Quantify extracted DNA using the Qubit Fluorometer. A minimum of 1 ng of DNA is required for library construction.
2. Construct libraries using the Nextera XT kit for Illumina MiSeq preparation, according to the manufacturer’s instructions. This protocol is optimized for a 260/280 > 1.8. Furthermore, samples must not contain EDTA.
3. Perform sequencing on the Illumina MiSeq platform using the MiSeq Reagent Kit v2 (500-cycles) to generate paired-end reads of up to 250 \times 2. Per current Illumina kit specifications, this kit will produce 7.5–8.5 Gb of data.
4. Use the on-instrument MiSeq Reporter tool for de-multiplexing, remove adapter sequences, and produce the final FASTQ files.

3.3 Sequencing, Data Quality Control, and Assembly

The following procedure is presented for implementation on a machine with a UNIX-based operating system. Machines with greater than 8 GB of RAM are recommended (per requirements of **step 3** below).

1. Trim read sets to remove low quality bases. There are numerous tools currently available for trimming FASTQ files, e.g., Sickle [13], Trimmomatic [22], and Scythe [23] among others. Here we detail the use of one of these tools—Sickle [13]. Sickle is a windowed adaptive trimming program. The following command can be used to trim reads:

```
sickle pe -f file_R1.fastq -r file_R2.fastq -t sanger -o
file_R1_trimmed.fastq -p file_R1_trimmed.fastq -s
file_singles.fastq -l 150.
```

This command will output all reads that, when trimmed for quality, are at least 150 bases in length (“-l” flag). (*See Note 5* for further details regarding the length threshold.) Quality is evaluated using the quality scores listed for each read in the FASTQ format; reads produced using an Illumina quality scoring version 1.8 or later needs to specify the “sanger” quality format for the “-t” flag. Each trimmed read is then assessed relative to the length threshold. Read pairs in which both reads of the pair pass this threshold will be written to the R1 and R2 files. If one read of a pair does not meet the threshold but the other does, the read that meets the threshold will be written to the file specified by the “-s” flag.

2. Evaluate the quality of the trimmed reads using the Java tool FastQC [13]. This tool includes graphical and statistical reports regarding the quality of the reads. These reports can identify datasets that require further trimming as well as bad samples or sequencing runs. The FastQC documentation provides details for interpreting FastQC reports [13].
3. Perform contig assembly on the trimmed FASTQ files using SPAdes [15]. Here SPAdes will be used with the “meta” option (also referred to as metaSPAdes). metaSPAdes can integrate the paired-end reads as well as the singles produced by the Sickle trimming. While several other metagenomic assemblers are available (*see Note 6*), recent studies suggest that the SPAdes algorithm performs better than other solutions for the assembly of complete phage genomes from virome datasets [21, 24]. The command listed here is an adaptation of that recommended from this prior work [21]:

```
spades -o /assembly --pe1-1 file_R1_trimmed.fastq --pe1-2  
file_R2_trimmed.fastq -s file_singles.fastq --only-assembler -k  
33,55,77,99,127
```

Results will be written to the folder specified by the “-o” flag and several values of k (here specified as 33, 55, 77, 99, and 127) will be tested to identify the best assembly. The “-t” flag (although not included in the sample command above) can be specified to multi-thread the process; this should be specified depending upon the number of threads available on the user’s machine to expedite the assembly process. For further details regarding the SPAdes algorithm and the aforementioned flags, please refer to the SPAdes documentation included within the installation of the tool and [15].

3.4 Virome Classification Tools

3.4.1 Online Resources

Homology between characterized sequences available in existing databases and user-supplied virome data is frequently assessed via BLAST queries. Given the sheer volume of sequence data generated by metagenomic studies, BLAST queries via the NCBI web interface or one's local machine are often impractical. Several online resources are available for homology-based analyses of virome data, supported by server or cloud resources to expedite homology searches. Here we review a few of the tools frequently used by studies in the literature.

1. Metavir [25]: Metavir first compares uploaded data (either assembled contigs or reads) to the complete viral genomes of the RefSeq viral database via BLAST. Several publicly available datasets are also available via Metavir, facilitating comparisons between viromes. (At the time of publication, the Metavir server is not accepting new projects.)
2. VIROME [26]: The web server VIROME categorizes sequences based on BLAST (blastp) comparisons to the Uni-Ref 100 peptide database. Users upload viral assemblies and the pipeline predicts coding regions and performs BLAST comparisons.
3. MG-RAST [27]: The MG-RAST pipeline can accept reads or assemblies to perform functional and taxonomic classification. It predicts functionality via sBLAT homology queries against several different databases. The pipeline includes several features for data analysis, including functional predictions, abundance reports, and taxonomic predictions.
4. iVirus [28]: The iVirus Data Commons uses the CyVerse cyberinfrastructure for virome analysis. Upload raw reads. iVirus will then perform quality control and assembly. Viral sequences are then identified via VirSorter [29]. These viral contigs can then be examined via a variety of different methods, performing comparative genomics, gene ecology, and community ecology (as detailed in iVirus documentation).

A recent study by Tangherlini et al. [30] compared the performance of BLAST, Metavir, VIROME, MG-RAST, and a few other tools. For the dataset examined in this study—a viral assemblage from a sample of the deep-sea, tBLASTx and Metavir identified the most viral sequences, representative of greater diversity of viral families [30].

3.4.2 Resources Run Locally

As previously mentioned, running BLAST locally against all publicly available sequences, or even curated datasets, can quickly exceed available resources (time and memory). The Metagenome Analyzer (MEGAN) software [31], now in version 6, facilitates analysis and interactive tools for examining results in a single tool. Homology is determined using a BLAST-like algorithm called

DIAMOND [32]. Like MG-RAST, MEGAN performs functional analysis using several different databases. Two editions are available: the “Community Edition” which is free (GPL license) and the “Ultimate Edition” which is not free and includes additional features and user email support.

Alternatively, BLAST-based analyses can be targeted to a small subset of marker (“signature”) genes. These curated datasets can be used to identify viral sequences and predict taxonomic classification. The Prokaryotic Virus Orthologous Groups (known as pVOGs) are one such curated dataset [33]. The following outlines the steps for this process.

1. Download the pVOG dataset from <http://dmk-brain.ecn.uiowa.edu/pVOGs/>.
2. Create a BLAST database of pVOG sequences. pVOG sequences are amino acid sequences, thus create a protein database:

```
makeblastdb -in pVOG.fasta -title pVOG -out pVOG -dbtype
prot
```

This command will output a protein BLAST database named pVOG for the pVOG.fasta file.

3. If contigs are going to be compared to the signature gene database, proceed to the next step. Otherwise, perform open reading frame (ORF) prediction using GeneMarkS [17] (*see Note 7*). The predicted ORFs can be output as either nucleotide sequences or amino acid sequences.
4. Compare predicted ORFs against the pVOG database. Predicted nucleotide sequences can be queried via the blastx algorithm. Predicted amino acid sequences can be queried via the blastp algorithm. The following sample command will perform a blastp query:

```
blastp -query predicted_ORFs.file -db pVOG -outfmt 10 -out
output_file -max_target_seqs 1
```

5. Evaluate BLAST output using, e.g., Python, R, or another programming language or statistical tool.

Taxonomic classification of viromes can also be assessed using an alternative homology-based approach. Recently, we developed a tool for evaluating BLAST-based results; BLAST queries are evaluated with respect to its information content [18]. The following outlines the steps for this process.

1. Retrieve a local collection of the genome for the viral taxon of interest, a near-neighbor genome, and the sequences of evolutionarily distant viruses (outgroup). *See Note 8* for further details regarding these sequences.

2. The following command will execute the method:

```
informativity taxon_of_interest_nucl.fasta  
taxon_of_interest_aa.fasta near_neighbor_aa.fasta  
outgroup_aa.fasta virome_nucl.fasta
```

The user will be prompted for a name for the run; output will be written to files by this name. Briefly, this command will first compare the taxon of interest to the near-neighbor and out-group sequences, quantifying the prevalence of each individual gene within the taxa of interest in other viral species. For the provided assembly, ORFs will be predicted for the contig sequences and examined for the taxon of interest. Scores are returned for each gene within the taxa of interest providing a quantifiable means of ascertaining if the genome/species is present within the virome or if individual genes are present (a likely indicator of the presence of another viral taxa). Further details regarding this tool and methods for interpretation can be found in [18].

4 Notes

1. While the volume to be sampled may be dependent upon availability, samples collected should include a sufficient representation of the viral fraction. Metagenomic studies of freshwater viruses have thus varied in the size of the samples processed. Antarctic lake samples have included 100 and 350 L [10]; similar large quantities (150 L) were collected from freshwater reservoirs in Taiwan [34]. This is in contrast to sample sizes collected from large freshwater lakes such as Lough Neagh in which 5 L was collected [20] and Lake Michigan in which 4 L was collected [19].
2. Successful use of tangential flow filtration for processing freshwater samples is heavily reliant on keeping the equipment and filters as clean as possible. Therefore between each sample the TFF system must be flushed thoroughly with the following: 0.01% NaOCl (250 mL), 0.1% Tergazyme (500 mL), phosphate buffered saline (250 mL) (as recommended by the manufacturer).
3. Concentrated filtrate can be treated with chloroform which will rupture cells that may have passed through filters. It can, however, also degrade chloroform-sensitive viruses. DNase and RNase treatment will remove nonviral, e.g., eukaryotic and prokaryotic cell debris, DNA and RNA.
4. A common issue encountered is small quantities of extracted DNA—each sample must be visualized on an agarose gel and subject to PCR with primers for the prokaryotic 16S rRNA

gene [35] to check for bacterial contamination, alongside appropriate controls. Extractions may be subject to multiple displacement amplification (MDA) reactions in order to amplify quantities of DNA too small to be seen on an agarose gel—however, this introduces known bias to the final dataset [6].

5. The length threshold should be tested to best reflect the length of the reads produced. Should another reagent kit be used, the number of cycles will differ. As such, the “-l” flag should be set relative to the average length of the paired-end read produced. In addition to using a kit with fewer cycles, reads produced can be shorter if DNA was fragmented to smaller sizes. Here we have assumed fragmentation ~800–1000 bp (per Illumina kit protocol).
6. Commonly used alternative resources for assembly include MetaVelvet [36], Ray Meta [37], and MEGAHIT [38].
7. Commonly used alternative resources for ORF prediction include GetORF [39] and prodigal [40].
8. The informativity method requires:
 - (a) The nucleotide sequences for coding regions of the genome of the taxon of interest in multi-FASTA format.
 - (b) The amino acid sequences for coding regions of the genome of the taxon of interest in multi-FASTA format.
 - (c) The amino acid sequences for coding regions of a genome for a near-neighbor to the taxon of interest in multi-FASTA format.
 - (d) The amino acid sequences for coding regions of the outgroup.

The near-neighbor genome here refers to the genome of a related species. The taxonomic distance between the taxon of interest and near-neighbor will determine the granularity of which the taxonomic classification can be made. For instance, to predict the presence of the phage species *Pseudomonas* phage PB1 (genus: Pbunavirus), using the near-neighbor *Burkholderia* phage BcepF1 (genus: Pbunavirus) will permit one to classify virome contigs to the species level. In contrast, if a near-neighbor is selected from another genus, then the prediction can be classified only to the genus level. The outgroup will contain the coding sequences for all viruses not belonging to the species or genus (depending upon the level selected). Sequences can be in a private collection and/or can be retrieved through FTP from GenBank. Further details regarding the process implemented by this approach can be found in [18].

Acknowledgments

This work was supported by the NSF (1149387) (CP). The authors would like to thank all who assisted in previous studies mentioned here, including Katherine Bruder, Alexandria Cooper, Thomas Hatzopoulos, Alex Kula, Kema Malki, Zachary Romer, Jason Shapiro, and Emily Sible.

References

1. Hug L, Baker B, Anantharaman K et al (2016) A new view of the tree of life. *Nat Microbiol* 1:16048
2. Halary S, Temmam S, Raoult D, Desnues C (2016) Viral metagenomics: are we missing the giants? *Curr Opin Microbiol* 31:34–43
3. Mokili JL, Rohwer F, Dutilh BE (2012) Metagenomics and future perspectives in virus discovery. *Curr Opin Virol* 2:63–77
4. Brum JR, Sullivan MB (2015) Rising to the challenge: accelerated pace of discovery transforms marine virology. *Nat Rev Microbiol* 13:147–159
5. Bruder K, Malki K, Cooper A et al (2016) Freshwater metaviromics and bacteriophages: a current assessment of the state of the art in relation to bioinformatic challenges. *Evol Bioinforma* 12(S1):25–33
6. Kim K-H, Bae J-W (2011) Amplification methods bias metagenomic libraries of uncultured single-stranded and double-stranded DNA viruses. *Appl Environ Microbiol* 77:7663–7668
7. Hendrix RW, Smith MCM, Burns RN, Ford ME, Hatfull GF (1999) Evolutionary relationships among diverse bacteriophages and prophages: all the world's a phage. *Proc Natl Acad Sci U S A* 96:2192–2197
8. Székely AJ, Breitbart M (2016) Single-stranded DNA phages: from early molecular biology tools to recent revolutions in environmental microbiology. *FEMS Microbiol Lett* 363:fnw027
9. Djikeng A, Kuzmickas R, Anderson NG, Spiro DJ (2009) Metagenomic analysis of RNA viruses in a fresh water Lake. *PLoS One* 4: e7264
10. López-Bueno A, Tamames J, Velazquez D, Moya A, Quesada A, Alcamí A (2009) High diversity of the viral community from an antarctic lake. *Science* 326:858–861
11. Adriaenssens EM, van Zyl LJ, Cowan DA, Trindade MI (2016) Metaviromics of Namib Desert salt pans: a novel lineage of Haloarchaeal Salterproviruses and a rich source of ssDNA viruses. *Viruses* 8:14
12. Zeigler AL, McCrow JP, Ininbergs K et al (2017) The Baltic Sea virome: diversity and transcriptional activity of DNA and RNA viruses. *mSystems* 2:e00125–16
13. Joshi NA, Fass JN (2011) Sickle: a sliding-window, adaptive, quality-based trimming tool for FastQ files (Version 1.33) [Software]. <https://github.com/najoshi/sickle>
14. Andrews S (2010) FastQC: a quality control tool for high throughput sequence data. <http://www.bioinformatics.babraham.ac.uk/projects/fastqc>
15. Nurk S, Meleshko D, Korobeynikov A, Pevzner PA (2017) metaSPAdes: a new versatile metagenomic assembler. *Genome Res* 27:825–834
16. Carmacho C, Coulouris G, Avagyan V et al (2009) BLAST+: architecture and applications. *BMC Bioinformatics* 10:421
17. Borodovsky M, Lomsadze A (2014) Gene identification in prokaryotic genomes, phages, metagenomes, and EST sequences with GeneMarkS suite. *Curr Protoc Microbiol* 32: E:1E.7:1E.7.1–1E.7.17
18. Watkins SC, Putonti C (2017) The use of informativity in the development of robust viromics-based examinations. *PeerJ* 5:e3281
19. Watkins SC, Kuehnle N, Ruggeri CA et al (2015) Assessment of a metaviromic dataset generated from nearshore Lake Michigan. *Mar Freshw Res* 67:1700–1708
20. Skvortsov T, Leeuwe C, Quinn J et al (2016) Metagenomic characterisation of the viral community of lough Neagh, the largest freshwater lake in Ireland. *PLoS One* 11:e0150361
21. Rihtman B, Meaden S, Clokie M, Koskella B, Millard A (2016) Assessing Illumina technology for the high-throughput sequencing of bacteriophage genomes. *PeerJ* 4:e2055
22. Bolger AM, Lohse M, Usadel B (2014) Trimomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics* 30:2114–2120

23. Buffalo V (2014) Scythe—A Bayesian adapter trimmer (version 0.994 BETA). <https://github.com/ucdavis-bioinformatics/scythe>
24. Roux S, Emerson JB, Eloe-Fadrosh EA, Sullivan MB (2017) Benchmarking viromics: an *in silico* evaluation of metagenome-enabled estimates of viral community composition and diversity. PeerJ 5:e3817
25. Roux S, Tournayre J, Mahul A et al (2014) Metavir 2: new tools for viral metagenome comparison and assembled virome analysis. BMC Bioinformatics 15:76
26. Wommack K, Bhavsar J, Polson S et al (2012) VIROME: a standard operating procedure for analysis of viral metagenome sequences. Stand Genomic Sci 6:427–439
27. Wilke A, Bischof J, Gerlach W et al (2016) The MG-RAST metagenomics database and portal in 2015. Nucleic Acids Res 44:590–594
28. Bolduc B, Youens-Clark K, Roux S, Hurwitz BL, Sullivan MB (2017) iVirus: facilitating new insights in viral ecology with software and community data sets imbedded in a cyberinfrastructure. ISME J 11:7–14
29. Roux S, Enault F, Hurwitz BL, Sullivan MB (2015) VirSorter: mining viral signal from microbial genomic data. PeerJ 3:e985
30. Tangherlini M, Dell’Anno A, Allen LZ, Riccioni G, Corinaldesi C (2016) Assessing viral taxonomic composition in benthic marine ecosystems: reliability and efficiency of different bioinformatic tools for viral metagenomic analyses. Sci Rep 6:28428
31. Huson DH, Weber N (2013) Microbial community analysis using MEGAN. Methods Enzymol 531:465–485
32. Buchfink B, Xie C, Huson DH (2015) Fast and sensitive protein alignment using DIAMOND. Nat Methods 12:59–60
33. Grazziotin AL, Koonin EV, Kristensen DM (2017) Prokaryotic Virus Orthologous Groups (pVOGs): a resource for comparative genomics and protein family annotation. Nucleic Acids Res 45:D491–D498
34. Tseng C-H, Chiang P-W, Shiah F-K et al (2013) Microbial and viral metagenomes of a subtropical freshwater reservoir subject to climatic disturbances. ISME J 7:2374–2386
35. Marchesi J, Weightman A, Cragg B (2001) Methanogen and bacterial diversity and distribution in deep gas hydrate sediments from the Cascadia Margin as revealed by 16S rRNA molecular analysis. FEMS Microbiol Ecol 34:221–228
36. Namiki T, Hachiya T, Tanaka H et al (2012) MetaVelvet: an extension of Velvet assembler to de novo metagenome assembly from short sequence reads. Nucleic Acids Res 40:e155–e155
37. Boisvert S, Raymond F, Godzardis É, Laviolette F, Corbeil J (2012) Ray Meta: scalable *de novo* metagenome assembly and profiling. Genome Biol 13:R122
38. Li D, Luo R, Liu CM et al (2016) MEGAHIT v1.0: a fast and scalable metagenome assembler driven by advanced methodologies and community practices. Methods 102:3–11
39. Rice P, Longden I, Bleasby A (2000) EMBOSS: the European molecular biology open software suite. Trends Genet 16:276–277
40. Hyatt D, Chen GL, Locascio PF, Land ML, Larimer FW, Hauser LJ (2010) Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC Bioinformatics 11:119



Chapter 3

Characterization of Eukaryotic Microbiome Using 18S Amplicon Sequencing

Ana Popovic and John Parkinson

Abstract

With the advent of low-cost, high-throughput sequencing, taxonomic profiling of complex microbial communities through 16S rRNA marker gene surveys has received widespread interest, uncovering a wealth of information concerning the bacterial composition of microbial communities, as well as their association with health and disease. On the other hand, little is known concerning the eukaryotic components of microbiomes. Such components include single-celled parasites and multicellular worms that are known to adversely impact the health of millions of people worldwide. Current molecular methods to detect eukaryotic microbes rely on the use of directed PCR analyses that are limited by their inability to inform beyond the taxon targeted. With increasing interest to develop equivalent marker-based surveys as used for bacteria, this chapter presents a stepwise protocol to characterize the diversity of eukaryotic microbes in a sample, using amplicon sequencing of hypervariable regions in the eukaryotic 18S rRNA gene.

Key words Microbiome, Eukaryotic microbes, Amplicon sequencing, Taxonomic assignment, 18S rRNA, Illumina MiSeq

1 Introduction

The advent of inexpensive, massively parallel sequencing technologies has enabled characterization of complex microbial communities, through direct sequencing of extracted genomic material. As a majority of environmental microbes are not amenable to laboratory culturing, these methods have provided direct access to largely unexplored biota. Development of targeted gene sequencing, through PCR amplification of a single taxonomically informative “marker gene” from organisms of interest, also known as amplicon sequencing, has further reduced cost and increased depth of taxonomic information. In bacteria, this approach, targeting the 16S rRNA gene, has garnered much data on the bacterial diversity in the environment and the human microbiome. As a result, links have been uncovered between the taxonomic

composition of gut bacteria and function of the immune system, obesity, drug metabolism, nutrition, and even behavior [1–5].

A similar approach may be employed to characterize the eukaryotic microbial diversity. In the context of the human microbiome, such studies will elucidate the little-explored diversity and role of eukaryotic microbes in human health and disease. While single-celled eukaryotes such as *Giardia*, *Cryptosporidium*, and *Entamoeba* are known to infect hundreds of millions of people worldwide [6, 7], the majority of infected individuals remain asymptomatic. In addition, recent studies have identified *Blastocystis* spp., *Dientamoeba fragilis*, and over 50 genera of fungi in healthy individuals, suggesting that eukaryotic microbes may play a larger role than previously appreciated in host health [8–11]. Here we present a protocol that exploits the properties of the 18S rRNA gene to survey the eukaryotic component of the microbiome.

As with the 16S rRNA gene used to survey bacterial taxa, the 18S rRNA gene is a suitable marker for eukaryotes based on the presence of conserved genetic regions, amenable to the design of universal DNA primers, separated by relatively short variable stretches, which serve as barcodes to identify specific taxa [12–17]. Furthermore, the 18S gene has a large curated set of reference sequences available through the SILVA non-redundant ribosomal RNA database (67,380 genes, version 128) [18], essential for accurate taxonomic classification. Current recommendations include the use of 18S variable region 4 for broad taxonomic resolution, proposed by the Consortium for the Barcode of Life (CBOL) Protist Working Group [16], and 18S variable region 9 proposed by the Earth Microbiome Project (EMP) [19]. However, adoption of the 18S rRNA gene as a suitable marker faces a number of experimental and computational challenges. For example, differences in cell wall or membrane composition can bias extraction procedures, skewing recovery of certain taxa [20]. Conserved DNA regions used for primer binding may not be 100% identical across taxa resulting in primer mismatches and subsequent biases in PCR reactions. Consequently, new primers are adopted as studies continue to reveal new eukaryotic taxa that are not adequately captured with existing 18S primers [13, 17, 21, 22]. Moreover, the variable region targeted can result in additional taxon bias in PCR products arising from differences in amplicon size and/or the presence of introns. For example, an amplicon containing variable regions 4 and 5 is approximately 400 nucleotides long in *Trichomonas vaginalis* and over 1000 nucleotides long in *Trypanosoma cruzi*. Such differences in length can also impact downstream computational analyses. For example, sequencing with Illumina MiSeq v3 chemistry, which generates 300 nucleotide paired-end sequences, would result in nonoverlapping reads for taxa with such extended variable regions, which

would be discarded during the read “merging” step employed by many standard workflows. A further challenge is variation in copy number of rRNA genes across taxa that, as for bacteria, can make it challenging to quantify taxon abundance within samples based on sequence counts alone. As an example, *Trichomonas vaginalis* is predicted to encode ~250 copies of the 18S rRNA gene [23], while *Plasmodium falciparum* encodes only 5–8 copies [24]. As a result, marker-based studies typically examine relative differences in abundance between samples.

In terms of taxonomic classification, lineage-specific sequence variability can significantly impact the level to which different taxa may be resolved. While a specific variable region may be informative enough to differentiate genera or species in one lineage, it may be relatively more conserved in another, allowing only high-level taxonomic assignment (e.g., order or family) [16]. For example, the 119 18S rRNA genes associated with *Blastocystis hominis* in the SILVA database (version 128) exhibit 78–100% pairwise sequence identity in variable regions 4 and 5; conversely the same region in *Entamoeba histolytica* shares 97% sequence similarity with *Entamoeba dispar*. Thus, additional marker gene regions may be necessary to resolve genera or species in particular taxonomic groups. Other genetic biomarkers which have been proposed to resolve particular lineages of eukaryotes include the large ribosomal subunit and internal transcribed spacers (ITS), mitochondrial cytochrome oxidase I, nuclear proteins actin and alpha- and beta-tubulin, myosins, Hsp70 and chloroplast rbcL [16, 25–29]. However to date, systematic comparisons to assess performance across multiple eukaryotic lineages are limited, and reference databases are sparse or lacking.

As a further challenge to diversity analyses, the set of reference 18S rRNA sequences is limited and unevenly distributed. The SILVA database (version 128) consists of 67,380 18S rRNA sequences, compared to 537,351 16S rRNA sequences. The majority of these derive from a limited set of taxa (e.g., 10,789 Alveolates and 22,723 Animalia), with only 676 representing Amoebozoa. Finally, we note that a number of bioinformatics tools and platforms have been developed and continue to evolve, as we learn how to best manipulate the data. Clustering algorithms and chimera detection each have their biases and exhibit some disagreements in predictions. These are challenges that one must be aware of during analysis, and which continue to be addressed.

Here we describe a workflow to characterize the eukaryotic component of a microbiome (Fig. 1), beginning with generation of 18S variable region 4 amplicons from extracted genomic DNA, followed by data processing of sequences generated by the Illumina MiSeq v3 platform (2× 300 bases), taxonomic profiling, and diversity analyses. We chose the variable region 4 as we have previously

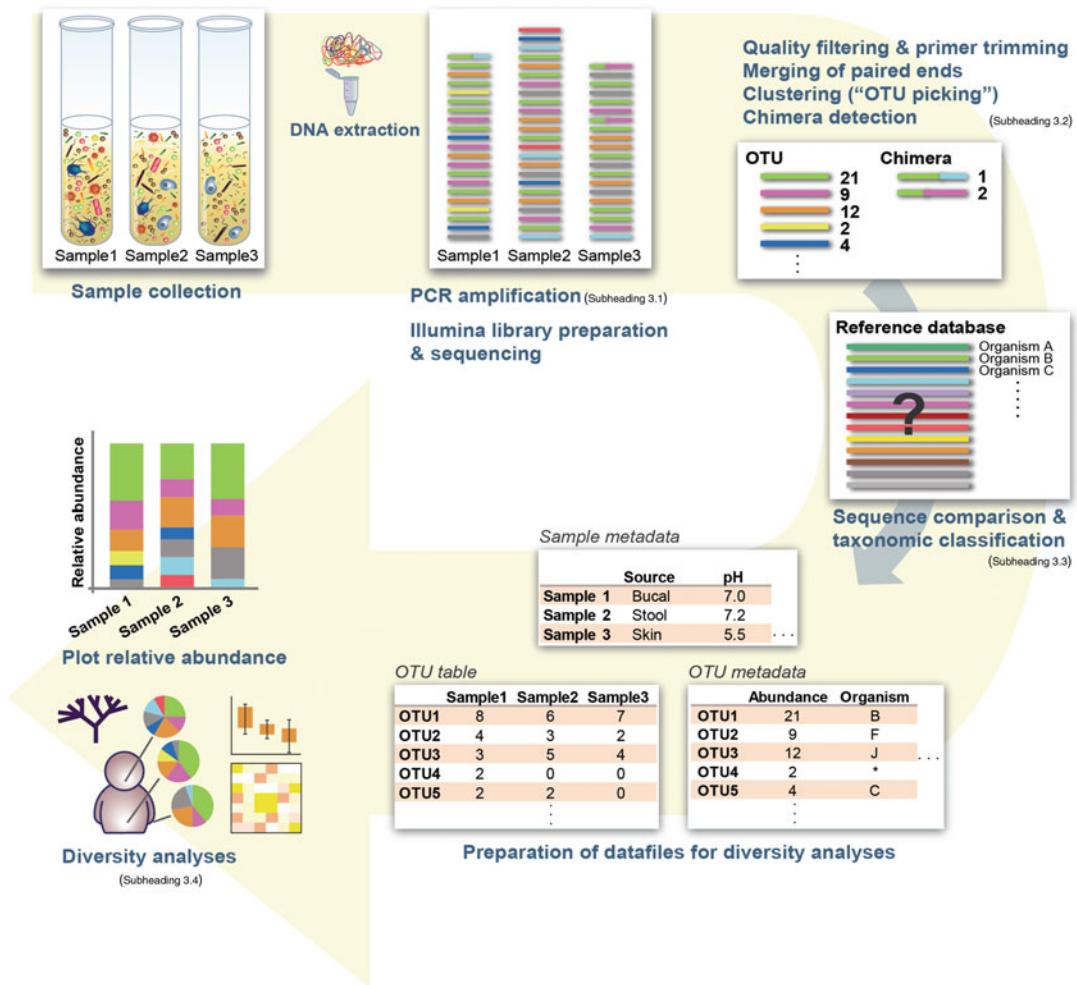


Fig. 1 Taxonomic classification of eukaryotic microbes using 18S amplicon sequencing. The workflow includes generation of 18S amplicons (Subheading 3.1), Illumina MiSeq sequence processing (Subheading 3.2), taxonomic classification (Subheading 3.3), and subsequent data visualization and diversity analyses (Subheading 3.4)

found this region to offer reasonable taxonomic resolution, consistent with previous studies [13]. The same pipeline may be applied to other marker gene regions, keeping in mind suitability of sequences for taxonomic classification, presence of conserved regions for design of universal primers, amplicon length, and availability of reference sequences. While we employ a particular set of open-source tools and scripts for this workflow, there are a myriad of bioinformatic tools which may be used, some of which are mentioned in the accompanying notes.

2 Materials

2.1 Amplicon Generation

High-fidelity DNA polymerase (*see Note 1*).

iProof High-Fidelity polymerase (BioRad cat. # 1725301)

or KAPA2G Robust HotStart polymerase (Kapa Biosystems cat. # KK5522; recommended by Illumina).

Thermocycler.

Universal DNA primers to generate 18S V4 amplicons [30].

515f 5'-GTGCCAGCMGCCGCGTAA-3'

1119r 5'-GGTGCCTTCCGTCA-3'

PCR reaction components, prepared as below:

Template DNA (test several volumes or ng—e.g., 2 ng)

Buffer 1×

dNTP 0.2 mM

Forward primer 0.4 mM

Reverse primer 0.4 mM

DMSO 5%

Polymerase 2 units per reaction BioRad iProof

ddH₂O (PCR-grade) Up to 25 μL total volume

Illumina adapter sequences and multiplexing barcode sequences may be ligated by the sequencing facility. Alternatively, validated primers that encode Illumina adapter sequences are provided by the EMP (www.earthmicrobiome.org/protocols-and-standards/18s/), albeit for the short 18S V9 region, which has lower power for taxonomic resolution.

2.2 Computational Analysis

Suggested software is available for download from the indicated links and requires access to sufficient computing power.

2.2.1 Sequence Data Processing

- (a) We perform sequence processing in a Linux environment (Ubuntu 16.04 ubuntu.com) through Oracle VM VirtualBox 5.0.4 (virtualbox.org), installed in Windows.
- (b) FastQC 0.11.5 (bioinformatics.babraham.ac.uk/projects/fastqc/) [31].
- (c) Trimmomatic-0.36 (usadellab.org/cms/?page=trimmomatic) [32].
- (d) Cutadapt 1.12 (cutadapt.readthedocs.io/en/stable/installation.html) [33].

(e) VSEARCH 1.11.1 (github.com/torognes/vsearch) [34], a free open-source tool modeled after USEARCH (drive5.com/usearch) [35].

(f) SILVA v128 rRNA reference databases, modified for QIIME (arb-SILVA-de/no_cache/download/archive/qiime).

Once programs are installed, the paths may be added to the end of the .bashrc file located in the home directory (“`export PATH=$PATH:/path/to/program`”) in order to make them accessible from any directory.

(g) R ([r-project.org](https://www.r-project.org)) and RStudio (rstudio.com). R may already be available in Ubuntu (type “`R --version`” in a terminal) by command line. RStudio provides a more user-friendly environment.

2.2.2 Analysis

(h) R packages used for taxonomic analysis and visualization are listed below:

- Phyloseq [36], visualization and analysis of microbiome data.
- Vegan 2.4-2 [37], also used for ecological data ordination, includes rarefaction function.
- Stringr 1.2.0 [38], manipulation of strings and data tables.
- Reshape2 [39], data reconfiguration.
- ggplot2 [40], data visualization.
- *Optional:* ape [41], visualization and manipulation of phylogenetic trees.

These packages can be installed from within the R environment.

3 Methods

The following method was applied to generate amplicon sequences from DNA purified from human stool samples, and to subsequently identify eukaryotic microbes in the microbiome. We will not discuss DNA purification methods as they may vary according to organisms of interest and sampling site, but generally we recommend mechanical disruption of samples with beads, followed by extraction with the MoBio PowerSoil DNA Isolation Kit (now Qiagen DNeasy PowerSoil Kit).

3.1 Amplicon Generation

Eukaryotic 18S rRNA variable region 4 amplicons are generated as below. There is no consensus on the best approach to standardize template input between different samples. The protocols recommended by the EMP, by Amaral-Zettler et al. [19], suggest using 2 µL template volume. We recommend carrying out test reactions

on several samples to determine the appropriate template volume or mass, as the proportion of eukaryotic DNA may differ depending on the sample source. For genomic DNA extracted from stool samples, we suggest a minimum of 1–5 ng template DNA, as the majority of microbes and therefore genomes are expected to be bacteria. Annealing temperatures and extension times may be modified, depending on the length of amplicon region and specific primers used. Suggested PCR conditions are indicated below:

1. Prepare 25 µL PCR reactions on ice (*see Subheading 2.1* for reaction mixture components and concentrations). All PCR reactions, including the “no template” control, should be carried out in triplicate.
2. Run PCR program as follows:

Denaturation	95°C	3 min
Melting	95°C	45 s
Annealing	57°C	45 s
Extension	72°C	50 s (<i>see Note 2</i>)
<i>30 cycles</i>		
Final extension	72°C	10 min
Incubation	4°C	∞

3. Pool the triplicate PCR reactions and submit for (or proceed to) Illumina library generation. A protocol for 16S sequencing library preparation is available on the Illumina website (https://support.illumina.com/downloads/16s_metagenomic_sequencing_library_preparation.html), which may be applied to other amplicons, including 18S. The protocol involves PCR cleanup, attachment of adapter sequences through an additional PCR step (alternatively adapter sequences may be ligated), library quantification and normalization, DNA denaturation and addition of >5% PhiX DNA library. Sequencing should be carried out using the Illumina paired-end MiSeq v3 (2 × 300 nt) platform.

3.2 Sequence Data Processing

This workflow makes use of FastQC for sequence quality assessment, and Trimmomatic, VSEARCH, and cutadapt for sequence processing (*see Note 3*). It also relies on a number of Linux-based `awk`, `sed`, and `bash` commands joined using UNIX pipes (‘|’) to create data tables. Sequence datasets should be downloaded as FASTQ files, and pre-processed to remove adapter and primer sequences, low-quality base calls, and to predict and remove chimeric sequences prior to taxonomic classification (*see Note 4*). In many of the steps in the following sections, many commands wrap over multiple lines (e.g., each of the two `awk` commands in **step 3** and the `vsearch` commands in **steps 7–9, 12–14**). Care must be

taken when copying such commands to include the full set of arguments. The # symbol precedes comments and explanations within the code.

1. Move all sequence FASTQ files to a new project folder.
2. Create a list of samples, matching sequence file names. It may be useful to rename the files to something simple or meaningful: SampleA_R1.fastq, SampleA_R2.fastq, SampleB_R1.fastq, etc.

```
ls *.fastq | sed 's/_R.*fastq//g' | uniq > samplelist.txt
more samplelist.txt
SampleA
SampleB
SampleC
```

3. (Optional) Write a bash script to automatically generate and run commands (**steps 4**, and **6–9**) with each sequence file. The example below uses a for loop to run commands over all files.

```
#!/bin/bash
#assign total number of samples to a variable
NUMFILES=$(wc -l < samplelist.txt)
for i in $(seq 1 $NUMFILES); do
    #assign sample name to variable
    FILE=$(sed -n ${i}p samplelist.txt)
    #insert data processing command. For example (step 4):
    awk -v VAR="@M/@M/{print "@VAR" ++i; next}
{print}' ${FILE}_R1.fastq> ${FILE}_R1mod.fastq
    awk -v VAR="@M/@M/{print "@VAR" ++i; next}
{print}' ${FILE}_R2.fastq> ${FILE}_R2mod.fastq
done
```

4. Rename individual sequences (reads) in each file to reflect the sample name and read number. This will both reduce the file size and allow for downstream scripts to identify the read source. Forward and reverse sequencing files (often labeled R1 and R2) must be processed in a consistent manner for downstream tools to be able to merge the reads.

```
awk '/@M/ {print "@SampleA: " ++i; next} {print}'
SampleA_R1.fastq > SampleA_R1mod.fastq
#view the first line of the original and modified FASTQ files
head -n 1 SampleA_R1.fastq
@M00331:25:00000000-ANL48:1:1101:15398:1343 1:N:0:125
head -n 1 SampleA_R1mod.fastq
@SampleA:1
```

In order to automatically change read names in all sample files, a bash script may be written (*see example in step 3*).

5. Generate sequence quality reports using FastQC, and output into a new directory. FastQC will generate HTML files, with graphical summaries of quality statistics for each sample (*see Note 5*).

```
mkdir fastqcOUT
fastqc *mod.fastq -outdir fastqcOUT
```

6. Trim primer sequences and low-quality bases using Trimmomatic (*see Note 6*). The primer sequences may be provided in a FASTA file (“primers.fasta” in the command below). If Illumina adapters have not been removed by the sequencing facility, they may be removed in this step as well. The following allows two mismatches to primer sequences (*see Note 7*). Note this command wraps over multiple lines.

```
java -jar /path/to/trimmomatic/trimmomatic-0.36.jar PE -
phred33 -trimlog SampleA_trim.log SampleA_R1mod.fastq
SampleA_R2mod.fastq SampleA_R1_trimmedP.fastq
SampleA_R1_trimmedUnP.fastq SampleA_R2_trimmedP.fastq
SampleA_R2_trimmedUnP.fastq
ILLUMINACLIP:primers.fasta:2:30:10 LEADING:3 TRAILING:3
SLIDINGWINDOW:4:15 MINLEN:30
```

7. Merge paired ends into a single read, using VSEARCH (*see Note 8*). A low proportion of merged reads may result, as the amplicon length range is 400–600 nt in a majority of organisms, up to 1000 nt in select taxa. In this case, R1 and R2 read files should be trimmed individually to a length of 200–250 nt in **step 6**, and **step 7** should be omitted (*see Note 7*). The remaining commands may also be applied to the trimmed R1 and R2 files separately.

```
vsearch --fastq_mergepairs SampleA_R1trimmedP.fastq --
reverse SampleA_R2trimmedP.fastq --fastq_ascii 33
--fastqout SampleA_merged.fastq --
fastq_allowmergestagger --threads 4 2>
SampleA_merged.log
```

8. Remove sequences with ambiguous bases and convert the FASTQ file to FASTA.

```
vsearch --fastq_filter SampleA_merged.fastq --fastq_maxns
0 --fastaout SampleA_merged.fasta 2>
SampleA_noambiguities.log
```

9. DerePLICATE sequences. Removing identical sequences will reduce computer processing times, while information on total numbers of reads will be retained with the --sizeout option.

```
vsearch --derep_fulllength SampleA_merged.fasta --output
SampleA_derep.fasta --log SampleA_derep.log --sizeout -
-fasta_width 0
```

10. Concatenate all dereplicated FASTA files into one file (*see Note 9*).

```
cat Sample*_derep.fasta > Samples.fasta
```

11. (*Optional*) Remove contaminating sequences from PhiX using bowtie2 (*see Note 10*).

```
#index PhiX genome.
bowtie2-build genome.fasta DATABASENAME
#remove contaminating sequences.
bowtie2 -f -x /path/to/reference/DATABASENAME -U
Samples.fasta --un Samples_nophix.fasta 2>
Samples_phix.log
```

12. Cluster highly similar sequences, above a user-defined threshold, into *de novo* Operational Taxonomic Units (OTUs), also known as “OTU picking,” choosing a representative sequence from each OTU for further analysis (*see Note 11*). The --sizein and --sizeout flags ensure that information on read abundance, required for chimera detection (**step 14**), is retained.

```
vsearch --cluster_fast Samples.fasta --id 0.97 --threads
4 --centroids otus.fasta --uc otus.uc --sizein --sizeout
--fasta_width 0 --relabel OTU --log otus.log
```

13. Map reads to OTUs (*see Note 11*). The output file in this step will provide the information on read abundance in samples that will be used to generate the OTU table.

```
vsearch --usearch_global Samples.fasta --id 0.97 --db
otus.fasta --uc otu_readmembership.uc --log
otu_readmembership.log
```

14. Predict chimeric sequences. This step is based on the uchime_denovo chimera-detection algorithm by Edgar [35], which segments sequences and identifies those where the 5' and 3' regions are highly similar or identical to two different, higher abundance clusters.

```
vsearch --uchime_denovo otus.fasta --nonchimeras
otus_nochimera.fasta --chimeras otus_chimera.fasta --
uchimeout otus_chimera.uc --threads 4 --sizein --sizeout -
-fasta_width 0 --log chimera.log
```

3.3 Taxonomic Classification

Once OTU representative sequences have been picked, they can be classified through sequence comparisons with the reference database. Data files required for downstream taxonomy and diversity analyses will also be generated.

1. Prepare reference sequence files for taxonomic assignment using cutadapt. Extract the variable region from the genes by mapping amplicon primer sequences, and trimming the boundaries (*see Note 12*). The reverse amplicon primer should be reverse complemented in the command.

```
cutadapt -e 0.2 -g FORWARDPRIMERSEQUENCE
ReferenceSequences.fasta > temp.fasta 2> cutadapt.log
cutadapt -e 0.2 -a REVERSEPRIMERSEQUENCE temp.fasta >
ReferenceSequence_Amplicon.fasta 2>> cutadapt.log
```

2. Assign taxonomy based on sequence similarity using global pairwise alignment in VSEARCH (*see Note 13*). A low sequence-identity threshold is used to identify the closest relatives for organisms not represented in the reference database. Sequences with low percentage identity may be filtered out at a later stage. Note that the algorithm returns more than one hit per OTU, so a second step is included to retain the hit with the longest alignment.

```
vsearch --usearch_global otus.fasta --db
/path/to/reference/ ReferenceSequence_Amplicon.fasta --
maxaccepts 0 --maxrejects 0 --top_hits_only --
output_no_hits --strand both --id 0.5 --iddef 1 --userout
assigned_otus.uc --userfields
target+query+id+eval+alnlen --threads 4 --rowlen 0 --
alnout otus.aln 2> assigned_otus.log

#extract only top hit for each OTU
sort -k5,5nr assigned_otus.uc | sort -u -k2,2 | awk -F
'\t' '{gsub("../", "\t", $1); print}' | sed
's/;size=.*//g' | sed 's/ /\t/g' > assigned_otus.tophit
```

3. Construct an OTU table, containing the number of reads of each OTU found in each sample. The awk command below creates a sparse matrix from the otu_readmembership.uc file (*see Subheading 3.2, step 13*), containing three columns: OTU, sample name, and number of reads. A sparse data matrix is a more efficient way to store this data as a majority of OTUs will be found only in a few samples. It can easily be converted to a contingency table in R (*see Subheading 3.4, step 1*).

```
awk -F '\t' '{print $10, $9}' otu_readmembership.uc | sed
's/;size=.*Sample/_Sample/g' | sed 's/.*size=/\t/g' |
sed 's///g' | awk '{arr[$1]+=$2} END {for (i in arr)
{print i,arr[i]}}' | sed 's/[ ]/\t/g' > otu.table
```

4. Construct an OTU metadata file, containing information on the reference database matching sequence and percent sequence identity, taxonomic information, sequence length, number of reads in the OTU, and results from chimera prediction. This file may be used to filter out OTUs based on various sequence similarity cutoffs, putative chimeric status, etc. Taxonomic information should be obtained from QIIME-formatted SILVA taxonomy files.

```

#OTU read length
awk '/^>/ {print}' otus.fasta | sed
's/>\(.*\);size=.*\1/g' > templen1
awk '!/^>/ {print length}' otus.fasta > templen2
paste templen1 templen2 | sort -k1,1 > temp.length
rm templen1 templen2

#Chimeric status
awk -F ' ' '{print $2, $18}' otus_chimera.uc | sed
's/;size=.*; /\t/g' | sort -k1,1 > temp.chimera

#OTU sizes (from otu_readmembership.uc file, Subheading
3.2, step 13) (see Note 14)
awk -F '\t' '{print $10, $9}' otu_readmembership.uc | sed
's/;size=.*size=/\t/g' | sed 's//\t/g' | awk
'{arr[$1]+=$2} END {for (i in arr) {print i, arr[i]}} ' |
sort -k1,1 > otu.sizes

#Taxonomy information (assumes a 7-level taxonomy from a
QIIME-formatted taxonomy file) (see Note 15)
sed 's/ _/_g' /path/to/taxonomy/taxonomyfile.txt | awk -F
'\t' '{gsub("\..*", "\t", $1); print}' > temp.taxonomy
awk 'NR==FNR {arr[$1] = $2; next} {if($1 in arr) {print
$0, arr[$1]} else print $0, "D_0__unknown;D_1__unknown;
D_2__unknown;D_3__unknown;D_4__unknown;D_5__unknown;
D_6__unknown"}' temp.taxonomy assigned_otus.tophit | sort
-k2,2 > otus.taxonomy

#Merge OTU information with length, chimeric prediction
and size information.
awk 'NR==FNR {arr[$1] = $2; next} {if($1 in arr) {print
$0, arr[$1]} else print $0, "*"}' temp.length temp.chimera
> temp.lengthchimera
awk 'NR==FNR {arr[$1] = $2; next} {if($1 in arr) {print
$0, arr[$1]} else print $0, "*"}' otu.sizes
temp.lengthchimera > temp.lengthchimerasize

join -1 1 -2 2 -o 1.1 1.4 1.3 1.2 2.1 2.3 2.4 2.5 2.6
temp.lengthchimerasize otus.taxonomy | sort -nr -k2,2 |
sed 's/ /\t/g' | sed '1

```

```
i\OTU\tReads\tLength\tChimera\tMatch\tID\tValue\tAInLength\tMatchTaxonomy' > otu.metadata

#Remove temporary files
rm temp.length temp.chimera temp.lengthchimera
temp.lengthchimerasize otu.sizes
```

5. Construct a table containing sample metadata, including sample names as the first row, followed by other relevant data (sample type, source, clinical or environmental data).

3.4 Taxonomic Analysis

Analysis of taxonomic data, including plotting species richness, rarefaction and relative proportions of taxa are carried out in R, using the Phyloseq package [36] (see Note 16). As diversity analysis and ordination methods are discussed in detail in other chapters, we will only demonstrate how to calculate initial species richness, rarefy the sequences, and plot taxonomic composition. To create a Phyloseq object, an OTU table, OTU metadata including taxonomic information, sample metadata, and optionally a phylogenetic tree are required.

Phyloseq tutorials and an example demonstration may be found on the Phyloseq GitHub page: <https://joey711.github.io/phyloseq/tutorials-index.html>, and <https://joey711.github.io/phyloseq-demo/phyloseq-demo.html>.

1. Load packages in RStudio and prepare the OTU table and metadata file, as well as the sample data file. Note that the symbol “>” denotes a command prompt in R.

```
# Load packages.
library("vegan")
library("reshape2")
library("ggplot2")
library("phyloseq")
library("stringr")

# Import the sparse OTU table and convert it into a
contingency table.
otu.sparsetable <- read.delim(file = "otu.table", sep = "\t",
header = FALSE)
colnames(otu.sparsetable) <- c("otu", "sample", "count")
otutable <- dcast(otu.sparsetable, otu ~ sample,
value.name="count", fill = 0)
rownames(otutable) <- otutable$otu
otutable <- otutable[,2:ncol(otutable)]
```

```

#Import the otu.metadata file. You may filter out OTUs based
on any desired parameters in the metadata file (chimeric
prediction, length, size, percentage identity to matched
reference sequence, etc.)
otumetadata <- read.delim(file = "otu.metadata", sep =
"\t", header = TRUE)
otutaxonomy <- data.frame(str_split_fixed(otumetadata$Match-
Taxonomy, ";", 7), row.names = otumetadata$OTU)
colnames(otutaxonomy) <- c("Domain", "Kingdom", "Phylum", "
Class", "Order", "Genus", "Species") #(see Note 17)
otutaxonomy <- apply(otutaxonomy, MARGIN = 2, function
(x) (gsub(pattern = "^D.*__", replacement = "", x)))

#Import the sample.metadata file.
samplemetadata <- read.delim(file = "sample.metadata", sep =
"\t", header = TRUE, row.names = 1)

```

2. Create a Phyloseq object, consisting of an OTU table, OTU taxonomic data, and sample metadata (and optionally a phylogenetic tree).

```

OTU <- otu_table(otutable, taxa_are_rows = TRUE)
TAX <- tax_table(otutaxonomy)
SAMPLEMAP <- sample_data(samplemetadata)
data <- merge_phyloseq(OTU, TAX, SAMPLEMAP)

#View components of Phyloseq object.
otu_table(data)
tax_table(data)
sample_data(data)
#Remove samples with fewer than 1000 sequences (see Note 18).
data <- prune_samples(sample_sums(data)<1000, data)

```

3. Calculate and plot alpha diversities of samples. You may compare alpha diversities based on a variable in the sample metadata file (e.g., age group of subject).

```

plot_richness(data)
plot_richness(data, measures = c("Chao", "Shannon"), x =
"age_group") + geom_boxplot(alpha = 0.1)

```

4. Plot rarefaction curves.

```

minReads <- min(colSums(otu_table(data)))
rarecurve(t(otu_table(data)), step = 20, sample =
minReads, xlab = "Sample Size", ylab = "Species", label =
TRUE)

```

5. Remove OTUs with fewer than five reads. Most OTUs will be present in only a few samples and in very low abundance. The confidence in the identification of taxa based on only one or two reads is debatable.

```
data_subset <- data %>% prune_taxa(taxa_sums(.) > 4, .)
```

6. Rarefy data to some minimum threshold, based on the rarefaction plot—e.g., 10,000 reads (*see Note 18*).

```
data_rarefied <- rarefy_even_depth(data_subset, sample.size = 10000)
```

7. Agglomerate data to higher taxonomic level and plot relative abundances of organisms in each sample.

```
#Agglomerate OTUs to phylum level
data_phylum <- tax_glom(data_subset, taxrank = 'Phylum')

#Convert read counts to relative abundance
data_phylum.r <- transform_sample_counts(data_phylum,
function(x) x/sum(x))

#Plot data
plot_bar(data_rarefied, fill = "Phylum")
plot_bar(data_phylum.r, fill = "Phylum")
```

More details on beta-diversity calculations and ordinations can be found in other chapters.

4 Notes

1. Illumina recommends Kapa polymerase. Other polymerases used in published amplicon studies include high-fidelity Taq and Phusion polymerases. We have best results with the BioRad iProof polymerase.
2. Annealing temperature and elongation time may be adjusted according to the primers used and the amplicon length.
3. The most widely used programs for 16S bacterial diversity analysis are QIIME [42] and Mothur [43]. QIIME is a data-processing pipeline, which makes use of a large number of external algorithms developed by other research groups for various steps. Installation is not trivial, and requires setup of many dependencies; however, this enables the user to pick and choose the latest developed tools for individual steps. Mothur, on the other hand, is a standalone program easy to install. Any algorithms used that were developed by outside groups were reimplemented into Mothur, and in some cases improved. QIIME and Mothur follow similar data-processing pipelines and require curated reference sequence databases from

Greengenes [44], RDP (16S) [45], or SILVA (16S and 18S) [18]. Sequence data processing described in our workflow relies primarily on VSEARCH, along with several other referenced tools. VSEARCH is an alternative open-source suite modeled after USEARCH. It has recently been integrated as an option in QIIME, and its clustering algorithm has been added to Mothur. We find it easier to directly manipulate the parameters of the individual algorithms and to customize the data workflow.

4. Portions of this workflow were adapted from the protocol published on a Wiki page by Frederic Mahé (<https://github.com/frederic-mahe/swarm/wiki/Fred's-metabarcoding-pipeline>).
5. Quality statistics on sequence files may also be generated using VSEARCH, as below; however, VSEARCH does not output graphical summaries.

```
vsearch --fastq_chars ${FILE}_R1mod.fastq 2>
${FILE}_R1.stats
```

6. Alternative tools which may be used for removal of adaptor or primer sequences include cutadapt and the Fastx Toolkit (http://hannonlab.cshl.edu/fastx_toolkit/). Fastx and VSEARCH may also be used to remove low-quality bases. VSEARCH, however, truncates the read at the first base that falls beneath a specified threshold, rather than inspect the average quality over a sliding window, as Trimmomatic does.
7. Trimmomatic outputs four files, two containing forward and reverse reads where both paired reads survived trimming, and two files where only one of the paired reads survived trimming. If a high percentage of reads are merged in step 7, only trimmed paired read files should be used in subsequent processing steps. If a low percentage of reads are merged, R1 and R2 files should be trimmed individually as follows, and processing continued individually for the resulting files.

```
java -jar /path/to/trimmomatic/trimmomatic-0.36.jar SE
-phred33 -trimlog SampleA_trim.log SampleA_R1mod.fastq
SampleA_R1_trimmed.fastq
ILLUMINACLIP:primers.fasta:2:30:10 LEADING:3 TRAILING:3
SLIDINGWINDOW:4:15 MINLEN:250 CROP:250
```

8. QIIME and Mothur protocols first merge sequences, then trim low-quality bases and remove reads with ambiguous bases. We have found that this results in a higher proportion of unmerged, discarded reads, using the VSEARCH algorithm, due to high numbers of mismatches in the overlap region.

9. It is possible to further reduce the data with a second dereplication step, once all sequences have been concatenated into one file. However, if identical sequences from different samples are grouped together, information on the source of sequences and abundance will be lost, unless this information is tabulated first.
10. The PhiX genome is spiked into amplicon sequencing runs, at 5–10% generally, as the Illumina platform performs poorly on samples with low diversity (in this case sequencing of many similar copies of one gene). PhiX is usually removed by the sequencing facility, but the step may be added as a precaution. The PhiX genome may be downloaded from the Illumina website.
11. The `--cluster_fast` algorithm is a *de novo* OTU picking method, sensitive to the order of input reads. To circumvent some of the associated biases, once the OTU representative sequences have been chosen, a separate step (**step 13**) applies a global sequence aligner (`--usearch_global`) to assign reads to the OTU with the highest sequence similarity. Two other commonly used OTU picking strategies include closed-reference and open-reference. Closed-reference OTU picking clusters reads against a reference database, discarding those without sequence similarity above the defined threshold, whereas in open-reference OTU picking, reads not assigned to known references are clustered *de novo* and included in downstream analyses. We use *de novo* OTU picking, where reads are naïvely clustered in the absence of reference sequence information, as relatively few reference sequences are available for eukaryotic microbes, particularly for protist taxa. We prefer this approach to maximize potential discovery of uncharacterized taxa. Clustering algorithms are continually being improved, and included here are several alternative programs which may be used: UCLUST [35], SUMACLUST [46], a read-order independent algorithm called Swarm [47], and Mothur [48].
12. Alternatively, the variable region may be extracted from a sequence alignment of the reference genes, if the start and end positions are known. Note that in alignments available for download from the SILVA archive (`*_full_align_trunc.fastq`), unaligned nucleotides were truncated from the genes; therefore, the genes are not complete. Information about the archived files may be found in the `README.txt` file within each folder.
13. There are a number of taxonomic assignment methods for amplicon data. QIIME and Mothur offer several assignment methods, including the RDP classifier, global alignment methods, UCLUST, and BLAST. While the RDP classifier has a 16S training data set popular for taxonomic classification of

bacteria, no training data set exists for 18S data, and the user would have to create his or her own. The SINA classifier [49] is a good alternative for classification of 18S sequence data against the SILVA reference database. However, the training data sequences are not publicly available.

14. Not all OTUs will necessarily have reads assigned to them. During OTU representative sequence picking, the `--cluster_fast` algorithm clusters reads within the defined threshold (e.g., 97%) in order that they appear in the FASTA file. During the subsequent global alignment step, some reads may be found to have higher identity to a downstream OTU representative sequence.
15. Some 18S gene accessions correspond to genome positions. In some cases, there are multiple genes from the same genome, but encoded at different genomic positions. To remove some inconsistencies between accessions in the 18S reference sequence set and the taxonomy files, the genome position numbers are stripped using the `sed` command.
16. Other tools for taxonomic diversity analyses are available through QIIME or the R package `vegan`.
17. Note that the taxonomic information in each column of the QIIME 18S 7-level taxonomy files does not always belong to the same rank. Manual curation may be required to fix these issues.
18. Samples may be rarefied to “minimum depth,” that is to the number of reads contained in the smallest sample. It is best to discard samples containing too few reads. While there is no consensus on a minimum read threshold, and as few as 150 sequences per sample have been used [9], we apply a minimum cutoff of 1000 sequences. A rarefaction curve generated on the sample would help inform on the cutoff above which the number of taxa begins to plateau.

References

1. Gill SR et al (2006) Metagenomic analysis of the human distal gut microbiome. *Science* 312:1355–1359. <https://doi.org/10.1126/science.1124234>
2. Martin F-PJ et al (2007) A top-down systems biology view of microbiome-mammalian metabolic interactions in a mouse model. *Mol Syst Biol* 3:112. <https://doi.org/10.1038/msb4100153>
3. Maurice CF, Haiser HJ, Turnbaugh PJ (2013) Xenobiotics shape the physiology and gene expression of the active human gut microbiome. *Cell* 152:39–50. <https://doi.org/10.1016/j.cell.2012.10.052>
4. Mazmanian SK, Liu CH, Tzianabos AO, Kasper DL (2005) An immunomodulatory molecule of symbiotic bacteria directs maturation of the host immune system. *Cell* 122:107–118. <https://doi.org/10.1016/j.cell.2005.05.007>
5. Turnbaugh PJ, Ley RE, Mahowald MA, Magrini V, Mardis ER, Gordon JI (2006) An obesity-associated gut microbiome with increased capacity for energy harvest. *Nature* 444:1027–1031. <https://doi.org/10.1038/nature05414>

6. Stanley SL Jr, Reed SL (2001) Microbes and microbial toxins: paradigms for microbial-mucosal interactions. VI. Entamoeba histolytica: parasite-host interactions. Am J Physiol Gastrointest Liver Physiol 280: G1049–G1054. <https://doi.org/10.1152/ajpgi.2001.280.6.G1049>
7. Upcroft P, Upcroft JA (2001) Drug targets and mechanisms of resistance in the anaerobic protozoa. Clin Microbiol Rev 14:150–164. <https://doi.org/10.1128/cmr.14.1.150-164.2001>
8. Andersen LO, Vedel Nielsen H, Stensvold CR (2013) Waiting for the human intestinal Eukaryotome. Isme j 7:1253–1255. <https://doi.org/10.1038/ismej.2013.21>
9. Parfrey LW et al (2014) Communities of microbial eukaryotes in the mammalian gut within the context of environmental eukaryotic diversity. Front Microbiol 5:298. <https://doi.org/10.3389/fmicb.2014.00298>
10. Scanlan PD, Stensvold CR, Rajilic-Stojanovic M, Heilig HG, De Vos WM, O'Toole PW, Cotter PD (2014) The microbial eukaryote *Blastocystis* is a prevalent and diverse member of the healthy human gut microbiota. FEMS Microbiol Ecol 90:326–330. <https://doi.org/10.1111/1574-6941.12396>
11. Underhill DM, Iliev ID (2014) The mycobiota: interactions between commensal fungi and the host immune system. Nat Rev Immunol 14:405–416. <https://doi.org/10.1038/nri3684>
12. Hadziavdic K, Lekang K, Lanzen A, Jonassen I, Thompson EM, Troedsson C (2014) Characterization of the 18S rRNA gene for designing universal eukaryote specific primers. PLoS One 9:e87624. <https://doi.org/10.1371/journal.pone.0087624>
13. Hugert LW et al (2014) Systematic design of 18S rRNA gene primers for determining eukaryotic diversity in microbial consortia. PLoS One 9:e95567. <https://doi.org/10.1371/journal.pone.0095567>
14. Loy A, Horn M, Wagner M (2003) probeBase: an online resource for rRNA-targeted oligonucleotide probes. Nucleic Acids Res 31:514–516
15. Machida RJ, Knowlton N (2012) PCR primers for metazoan nuclear 18S and 28S ribosomal DNA sequences. PLoS One 7:e46180. <https://doi.org/10.1371/journal.pone.0046180>
16. Pawłowski J et al (2012) CBOL protist working group: barcoding eukaryotic richness beyond the animal, plant, and fungal kingdoms. PLoS Biol 10:e1001419. <https://doi.org/10.1371/journal.pbio.1001419>
17. Wang Y, Tian RM, Gao ZM, Bougouffa S, Qian PY (2014) Optimal eukaryotic 18S and universal 16S/18S ribosomal RNA primers and their application in a study of symbiosis. PLoS One 9:e90053. <https://doi.org/10.1371/journal.pone.0090053>
18. Quast C et al (2013) The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. Nucleic Acids Res 41:D590–D596. <https://doi.org/10.1093/nar/gks1219>
19. Amaral-Zettler LA, McCliment EA, Ducklow HW, Huse SM (2009) A method for studying protistan diversity using massively parallel sequencing of V9 hypervariable regions of small-subunit ribosomal RNA genes. PLoS One 4:e6372. <https://doi.org/10.1371/journal.pone.0006372>
20. Hart ML, Meyer A, Johnson PJ, Ericsson AC (2015) Comparative Evaluation of DNA Extraction Methods from Feces of Multiple Host Species for Downstream Next-Generation Sequencing. PLoS One 10: e0143334. <https://doi.org/10.1371/journal.pone.0143334>
21. Dawson SC, Pace NR (2002) Novel kingdom-level eukaryotic diversity in anoxic environments. Proc Natl Acad Sci U S A 99:8324–8329. <https://doi.org/10.1073/pnas.062169599>
22. Kim E et al (2011) Newly identified and diverse plastid-bearing branch on the eukaryotic tree of life. Proc Natl Acad Sci U S A 108:1496–1500. <https://doi.org/10.1073/pnas.1013337108>
23. Carlton JM et al (2007) Draft genome sequence of the sexually transmitted pathogen *Trichomonas vaginalis*. Science 315:207–212. <https://doi.org/10.1126/science.1132894>
24. Gardner MJ et al (2002) Genome sequence of the human malaria parasite *Plasmodium falciparum*. Nature 419:498–511. <https://doi.org/10.1038/nature01097>
25. Hebert PD, Cywinski A, Ball SL, deWaard JR (2003) Biological identifications through DNA barcodes. Proc Biol Sci 270:313–321. <https://doi.org/10.1098/rspb.2002.2218>
26. Odronitz F, Kollmar M (2007) Drawing the tree of eukaryotic life based on the analysis of 2,269 manually annotated myosins from 328 species. Genome Biol 8:R196. <https://doi.org/10.1186/gb-2007-8-9-r196>
27. Parfrey LW et al (2010) Broadly sampled multigene analyses yield a well-resolved eukaryotic tree of life. Syst Biol 59:518–533. <https://doi.org/10.1093/sysbio/syq037>
28. Rodriguez-Ezpeleta N, Brinkmann H, Burger G, Roger AJ, Gray MW, Philippe H,

- Lang BF (2007) Toward resolving the eukaryotic tree: the phylogenetic positions of jakobids and cercozoans. *Curr Biol* 17:1420–1425. <https://doi.org/10.1016/j.cub.2007.07.036>
29. Tekle YI, Grant JR, Kovner AM, Townsend JP, Katz LA (2010) Identification of new molecular markers for assembling the eukaryotic tree of life. *Mol Phylogenet Evol* 55:1177–1182. <https://doi.org/10.1016/j.ympev.2010.03.010>
30. Bates ST, Berg-Lyons D, Lauber CL, Walters WA, Knight R, Fierer N (2012) A preliminary survey of lichen associated eukaryotes using pyrosequencing. *Lichenologist* 44:137–146. <https://doi.org/10.1017/S0024282911000648>
31. Andrews S (2017) FastQC. <http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>. Accessed 6 Mar 2017
32. Bolger AM, Lohse M, Usadel B (2014) Trimomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics* 30:2114–2120. <https://doi.org/10.1093/bioinformatics/btu170>
33. Martin M (2011) Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet J* 17:10–12. <https://doi.org/10.14806/ej.17.1.200>
34. Rognes T, Flouri T, Nichols B, Quince C, Mahe F (2016) VSEARCH: a versatile open source tool for metagenomics. *PeerJ* 4:e2584. <https://doi.org/10.7717/peerj.2584>
35. Edgar RC (2010) Search and clustering orders of magnitude faster than BLAST. *Bioinformatics* 26:2460–2461. <https://doi.org/10.1093/bioinformatics/btq461>
36. McMurdie PJ, Holmes S (2013) phyloseq: an R package for reproducible interactive analysis and graphics of microbiome census data. *PLoS One* 8:e61217. <https://doi.org/10.1371/journal.pone.0061217>
37. Oksanen JB, Guillaume F, Friendly M, Kindt R, Legendre P, McGlinn D, Minchin PR, O'Hara RB, Simpson GL, Solymos P, Henry M, Stevens H, Szoecs E, Wagner H (2017) Vegan: Community Ecology Package. <https://CRAN.R-project.org/package=vegan>
38. Wickham H (2017) stringr: Simple, consistent wrappers for common string operations. R package version 1.2.0.
39. Wickham H (2007) Reshaping data with the reshape package. *J Stat Softw* 21:1–20
40. Wickham H (2009) ggplot2: Elegant Graphics for Data Analysis. Springer, New York
41. Paradis E, Claude J, Strimmer K (2004) APE: analyses of phylogenetics and evolution in R language. *Bioinformatics* 20:289–290
42. Caporaso JG et al (2010) QIIME allows analysis of high-throughput community sequencing data. *Nat Methods* 7:335–336. <https://doi.org/10.1038/nmeth.f.303>
43. Schloss PD et al (2009) Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Appl Environ Microbiol* 75:7537–7541. <https://doi.org/10.1128/aem.01541-09>
44. DeSantis TZ et al (2006) Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB. *Appl Environ Microbiol* 72:5069–5072. <https://doi.org/10.1128/AEM.03006-05>
45. Cole JR et al (2014) Ribosomal Database Project: data and tools for high throughput rRNA analysis. *Nucleic Acids Res* 42:D633–D642. <https://doi.org/10.1093/nar/gkt1244>
46. Mercier C (2018) Sumaclust: fast and exact clustering of sequences. <https://git.metabarcoding.org/obitools/sumaclust.git>. Accessed 28 Jan 2018
47. Mahe F, Rognes T, Quince C, de Vargas C, Dunthorn M (2014) Swarm: robust and fast clustering method for amplicon-based studies. *PeerJ* 2:e593. <https://doi.org/10.7717/peerj.593>
48. Schloss PD, Westcott SL (2011) Assessing and improving methods used in operational taxonomic unit-based approaches for 16S rRNA gene sequence analysis. *Appl Environ Microbiol* 77:3219–3226. <https://doi.org/10.1128/aem.02810-10>
49. Pruesse E, Peplies J, Glockner FO (2012) SINA: accurate high-throughput multiple sequence alignment of ribosomal RNA genes. *Bioinformatics* 28:1823–1829. <https://doi.org/10.1093/bioinformatics/bts252>



Chapter 4

Culture and Molecular Profiling of the Respiratory Tract Microbiota

Fiona J. Whelan, Laura Rossi, Jennifer C. Stearns, and Michael G. Surette

Abstract

Microbiome research of host-associated communities has been advanced recently through improvements in sequencing technologies and bioinformatic methods. Traditional microbiological culture, when combined with molecular techniques, can provide a sensitive platform to comprehensively study the airway microbiota. Here we describe the culture methods necessary to capture a large proportion of the airway microbiota and molecular methods for profiling bacterial communities through the 16S rRNA gene, which, when combined, offer a more complete picture of the diversity of airway microbial communities than either method alone.

Key words Culture enrichment, Respiratory tract microbiota, Microbiome, 16S rRNA gene, Human microbiota

1 Introduction

The human respiratory tract generally describes the space between the nasal cavity and the lungs and is often split into “upper” and “lower” regions at the larynx. The upper respiratory tract (URT) houses microbiota with a density estimated to be in the range of 10^3 – 10^6 CFU per nasal swab (nasopharynx) or mL of oral wash (oropharynx) [1]; previous research has determined that the composition of the microbiota at various regions within the URT—including the anterior nares, nasopharynx, oropharynx, and mouth—are distinct from each other [2–4]. These distinctions become evident in childhood [2, 5] and persist until old age, at which time these distinct communities again become more similar [6, 7]. In contrast, the lower respiratory tract (LRT) of healthy adults has a lower microbial density (approximately 10^2 CFU/mL) [1]. Although these sites were once considered sterile (Laurenzi 1961), recent improvements in culture-independent techniques have consistently observed microbial signatures suggesting a bacterial presence in these locales [3, 8–10].

Although improvements in culture-independent sequencing technologies has increased our knowledge of the microbiota of the respiratory tract, using these methods in conjunction with culture-dependent techniques can provide more sensitive and complete information about these communities. Previous studies have shown that a substantial proportion of microbes can readily be cultured from many sites within the human body including the gut and respiratory tract [11–13]; culture enrichment captures greater microbial diversity by allowing low abundance organisms normally missed by culture-independent techniques to multiply and be captured by 16S rRNA gene sequencing. Further, the combined use of culture-dependent and -independent approaches ensures that only viable organisms are identified, aiding in a better understanding of low-biomass communities in locations such as the LRT. Lastly, being able to freeze the cultured growth as stocks allows for future investigations of the individual members of these communities in laboratory experiments, which is the only avenue for study of the mechanisms important to host–microbe interactions.

Culture enrichment can be applied to any respiratory tract specimen including nasal swabs, oral washes, and bronchoalveolar lavages (BALs). Upon sample collection, a subset of homogenized specimen is diluted in laboratory broth media and plated on various media types including nonselective and selective agars. Selective media are those which only allow a subset of bacterial species to proliferate due to the inclusion of antibiotics or particular nutrients in their composition. For example, fastidious bacteria have a complex set of nutrients which are required for their growth on agar media. Further, environmental oxygen levels can be controlled to produce anaerobic, hypoxic, or aerobic growth environments in order to culture a wide array of organisms. These approaches can be used to target the isolation of a particular organism of interest (for example, the *Lachnospiraceae* [10]) or to generally enrich a respiratory microbial community [11].

2 Materials

2.1 Sample Types

1. Swab of the nasopharynx: The nasopharynx is the area at the very back of the nasal cavity. This area can be swabbed by inserting a swab soaked in transport media (e.g., Copan's Universal Transport Medium™) through the nasal cavity [2].
2. Swab of the anterior nares: In contrast to sampling of the nasopharynx, the anterior nares represent the area just inside the nasal cavity. This area can similarly be sampled with a soaked cotton swab [6, 14].
3. Swab of the oropharynx: The oropharynx is the area at the very back of the throat, above the laryngopharynx. This area can be swabbed by inserting a swab soaked in transport media to the back of the throat [6, 14].

4. Saliva: The oral microbiota can be sampled by examining the contents of saliva. This sample is collected by asking participants to produce saliva into a sterile collection tube [15].
5. Dental plaque: Plaque scrapings can be taken from either the periodontal pockets (subgingival) or from the exposed tooth surface (supragingival) using a sterile curette [16].
6. Sputum: In the case of sampling the LRT, sputum samples can be collected via the oral cavity. It is important to differentiate saliva from sputum; generally, sputum can only be collected from individuals with respiratory infection or diseases which dramatically increase the microbial density of the LRT; this includes diseases such as chronic obstructive pulmonary disease (COPD), asthma, and cystic fibrosis (CF). Expectorated sputum can be collected during activities such as chest physical therapy to increase the sample volume.
7. Bronchoalveolar lavage or brushing: A sample from the LRT can be collected by inserting a tube into one of the bronchi and either a sterile solution can be injected into the space then aspirated up the tube or a sterile brush can be inserted through the tube to collect material from the lung tissue itself [9, 10].

2.2 Culture Recipes

Below outlines the various culture recipes and the associated stock solutions. There are a number of high-quality suppliers of the various reagents needed including Sigma-Aldrich, Thermo Fisher Scientific, BD.

2.2.1 Stock Solutions

1. L-Cysteine: 500 mg/mL dissolved in H₂O and filter sterilized.
2. Brain Heart Infusion (BHI) broth supplemented with L-cysteine (0.5 g/L).
3. Hemin: 5 mg/mL stock made in 25 mM NaOH and filter sterilized.
4. Colistin: 5 mg/mL stock made in H₂O and filter sterilized.
5. Kanamycin: 100 mg/mL dissolved in H₂O and filter sterilized.
6. Vancomycin: 100 mg/mL dissolved in H₂O and filter sterilized.
7. Vitamin K: mix Vitamin K powder in ethanol in a 1:1 ratio.

Specifically for McKay Media

1. M-stock hemin: 0.015 g to 1 mL of 1 M NaOH; bring up to 30 mL with H₂O.
2. M-stock colistin: 0.03 g colistin sulfate in 30 mL H₂O (filter sterilize).
3. M-stock oxolinic acid: 0.015 g in 30 mL 0.1 M NaOH (filter sterilize).
4. M-stock L-arginine-HCl: 1.5 g of L-Arg-HCl to 60 mL H₂O (filter sterilize).

5. M-stock sulfadiazine: 1.6 g to 90 mL 0.1 M NaOH; bring up to 100 mL (filter sterilize).
6. M-stock: Salt solution: add 5 g NaHCO₃, 1 g NaCl, 0.5 g K₂HPO₄, 0.25 g MgSO₄·7H₂O, and 500 mL H₂O, then filter sterilize through a 0.22 µm filter.

2.2.2 Nonselective Microbiological Media

1. Brain Heart Infusion Agar with Supplements (BHI + sup) (*see Note 1*). Follow the manufacturer's instructions to mix the appropriate amount of BHI agar powder with 500 mL of ddH₂O, then autoclave (*see Note 2*) and cool. Next add supplements (*see Note 3*) and pour into sterile Petri dishes.
2. Cooked Meat Agar with Supplements (Beef+sup) (*see Note 1*). Follow the manufacturer's instructions to mix the appropriate amount of Cooked Meat with Agar Powder with 500 mL of ddH₂O, then autoclave (*see Note 2*) and cool. Next add supplements (*see Note 3*) and pour into sterile Petri dishes.
3. Tryptic Soy Yeast Agar with Supplements (TSY + sup). Follow the manufacturer's instructions to mix the appropriate amount of Tryptic Soy Agar powder and 1.5 g yeast extract with 500 mL of ddH₂O, then autoclave (*see Note 2*) and cool. Next add supplements (*see Note 3*) and pour into sterile Petri dishes.

2.2.3 Enrichment Microbiological Media

1. Actinomycete Isolation Agar (AIA) for enrichment of Actinomycetes. Follow manufacturer's instructions to mix the appropriate amount of AIA powder with 500 mL of ddH₂O, then add 5 mL of glycerol, autoclave (*see Note 2*), and pour into sterile Petri dishes.
2. Columbia Blood Agar (CBA) for enrichment of fastidious organisms. Follow the manufacturer's instructions to mix the appropriate amount of CBA powder with 500 mL of ddH₂O, then autoclave (*see Note 2*) and cool. Add 25 mL sheep's blood to a final concentration of 5% and pour into sterile Petri dishes.
3. Chocolate Agar (CHOC) for enrichment of *Haemophilus influenzae*, *Neisseria meningitidis*, *Streptococcus pneumoniae*, and other fastidious pathogenic respiratory bacteria. Follow the manufacturer's instructions to mix the appropriate amount of GC Medium with 200 mL of ddH₂O, then autoclave (*see Note 2*) and cool. Next add 200 mL of hemoglobin (warmed to room temperature) and 2 mL of IsoVitaleX Enrichment (BD# B211875) and pour into sterile Petri dishes.
4. Fastidious Anaerobe Agar (FAA) for enrichment of fastidious anaerobic bacteria. Follow the manufacturer's instructions to mix the appropriate amount of FAA powder with 500 mL of ddH₂O, then autoclave (*see Note 2*) and pour into sterile Petri dishes.

2.2.4 Selective Microbiological Media

1. Columbia Agar with Colistin, Nalidixic Acid, and Blood (CNA) for selection of Gram positive cocci. Follow the manufacturer's instructions to mix the appropriate amount of CNA powder with 500 mL of ddH₂O, then autoclave (*see Note 2*) and cool. Next add 25 mL sheep's blood to a final concentration of 5% and pour into sterile Petri dishes.
2. Kanamycin/Vancomycin Laked Blood Agar (KVLB) for selection of Gram negative, anaerobic bacilli. Follow the manufacturer's instructions to mix the appropriate amount of Trypticase Soy Agar and 2.5 g of yeast extract with 500 mL of ddH₂O, then autoclave (*see Note 2*) and cool. Next add 10 µL of stock Vitamin K, 500 µL of stock hemin, 400 µL of stock L-cysteine, 500 µL of stock Kanamycin, 37.5 µL of stock Vancomycin, and 25 mL laked blood (*see Note 4*) and pour into sterile Petri dishes.
3. MacConkey Agar (MAC) for selection of Gram negative and enteric bacilli. Follow the manufacturer's instructions to mix the appropriate amount of MacConkey agar powder with 500 mL of ddH₂O, then autoclave (*see Note 2*) and pour into sterile Petri dishes.
4. McKay Agar (McKay) for selection of *Streptococcus milleri* group bacteria. First prepare solution A by mixing 40 g nutrient broth powder with 1 L of ddH₂O. Next in a 2.8 L wide-mouth flask, mix 15 g glucose, 30 g yeast extract, 15 g tryptone, and 6 g K₂HPO₄ with 600 mL ddH₂O until dissolved. While stirring, add 2 mL Tween-80, 120 mL of M-stock salt solution, and all of solution A. To this, slowly add 200 µL of 1 M CaCl₂ to avoid precipitation, then add 30 mL M-stock hemin and 6 µL of stock Vitamin K. Next, bring the volume up to 2800 mL and transfer to a 6 L flask. To this add 45 g Bacto Agar and adjust the pH to 7.2 if necessary. Next add two indicator dyes: 0.18 g bromocresol purple and 0.003 g crystal violet (included in order to indicate spots of high acidity on the resulting media), then autoclave (*see Note 2*) and cool to 55 °C. Then while stirring, add 60 mL M-stock L-arginine-HCl, 30 mL M-stock L-cysteine, 30 mL M-stock oxolinic acid, and 93.75 mL M-stock sulfadiazine and pour into sterile Petri dishes.
5. Mannitol Salt Agar (MSA) for selection of Staphylococci. Follow the manufacturer's instructions to mix the appropriate amount of product with 500 mL of ddH₂O, autoclave (*see Note 2*) and pour into sterile Petri dishes.
6. Phenylethyl Alcohol Agar (PEA) for selection of Gram positive obligate anaerobic bacteria. Follow the manufacturer's instructions to mix the appropriate amount of PEA powder with 500 mL of ddH₂O, autoclave (*see Note 2*), cool, then add 25 mL sheep's blood to a final concentration of 5% and pour into sterile Petri dishes.

2.3 Solutions Needed for the Isolation of Bacterial Genomic DNA

1. GES Solution. Add 60 g guanidine thiocyanate (*see Note 5*) and 20 mL 0.5 M EDTA (pH = 8) to 20 mL DNA-free ddH₂O and heat to 65 °C. After cooling to room temperature, add 1 g *N*-lauroyl sarcosine and adjust volume to 100 mL with DNA-free ddH₂O. Filter sterilize and store at room temperature.
2. Sodium phosphate monobasic (200 mM). Add 4.8 g of mono-basic NaH₂PO₄ to 150 mL DNA-free H₂O, adjust the pH to 8. Next, adjust the volume to 200 mL with DNA-free ddH₂O, filter sterilize, and store at room temperature.
3. Lysozyme (100 mg/mL). Add 1 g of lysozyme to 10 mL DNA-free ddH₂O, then dissolve and filter sterilize. Store in 1 mL aliquots at -20 °C.
4. RNase A (10 mg/mL in H₂O from Qiagen, #19101).
5. 25% SDS diluted in ddH₂O then filter sterilized.
6. Proteinase K. Mix 150 µL 1 M Tris (pH = 8) and 1.5 mL 0.1 M calcium acetate with 8 mL DNA-free ddH₂O, then add 0.2 g proteinase K, dissolve, and filter sterilize. Store in 1 mL aliquots at -20 °C.
7. Sterile 5 M NaCl.
8. Buffered 25:24:1 phenol-chloroform-isoamyl alcohol (Sigma-Aldrich).

2.4 Reagents for Illumina Sequencing 16S rRNA Gene Tags

1. 10× PCR buffer (Life Technologies, #10342020).
2. MgCl₂ (50 mM; Life Technologies, #10342020).
3. dNTPs (10 mM; New England Biolabs, #N0447L).
4. Taq polymerase (Life Technologies, #10342020).
5. DNA-free H₂O (Thermo Fisher, #10977015).
6. Bovine Serum Albumin (BSA) (10 mg/mL): Mix 100 mg BSA powder with 10 mL DNA-free H₂O. Filter sterilize, aliquot into 1 mL aliquots, irradiate with UV for 20 min, and then store at -20 °C.
7. Barcoded Illumina primers for the v3 region of the bacterial 16S rRNA gene (*see Note 13*).

3 Methods

It is imperative to keep specimens in an anaerobic environment upon collection. Swabs should be collected directly into transport media which promotes the survival of anaerobic bacteria. Other sample types, such as saliva and sputum, should be transported on ice in airtight containers or bags with the appropriate anaerobic sachets (e.g., BD GasPak™ EZ pouch systems). Samples should be cultured as soon as possible and definitely within 4 h of production/isolation.

3.1 Plating

1. In an anaerobic environment, prepare the sample. In the case of swabs in transport medium, no additional work is required. In the case of saliva/sputum/plaque samples, use an 18 1 ½ gauge needle and syringe to homogenize the sample. More viscous samples may need to first be processed with the syringe only.
2. Save a 300 µL aliquot of the sample for molecular workup.
3. Make tenfold serial dilutions in BHI broth supplemented with L-cysteine (0.5 g/L).
4. Pipette 100 µL of the desired dilutions (usually 10^{-3} and 10^{-5} CFU/mL dilutions are plated) onto each of the 13 culture media types (*see Note 6*) listed in Subheading 2.2. Two of each media type for each dilution can be made since one will be incubated aerobically and the other anaerobically.
5. Disperse the sample using a sterile hockey stick or glass beads until it is spread homogeneously across the plate.
6. Save any unused sample immediately at -80°C .
7. Incubate anaerobic plates at 37°C for 3–5 days.
8. Incubate aerobic plates at 37°C with 5% CO_2 for 2–3 days.

3.2 Culture Plate Collection

After the appropriate number of days, profile and collect the bacterial growth on each plate.

1. Photograph each plate on the benchtop or inside the anaerobic environment (depending on the culture conditions).
2. If desired, count morphotypes and record phenotypic information about each colony type; estimate the total number of colonies per morphotype.
3. If conducting Subheading 3.3, pick an isolate of each morphotype into 5% Chelex solution (*see* Subheading 3.3 for more detail).
4. Add 2–3 mL of BHI + L-cysteine broth to the plate surface and use a hockey stick to gently scrape colonies into the media.
5. Pipette off the media and set aside 300 µL for Subheading 3.4, then mix the rest with 20% skim milk (to increase stock viability after thawing) in a 1:1 ratio and store at -80°C .

3.3 Bacterial Colony PCR

1. Pick fresh colonies from the agar surface (during Subheading 3.2) using a sterile pipette tip or toothpick into 50 µL aliquots of 5% autoclaved Chelex solution.
2. Boil aliquots for 15 min, then centrifuge aliquots at 6,000 rpm ($2,000 \times g$) for 5 min.
3. Use 2–5 µL of the supernatant in a 50 µL PCR amplification reaction containing 10 pmole of each 8F and 926R primers (*see Note 7*), 1.5 mM MgCl_2 , 0.2 mM dNTPs, 1× PCR buffer, and

1.25 units of Taq polymerase. Thermocycler conditions will vary slightly according to the Taq polymerase used but as a general guideline this should include activation of hotstart Taq (if using), then an initial denaturation step at 94 °C for 2 min, then 29 cycles of denaturation at 94 °C for 30 s, annealing at 56 °C for 30 s, extension at 72 °C for 1 min, then one final extension at 72 °C for 10 min.

4. The resulting PCR products can be sent for Sanger sequencing.

3.4 Isolation of DNA

This method can be used to isolate DNA (*see Note 8*) from clinical samples directly or from bacteria collected from culture plates as in Subheading 3.1.

1. 24 samples at a time is a manageable amount. Remove samples from the –80 °C and thaw at room temperature. For each sample, label a 2 mL screwcap, a 2 mL Eppendorf tube, and two 1.75 mL Eppendorf tubes.
2. Add 800 µL of 200 mM of monobasic NaPO₄ (pH = 8) and 100 µL of GES to a 2 mL plastic screw top tube containing glass beads (*see Note 9*). Add to this 300 µL or 0.1 g of the sample (*see Note 10*), then mechanically lyse cells using a PowerLyzer 24 Homogenizer bead beater for 3 min set to homogenize at 3000 rpm, with an additional cycle added after a 45 s delay for biopsies or other solid sample types.
3. To the sample add 50 µL lysozyme and 10 µL RNase A, mix by vortexing, then incubate in a 37 °C water bath for 1–1.5 h.
4. To the sample add 25 µL 25% SDS, 25 µL proteinase K, and 62.5 µL 5 M NaCl, mix by vortexing, incubate in a 65 °C water bath for 0.5–1.5 h, then centrifuge at 13,500 rpm (15,000 × *g*) for 5 min.
5. Add 900 µL phenol-chloroform-isoamyl alcohol to 2 mL eppendorf tubes. To this add 900 µL of the supernatant from the samples, mix by vortexing, and centrifuge at no more than 13,500 rpm (15,000 × *g*) for 10 min (*see Note 11*).
6. Column purify the DNA with a DNA cleanup kit such as DNA Clean and Concentrator-25 (Cedarlane Laboratories, #D4034) by following the manufacturer's instructions and elute the DNA in DNA-free H₂O.
7. Quantify DNA using a spectrophotometer and store at –20 °C.

3.5 PCR

Amplification of the v3 Region of the 16S rRNA Gene for Illumina Sequencing

The bacterial small subunit ribosomal (16S) gene has been adopted as the marker gene of choice for phylogenetic-based taxonomic assignment of bacteria, especially from mixed communities. This gene contains nine variable regions (v1–v9), that have diverged through time as bacteria evolved, that can be used to calculate

phylogenetic distance between bacterial taxa. Several studies have assessed the suitability of each variable region for use in taxonomic assignment; however, we have had success with the v3 region. As the Illumina chemistry improves it will be possible to reliably sequence longer reads, and hence collect more information about phylogeny, that extend into the v4 region. This section describes only v3 16S rRNA gene sequencing; however, primers described here can be exchanged for those of other variable regions of interest.

This protocol, adapted from [17], is used to PCR amplify the v3 region of the 16S rRNA gene while including up to 200 unique barcode tags. The use of a 6 bp barcode tag, appended to the 5' end of the forward primer, uniquely identifies each sample during multiplex Illumina sequencing (*see Subheading 3.6*). In order to avoid bias the lowest possible number of PCR cycles should be used, which may require several PCR reactions (usually 3) to be pooled for each sample (*see Note 12* before starting).

1. Use 30–50 ng (or 100–200 ng for host-associated communities with high amounts of host DNA) of DNA in a 50 µL PCR amplification reaction containing 5 pmole of each v3F and v3R primers (*see Note 13*) with barcodes added to one primer and Illumina adaptors added to both primers [17], 1.5 mM MgCl₂, 0.2 mM dNTPs, 0.4 µg BSA, 1× PCR buffer and 1.25 units of Taq polymerase. Thermocycler conditions will vary slightly according to the Taq polymerase used but as a general guideline this should include activation of hotstart Taq (if using), then an initial denaturation step at 94 °C for 2 min, then 15–25 cycles of: denaturation at 94 °C for 30 s, annealing at 50 °C for 30 s, extension at 72 °C for 30 s; then one final extension at 72 °C for 10 min.
2. Use gel electrophoresis to determine if each PCR reaction produced a product approximately 300 bp in size by running 5 µL of each reaction on a 1.5–2% agarose gel.
3. Run pooled libraries of triplicate reactions on the Illumina MiSeq platform. Check with your sequencing facility for normalization procedures prior to pooling the library.

3.6 Illumina Sequencing

Illumina sequencing involves the attachment of single molecules to the flat surface of a flow cell, *in situ* amplification of the sequence via “bridging,” followed by the incorporation of fluorescently labeled reversible terminator deoxyribonucleotides [18]. These specialized representations of adenine, cytosine, guanine, and thymine are added, and a fluorescent image is taken as they are excited, and translated into a DNA sequence based on the unique color of each base’s fluorophore. Because of the 3' modifications made to the nucleotides, only one can be incorporated into any given sequence

at a time. After the image is taken, this termination chemistry is reversed, and another cycle of nucleotide addition ensues [18]. Using this technology, the Illumina MiSeq Personal Sequencer system allows for up to 25 million 2×300 bp paired-end reads per 56 h run. Multiplexing (or barcoding) [17] with short signature sequences (*see Note 13*) allows for multiple samples to be combined on a single run.

3.7 Sequence Analysis

Sequencing with Illumina sequencers, as well as other popular technologies, result in FASTQ formatted files which report each base pair in the sequence as well as the quality of the call. The goal of sequence analysis is to interpret this output to determine the taxonomic distribution of a given sample, and how the relative abundance of these organisms compares across a dataset.

Generally, this procedure involves quality control (including trimming and/or culling sequences for which the quality is poor), checking for chimeric sequences which can form as part of the PCR protocol, clustering into Operational Taxonomic Units (OTUs), and assigning a taxonomy to each OTU. Once this data processing is complete, it is possible to use this information to determine the distribution of diversity within a sample (alpha-diversity), or between samples (beta-diversity), as well as the determination of differential abundance of OTUs, genera, or phyla.

There are many commonly used software tools which can assist in this data processing and sequence analysis; these include, but are not limited to, Quantitative Insights into Microbial Ecology (QIIME) [19], mothur [20], and phyloseq [21].

4 Notes

1. Supplements promote growth of anaerobic bacteria. When culturing from cystic fibrosis sputum (or other sample types high in *Pseudomonas aeruginosa*) onto media + supplements, add 1 mL of stock of colistin (final concentration 10 mg/L) in order to suppress the growth of *P. aeruginosa*.
2. Autoclaving should be done in a steam autoclave at 121 °C and at least 15 psi for an appropriate amount of time. 500 mL of media can be autoclaved for 30 min to achieve complete sterility; however, smaller volumes and reagents sensitive to heat should be autoclaved for a shorter time.
3. Supplements to add: 500 µL of stock L-cysteine (final concentration 0.5 g/L), 1 mL of stock hemin (final concentration 10 mg/L), and 1 µL of stock Vitamin K (final concentration 1 mg/L).

4. Preparing laked blood: Mix 12.5 mL of sheep's blood with 12.5 mL of sterile H₂O and freeze overnight. Before use, warm laked blood to room temperature.
5. Guanidine thiocyanate is corrosive. Wear a mask when weighing out the powder.
6. 100 µL of sample can be plated on solid media in 100 × 15 mm Petri dishes. If larger plates are used then more sample is required.
7. 16S rRNA gene primer sequences are 8F: 5'-AGAGTTGA TCCTGGCTCAG and 926R: 5'-CCGTCAATTCTTTRA GTTT.
8. Changes to DNA isolation methods introduce a significant source of variation and standard protocols have been put forward by many groups [22]. It is, therefore, imperative to use a consistent isolation protocol on all samples in the same study. Samples can be processed at different times without batch effects as long as a consistent protocol is used and careful attention is taken to avoid contamination of low biomass samples, such as nasal swabs, with high biomass samples, such as sputum. Here, we propose the following DNA isolation protocol to maximize the recovery from Gram positive organisms such as the Bacilli.
9. For sputum, plate pools, and liquid samples use 0.2 g of 0.1 mm or 0.5 mm glass beads. For biopsies and other solid sample types use 0.2 g of both 2.8 mm ceramic and 0.1 mm glass beads.
10. For plate pools: this step may require the use of pipette tips whose tip has been cut back 0.5 cm with scissors before autoclaving. This provides a bit more space for highly viscous samples. For sputum samples, the addition of 300 µL should be completed by repeated passage through a 1 mL tuberculin tip syringe with 18 gauge needle.
11. The phenol-chloroform weakens the plastic and eppendorfs can crack at higher speeds.
12. Preparation of all solutions and reactions for PCR amplification of v3 16S rRNA gene tags should be done in a PCR Workstation (CanadaWide Scientific, #95-0438-01) to avoid DNA contamination from outside sources.
13. v3 16S rRNA gene Illumina primer sequences are F: 5'-caagca-gaagacggcatacgagat**CGTGAT**tgactggagttcagacgtgtctcc-gatctCCTACGGGAGGCAGCAG, where the lower case region is the Illumina adapter, the capitals in bold are the barcode, the lower case underlined region is the Illumina priming region, and the following capitals are the 341F of the 16S gene; and R: 5'-aatgatacggcgaccaccgagatctacactttccc-tacacgacgcttccgatctNNNNATTACCGCG GCTGCTGG,

where the lower case region is the Illumina adapter region, the lower case underlined region is the Illumina priming region, and the following capitals are the 518R of the 16S gene. There are over 100 compatible barcodes listed in [17].

Acknowledgments

Thank you to the members of the Surette laboratory as well as the clinicians and patients who have helped us improve the culturing conditions and techniques described within over the years.

References

- Man WH, de Steenhuijsen Piters WAA, Bogaert D (2017) The microbiota of the respiratory tract: gatekeeper to respiratory health. *Nat Rev Microbiol* 15(5):259–270
- Stearns JC, Davidson CJ, McKeon S, Whelan FJ, Fontes ME, Schryvers AB et al (2015) Culture and molecular-based profiles show shifts in bacterial communities of the upper respiratory tract that occur with age. *ISMEJ* 9:1246–1259
- Charlson ES, Bittinger K, Haas AR, Fitzgerald AS, Frank I, Yadav A et al (2011 Oct) Topographical continuity of bacterial populations in the healthy human respiratory tract. *Am J Respir Crit Care Med* 184(8):957–963
- Bassis CM, Tang AL, Young VB, Pynnonen MA (2014) The nasal cavity microbiota of healthy adults. *Microbiome* 2:27
- Bogaert D, Keijser B, Huse S, Rossen J, Veenhoven R, van Gils E et al (2011) Variability and diversity of nasopharyngeal microbiota in children: a metagenomic analysis. Semple M, editor. *PLoS One* 6(2):e17035
- Whelan FJ, Verschoor CP, Stearns JC, Rossi L, Johnstone J, Surette MG et al (2014) The loss of topography in the microbial communities of the upper respiratory tract in the elderly. *Ann Am Thorac Soc* 11(4):513–521
- Chotirmall SH, Burke CM (2015) Aging and the microbiome: implications for asthma in the elderly? *Expert Rev Respir Med* 9(2):125–128
- Dickson RP, Erb-Downward JR, Freeman CM, McCloskey L, Beck JM, Huffnagle GB et al (2015) Spatial variation in the healthy human lung microbiome and the adapted island model of lung biogeography. *Ann Am Thorac Soc* 12 (6):821–830
- Segal LN, Alekseyenko AV, Clemente JC, Kulkarni R, Wu B, Gao Z et al (2013) Enrichment of lung microbiome with supraglottic taxa is associated with increased pulmonary inflammation. *Microbiome* 1(1):19
- Dickson RP, Erb-Downward JR, Freeman CM, McCloskey L, Falkowski NR, Huffnagle GB et al (2017) Bacterial topography of the healthy human lower respiratory tract. *MBio* 8(1): e02287-16. <https://doi.org/10.1128/mBio.02287-16>
- Lau JT, Whelan FJ, Herath I, Lee CH, Collins SM, Bercik P et al (2016) Capturing the diversity of the human gut microbiota through culture-enriched molecular profiling. *Genome Med* 8(1):72
- Sibley CD, Grinwis ME, Field TR, Eshaghurshan CS, Faria MM, Dowd SE et al (2011) Culture enriched molecular profiling of the cystic fibrosis airway microbiome. *PLoS One* 6(7):11
- Browne HP, Forster SC, Anonye BO, Kumar N, Neville BA, Stares MD et al (2016) Culturing of “uncultivable” human microbiota reveals novel taxa and extensive sporulation. *Nature* 533(7604):543–546
- Gevers D, Knight R, Petrosino JF, Huang K, McGuire AL, Birren BW et al (2012) The human microbiome project: a community resource for the healthy human microbiome. *PLoS Biol* 10(8):e1001377
- Human Microbiome Project Consortium (2012) Structure, function and diversity of the healthy human microbiome. *Nature* 486 (7402):207–214
- Stearns JC, Lynch MDL, Senadheera DB, Tenenbaum HC, Goldberg MB, Cvitkovitch DG et al (2011) Bacterial biogeography of the human digestive tract. *Sci Rep* 1:170
- Bartram AK, Lynch MDJ, Stearns JC, Moreno-Hagelsieb G, Neufeld JD (2011) Generation of multimillion-sequence 16S rRNA gene libraries from complex microbial communities by assembling paired-end illumina reads. *Appl Environ Microbiol* 77(11):3846–3852

18. Bentley DR, Balasubramanian S, Swerdlow HP, Smith GP, Milton J, Brown CG et al (2008) Accurate whole human genome sequencing using reversible terminator chemistry. *Nature* 456(7218):53–59
19. Caporaso GJ, Kuczynski J, Stombaugh J, Bittinger K, Bushman FD, Costello EK et al (2010) QIIME allows analysis of high-throughput community sequencing data. *Nat Methods* 7:335–336
20. Schloss PD, Westcott SL, Ryabin T, Hall JR, Hartmann M, Hollister EB et al (2009) Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Appl Environ Microbiol* 75(23):7537–7541
21. McMurdie PJ, Holmes S (2013) Phyloseq: an R package for reproducible interactive analysis and graphics of microbiome census data. *PLoS One* 8(4):e61217
22. Kuczynski J, Lauber CL, Walters WA, Parfrey LW, Clemente JC, Gevers D et al (2011) Experimental and analytical tools for studying the human microbiome. *Nat Rev Genet* 13(1):47–58



Chapter 5

Methods and Strategies to Examine the Human Breastmilk Microbiome

Lauren LeMay-Nedjelski, Julia Copeland, Pauline W. Wang,
James Butcher, Sharon Unger, Alain Stintzi, and Deborah L. O'Connor

Abstract

It has recently been discovered that breastmilk is not sterile, but contains a vast array of microbes, known collectively as the breastmilk microbiome. The breastmilk microbiome field is in its infancy, but over the last decade, our understanding of the microbial communities that inhabit the human body has increased exponentially, due in large part to novel next-generation sequencing technologies. These culture-independent, high-throughput molecular technologies have allowed biologists to investigate the entirety of microbiota present in breastmilk, which was previously poorly known. These approaches are novel and the methodologies surrounding the exploration of the breastmilk microbiota remain in flux. The objectives of this chapter are to outline what is known thus far and detail the optimal methods and strategies to conducting a breastmilk microbiome study from subject recruitment and milk collection to DNA extraction, high-throughput sequencing and bioinformatics analyses.

Key words Breastmilk, Breastfeeding, Microbiome, Lactation, Bacteria, Gastrointestinal tract, DNA extraction, 16S rRNA gene, Illumina MiSeq, Bioinformatics

1 Introduction

Exclusive breastfeeding is the recommended dietary regime for infants during the first 6 months of life, with introduction of solid foods at this time and continued breastfeeding thereafter [1]. Breastmilk provides essential nutrients to newborns and contains bioactive components conferring immunoprotective properties [2]. Breastmilk has well-established benefits as compared to infant formulas in reducing morbidities such as gastrointestinal tract infections and necrotizing enterocolitis (NEC) in preterm infants, as well as reducing all-cause mortality [2–6]. Consumption of breastmilk has also been associated with a reduced risk of overweight and obesity, potentially due to the reduced protein content of breastmilk compared to formula and/or the reduced insulin response to feeding [2, 7, 8]. The bioactive components in

breastmilk such as lactoferrin, oligosaccharides, immunoglobulins, and insulin-like growth factor (IGF) are thought to be responsible for its benefits as these constituents can regulate both metabolic and immunologic programming in the infant, in both the short and long term [4, 8–11].

It has recently been discovered that breastmilk within the mammary gland is not sterile, but contains numerous microbes, in addition to the macronutrients, micronutrients, and bioactive components, known collectively as the breastmilk microbiome [11, 12]. Microbes found in breastmilk are not typically harmful, and are often both commensal and mutualistic [12]. One of the mechanisms whereby breastfeeding is now thought to confer important health benefits is through provision of a constant supply of microbes, which aid in the development of the community of microorganisms found within the infant gastrointestinal (GI) tract, the gut microbiome, as well as the infant's immune system and the naïve GI tract itself [11–15]. Emerging evidence suggests that failure to establish an abundant and diverse gut microbiome during the first few years of life may contribute to the development of allergies, autoimmune disease, asthma, obesity, and cardiovascular disease through host–microbiota interactions [16–19]. Minimal research has been conducted on the breastmilk microbiome and its composition and stability in the face of environmental influences, such as mode of delivery and maternal diet, remain unknown (Table 1). With that said, a few studies have shown that the same bacteria that reside in maternal breastmilk are found concurrently in the offspring's stool, displaying a transfer of microbiota from mother to infant [20, 21]. It is crucial to enrich our understanding of the composition and functionality of the breastmilk microbiome if we wish to further examine the infant gut microbiome. After mode of delivery, diet (breastmilk, if provided) is the next major postnatal exposure to ingested microbes, likely impacting infants' future health [6, 11, 12, 22, 23]. It is important to note that the microbiome also encompasses viruses, fungi, and archaea; however, for the purposes of this chapter, the focus will be exclusively on the study of the bacteria in the breastmilk microbiome [24].

1.1 Breastmilk Biogenesis

The human breast begins its maturation during the first trimester of pregnancy. Breast maturation is directly orchestrated by the rising levels of circulating lactogenic hormones, including estrogen, progesterone, and prolactin [25]. This hormonal combination, along with additional regulators such as placental lactogen and epidermal growth factor, triggers ductal branching, alveolar development, morphogenesis, as well as secretory differentiation [25]. In the second trimester of pregnancy, the steady surge of prolactin prompts the differentiation of mammary epithelial cells into lactocytes [10]. This stage, known as lactogenesis I, typically occurs at approximately 24 weeks gestation and may present with the first secretion of colostrum, the earliest form of milk produced [9]. Approximately

Table 1
Breastmilk microbiome studies published to date

Milk collected, time point, breast cleaned	Sample size (n=)	Study location	Methods	Summary of results	References
Mature milk, complete breast expression, three time points 1–2 weeks apart, iodine swab	16	Washington, and Idaho, USA	QIAamp DNA mini kit Hypervariable region: V1–V2 Primers: 27F and 338R 35 PCR cycles 454 pyrosequencing	– Breastmilk was collected at three time points; stability of microbiome was inconsistent and varied between women	[32]
Colostrum and mature milk, three time points (colostrum, 1 and 6 months postpartum), iodine swab	18	Turku, Finland	QIAamp DNA stool mini kit Hypervariable region: V1–V2 Primers: 27F and 533R 20 PCR cycles 454 pyrosequencing	– Breastmilk microbiome changed over the course of lactation (colostrum to 6 months postpartum) – Breastmilk from obese mothers was less diverse compared to normal weight mothers – Mothers who underwent vaginal births and nonselective C-sections had breastmilk microbiomes dissimilar from elective C-section mothers	[11]
Transitional milk and mature milk, foremilk, one time point, ≥day 6 postpartum sterile saline swab	39	Ontario, Canada	QIAamp DNA stool kit Hypervariable region: V6 Primers: 25 PCR cycles Illumina MiSeq	No statistically significant differences between preterm and term births, C-section (both elective and nonelective) and vaginal deliveries and infant sex were seen in the breastmilk microbiome	[31]

(continued)

Table 1
(continued)

Milk collected, time point, breast cleaned	Sample size (n=)	Study location	Methods	Summary of results	References
Mature milk, hindmilk, three sampling points, e.g., At days 3–6, 9–14, 25–30 postpartum, aseptic soap	7	Zurich, Switzerland	Fast DNA SPIN kit for soil Hypervariable region: V5–V6 Primers: 8F and 1391R 25 cycles PCR 454 pyrosequencing	<ul style="list-style-type: none"> <i>Bifidobacterium breve</i> was found in maternal stool, breastmilk and infant stool Butyrate-producing bacteria also shared between maternal stool and breastmilk Evidence for vertical transfer of obligate gut-associated anaerobes from mother to neonate via breastfeeding 	[20]
Mature milk, foremilk and/or hindmilk, one sample post partum, soap and water	10	California, USA (Cohort of Mexican-American women)	QIAamp ultraclean production pathogen mini kit Hypervariable region: V4 Primers: 515F and 806R 35 PCR cycles Illumina HiSeq 2500	<ul style="list-style-type: none"> <i>Streptococcus</i> and <i>Staphylococcus</i> [76] dominated breastmilk microbiome High BMI pre-pregnancy was correlated with lower <i>Streptococcus</i> abundance and higher microbial diversity in breastmilk 	
Colostrum, transitional milk, mature milk; 9 samples from day 2 to 6 months postpartum, full breast expression	21	Washington State University and University of Idaho	QIAamp DNA mini kit (Qiagen) Hypervariable region: V1–V3 Primers: 27F and 534R	<ul style="list-style-type: none"> Milk microbiota fairly stable over time Relative abundances of several taxa were associated with BMI, delivery mode, infant sex and maternal diet. Overweight and obese 	[77]

20 PCR cycles Illumina MiSeq	mothers = Mother's who were overweight and obese displayed an increased incidence of Granulicatella in their milk compared to mother's of a healthy body size <ul style="list-style-type: none"> - Saturated and monounsaturated fatty acids inversely associated with relative abundance of Corynebacterium - Total carbohydrates, disaccharides, and lactose = negatively associated with Firmicutes - Protein= positively correlated with the relative abundance of Gemella 	Mature milk samples (1 month postpartum), foremilk discarded, soap and sterile water and soaked in chlorhexidine <ul style="list-style-type: none"> 80 China (Beijing), South Africa (Cape town), Finland (southwestern area), Spain (Valencia, Mediterranean area) 	InviMag stool DNA kit and bead beating with FastPrep Hypervariable region: V4 Primers: 5'15F and 806R Illumina MiSeq	<ul style="list-style-type: none"> - Milk microbiota composition differed significantly based on geography - Vaginally delivered women- Spanish women = highest Bacteroidetes; Chinese women = highest Actinobacteria - Cesarean section = higher Proteobacteria in Spanish and South African women - Spanish and South African women = significantly higher bacterial genes mapped to lipid, amino acid and carbohydrate metabolism - Monounsaturated fatty acids in milk were negatively associated with Proteobacteria, while <i>Lactobacillus</i> was positively associated <p>[78]</p>
---------------------------------	---	---	---	--

(continued)

Table 1
(continued)

Milk collected, time point, breast cleaned	Sample size (n=)	Study location	Methods	Summary of results	References
Milk samples were collected within 5 days postpartum (colostrum), between 6 and 15 days (transitional milk) and after 15 days (mature milk), manual expression (discarded first drops) soap and sterile water and soaked in chlorhexidine	21	Valencia, Spain	- MasterPure complete DNA and RNA purification kit - qPCR targeted to single-copy gene <i>fusA</i> Primers: 8F and 785R 20 PCR cycles 454 sequencing	- High interindividual variability [79] - Predominant genera = <i>Staphylococcus</i> , <i>Streptococcus</i> , <i>Pseudomonas</i> - Median bacterial load was 10^6 / mL - No correlation between bacterial load and amount of immune cells in human milk	[79]
Manual or electric breastpump	107	Los Angeles, California; St. Petersburg, Florida	BiOstic Bacteremia DNA isolation kit (MOBIO) Hypervariable region: V4 Primers: 340F, 806R Illumina sequencing	- Proteobacteria was the predominant phyla, followed by Firmicutes - Alpha diversity did not change over the first year of life - Beta diversity increased between mother's over the first 6 months postpartum, then reduced	[80]
Samples collected at 1, 3, 6, and 12 weeks postpartum, sterile gloves worn, breast cleaned with chlorhexidine wipes, first few drops of milk discarded	10		QIAamp DNA stool mini kit Hypervariable region: V3-V4 30 cycles PCR Illumina MiSeq	207 unique bacterial genera detected Primary phyla = Proteobacteria, Firmicutes Primary genera = <i>Staphylococcus</i> , <i>Pseudomonas</i> 12 core genera represented 81% of the microbiota relative abundance	[81]

Samples collected at 1 month postpartum manually in sterile tube (first 500 µL discarded), nipples and mammary areola cleaned with soap and sterile water and soaked in chlorhexidine	10	Valencia, Spain	QIAamp DNA stool mini kit Hypervariable region: V1–V3 20 PCR cycles Primers: 27F and 533R	- PCoA plots showed separated milk microbiome from mothers with vaginal delivery vs. C-section - Higher bacterial diversity and richness were found in samples from vaginal deliveries [82]
Manual expression using a sterile glove from each breast, nipples and areola cleaned with 70% ethyl alcohol, first drops of milk discarded	18 controls and 32 women with mastitis	Gujarat, India	QIAamp stool DNA fast mini kit Hypervariable region: V2–V3 25 cycles PCR Primers 101F and 518R	- Women with mastitis had lower microbial diversity, increased abundance of potential pathogens, fewer obligate anaerobes and greater aerotolerant bacteria than controls [83]

48–72 h following parturition, progesterone decreases and prolactin increases, leading to full secretory activation or lactogenesis II, which involves producing transitional milk before finally producing mature breastmilk (lactogenesis III) [10, 25].

Colostrum is higher in protein, lower in carbohydrates, and contains large quantities of immune-modulating factors such as immunoglobulins, lactoferrin, and oligosaccharides as well as additional active immune cells like leukocytes [9, 25–27]. Human milk oligosaccharides (HMOs), for example, also function as prebiotics as they are resistant to digestion in the stomach and the small intestine and thus provide an important source of fuel for beneficial commensal bacteria (e.g., *Bifidobacteria* and *Lactobacillus* species) in the large intestine [12]. Colostrum also contains growth factors and mitogen-containing compounds thought to be involved in the development of not only the newborn's gastrointestinal tract but also immune development and hematopoiesis [28, 29]. Over the first week following parturition, transitional and then mature milk is secreted, which contains overall more calories than colostrum, as well as higher quantities of fat and lactose. Mature milk continues to contain bioactive components, such as growth and immunological factors, albeit often in lower concentrations compared to colostrum [9, 10, 30].

1.2 Breastmilk Microbiome

The most predominant phylum seen in the breastmilk microbiome is Firmicutes, followed by Proteobacteria [11, 31, 32]. The origin of these bacteria in breastmilk remains controversial, however, several theories have been proposed to explain their presence. In brief, the neonate is primarily colonized with microbiota during the birthing process, either through Cesarean delivery (C-section) or vaginal delivery, and the bacterial species that colonize the infant differ significantly depending on the mode of delivery [33, 34]. During suckling, breastmilk can flow retrograde from the infant's newly colonized oral cavity back into the mammary ducts, transferring bacteria from the infant's mouth into the mother's mammary duct [35]. Therefore, mode of delivery, which colonizes the oral microbiome, may impact the breastmilk microbiome via this backwash mechanism [11].

A second theory involves dendritic cells (DCs), antigen-presenting cells of the immune system, which are able to take up whole commensal bacteria from the maternal gut lumen using their cellular projections known as dendrites [20, 36, 37]. These DCs can then house a small quantity of these live nonpathogenic bacteria for several days while they migrate to new sites in the body, such as the lactating mammary gland, and unload the bacteria [38–40]. Lastly, breast tissue itself has been shown to have its own microbiota, and the major bacterial phyla present in the breast tissue match those found in breastmilk, suggesting a transfer of bacteria from the glandular and/or adipose tissue to the mammary ducts [6, 31].

The origin of the microorganisms present in breastmilk warrants further investigation, as it is crucial to understand how they colonize the mammary duct and if the breast selects for the bacterial composition.

2 Study Design and Methods

2.1 Experimental Design

Conducting a clinical study to examine the breastmilk microbiome is a complex, labor-intensive process that requires rigorous planning and meticulous attention to detail. In the initial stages, a clear research question or primary objective(s) and hypothesis(es) to be tested must be formulated to guide the development of the research proposal, which, in turn, will dictate the procedures for breastmilk sample collection, measuring the health outcomes of interest and dealing with confounding variables. Typical research designs in human milk and lactation research include randomized controlled trials (blinded or unblinded), prospective/retrospective cohort studies, and case-control and cross-sectional studies (Table 2). Regardless of the study design chosen, it needs to be

Table 2
Study designs frequently used in human milk research

Study design		Strengths	Limitations
<i>Observational studies</i>			
Cohort		Strongest observational study Can study multiple outcomes Can establish incidence Determines associations	Selection bias and confounding factors Risk of recall bias increased if retrospective design. No randomization to exposures Time consuming Loss to follow-up
<i>Experimental studies</i>			
Case-control		Helpful for identifying risk factors for rare conditions Less time consuming and expensive Can study multiple risk factors Determines associations	Recall bias due to retrospective nature Challenging to find a control group that is similar to cases Temporal sequence of events may be difficult to ascertain
Cross-sectional		Can study multiple outcomes Measures prevalence Fast Less expensive	Weakest observational design Not suitable for rare conditions Temporal sequence of events may be difficult to ascertain
Randomized controlled trials		Gold standard research design Random allocation should ensure groups to be compared are equal in both known and unknown confounding factors Enables determination of cause and effect	Very expensive Ethically problematic at times Frequent short duration makes it challenging to examine chronic disease endpoints

realistic, feasible, and answer the question at hand. For example, a randomized controlled trial (RCT) is the gold standard of clinical research. However, if the research question was to examine the benefits of the human milk microbiome on infant gut development compared to formula-fed controls, this design would be inappropriate as it would be unethical to randomize women to breastfeed or formula-feed their infants. Alternatively, prospective cohort studies allow for preplanning of breastmilk sample and data collection, but careful attention must be paid to confounding variables positively associated with breastfeeding intensity such as maternal education and socioeconomic status, which may independently impact the outcome variables of interest.

After selecting the study design, the sample size (this varies depending on primary outcome and effect size) and the milk collection timepoint(s), must be determined. For a lactation study, it is important to recruit woman during pregnancy, and ensure that the eligibility criteria for enrollment include the intent to breastfeed. Depending on the goal of the study, women with conditions that could affect the breastmilk microbiome, such as mastitis, may be excluded, or at the very least, noted, to ascertain if these conditions impact the milk microbiome. Antibiotics have the potential to significantly perturb all microbiomes of the human body, and so individuals with antibiotic usage within 6 months of sampling may also need to be excluded from microbiome studies [41–45]. Lastly attrition is a unique issue of lactation studies as women wean their infants at different times. The timing of weaning varies by geographical location and is also impacted by various sociodemographic factors [46]. Unless care is taken, the study may be underpowered due to unplanned attrition and unintended bias could be introduced into study findings as women with higher socioeconomic status, education, and social support breastfeed longer [47].

2.2 Breastmilk Collection

Breastmilk is a bodily fluid and therefore must be collected and stored in a manner that reduces the risk of microbial contamination from external sources, particularly important in microbiome studies. As breastmilk has been shown to contain harmful pathogens such as human immunodeficiency virus (HIV), cytomegalovirus, and zika virus, equivalent to blood, it meets laboratory biosafety 2 criteria and appropriate precautions must be followed when handled [48]. There is controversy in the field whether or not the breast should be cleaned with either an iodine swab or a saline solution prior to breastmilk expression for milk microbiome studies. Proponents for cleaning advocate that sanitizing the breast ensures that the only bacteria obtained are from the milk microbiome and not from the skin microbiome surrounding the nipple. However, in everyday practice, mothers are neither recommended to, nor wash their breasts, before every feeding and so not washing

the breast before breastmilk expression is likely a better reflection of the microbes ingested by the infant as a result of breastfeeding.

Sample collection in breastmilk microbiome studies typically involves a complete breastmilk expression with a sterile electric breast pump. Breastmilk is dynamic and it is ideal to carry out a complete breast expression in one sitting, even if the volume of breastmilk required for the study is small. Milk from the complete breast expression should be inverted several times in a leak-proof container, ideally while still warm, before being aliquoted out into smaller vials to ensure a consistent sample composition. If nurses or lactation consultants assist the subjects in the study, it is important that gloves are worn to prevent sample contamination with foreign microbes.

The human milk microbiome studies conducted to date have collected breastmilk samples at varying time points during the postpartum period, often occurring after 1-month postpartum. This approach ensures that the mother has time to recuperate from childbirth and there is an adequate volume of milk obtained as there is a higher likelihood that the sampled milk will be mature milk [49]. Sampling breastmilk anywhere from 1 to 6 months has been reported, but usually not past 6 months as many women begin weaning at this time [11, 31, 32, 49, 50]. Sampling the breastmilk microbiome at only one time point during the study may not provide a complete picture of the composition of the microbial communities present in breastmilk [45]. The stability of the breastmilk microbiome is also unknown, and time-series data would help to illuminate how much the breastmilk microbiome changes over time, if at all, especially following any perturbation [45]. For this reason, repeated sampling over a few months postpartum would provide a more comprehensive picture regarding the dynamics of the breastmilk microbiome. Lastly, study protocols should standardize the time of day all mothers provide breastmilk samples, as well as when their last expression/feeding took place prior to the study milk expression time.

2.3 Breastmilk Storage

Once the breastmilk has been expressed, it is optimal to immediately pellet the milk by centrifugation at $10,000 \times g$ for 10 min in 1–2 mL aliquots, remove the fatty layer and leave approximately 100 μ L liquid for storage at -80°C . If this is not feasible, well-mixed milk can simply be aliquoted into 1–2 mL labeled containers or tubes and immediately stored at -80°C . Rapid freezing to -80°C after breast expression is ideal to preserve microbial communities [51]. It is crucial that the number of freeze-thaw cycles is minimized as that may have an impact on the microbial community since certain bacteria will lyse in the thaw cycle [52]. Alternatively, a process known as lyophilization, or freeze drying of milk samples followed by storage at room temperature, is also a viable option for long-term storage of breastmilk samples [53, 54]. Lyophilization

has been shown to only marginally alter the microbial composition of breastmilk similar to storage at -80°C , it is superior to -20°C storage and thus remains an option for storing human milk [54].

3 Materials to Identify Taxonomic Units in Breastmilk

3.1 DNA Extraction from Breastmilk

1. 5–200 mg sample material (1–2 mL breastmilk).
2. NucleoSpin Food kit (Macherey-Nagel) (*see Note 1*).
3. 96–100% ethanol.
4. 1.5 mL microcentrifuge tubes.
5. Manual pipettors ranging from 20 to 1000 μL .
6. Disposable pipette tips.
7. Centrifuge for microcentrifuge tubes.
8. Vortex mixer.
9. Heating block for incubation at 65°C .
10. Appropriate personal protective equipment (lab coat, gloves, goggles).
11. Optional: Gel extraction kit (e.g., Macherey-Nagel PCR cleanup gel extraction kit).

3.2 Amplification of the V4 Hypervariable Region of the 16S rRNA Gene (See Note 2)

1. DNA extracted from breastmilk.
2. KAPA2G Robust HotStart Ready Mix (KAPA Biosystems).
3. 10 μM 515FV4 forward primer [55] (Table 3; *see Notes 3 and 4*).
4. 10 μM 806RV4 reverse primer [55] (Table 3; *see Notes 3 and 4*).
5. 100 μM Read 1 sequencing primer (Table 3).
6. 100 μM Read 2 sequencing primer (Table 3).
7. 100 μM Index sequencing primer (Table 3).
8. Molecular grade Agarose.
9. TBE buffer (10 \times buffer diluted to 1 \times): Tris base (1000 mM), boric acid (1000 mM), EDTA (20 mM). Dilute to 1 \times prior to use.
10. TAE buffer (50 \times stock solution diluted to 1 \times): Tris base (40 mM), acetic acid (2 mM), EDTA (20 mM). Dilute to 1 \times prior to use.
11. Agencourt® AMPure® XP (Beckman Coulter).
12. Sterile water.
13. Sterile PCR tubes or plates.
14. Thermal cycler.
15. MiSeq sequencer (Illumina).

Table 3**Primers sequences used for V4 amplification and sequencing [55]**

Primer	Sequence (5' – 3')
515F V4 forward	AATGATACGGCGACCACCGAGATCTACACTATGGTAATTGTGT GCCAGCMGCCGCGTAA
806R V4 reverse (+ barcode ^a)	CAAGCAGAACGGCATACGAGATXXXXXXXXXXXXAGTCAG TCAGCCGACTACHVGGGTWTCTAAT
Read 1 sequencing primer	TATGGTAATTGTGTGYCAGCMGCCGCGTAA
Read 2 sequencing primer	AGTCAGCCAGCCGGACTACNVGGGTWTCTAAT
Index sequencing primer	AATGATACGGCGACCACCGAGATCTACACGCT

^aThe barcode is indicated in the 806R primer by a 12mer of X bases

16. Fluorometric method for quantifying concentration of DNA (i.e., Quant-iT PicoGreen® dsDNA Assay kit or Qubit® High Sensitivity dsDNA kit [Thermo Life Sciences]).
17. Disposable lab equipment such as 1.5 mL microcentrifuge tubes and disposable pipette tips.
18. Manual pipettors ranging from 20 to 1000 µL.
19. Agilent Bioanalyzer 2100 and DNA1000 chips for final library assessment and quantitation (Agilent Technologies).

3.3 Bioinformatics Analyses

1. USEARCH version 8.1 (32-bit version, maximum 4Gb RAM), accessed through Linux, Mac OSX, or Windows.
2. Chimera database: Ribosomal Database Project (RDP) 16S gold database, derived from the RDP training set version 9, accessed through USEARCH (USEARCH website).
3. Taxonomy database: UPARSE compatible with RDP database version 15 (USEARCH website).
4. MacQIIME 1.9.1-20150604 (default scripts).

4 Methods to Identify Taxonomic Units in Breastmilk

4.1 DNA Extraction from Breastmilk

1. Thaw 1 mL of breastmilk on ice if previously frozen at –80 °C.
2. Centrifuge at 10,000 × g for 10 min to separate out the fat layer.
3. Pipette off the top fat layer and the majority of the supernatant, leaving behind approximately 100 µL of liquid, and resuspend the pellet.
4. Transfer the resuspended pellet into a new tube containing 550 µL of lysis buffer preheated to 65 °C from the Nucleospin

Food genomic DNA extraction kit. Vortex briefly, add 10 µL of the supplied Proteinase K, and invert the tube gently to mix.

5. Incubate this mixture at 65 °C for 120 min (*see Note 5*).
6. Transfer this mixture into a clean microcentrifuge tube and add equivalent volumes of Buffer C4 and 96–100% ethanol. Vortex this mixture for 30 s.
7. Place one NucleoSpin Food Column into a collection tube and add 700 µL of the mixture to the column. Centrifuge for 1 min at 11,000 × g and discard the flow-through. Repeat until the entire sample volume has been passed through the column.
8. Pipette 400 µL Buffer CQW into the NucleoSpin Food Column. Centrifuge for 1 min at 11,000 × g and discard flow-through.
9. Pipette 700 µL Buffer C5 into the NucleoSpin Food Column. Centrifuge for 1 min at 11,000 × g and discard flow-through.
10. Pipette 200 µL Buffer C5 into the NucleoSpin Food Column and centrifuge for 2 min at 11,000 × g to completely remove C5.
11. Place the NucleoSpin Food Column into a new 1.5 mL microcentrifuge and pipette 30 µL of elution buffer (preheated to 70 °C) onto the center of the membrane. Incubate for 5 min at room temperature (18–25 °C) and centrifuge for 1 min at 11,000 × g to elute DNA (*see Note 6*).
12. Lastly, run a negative extraction control along with the breast-milk samples (*see Note 7*).

4.2 Amplification of the V4 Hypervariable Region of the 16S rRNA Gene

1. PCR reactions should be set up following the manufacturer's recommendations including 12.5 µL of KAPA2G Robust Hot-Start ReadyMix, 1.5 µL of 10 µM forward and 1.5 µL of 10 µM reverse primer, 3.5 µL of sterile water, and 6 µL of DNA (or a volume appropriate for amplification, plus sterile water up to 25 µL final volume) (*see Notes 3, 4, and 8*).
2. Amplify the V4 region by cycling the reaction at 95 °C for 3 min, 25–30 cycles of 95 °C for 15 s, 50 °C for 15 s, and 72 °C for 15 s, followed by a 5 min 72 °C extension. Cycle number must also be adjusted based on DNA concentration (*see Notes 9 and 10*).
3. It is highly recommended to perform all amplifications in triplicate and electrophorese the resulting amplicons on a 1% TBE agarose gel to check for proper amplification (amplicon size ~390 bp). A negative control lacking template DNA should be amplified alongside the samples to check for contamination and a positive control with DNA from a known bacterial species should also be included to check for successful amplification (*see Note 11*).

4. If bands of the same size and intensity are observed in each triplicate, samples are pooled. The replicate pools can then be quantified and subsequently combined in equal amounts to generate the pooled sequence library (*see Note 12*).
5. (Optional step): If multiple PCR bands are present, this can indicate cross-contamination with human mitochondrial DNA. To purify the bacterial V4 amplicons electrophorese your pooled libraries on a 2% agarose TAE gel, cut the desired 390 bp band from the gel and purify using a gel purification kit (e.g., the Macherey-Nagel PCR clean-up Gel extraction kit) (*see Note 13*). Quantify gel extracted libraries and pool with other replicates.
6. Further purify the pooled library with AMPure XP beads by adding 0.8× volume of beads to 1× volume of library DNA and follow the manufacturer's standard protocol to clean and elute the final product.
7. Quantify the purified library using a fluorometric kit or Agilent DNA1000 Bioanalyzer chip.
8. Load quantified library on an Illumina MiSeq according to the manufacturer's instructions. Sequence using the Miseq-V2-300 cycle chemistry to generate 150 PE reads (*see Note 14*). It is recommended to always include a mock community when characterizing the microbiota of breastmilk samples (*see Note 15*).

4.3 Bioinformatics Analyses of the Bacterial Microbiome

The UPARSE pipeline, available through USEARCH can be used for sequence analysis (*see Note 16*). The pipeline filters out low-quality sequences, assembles the paired-end reads into a single V4 amplicon, clusters sequences into Operational Taxonomic Units (OTUs) at 97% similarity and assigns a taxonomy to these OTUs [56, 57]. Default settings were used in the following example unless otherwise specified. The parameters written in the pipeline can be altered to suit one's specific needs.

1. Assemble the raw paired-end sequences using USEARCH and the fastq_mergepairs option and the parameter -fastq_merge_maxee = 1.0. This indicates a maximum expected error value of 1, recommended by USEARCH [56].
2. Filter the merged fastq files with the -fastq_filter option and the parameter fastq_maxee = 0.5 (*see Note 17*).
3. Filter out sequences shorter than 225 base pairs using the -fastq_filter option and the parameter -fastq_minlen 225 (*see Note 18*).
4. De-rePLICATE sequencing using the -derep_full length command.
5. Sort dereplicated sequences to remove singlettons using the -sortbyysize command.

6. Cluster sequences into operational taxonomic units (OTUs) at 97% identity using the `-cluster_otus` command and set `id = 0.97` (*see Note 19*).
7. Detect and remove chimeras using a referenced based method and the `-uchime2_ref` command. The Ribosomal Database Project (RDP) 16S gold database, accessed through the USEARCH website, is used to search for chimeric sequences in the OTUs [58]. The parameter `mode = high confidence` is set to minimize the number of false positive chimeras detected.
8. Map assembled sequences back at 97% identity to the chimera-free OTU sequences using the `-usearch_global` command. The assembled sequences that were originally input into the de-replication step are queried against the chimera filtered OTU file as the database and an `id = 0.97` is set.
9. Execute taxonomic assignment using the `-utax` command and the USEARCH compatible RDP database version 15 as the reference. A minimum confidence cutoff of 0.8 is set with the parameter `-utax_cutoff = 0.8`.
10. Align OTU fasta sequences using PyNast accessed through the QIIME python script `align_seqs.py` [55].
11. Construct a phylogenetic tree from the aligned sequence data using FastTree with the QIIME python script `make_phylogeny.py` [59]. This phylogenetic tree is used for downstream analysis in QIIME, such as weighted or unweighted UniFrac beta diversity comparisons.
12. Convert the mapped sequences and the taxonomic information to a tab-delimited format using the python script `uc2otu-tab.py` available through USEARCH. This file can be converted into a BIOM format file using QIIME and the `biom` command `biom-convert` [55]. BIOM format files can be further analyzed in QIIME for compositional and diversity analyses.
13. Remove low relative abundance (RA) OTUs (<0.005%) from the OTU table using the QIIME python script `filter_otus_from_ottu_table.py` and the parameter `-min_count_fraction = 0.00005` [60] (*see Note 20*).
14. The resulting OTU table can be rarefied to normalize the number of sequences per samples. The rarefaction level can be determined by rarefaction curves created through QIIME using the `multiple_rarefactions.py`, `alpha_diversities.py`, `collate_alpha.py`, and `make_rarefaction_plots.py` commands (*see Note 21*).
15. Relative abundances of the community members can be determined using the rarefied data, summarized at each taxonomic level using the `summarize_taxa.py` command through QIIME.

16. Alpha and beta diversity are also determined using the rarefied OTU data in QIIME and the alpha_diversity.py and beta_diversity.py commands [55].
17. Visualize beta diversity distances using principal coordinate analysis (PCoA) in EMPERor [55, 61].
18. Following microbiome bioinformatics analyses, one may examine associations between independent variables (metadata) and dependent variables (the microbial data) to identify prognostic independent variables. This could be accomplished with the use of generalized linear mixed models (GLMM) in SAS/R, in which the dependent variables would include the proportions of select OTUs or groups of bacteria (i.e., Phyla or Genera), and the independent variables would include categorical metadata, such as mode of delivery, maternal body mass index, or other factors of maternal health.

5 Notes

1. Two commonly used DNA extraction kits for microbiome research were tested: the MoBio Power Soil DNA Isolation kit (MoBio) and the DNeasy Blood and Tissue kit (Qiagen). These kits perform well for DNA extraction from many other human biological samples, such as skin and stool samples, with slight modifications based on sample type [62, 63]. When tested with breastmilk, we found that these kits were not effective at removing the high levels of fat. This led to residual fats observed in the extracted DNA, low-quality DNA, and difficulties with downstream processing. Instead, we tested the Macherey-Nagel NucleoSpin Food kit as it optimized for DNA extraction from high protein and high fat foods such as cheese. The combination of this kit and an upfront centrifugation step to separate the fat from the rest of the sample resulted in high-quality DNA.
2. Due to the current limitations of DNA-sequencing technology, it is impractical to sequence the full 16S rRNA gene using a high-throughput next-generation platform. The V4 region of the 16S rRNA gene was selected for sequencing because as a single hypervariable region, it most closely recapitulates the topology of a full-length 16S rRNA gene sequence phylogenetic tree, while being the appropriate length for current technologies [64].
3. Forward, reverse, or both forward and reverse primers can be barcoded for multiplexing. This is achieved by synthesizing primers targeting the V4 region that are flanked by error-proof Golay sequence barcodes. The barcodes are typically

8–12 bp, and contain combinations of base pairs that allow barcode sequences to remain unique even in the event of incorrect base pair calls (number of tolerated errors depends on the length of the barcode). These primers also contain the Illumina adapter sequence, a 10-base pad to prevent hairpin formation, and a 2-base linker that is not complementary to the 16S rRNA gene region. The use of these customized primers requires the addition of sequencing primers to the Illumina MiSeq cartridge [65].

4. *Bifidobacterium* species (sp.), especially in culture-based studies, have been shown to be important colonizers of the infant gut and have also been found in breastmilk; however, many have struggled to detect notable quantities of *Bifidobacterium* sp. in breastmilk [66, 67]. The challenges regarding *Bifidobacterium* sp. sequencing have been postulated to be the result of specific primers employed [68–70]. The bacterial 27F/338R primer set has been shown to be biased against the amplification of Bifidobacterial 16S rRNA genes, and as such should be avoided when examining the breastmilk microbiome [70]. Instead, primers that have been shown to amplify *Bifidobacteria* for 16S gene sequencing of the breastmilk microbiome include 27F/533R and 515F/806R [11, 70, 71].
5. This is a modification of the standard protocol which uses a 30 min incubation. In the “Troubleshooting” guide of the protocol it suggests 1–2 h of incubation in the lysis buffer to improve cell lysis.
6. The recommended elution volume for the kit is 100 µL; however the concentration of microbial DNA in breastmilk is usually low. To retain a concentration high enough for successful PCR, a lower elution volume of 30 µL was used.
7. It is recommended to run a negative extraction control (sterile water passed through the extraction kit) to identify true breastmilk microbiota from contaminating microbial DNA in the extraction kit.
8. The volume of DNA and water depends on the concentration of DNA, so these volumes are simply guidelines and will need to be adjusted. The amount of input DNA can be varied depending upon the load of bacterial versus human cells per sample. On average, 6 µL of breastmilk DNA are required to amplify the breastmilk microbiome. However, the DNA volume can range from 1 to 9.5 µL.
9. The PCR conditions described are optimized for KAPA2G Robust HotStart ReadyMix polymerase mixture. This mixture contains a very robust polymerase with high sensitivity and tolerates the presence of common inhibitors. Other high fidelity DNA polymerases may be used instead as long as the

reaction conditions are optimized. The volume of DNA required for input into the PCR reaction and the PCR cycles required to create a library with sufficient concentration for purification and sequencing requires optimization.

10. Even with extraction optimization, the yield of microbial DNA from breastmilk is often much lower than other sample types and requires many cycles of polymerase chain reaction (PCR) to create enough product for sequencing. Cycle number must also be adjusted based on bacterial DNA concentration. Twenty-five to thirty cycles is the typical range required for breastmilk samples, but specific sample optimization may be required to determine the ideal cycle number without over-amplifying and biasing the resulting library. This is determined by identifying the minimum cycle number where amplicons appear as strong, clear bands on an agarose gel.
11. A negative (PCR master mix, no template DNA) and a positive (PCR master mix, known single bacterial species genomic DNA template) control are performed alongside the samples. It is recommended to sequence a few negative (non-template) controls alongside to better understand the taxa that may be present in the sequence data due to low-level contamination resulting from a high PCR cycle number. For every PCR, a positive band should be approximately 390 bp in size and the negative control should not produce any visible amplified product. If a band is observed in the negative control, but the PCR cycle number cannot be reduced because of loss of the desired target band, a few of the negative controls should be sequenced along with the samples. To check the success of the PCR amplification, the samples are run on a 1% TBE gel.
12. Triplicate PCRs and sample pooling are performed to reduce sample amplification biases that may occur in single samples. Quantitation is performed using either the Picogreen or Qubit, both of which are highly sensitive fluorometric quantitation assays. The Quant-iT PicoGreen® dsDNA Assay kit is particularly suited to high-throughput quantitation.
13. In breastmilk samples, there is often a large proportion of human DNA extracted with the target bacterial DNA. The V4 primers, especially in situations with large amount of human DNA present, can cross-amplify with human mitochondria. The V4 amplicon with sequencing adapters and barcodes should be approximately 390 bp (www.earthmicrobiome.org), whereas cross-amplified mitochondrial DNA will produce a band approximately 300 bp in size. This cross-amplification requires additional purification of the target bacterial band prior to sequencing, which is often not required for other human microbiome sample types that have a much

higher relative abundance of bacterial to human DNA. A 2% agarose gel is used to allow for clear separation of the bacterial amplicons from the cross-amplified mitochondria. TAE gels are typically used in this stage, as opposed to TBE, because of the potential interference of boric acid in TBE with sequencing.

14. The MiSeq platform was selected and a 150×2 (300 bp) paired-end run was performed. Based on the number of bar-codes available and samples in the dataset, alternative higher-throughput Illumina platforms can be used (i.e., HiSeq, Next-Seq) as long as the read lengths are sufficient to generate overlapping paired-end reads. Choosing a higher-throughput instrument is determined based on the desired number of sequences per sample. This can be determined through rarefaction alpha diversity analysis of a preliminary subset of the data before sequencing all samples.
15. A mock community is a known mixture of microbial cells or nucleic acids compiled *in vitro* to be used as a control in microbiome analyses [72]. The mock community can be used to reveal the effect of sequencing error rates on the number of operational taxonomic units (OTUs) [73]. This is an effective method since the mock community composition is known, and so divergence from the expected illuminates the quality of the sequencing run [74].
16. This method, following the standard protocol and default parameters unless otherwise specified, was selected because it was shown to be superior in detecting correct bases in artificial microbial communities compared to other commonly used pipelines [57]. However multiple analysis methods were not tested for the breast milk microbiome specifically, and further optimization when analyzing this community could be possible.
17. This is an even more strict quality requirement than in the original merging step. Individual base quality scores give the probability of a single location having an error, and thus the overall probability of a read containing an error can be calculated from these values. The fastq_maxee option removes reads that have greater than the specified threshold of expected errors.
18. After removing the adapter and primer sequences, the assembled V4 amplicon is on average 253 bp long. Filtering out sequences less than 225 bp long allows sequences with naturally shorter V4 regions to be retained, but those present due to sequencing error or nonbacterial amplicons will be removed. As mentioned, the breast milk microbiome often contains a high proportion of human mitochondrial DNA, resulting in the increased chance of sequencing mitochondrial amplicons

that are approximately 100 bp shorter than the bacterial amplicons of interest. These should therefore be removed using the 225 bp cutoff.

19. The UPARSE pipeline used a de novo OTU clustering command that employs distance-based greedy clustering (DGC) and abundance-based greedy clustering (AGC). De novo clustering methods do not rely on a reference for clustering, allowing for the discovery of previously uncharacterized OTUs, and DGC/AGC methods performed more accurately and reproducibly, compared to reference-based methods [75].
20. The removal of low abundance OTUs from the whole OTU table, along with upfront quality filtering, are the key steps in eliminating erroneous OTU sequences [60]. The exact settings used will depend on each dataset and thus these parameters will need to be defined for each experimental setup.
21. The minimum depth for rarefaction is determined by the curve. Ideally, the depth should be selected from a region of the curve that shows stable levels of alpha diversity with increasing sequencing depth. This region will be observed as the curve reaches an asymptote when rarefaction depth vs. diversity score is plotted.

References

1. WHO (2014) World Health Organization: Breastfeeding [Online]. <http://www.who.int/topics/breastfeeding/en/>
2. Victoria CG, Bahl R, Barros AJD et al (2016) Breastfeeding in the 21st century: epidemiology, mechanisms and lifelong effect. *Lancet* 387:475–490
3. Herrmann K, Carroll K (2014) An exclusively human milk diet reduces necrotizing enterocolitis. *Breastfeed Med* 9(4):184–190
4. Horta BL, Victoria CG (2013) Short-term effects of breastfeeding: a systematic review of the benefits of breastfeeding on diarrhoea and pneumonia mortality. World Health Organization (WHO), Geneva
5. Rollins NC, Ndirangu J, Bland RM et al (2013) Exclusive breastfeeding, diarrhoeal morbidity and all-cause mortality of HIV-infected and HIV uninfected mothers: a intervention cohort study in KwaZulu Natal, South Africa. *PLoS One* 8(12):e81307
6. Ward TL, Hosid S, Ioshikhes I et al (2013) Human milk metagenome: a functional capacity analysis. *BMC Microbiol* 13(116):1–12
7. Scholtens S, Brunekreef B, Smit HA et al (2008) Do differences in childhood diet explain the reduced overweight risk in breastfed children? *Obesity* 16:2498–2503
8. Horta BL, de Mola CL, Victora CG (2015) Long-term consequences of breastfeeding on cholesterol, obesity, systolic blood pressure, and type-2 diabetes: systematic review and meta-analysis. *Acta Paediatr Suppl* 104:30–37
9. Hassiotou F, Geddes DT, Hartmann PE (2013) Cells in human milk: state of the science. *J Hum Lact* 29(2):171–182
10. Lawrence RA, Lawrence RM (2016) Breastfeeding: a guide for the medical profession, 8th edition. Elsevier, Saunders
11. Cabrera-Rubio R, Collado MC, Laitinen K et al (2012) The human milk microbiome changes over lactation and is shaped by maternal weight and mode of delivery. *Am J Clin Nutr* 96(3):544–551
12. Fernandez L, Langa S, Martin V et al (2013) The human milk microbiota: origin and potential roles in health and disease. *Pharmacol Res* 69(1):1–10
13. Jakobsson HE, Abrahamsson TR, Jenmalm MC et al (2014) Decreased gut microbiota diversity, delayed Bacteroidetes colonization and reduced Th1 responses in infants delivered by caesarean section. *Gut* 63:559–566

14. LeBouder E, Rey-Nores JE, Raby AC et al (2006) Modulation of neonatal microbial recognition: TLR-mediated innate immune responses are specifically and differentially modulated by human milk. *J Immunol* 176:3742–3752
15. Stockinger S, Hornef MW, Chassan C (2011) Establishment of intestinal homeostasis during the neonatal period. *Cell Mol Life Sci* 68:3699–3712
16. Candela M, Rampelli S, Turroni S et al (2012) Unbalance of intestinal microbiota in atopic children. *BMC Microbiol* 12:1–9
17. Kalliomaki M, Collado MC, Salminen S et al (2008) Early differences in fecal microbiota composition in children may predict overweight. *Am J Clin Nutr* 87:534–538
18. White RA, Bjornholt JV, Baird DD et al (2013) Novel developmental analyses identify longitudinal patterns of early gut microbiota that affect infant growth. *PLoS Comput Biol* 9: e1003042
19. Carding S, Verbeke K, Vipond DT et al (2015) Dysbiosis of the gut microbiota in disease. *Microb Ecol Health Dis* 26:26191. <https://doi.org/10.3402/mehd.v26.26191>
20. Jost T, Lacroix C, Braegger CP et al (2014) Vertical mother-neonata transfer of maternal gut bacteria via breastfeeding. *Environ Microbiol* 16(9):2891–2904
21. Martin V, Maldonado-Barragan A, Moles L et al (2012) Sharing of bacterial strains between breast milk and infant feces. *J Hum Lact* 28:36–44. <https://doi.org/10.1177/0890334411424729>
22. Morelli L (2008) Postnatal development of intestinal microflora as influenced by infant nutrition. *J Nutr* 138:1791S–1795S
23. Guaraldi F, Salvatori G (2012) Effect of breast and formula feeding on gut microbiota shaping in newborns. *Front Cell Infect Microbiol* 2:94
24. Levy M, Thaiss CA, Elinav E (2015) Metagenomic cross-talk: the regulatory interplay between immunogenetics and the microbiome. *Genome Med* 7:120. <https://doi.org/10.1186/s13073-015-0249-9>
25. Pang WW, Hartmann PE (2007) Initiation of human lactation: secretory differentiation and secretory activation. *J Mammary Gland Biol Neoplasia* 12(4):211–221
26. Godhia ML, Patel N (2013) Colostrum- its composition, benefits as a nutraceutical- a review. *Curr Res Nutr Food Sci* 1(1):37–47
27. Castellote C, Casillas R, Ramirez-Santana C et al (2011) Premature delivery influences the immunological composition of colostrum and transitional and mature human milk. *J Nutr* 141(6):1181–1187
28. Playford RJ, MacDonald CE, Johnson WS (2000) Colostrum and milk-derived peptide growth factors for the treatment of gastrointestinal disorders. *Am J Clin Nutr* 72(1):5–14
29. Bode L, Jantscher-Krenn E (2012) Structure-function relationships of human milk oligosaccharides. *Adv Nutr* 3(3):383S–391S
30. Ballard O, Morrow AL (2013) Human milk composition: nutrients and bioactive factors. *Pediatr Clin N Am* 60(1):49–74
31. Urbaniak C, Angelini M, Gloor GB et al (2016) Human milk microbiota profiles in relation to birthing method, gestation and infant gender. *Microbiome* 4:1–9
32. Hunt KM, Foster JA, Forney LJ et al (2011) Characterization of the diversity and temporal stability of bacterial communities in human milk. *PLoS One* 6(6):E21313
33. Dominguez-Bello MG, De Jesus-Laboy KM, Shen N et al (2016) Partial restoration of the microbiota of cesarean-born infants via vaginal microbial transfer. *Nat Med* 22(3):251–254
34. Dominguez-Bello MG, Costello EK, Contreras M et al (2010) Delivery mode shapes the acquisition and structure of the initial microbiota across multiple body habitats in newborns. *Proc Natl Acad Sci U S A* 107(26):11971–11975
35. Ramsay DT, Kent JC, Owens RA et al (2004) Ultrasound imaging of milk ejection in the breast of lactating women. *Pediatrics* 113(2):361–367
36. Rescigno M, Urbano M, Valzasina B et al (2001) Dendritic cells express tight junction proteins and penetrate gut epithelial monolayers to sample bacteria. *Nat Immunol* 2(361):1–7
37. Macpherson AJ, Uhr T (2004) Induction of protective IgA by intestinal dendritic cells carrying commensal bacteria. *Science* 303(5664):1662–1665
38. Perez PF, Dore J, Leclerc M et al (2007) Bacterial imprinting of the neonatal immune system: lessons from maternal cells? *Pediatrics* 119(3):E724–E732
39. Donnet-Hughes A, Duc N, Serrant P et al (2000) Bioactive molecules in milk and their role in health and disease: the role of transforming growth factor-β. *Immunol Cell Biol* 78(1):74–79
40. Qutaishat SS, Stemper ME, Spencer SK et al (2003) Transmission of *Salmonella enterica* serotype typhimurium DT104 to infants through mother's breastmilk. *Pediatrics* 111(6 Pt 1):1442–1446

41. Cho I, Blaser MJ (2012) The human microbiome: at the interface of health and disease. *Nat Rev Genet* 21(4):260–270
42. Dethlefsen L, Huse S, Sogin ML et al (2008) The pervasive effects of an antibiotic on the human gut microbiota, as revealed by deep 16S rRNA sequencing. *PLoS Biol* 6:e280. <https://doi.org/10.1371/journal.pbio.0060280>
43. Dethlefsen L, Relman DA (2011) Incomplete recovery and individualized responses of the human distal gut microbiota to repeated antibiotic perturbation. *Proc Natl Acad Sci U S A* 108(Suppl 1):4554–4561. <https://doi.org/10.1073/pnas.1000087107>
44. Ubeda C, Taur Y, Jenq RR et al (2010) Vancomycin-resistant enterococcus domination of intestinal microbiota is enabled by antibiotic treatment in mice and precedes bloodstream invasion in humans. *J Clin Invest* 120:4332–4341
45. Goodrich JK, Di Rienzi SC, Poole AC et al (2014) Conducting a microbiome study. *Cell* 158:250–262
46. Canadian Paediatric Society (2004) Weaning from the breast. *Paediatr Child Health* 9 (4):249–253
47. Brand E, Kothari C, Stark MA (2011) Factors related to breastfeeding discontinuation between hospital discharge and 2 weeks post-partum. *J Perinat Educ* 20(1):36–44. <https://doi.org/10.1891/1058-1243.20.1.36>
48. Jones CA (2001) Maternal transmission of infectious pathogens in breastmilk. *J Paediatr Child Health* 37(6):576–582
49. Lovelady CA, Dewey KG, Picciano MF et al (2002) Guidelines for collection of human milk samples for monitoring and research of environmental chemicals. *J Appl Toxicol Environ Health* 65:1881–1891
50. Gonet L (2015) Breastfeeding trends in Canada. Statistics Canada Catalogue No. 82-624-X
51. Choo JM, Leong LEX, Rogers GB (2015) Sample storage conditions significantly influence faecal microbiome profiles. *Sci Rep* 5:16350
52. Sergeant MJ, Constantinidou C, Cogan T et al (2012) High-throughput sequencing of 16S rRNA gene amplicons: effects of extraction procedure, primer length and annealing temperature. *PLoS One* 7(5):e38094
53. Koren O, Goodrich JK, Cullender TC et al (2012) Host remodeling of the gut microbiome and metabolic changes during pregnancy. *Cell* 150(3):470–480
54. Salcedo J, Gormaz M, Lopez-Mendoza MC, Nogarotto E, Silvestre D (2015) Human milk bactericidal properties: effect of lyophilization and relation to maternal factors and milk components. *J Pediatr Gastroenterol Nutr* 60 (4):527–532
55. Caporaso JG, Lauber CL, Walters WA et al (2012) Ultra-high-throughput microbial community analysis on the Illumina HiSeq and MiSeq platforms. *ISME J* 6:1621–1624
56. Edgar RC (2010) Search and clustering orders of magnitude faster than BLAST. *Bioinformatics* 26:2460–2461
57. Edgar RC (2013) UPARSE: highly accurate OTU sequences from microbial amplicon reads. *Nat Methods* 10(10):996–998. <https://doi.org/10.1038/nmeth.2604>
58. Wang Q, Garrity GM, Tiedje JM et al (2007) Naïve Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. *Appl Environ Microbiol* 73 (16):5261–5267
59. Price LB, Liu CM, Melendez JH et al (2009) Community analysis of chronic wound bacteria using 16S rRNA gene-based pyrosequencing: impact of diabetes and antibiotics on chronic wound microbiota. *PLoS One* 4:e6462
60. Bokulich NA, Subramanian S, Faith JJ et al (2013) Quality-filtering vastly improves diversity estimates from Illumina amplicon sequencing. *Nat Methods* 10:57–59
61. Vazquez-Baeza Y, Pirrung M, Gonzalez A et al (2013) EMPeror: a tool for visualizing high-throughput microbial community data. *GigaScience* 2:16
62. Mackenzie BW, Waite DW, Taylor MW (2015) Evaluating variation in human gut microbiota profiles due to DNA extraction method and inter-subject differences. *Front Microbiol* 6:130. <https://doi.org/10.3389/fmicb.2015.00130>
63. Castelino M, Eyre S, Moat G et al (2017) Optimisation of methods for bacterial skin microbiome investigation: primer selection and comparison of the 454 versus MiSeq platform. *BMC Microbiol* 17:23. <https://doi.org/10.1186/s12866-017-0927-4>
64. Yang B, Wang Y, Qian PY (2016) Sensitivity and correlation of hypervariable regions in 16S rRNA genes in phylogenetic analysis. *BMC Bioinformatics* 17:135
65. Kozich JJ, Westcott SL, Baxter NT et al (2013) Development of a dual-index sequencing strategy and curation pipeline for analyzing amplicon sequence data on the MiSeq Illumina sequencing platform. *Appl Environ Microbiol* 79(17):5112–5120

66. Martin R, Jimenez E, Heilig H et al (2009) Isolation of bifidobacteria from breast milk and assessment of the bifidobacterial population by PCR-denaturing gradient gel electrophoresis and quantitative real-time PCR. *Appl Environ Microbiol* 75(4):965–969
67. Soto A, Martin V, Jimenez E et al (2014) Lactobacilli and bifidobacteria in human breast milk: influence of antibiotic therapy and other host and clinical factors. *J Pediatr Gastroenterol Nutr* 59(1):78–88
68. Milani C, Hevia A, Foroni E, Duranti S, Turroni F, Lugli GA, Sanchez B, Martin R, Gueimonde M, van Sinderen D et al (2013) Assessing the fecal microbiota: an optimized ion torrent 16S rRNA gene-based analysis protocol. *PLoS One* 8(7):e68739. <https://doi.org/10.1371/journal.pone.0068739>
69. Walker AW, Martin JC, Scott P, Parkhill J, Flint HJ, Scott KP (2015) 16S rRNA gene-based profiling of the human infant gut microbiota is strongly influenced by sample processing and PCR primer choice. *Microbiome* 3:26
70. Sim K, Cox MJ, Wopereis H, Martin R, Knol J, Li MS, Cookson WO, Moffatt MF, Kroll JS (2012) Improved detection of bifidobacteria with optimised 16S rRNA-gene based pyrosequencing. *PLoS One* 7(3):e32543. <https://doi.org/10.1371/journal.pone.0032543>
71. Hayashi H, Sakamoto M, Benno Y (2004) Evaluation of three different forward primers by terminal restriction fragment length polymorphism analysis for determination of fecal bifidobacterium spp. in healthy subjects. *Microbiol Immunol* 48(1):1–6
72. Highlander S (2014) Mock community analysis. Encyclopedia of Metagenomics. Springer, New York, pp 1–7
73. Huse SM, Huber JA, Morrison HG, Sogin ML, Welch DM (2007) Accuracy and quality of massively parallel DNA pyrosequencing. *Genome Biol* 8:R143
74. Kunin V, Engelbrekston A, Ochman H et al (2010) Wrinkles in the rare biosphere: pyrosequencing errors can lead to artificial inflation of diversity estimates. *Environ Microbiol* 12:118–123
75. Westcott SL, Schloss PD (2015) De novo clustering methods outperform reference-based methods for assigning 16S rRNA gene sequences to operational taxonomic units. *PeerJ* 3:e1487
76. Dave V, Street K, Francis S et al (2016) Bacterial microbiome of breast milk and child saliva from low-income Mexican-American women and children. *Pediatr Res* 79(6):846–854
77. Williams JE, Carrothers JM, Lackey KA et al (2017) Human milk microbial community structure is relatively stable and related to variations in macronutrient and micronutrient intakes in healthy lactating women. *J Nutr* 147(9):1739–1748
78. Kumar H, du Tolt E, Kulkarni A et al (2016) Distinct patterns in human milk microbiota and fatty acid profiles across specific geographic locations. *Front Microbiol* 7:1619
79. Boix-Amorós A, Collado MC, Mira A (2016) Relationship between milk microbiota, bacterial load, macronutrients, and human cells during lactation. *Front Microbiol* 7:492
80. Pannaraj PS, Li F, Cerini C et al (2017) Association between breast milk bacterial communities and establishment and development of the infant gut microbiome. *JAMA Pediatr* 171(7):647–654
81. Murphy K, Curley D, O'Callaghan TF et al (2017) The composition of human milk and infant faecal microbiota over the first three months of life: a pilot study. *Sci Rep* 7:40597
82. Cabrera-Rubio R, Mira-Pascual L, Mira A et al (2016) Impact of mode of delivery of the milk microbiota composition of healthy women. *J Dev Orig Health Dis* 7(1):54–60
83. Patel SH, Vaidya YH, Patel RJ et al (2017) Culture independent assessment of human milk microbial community in lactational mastitis. *Nat Sci Rep* 7:7804



Chapter 6

Quantification of Vitamin B₁₂-Related Proteins in Marine Microbial Systems Using Selected Reaction Monitoring Mass Spectrometry

Erin M. Bertrand

Abstract

Mass spectrometry-based proteomic approaches to studying microbial systems enable assessment of taxonomically resolved functional capacity. A subset of these proteomic approaches are absolutely quantitative, enabling comparisons of protein expression patterns between different studies and across environments. This chapter outlines a method for applying quantitative assays in marine microbial communities, using proteins involved in vitamin B₁₂ (cobalamin) utilization and production as specific examples. This approach involves identifying important protein targets, determining taxonomic resolution of the required assays, identifying suitable peptides, developing and optimizing liquid chromatography-selected reaction monitoring mass spectrometry assays (LC-SRM-MS), and processing the resulting data. Implementing the method outlined here results in measurements (fmol diagnostic peptide per µg of total bulk protein) that, in this case, define the nutritional status of microbial community members with respect to vitamin B₁₂, and are comparable across and between marine microbial systems.

Key words Targeted metaproteomics, Mass spectrometry, Microbial systems, Selected reaction monitoring

1 Introduction

Proteomics research seeks to identify and quantify proteins present in biological systems [1]. Because proteins catalyze reactions and facilitate resource acquisition, they largely define function. Functional characterizations are imperative for discerning the relationships between microbes and their surrounding environment or host system. Since protein sequences carry taxonomic information, functional community composition can also be inferred from detected proteins [2]. And because proteins represent major metabolic investment, quantifying the proteome uncovers resource allocation strategies and costs [3, 4]. In order to simultaneously realize each of these levels of information, the proteomic approach applied needs to be absolutely quantitative. This chapter details the process

for designing and applying quantitative selected reaction monitoring (SRM) mass spectrometry-based methods to measure the abundance of cobalamin-related proteins, MetE (cobalamin-independent methionine synthase; 5-methyltetrahydropteroyltri-L-glutamate:L-homocysteine S-methyltransferase) and MetH (cobalamin-dependent methionine synthase; 5-methyltetrahydrofolate:L-homocysteine S-methyltransferase), and CbiA, a cobalamin biosynthesis protein, cobyrinic acid a,c-diamide synthase (Table 1). The MetE and MetH measurements primarily target these enzymes produced by diatoms, important primary producers that require exogenous sources of cobalamin, while the CbiA measurement targets this protein from select *Gammaproteobacteria* which are important cobalamin producers. These methods have successfully been applied in laboratory cultures as well as microbial communities from the Southern Ocean [4, 5]. When applied together, they can provide a picture of cobalamin production and demand, a key inter-kingdom interaction, in marine microbial systems.

Table 1
Peptides for characterizing cobalamin production and demand in select marine microbial systems

Peptide sequence	Protein	Taxonomic coverage	Interpretation	Solvent for initial dilution
(K/R) ISGGISNLSFGFR	Cobalamin-dependent methionine synthase MetH	Present in all available MetH sequences from diatoms [4, 16]	Reflects diatom-use of cobalamin for key metabolic reactions [4, 17]	95% acetonitrile 0.5% formic acid
(K/R) VIQVDEPALR	Cobalamin-independent methionine synthase MetE	Present in all available MetE sequences from diatom as well as MetE sequences from select fungi and bacteria [4, 16, 18]	In diatom-dominated communities, high expression reflects cobalamin starvation in diatoms [4, 17]	95% acetonitrile 0.5% formic acid
(K/R) HLGLVQAHEVR	Cobyrinic acid a,c-diamide synthase, CbiA	Present in the dominant group of cobalamin biosynthesizers in the Southern Ocean [5, 19], gammaproteobacteria	Reflects cobalamin biosynthesis capacity in Southern Ocean communities [5, 19]	100% acetonitrile

^aL residues highlighted in bold are those that are 13C and 15N labeled (6× 13C, 1× 15N)

SRM mass spectrometry can be highly sensitive and quantitative, offering potential for measuring low-abundance targets, such as specific peptides, in complex samples [3, 6–9]. In contrast to global or survey proteomic approaches, SRM, as applied here, targets only a subset of proteins for analysis and detects them in an absolutely quantitative mode. Peptides are separated by reverse phase liquid chromatography, then introduced to a triple quadrupole mass spectrometer via electrospray ionization (LC-SRM-MS). Specific peptides, in a predetermined mass window, are collected in the mass spectrometer's first quadrupole. Those ions are then fragmented and the generation of predetermined product ions is monitored. The amount of the product ions detected is proportional to the abundance of the peptide of interest in a sample. This abundance is absolutely calibrated using stable-isotope-labeled versions of each peptide as internal standards [9]. Such measurements can provide a quantitative picture of resources allocated by diatoms to coping with low environmental availability of vitamin B₁₂ [4] and can quantify the contribution of specific taxonomic groups to vitamin B₁₂ biosynthesis capacity [5].

2 Materials

This application requires a mixture of peptides digested and purified from biomass samples derived from marine microbial communities. Since protein extraction procedures can be highly specific and significantly tailored for different sample types, a specific procedure is not included here (*see Note 1*).

2.1 Prepared Tryptic Peptide Digest from Environmental Samples

1. One sample, containing tryptic peptide digest of a total of 60–100 µg total protein, with equal parts pooled from each sample to be analyzed.
2. A minimum of 10 µg of tryptic peptide digest of each sample to be quantitatively analyzed (*see Note 2*).

2.2 Reagents

1. Known concentrations of stable-isotope-labeled versions of specific peptides of interest, in this case acquired from Sigma Aldrich (AQUA Peptides) and described in Table 1. Alternative suppliers include JPT Peptide Solutions (SpikeTides).
2. LC/MS grade water (e.g., Fisher Optima).
3. LC/MS grade acetonitrile (e.g., Fisher Optima).
4. LC/MS grade formic acid (e.g., Fisher Optima).
5. Methanol-rinsed microcentrifuge tubes (rinsed to reduce plasticizers).
6. Low-binding pipette tips.

7. Trapping column appropriate for capillary flow rates, for example: Acclaim Pepmap C18 micro precolumn.
8. C18 reverse phase chromatography column appropriate for peptide analysis at capillary flow rates. Examples:
 - (a) Magic C18AQ column (0.2×50 mm, $3 \mu\text{m}$ particle size, 200 \AA pore size, Michrom Bioresources).
 - (b) Acclaim Pepmap RSLC C18 column (0.3×100 mm $2 \mu\text{m}$ particle size, 100 \AA pore size).

2.3 Software Required for Building Analytical Methods and Processing Data

1. The open source Skyline package [10] available at: <https://skyline.ms/project/home/software/Skyline/begin.view>.
2. UniPept metaproteomics toolbox [11, 12], available in web-server and command line form at <http://uniPept.ugent.be/> or METATRYP [13], available at <https://github.com/saitomics/metatryp>.
3. Algorithms to assess the likelihood that peptides will be amenable to mass spectrometry, such as CONSeQuence [14], available at <http://king.smith.man.ac.uk/CONSeQuence/>.

2.4 Required Instrumentation

1. An HPLC system capable of controlled flow gradient elution at capillary flow rates ($2\text{--}5 \mu\text{L min}^{-1}$) interfaced via positive ion mode electrospray ionization with a triple quadrupole mass spectrometer.
2. An autosampler capable of injecting $1 \mu\text{L}$ samples without carryover and without more than $1 \mu\text{L}$ of sample loss (*see Notes 3 and 4*).

3 Methods

1. Using Skyline, prepare a list of potential peptide targets for each protein of interest:
 - (a) Assemble a .fasta file containing protein sequences to be analyzed (*see Note 5*).
 - (b) Open Skyline and paste the .fasta file containing all MetH sequences from diatoms into the “targets” window.
 - (c) Click on the “Settings” tab and “Peptide Settings.” Select Enzyme: Trypsin and 0 missed cleavages. Click the “Filter” tab and select the boxes that allow you to exclude peptides containing Cys and Met since these peptides can be differentially modified in your analytical matrix and should be avoided. Set the minimum peptide length to 6 and maximum length to 25. Click “OK.” A list of possible tryptic peptides will be generated.

- (d) Export this list by selecting File, Export, Transition list. The resulting .csv file will contain all possible tryptic peptides generated from diatom MetH sequences along with an identifier denoting their protein sequence of origin. From this list, select the peptide that is present in the most MetH sequences. In this case, it will be ISGGISNLSFGFR.
 - (e) Repeat for all targets, in this case MetE and CbiA.
2. To determine the taxonomic distribution of your selected peptides, you can employ tools including the “Tryptic peptide analysis” function in UniPept [11, 12]. This function will identify which genomes from RefSeq possess the peptide of interest and provide the lowest common ancestor. This information is required for interpreting the abundance of your target peptides in environmental samples. The results of these analyses are given in Table 1.
 3. These peptides should be assessed for likelihood of detectability by electrospray ionization mass spectrometry. This can be done by applying previously developed algorithms [14, 15] or by synthesizing the peptides and empirically observing their suitability in your LC-MS system. This empirical validation was previously performed for the peptides in Table 1 by verifying that they were observable under LC-MS conditions optimized for peptide analysis, as described below [4, 5].
 4. Once the target peptides have been determined, SRM methods should be developed for their detection. Using Skyline, prepare a transition list to optimize the detection method and monitor each native peptide and heavy-isotope-labeled internal standard peptide:
 - (a) Open Skyline and paste each of the three peptide sequences in Table 1 into the “Targets” window.
 - (b) Click on the “Settings” pulldown menu and select “Transition Settings.” In the “Prediction” tab select: Precursor mass and product ion mass: monoisotopic; Collision Energy: whichever instrument you are employing, in this case the TSQ Vantage; all others leave as “none.” In the “Filter” tab, select Precursor charge of 2, Ion charge of 1, and Ion type: y. Select “From *m/z* precursor” to “6 ions” and select “N-terminal to proline” in the “Special ions” box and check the box “Auto-select all matching transitions.”
 - (c) Click on the “Settings” pulldown menu and select “Peptide Settings” and “Modifications” tab. Select “Isotope Label Type: heavy” and “Label:13C(6)15N(1) (L)” from the list of possible modifications.

- (d) You should now see your three peptides with two parent ions each, the native and the heavy-isotope version, in your “Targets” window.
 - (e) You’ll need to modify the heavy version of “HLGLVQA-HEVR” since the modification applied altered both L residues. Right click on the peptide and select “Modify.” Then change the first heavy “L” residue back to the native version (*see Table 1*).
 - (f) Go to the “File” tab, and select “Export Transitions List.” Select the instrument type, in this case “Thermo” and “single method.” In the “optimizing” window, select “Collision Energy” and then click “Okay.” A .csv file will be generated that can be imported into your instrument method building software (in this case Thermo’s Xcalibur). This file will allow you to verify the optimal collision energies and y-ions to monitor in the next few steps.
5. Solubilize the isotopically labeled standard peptides (e.g., AQUA peptides) according to the manufacturer’s instructions.
 - (a) In the case of AQUA peptides, add 20 µL of one of several solvent mixtures, depending on peptide hydrophobicity and charge, to 1 nmol lyophilized vials. Solvent mixtures for each peptide are given in Table 1.
 - (b) Vortex these dilutions for 1 min at moderate speed, heat to 45 °C for 15 min, vortex again for 30 s, and then place in a sonication bath for 5 min to aid in peptide resuspension.
 - (c) Dilute these mixtures by *very slowly* adding 180 µL of 0.1% formic acid, 2% acetonitrile, and 98% water (*see Note 6*).
 6. Prepare a mixture of 100 fmol of each heavy labeled standard peptide per µL of 5% acetonitrile, 0.1% formic acid, and 95% water using the 5 pmol/µL stock.
 7. Prepare an instrument method which conducts the injection of 1 µL of the 100 fmol/µL stocks, loading onto a C18 trapping column and then onto a C18 analytical column, separates the peptides efficiently, and then interfaces, via electrospray ionization, into the triple quadrupole mass spectrometer (*see Note 7*). The electrospray ionization source should be operated in positive ion mode and the mass spectrometer should be programmed to acquire spectra via selected reaction monitoring to detect both the heavy and native versions of these peptides throughout the run using the file generated in Skyline (**step 4**).
 8. Prepare a series of at least three injections and complete LC-SRM-MS mass runs in order to verify the collision energies and optimal y-ions for monitoring. The y-ions and collision energies (Table 2) were previously identified via direct infusion of

each peptide (50 fmol/ μ L) in 20% acetonitrile, 80% water, and 0.1% formic acid at 4 μ L/min.

9. Process the data in order to verify the parameters given in Table 2 for your specific instrumentation.
 - (a) Raw data should be imported into Skyline through “File”; “Import”; “Results.”
 - (b) Return to the “Transition Settings” menu in the Settings pulldown and select “Use optimization values when present” and “optimize by Transition.”
 - (c) Observe the chromatograms to select the y-ions to monitor that generate the highest intensity. To do this, select the “View” pulldown, then “replicate comparison” and a bar graph showing the intensity of the y-ions generated will appear. Verify that the most abundant y-ions are those given in Table 2.
 - (d) Manually select the most abundant y-ions in the “Targets Window” and delete the other ions.
 - (e) Go to the “File” tab, and select “Export Transitions List.” Select the instrument type, in this case “Thermo” and “single method.” In the “optimizing” window, select “None” and then click “Okay.” A .csv file will be generated that can be used to generate your optimized transition list for quantifying these peptides.
10. Create a new instrument method with the same injection and liquid chromatography parameters as step 7, replacing the mass spectrometry SRM transition file with your newly optimized .csv transition file.
11. Create your standard curves for each peptide in order to verify their linearity and the dynamic range.
 - (a) Prepare a series of five solutions with increasing amounts of all three isotopically labeled peptide standards, ranging from 10^{-15} to 10^{-11} moles, added to 10 μ g of a mixture of your extracted peptide samples in 0.1% formic acid and 5% acetonitrile (*see Note 8*).
 - (b) Prepare a matrix blank: 10 μ g of total extracted peptides in 0.1% formic acid and 5% acetonitrile without heavy labeled standard additions.
12. After conducting several instrument blank injections (0.1% formic acid, 5% acetonitrile in LC/MS grade water), set up an injection of the highest concentration sample followed by two more instrument blanks. This is done in order to verify that your instrumentation does not have carryover between samples. Once the lack of peptide detection in the blank injections is verified, conduct triplicate 1 μ L injections of each

Table 2

Optimal selected reaction monitoring parameters optimized for a TSQ Vantage

Protein	Peptide	Parent ion (+z)	Parent (<i>m/z</i>)	Product (<i>m/z</i>)	Collision energy
MetE	VIQVDEPA[L_C13N15]R	2	573.833	934.507	16
		2	573.833	806.448	21
		2	573.833	707.380	20
MetH	VIQVDEPALR	2	570.325	927.489	16
		2	570.325	799.431	21
		2	570.325	700.362	20
CbiA	ISGGISN[L_C13N15] SFGFR	2	681.368	1161.612	21
		2	681.368	934.486	20
		2	681.368	613.309	24
	ISGGISNLSFGFR	2	677.859	1154.595	21
		2	677.859	927.468	20
		2	677.859	613.309	24
	HLG[L_C13N15] VQAHEVR	2	633.363	1015.576	28
		2	633.363	838.453	31
		2	633.363	739.386	27
		2	633.363	1128.660	28
		2	629.854	1008.559	28
		2	629.854	838.453	31
		2	629.854	739.386	27
		2	629.854	1121.643	28

solution, including the matrix blank using the method prepared in **step 10**.

13. Import the data into Skyline as in **step 9a**.
14. Manually inspect results to verify transition consistency, retention times, and replicate precision. To facilitate this inspection, select “View” “Peak Areas” and “Replicate Comparisons” and “View” “Retention Times” and “Replicate Comparisons.” This will generate bar graphs to facilitate comparison of multiple replicates (*see Note 9*).
15. Create an exportable report in which to manipulate data.

- (a) Choose “File” “Export” “Report.” Click on the Edit Report form, and “Add”. Name the report type “Summary” and select “PeptideSequence” “IsotopeLabelType,” BestRetentionTime, and TotalArea. Save and export. This will generate a .csv file that you can use to plot your results and conduct in-depth manual comparisons in R or Excel, for example (*see Note 10*).
- 16. For each isotope-labeled standard peptide, plot SRM peak area versus added concentration in order to verify linearity of peptide behavior across the dynamic range examined.
- 17. Conduct the quantification of the peptides of interest in your field samples:
 - (a) Prepare a solution of 20 fmol of each standard peptide per 1 μ L of 5% acetonitrile, 0.1% formic acid, and 95% water and use this to resuspend 10 μ g of each of your peptide samples to a concentration of 1 μ g/ μ L.
 - (b) Execute triplicate injections of 1 μ g of total bulk peptide per sample using the method applied in **step 10**.
- 18. Import raw data files into Skyline and process as in **step 15**.
- 19. To quantify the concentration of each peptide per sample, assume that the mass spectrometry response is the same for the heavy and native versions of these peptides and that the instrument response is linear (as verified in **step 16**) and use Eq. (1) [9]. *See Notes 11 and 12.*

$$\text{fmol native peptide} = \frac{\text{Peak area (native)}}{\text{Peak area (heavy)}} \cdot 20 \text{ fmol heavy peptide} \quad (1)$$

4 Notes

1. An example of a suitable method for sample collection, protein extraction, and digestion into tryptic peptides from marine samples can be found in Saito et al. 2014 [20], in particular pages 2–3 of the Supplemental Materials. It is imperative that the concentration of total bulk peptides (μ g/ μ L) be known. A suggested method for verifying the concentration of purified bulk peptides is tryptophan fluorescence [21]. Particular care must be taken to avoid selecting a bulk protein assay with significant potential interferences with likely contaminants in your samples of interest. For example, chlorophyll interferes with both Bradford and Lowry-type determinations [22].

2. 1 µg of total bulk peptides is required per sample injection, and a minimum of three injections is required for quantification. Accounting for sample loss during injection, the minimum total bulk peptide digest required per sample is 10 µg. Purified and lyophilized or concentrated peptide mixtures should be stored at –80 °C until analysis.
3. Higher sample loss per injection will require larger sample sizes than those discussed here.
4. Systems on which this method has been effectively applied include: (1) a Paradigm MS4 system (Michrom Bioresources Inc.) with an ADVANCE nanocapillary electrospray source (Michrom Bioresources Inc.) coupled to a Vantage TSQ Triple Quadrupole Mass Spectrometer (Thermo) or (2) an Ultimate 3000 RSLCnano system (Dionex), with the autosampler functioning in micro pickup mode, interfaced with either a SciEx QTRAP 5000 or a Quantiva TSQ Triple Quadrupole Mass Spectrometer (Thermo).
5. A .fasta file containing the representative proteins you would like to include in your analysis is required. In the case of this application, protein MetE and MetH sequences from available diatom genomes and transcriptomes [16] were identified via BLASTP searching with query protein sequences obtained from diatoms available in RefSeq [18]; *Phaeodactylum tricornutum* MetE and MetH sequences were used here. CbiA sequences derived from the dominant Southern Ocean cobalamin producers were identified in previous studies [5, 19] and acquired from NCBI's nr database.
6. Store these 5 pmol/µL stocks at 4 °C for up to 5 weeks; keep at –80 °C for longer-term storage.
7. A liquid chromatographic separation that has worked efficiently for these peptides includes applying a hyperbolic gradient from 5% to 35% buffer B over 40 min, where buffer A is 0.1% formic acid in water and buffer B is 0.1% formic acid in acetonitrile at a flow rate of 4 µL/min.
8. The objective is to inject increasing amounts of each peptide standard into the mass spectrometer in your experimental matrix. 1 µg of total bulk peptide is required per injection; 10^{-16} to 10^{-12} moles of each heavy standard are monitored in each injection.
9. In particular, observe whether the relative peak areas for each transition remain consistent. Major deviations suggest possible contaminating or interfering peptides. These will need to be excluded via either (a) chromatographic alterations to separate a contaminating peptide with the same parent mass and retention time as the analyte peptides, or (b) changes to the mass

spectrometry method to exclude the interfering product ion mass.

10. In addition, there are many tools within Skyline for processing this data. For an overview, view this tutorial: https://skyline.ms/tutorials/ExistingQuant-1_1.pdf.
11. The interpretation of the abundance of these peptides will be aided by assessments of the percentage of the microbial community that comprises diatoms, preferably as a function of biomass or total protein. Possible methods for conducting this include estimates based on cell counts.
12. The interpretation of the abundance of these peptides should also be periodically reevaluated as additional MetE, CbiA, and MetH sequences become available for analysis.

References

1. Vogel C, Marcotte EM (2012) Insights into the regulation of protein abundance from proteomic and transcriptomic analyses. *Nat Rev Genet* 13:227–232
2. Mesuere B, Devreese B, Debyser G et al (2012) UniPept: tryptic peptide-based biodiversity analysis of metaproteome samples. *J Proteome Res* 11:5773–5780
3. Saito MA, Bertrand EM, Dutkiewicz S et al (2011) Iron conservation by reduction of metalloenzyme inventories in the marine diazotroph *Crocospaera watsonii*. *Proc Natl Acad Sci U S A* 108:2184–2189
4. Bertrand EM, Moran DM, McIlvin MR et al (2013) Methionine synthase interreplacement in diatom cultures and communities: implications for the persistence of B12 use by eukaryotic phytoplankton. *Limnol Oceanogr* 58:1431–1450
5. Bertrand EM, Saito MA, Jeon YJ et al (2011) Vitamin B12 biosynthesis gene diversity in the Ross Sea: the identification of a new group of putative polar B12 biosynthesizers. *Environ Microbiol* 13:1285–1298
6. Picotti P, Bodenmiller B, Aebersold R (2013) Proteomics meets the scientific method. *Nat Methods* 10:24–27
7. Lange V, Malmström JA, Didion J et al (2008) Targeted quantitative analysis of *Streptococcus pyogenes* virulence factors by multiple reaction monitoring. *Mol Cell Proteomics* 7:1489–1500
8. Picotti P, Aebersold R (2012) Selected reaction monitoring-based proteomics: workflows, potential, pitfalls and future directions. *Nat Methods* 9:555–566
9. Gerber SA, Rush J, Stemman O et al (2003) Absolute quantification of proteins and phosphoproteins from cell lysates by tandem MS. *Proc Natl Acad Sci U S A* 100:6940–6945
10. MacLean B, Tomazela DM, Shulman N et al (2010) Skyline: an open source document editor for creating and analyzing targeted proteomics experiments. *Bioinformatics* 26:966–968
11. Mesuere B, Debyser G, Aerts M et al (2014) The UniPept metaproteomics analysis pipeline. *Proteomics* 15:1437–1442
12. Mesuere B, Willems T, Jeugt F, Van d et al (2016) UniPept web services for metaproteomics analysis. *Bioinformatics* 32:1746–1748
13. Saito MA, Dorsk A, Post AF et al (2015) Needles in the blue sea: sub-species specificity in targeted protein biomarker analyses within the vast oceanic microbial metaproteome. *Proteomics* 15:3521–3531
14. Eyers CE, Lawless C, Wedge DC et al (2011) CONSeQuence: prediction of reference peptides for absolute quantitative proteomics using consensus machine learning approaches. *Mol Cell Proteomics* 10:M110.003384
15. Mallick P, Schirle M, Chen SS et al (2007) Computational prediction of proteotypic peptides for quantitative proteomics. *Nat Biotechnol* 25:125–131
16. Keeling PJ, Burki F, Wilcox HM et al (2014) The Marine Microbial Eukaryote transcriptome Sequencing Project (MMETSP): illuminating the functional diversity of eukaryotic life in the oceans through transcriptome sequencing. *PLoS Biol* 12:e1001889
17. Bertrand EM, Allen AE, Dupont CL et al (2012) Influence of cobalamin scarcity on diatom molecular physiology and identification of

- a cobalamin acquisition protein. Proc Natl Acad Sci U S A 109:E1762–E1771
18. Pruitt KD, Tatusova T, Maglott DR (2005) NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. Nucleic Acids Res 33: D501–D504
19. Bertrand EM, McCrow JP, Moustafa A et al (2015) Phytoplankton-bacterial interactions mediate micronutrient colimitation at the coastal Antarctic sea ice edge. Proc Natl Acad Sci U S A 112:9938–9943
20. Saito MA, McIlvin MR, Moran DM et al (2014) Multiple nutrient stresses at intersecting Pacific Ocean biomes detected by protein biomarkers. Science 345:1173–1177
21. Wiśniewski JR, Gaugaz FZ (2015) Fast and sensitive total protein and peptide assays for proteomic analysis. Anal Chem 87:4110–4116
22. Eze JMO, Dumbroff EB (1982) A comparison of the Bradford and Lowry methods for the analysis of protein in chlorophyllous tissue. Can J Bot 60:1046–1049



Chapter 7

Single-Cell Genomics of Microbial Dark Matter

Christian Rinke

Abstract

Single-cell genomics allows bypassing the culturing step and to directly access environmental microbes one cell at a time. The method has been successfully applied to explore archaeal and bacterial candidate phyla, referred to as microbial dark matter. Here I summarize the single-cell genomics workflow, including sample preparation and preservation, high-throughput fluorescence-activated cell sorting, cell lysis and amplification of environmental samples. Furthermore I describe phylogenetic screening based on 16S rRNA genes and suggest a suitable library preparation and sequencing approach.

Key words Single-cell genomics, Microbial dark matter, Fluorescence-activated cell sorting, FACS, Multiple genome amplification, 16S rRNA gene, ILLUMINA Nextera XT libraries, Biofilm, Sludge, Sediment

1 Introduction

Genomic exploration of microbes was initially restricted to a small fraction of organisms which could be cultured in the laboratory, while the vast majority of microbes, up to 99% in certain habitats [1], remained unexplored. The existence of novel archaeal and bacterial phyla was known from 16S rRNA gene surveys, but these candidate phyla were without genomic information and have been referred to as microbial dark matter. Advances in methodology over the last decade now allow us to omit the culturing step and target microbes directly in environmental samples. Genomes are accessed either via metagenomics, the assembly and subsequent binning of shotgun sequencing data, or by single-cell genomics (SCG), the separation, amplification, and sequencing of DNA from single cells. SCG has the advantage that natural populations with a high degree of genomic strain heterogeneity can be dissected one cell at a time, avoiding the co-assembly of multiple strains. SCG has been employed to sequence uncultured bacterial phyla including OP9 (Atribacteria) from a hot spring [2], Poribacteria from sponges [3], SR-1 from human oral mucus [4], TM6

from a hospital sink biofilm [5], and TM7 from the human mouth [6]. Recently the potential of high-throughput SCG was showcased in the Microbial Dark Matter project [7] which recovered draft genomes from over 200 cells representing more than 20 major uncultivated archaeal and bacterial lineages. The single-cell genomes from this project made it possible to resolve numerous inter-phylum level relationships, to propose new archaeal and bacterial superphyla, and to discover unexpected metabolic features such as stop codon reassessments. Furthermore SCG has also been applied to dissect the genomic diversity within coexisting members of uncultured microbial species [8], and to explore interactions between marine protists, bacteria, and viruses *in situ* [9].

Microbial SCG follows a general workflow, starting with the separation of single cells from environmental samples. This can be carried out via a multitude of methods including micromanipulation [10], optofluidics (optical tweezing in conjunction with microfluidics [11]), laser-capture microdissection of tissue samples [12], and Fluorescence-activated cell sorting (FACS) [13]. The majority of microbial single-cell studies rely on FACS because of the high throughput and the ability to separate individual environmental cells on the basis of various cellular properties (e.g., size, fluorescence, granularity). The primary limitation is the inability to visually inspect the morphology of individual cells; however, cells can be sorted on slides and spot checked by microscopy.

Once single cells are separated, the next step is to make the cell envelope permeable to allow access for SCG reagents. This is generally achieved by exposure to a high pH solution (alkaline lysis), which has been shown to effectively lyse a wide range of Bacteria and Archaea [14]. The now accessible microbial DNA is then copied via whole genome amplification (WGA) to gain sufficient material for the genomic analysis. WGA methods include temperature cycled PCR, isothermal multiple displacement amplification (MDA), and recently a combination of both approaches has been proposed, called Multiple Annealing and Looping-Based Amplification Cycles (MALBAC) [15]. The most popular method for microbial single-cell genomics remains MDA, which combines straightforward chemistry, good genome coverage, and a very low error rate [16]. The single-cell amplified genomes (SAGs) are then used as input for sequencing library preparation. Low input DNA library methods, such as Illumina Nextera XT which produces usable assemblies from as little as 1 pg input DNA [17], are preferred since they allow multiple library creations from one SAG. Successful libraries are then shotgun sequenced using Illumina sequencing platforms.

Here I describe the FACS-based protocol (Fig. 1), starting with sample preparation and preservation, single-cell separation, cell lysis, and whole genome amplification to obtain single-cell amplified genomes (SAGs).

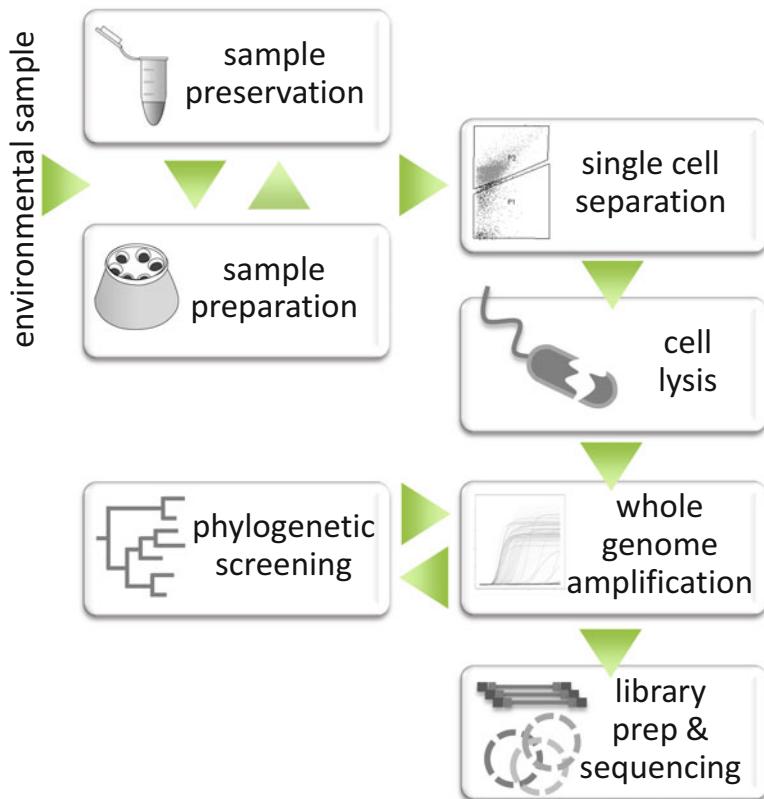


Fig. 1 Single-cell genomics workflow. The workflow includes sample preparation (Subheading 3.1), sample preservation (Subheading 3.2), single-cell separation (FACS based; Subheadings 3.3 and 3.4), cell lysis (alkaline solution; Subheading 3.5), whole genome amplification (MDA; Subheading 3.6), and optional phylogenetic screening (marker genes, e.g., 16S rRNA; Subheading 3.7). The amplified genomic DNA is then ready for sequencing library creation (Nextera XT) and subsequent shotgun sequencing. Please note that only a small aliquot of DNA generated in the WGA is used for the optional phylogenetic screening

2 Materials

2.1 Reagents

2.1.1 Sample Preparation and Preservation

1. Sample (for example, biofilm, fecal, seawater, sediment, soil).
2. Sterile, filtered buffer solution, for example, 1× PBS.
3. 100× TE, pH 8.0.
4. Milli-Q water (Millipore).
5. Molecular-grade glycerol.
6. Sterile UV-treated seawater.

2.2 Single-Cell Separation via FACS

1. Ultrapure water, such as Milli-Q water (Millipore), or filtered molecular biology-grade water.
2. Household bleach, a 3–8% (wt/vol) solution of sodium hypochlorite.

3. PBS liquid concentrate, 10×, sterile.
4. SYBR Green nucleic acid stain (e.g., Invitrogen SYBR Green nucleic acid gel stain, 10,000× concentrate, Thermo Fisher Scientific).

2.3 Single-Cell Lysis and Whole Genome Amplification via MDA

1. REPLI-g Single Cell Kit (Qiagen).
2. 5 mM SYTO 13 (Thermo Fisher Scientific).
3. Household bleach, a 3–8% (wt/vol) solution of sodium hypochlorite.

2.4 Phylogenetic Screening

1. Ultrapure water, such as Milli-Q water (Millipore) or filtered molecular biology-grade water.
2. SsoAdvanced SYBR Green Supermix (Biorad).
3. Primer set for marker gene of choice, for example, 16S rRNA gene universal primers 926wF (5'-GAAACTYAAAKGAATT GRCGG-3'; 10 μM) and 1392R (5'-ACGGGCGGTGTGT RC-3'; 10 μM).
4. ExoSAP-IT (Affymetrix).

2.5 Equipment

2.5.1 Sample Preparation and Preservation

1. Microcentrifuge tubes or cryovials, 2 mL (e.g., Eppendorf Safe-Lock tubes 2.0 mL, clear).
2. Microcentrifuge (e.g., Eppendorf 5424 ventilated microcentrifuge).
3. Vortex (e.g., VWR Scientific, Vortex Genie 2).
4. Centrifuge.
5. Standard light microscope.
6. Sterile cotton swabs (e.g., Fisher Scientific, Fisherbrand cotton-tipped applicators).
7. Ultrasonic water bath (e.g., Spectralab ultrasonic cleaning bath; Spectralab Instruments).
8. Falcon tube, 50 mL.
9. Glass beads, 2 mm diameter (e.g., solid-glass beads, borosilicate, diameter 2 mm; Sigma-Aldrich).
10. BD Falcon 40-μm nylon cell strainer (BD Biosciences).

2.6 Single-Cell Separation via FACS

1. Fluorescence-activated cell sorter (FACS; e.g., FACSAria II, BD Biosciences or MoFlo, Beckman Coulter; *see Note 1*).
2. PCR cabinet with UV light for decontamination of sheath fluid (e.g., Labconco).
3. Two 2 L quartz flasks for UV treatment of sheath fluid.
4. Two stir plates and stir bars for sheath fluid UV treatment.
5. BD Falcon 40-μm nylon cell strainer (BD Biosciences).

6. Polypropylene round-bottom tubes, 5 mL (e.g., BD Falcon 12 × 75-mm style, disposable tubes (BD Biosciences)).
7. Pall Acrodisc, 32-mm syringe filter with 0.1-μm Supor membrane (Pall).
8. BD Luer-Lok tip disposable syringe, 10 mL (BD Biosciences).
9. Optical micro-well plates to receive sorted single cells (e.g., Eppendorf twin.tec® PCR Plate 384, skirted).

2.7 Single-Cell Lysis and Whole Genome Amplification via MDA

1. Spectraline XL-1500 UV cross-linker (Fisher Scientific).
2. PCR cabinet with UV light for decontaminating work surfaces (e.g., Labconco).
3. Plate reader with temperature control or a real-time thermocycler (e.g., ViiA™ 7 Real-Time PCR System with 384-Well Block, Fisher Scientific).
4. Eppendorf Safe-Lock tubes, 1.5 mL (Eppendorf).
5. EMD colorpHast pH strips (Fisher Scientific).

2.8 Phylogenetic Screening

1. Standard thermocycler or a real-time thermocycler (e.g., ViiA™ 7 Real-Time PCR System with 384-Well Block, Fisher Scientific).
2. Plate shaker (e.g., Eppendorf MixMate Vortex Mixer; VWR).
3. Optical microtiter plate (e.g., Eppendorf twin.tec® PCR Plate 384, skirted).

3 Methods

3.1 Sample Preparation

Sample processing procedures vary and are specific for the type of sample. The common goal is to obtain a solution containing single microbial cells. Procedures to achieve this include dislodging cells from particles such as sediments or soil, disrupting microbial biofilms, or separating cells from sludge. I focus here on three sample types: sediment, biofilm, and sludge samples, since they require more involved sample preparations.

3.1.1 Sediment/Soil Samples

1. Mix 5 g of sediment sample with 10–30 mL of sterile buffer in a 50-mL Falcon tube. For soil samples and freshwater sediments, use 1× PBS as buffer. For marine sediments use sterile-filtered, UV-treated seawater.
2. Vortex the sample for 30 s. This step will dislodge the microbial cells from the sediment/soil particles.
3. Centrifuge the sample at $2500 \times g$ at room temperature (25 °C) for 30 s to remove large particles.
4. Collect the supernatant, and proceed to Subheading 3.2.

3.1.2 Biofilm Sample

1. Collect a sample of biomass using a cotton swab, and deposit it into microcentrifuge tubes containing a sterile-filtered, isotonic buffer solution, e.g., 1× PBS or seawater.
2. Sonicate the sample tube in an ultrasonic water bath by floating the tube for 10 min at the default setting at room temperature.
3. Shake the tube by hand for an additional 5 min.
4. Examine the sample under a microscope to ensure a sufficient separation of cells. If necessary, repeat **steps 2–4**.

3.1.3 Granular Sludge Sample

1. Sample ~5 mL of liquid sludge and transfer it into a 50-mL Falcon tube. Fill the tube up to a 50 mL volume with a sterile-filtered, isotonic buffer solution, e.g., 1× PBS or seawater.
2. Shake the Falcon tube by hand for 1 min.
3. Centrifuge the sample at $14,000 \times g$ for 15 min. The cells will form a pellet.
4. Remove most of the supernatant to reduce the total liquid volume to 15 mL.
5. Add 0.1 g of 2-mm-diameter glass beads into the tube.
6. Shake the tube by hand for 10 min. This will dislodge the cells in the pellet.
7. Let the tube stand for 3 min, and then collect the upper half of the suspension and transfer it into a new tube.
8. Remove the glass beads by filtration through a 40-μm nylon cell strainer.
9. Examine the sample under a microscope to ensure a sufficient separation of cells. If necessary, repeat **steps 2–8**.

3.2 Sample Preservation

After sample preparation the single-cell samples should be cryopreserved in a glycerol stock and stored at –80 °C.

1. Aliquot nuclease-free water into sterile 1.5-mL Eppendorf Safe-Lock tubes and UV-irradiate tubes in a Spectraline XL-1500 UV cross-linker for 1 h, to obtain UV-treated nucleic-acid-free water.
2. For cryo-protection make a GlyTE stock in a 250-mL flask. Add 20 mL of 100× TE (pH 8.0), 60 mL of UV-treated nuclease-free water. Next use a syringe to add 100 mL of molecular-grade glycerol (*see Note 2*). Store the stock at –20 °C for up to 1 year.
3. Transfer 100 μL of GlyTE stock and 1 mL of sample to a sterile cryovial.
4. Mix the cryovial gently and incubate it for 1 min at room temperature.
5. Flash freeze in liquid nitrogen and store at –80 °C. Prepare several replicate vials for each sample.

3.3 FACS Preparation for Aseptic Sort

A sterile/aseptic FACS is essential to minimize exogenous contamination. Sheath fluid and the FACS fluidic lines are possible contamination sources. Effective decontamination can be achieved by UV treatment of the sheath fluid and by running bleach through the fluidic lines. It is recommended to perform this preparation for aseptic sorting before every run.

1. In a clean UV cabinet prepare 4 L of 1× PBS in two 2-L quartz flasks. Move the flasks onto stir plates, add a stir bar, and start stirring within the cabinet. Place the empty sheath fluid tank and its lid with the inner surfaces face up in the UV cabinet. Close the cabinet and start the overnight (≥ 16 h) UV exposure. After UV exposure, transfer the sheath fluid to the tank in the UV cabinet (see Note 3). Reserve at least 10 mL of clean sheath fluid for later use while sorting.
2. Prepare a second sheath tank with 1 L of 10% (wt/vol) bleach (0.3–0.8% (wt/vol) sodium hypochlorite, final concentration) and run it through the FACS for 2 h to decontaminate fluidic lines (see Note 4).
3. Dispose of any remaining bleach and rinse the sheath tank with sterile water. Run 1 L of sterile water through the FACS for 30 min to rinse fluidic lines.
4. Switch to the tank with sheath fluid and start the sort. Follow the procedure specified in your FACS manual to center the stream, adjust the laser and detectors, and to calculate drop delay.
5. Before running samples on the FACS, UV-treat the microtiter plates without a cover/seal for 10 min. Seal the plates and UV-treat again for 10 min (see Note 5).

3.4 Single-Cell Separation via FACS

The operation of cell sorters differs by brand and model and the type of samples used for microbial single-cell genomics can vary substantially. Therefore we focus here on general guidelines which should be applicable to most instruments and work for the majority of environmental samples.

1. Filter each sample through an appropriately sized filter (e.g., 40- μm filter when using a 70- μm nozzle). This step is essential to avoid clogging the nozzle in the FACS.
2. Stain the sample with 1× SYBR Green fluorescent nucleic acid stain for 15 min at 4 °C in the dark (see Note 6).
3. Run the stained sample and target your desired microbial population with a sort gate.
4. Sort the target cells into the UV-treated microtiter plates. I recommend sorting the cells into 96 or 384 well plates, including columns of negative and positive controls (Fig. 2). A

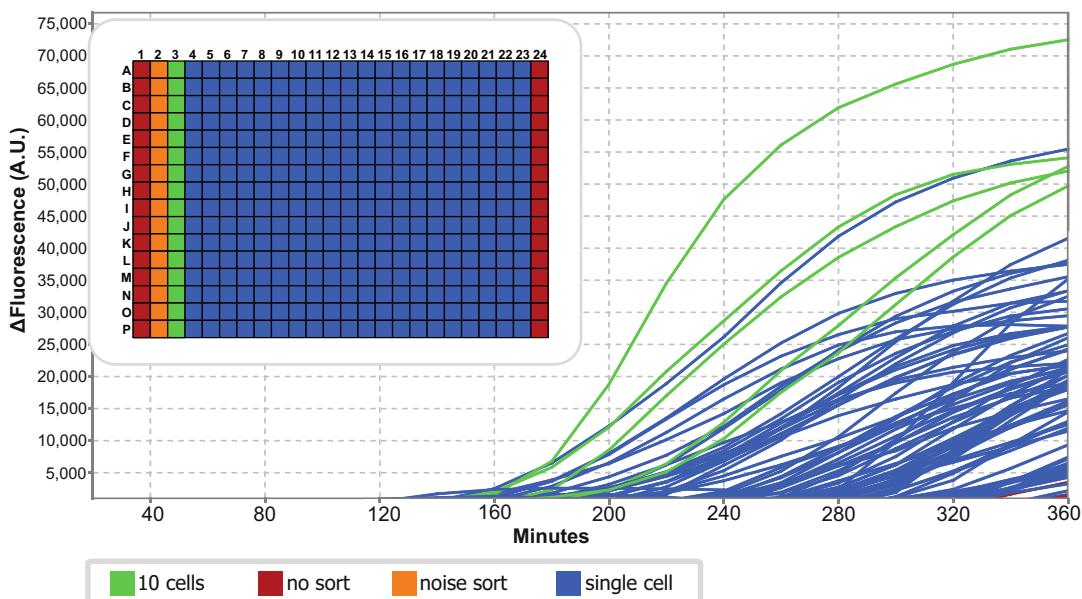


Fig. 2 Real-time whole genome amplification. The graph shows MDA kinetics for a 384-well plate of sorted single cells with positive controls (10 cells per well) and negative controls (no sort; noise sort). MDA success is measured by a fluorescence value above 5000 arbitrary units (A.U.). Positive controls (green) should be among the first wells to amplify, whereas some negative controls (red) usually amplify toward the end of the 6 h (360 min) amplification. The **insert** shows the plate layout for a 384 well plate including single cells and controls. “No sort” are no-template negative controls, and “noise sort” are negative controls where only the sheath fluid is sorted

two-step sort can be performed to dilute free DNA in the sample (*see Note 7*).

- After sort is complete, seal plates with a sterile cover and store at -80°C .

3.5 Single-Cell Lysis

The Qiagen REPLI-g Single Cell Kit includes all necessary reagents for cell lysis and whole genome amplification (WGA). Refer to the manufacturer’s protocol for detailed information. Note the recommended total reaction volume, including lysis and amplification reagents, is 50 μL . However, this volume can be considerably reduced without any adverse effects (*see Note 8*). The entire workflow including cell lysis and WGA should be performed in a sterile and DNA-free cabinet (*see Note 9*).

- Before performing any work, wipe down all cabinet surfaces, pipettes, and equipment with 10% (wt/vol) bleach. UV-treat the clean cabinet for 60 min with equipment inside.
- Prepare lysis buffer D2 according to the manufacturer’s protocol.

3. UV-sterilize the PBS (PBS sc), lysis buffer D2 and the STOP buffer for 1 h in the UV cross-linker. Store on ice.
4. Check the pH of the lysis buffer D2 by pipetting 4 µL onto a pH strip. The desired pH is 12–14 and buffers with a pH below 12 should not be used since the cell lysis is compromised.
5. Add 4 µL PBS (PBS sc) to each well, mix by tapping.
6. Add 3 µL buffer D2 to each well, mix by tapping and spin down at $1000 \times g$ for 30 s. Incubate for 10 min at 65 °C.
7. Add 3 µL of STOP buffer to each well. Spin down the plate at $1000 \times g$ for 30 s. Store lysed cells at 4 °C for up to 1 h while preparing the WGA mastermix.

3.6 Whole Genome Amplification (WGA)

I recommend using the Qiagen REPLI-g Single Cell Kit, since it is designed for the amplification of low starting material including single microbial cells. All necessary components for the single-cell whole genome amplification are included in the kit.

1. Thaw all kit reagents at room temperature, except the Phi29 polymerase (REPLI-g sc DNA Polymerase) which needs to be thawed on ice.
2. Prepare the mastermix according to the instructions in the Qiagen REPLI-g Single Cell Kit protocol in a sterile, DNA-free cabinet. First combine the reaction buffer with the sterile water (provided in the Qiagen kit) and mix by vortexing and a quick spin.
3. To monitor the single-cell amplification in real time add SYTO 13 to the master mix. The final concentration should be ~0.5 µM (*see Note 10*).
4. Add the Phi29 polymerase, vortex for 1 s, and quick spin.
5. To each well, containing already 10 µL of reagents (3 µL PBS, 4 µL Lysis buffer, 3 µL STOP buffer), add 40 µL mastermix for a total reaction volume of 50 µL. Cover the plate with an optical seal and centrifuge at $1000 \times g$ for 1 min (*see Note 11*).
6. Incubate the sealed plate at 30 °C for 6 h in a real-time thermocycler (e.g., ViiA™ 7 Real-Time PCR System with 384-Well Block, Fisher Scientific) (Fig. 2; *see Note 12*).
7. Heat-inactivate the Phi29 polymerase by incubating the plate at 65 °C for 10 min. MDA products can be stored for up to a year at –80 °C (*see Note 13*).

3.7 Phylogenetic Screening

Phylogenetic screening of single-cell amplified genomes (SAGs) is optional, but it is useful to pre-screen and identify SAGs of interest for downstream analysis such as sequencing. I focus on the 16S rRNA gene, since it is the most commonly used marker gene to establish taxonomic classifications.

1. Dilute the MDA product 1:20 in nuclease-free water. Mix the dilution thoroughly by hand-pipetting up and down, or alternatively use a plate shaker for 15 min at the maximum setting (*see Note 14*).
2. Transfer 2 µL of diluted MDA product as template to an optical microtiter plate (e.g., Eppendorf twin.tec® PCR Plate 384).
3. Thaw the reagents for the 16S rRNA PCR on ice: SsoAdvanced SYBR Green Supermix, 10 µM 926wF primer and 10 µM 1392R primer (*see Note 15*).
4. Prepare a sufficient volume of master mix, where each reaction should contain: 2.6 µL nuclease-free water, 5 µL SsoAdvance SYBR Green Supermix (2×), 0.2 µL 926wF primer (10 µM), and 0.2 µL 1392R primer (10 µM). Mix by vortexing and spin down.
5. To each well containing 2 µL of diluted MDA product as template, add 8 µL of PCR master mix, for a total reaction volume of 10 µL. Cover the plate with an optical seal and centrifuge at 1000 × *g* for 1 min.
6. Amplify samples in a thermocycling instrument, using the cycling program according to the PCR kit manufacturer's instructions. Using a real-time instrument allows adding a melt curve step to the cycling program which will aid in the analysis of PCR products.
7. Purify and sequence PCR products to identify the single-cell genomes. For example, purify with the ExoSAP-IT according to the manufacturer's instructions, and Sanger sequence. The resulting 16S rRNA gene sequences can be analyzed via online databases such as GreenGenes (<http://greengenes.soechardgenome.com/downloads>) or SILVA (<http://www.arb-silva.de/>).

4 Notes

1. We use a BD FACS Aria II (Becton, Dickinson and Company) with a 70-nm nozzle and 488-nm excitation laser to detect and sort prokaryotic cells labeled with the DNA stain SYBR green.
2. Glycerol is extremely viscous and is therefore most accurately transferred via syringe.
3. We suggest dedicating a tank to clean sheath fluid for single-cell genomics use only.
4. Some FACS instruments such as the BD FACS Aria II do not allow running bleach through all fluidic lines due to a bleach sensitive filter in the sheath fluid line. We successfully replaced this filter with a custom filter assembly (e.g., Balston high

chemical resistance disposable filter; company, L9922-05-AQ, cat. no. 9922-05-AQ) tested here at the School of Chemistry and Molecular Biosciences, UQ.

5. Traditionally, liquid (such as buffer) was added to the empty wells to prevent cells from drying out. However I found that a “dry sort” into empty wells does not result in degradation of the genome recovered downstream.
6. Minimize light exposure of SYBR green since the stain is light sensitive.
7. Environmental DNA (eDNA), defined as genetic material obtained directly from environmental samples (soil, sediment, water, etc.) without any obvious signs of biological source material, is present in almost all environments [18]. The FACS can be used to efficiently dilute eDNA down to an undetectable level. Sort at least 10,000 target cells into 1 mL of UV-treated 1× PBS or UV-treated sheath fluid. Backflush the sample line for 1 min, run 10% bleach solution through the sample line for 5 min and backflush for an additional 5 min. Sort this “pre-sorted” population of cells into the UV-treated plates containing 4 µL 1× PBS per well.
8. Reducing the 50 µL total reaction volume to 25 µL can be achieved by halving the volumes provided in the manufacturer’s protocol. When reducing the volume further it is important to maintain a lysis buffer concentration of approximately 6–7% (vol/vol) in the final WGA mastermix. The STOP solution needs to be in the exact same concentration as the lysis buffer. I have successfully tested volume reductions down to 1/10th, resulting in a total reaction volume of only 5 µL.
9. Single-cell lysis and amplification are highly susceptible to outside contamination; therefore, it is important to reserve a sterile and DNA-free space for single-cell work. The utilized equipment should be dedicated to single-cell tasks only and maintained as sterile and DNA free as possible.
10. Minimize the exposure of SYTO13 to direct light before adding it to the master mix.
11. Check the centrifuged plate to ensure that no air bubbles remain in the wells. The presence of air bubbles may interfere with the real-time readings measured by the instrument. Also do not let the MDA master mix warm to a temperature above 30 °C. The Phi29 enzyme activity suffers at temperatures above 30 °C.
12. If the WGA progress is monitored directly, the reaction can be cut short when the negative controls start to amplify.
13. Freezing and thawing of the hyper-branched MDA products results in a nonhomogeneous DNA product which is difficult

to dilute accurately. Therefore, if possible, proceed directly from WGA to phylogenetic screening, skipping the freezing step.

14. The MDA product is highly viscous. It is crucial to ensure that the MDA product and the dilution is mixed thoroughly. Robotic instrumentation may increase mixing efficiency.
15. Minimize the light exposure of SsoAdvance SYBR Green Supermix.

Acknowledgments

The work was conducted at the Australian Centre for Ecogenomics (ACE) at UQ, and was supported by the ARC Discovery Project DP160103811. I would like to thank the ACE team for support, especially Dr. Michael Nefedov and Alexander Baker.

References

1. Amann RI, Ludwig W, Schleifer KH (1995) Phylogenetic identification and in situ detection of individual microbial cells without cultivation. *Microbiol Rev* 59:143–169
2. Dodsworth JA, Blainey PC, Murugapiran SK, Swingley WD, Ross CA, Tringe SG, Chain PSG, Scholz MB, Lo C-C, Raymond J, Quake SR, Hedlund BP (2013) Single-cell and metagenomic analyses indicate a fermentative and saccharolytic lifestyle for members of the OP9 lineage. *Nat Commun* 4:1854. <https://doi.org/10.1038/ncomms2884>
3. Kamke J, Sczyrba A, Ivanova N, Schwientek P, Rinke C, Mavromatis K, Woyke T, Hentschel U (2013) Single-cell genomics reveals complex carbohydrate degradation patterns in poribacterial symbionts of marine sponges. *ISME J* 7 (12):2287–2300. <https://doi.org/10.1038/ismej.2013.111>
4. Campbell JH, O'Donoghue P, Campbell AG, Schwientek P, Sczyrba A, Woyke T, Söll D, Podar M (2013) UGA is an additional glycine codon in uncultured SRI bacteria from the human microbiota. *Proc Natl Acad Sci* 110:5540–5545. <https://doi.org/10.1073/pnas.1303090110>
5. McLean JS, Lombardo M-J, Badger JH, Edlund A, Novotny M, Yee-Greenbaum J, Vyahhi N, Hall AP, Yang Y, Dupont CL, Ziegler MG, Chitsaz H, Allen AE, Yoosheph S, Tesler G, Pevzner PA, Friedman RM, Nealson KH, Venter JC, Lasken RS (2013) Candidate phylum TM6 genome recovered from a hospital sink biofilm provides genomic insights into this uncultivated phylum. *Proc Natl Acad Sci* 110(26):E2390–E2399. <https://doi.org/10.1073/pnas.1219809110>
6. Marcy Y, Ouverney C, Bik EM, Lösekann T, Ivanova N, Martin HG, Szeto E, Platt D, Hugenholz P, Relman DA, Quake SR (2007) Dissecting biological “dark matter” with single-cell genetic analysis of rare and uncultivated TM7 microbes from the human mouth. *Proc Natl Acad Sci* 104:11889–11894. <https://doi.org/10.1073/pnas.0704662104>
7. Rinke C, Schwientek P, Sczyrba A, Ivanova NN, Anderson IJ, Cheng J-F, Darling A, Malfatti S, Swan BK, Gies EA, Dodsworth JA, Hedlund BP, Tsiamis G, Sievert SM, Liu W-T, Eisen JA, Hallam SJ, Kyripides NC, Stepanauskas R, Rubin EM, Hugenholz P, Woyke T (2013) Insights into the phylogeny and coding potential of microbial dark matter. *Nature* 499:431–437. <https://doi.org/10.1038/nature12352>
8. Kashtan N, Roggensack SE, Rodrigue S, Thompson JW, Biller SJ, Coe A, Ding H, Marttinen P, Malmstrom RR, Stocker R, Follows MJ, Stepanauskas R, Chisholm SW (2014) Single-cell genomics reveals hundreds of coexisting subpopulations in wild Prochlorococcus. *Science* 344:416–420. <https://doi.org/10.1126/science.1248575>
9. Yoon HS, Price DC, Stepanauskas R, Rajah VD, Sieracki ME, Wilson WH, Yang EC, Duffy S, Bhattacharya D (2011) Single-cell genomics reveals organismal interactions in uncultivated marine protists. *Science*

- 332:714–717. <https://doi.org/10.1126/science.1203163>
10. Woyke T, Tighe D, Mavromatis K, Clum A, Copeland A, Schackwitz W, Lapidus A, Wu D, McCutcheon JP, McDonald BR, Moran NA, Bristow J, Cheng J-F (2010) One bacterial cell, one complete genome. PLoS One 5:e10314. <https://doi.org/10.1371/journal.pone.0010314>
11. Landry ZC, Giovanonni SJ, Quake SR, Blainey PC (2013) Chapter 4: Optofluidic cell selection from complex microbial communities for single-genome analysis. In: DeLong EF (ed) Methods Enzymol. Academic Press, Cambridge, pp 61–90
12. Frumkin D, Wasserstrom A, Itzkovitz S, Harmelin A, Rechavi G, Shapiro E (2008) Amplification of multiple genomic loci from single cells isolated by laser micro-dissection of tissues. BMC Biotechnol 8:17. <https://doi.org/10.1186/1472-6750-8-17>
13. Rinke C, Lee J, Nath N, Goudeau D, Thompson B, Poulton N, Dmitrieff E, Malmstrom R, Stepanauskas R, Woyke T (2014) Obtaining genomes from uncultivated environmental microorganisms using FACS-based single-cell genomics. Nat Protoc 9:1038–1048
14. Clingenpeel S, Clum A, Schwientek P, Rinke C, Woyke T (2014) Reconstructing each cell's genome within complex microbial communities - dream or reality? Microb Physiol Metab 5:771. <https://doi.org/10.3389/fmicb.2014.00771>
15. Zong C, Lu S, Chapman AR, Xie XS (2012) Genome-wide detection of single-nucleotide and copy-number variations of a single human cell. Science 338:1622–1626. <https://doi.org/10.1126/science.1229164>
16. de Bourcy CFA, De Vlaeminck I, Kanbar JN, Wang J, Gawad C, Quake SR (2014) A quantitative comparison of single-cell whole genome amplification methods. PLoS One 9:e105585. <https://doi.org/10.1371/journal.pone.0105585>
17. Rinke C, Serene L, Ben W, Jean-Baptiste R, Adam S, Xuyen L, Margaret KB, Stocker R, Seymour J, Tyson GW, Hugenholtz P (2016) Validation of picogram-input DNA libraries for microscale metagenomic. PeerJ 4:e2486. <https://doi.org/10.7717/peerj.2486>
18. Thomsen PF, Willerslev E (2015) Environmental DNA—an emerging tool in conservation for monitoring past and present biodiversity. Biol Conserv 183:4–18. <https://doi.org/10.1016/j.biocon.2014.11.019>



Chapter 8

16S rRNA Gene Analysis with QIIME2

Michael Hall and Robert G. Beiko

Abstract

Microbial marker-gene sequence data can be used to generate comprehensive taxonomic profiles of the microorganisms present in a given community and for other community diversity analyses. The process of going from raw gene sequences to taxonomic profiles or diversity measures involves a series of data transformations performed by numerous computational tools. This includes tools for sequence quality checking, denoising, taxonomic classification, alignment, and phylogenetic tree building. In this chapter, we demonstrate how the Quantitative Insights Into Microbial Ecology version 2 (QIIME2) software suite can simplify 16S rRNA marker-gene analysis. We walk through an example data set extracted from the guts of bumblebees in order to show how QIIME2 can transform raw sequences into taxonomic bar plots, phylogenetic trees, principal co-ordinates analyses, and other visualizations of microbial diversity.

Key words Microbial ecology, Marker gene, 16S rRNA gene, QIIME, Bioinformatics

1 Introduction

Molecular techniques give us the ability to characterize microorganisms and gain insight into the important biological processes that they drive. Modern high-throughput methods allow for the interrogation of entire communities of microorganisms in parallel. One such method is marker gene analysis. In this type of microbial community analysis, a phylogenetically informative and universal gene (or gene fragment) is isolated and amplified with the polymerase chain reaction (PCR). The amplified gene product is sequenced and the variation within the gene sequences is exploited to predict the taxonomic groups that are present in a sample, their relative abundances, and community-diversity measures.

These community descriptors are generated by a series of computational transformations of the original sequence data. Some of these transformations, such as sequence quality filtering, sequence alignments, and phylogeny building, are common bioinformatic tasks that can be accomplished with more general tools. Other transformations, such as taxonomic classification, or the

quantification of community-profile similarity are more likely to require databases and tools designed specifically for marker-gene analyses. Dozens of software tools and dependencies are required for a complete analysis, written in various programming languages and with documentation spread across many different locations. These tools interact with one another in a process workflow, feeding from one to the next. Often, the output format of one step does not match the input requirements of the next step, so an additional transformation is required. This can lead to complicated analyses that are performed with ad-hoc commands and data manipulations that make analysis replication and data-provenance tracking difficult or impossible.

This chapter demonstrates a microbial marker-gene analysis using the Quantitative Insights Into Microbial Ecology version 2 (QIIME2, pronounced “chime two”) software suite [3]. QIIME2 provides a software environment, data standards, and tool wrappers that allow for seamless interoperability between tools used for microbial community analysis. We describe a typical analysis pipeline using QIIME2, and demonstrate how study replication and data provenance can be simplified with scripting and QIIME artifacts. This chapter is accompanied by a GitHub repository (https://github.com/beiko-lab/mimb_16S) which contains scripts to download an example data set and process the data using the marker-gene analysis pipeline described here.

2 Materials

2.1 Sequence Data

This protocol requires a marker-gene data set generated from a 16S rRNA gene fragment and sequenced as paired-end reads on an Illumina platform. While in principle other genes and sequencing platforms could be used with QIIME2 and its associated tools, the default parameters and databases are tuned for the 16S rRNA gene and Illumina paired-end sequences. Use of sequence data from other genes or sequencing platforms would necessitate substituting an appropriate reference sequence set and a critical re-evaluation of default parameters and models on many steps, but particularly those associated with sequence denoising and taxonomic classification.

Sequence data should be in FASTQ format and must be named using the Illumina naming convention. For example, a gzip-compressed FASTQ file may be called `SampleName_S1_L001_R2_001.fastq.gz`, where `SampleName` is the name of the sample, `S1` indicates the sample number on the sample sheet, `L001` indicates the lane number, `R2` indicates that the file contains the reverse reads (with `R1` indicating forward reads), and the last three numbers are always `001` by convention. The files should be demultiplexed, which means that there is one FASTQ sequence file for every sample, and all of the FASTQ files should be placed

into the same directory. This directory should contain only two other files. The first is `metadata.yaml`. This is a simple text file that contains only the text `{phred-offset: 33}` on a single line (*see Note 1*). The second file is named `MANIFEST` and is a three-column comma-separated text file with the first column listing the sample name (matching the FASTQ filename convention), the second column listing the FASTQ file name, and the third column listing whether the reads are “forward” or “reverse.” Here are the first 5 lines of the `MANIFEST` file from the example data set:

```
sample-id,filename,direction
SRR3202913,SRR3202913_S0_L001_R1_001.fastq.gz,forward
SRR3202913,SRR3202913_S0_L001_R2_001.fastq.gz,reverse
SRR3202914,SRR3202914_S0_L001_R1_001.fastq.gz,forward
SRR3202914,SRR3202914_S0_L001_R2_001.fastq.gz,reverse
```

To import a directory of sequence files into QIIME, the directory must contain all of the FASTQ files, a `metadata.yaml` file, and a complete `MANIFEST` file listing each of the FASTQ files in the directory. Subheading 3.1 describes the import process.

2.2 Sample Metadata

Sample metadata is stored in a tab-separated text file. Each row represents a sample, and each column represents a metadata category. The first line is a header that contains the metadata category names. These cannot contain special characters and must be unique. The first column is used for sample names and must use the same names as in the `sample-id` column of the `MANIFEST` file. The QIIME developers host a browser-based metadata validation tool, Keemei (<https://keemei.qiime2.org/>), that checks for correct formatting and helps identify any errors in the metadata file [16]. Metadata files used in QIIME1 analyses are compatible with QIIME2 and can be used without modification.

2.3 Software

For this computational pipeline, we will be using the 2018.2 distribution of the QIIME2 software suite. The installation process has been significantly simplified over previous iterations of QIIME. The entire package, including all dependencies and tools, can be automatically installed with the Anaconda/Miniconda package and environment manager (available at <https://anaconda.org/>). The QIIME software is placed in a virtual environment so that it does not interfere or conflict with any existing software on the system. Once installed, the environment must be activated with the command `source activate qiime2-2018.2`, giving access to QIIME as well as the tools that it wraps. It is important to note that changes to the command line interface can occur between QIIME2 releases and with plugin updates. The companion GitHub repository will list any necessary changes to the protocol that may arise over time.

3 Methods

3.1 Import Data

In QIIME2, there are two main input/output file types: QIIME artifacts (.qza) and QIIME visualizations (.qzv). QIIME artifacts encapsulate the set of (potentially heterogeneous) data that results from a given step in the pipeline. The artifact also contains a variety of metadata including software versions, command parameters, timestamps, and run times. QIIME visualization files are analysis endpoints that contain the data to be visualized along with the code required to visualize it. Visualizations can be launched in a web browser with the `qiime tools view` command, and many feature interactive elements that facilitate data exploration. Data can be extracted from .qza or .qzv files using the `qiime tools extract` command (*see Note 2*).

If the sequences are in a directory named `sequence_data` (along with a `metadata.yml` and `MANIFEST` file, as described in Subheading 2.1), then the command to import these sequences into a QIIME artifact is:

```
qiime tools import --type  
'SampleData[PairedEndSequencesWithQuality]' --input-path  
sequence_data --output-path reads
```

The data type is specified as `SampleData[PairedEndSequencesWithQuality]`. This is QIIME's way of indicating that there are paired forward/reverse FASTQ sequence files for each sample (*see Note 3*). An output artifact named `reads.qza` will be created, and this file will contain a copy of each of the sequence data files (*see Note 4*).

3.2 Visualize Sequence Quality

The quality profile of sequences can vary depending on sequencing platform, chemistry, target gene, and many other experimental variables. The sequence qualities inform the choices for some of the sequence-processing parameters, such as the truncation parameters of the DADA2 denoising step [2]. QIIME includes an interactive sequence quality plot, available in the “q2-demux” plugin. The following command will sample 10,000 sequences at random and plot box plots of the qualities at each base position:

```
qiime demux summarize --p-n 10000 --i-data reads.qza  
--o-visualization qual_viz
```

The plots, contained within a QIIME visualization .qzv file, can be viewed in a web browser by providing the .qzv file as an argument to the `qiime tools view` command. The sample size should be set sufficiently high to ensure an accurate representation of the qualities, but the run time of this command will increase with the sample size.

Since the `reads.qza` file was created from paired-end reads, the visualization will automatically display the quality distributions for a random sample of both the forward and reverse sequences. With Illumina paired-end data it is expected for there to be a decrease in the quality at the higher base positions. The point at which the quality begins to decrease should inform the truncation parameter used in the subsequent sequence denoising step. The truncation value is provided separately for forward and reverse sequence reads, so it is important to note where the quality decrease occurs for both sets.

3.3 Denoise Sequences With DADA2

As an alternative to OTU clustering at a defined sequence-identity cut-off (e.g., 97%), QIIME2 offers Illumina sequence denoising via DADA2 [2]. The `qiime dada2 denoise-paired` will both merge and denoise paired-end reads. The command has two required parameters: `--p-trunc-len-f` indicates the position at which the forward sequence will be truncated and `--p-trunc-len-r` indicates the position at which the reverse read will be truncated. Optional parameters include `--p-max-ee` which controls the maximum number of expected errors in a sequence before it is discarded (default is 2), and `--p-truncq` which truncates the sequence after the first position that has a quality score equal to or less than the provided value (default is 2). DADA2 requires the primers to be removed from the data to prevent false positive detection of chimeras as a result of degeneracy in the primers. If primers are present in the input sequence files, the optional `--p-trim-left-f` and `--p-trim-left-r` parameters can be set to the length of the primer sequences in order to remove them before denoising. The denoising process outputs two artifacts: a table file and a representative sequence file. The table file can be exported to the Biological Observation Matrix (BIOM) file format (an HDF5-based standard) using the `qiime tools export` command for use in other utilities [11]. The representative sequence file contains the denoised sequences, while the table file maps each of the sequences onto their denoised parent sequence.

3.4 Filter Sequence Table

After denoising with DADA2, many reads may have been excluded because they could not be merged or were rejected during chimera detection. You may wish to exclude any samples that have significantly fewer sequences than the majority. The `qiime feature-table summarize` command produces a visualization file that shows the spread of sequence depths across the samples. Use this visualization to identify a lower bound on the sequence depth and (if desired) filter out low sequence depth samples with the `qiime feature-table filter-samples` command with the `--p-min-frequency` parameter.

3.5 Taxonomic Classification

The QIIME2 software leverages the machine learning Python library scikit-learn to classify sequences [14]. A reference set can be used to train a naïve Bayes classifier which can be saved as a QIIME2 artifact for later re-use. This avoids re-training the classifier between runs, decreasing the overall run time. The QIIME2 project provides a pre-trained naïve Bayes classifier artifact trained against Greengenes (13_8 revision) trimmed to contain only the V4 hypervariable region and pre-clustered at 99% sequence identity [12]. To train a naïve Bayes classifier on a different set of reference sequences, use the `qiime feature-classifier fit-classifier-naive-bayes` command. Other pre-trained artifacts are available on the QIIME2 website (<https://docs.qiime2.org/>). Once an appropriate classifier artifact has been created or obtained, use the `qiime feature-classifier classify` command to generate the classification results.

3.6 Visualize Taxonomic Classifications

The taxonomic profiles of each sample can be visualized using the `qiime taxa barplot` command. This generates an interactive bar plot of the taxa present in the samples, as determined by the taxonomic classification algorithm and reference sequence set used earlier. Bars can be aggregated at the desired taxonomic level and sorted by abundance of a specific taxonomic group or by metadata groupings. Color schemes can also be changed interactively, and plots and legends can be saved in vector graphic format.

3.7 Build Phylogeny

A phylogenetic tree must be created in order to generate phylogenetic diversity measures such as unweighted and weighted UniFrac [9, 10] or Faith’s phylogenetic diversity (PD) [7]. The process is split into four steps: multiple sequence alignment, masking, tree building, and rooting. QIIME2 uses MAFFT for the multiple sequence alignment via the `qiime alignment mafft` command [8]. The masking stage will remove alignment positions that do not contain enough conservation to provide meaningful information (default 40%) and can also be set to remove positions that are mostly gaps. The `qiime alignment mask` command provides this functionality. The tree building stage relies on FastTree (see Note 5) and can be invoked with `qiime phylogeny fasttree` [15]. The final step, rooting, takes the unrooted tree output by FastTree and roots it at the midpoint of the two leaves that are the furthest from one another. This is done using the `qiime phylogeny midpoint-root` command. The end result is a rooted tree artifact file that can be used as input to generate phylogenetic diversity measures.

3.8 Compute Diversity Measures

An array of alpha- and beta-diversity measures can be generated with a single command with QIIME2. The `qiime diversity core-metrics-phylogenetic` command will produce both phylogenetic and non-phylogenetic diversity measures, as well as alpha- and beta-diversity measures. As input, this command

requires a sequence/OTU table, a phylogenetic tree, and a sampling depth for random subsampling. A good value for the sampling depth is the number of sequences contained in the sample with the fewest sequences. It can be found by visualizing the `table_summary_output.qzv` file from the `qiime feature-table summarize` command. The `qiime diversity core-metrics-phylogenetic` command generates Faith's phylogenetic diversity, Shannon diversity, evenness, and observed OTUs (see Note 6) as alpha-diversity measures and weighted/unweighted UniFrac, Bray-Curtis, and Jaccard as beta-diversity measures. For each of the beta-diversity measures, QIIME2 automatically generates principal co-ordinates analysis visualizations. These are three-dimensional visualizations of the high-dimensional pairwise distance (or dissimilarity) matrices. These plots allow the researcher to identify groupings of similar samples at a glance.

3.9 Test for Diversity Differences Between Groups

We can test for significant differences between different sample groups using the `qiime diversity alpha-group-significance` and `qiime diversity beta-group-significance` commands. The alpha-diversity group significance command creates boxplots of the alpha-diversity values and significant differences between groups are assessed with the Kruskal-Wallis test. The beta-diversity command uses boxplots to visualize the distance between samples aggregated by groups specified in the metadata table file. Significant differences are assessed using a PERMANOVA analysis [1] or optionally with ANOSIM [5].

3.10 Alpha Rarefaction

An alpha rarefaction analysis is used to determine if an environment has been sequenced to a sufficient depth. This is done by randomly subsampling the data at a series of sequence depths and plotting the alpha diversity measures computed from the random subsamples as a function of the sequencing depth. A plateau on the rarefaction curve of a given sample provides evidence that the sample has been sequenced to a sufficient depth to capture the majority of taxa. Use the `qiime diversity alpha-rarefaction` command to generate the visualization file.

3.11 Data Provenance

Experimenting with different parameters and plugins can result in an accumulation of output files. It can be quite easy to lose track of which commands and parameters were used for which file. Thankfully, QIIME2 tracks the provenance of each artifact and visualization file. Using the online viewer (<https://view.qiime2.org>), an artifact or visualization file can be imported, presenting a provenance tab that shows the history of the file. The viewer lists each of the parameters used to create the file as well as the run time for the command and a comprehensive list of plugin and software versions. This information is provided not only for the imported file, but also for each of the files that were provided as input, and the input to

those files, and back until the data import command. This means that each QIIME artifact comes bundled with all the knowledge of how it was created.

4 Example

4.1 Retrieve Example Sequence Data

For the example analysis, we will be retrieving data from a recent study on the gut microbiome of the bumblebee, *Bombus pascuorum* [13]. In this study, 106 samples were collected from four different types of bumblebees. Twenty-four samples were collected from larvae (La), 47 from nesting bees (Nu), 18 from foragers that lived in the nest (Fn), 16 from foragers that were collected from the nearby environment (Fo), and one from the queen (Qu). The DNA from the microbiota in the midgut and hindgut of each insect was extracted and amplified using the 515f/806r primer pair (16S rRNA gene V4 region). DNA sequencing was performed on an Illumina MiSeq, resulting in a set of paired-end 16S rRNA gene fragment sequences with an insert size of approximately 254 bp in length.

The data were deposited in the European Bioinformatics Institute Short Read Archive (EBI SRA) at project accession PRJNA313530. The GitHub repository that accompanies this chapter (https://github.com/beiko-lab/mimb_16S) contains a BASH script named `fetchFastq.sh`. This script automatically downloads the raw FASTQ files and sample metadata from the EBI SRA.

4.2 Import Data

The `fetchFastq.sh` script creates a directory named `sequence_data/import_to_qiime/` that contains the forward and reverse FASTQ data files, the `MANIFEST` file, and the `metadata.yml` file. While in the directory containing the `fetchFastq.sh` script, the following command will import the FASTQ files into a QIIME artifact named `reads.qza`:

```
qiime tools import --type
'SampleData[PairedEndSequencesWithQuality]' --input-path
sequence_data/import_to_qiime --output-path reads
```

4.3 Visualize Sequence Quality

To generate visualizations of the sequence qualities, we run the command:

```
qiime demux summarize --p-n 10000 --i-data reads.qza
--o-visualization qual_viz
```

Next, the command `qiime tools view qual_viz.qzv` will launch the visualization in a web browser. Figure 1 shows the quality profile across a sample of 10,000 reverse reads. The quality

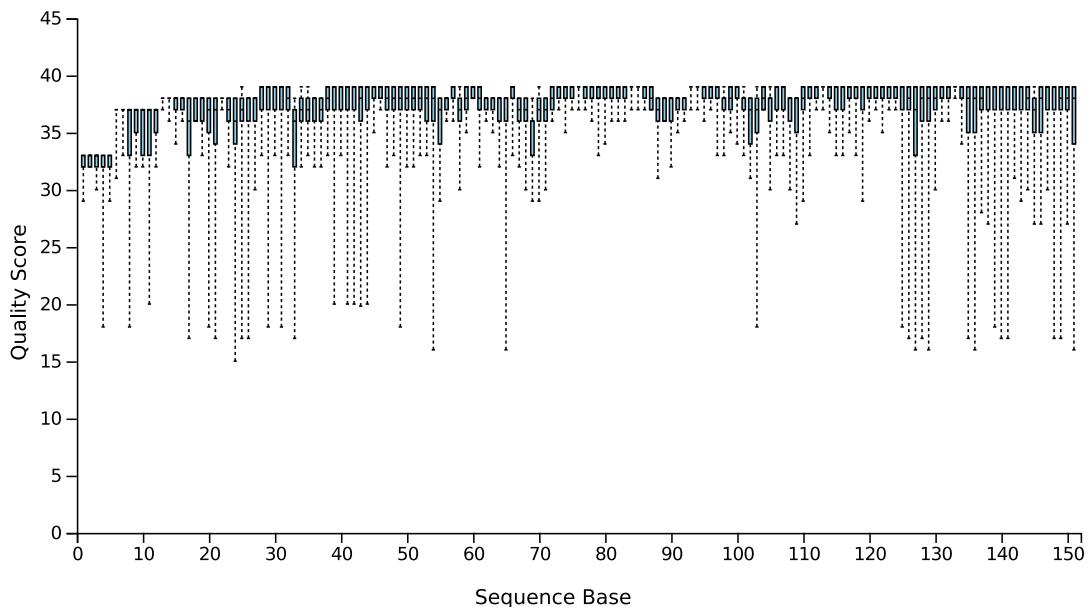


Fig. 1 Quality score box plots sampled from 10,000 random reverse reads

scores begin slightly lower, which is expected from the bases that belong to the primer sequences (see Note 7). We will be removing the primer sequences at the denoising stage. While the median quality scores remain fairly stable, the variance of the quality scores increases around position 130. We will trim some of the bases at the 5' end of the reverse reads during the denoising stage.

4.4 Denoise Sequences With DADA2

The example sequence data are 2×151 bp paired-end reads from an Illumina MiSeq using the 515f/806r primer set [4]. The quality plots (e.g., Fig. 1) indicate that the primers should be trimmed. The forward primer is 19 bp in length and the reverse primer is 20 bp in length, informing our choice for the parameters `--p-trim-left-f` and `--p-trim-left-r`, respectively. We see an increase in the variance of the quality scores for the reverse reads, so we will truncate the reverse reads at position 140. We will not truncate the forward reads, as the same dramatic increase in quality variance is not observed. Therefore, the `--p-trunc-len-f` parameter will be set to 151, and the `--p-trunc-len-r` parameter will be set to 140. At an average amplicon length of 254 bp, trimming the reverse reads by 11 bp would leave an average of 37 bp of overlap. This is sufficient for DADA2, which requires a minimum of 20 bp of overlap for the read merging step. Using `--p-n-threads 4` allows the program to perform parallel computations on 4 threads, and the `--verbose` option displays the DADA2 progress in the terminal.

```
qiime dada2 denoise-paired --i-demultiplexed-seqs reads.qza
--o-table table --o-representative-sequences
representative_sequences --p-trunc-len-f 151 --p-trunc-len-r 140
--p-trim-left-f 19 --p-trim-left-r 20 --p-n-threads 4 --verbose
```

The information printed to the terminal with the `--verbose` option shows a sampling of the sequence counts at each stage of the denoising process. In order to view the number of successfully denoised sequences for each sample, we create a summary of the output table file:

```
qiime feature-table summarize --i-table table.qza
--o-visualization table_summary
```

The visualization file provides detailed information about the denoised sequence counts, including the number of sequences per sample. Sample SRR3203007 had the fewest sequences, with 3615 non-chimeric denoised sequences identified by DADA2. The sample with the second-lowest sequencing depth was SRR3203003 with 42,138 sequences.

4.5 Filter Sequence Table

Since SRR3203007 has a significantly lower sequencing depth than all of the other samples, we will remove it from the table and exclude it from further analysis.

```
qiime feature-table filter-samples --i-table table.qza
--p-min-frequency 5000 --o-filtered-table filtered_table
```

This removes samples with fewer than 5000 sequences, which will remove only sample SRR3203007.

4.6 Taxonomic Classification

First, we must download the trained naïve Bayes classifier artifact. We will fetch this from the QIIME website with the command `wget`:

```
wget
https://data.qiime2.org/2018.2/common/gg-13-8-99-515-806-nb-
classifier.qza
```

This classifier artifact is trained on Greengenes August 2013 revision, trimmed to the V4 hypervariable region using primers 515f/806r, and clustered at 99% sequence identity. We then instruct QIIME to classify using this classifier artifact and the `scikit-learn` Python library:

```
qiime feature-classifier classify-sklearn --i-classifier
gg-13-8-99-515-806-nb-classifier.qza --i-reads
representative_sequences.qza --o-classification taxonomy
```

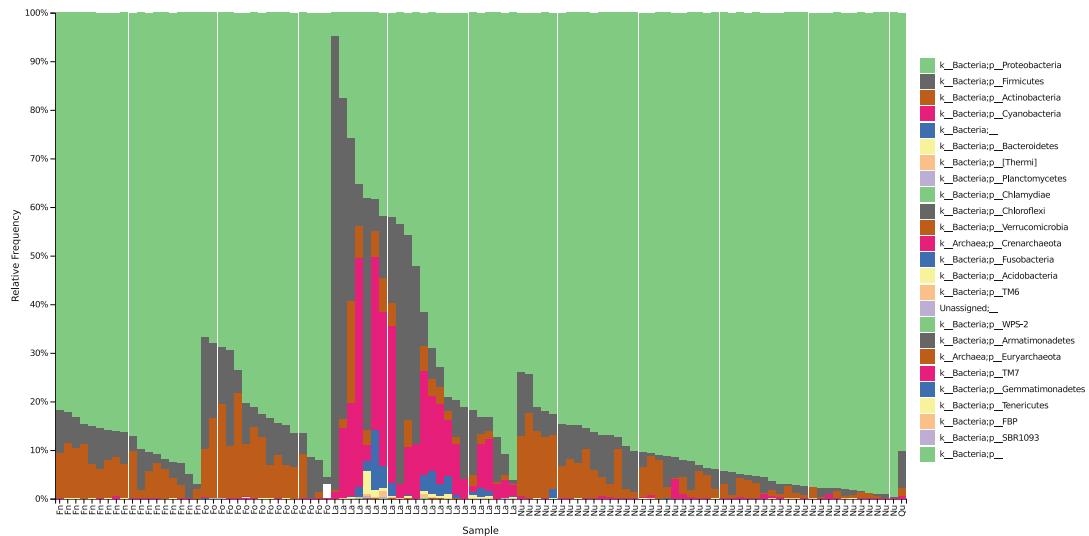


Fig. 2 Taxonomic profiles for the bumblebee gut samples at phylum level. Samples are sorted first by bee type and then by the abundance of the phylum *Proteobacteria*. Bee types are larval samples (La), nesting bees (Nu), foragers from the nest (Fn), foragers from the environment (Fo), and the queen (Qu)

4.7 Visualize Taxonomic Classifications

QIIME can generate interactive bar plots of the taxonomic profiles. The profiles can be sorted by metadata categories, so we have to provide the tab-separated metadata file which was generated by the data download script. The metadata file for the example data is located in `sequence_data/METADATA.txt`.

```
qiime taxa barplot --i-table filtered_table.qza --i-taxonomy taxonomy.qza --m-metadata-file sequence_data/METADATA.txt --o-visualization taxa-bar-plots
```

The resulting `taxa-bar-plots.qzv` visualization file can be launched with the command `qiime tools view taxa-bar-plots.qzv`. Figure 2 shows taxonomic profiles sorted by `bee_type` and then by the relative abundance of the phylum *Proteobacteria*.

4.8 Build Phylogeny

We build a phylogeny based on the 16S rRNA gene fragments in the four-step process described in Subheading 3.7. First, we align the denoised sequences with MAFFT.

```
qiime alignment mafft --i-sequences representative_sequences.qza --o-alignment aligned_representative_sequences
```

Next, we mask the uninformative positions.

```
qiime alignment mask --i-alignment aligned_representative_sequences.qza --o-masked-alignment masked_aligned_representative_sequences
```

We build the phylogeny using the FastTree method.

```
qiime phylogeny fasttree --i-alignment  
masked_alignedRepresentativeSequences.qza --o-tree unrooted_  
tree
```

Finally, we root the tree at the midpoint, producing the `rooted_tree.qza` artifact that will be used as input to generate phylogenetic-diversity measures.

```
qiime phylogeny midpoint-root --i-tree unrooted_tree.qza  
--o-rooted-tree rooted_tree
```

4.9 Compute Diversity Measures

We can use a single command to generate a series of phylogenetic and non-phylogenetic diversity measures. In order to compare samples with uneven sequencing depth, QIIME2 randomly subsamples or “rarefies” the sequences present in each environmental sample, at a user-specified depth. After filtering, our sample with the fewest sequences is SRR3203003 with 42,138 sequences. The smallest number of sequences in a given sample can be used as the subsampling depth, but here we will go slightly lower and use a depth of 41,000.

```
qiime diversity core-metrics-phylogenetic --i-phylogeny  
rooted_tree.qza --i-table filtered_table.qza --p-sampling-depth  
41000 --output-dir diversity_41000 --m-metadata-file  
sequence_data/METADATA.txt
```

This command will generate several QIIME artifact files that contain the Faith’s phylogenetic diversity, observed OTUs, Shannon diversity, and evenness alpha-diversity measures for each sample. It also generates beta-diversity distance matrices for the Bray-Curtis, Jaccard, unweighted UniFrac, and weighted UniFrac measures, as well as visualizations of the principal co-ordinates analyses based on these distance measures. These visualization files have the filename suffix `_emperor.qzv` and when viewed will display a three-dimensional ordination plot. A static example of the interactive principal co-ordinates visualization is shown in Fig. 3.

4.10 Test for Diversity Differences Between Groups

We will test for significant differences in the microbial community diversity measures of the bee types. QIIME2 will perform the statistical tests for each of the sample groupings present in the metadata file. To run the tests for Faith’s phylogenetic diversity, we run:

```
qiime diversity alpha-group-significance --i-alpha-diversity  
diversity_41000/faith_pd_vector.qza --m-metadata-file  
sequence_data/METADATA.txt --o-visualization  
diversity_41000/alpha_PD_significance
```

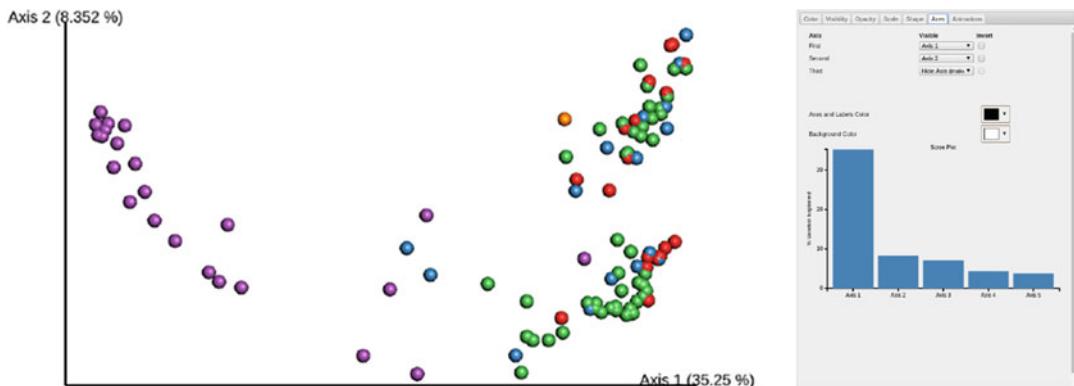


Fig. 3 A two-dimensional principal co-ordinates analysis ordination of the bee gut samples based on unweighted UniFrac distances. Samples are colored by bee type (blue: foragers from the environment, red: foragers from the nest, green: nesting bees, purple: larvae, orange: queen). Legend is visible in the “Color” tab of the interactive visualization. The “Axis” tab, shown here, allows any of the principal co-ordinates to be selected, allows the background and font colors to be changed, and shows the Scree plot that indicates the proportion of the data variation captured by each principal co-ordinate

This runs all-group and pairwise Kruskal-Wallis tests, a non-parametric analysis of variance. The visualization file, `alpha_PD_significance.qzv`, presents boxplots and test statistics for each metadata grouping. In this example data set, the larval samples had a significantly higher phylogenetic diversity value compared with each of the other bee types ($p < 0.001$).

The test for beta-diversity group distances can be performed with PERMANOVA (the default method) or ANOSIM. The `qiime diversity beta-group-significance` command computes only one metadata grouping at a time, so to test the differences between bee types we have to supply the appropriate column name from the metadata file:

```
qiime diversity beta-group-significance --i-distance-matrix
diversity_41000/bray_curtis_distance_matrix.qza --m-metadata-file
sequence_data/METADATA.txt --m-metadata-column bee_type
--o-visualization diversity_41000/beta_bray_beetype_significance
```

This runs an all-group PERMANOVA analysis on the Bray-Curtis dissimilarity measures for each bee type. For this data set, the PERMANOVA test reveals that the five bee types have significant differences in their community compositions. The pairwise distance boxplots show that this is largely driven by the larval samples, an observation corroborated by the relatively robust clustering of larval samples visible in the principal co-ordinates ordination (Fig. 3).

4.11 Alpha Rarefaction

Our final analysis is to create alpha rarefaction curves in order to determine if the samples have been sequenced to a sufficient depth. The qiime diversity alpha-rarefaction command will generate rarefaction curves based on the Shannon diversity and observed OTUs measures by default, and will additionally generate phylogenetic diversity-based curves if the phylogenetic tree created above is provided using the `--i-phylogeny` parameter. The desired alpha-diversity measure is selected interactively after the visualization file is launched.

```
qiime diversity alpha-rarefaction --i-table filtered_table.qza
--p-max-depth 41000 --o-visualization
diversity_41000/alpha_rarefaction.qzv --m-metadata-file
sequence_data/METADATA.txt --i-phylogeny rooted_tree.qza
```

Figure 4 (see Note 8) shows the alpha rarefaction curves with the results average by bee type. We can derive a few insights from this table. The first is that each of the bee type categories appear to plateau. Although the diversity measure does generally continue to increase as a function of the sequencing depth, the accumulation slows significantly, suggesting that we have sufficient sample sequence depth to have captured the majority of taxa present in the sample. New taxa that may be picked up by additional sequencing effort are likely to be either rare microorganisms or the result of sequencing error. The second thing we can learn from Fig. 4 is that the larval samples have a significantly higher phylogenetic diversity than the other bee types. This result agrees with the Kruskal-Wallis test performed in Subheading 4.10.

4.12 Data Provenance

Marker-gene analyses have the potential to generate many output files. Data-provenance tracking ensures that we do not lose track of how each file was generated. If you forget or are unsure how a file

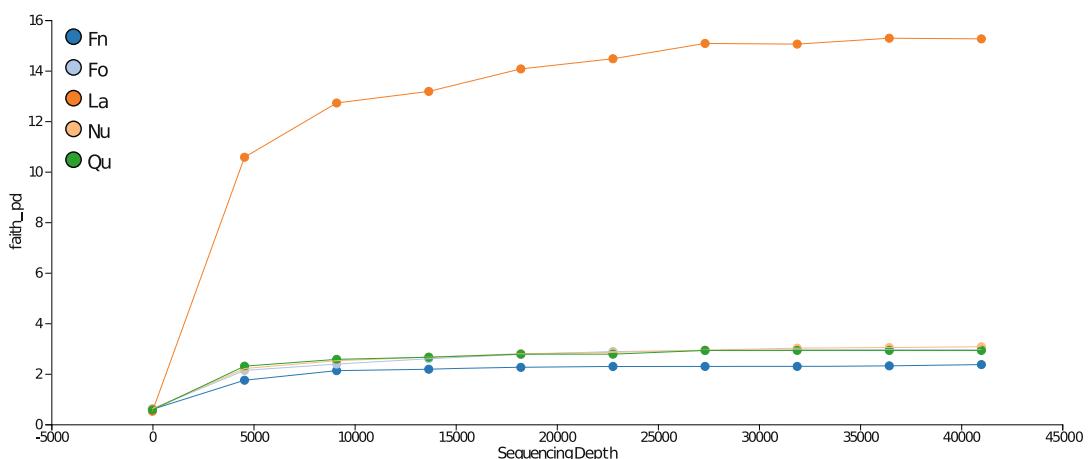


Fig. 4 Rarefaction curves based on the phylogenetic diversity measure, with samples aggregated by bee type

was generated, it can be imported to the viewer at <https://view.qiime2.org> and the history is available in the “Provenance” tab. For example, Fig. 5 shows the provenance graph for the unweighted UniFrac principal co-ordinates file generated in Subheading 4.9, `unweighted_unifrac_emperor.qzv`. Clicking on the circles in the graph reveals the output file type, format, and unique identifier. Clicking on the blocks reveals the command that generated the output file(s). The arrows show the flow of output files from one step as input to the next.

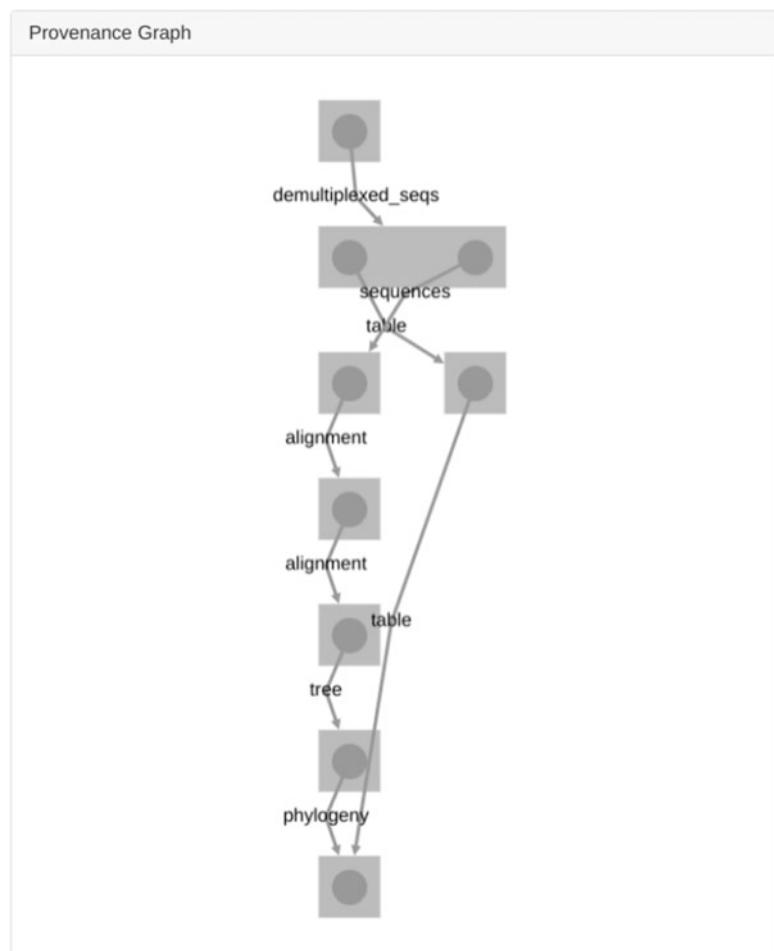


Fig. 5 Data provenance graph for the `unweighted_unifrac_emperor.qzv` file generated in the example analysis

5 Notes

1. Most recent Illumina sequence data is produced with the CASAVA pipeline 1.8+ which uses the PHRED+33 encoding indicated by the “metadata.yml” text `{phred-offset: 33}`. If your FASTQ data was generated by an older version of the CASAVA pipeline, the qualities will be in PHRED+64 format. In this case, the “metadata.yml” text should be changed to `{phred-offset: 64}`.
2. The .qza and .qzv files are simply re-named ZIP files. The file extension can be changed from .qza or .qzv to .zip and extracted with any ZIP file decompression tool on systems where QIIME is not installed or available, such as Windows PCs.
3. A list of the data types in QIIME2 (2018.2 distribution) is available at <https://docs.qiime2.org/2018.2/semantic-types/>. Single-end FASTQ files correspond to the data type “`SampleData[SequencesWithQuality]`.”
4. Preserving copies of the raw input data within an artifact enhances research reproducibility by allowing the source data to be easily identified and ensures that the data are more closely coupled to the analysis. Just keep an eye on your hard drive usage!
5. The QIIME2 2018.2 distribution comes with FastTree version 2.1.10 compiled with double precision. This mitigates issues with resolving short branch lengths that could occur when using the version of FastTree that was distributed with earlier versions of the QIIME software suite.
6. Even though we are not using the common 97% OTU clustering approach, the denoised sequences from DADA2 can still be considered operational taxonomic units, so the name “observed OTUs” is still appropriate for this diversity measure. However, to avoid confusion, you may wish to describe the “observed OTUs” measure as “observed taxa” or “observed representative sequences.”
7. The lower quality scores at the beginning of each read are caused by the homogeneity of the primer sequences. This makes it difficult for the Illumina sequencer to properly identify clusters of DNA molecules [6].
8. Several of the QIIME visualizations do not have a clear way to export the plots in high-quality scalable vector graphic (SVG) format. The alpha rarefaction plot is one such visualization. A simple screenshot will result in a raster graphic that may have sufficient resolution for inclusion in a research article. The New York Times’ SVG Crowbar utility (<https://nytimes.com/crowbar>).

github.io/svg-crowbar/) can be used to extract many of these plots in SVG format. These files can then be easily manipulated using Inkscape or similar vector graphic-editing programs.

References

1. Anderson M (2005) PERMANOVA: a FORTRAN computer program for permutational multivariate analysis of variance, 24th edn. Department of Statistics, University of Auckland, Auckland
2. Callahan BJ, McMurdie PJ, Rosen MJ, Han AW, Johnson AJA, Holmes SP (2016) DADA2: high-resolution sample inference from Illumina amplicon data. *Nat Methods* 13(7):581
3. Caporaso JG, Kuczynski J, Stombaugh J, Bittinger K, Bushman FD, Costello EK, Fierer N, Pena AG, Goodrich JK, Gordon JI et al (2010) QIIME allows analysis of high-throughput community sequencing data. *Nat Methods* 7(5):335
4. Caporaso JG, Lauber CL, Walters WA, Berg-Lyons D, Lozupone CA, Turnbaugh PJ, Fierer N, Knight R (2011) Global patterns of 16S rRNA diversity at a depth of millions of sequences per sample. *Proc Natl Acad Sci USA* 108:4516–4522
5. Chapman M, Underwood A (1999) Ecological patterns in multivariate assemblages: information and interpretation of negative values in ANOSIM tests. *Mar Ecol Prog Ser* 180:257–265
6. Fadrosh DW, Ma B, Gajer P, Sengamalay N, Ott S, Brotman RM, Ravel J (2014) An improved dual-indexing approach for multiplexed 16S rRNA gene sequencing on the Illumina MiSeq platform. *Microbiome* 2(1):6
7. Faith DP (1992) Conservation evaluation and phylogenetic diversity. *Biol Conserv* 61 (1):1–10
8. Katoh K, Standley DM (2013) MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Mol Biol Evol* 30(4):772–780
9. Lozupone C, Knight R (2005) UniFrac: a new phylogenetic method for comparing microbial communities. *Appl Environ Microb* 71 (12):8228–8235
10. Lozupone CA, Hamady M, Kelley ST, Knight R (2007) Quantitative and qualitative β diversity measures lead to different insights into factors that structure microbial communities. *Appl Environ Microb* 73(5):1576–1585
11. McDonald D, Clemente JC, Kuczynski J, Rideout JR, Stombaugh J, Wendel D, Wilke A, Huse S, Hufnagle J, Meyer F et al (2012) The Biological Observation Matrix (BIOM) format or: how I learned to stop worrying and love the ome-ome. *GigaScience* 1(1):7
12. McDonald D, Price MN, Goodrich J, Nawrocki EP, DeSantis TZ, Probst A, Andersen GL, Knight R, Hugenholtz P (2012) An improved Greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea. *ISME J* 6(3):610
13. Parmentier A, Meeus I, Nieuwerburgh F, Deforce D, Vandamme P, Smagghe G (2018) A different gut microbial community between larvae and adults of a wild bumblebee nest (*Bombus pascuorum*). *Insect Sci* 25 (1):66–74
14. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V et al (2011) Scikit-learn: machine learning in Python. *J Mach Learn Res* 12 (Oct):2825–2830
15. Price MN, Dehal PS, Arkin AP (2010) FastTree 2-approximately maximum-likelihood trees for large alignments. *PLoS One* 5(3): e9490
16. Rideout JR, Chase JH, Bolyen E, Ackermann G, González A, Knight R, Caporaso JG (2016) Keemei: cloud-based validation of tabular bioinformatics file formats in Google Sheets. *GigaScience* 5(1):27



Chapter 9

Processing a 16S rRNA Sequencing Dataset with the Microbiome Helper Workflow

Gavin M. Douglas, André M. Comeau, and Morgan G. I. Langille

Abstract

Sequencing microbiome samples has recently become a fast and cost-effective method to taxonomically profile communities. The growing interest in analyzing microbial sequencing data has attracted many new researchers to the field. Here, we present a straightforward bioinformatic pipeline that aims to streamline the processing of 16S rRNA sequencing data. This workflow is part of the larger project called Microbiome Helper (Comeau et al. mSyst 2:e00127-16, 2017), which includes other bioinformatic workflows, tutorials, and scripts available here: https://github.com/mlangill/microbiome_helper/wiki.

Key words Microbiome Helper, 16S rRNA gene, VirtualBox image, Tutorial, Standard operating procedure, Amplicon sequencing

1 Introduction

Identifying prokaryotic operational taxonomic units (OTUs) based on the DNA sequence of the 16S rRNA gene (16S) has long been a method for taxonomically profiling microbial communities [1]. Several bioinformatic pipelines have been proposed to process this sequencing data, the most popular of which are mothur [2] and “Quantitative Insights into Microbial Ecology” (QIIME) [3]. However, the many different possible workflows and parameter choices often leave new users unsure how to proceed. We developed a set of tutorials, straightforward workflows, and scripts called Microbiome Helper [4] to help new users get acquainted with standard microbial sequencing analyses. Microbiome Helper includes information on processing shotgun metagenomic and other microbial amplicon sequencing data, which you can find on our website. We have also created a virtual image, which will allow users to start processing sequencing data with little or no configuration. The aim of this chapter is to summarize the full pipeline for processing 16S data starting from raw paired-end (PE) reads, which is based on the 16S tutorial on our website.

2 Materials

2.1 Microbiome Helper VirtualBox Image

All the materials required for running the below workflow are preinstalled on our Ubuntu (v16.04) VirtualBox image (VBox), which can be downloaded here: https://github.com/mlangill/microbiome_helper/wiki/Microbiome-Helper-Virtual-Box. Installation instructions and requirements are on this page. This VBox comes as an Open Virtualization Application (OVA) file, which is typically more than 8 gigabytes (GB) depending on the version. While importing this file, your system will require roughly twice this much disk space, so make sure you have enough space available. After installing the VBox you can delete the OVA file to save space.

You can configure this VBox to suit your host computer. If you have multiple cores or 8 GB or more of random access memory (RAM) then you should increase the VBox's resource allocation through the *Settings* window in the VBox manager while the VBox is shutdown. You can also set up a shared folder with your local computer to make it easier to transfer files between the two environments. One downside of using this virtual environment is that whenever updates are made you will need to manually delete your current version and start over with the new version if you want the latest features (however, an updating tool will be forthcoming to circumvent this problem).

If you have never used command-line Linux, then you should look up some basic online tutorials (such as the one at: <http://korflab.ucdavis.edu/bootcamp.html>) once the VBox is installed. You can access the command-line by opening the *Terminal* application on the left sidebar in the VBox environment. Before proceeding with the below workflow (*see Subheading 3*), you will need to be familiar with basic command-line commands (e.g., how to navigate directories, how to copy/move files).

2.2 Microbiome Helper GitHub Repository

The scripts referred to below are part of the Microbiome Helper GitHub repository (https://github.com/LangilleLab/microbiome_helper/wiki). The wiki site for this repository includes tutorials and workflows for how to use these scripts and other resources for processing data. Note that the GitHub repository is already installed on our VBox, but you can get the latest version on a different system by running:

```
git clone https://github.com/LangilleLab/microbiome_helper.git
```

Note there are multiple dependencies that you will need to properly install if not using the VBox. The scripts in this repository are mainly “wrapper” scripts, which means they are used to read in input and then run other programs. These scripts are intended to simplify running the same commands on multiple samples while

keeping track of log information. GNU Parallel [5] is a straightforward alternative approach to multiprocessing these programs (*see Note 1*).

2.3 16S rRNA Gene Databases

Databases of 16S sequences are required for both the chimera checking and OTU-picking steps, which are available on our wiki site. For chimera checking, we use the Ribosomal Database Project's version 16 training sequences [6]. We use the default database for OTU picking with QIIME, which is Greengenes 13_8 [7]. Importantly, if you want to process 18S rRNA gene or internal transcribed spacer 2 (ITS2) amplicon sequencing data you will need to use different databases (*see Note 2*).

2.4 Pre-processing Software

We use several tools for the pre-processing steps of our pipeline, which are outlined below. We have written wrapper scripts to enable running most of these tools in parallel (*see Subheading 3*).

1. FastQC (<https://www.bioinformatics.babraham.ac.uk/projects/fastqc/>) is a useful tool for performing quality control on your raw reads. Some of the basic statistics returned by this tool include the number, length, and base quality distribution of your reads.
2. Paired-End reAd mergeR (PEAR) [8] is an efficient tool for stitching (i.e., merging) PE reads. This tool keeps track of all reads, even those that were not successfully stitched, which allows users to easily decide whether it would be better to proceed with stitched or unstitched reads.
3. FASTX-Toolkit (http://hannonlab.cshl.edu/fastx_toolkit/) is a collection of tools for pre-processing sequence files. We use this tool to filter reads by base quality score.
4. BBMap (<https://sourceforge.net/projects/bbmap/>) is a tool focused on short read aligning, but also contains many useful pre-processing features. We use this tool to filter reads by a set length and to check for matches to primer sequences.
5. VSEARCH [9] is the open-source alternative to USEARCH [10], which supports both de novo and reference-based chimera checking, based on the UCHIME algorithm [11]. We perform reference-based chimera checking in our standard workflow.

2.5 Open-Reference OTU Picking

We typically run open-reference OTU picking at 97% identity, which involves first clustering reads against a database of reference sequences and then clustering the remaining reads against themselves [12]. We use SortMeRNA [13] for the reference-based and SUMACLUST (<https://git.metabarcoding.org/obitools/sumatra/wikis/home>) for the de novo clustering steps. Both programs

are wrapped into QIIME OTU picking scripts. Currently only SortMeRNA v2.0-dev time stamped 29/11/2014 works with the QIIME pipeline.

2.6 Statistical Analyses and Visualization

QIIME is used to generate basic alpha-diversity and UniFrac beta-diversity [14] values from the OTU table. Visualizations of these metrics can be generated with the EMPeror package [15], which can be displayed and explored in your web browser. STAMP [16] is used for identifying individual taxa that have different relative abundances between groups and performing statistical tests on these.

3 Methods

3.1 Tutorial and Example Dataset

Our online 16S tutorial aims to teach the basics of the workflow, which shows example commands for each of the steps discussed below. You can access this tutorial on our website. You can download this dataset from the tutorial page or with these commands (if you have *wget* installed):

```
wget https://www.dropbox.com/s/r2jqqc7brxg4jhx/16S_chemerinTutorial.zip?dl=1 -O
16S_chemerinTutorial.zip; unzip 16S_chemerinTutorial.zip
```

3.2 Check Data Quality

The first step whenever processing sequencing data should be to run quality control on the raw data. However, to perform quality control, it is important to understand the format of your sequence data. FASTQ format is currently the most common format for raw sequence data, which is what we provide in the above link. Each read in a FASTQ file is represented over four lines, which correspond to (1) the read identifier, (2) the sequence, (3) a line typically containing just “+”, and (4) the base quality scores. For PE sequencing data, there will be two FASTQs per sample, which correspond to the forward (“R1”) and reverse (“R2”) reads. FastQC is a useful tool to summarize these files, which will produce a summary of the quality of each file that you can view in your web browser. To generate a summary of all sequence data, first create a directory (*fastqc_out_combined* in our example) using the command “*mkdir fastqc_out_combined*”. Then execute the following command to create the summary file in this directory:

```
cat *fastq | fastqc stdin -o fastqc_out_combined
```

This command relies on the UNIX pipe “|” command to chain multiple actions on a single line. Although this combined summary is useful, you should not rely on the dataset-wide quality report alone. Individual FastQC reports will reveal outliers in your data, such as issues with specific samples or sequencing runs.

3.3 Stitch Paired-End Reads

To stitch PE reads together run this command in your working directory:

```
run_pear.pl -p 4 -o stitched_reads fastq/*
```

This command will output four FASTQs per sample: the assembled reads, discarded reads, and unassembled forward and reverse reads separately. These output files are created in a new folder named *stitched_reads*. The *-p* option in the above command specifies that PEAR will be run over four cores. Discarded reads are those that fail PEAR's quality filters. It is important to check the assembly rate for all your samples, which will be summarized in the file *pear_summary_log.txt* by default. Note that it is not always possible to stitch PE reads since it depends on both your read lengths and the insert size. Looking at the assembly rates for your samples should help you decide whether it is better to proceed with your stitched reads or your forward and/or reverse reads separately. Outlier samples can also be identified at this step since low-quality reads will have a lower assembly rate.

3.4 Read Filtering

A key step when pre-processing sequencing data is to quality filter the reads. The exact cutoffs should be chosen based on the FASTQC reports generated above. Unique to amplicon sequencing is that primer sequences will be found at the start of both the forward and reverse reads. Confirming that the expected primer sequences are present is a useful check that the data is of adequate quality and in the correct orientation. The specified primers below are for the V6–V8 variable regions (note that the reverse complement of the reverse primer should be specified), which should be substituted with the primers you used. BBMap and the FASTX Toolkit are two bioinformatic tools that can quality filter FASTQs. Run this command to run these tools in parallel over the successfully stitched FASTQs:

```
read_filter.pl -f ACGCGHNRAACCTTACC -r TTGYACWCACYGCCGT --primer_check both --thread 4 -q 30 -p 90 -l 400 stitched_reads/*.assembled.*
```

This program requires the minimum base quality (*-q*) over a given proportion of bases (*-p*) and the minimum read length (*-l*) to be specified. You may also need to point to where BBMap is installed (*--bbmap* option, */usr/local/prg/bbmap* is checked by default). As mentioned above, users can also check whether the first and last sequences in each read match the primers used for PCR amplification (*--primer_check [none|both|forward]*). By default, primer sequences are not checked. The user can also select to check forward sequences only if reads are unstitched. If primer sequences are being checked, the user also has the option to trim off these sequences (*--primer_trim* flag), although there is some disagreement about whether this is necessary (see Note 3). Since this program wraps several tools and different commands, the

intermediate files are removed for simplicity. However, users can also specify that these intermediate files should be kept (*--keep* flag), which can be helpful for troubleshooting. Again, this command runs the filtering programs in parallel over four cores, as specified by the *-thread* option. The outputs of this command are FASTQs containing the filtered reads in the directory *filtered_reads* by default and a log-file containing the numbers of reads dropped at each step called *read_filter_log.txt* by default. Once the reads are filtered they should be converted to FASTA format with this command:

```
run_fastq_to_fasta.pl -p 4 -o fasta_files filtered_reads/*fastq
```

This command again runs over four cores as specified by the *-p* option. The output FASTA files will be found in the *fasta_files* directory.

3.5 Chimera Checking

Chimeras are sequences that contain DNA from multiple source templates. These artifacts are of concern when processing amplicon sequencing data since they commonly occur during PCR amplification of similar sequences. A common method for identifying chimeras is to identify sequences that match two or more parent sequences in a database. High-quality chimera-free 16S sequences are found in the database file *RDP_trainset16_022016.fa*, which are used as the reference set below. Run this command to remove chimeras from your dataset:

```
chimera_filter.pl --thread 4 -type 1 -db
/home/shared/rRNA_db/RDP_trainset16_022016.fa fasta_files/*
```

The chimera-filtered FASTQs will be output to the folder *non_chimeras* by default. A summary of the reads called as non-chimeric, chimeric, and borderline (denoted “unclear”) will be output in the file *chimera_filter_log.txt* by default. Your choice of database will likely have the largest effect on chimera checking (see Subheading 2.3); however, there are several other options to be aware of as well. Firstly, you can specify whether you want to keep only sequences that are clearly non-chimeric (*--type 1*) or to keep borderline sequences, which have marginal chimera scores, as well (*--type 0*). Restricting to clearly non-chimeric sequences is more conservative, but you may want to include borderline sequences as well depending on how much data you are losing. As for read filtering, you can optionally keep intermediate files (*--keep* flag) and the jobs are run over four cores as specified by the *-thread* option.

3.6 Open-Reference OTU Picking

Before running OTU picking, a QIIME-formatted mapping file needs to be created with this command:

```
create_qiime_map.pl non_chimeras/* > map.txt
```

The output file, *map.txt*, is the minimal QIIME-formatted map file required for the subsequent commands, but users should add sample information as additional columns for data analysis (see Subheading 3.8). A combined FASTA can then be generated for all samples based on this map file with this QIIME command:

```
add_qiime_labels.py -i non_chimeras/ -m map.txt -c FileInput -o combined.fasta
```

The combined FASTA file will be in the *combined.fasta* directory. Finally, before running OTU picking you can specify non-default options in a parameter file. Options in this file are given with the format *(script name):(option name) setting*. Note the script name should not include an extension. For example, to set four threads for the *pick_otsu.py* program you would add this line to the file: *pick_otsu:threads 4*. Typically, we set that number of threads and the *sortmerna_coverage* option to be 0.8 for *pick_otsu.py*, which means that 80% of a query sequence needs to match a reference sequence for it to be considered a match. The Greengenes 16S rRNA gene database will be used by default and does not need to be specified. If you will be re-running OTU picking a few times you can save time by indexing the 16S rRNA gene database with SortMeRNA and then passing *pick_otsu.py* the *sortmerna_db* option to avoid re-indexing this database each time (assumed to be in */home/shared/pick_otu_indexdb_rna/* below). These options can quickly be added to the parameter file with these commands (note that *>>* means append to the output file):

```
echo "pick_otsu:threads 4" >> clustering_params.txt
echo "pick_otsu:sortmerna_coverage 0.8" >> clustering_params.txt
echo "pick_otsu:sortmerna_db /home/shared/pick_otu_indexdb_rna/97_otsu" >>
clustering_params.txt
```

After preparing the parameter file, OTU picking is run with this command:

```
pick_open_reference_otsu.py -i $PWD/combined.fasta/combined_seqs.fna -o
$PWD/clustering/ -p $PWD/clustering_params.txt -m sortmerna_sumaclust -s 0.1 -v --
min_otu_size 1
```

Many output files will be created by this script, which will be found in the *clustering* directory. The key output file is the resulting OTU table (*otu_table_mc1_w_tax_no_pynast_failures.biom*). Note that *\$PWD* is a bash variable for the full path of the current working directory. The *-s 0.1* option indicates that 10% percent of reads that failed to be classified should be sub-sampled to create a new reference database for de novo clustering. The *-min_otu_size* option specifies that all OTUs, including singletons (i.e., OTUs called by a single read), will be outputted.

3.7 Processing the OTU Table

After calling OTU picking, all OTUs that are called by fewer than 0.1% of the total reads are removed. This cutoff is based on the proportion of bleed-through expected to occur between Illumina MiSeq sequencing runs (as reported by Illumina). The filtering is performed by this command:

```
remove_low_confidence_otus.py -i
$PWD/clustering/otu_table_mc1_w_tax_no_pynast_failures.biom -o
$PWD/clustering/otu_table_high_conf.biom
```

The filtered BIOM table will be *clustering/otu_table_high_conf.biom*. After filtering out these OTUs, a reasonable read depth for rarefaction must be chosen. This cutoff should be chosen after considering the summary of per-sample depths, which is generated by this command:

```
biom summarize-table -i clustering/otu_table_high_conf.biom -o
clustering/otu_table_high_conf_summary.txt
```

The per-sample depths will be sorted from lowest to highest in *clustering/otu_table_high_conf_summary.txt*. Once this value is chosen, you should create the folder *final_otu_tables* and rarefy the OTU table with this QIIME command, which subsamples the number of reads for each sample to the same number:

```
single_rarefaction.py -i clustering/otu_table_high_conf.biom -o
final_otu_tables/otu_table.biom -d X
```

The rarified OTU table will be *final_otu_tables/otu_table.biom*. Note that the *-d* option specifies the rarefaction depth; the above *X* should be replaced with your chosen depth.

3.8 Statistical Analyses and Visualization

The read depth choice for rarefaction can be chosen based on comparing your samples' rarefaction curves. Viewing rarefaction curves helps users decide whether increased sequencing depth would have resulted in many more OTUs being identified. You can create these curves by using this QIIME command:

```
alpha_rarefaction.py -i final_otu_tables/otu_table.biom -o plots/alpha_rarefaction_plot
-t clustering/rep_set.tre -m map.txt --min_rare_depth X --max_rare_depth Y --num_steps
Z
```

The output of this command are interactive plots that you can explore in your web browser, which will be found in the *plots/alpha_rarefaction_plot* folder. Each of your samples will be rarified from depths of *-min_rare_depth* to *-max_rare_depth* over *-num_steps* steps. These parameter settings will depend on your own data (*X*, *Y*, and *Z* are placeholders), but typically you would set the

`-max_rare_depth` argument to be the rarefaction depth specified by the `single_rarefaction.py` command.

Plots displaying the Principal Coordinates Analysis summarizing measures of beta-diversity in your samples can also quickly be generated with this QIIME command:

```
beta_diversity_through_plots.py -mmap.txt -t clustering/rep_set.tre -i
final_otu_tables/otu_table.biom -o plots/bdiv_otu
```

This command also outputs interactive plots that you can open in your browser to compare the UniFrac distances between your samples, which are found in the `plots/bdiv_otu` folder. For both the alpha- and beta-diversity scripts, you can compare samples by your groupings of interest by providing the mapping file with the `-m` option and the tree of all representative sequences built during OTU picking is specified with the `-t` option. For identifying specific taxa that have differing relative abundances between your samples you can convert your OTU table to STAMP format with this command:

```
biom_to_stamp.py -m taxonomy final_otu_tables/otu_table.biom
>final_otu_tables/otu_table.spf
```

The resulting table can be read into STAMP along with your mapping file. Example plots generated by this pipeline are shown in Fig. 1.

4 Notes

1. *GNU Parallel* is a helpful program for multiprocessing repetitive commands. Briefly, this tool enables the same commands to be run on large numbers of files while providing simple syntax for specifying outputs for each file. There are many options available for this tool, which gives users sophisticated control. Some of the most helpful options include specifying how many jobs can be run simultaneously (or alternatively what percent of a system’s CPUs can be used), provide expected time until all jobs finish, and set a maximum memory usage threshold. You can read more about this tool on the GNU website: <https://www.gnu.org/software/parallel/>. We have also written a brief tutorial that demonstrates how *GNU Parallel* can be integrated with bioinformatic pipelines on the Microbiome Helper website.
2. For identifying chimeras in 18S rRNA gene (18S) sequencing data (i.e., eukaryotic profiling), we recommend using a sequence database provided by SILVA [17]. For 18S OTU

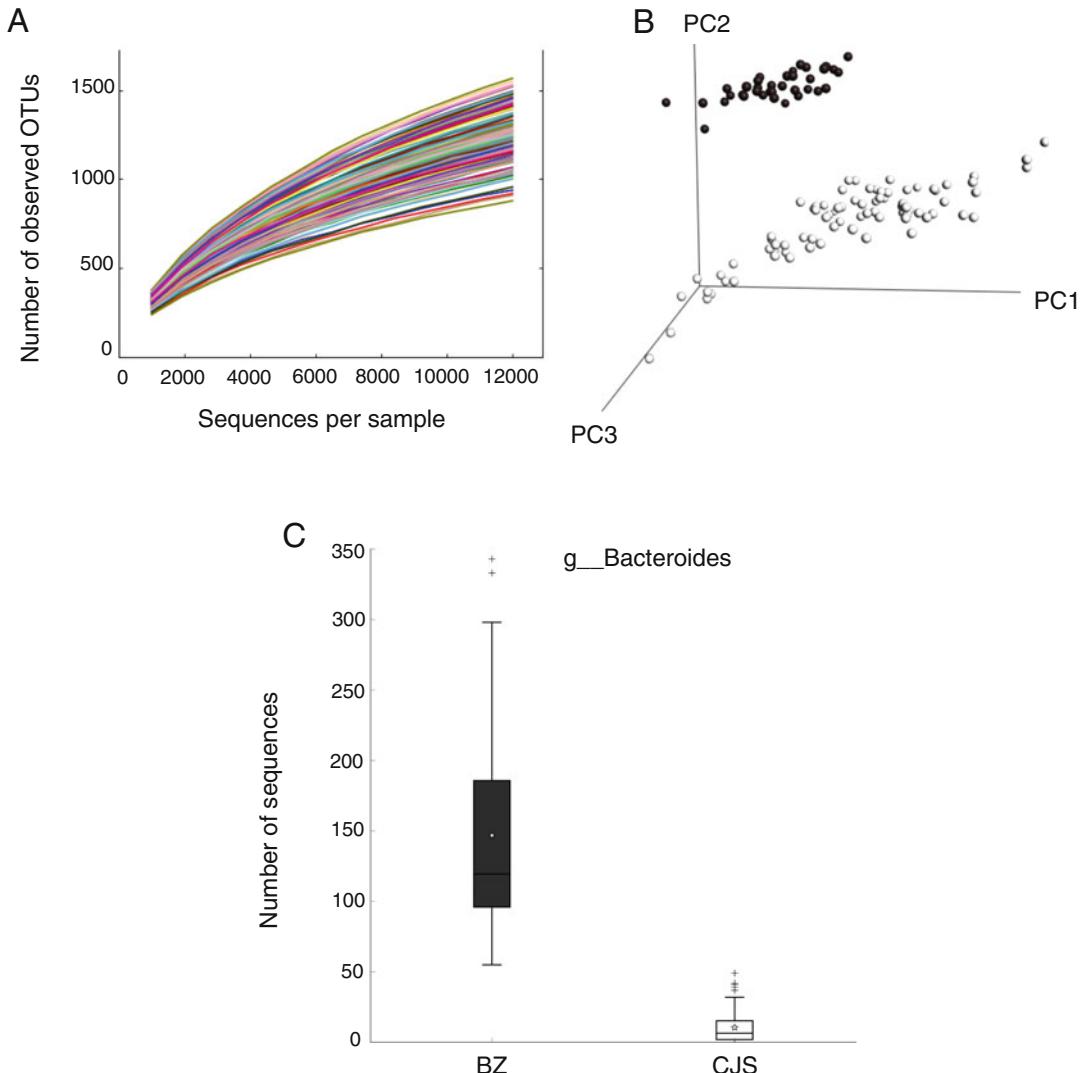


Fig. 1 Example plots generated by the Microbiome Helper pipeline. These plots are based on the full 16S rRNA gene tutorial dataset of mouse stool samples, which is available on the Microbiome Helper website. **(a)** Rarefaction curves showing the increase in richness per sample as sequencing depth increases. **(b)** Principal Coordinates Analysis plot of weighted UniFrac distances between samples. Samples are colored by which source facility the mice came from (black: BZ1, white: CJS). **(c)** Boxplot comparing the number of reads corresponding to the genus *Bacteroides* across the two source facilities generated in STAMP. All plot label sizes were increased to make them suitable for publication

picking, we use the SILVA 90% database for the alignment step and sequences from the Protist Ribosomal Reference database [18] for all other steps. For users profiling fungi with the internal transcribed spacer 2 (ITS2), we recommend using sequences for both chimera checking and OTU picking from the UNITE database [19]. These databases are all available in the Microbiome Helper VBox and website.

3. Primers are often trimmed from 16S sequences due to the concern that sequences with mismatches to the primers will still be amplified. This bias means that these amplified sequences could incorrectly start with the primer sequences rather than the true biological sequence. However, a small number of nucleotide differences is unlikely to have a large effect on OTU clustering at 97%. Also, since the primers match conserved regions of the 16S rRNA gene these sequences can be useful for anchoring the sequences during alignment.

References

1. Stackebrandt E, Goebel BM (1994) Taxonomic note: a place for DNA-DNA Reassociation and 16S rRNA sequence analysis in the present species definition in bacteriology. *Int J Syst Bacteriol* 44:846–849
2. Schloss PD, Westcott SL, Ryabin T et al (2009) Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Appl Environ Microbiol* 75:7537–7541
3. Caporaso JG, Kuczynski J, Stombaugh J et al (2010) QIIME allows analysis of high-throughput community sequencing data. *Nat Publ Gr* 7:335–336
4. Comeau AM, Douglas GM, Langille MGI (2017) Microbiome Helper: a Custom and Streamlined Workflow for Microbiome Research. *mSyst* 2:e00127-16
5. Tange O (2011) GNU Parallel: the command-line power tool. *Login USENIX Mag* 36:42–47
6. Cole JR, Wang Q, Fish JA et al (2014) Ribosomal database project: data and tools for high throughput rRNA analysis. *Nucleic Acids Res* 42:633–642
7. DeSantis TZ, Hugenholtz P, Larsen N et al (2006) Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB. *Appl Environ Microbiol* 72:5069–5072
8. Zhang J, Kobert K, Flouri T, Stamatakis A (2014) PEAR: a fast and accurate Illumina paired-end reAd mergeR. *Bioinformatics* 30:614–620
9. Rognes T, Flouri T, Nichols B et al (2016) VSEARCH: a versatile open source tool for metagenomics. *PeerJ* 4:e2584
10. Edgar RC (2010) Search and clustering orders of magnitude faster than BLAST. *Bioinformatics* 26:2460–2461
11. Edgar RC, Haas BJ, Clemente JC et al (2011) UCHIME improves sensitivity and speed of chimer detection. *Bioinformatics* 27:2194–2200
12. Rideout JR, He Y, Navas-Molina JA et al (2014) Subsampled open-reference clustering creates consistent, comprehensive OTU definitions and scales to billions of sequences. *PeerJ* 2:e545
13. Kopylova E, Noé L, Touzet H (2012) Sort-MeRNA: fast and accurate filtering of ribosomal RNAs in metatranscriptomic data. *Bioinformatics* 28:3211–3217
14. Lozupone C, Knight R (2005) UniFrac: a new phylogenetic method for comparing microbial communities. *Appl Environ Microbiol* 71:8228–8235
15. Vázquez-Baeza Y, Pirrung M, Gonzalez A, Knight R (2013) EMPeror: a tool for visualizing high-throughput microbial community data. *Gigascience* 2:16
16. Parks DH, Tyson GW, Hugenholtz P, Beiko RG (2014) STAMP: statistical analysis of taxonomic and functional profiles. *Bioinformatics* 30:3123–3124
17. Quast C, Pruesse E, Yilmaz P et al (2013) The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. *Nucleic Acids Res* 41:590–596
18. Guillou L, Bachar D, Audic S et al (2013) The protist ribosomal reference database (PR2): a catalog of unicellular eukaryote small sub-unit rRNA sequences with curated taxonomy. *Nucleic Acids Res* 41:597–604
19. Koljalg U, Nilsson RH, Abarenkov K et al (2014) Towards a unified paradigm for sequence-based identification of fungi. *Mol Ecol* 22:5271–5277



Chapter 10

Normalization of Microbiome Profiling Data

Paul J. McMurdie

Abstract

Normalization is a term that is often used but rarely defined and poorly understood. The number of choices of *normalization* procedure is large—some are inappropriate or inadmissible—and all are narrowly relevant to a specific analysis that depends on both the nature of the data and the question being asked. This chapter describes key definitions of *normalization* as they apply in metagenomics, mainly for taxonomic profiling data; while also demonstrating specific, reproducible examples of normalization procedures in the context of analysis techniques for which they were intended. The analysis and graphics code is distributed as a supplemental companion to this chapter so that the motivated reader can re-use it on new data.

Key words Normalization, Microbiome, Metagenomics, DNA sequencing, Statistics

1 Introduction

The word *normalization* is overused and its meaning is overloaded. I avoid using the unqualified term in my work because what I need to communicate is more specific than the splintered, drifting, and contextual meanings in common usage. Nevertheless, good analysis almost always requires transformational procedures that are described as *normalization* by someone(s). The goal of this section, and chapter, is to define a specific subset of *normalization* procedures that are commonly required during the analysis of taxonomic (e.g., microbiome) profiling data; while also providing a reproducible example of the application of these techniques in the context of the analysis for which they are intended.

1.1 Microbiome Profiling Data

There are two complementary techniques most-often used for culture-independent estimates of the profile of microbes in a biological sample:

Electronic supplementary material: The online version of this chapter (https://doi.org/10.1007/978-1-4939-8728-3_10) contains supplementary material, which is available to authorized users.

1. **Amplicon Sequencing.** This is defined as DNA-sequencing of the molecules, called *amplicons*, that result from a PCR-based amplification reaction. It is a general-purpose technique used in many different settings in molecular biology. Molecular profiling of a microbiome can be achieved by amplicon sequencing applied to metagenomic DNA when the PCR primers target a phylogenetically informative genetic marker (*see Note 1*). The resulting data is thus comprised of a population of sequences from the same genetic locus, but potentially every microbe in the sample. Though influenced by a number of factors, the counts of sequences from each microbe are usually well-correlated to that microbe's abundance in the microbiome sample. As one might guess, there is an enormous number of choices of PCR target when conducting amplicon sequencing. For bacteria and archaea, the 16S rRNA gene has been the most popular, but many others have been demonstrated to be useful.
2. **Shotgun Metagenome Sequencing.** Shotgun Metagenome Sequencing is the sequencing of genomic DNA extracted from a microbiome. There is a whole field devoted to the challenge of assembling metagenome sequence fragments into contigs. Microbiome profiling is often achieved by mapping these fragments (or contigs) to a reference set of genomes in order to generate a microbiome profile. The counts of reads mapped to each genome or taxon serve as the primary data for our normalization discussion here.

While both of these methods are based on DNA-sequencing, they differ tremendously in approach and in the methods by which the DNA sequences can be classified and grouped together. The distinction between these methods can have important consequences on your choice of analysis methods, including normalization.

Also note that, unlike amplicon sequencing data, shotgun metagenome sequencing can also be used to evaluate gene content of a biological sample, and has also been used to characterize microorganisms from multiple kingdoms from the same set of data. Due to technical constraints related to PCR primer design, a single amplicon sequencing result might be restricted to a single domain of life, even if the marker gene itself is universal. Many marker genes are even more taxonomically narrow than domain. On the other hand, amplicon sequencing data has a higher density of profiling information, and can thus operate at a substantially lower cost per sample. If profiling a particular domain or taxonomic group is your sole or primary focus, then amplicon sequencing may be the most appropriate method for your experimental design. This depends of course on other factors of your experiment, like sample availability, power estimation of the phenomena being characterized, etc. The profiling of viruses [1] is a particularly good example of this trade-off, as there are no *universal* genes with which to

target all species of virus in a sample via amplicon sequencing [2], and yet many studies have reported useful data from the application of amplicon sequencing that targets a specific locus within a virus of interest [3].

1.2 What Is Normalization?

The term *normalization* has many context-dependent meanings, both colloquial and technical. In technical settings the term is often used to mean *standardization of measurement to a common scale*. Alternatively or simultaneously the term is sometimes used to mean *standardization of uncertainty to a common value*, which statisticians call *variance stabilization* [4]. It is important to disambiguate the two meanings as a first step in understanding why a particular transformation of your data may (or may not) be appropriate. This chapter aims to demonstrate the usage of methods that address either or both of these important aspects. In data based on modern DNA-sequencing counts, both aspects are strongly dependent on the number of sequencing reads observed from the same biological specimen, which we refer to here as the *library size* (see Note 2). The library size typically varies substantially from one sequencing technology or model to another, as well as from one sample to another, even on the same sequencing run after pains have been taken in the laboratory to provide an equal quantity of DNA from each sample. In most applications these differences in library size are non-informative, yet the library size of a sample constrains the number of reads available to any of its components/features. It follows theoretically and empirically that the proportion of reads for each feature relative to the library size is the meaningful reproducible value, rather than the absolute read counts themselves. Data with this inherently proportional nature is called *compositional* [5], a distinction that is important when it applies because compositional data can lead to spurious correlations when analyzed as if the features were independent, a phenomenon that was first described by Karl Pearson in 1897 [6].

1.3 The Right Normalization for Your Analysis

A common misconception in many bioinformatic areas, and microbiome discussions in particular, is that a user should choose a single filtering and normalizing transformation of their data for all analyses applied; and moreover, that it is better to use the same method as everyone else so that it is comparable, regardless of whether this transformation is appropriate for the data and analysis at hand. It is unclear where this myth originated, but it is clearly perpetuated by the notion and language around a *pipeline*, which implies a linear nature of the analysis process.

In reality, microbiome analysis is often an iterative exercise with an early exploratory phase and a large variation in study designs, microbial community structures, sources of variability, and scientific goals. Even precisely defined hypothesis-testing experiments lend themselves to some exploration during quality control evaluations, or opportunistic discoveries that inspire effort in

unanticipated directions. That is to say, there are many different legitimate options for analysis. Not all analyses are appropriate (or even recommended) for a particular dataset. It is your responsibility as the analyst to understand when and why a particular distance, test, or graphic is appropriate.

Most importantly for this chapter, *different analyses and different distances may require different normalization procedures*. For the following normalization methods I will try to emphasize which common analysis methods are appropriate, acknowledging in advance that your favorite method may not be among the list. The methods mentioned here are not intended to be comprehensive, and so its absence does not mean it is ill-advised. That said, it is important that you understand your biological question well enough to know why your choice of analysis—and therefore, which normalization method—is appropriate. In the end, a discussion with your local statistician can go a long way to better understand what to do, or not do, with your data, especially before resources are spent and the experiment is conducted.

1.4 Admissible and Inadmissible Normalization by Resampling

Regrettably, an inadmissible approach we will describe here as *rarefying* has become common enough in recent microbiome literature to warrant its mention, even though it is not recommended. *Rarefying* was never formally proposed or justified, but gained popularity through its use as a default step in popular analysis software pipelines such as QIIME [7] and mothur [8].

Rarefying is a one-off resampling procedure with the following steps:

1. Review dataset and select a minimum threshold for the number of reads a library must contain to be included in analysis.
2. Discard from the data those libraries that have fewer reads than this threshold.
3. Down-sample the remaining libraries a single time, such that each has the same number of reads.
4. Carry-on with analysis as if this was the original data.

The recent popularity of this approach in microbiome literature is strangely dissociated from the decades-mature theoretical framework of resampling-based statistical analyses that form the basis for much of modern statistics [9]. Simply stated, *rarefying* as defined above (the most common definition) is a formally inadmissible procedure. This is largely because it is incomplete, treating available valid data *as if it never existed*, including whole samples (**steps 2 and 3**). Hopefully your first-impression after reading the definition of *rarefying* was to recoil with skepticism. I like to joke that this is a rare example in statistics where your intuition is actually correct!

Based on discussions with users and peers, my impression is that the popularity of *rarefying* in the microbiome field originates

from the conflation of distinct processes of sequence count-based microbiome data analysis:

1. **Error-correction.** Filter sequences with errors and assign them to their parent true sequence.
2. **Sequence-effort standardization.** Accounting for library-size dependency on estimated parameter values.
3. **Variance-stabilization.** Accounting for library-size dependency on the uncertainty of estimated parameters.

Note how 2 and 3 are the two common meanings attributed to the term *normalization* that were described earlier.

It is common in the literature—but not recommended—to approximate (1) by a fixed-radius sequence clustering heuristic (“OTU clustering,” [10]) and to use *rarefying* to simultaneously address (2) and (3). One of the apparent self-reinforcing features of this tandem is that OTU clustering is notoriously prone to generating many more features than are actually present in the sample or in nature [11], with the number of artifact features having a strong positive dependency on *library size*. The *rarefying* transformation results in a misleading impression that this issue has been addressed, when in fact all rare features in the data have simply been discarded, even those that were real. The tragic irony is that sensitivity for rare features is one of the major benefits of amplicon sequencing relative to shotgun metagenomics. Meanwhile, *rarefying* also reduces available precision to that of the smallest libraries in the dataset and adds artificial sampling noise [12].

It is encouraging to note that new, highly effective denoising tools are becoming available that make the distinctions above more obvious, and offer a complete alternative to OTU-clustering-and-rarefying [10]. In particular, DADA2 [13] has shown very promising analytical performance, including very low dependence of false positive sequences on library size, blunting one of the lingering arguments in support of rarefying.

Most directly to the topic of *normalization*, there are already resampling procedures available for DNA sequence count data, originally developed for the RNA-Seq setting that apply quite well by analogy to microbiome data. The *SAMseq* and *npSeq* packages [14] are examples of statistically admissible resampling-based procedures for making inferences of differential abundance. See Subheading 3.1.4 for details on using this method.

1.5 DNA-Sequencing Is a Discrete Sampling Process

Technical variation among aliquots of the same physical sample of DNA, but separate trials in the same sequencer, are modeled very well by a Poisson process [15]. The following is a simple but illustrative simulation of the molecular sampling process, intended to provide some intuition for this process at work on microbiome samples, especially with respect to uncertainty and the effect of

different library sizes. Let's define a simple imaginary vector of proportions of bacterial taxa, named A–Z, in a single preparation of a single sample. We'll call the vector `taxaProportions`, and give it the following values:

```
##      A      B      C      D      E      F      G      H
## 0.3000 0.2000 0.1000 0.0500 0.0500 0.0500 0.0500 0.0500
##      I      J      K      L      M      N      O      P
## 0.0250 0.0250 0.0250 0.0250 0.0250 0.0250 0.0010 0.0010
##      Q      R      S      T      U      V      W      X
## 0.0010 0.0010 0.0010 0.0010 0.0001 0.0001 0.0001 0.0001
##      Y      Z
## 0.0001 0.0001
```

We can draw taxa from this vector at random with replacement, such that our probability of picking a specific taxon at each draw is equal to the *true* proportion that we defined above. To characterize this further, we can repeat this procedure many times, as if they were separate sequencing trials at each of a few different library sizes. The results of this simple simulation are summarized graphically in Fig. 1, where the count of each taxon in each simulation trial has been transformed to proportion of the total for comparison against the known truth.

If we summarize the uncertainty in each measurement as the variance and plot the variance versus the mean count, we get the

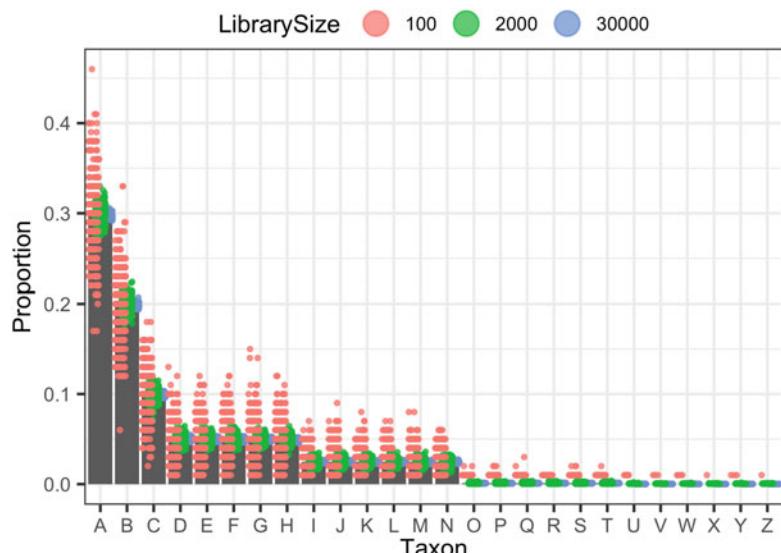


Fig. 1 Result of a simple microbiome simulation at different library sizes. Each letter represents a taxon. Each point indicates the proportion of reads that were observed for that taxon, shaded by the number of total reads (library size) in the respective simulation trial. Gray bars indicate the true proportion for each taxon

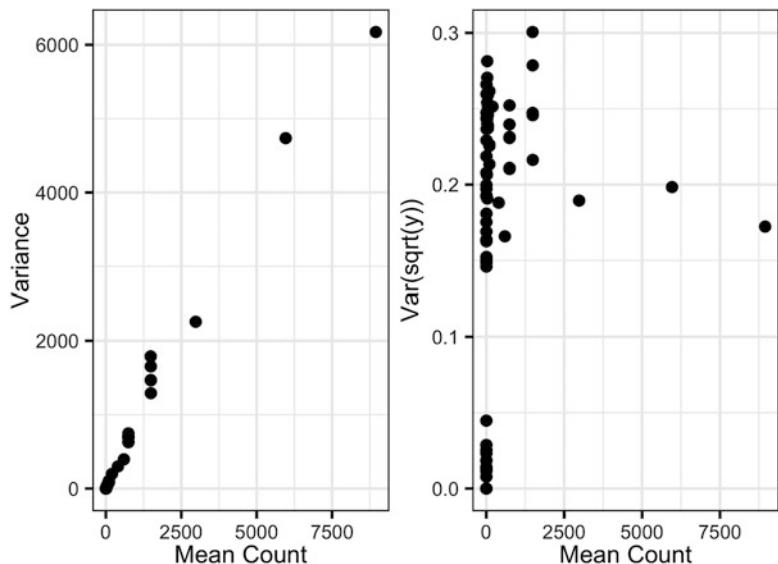


Fig. 2 Relationship of variance and mean count in the simulated microbiome example described in Subheading 1.5

following (Fig. 2), recapitulating the well-known equality between variance and mean for a Poisson process.

The result in Fig. 2 illustrates how uncertainty in the abundance of each taxon is dependent on the number of times that taxon was observed (*read* in the sequencer, *sampled* in our simulation). It is clear from both figures that the set of observed proportions derived from a larger library (more reads) are more precisely estimated (have a narrower confidence interval) than the same proportions measured using a smaller library, even though the underlying proportions are identical. Similarly, when different taxon proportions within the same library are compared across a range of values, the taxa at smaller count values have a larger uncertainty associated with their estimate (e.g., proportion), due to the smaller number of reads contributing to the measure. This scale-dependent uncertainty (formally in statistics, *heteroskedasticity*) poses a problem for many analyses that assume otherwise, and is the theoretical underpinning for many commonly encountered transformations in bioinformatic analyses, including especially *square-root*, *natural logarithm*, *regularized logarithm*, and *hyperbolic arcsin* [4].

In our simulation example the square-root transformation of the count values (Fig. 2, right panel) appears to stabilize the variance to a constant value that is no longer dependent on the scale of the observation (the mean). That a square-root transformation would have this variance-stabilizing effect is a mathematically derived behavior of Poisson distributed data [4]. While the details of this derivation or its generalization to arbitrary distributions is outside the scope of this chapter, it suffices to state that an optimal variance-stabilizing transformation is known for many commonly

encountered noise models, including a particular hierarchical Poisson distribution, the negative binomial, that is commonly used to model uncertainty among biological replicates in modern DNA-sequencing data [4].

1.5.1 Negative Binomial

Because this chapter uses methods that are based on the negative binomial distribution as a model for uncertainty in our example dataset, it seems worthwhile to provide a little more intuition and motivation behind the choice. In our simulation above, which is a good approximation of the sampling noise among technical replicates, the true mean among replicates was always exactly the value of the proportions we defined at the outset (the vector of true proportions). What about when you are estimating the mean value of a taxon from a set of biological replicates? For example, different mice that were treated identically in a therapeutic trial, or separate identically treated enrichment cultures from the same environmental inocula, or (perhaps much noisier) stool samples from patients within the same demographic cohort. It is hopefully intuitive that the true proportion of a given taxon in each sample in these experiments, as a real continuous random variable, is not identical among these biological replicates. It follows that each biological specimen contains a different *true* proportion for Taxon A, with variation that we are scientifically interested in characterizing, in addition to our estimate of the *true mean* value among the replicates. In statistics, this is called a *hierarchical model*, and in this case is also naturally an *infinite mixture model*, because we expect a different (perhaps very different!) true value for the proportion of Taxon A for every independent observation. If the distribution of true proportion values is modeled using the gamma distribution, the resulting Gamma-Poisson mixture model is mathematically equivalent to the *Negative Binomial*. The *Negative Binomial* distribution has been used effectively to model RNA-Seq data for some years now [16], and appears to work well in many microbiome profiling settings as well [12].

Just as in our simple example of Poisson distributed data, it is possible to derive an optimal variance-stabilizing transformation for *negative binomial* distributed data. This is a fact we will leverage in examples of whole-microbiome comparisons where the approach requires that the data is homoskedastic.

In some of our examples of differential abundance testing, such a transformation is not necessary, because the hypothesis test explicitly models the data as being negative binomial.

1.6 Goals and Scope

The remainder of this chapter is focused on providing examples of *normalization* as defined in Subheading 1.2, in the context of specific analysis aims.

2 Materials

All of the methods described herein are freely available as open-source software, with a preference for the R language in particular.

2.1 Recommended Software, Environment

- R Language and Environment [17]
- RStudio IDE [18]
- Bioconductor [19]

2.2 Program Availability

All of the analytical tools described in the following are freely available from the World Wide Web or from the authors, and any source code not shown is available as supplemental code.

The following are links and references to software packages directly referenced in this chapter. Additional dependencies exist, and they are explicitly defined by each package through R's package management system.

- phyloseq [20]
- DESeq2 [21]
- ALDEx2 [22]
- metagenomeSeq [23]
- SAMseq and npSeq [14]
- edgeR [24]
- limma-voom [25, 26]

3 Methods

This section is divided into two parts, where each part has a different analysis goal:

1. Testing for differential abundance of taxa between two sample groups
2. Exploring the relative similarities between whole-microbiome samples

The latter analysis goal is a descriptive approach found in much of the early amplicon sequencing-based microbiome literature. The former goal is precisely defined for the example dataset, and is a somewhat minimal example of a multiple hypothesis testing approach that is becoming more common with the increase of clinical microbiome data. In both approaches, choices related to normalization can strongly affect both the results and their interpretation.

If a required package is missing in your current R installation, use the Bioconductor installer.

```
# try http:// if https:// URLs are not supported
source("https://bioconductor.org/biocLite.R")
# For installing the phyloseq package
biocLite("phyloseq")
```

Load packages, code, example data.

```
library("phyloseq"); packageVersion("phyloseq")
```

```
## [1] '1.22.3'
```

```
library("ggplot2"); packageVersion("ggplot2")
```

```
## [1] '2.2.1'
```

```
source("supportingcode.R")
exdat = readRDS("data/Kostic2012StageII.RDS")
```

3.1 Normalization for Differential Abundance Testing

Often, biological questions motivate a formal test for differential abundance of sequences/taxa between two (or more) sets of microbiome samples. This is inherently a multiple comparisons/testing scenario, and so corrections for multiple-testing [27] are necessary. Many of the workflows will perform this correction by default, and you can assume that to be the case if the correction is not explicitly shown in the example. Generally, these corrected p -values are what you would communicate to others in practice and publication. Please bear this in mind, even though the following volcano charts in this section display the nominal p -values.

Most importantly for this chapter on normalization, the methods shown take account for aspects of the data that require normalization, e.g., differences in library size or number of samples between groups, and this is often accomplished implicitly within the relevant function(s).

3.1.1 Example Data

In the following examples I use publicly available data from a study on colorectal cancer [28]. This data consists of 454 FLX Titanium sequencing counts of PCR amplicons from the V3–V5 variable region of the 16S rRNA gene. In the original study 190 separate samples were obtained in the form of samples from 95 patient biopsy pairs from either tumor or non-tumor tissue. For simplicity of illustration, this data has been truncated to include only samples from patients with a Stage-II tumor. Details of the data selection are included in the supplemental information.

3.1.2 Differential Abundance Testing with DESeq2

The negative binomial model for DNA sequence count data was explained and motivated in Subheading 1.5.1. In the following Subheadings 3.1.2 and 3.1.3 the steps for executing two commonly used implementations of the negative binomial are provided, for testing differential abundance of modern DNA-sequencing derived features, *DESeq2* [21], and *edgeR* [24]. Note that *DESeq2* has an official extension within the *phyloseq* package and a vignette within *phyloseq*.

1. Convert example data into a format expected for *DESeq2*, including a definition for the experimental design that defines the form and meaning of the test. The function *phyloseq_to_deseq2* converts your *phyloseq*-format microbiome data into a *DESeqDataSet* with dispersions estimated, using the experimental design formula, also shown (the `~ DIAGNOSIS` term). In this case a custom calculation for geometric means is also included.

```
library("DESeq2"); packageVersion("DESeq2")

## [1] '1.18.1'

dds = phyloseq_to_deseq2(exdat, ~DIAGNOSIS)
geoMeans = apply(counts(dds), 1, gm_mean)
dds = estimateSizeFactors(dds, geoMeans=geoMeans)
```

2. Execute the *DESeq2* multiple-testing wrapper function.
3. Estimate library size factors
4. Estimate group and gene-wise dispersions
5. Fit a mean-dispersion relationship
6. Compute final dispersion estimates
7. Fit negative binomial model and perform significance test (e.g., Wald). Item 2 through item 7 are accomplished with the *DESeq* function (*see Note 3*).

```
dds = DESeq(dds, test="Wald", fitType="local")
```

8. Organize and summarize results. The *DESeq2* results are stored with the original count data and fit details in the *dds* object returned by the *DESeq* function. The *DESeq2* *results* function extracts the table of test results from the much more complex *dds* object.

```
res = results(dds)
```

It is helpful to further organize these multiple-testing results; for instance, order by significance, remove NA values from failed fits or missing data, and join with taxonomic information. Additional bookkeeping code and the custom plot function show here,

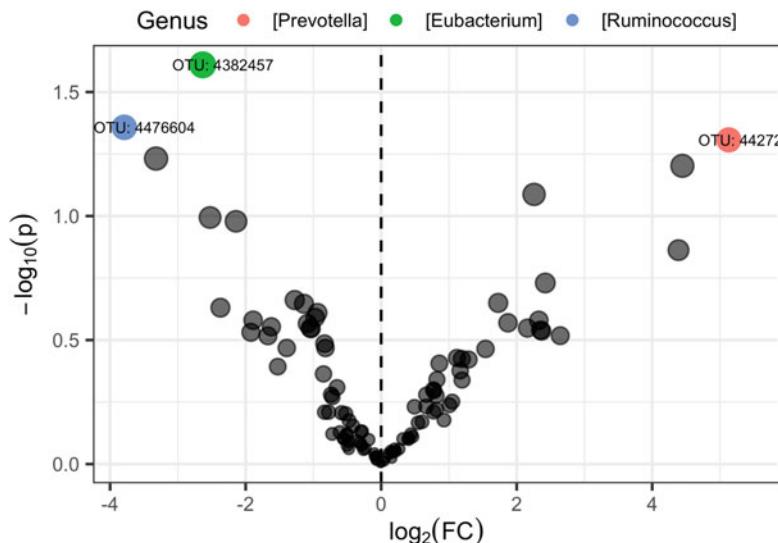


Fig. 3 Volcano plot summarizing the DESeq2 differential abundance multiple-testing results. The horizontal axis shows directional effect size as the log-2 ratio (Fold Change, FC), while the vertical axis indicates increasing significance away from the origin, as the negative log-10 transform of the nominal p -value. The most-significant few taxa are shaded according to their classified genera

`plot_deseq`, are included in the supplemental code for this chapter. It produces Fig. 3.

```
plot_deseq(res1, alpha = 0.05)
```

3.1.3 Differential Abundance Testing with edgeR

The `edgeR` package provides an alternative differential abundance testing method, including a variation that the authors propose is more robust [24]—less sensitive to outliers and other deviations from assumptions inherent to the Negative Binomial model.

1. Convert example data into format expected for `edgeR`, including a definition for the experimental design that defines the form and meaning of the test. The function `phyloseq_to_edgeR` converts your `phyloseq`-format microbiome data into a `DGEList` for `edgeR` package.

```
library("edgeR"); packageVersion("edgeR")

## [1] '3.20.9'

dge = phyloseq_to_edgeR(exdat, group="DIAGNOSIS")
design <-
  model.matrix(
  ~DIAGNOSIS,
  data=data.frame(sample_data(exdat)))
```

2. Use edgeR's *robust* variant of the dispersion estimation procedure, `estimateGLMRobustDisp`.

```
rer = estimateGLMRobustDisp(dge, design)
```

3. Compute the two-class test using `estimateGLMRobustDisp`, empirical robust bayes tagwise dispersions for negative binomial GLMs using observation weights.

```
rertest = exactTest(rer)
```

4. Extract test results using the `topTags` function, and add taxonomic information to the table.

```
rertab <-  
  topTags(rertest,  
           n = nrow(dge$table),  
           adjust.method = "BH",  
           sort.by = "PValue")@.Data[[1]]  
  
rertab <-  
  cbind(  
    as(rertab, "data.frame"),  
    as(tax_table(exdat)[rownames(rertab), ], "matrix"))
```

5. Plot results using the provided `plot_edgeR` function (see supplemental code and **Note 4**). This produces Fig. 4.

```
plot_edgeR(rertab, alpha = 0.03)
```

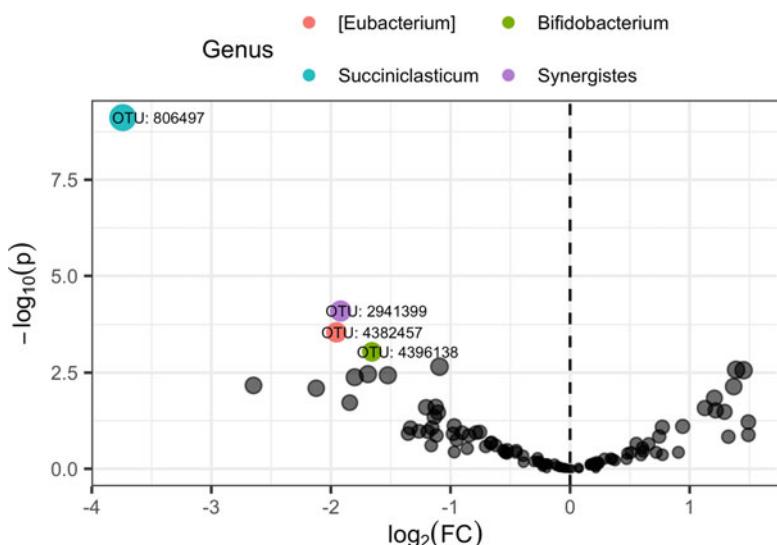


Fig. 4 Volcano plot for edgeR-robust differential abundance multiple-testing results, displayed as in Fig. 3

3.1.4 Differential Abundance Testing via Resampling

```
source("https://bioconductor.org/biocLite.R")
biocLite("impute")
devtools::install_github("npSeq", "joey711")
```

Jun Li and Robert Tibshirani proposed *SAMseq* as a nonparametric approach for identifying differential expression in RNA-Seq data[14]. The method was implemented in an existing package for the Statistical Analysis of Microarrays (SAM) in R[29]. Jun Li has since revised part of the approach and the R interface, implementing this as a separate package, called *npSeq*, which we show here. I consider it simpler to use and with better documented methods and output, and therefore more appropriate to show here in a reproducible example. The only analytical difference from SAMSeq is that *npSeq* uses symmetric rather than asymmetric thresholds, which, for some datasets, alleviates a common artifact in which all significant genes (or taxa) obtained by SAM are either all up or all down. The *npSeq* implementation more often returns a set of significant features that are a mixture of directions.

1. Define the response variable for this test, call it *Y*.

```
Y = as.integer(factor(get_variable(exdat, "DIAGNOSIS")))
```

3. Execute *npSeq* method using the custom *run_npSeq* wrapper around the *npSeq.Main* function in the *npSeq* R package. This also organizes the results into a *data.table* with taxonomy.

```
resnpSeq = run_npSeq(exdat, y = Y, nperms = 5000, nsam = 50)
```

4. Plot the results using the custom function *plot_npSeq*, shown in Fig. 5.

```
plot_npSeq(resnpSeq, alpha = 0.3, lfcmin = 0.5)
```

3.1.5 Differential Abundance Testing with *metagenomeSeq2*

metagenomeSeq is an official Bioconductor package, with the explicit goal of detecting differential abundance in microbiome experiments with an explicit design. It models metagenome count data using the Zero-Inflated Gaussian (ZIG) with scale adjustment via cumulative sum scaling [23]. Note that other than zero-inflation, this is not a discrete model, with some analytical performance limitations already acknowledged [12]. A better performing modification has been proposed using the zero-inflated log-normal instead.

1. Load *metagenomeSeq* package

```
library("metagenomeSeq"); packageVersion("metagenomeSeq")
```

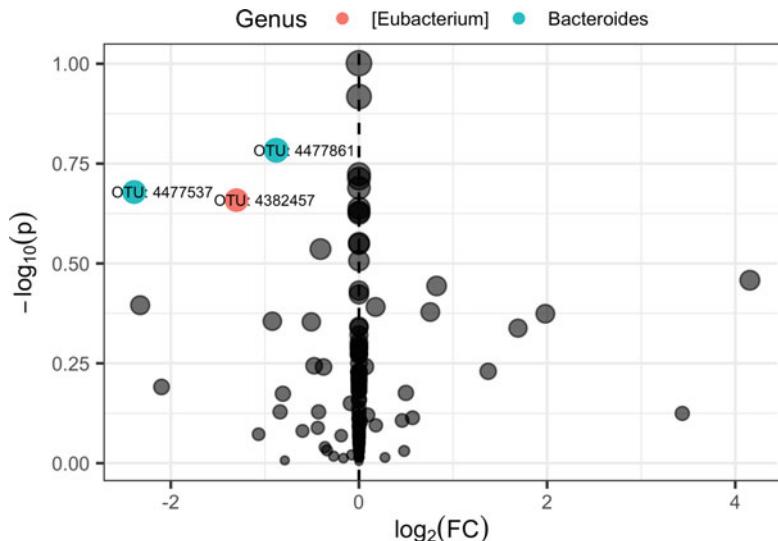


Fig. 5 Volcano plot of npSeq[14] differential abundance multiple-testing results, displayed as in Fig. 3

```
## [1] '1.20.1'
```

2. Convert the phyloseq example dataset into a metagenomeSeq object using phyloseq's included `phyloseq_to_metagenomeSeq` function.

```
mgs = phyloseq_to_metagenomeSeq(exdat)
```

3. Fit the Zero-Inflated Gaussian model to the data, using `metagenomeSeq`'s `fitZig` function.
4. Perform differential abundance test using `metagenomeSeq`'s `MRfulltable` function.
5. Organize table and include taxonomic information for interpretation.
6. The previous **item 3** through **item 5** are wrapped into the `test_metagenomeSeq` function, provided in the supplemental code
7. The provided `plot_metagenomeSeq` function graphically summarizes the `metagenomeSeq` results on this same dataset as Fig. 6

```
plot_metagenomeSeq(mgsres)
```

3.1.6 Differential Abundance Testing with Limma-Voom

An alternative method also intended for a variance stabilized multiple-testing framework for RNA-Seq data, “limma-voom” [26] is a modification to a theoretical and software framework originally developed for microarrays (“limma”) so that it is compatible with sequencing-derived discrete count data. The ability to

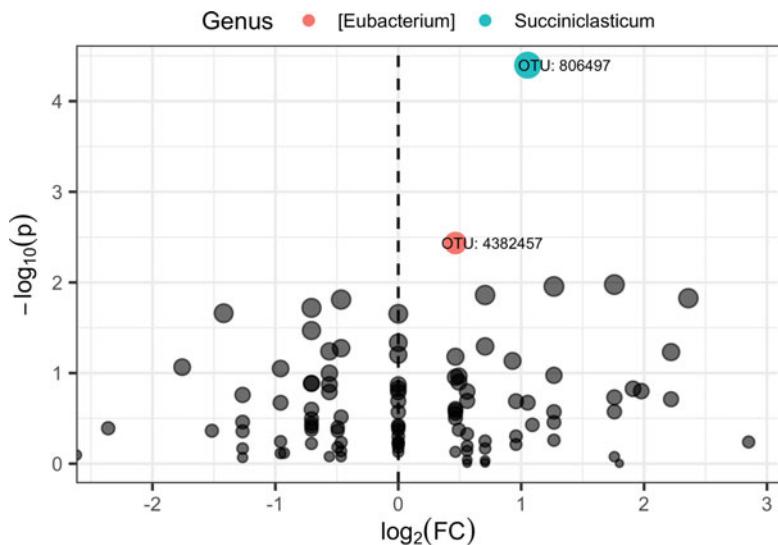


Fig. 6 Volcano plot of metagenomeSeq differential abundance multiple-testing results, displayed as in Fig. 3

interface with the large number of microarray-oriented statistical modeling tools is compelling, and the expectation is that this approach is a good-enough approximation of the error-process to be useful for many cases.

1. Load limma [25] package.

```
library("limma"); packageVersion("limma")
```

```
## [1] '3.34.9'
```

2. Convert the phyloseq example dataset into a standard R matrix suitable for use by the `voom` function. Taxa should be rows.

```
x = as(otu_table(exdat), "matrix")
if(!taxa_are_rows(exdat)){x <- t(x)}
```

3. Define the design matrix for linear modeling, specifying sample classes in the `DIAGNOSIS` factor, and save this as `design`.

```
design <-
  model.matrix(
    ~DIAGNOSIS,
    data = data.frame(sample_data(exdat)))
```

4. Compute `voom` transformation for linear modeling.

```
y <- voom(x, design, plot=FALSE)
```

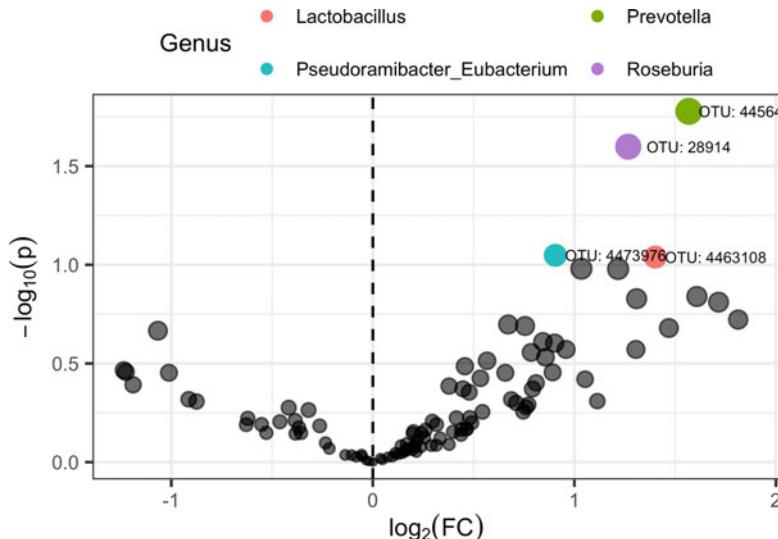


Fig. 7 Volcano plot of the limma-voom differential abundance multiple-testing results, displayed as in Fig. 3

5. Compute test using limma's `lmFit` and `eBayes` functions.

```
fit <- lmFit(y, design)
fit <- eBayes(fit)
```

6. Extract test results using limma's `topTable` command.

```
tt = as.data.frame(topTable(fit, coef=2, n=nrow(x), sort="p"))
```

7. The provided `plot_limmaVoom` function graphically summarizes the limma-voom results as shown in Fig. 7.

```
plot_limmaVoom(tt, exdat, alpha = 0.1)
```

3.1.7 Differential Abundance Testing with ALDEx2

A fundamental assumption in most of the previous methods (even not-recommended *rarefying*) is that the data can be modeled as counts, with library size accounted for as a nuisance variable that mostly affects the scale, while features are otherwise treated as approximately independent. As mentioned in Subheading 1.2, this assumption can be dangerous if the data is *compositional* [5], and there are many compelling arguments and examples in support of this concern [30].

ALDEx2 [22] (Anova-Like Differential Expression, 2) is a bioconductor package that approaches the problem of differential abundance of sequence feature counts as inherently compositional data. In this approach, data are transformed to a Centered Log-Ratio (CLR), while also making use of Monte Carlo resampling from a Dirichlet Multinomial to address sparsity (see Note 5).

1. Load ALDEx2 package.

```
library("ALDEx2")
aldat = as(otu_table(exdat), "matrix")
if(!taxa_are_rows(exdat)){aldat <- t(aldat)}
aldat <- as.data.frame(aldat)
Conditions = as.character(get_variable(exdat, "DIAGNOSIS"))
stopifnot(length(Conditions) == ncol(aldat))
```

2. Generate instances of the CLR-transformed values. The `mc.samples` parameter determines the number of Monte Carlo Dirichlet instances to generate. This is the sort of parameter that should not affect the results unless it is too small (the package authors claim 128 is “usually sufficient”), but larger values incur a larger computational burden. When in doubt: increase the value of `mc.samples` and check if the results changed appreciably.

```
x <- aldex.clr(
  reads = aldat,
 conds = Conditions,
  mc.samples=128,
  verbose=TRUE, useMC = TRUE)
```

3. **Perform test.** In this case a two-sample test via *Welch's t* and *Wilcoxon rank*, using `aldex.ttest()` function. Input values are the `aldex` object from `aldex.clr` in previous step, a vector of sample conditions (`Conditions`, a character vector), and a logical parameter `paired.test` indicating whether the test should be calculated as a paired-test.

```
x.tt <- aldex.ttest(x, Conditions, paired.test=TRUE)
```

```
## [1] "running tests for each MC instance:"
## |----- (25%) ----- (50%) ----- (75%) ----- |
```

Alternatively, one can calculate GLM and Kruskal Wallace tests using the `aldex.glm` function. This is typically much slower than the two-component test shown above.

4. **Summarize Results.** Estimate effect size, within and between condition values in the case of two conditions. This step is also required for plotting.

```
## [1] "operating in serial mode"
```

5. **Plot Results.** The resulting plot is shown in Fig. 8.

```
plot_aldex2(x.tt, x.effect, exdat, alpha = 0.3)
```

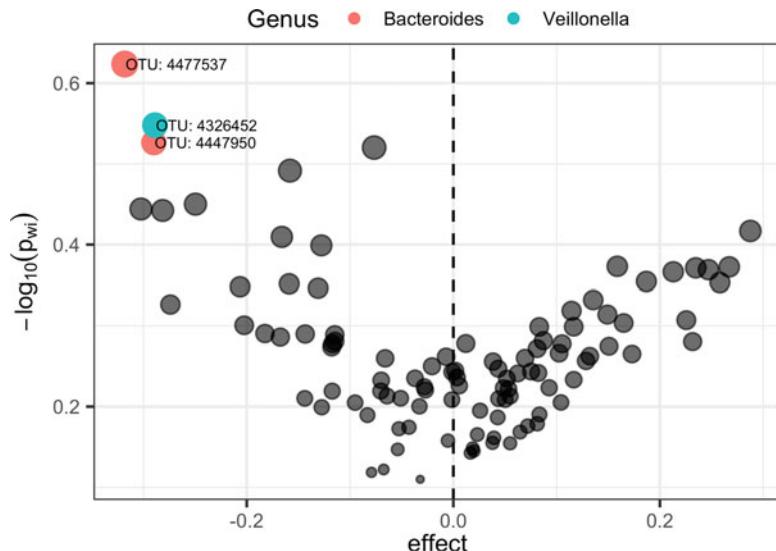


Fig. 8 Volcano plot of the ALDEx2 differential abundance multiple-testing results, displayed as in Fig. 3. Here effect is the median of the ratio of the between-group difference and the larger of the variance within groups, and p is the Wilcoxon test nominal p -value

3.1.8 Seeking Ground Truth for Differential Abundance

You probably noticed that different choices in multiple-testing method resulted in different interpretations, sometimes substantial, even though they were all applied to the same data. Whether a particular method is appropriate, including its corresponding normalization procedure(s), depends on the quality and form of the data itself. It is a decision that you as a researcher must make, and one you must also be able to defend after you have assigned meaning to your interpretation of the data.

The best supporting evidence would be independent observations, especially complementary measurements (see Subheading 3.3). In its absence, an alternative view of the data can be helpful, especially one that relates back to the primary observations. In our case, several of our procedures for differential abundance have a specific model of the data, which might be altogether wrong or at least sensitive to deviations from the model. There are many useful “Goodness of Fit” techniques from which to choose, and these are both recommended and left to the reader to explore.

Meanwhile, let us assume that the model-fit was acceptable for the data overall, and now you quite reasonably desire a graphic that summarizes the primary observations that underpin a particular *significant* taxon. As an example, we plot the abundance of OTU 806497 in the genus *Succinilasticum*, which appeared among the most-significant OTUs in three of our differential abundance tests. The provided `plot_otu` function created Fig. 9, in which each

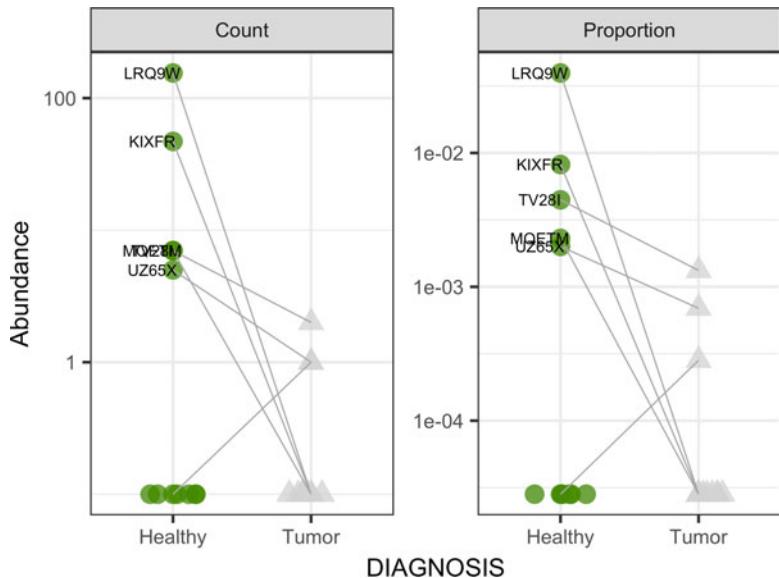


Fig. 9 Observed abundance values for the *Succinilasticum* OTU 806497. Each point represents a sample. Pairs of samples from the same human host are connected by thin lines. The count and proportion panels display the un-normalized read count and the proportion of reads in the sample, respectively. Vertical scales are both log10, but with different ranges

point represents the abundance of OTU 806497 in a sample, shown as both the unmodified count (number of sequencing reads) and its proportion of the total reads. The sample proportion is a simple normalization that we will revisit again in Subheading 3.2.

```
plot_otu(exdat, "806497")
```

Most importantly, Fig. 9 shows that OTU 806497 is only observed in a little more than one-third of host tissue-pairs. In three of these pairs OTU 806497 was not observed in tumor tissue, making for an ostensibly strong result, but observed with only a modest proportion difference in the remaining two pairs. Although OTU 806497 was among the most significant in our toy example data, it is actually a modest or even borderline result. The absence of this OTU in two-thirds of the subjects suggests it might not be a biologically useful observation, though this is strongly context dependent. Even if not considered reliable on its own, a larger cohort and a stratifying explanatory variables might help.

Finally, note that in Fig. 9 the two panels between raw counts and proportion appear quite similar. This is not surprising when library sizes are uniformly distributed in a narrow range, but in real data the library size distribution can be quite skewed. In extreme

cases these panels can deviate substantially due to serendipitous or structural patterns in library sizes. Even in this well-behaved comparison there are at least one or two cases where the rank order between the samples in each class appears to have changed.

3.2 Normalization for Comparing Samples

Early metagenomics literature is rich with the use of exploratory data analysis methods, especially those borrowed or adapted from ecology. These analyses are focused on high-level descriptions of microbial community profiles, and recognizable patterns of similarities and differences among them. The general approach is to define a distance that quantifies the pairwise dissimilarity between two microbial samples, and then compute this distance for all pairs of samples in the experiment. This distance matrix is then decomposed into orthogonal components that are much more amenable to graphical representation than the original feature-by-sample table. The distance matrix decomposition, a type of *ordination* procedure, is often achieved via multidimensional scaling (MDS, also referred to as *PCoA*) [31], or non-metric multidimensional scaling (NMDS) [32]. The choice of normalization procedure used prior to distance computation is often implied or constrained by the choice of distance measure. While the choice of distance measure and ordination method are chapters or books unto themselves, for simplicity here we will use just one distance measure, Bray-Curtis [33], and one ordination method, MDS—with the exception of the CLR normalization method, where standard principal components analysis (PCA) is used, and therefore, a separate distance measure is not needed. Note that the Bray-Curtis distance is sensitive to the total counts in each sample, so original non-normalized counts are not appropriate because the total count is an arbitrary artifact of the sequencing process.

The following examples are commonly used approaches for comparing whole microbiome samples to one another, with the main difference between each subsection being the normalization method employed. The phyloseq package [20] is especially useful for these examples, as it unifies the distance, ordination, and plotting commands in a consistent interface of just a few well-documented functions. For further examples and details, see [the phyloseq home page tutorials](#), or a recent open-source end-to-end workflow [34].

3.2.1 Simple Proportion

1. Transform count values to relative abundance, also known as sample proportion.

```
exRA = transform_sample_counts(exdat, function(x) x/sum(x))
```

2. Compute sample-wise distance and ordinate

```
OrdRA = ordinate(exRA, "MDS", "bray")
```

3. Plot

```
pRA = plot_ordination(exRA, OrdRA, title = "Proportions")
```

3.2.2 Variance-Stabilizing Transformation with DESeq2

DESeq2 provides two different variance-stabilizing transformations. One is based on the negative binomial model, and the other is a regularized logarithmic transformation [21].

1. Transform count values to vst or rlog. A few extra steps for replacing negative values with zero for Bray-Curtis distance.

```
dds = phyloseq_to_deseq2(exdat, ~DIAGNOSIS)
# Normalization, rlog
NormRlog = t(assay(rlog(dds)))
NormRlog[NormRlog < 0] <- 0.0
# Normalization, vst
NormVST = t(assay(varianceStabilizingTransformation(dds)))
NormVST[NormVST < 0] <- 0.0
# Replace in respective copies of phyloseq object
exRLog = exVST = exdat
otu_table(exRLog) <- otu_table(NormRlog, FALSE)
otu_table(exVST) <- otu_table(NormVST, FALSE)
```

2. Compute sample-wise distance and ordinate

```
OrdRLog = ordinate(exRLog, "MDS", "bray")
OrdVST = ordinate(exVST, "MDS", "bray")
```

3. Plot

```
pRLog = plot_ordination(exRLog, OrdRLog, title = "RLog")
pVST = plot_ordination(exVST, OrdVST, title = "VST")
```

3.2.3 Centered Log-Ratio

Considering our earlier discussion on compositional data [22], the following approach normalizes abundance values via the *Centered Log-Ratio* (CLR).

1. Transform count values to CLR. Requires some filtering and transformation of zeroes using the cmultRep1 function from the “zCompositions” package [35], as suggested by Gloor et al. [36].

```
exCLR = phyloseq_CLR(exdat)
```

2. Principal components analysis on the CLR-transformed abundance values [36].

```
OrdCLR = ordinate(exCLR, "RDA")
```

3. Plot

```
pCLR = plot_ordination(exCLR, OrdCLR, title = "CLR")
```

3.2.4 Show the Normalization Plots Together

```
plotList = lapply(list(pRA, pRLog, pVST, pCLR), add_theme)
do.call(what = gridExtra::grid.arrange,
       args = c(plotList, list(nrow = 2)))
```

3.2.5 Interpreting Ordination Plots

Figure 10 should be considered just a tiny example of the field of exploratory data analysis (EDA), and its application in metagenomics. The only clear pattern that I noticed in these charts is that samples from the same human host were usually more similar to each other than they were to most other samples, regardless of the choice of normalization or distance. This phenomenon, in which between-host variation dominates the sample-to-sample variability, has been observed in many human-associated microbiome samples and body types [37, 38]. Methods like Permutational Multivariate Analysis of Variance (PERMANOVA, vegan::adonis()) allow an investigator to evaluate the relative contribution of experimental design variables to the observed sample-to-sample microbiome variation encoded in the sample-wise distance matrix [39]—a multivariate analog of the univariate analysis of variance (ANOVA). In studies like this example, with repeated observations from the same human host, the host indicator variable can often account for as much or more of the sample-wise variability than key study variables.

Regardless of the challenge of high host-wise variability, there are still opportunities for interpretive modeling. In a larger dataset than this example (e.g., [38]), additional clinical variables could be explored, using one or more of the ordination axes as a response pseudo-variable (or vice versa), in a formal model. Statisticians call this “Principal Components (or Coordinates) Regression” [40], and is worth consideration if you are already in gazing at ordination plots, and attempting to interpret their patterns in the context of other sample covariates. While this deeper study-specific interpretation is beyond the scope of this chapter, the examples leading to the charts in Fig. 10 demonstrate the mechanics of normalization in this context of whole-sample comparisons.

3.3 Normalizing by Independent Complementary Measurements

Some of the most biologically informative approaches for normalizing metagenomic count data are those that incorporate independent complementary measurements of the same biological material. For example, this can provide a means to transform a sequence count proportion into an absolute scale, or a scale that is relative to mass of biological material. In many settings these latter representations of the metagenomic measurement are more generally

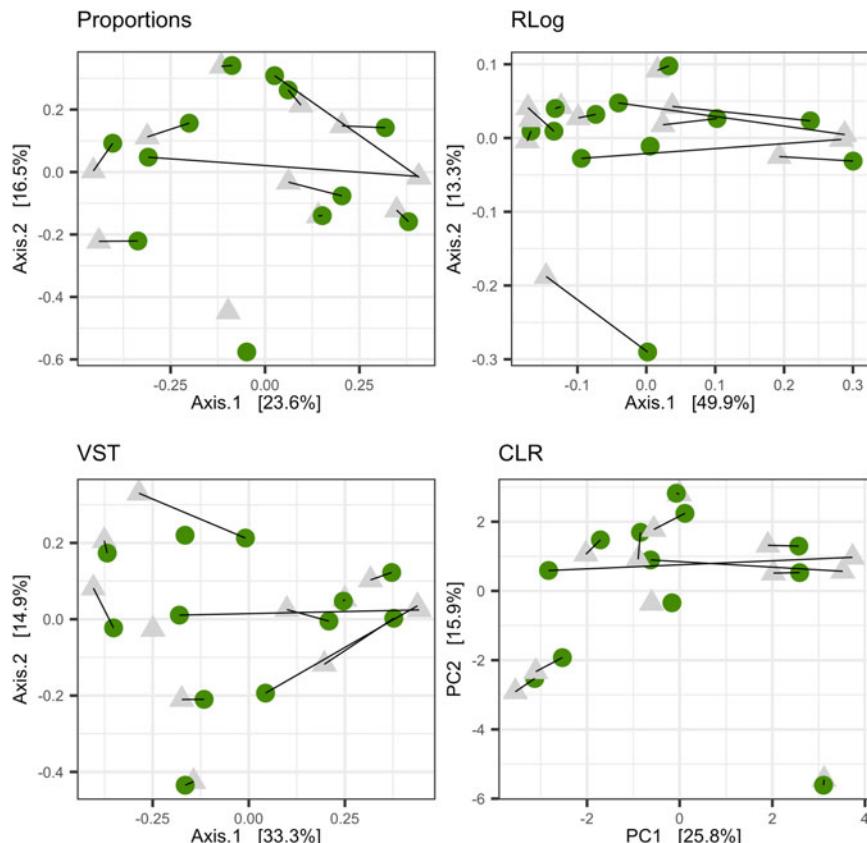


Fig. 10 Ordination results following four different whole-microbiome normalization methods. RLog, VST, and CLR are Regularized Logarithm, Variance-Stabilizing Transformation, and Centered Log-Ratio, respectively. Light gray triangles and green circles indicate Tumor and Healthy tissue, respectively. The thin black lines connect biospecimens from the same human host. The normalization and distance method specifications are shown in the previous sections

comparable across samples, materials, and experiments than is possible with sequencing counts alone. Some common complementary measures include mass of biological material used in DNA extraction, mass microbial DNA extracted, total cell counts, and quantitative PCR (e.g., qPCR or ddPCR) of the same amplicon. The use of an *internal* complementary measure—for example, the *spike-in* addition of a known quantity of exogenous DNA of known composition—is an approach common in the quantitative measurements of other scientific fields, and is gaining interest in metagenomics as well. At the time of this writing, the metagenomics field is still in the early phases of developing standard materials and protocols for more effective comparisons across studies, and there is currently no consensus on the minimum set of procedures or internal references appropriate for a metagenomics analysis.

4 Notes

1. PCR primers define both edges of the DNA amplified in PCR.
2. You might encounter other terms for *library size* in common usage, especially “read depth,” “sequencing effort,” number of reads per sample, etc.
3. In this case we’re using its default testing framework, but you can use alternatives via the Generalized Linear Model (GLM, `glm()`) interface. The default multiple-inference correction is to use the False Discovery Rate (FDR) [27], computed within the `DESeq` function.
4. The default method for multiple-inference adjustment is FDR [27].
5. The issue of sparsity in this context really refers to both the uncertainty related to the discrete sampling process described in Subheading 1.5 and functions with undesirable behavior in the presence of zeros, like the geometric mean.

References

1. Wolfs TF, Zwart G, Bakker M, Goudsmit J (1992) HIV-1 genomic RNA diversification following sexual and parenteral virus transmission. *Virology* 189:103–110
2. Lipkin WI (2010) Microbe hunting. *Microbiol Mol Biol Rev* 74:363–377
3. Beerenwinkel N, Günthard HF, Roth V, Metzner KJ (2012) Challenges and opportunities in estimating viral genetic diversity from next-generation sequencing data. *Front Microbiol* 3:329
4. Holmes S, Huber W (2018) Modern statistics for modern biology. Cambridge University Press, Cambridge (in press)
5. Aitchison J, Egozcue JJ (2005) Compositional data analysis: where are we and where should we be heading? *Math Geol* 37:829–850. <https://doi.org/10.1007/s11004-005-7383-7>
6. Pearson K (1897) Mathematical contributions to the theory of evolution. On a form of spurious correlation which may arise when indices are used in the measurement of organs. *Proc R Soc Lond* 60:489–498. <https://doi.org/10.1098/rspl.1896.0076>
7. Caporaso JG, Kuczynski J, Stombaugh J et al (2010) QIIME allows analysis of high-throughput community sequencing data. *Nat Methods* 7:335–336
8. Schloss PD, Westcott SL, Ryabin T et al (2009) Introducing mothur: open-source, platform-independent, community-supported software for phylogenetic analysis, classification, and comparison. *Genome Biol Evol* 1:3–12
9. Holmes S, Huber W, Hedges RM, et al (2018) for describing and comparing microbial communities. *Appl Environ Microbiol* 75:7537–7541
10. Efron B (2000) The bootstrap and modern statistics. *J Am Stat Assoc* 95:1293–1296
11. Callahan BJ, McMurdie PJ, Holmes SP (2017) Exact sequence variants should replace operational taxonomic units in marker-gene data analysis. *ISME J* 11:2639–2643
12. Kopylova E, Navas-Molina JA, Mercier C et al (2016) Open-source sequence clustering methods improve the state of the art. *mSystems* 1:e00003–e00015
13. McMurdie PJ, Holmes S (2014) Waste not, want not: why rarefying microbiome data is inadmissible. *PLoS Comput Biol* 10:e1003531
14. Callahan BJ, McMurdie PJ, Rosen MJ et al (2016) DADA2: high-resolution sample inference from Illumina amplicon data. *Nat Methods* 13:581–583
15. Li J, Tibshirani R (2013) Finding consistent patterns: a nonparametric approach for identifying differential expression in RNA-Seq data. *Stat Methods Med Res* 22:519–536
16. Mariotti JC, Mason CE, Mane SM et al (2008) RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome Res* 18:1509–1517
17. Rapaport F, Khanin R, Liang Y et al (2013) Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data. *Genome Biol* 14:R95

17. R Core Team (2016) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna
18. RStudio Team (2016) RStudio: integrated development environment for r. RStudio, Inc., Boston, MA
19. Huber W, Carey VJ et al (2015) Orchestrating high-throughput genomic analysis with Bioconductor. *Nat Methods* 12:115–121
20. McMurdie PJ, Holmes S (2013) phyloseq: an R package for reproducible interactive analysis and graphics of microbiome census data. *PLoS One* 8:e61217
21. Love MI, Huber W, Anders S (2014) Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome Biol* 15:550
22. Fernandes AD, Reid JN, Macklaim JM et al (2014) Unifying the analysis of high-throughput sequencing datasets: characterizing RNA-seq, 16S rRNA gene sequencing and selective growth experiments by compositional data analysis. *Microbiome* 2:1–13
23. Paulson JN, Stine OC, Bravo HC, Pop M (2013) Differential abundance analysis for microbial marker-gene surveys. *Nat Methods* 10:1200–1202. Advance online publication SP - EP :-1–6
24. Zhou X, Lindsay H, Robinson MD (2014) Robustly detecting differential expression in RNA sequencing data using observation weights. *Nucleic Acids Res* 42:e91
25. Ritchie ME, Phipson B, Wu D et al (2015) limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res* 43:e47
26. Law CW, Chen Y, Shi W, Smyth GK (2014) voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol* 15:R29
27. Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J R Stat Soc Ser B Methodol* 57:289–300
28. Kostic AD, Gevers D, Pedamallu CS et al (2012) Genomic analysis identifies association of *Fusobacterium* with colorectal carcinoma. *Genome Res* 22:292–298
29. Tusher VG, Tibshirani R, Chu G (2001) Significance analysis of microarrays applied to the ionizing radiation response. *Proc Natl Acad Sci USA* 98:5116–5121
30. Fernandes AD, Macklaim JM, Linn TG et al (2013) ANOVA-like differential expression (ALDEx) analysis for mixed population RNA-Seq. *PLoS One* 8:e67019
31. Gower JC (1966) Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika* 53:325–338
32. Minchin PR (1987) An evaluation of the relative robustness of techniques for ecological ordination. *Vegetatio* 69:89–107
33. Bray JR, Curtis JT (1957) An ordination of the upland forest communities of Southern Wisconsin. *Ecol Monogr* 27:325
34. Callahan B, Sankaran K, Fukuyama J et al (2016) Bioconductor workflow for microbiome data analysis: from raw reads to community analyses. *F1000Res* 5:1492
35. Palarea-Albaladejo J, Martín-Fernández JA (2015) zCompositions - R package for multivariate imputation of left-censored data under a compositional approach. *Chemom Intell Lab Syst* 143:85–96
36. Gloor GB, Reid G (2016) Compositional analysis: a valid approach to analyze microbiome high-throughput sequencing data. *Can J Microbiol* 62:692–703
37. Turnbaugh PJ, Gordon JI (2009) The core gut microbiome, energy balance and obesity. *J Physiol* 587:4153–4158. <https://doi.org/10.1113/jphysiol.2009.174136>
38. Kolde R, Franzosa EA, Rahnavard G et al (2018) Host genetic variation and its microbiome interactions within the human microbiome project. *Genome Med* 10:6. <https://doi.org/10.1186/s13073-018-0515-8>
39. Anderson M (2001) A new method for non-parametric multivariate analysis of variance. *Austral Ecol* 26:32–46
40. James G, Witten D, Hastie T, Tibshirani R (2013) An introduction to statistical learning. Springer, Berlin



Chapter 11

Predicting the Functional Potential of the Microbiome from Marker Genes Using PICRUSt

Gavin M. Douglas, Robert G. Beiko, and Morgan G. I. Langille

Abstract

Marker-gene sequencing is a cost-effective method of taxonomically profiling microbial communities. Unlike metagenomic approaches, marker-gene sequencing does not provide direct information about the functional genes that are present in the genomes of community members. However, by capitalizing on the rapid growth in the number of sequenced genomes, it is possible to *infer* which functions are likely associated with a marker gene based on its sequence similarity with a reference genome. The PICRUSt tool is based on this idea and can predict functional category abundances based on an input marker gene. In brief, this method requires a reference phylogeny with tips corresponding to taxa with reference genomes as well as taxa lacking sequenced genomes. A modified ancestral state reconstruction (ASR) method is then used to infer counts of functional categories for taxa without reference genomes. The predictions are written to pre-calculated files, which can be cross-referenced with other datasets to quickly generate predictions of functional potential for a community. This chapter will give an in-depth description of these methods and describe how PICRUSt should be used.

Key words Marker genes, Metagenomes, Functional prediction, Ancestral state reconstruction, Phylogenetic analysis

1 Introduction

Marker-gene and metagenomic studies can both provide useful information about the taxonomic composition of a microbiome sample. While metagenomic surveys can have the additional benefit of producing functional information, this extra information comes at a much higher overall sequencing cost. Metagenomic datasets do not provide the same depth of taxonomic sampling as marker-gene (typically 16S ribosomal RNA gene, henceforth 16S) studies, and the computational cost of sequence analysis is significant. Since there is a relationship (however imperfect) between the phylogenetic relatedness of organisms and their complement of functional genes [1, 2], it is reasonable to propose that reference genomes can serve as functional proxies for closely related microorganisms that

are present in a given sample. Given a set of 16S sequences from a sample, it is possible to identify the most closely related organisms with associated sequenced genomes, and propose that their associated functions are also present in the sampled microbiome. Phylogenetic Investigation of Communities by Reconstruction of Unobserved States (PICRUSt [3]) is an algorithm and software package that performs these predictions for 16S sequences, using a reference phylogeny to weight the relative functional contributions of closely related sequence genomes.

Two related advantages of PICRUSt are its ability to propose functions that may be characteristic of particular habitats, and to identify cases where distantly related organisms may provide the same critical functions and the presence of one or the other is sufficient for the function to be present in the community. PICRUSt was developed and tested on several paired 16S and metagenomic datasets and was shown to have surprisingly high accuracy (>90% in most cases). PICRUSt has since been applied to samples collected from a wide range of habitats including the human gastrointestinal tract [4], crops and surrounding soils [5], and coral reefs [6]. While it must be remembered that PICRUSt *predicts* the functional attributes of a microbiome rather than *identifying* them directly from DNA sequences, the predictions made by PICRUSt can serve as useful hypothesis-generating tools and provide alternative ways to probe the structure of the microbiome.

Two additional tools for functional inference from 16S sequences have been released since PICRUSt's publication: Tax4Fun [7] and Pipillin [8]. Tax4Fun is an R package that linearly combines precomputed functional profiles based on taxonomic abundances. It uses SILVA [9] as a reference database for 16S sequences, which differs from the default database used by PICRUSt (*see Subheading 2.1*). One possible advantage of Tax4Fun is that it is potentially more accurate for communities with a large proportion of poorly characterized phyla [7]. Pipillin is another recent tool, which differs from both PICRUSt and Tax4Fun in that it does not require a phylogenetic tree or a database of 16S sequences. Instead, Pipillin uses a nearest-neighbor algorithm to quickly map 16S sequences to reference genomes. It is possible that these tools may be better suited for different contexts, which may be reflected by the conflicting reports in their inference accuracies [7, 8]. Predictions from all of these tools should be interpreted extremely cautiously when analyzing communities that have a large proportion of 16S sequences distantly related to existing reference genomes.

2 Materials

PICRUSt can be used on the command line (*see* <http://picrust.github.io/picrust/install.html>) in a Linux or Mac OS system with minimal hardware requirements (single-core processor with >4GB

RAM), or as an online Galaxy implementation (*see* <http://huttenhower.sph.harvard.edu/galaxy/> or <http://galaxy.morganlangille.com/>). The inputs to PICRUSt follow standard formats that can be produced from a microbial ecology analysis done in QIIME [10] or mothur [11]. Several online tutorials for running PICRUSt are available, including one which was part of a Canadian Bioinformatics Workshop (https://github.com/LangilleLab/microbiome_helper/wiki/CBW-2016-PICRUSt-tutorial).

2.1 Reference OTUs

The input file for PICRUSt is a BIOM or tab-separated table of OTU abundances. By default, marker-gene sequences must be picked against a Greengenes reference [12] prior to use in PICRUSt in order to use the provided pre-calculated files. Updates to PICRUSt that allow the functional potential of de novo OTUs to be predicted is planned for PICRUSt version 2.0. However, currently users can create custom databases to be used for genome prediction of their study sequences of interest (*see* Subheading 3.1). Two versions of Greengenes are supported by PICRUSt version 1.1.0: “18may2012” and “13.5/13.8” (which correspond to updates in May and August 2013). Sequences can be assigned to OTUs using either the “closed” method (which generates OTUs only from sequences that match the reference at the appropriate level of sequence identity: *see Note 1*), or the “open” method (which can create new OTUs for sequences that do not match the reference). However, since PICRUSt cannot make predictions from novel OTUs, these must be removed from the file prior to performing the metagenome prediction (*see Note 2*). OTU files can be in BIOM [13] or tab-separated format.

2.2 Reference Genomes

Two files are necessary to represent the reference genomes. The first file summarizes the copy number, per genome, of the marker gene being used. This file must include two columns, separated by a tab, with one row for each genome: the first column must contain the identifier for a particular genome, and the second, the corresponding copy number. The standard reference file used by PICRUSt is obtained from the Joint Genome Institute Integrated Microbial Genomes (JGI IMG: [14]) resource and named “IMG_16S_counts.tab.”

The second file is also tab separated, with one row per genome. The first column in this file is again the genome identifier, while the remaining columns indicate the counts for each functional trait. In the case of the KEGG Orthology (KO) groups, each column will correspond to one specific KEGG identifier (for example, K04041 = “fructose-1,6-bisphosphatase III”). By default PICRUSt supports three ontology schemes for prediction of genome functions (*see Note 3*). IMG genomes are linked to Greengenes ids based on matching of identical 16S sequences in both of these default files.

2.3 Reference Phylogenetic Tree

The input phylogenetic tree will be used by PICRUSt to infer ancestral states based on genomes with known functions. All sequenced genomes should have an associated tip label, either with the genome ID present directly in the tree file, or cross-referenced in a two-column, tab-separated file that uniquely map tip IDs (first column) to genome IDs (second column). The tree should be in Newick format: currently the default tree is the Greengenes reference “gg_tree.nwk.” Users can add additional sequences to this phylogeny using pplacer [15] or the evolutionary placement algorithm [16].

3 Methods

PICRUSt uses ancestral state reconstruction (ASR) to infer the gene content of internal nodes in the reference tree (Fig. 1), then extends these to all tips in the tree to predict gene content associated with OTUs. PICRUSt includes a precomputed set of states for all tips in the Greengenes reference tree: if this is the desired set of predictions, then Subheading 3.1 can be skipped.

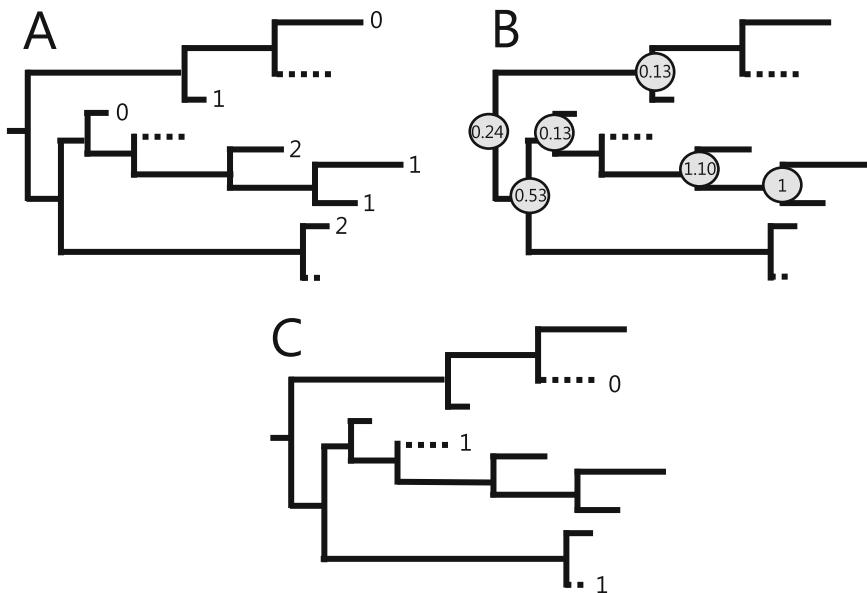


Fig. 1 Example of PICRUSt’s genome prediction workflow. A phylogenetic tree is built based on 16S rRNA gene sequences in reference genomes (solid lines) and sequences that do not correspond to a reference genome (dotted lines). **(a)** Counts for a trait of interest are inputted for reference genomes. **(b)** Ancestral state reconstruction is performed based on these trait values and the phylogenetic tree based on their 16S rRNA gene sequence identity to predict trait values at internal nodes (gray circles) in the tree. **(c)** These predicted internal nodes are extended to unknown trait values in the tree, which results in the predicted trait value based on sequence alone

3.1 Genome Prediction

- Step 1. Prepare trees for analysis (*format_tree_and_trait_table.py*): this step performs quality control on the input datasets, resolving polytomies (i.e., unresolved lineages) by introducing random short branches to create nested bifurcations, and generating a pruned tree that includes only those tips with marker-gene copy-number predictions.
- Step 2. Reconstruct ancestral states in the reference tree (*ancestral_state_reconstruction.py*): PICRUSt can use any of a series of different ASR methods to assign predictions to internal nodes in the pruned tree. The default method is phylogenetic independent contrasts (PIC; [17]), implemented in the R package ape [18]. This method assumes that trait evolution occurs along a random walk (Brownian motion). This assumption implies that the difference in trait values for two species in a phylogeny is normally distributed with mean = 0 and variance proportional to the amount of time separating them. Ancestral states are reconstructed through a least squares method based on this model, a phylogenetic tree, and observations for extant taxa. In PICRUSt's implementation of this method extant taxa with missing trait values also have their states inferred. Maximum-likelihood methods can be used instead, but require much longer computational times (see Note 4). This step produces two output files which correspond to ASRs for marker-gene counts and functional traits.
- Step 3. Predict traits for tips that lack sequenced genomes (*predict_traits.py*): OTUs that lack associated traits are assigned functions by identifying the nearest corresponding ASR in the pruned tree. The trait for the organism is then predicted based on a branch-length weighted average of the ancestral node and its close relatives. To save computational time, the user can specify a tab-separated file that limits the OTUs for which traits are to be predicted. This step can also produce confidence intervals for each predicted trait, which reflect the uncertainty in the prediction (see Note 5).

3.2 Metagenome Prediction

- Step 1. Normalize each OTU's abundance by their number of marker genes (*normalize_by_copy_number.py*): since organisms can have varying numbers of marker genes in their genome, normalization is needed to mitigate the overrepresentation of genomes with higher marker-gene counts. The marker-gene (typically 16S) copy number is predicted for each OTU using the ASR method described above.
- Step 2. Predict metagenome functional abundances (*predict_metagenomes.py*): multiply each OTU's normalized abundance by that OTU's pre-calculated functional category abundances to produce a table of functional category

counts per sample. At this point, different classification schemes can be outputted (*see Note 3*). Users also have the option of outputting the nearest sequenced taxon index (NSTI) for each sample (*see Note 6*), which can help them evaluate how well the genome database represents the community.

- Step 3. *Optionally* collapse hierarchical data to a specified level (*categorize_by_function.py*). This step is helpful for users who would like to interpret KEGG or COG (*see Note 3*) functional categories at higher levels. The most common use is to collapse KEGG_Description tables (outputted by **step 2** above) into KEGG_Pathways tables, which include more interpretable descriptions. Note that many functions are involved in multiple higher categories (e.g., many genes are involved in multiple pathways). In such cases where a many-to-one relationship exists, the lower functional category is counted once for each higher category, which means the total counts in the table will increase.
- Step 4. *Optionally* partition metagenome functional categories by OTU (*metagenome_contributions.py*). This script outputs the breakdown functions by OTUs. The metagenome predictions calculated in **step 2** are aggregates of these OTU-by-function predictions, but specific subdivisions by OTU are not output by default. When partitioning predictions by OTUs, users should specify a small number of functional categories of interest since otherwise the output table can be very large and challenging to analyze. An R script (*plot-metagenome_contributions.R*) can be used to quickly generate stacked bar charts to explore which OTUs (or higher taxonomic level) are contributing to a particular function prediction within each sample. Currently this script is available as part of the Microbiome Helper repository (https://github.com/mlangill/microbiome_helper).

3.3 Analyze PICRUSt Predictions

STAMP [19] is a straightforward program to use that allows researchers to explore and perform statistical tests within microbiome datasets. QIIME [10] is a Python package that provides a range of microbiome data analysis options on the command line. Detailed tutorials on how to use STAMP and/or QIIME with PICRUSt's functional predictions are available online: http://picrust.github.io/picrust/tutorials/downstream_analysis.html#downstream-analysis-guide. A brief description of these workflows is below.

The easiest way to analyze PICRUSt output is with the STAMP software package, which is designed specifically for analyzing microbial taxonomic and functional datasets. A major advantage of STAMP is that it can be run as a graphical user interface on most personal computers and does not require any command-line

knowledge. Also, STAMP produces several useful visualizations for exploring a dataset and can quickly perform a variety of statistical tests to identify differences between samples.

Since PICRUSt predictions are in BIOM format they can also be readily used with several useful QIIME scripts. For example, many alpha and beta-diversity metrics as well as summary plots can be quickly generated. Users could quickly compute the numbers of unique gene families predicted within each sample and compare whether functional richness differed based on sample variables of interest. Similarly, the difference in the abundance of functional categories can be computed between pairwise samples to allow the similarity between samples to be visualized through an ordination plot, such as principal coordinates analysis. In addition, users could create stacked bar charts of higher-level functional categories to identify large-scale differences between samples. There are also several scripts which can allow users to perform rarefaction, filter functions and/or samples, and sort tables. Since QIIME was created with marker-gene surveys and taxonomic predictions in mind, QIIME scripts often refer to taxa or OTUs in their descriptions. However, most of these scripts can be used with any BIOM table, including those produced by PICRUSt. A list of QIIME scripts that could be used with PICRUSt output is available here: <https://picrust.github.io/picrust/tutorials/qiimeTutorial.html>.

4 Notes

1. Typically OTU picking is performed based on an identity cutoff of 97%. This setting is arbitrary and drastically different clusters can be produced when this setting is changed. In particular, recently there have been several sequence denoising tools developed that can theoretically cluster reads at 100% identity (i.e., real biological sequences), such as DADA2 [20] and Deblur [21]. Currently PICRUSt does not support the use of these tools in the default pipeline since the pre-calculated files are restricted to the Greengenes database.
2. PICRUSt requires all OTUs to correspond to a defined reference sequence. This limitation is due to the requirement of PICRUSt's genome prediction steps for a database of known marker-gene sequences to produce the functional predictions for each OTU. In practice, most users use pre-calculated files that were run on the entire Greengenes 16S database, which do not contain novel OTUs identified in a particular dataset.
3. There are currently three functional classification schemes which can be used with PICRUSt. The first is the Kyoto Encyclopedia of Genes and Genomes (KEGG; [22]) database, which aims to systematically assign higher-level functions to all genes in sequenced genomes. The second is the framework of

clusters of orthologous groups (COGs; [23]), which is a similar effort to combine orthologous genes into functional categories. Lastly, Rfam [24] is a database of noncoding RNA families identified based on sequence similarity. Note that both KEGG and COG gene families can be collapsed to higher functional levels, but Rfam RNA families cannot. As genome annotation methods change over time, PICRUSt will include other functional classification systems such as MetaCyc and the Gene Ontology.

4. PICRUSt can use four different ASR methods. By default, PIC is implemented (*see* Subheading 3.1) because it offers an effective trade-off between speed and accuracy. This choice is largely due to how much fast this method runs. Another extremely fast method is Wagner Parsimony [25], also known as Maximum Parsimony. This method is the simplest method of ASR, which involves minimizing the total number of changes in the trait across the phylogeny. Maximum-likelihood and restricted maximum-likelihood methods of ASR are also implemented in the ape R package [18] used by PICRUSt. These methods are more accurate, but have much longer running times than simpler methods like PIC.
5. PICRUSt outputs 95% confidence intervals for each functional category prediction, which are also pre-calculated at the genome prediction stage. These intervals are based off the probability distributions returned by the ASR. These confidence intervals can be checked for functions of particular biological interest to ensure that differences observed between samples are significant.
6. The nearest sequenced taxon index (NSTI) quantifies how closely the genome database represents the community of OTUs. NSTI is the sum of branch lengths between each OTU and the nearest sequenced genome in the phylogenetic tree, weighted by the relative abundance of each OTU. This value is summed across all OTUs so that one NSTI is produced per sample. PICRUSt's inference accuracy is negatively correlated with NSTI values [3], so analyzing these values can help users determine how much confidence they should have in functional inferences.

Acknowledgments

We would like to acknowledge the other coauthors of PICRUSt who helped develop and test the software, as well as the many users from the PICRUSt mailing list that have provided insightful questions and comments.

References

1. Segata N, Huttenhower C (2011) Toward an efficient method of identifying core genes for evolutionary and functional microbial phylogenies. *PLoS One* 6:e24704
2. Snel B, Bork P, Huynen MA (1999) Genome phylogeny based on gene content. *Nat Genet* 21:108–110
3. Langille MG, Zaneveld J, Caporaso JG et al (2013) Predictive functional profiling of microbial communities using 16S rRNA marker gene sequences. *Nat Biotechnol* 31:814–821
4. Gevers D, Kugathasan S, Denson LA et al (2014) The treatment-naïve microbiome in new-onset Crohn's disease. *Cell Host Microbe* 15:382–392
5. Zarraonaindia I, Owens S, Weisenhorn P et al (2015) The Soil Microbiome Influences Grapevine-Associated Microbiota. *MBio* 6:e02527–e02514
6. Morrow KM, Bourne DG, Humphrey C et al (2015) Natural volcanic CO₂ seeps reveal future trajectories for host-microbial associations in corals and sponges. *ISME J* 9:894–908
7. Aßhauer KP, Wemheuer B, Daniel R, Meinicke P (2015) Tax4Fun: Predicting functional profiles from metagenomic 16S rRNA data. *Bioinformatics* 31:2882–2884
8. Iwai S, Weinmaier T, Schmidt BL et al (2016) Pipillin: improved prediction of metagenomic content by direct inference from human microbiomes. *PLoS One* 11:e0166104
9. Quast C, Pruesse E, Yilmaz P et al (2013) The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. *Nucleic Acids Res* 41:590–596
10. Caporaso JG, Kuczynski J, Stombaugh J et al (2010) QIIME allows analysis of high-throughput community sequencing data. *Nat Methods* 7:335–336
11. Schloss PD, Westcott SL, Ryabin T et al (2009) Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Appl Environ Microbiol* 75:7537–7541
12. McDonald D, Clemente JC, Kuczynski J et al (2012a) The Biological Observation Matrix (BIOM) format or: how I learned to stop worrying and love the ome-ome. *GigaScience* 1:7
13. McDonald D, Price MN, Goodrich J et al (2012b) An improved Greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea. *ISME J* 6:610–618
14. Markowitz VM, Chen IMA, Palaniappan K et al (2014) IMG 4 version of the integrated microbial genomes comparative analysis system. *Nucleic Acids Res* 42:D560–D567
15. Matsen FA, Kodner RB, Armbrust EV (2010) pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree. *BMC Bioinform* 11:538
16. Berger SA, Krompass D, Stamatakis A (2011) Performance, accuracy, and web server for evolutionary placement of short sequence reads under maximum likelihood. *Syst Biol* 60:291–302
17. Felsenstein J (1985) Phylogenies and the Comparative Method. *Am Nat* 125:1–15
18. Paradis E, Claude J, Strimmer K (2004) APE: analyses of phylogenetics and evolution in R language. *Bioinformatics* 20:289–290
19. Parks DH, Tyson GW, Hugenholtz P, Beiko RG (2014) STAMP: statistical analysis of taxonomic and functional profiles. *Bioinformatics* 30:3123–3124
20. Callahan BJ, McMurdie PJ, Rosen MJ et al (2016) DADA2: high-resolution sample inference from Illumina amplicon data. *Nat Methods* 13:581–583
21. Amir A, McDonald D, Navas-Molina JA et al (2017) Deblur rapidly resolves single-nucleotide community sequence patterns. *mSystems* 2:e00191–e00116
22. Kanehisa M, Goto S, Sato Y et al (2012) KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Res* 40:109–114
23. Tatusov RL, Koonin EV, Lipman DJ (1997) A genomic perspective on protein families. *Science* 278:631–637
24. Burge SW, Daub J, Eberhardt R et al (2013) Rfam 11.0: 10 years of RNA families. *Nucleic Acids Res* 41:226–232
25. Swofford DL, Maddison WP (1987) Reconstructing ancestral character states under Wagner parsimony. *Math Biosci* 87:199–229



Chapter 12

Metagenome Assembly and Contig Assignment

Qingpeng Zhang

Abstract

The recent development of metagenomic assembly has revolutionized metagenomic data analysis, thanks to the improvement of sequencing techniques, more powerful computational infrastructure and the development of novel algorithms and methods. Using longer assembled contigs rather than raw reads improves the process of metagenomic binning and annotation significantly, ultimately resulting in a deeper understanding of the microbial dynamics of the metagenomic samples being analyzed. In this chapter, we demonstrate a typical metagenomic analysis pipeline including raw read quality evaluation and trimming, assembly and contig binning. Alternative tools that can be used for each step are also discussed.

Key words Metagenomics, Assembly, Binning, Quality evaluation, Annotation

1 Introduction

One of the most remarkable developments in the metagenomics research in the past several years is that metagenomic assembly has become not only a feasible step in the pipeline of metagenomic data analysis, but also a routine and essential step for almost all metagenomic research projects. Reconstructing longer genomic fragments from short sequencing reads has been a challenging task because of the high diversity of microbial samples and limited computational resources. For example, a study investigating assembly of two soil metagenomes found that only 10–15% of the raw reads could be mapped back to assembled contiguous sequences (“contigs”) [1]. However, recent developments have contributed to significant improvement in the quality of assembly of metagenomic reads. Firstly, improvements in sequencing technology have led to a lower cost of acquiring more and longer sequencing reads, which makes it easier to achieve higher sequencing depth for a microbial sample and improves the ability of assemblers to reconstruct longer contigs from sequencing reads. Secondly, bottlenecks in contig assembly have been largely overcome through improvements in computational infrastructures, especially with the access to large-

memory computing nodes. Finally, novel assembly algorithms and related methods have been specifically developed to efficiently process metagenomic data.

The improved ability to obtain improved assemblies is revolutionary to the analysis of metagenomic data. With the longer assembled contigs compared to shorter raw reads, it is possible to acquire more complex and longer genetic elements of interest, because it is easier to get significant alignments between longer sequences and reference sequences compared to shorter reads. This significantly enhances metagenome annotation. For genome binning of sequences, which means the assignment of sequences into groups representing individual species genomes, using longer assembled contigs is also recommended when available [2]. The important features used for binning, such as compositional signals like GC content and k-mer distributions, as well as similarity-based signals that rely on reference databases are more noticeable and stable on longer sequences, hence the performance significantly can be improved through the use of longer sequences [2].

With an increasing reliance on assembled contigs, a whole new category of tools has been developed especially for binning contigs [2]. Many of them integrate composition-based signals with coverage profiles across multiple samples to achieve better binning performance [3–5]. Previous binning efforts were based on raw reads, but with improved quality of assembly and more sophisticated binning methods on contigs, it is routine to assemble the shorter raw reads into contigs before binning and other downstream analyses [2].

This chapter outlines a typical pipeline of metagenomic assembly and taxonomic binning, beginning with the quality evaluation and trimming of raw reads. For each step, one typical tool will be introduced as an example. Alternative tools will be discussed in Subheading 4 afterward. The fields of metagenomic assembly and binning are still being actively explored and new tools are continually being developed. It is difficult to claim a clear winner for each step in the pipeline although several benchmarking studies have tried to perform objective and systematic comparisons of the performance of those tools [6–8].

2 Data Sources and Software

2.1 Data Sources

In a sequencing facility, after the actual sequencing is complete, facility staff may perform initial and/or routine analysis of the raw reads, which may include some of the procedures described in Subheading 3 below, such as quality evaluation of sequence reads, filtering out reads with low quality or high error rate, and assembly. However, the raw reads should be available for download from the server provided by the sequencing facility. This allows the users to

make use of the methods introduced in this chapter to the raw reads, either to improve the quality of analysis or to build customized pipelines that fulfill the requirements of specific scientific problems.

2.2 Program Availability

All the tools discussed in this chapter are available online.

- FastQC (v0.11.5): <http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>
- Trimmomatic (v0.36): <http://www.usadellab.org/cms/?page=trimmomatic>
- FASTX-Toolkit (v0.0.13): http://hannonlab.cshl.edu/fastx_toolkit/index.html
- MultiQC (v1.4): <http://multiqc.info/>
- Cutadapt (v1.16): <https://github.com/marcelm/cutadapt>
- Khmer (v2.1.2): <https://github.com/dib-lab/khmer>
- MEGAHIT (v1.1.2): <https://github.com/voutcn/megahit>
- DISCO (v1.2): <https://github.com/abiswas-odu/Disco>
- QUAST (v4.6.3): <http://quast.sourceforge.net/quast>
- metaQuast (v4.6.3): <http://quast.sourceforge.net/metaquast>
- Bowtie 2 (v2.4.3): <http://bowtie-bio.sourceforge.net/bowtie2/index.shtml>
- MetaBAT (v0.30.3): <https://bitbucket.org/berkeleylab/metabat>
- CheckM (v1.0.9): <http://ecogenomics.github.io/CheckM/>

3 Methods

Below we will describe the general pipeline to perform metagenomics data analysis, from quality evaluation of raw sequencing reads, to the assignment of assembled contigs into separate taxonomic bins that represent different taxa. For each step, there are multiple software packages that can be used. Here we use the more popular packages as examples. Different software packages have their respective advantages and disadvantages and we will try to discuss these in more details in the next section of this chapter.

3.1 Quality Evaluation of Raw Sequencing Reads Using FastQC

1. We want to check the quality of the raw sequencing reads acquired from sequencers before we start any further analysis. FastQC is a popular tool developed by the Bioinformatics Group at the Babraham Institute, which can perform a modular set of analyses and automatically generate a sophisticated report about the quality of raw sequence data.

2. FastQC can be executed in two modes. One is a stand-alone application with interactive features. This is good for relatively smaller data sets and is convenient to obtain an evaluation report immediately. For larger data sets, or for use in an automated pipeline, FastQC also has a noninteractive mode with a command-line interface. The evaluation reports will be generated as a group of HTML files for further reference (*see Note 1*). For example, to evaluate the quality of a fastq file with sequencing reads called “sample.fastq,” execute the command:

```
fastqc sample.fastq .
```

After the computing is finished, in the directories such as *sample_fastqc*, you would find the report from FastQC as an html file such as *fastqc_report.html*. You can check the content of the HTML file with any browser, such as Firefox.

```
firefox sample_fastqc.html
```

3. FastQC performs a set of analyses and generates different sets of statistics. They include basic statistics such as the total number of sequences in the dataset and the average sequence length, and more sophisticated statistics such as sequence quality per base and per sequence, or sequence duplication levels showing the percent of unique sequences and duplicated sequences. For each module of analysis, FastQC uses three labels to give a straightforward pass/fail result: a green tick indicates “normal” data of satisfactory quality, an orange triangle indicates data that are “slightly abnormal,” while a red cross indicates that this aspect of the data is very unusual. We are more interested in the red crosses. We need to investigate and try to correct the parameters that are marked as red crosses in order to improve the quality of the sequencing data and the downstream analysis. These labels offer the user a guide by which specific results can be investigated further. It should be noted that these straightforward pass/fail tests conducted by FastQC do not take the characteristics of the specific experiment into account. Consequently users should be cautious to form decisions based only on the labels given by FastQC.
4. Out of the many statistics FastQC provides, per-base sequence quality is probably the most important for users to check. If a large number of sequences from a sample have abnormally low-quality scores, or the quality scores of the bases close to the 3' end drop dramatically, further investigation will be necessary and the user may wish to discard the sequences from this specific sample. The statistics FastQC provides give the users important information about how to trim and filter the sequences, which will be discussed in the next section.

3.2 Sequence Trimming and Filtering with Trimmomatic

1. Errors do occur in the sequence data, from sequencing error or from library preparation. It is important to remove those errors by trimming and filtering the sequences, which can reduce subsequent assembly errors.
2. To trim sequences, bases located on both ends of sequences with low-quality scores are removed. We also use reference adaptor sequences to identify and filter the adaptor segments in sequence reads. Finally sequences shorter than a given threshold such as 35 bps are discarded, as from experience, such short sequences may significantly impact assembly quality. Sequence trimming and filtering will reduce assembly error but will also remove contents that may be useful potentially. Careful consideration should therefore be used in deciding what parameters to use.
3. Trimmomatic is a popular software package that can be used for read trimming and filtering [9]. It implements multiple trimming steps as discussed above, such as removal of adapters, low-quality sequence, and short sequences. It has a command-line interface that allows filtering and trimming parameters to be predefined prior to integration into automated pipelines. To trim a fastq file of single-end sequence data called “sample.fastq” against a panel of adaptors specified in a file called “Adapters” we would run Trimmomatic by

```
java -jar Trimmomatic-0.36/trimmomatic-0.36.jar SE sample.fastq sample_trim.fastq ILLUMINACLIP:Adapters:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:50
```

For paired-end sequencing reads data, two input files and four output files are specified to handle the situation where one of the paired reads survives the filtering, while the other read does not (*see Note 2*).

4. After trimming and filtering sequences by Trimmomatic, it is advised to run FastQC again to check the quality of the trimmed sequences and to see if there are any remaining problems with the trimmed sequence data:

```
fastqc sample_trim.fastq
firefox sample_trim_fastqc.html
```

See Notes 3, 4, and 5 for additional discussion of FastQC and Trimmomatic.

3.3 Metagenome Assembly with MEGAHIT

1. After trimming and filtering, we need to assemble the reads into longer contigs. Unlike genome assembly, metagenome assembly is a more challenging task for a variety of reasons including high taxonomic diversity of a metagenome sample, low abundance of certain taxa, and lower sequencing depth. A variety of software packages have been developed specifically

for metagenome assembly, including software packages initially designed for genome assembly (*see Note 6*). Users can refer to benchmarking studies that use mock or real metagenome sequencing data to compare assembler performance [6–8]. Here we use MEGAHIT [10], widely used assembler designed for metagenomes that has been optimized for processing speed. MEGAHIT represents an extension of two other popular assembly tools, SOAPdenovo [11] and SOAPdenovo2 [12], which were initially designed for assembly of single genomes [12]. The command to run MEGAHIT is straightforward. Users should refer to the documentation for an up-to-date list of options.

2. MEGAHIT can accept sequence files in fasta or fastq formats, or in compressed or uncompressed format. For paired-end, interleaved-paired-end and single-end read datasets, the user should specify the following flags, $-1/-2$, -12 and $-r$, respectively. To assemble paired-end reads, the user can execute the following command:

```
megahit -1 sample_trim_pe_1.fastq -2 sample_trim_pe_2.
fastq -o sample_trim.megahit
```

The previous step of trimming and filtering will generate “orphan” reads where one read from a given pair has been filtered out and discarded while the other one was retained. We need to separate the filtered reads set into the two groups of paired-end and single-end to be fed into MEGAHIT for assembly. In that case, flags $-1/-2$, or -12 and $-r$ will both be used for paired-end reads and single-end reads as input files. Just the files in -1 and -2 should be paired and in proper order.

3. One of the most important assembly parameters is the size of k-mers to build the de Bruijn graph, which is the underlying data structure that many assemblers rely on (*see Note 7*). Unlike many other assembly tools (*see Note 8*), MEGAHIT makes use of a multiple k-mer size strategy. Multiple de Bruijn graphs are built iteratively from a small k to a larger k, where smaller k-mer sizes are used for filtering erroneous edges in the graph and bridging the gaps in the low-coverage regions and larger k-mer sizes are used for resolving repeats. MEGAHIT can accept a variety of k-mer sizes by defining a range with the options $--k-min$, $--k-max$ and $--k-step$. According to the recommendation of the developers, for highly complex metagenomics data like soil, a larger k-min such as 27 is recommended. For high-depth data, large k-max (25–31) is recommended. And for low-coverage data, smaller k-step such as 10 is recommended (*see Notes 9 and 10*).
4. Other options are available to further tune the performance of the assembler. For example, a “mercy k-mer” strategy may be

specified to deal with low-coverage sequence datasets. The option `--kmin-1pass` can be specified to reduce memory usage of datasets with low sequencing depth, such as soil metagenomics dataset. Users can refer to the online documentation for more assembly tips.

5. Once completed, the quality of the assembly needs to be evaluated. Generally the first thing to check is the log file generated by the assembler, in which basic assembly statistics are provided. These include the number of assembled contigs, the length of contigs, and the N50 of the assembly. The N50 length is defined as the length of the shortest sequence at 50% of the genome, which means the total number of bases in the contigs shorter than the N50 is similar to the total number of bases in the contigs longer than the N50. MetaQUAST [13] is a metagenomic extension of the QUAST package [11] that addresses metagenome-specific considerations such as accepting multiple references to make multi-genome tables and dealing with the commonly unknown species content. These assembly evaluation tools can be applied to different assemblies generated by different assemblers or to assess the impact of using different sets of parameters to identify the strategy that yields the optimal assembly.

3.4 Assignment of Assembled Contigs Using MetaBAT

1. After assembly we have contigs with various lengths. To answer specific questions about the individual species and the interactions between them, we need to assign the contigs into groups (bins) based on putative taxonomic origins. Ideally each group will consist of genomic content from a single taxon. This grouping process is referred to as “genome binning,” and may or may not assign taxonomic labels to the bins created [7]. The various contig-binning tools discussed in this section generally do not assign taxonomic label to contigs. There is a different category of tools that can do such taxonomic labeling to contigs (*see Note 11*).

Contig binning can be performed through a variety of approaches (*see Note 12*). Traditionally the characteristics of sequences themselves such as k-mer frequency profiles are used, with the assumption that sequences from the same genome tend to have similar characteristics like k-mer frequency. Subsequently, it was discovered from analysis of multiple samples across a variety of metagenomics projects that sequences from the same genome tend to have similar abundance profiles across samples. This resulted in the development of abundance profile-based binning methods that have recently proved effective, particularly with the integration of k-mer frequency profiles [1]. In the past few years, multiple contig-binning tools using these integrated strategies have been developed and

widely used [2]. Here we use MetaBAT [14], one of the most widely used automatic contig-binning tools that uses both abundance profiles and k-mer frequency distributions, to demonstrate a typical approach to binning contigs.

2. Since MetaBAT uses both abundance profile and k-mer frequency to bin contigs, in addition to the input file containing the assembled contigs to assign, MetaBAT also requires a file specifying coverage information across samples for each contig. We consequently first need to generate this coverage profile for the contigs across samples. There are multiple mappers available that can be used to map sequence reads against assembled contigs to obtain read coverage. Bowtie 2 [15] has been used widely for its speed and memory efficiency especially for aligning short reads that are common for metagenomic sequencing data (*see Note 13*). For the reads from each sample, we can run Bowtie with this command:

```
bowtie -S assembly.fa sample_x.fastq sample_x.sam
```

3. The alignment of reads back to assembly results in the generation of a SAM or BAM file, as above, which contains the alignment information [16]. SAM files are in plain text format while BAM files are binary versions of the SAM file, which can save storage space and facilitate the downstream analysis. The SAM file can be converted to BAM file and sorted with commands:

```
samtools view -bS -o sample_x.bam sample_x.sam  
samtools sort sample_x.bam sample_x.sorted.bam
```

4. There is a program “jgi_summarize_bam_contig_depths” included in MetaBAT that can be used to convert the BAM files of each sample into a “depth.txt” file after the mapping is finished.

```
jgi_summarize_bam_contig_depths --outputDepth depth.txt --  
pairedContigs paired.txt *.bam
```

The MetaBAT tool uses this file to assign the contigs provided in the input file into different bins.

```
metabat -i assembly.fasta -a depth.txt -o bins_dir/bin
```

5. After the contigs are assigned to bins by MetaBAT, the quality of bins is assessed. Of particular interest are completeness—how completely each bin represents an individual taxon, and contamination—how many contigs in a bin were clustered incorrectly. To assess the quality of a recovered genome, a limited number of single-copy, universally conserved core “marker” genes are examined. CheckM [17] uses a broad list of “marker genes” that are specific to individual genomes in a

reference genome tree, together with other key genomic characteristics such as G+C content and coding density. Assuming the putative genomes in bins are in the directory `bins_dir/bin` with “.fa” as the file extension, and the CheckM results are to be stored in `bins_dir/checkM`, the command to run CheckM would be:

```
checkm lineage_wf -t 8 -x fa bins_dir/bin bins_dir/checkM
```

6. CheckM uses the sequences from different genome bins as input and calls genes on the sequences automatically with the Prodigal microbial gene-finding program before examining the presence of marker genes. Users can also use an external gene-calling program and provide the amino-acid sequences of the identified genes to CheckM in a FASTA file with the flag --genes.

4 Notes

1. To reduce the memory requirement and increase the performance, for certain statistics FastQC uses only a subset of the full dataset. For example, duplicate sequence analysis is conducted only for the first 100,000 sequences in each file analyzed.
2. It is advisable to name the output files by adding more suffixes. The resulting file name may look redundant, but for a pipeline with multiple steps, using this filename convention will keep track of the steps more clearly and avoid potential mistakes.
3. Besides FastQC, the FASTX-Toolkit can be used for quality evaluation before and after sequence trimming and filtering. It can also be used for trimming like Trimmomatic, however our preference is for Trimmomatic, since we believe it has better performance and includes more features. An alternative for adaptor removal is Cutadapt [18].
4. Besides the routine trimming and filtering methods implemented in Trimmomatic, there is an optional trimming approach based on k-mer spectra [19]. Trimmomatic performs trimming using only the quality score, which represents the probability that a base call is an error, even bases with high scores can represent errors. As an alternative to the quality score-based trimming, k-mer spectral error trimming attempts to remove the low-abundance k-mers, based on the hypothesis that low-abundance k-mers in a sequence data set with high coverage are likely to be erroneous. There is a script “trim-low-abund.py” that can be used for such k-mer spectral error trimming in khmer [20], an efficient library and toolkit for k-mer based analysis and transformations, developed

specifically for scaling assembly of metagenomes and mRNA using short read sequences. To avoid trimming the low-abundance k-mers because of low sequence coverage or low species abundance, the script first check the overall coverage of a read and only does such low-abundance k-mers trimming on reads with high coverage. Further analysis demonstrates that such k-mer spectral error trimming generally outperforms quality score-based trimming [21].

5. If there are multiple samples to process, MultiQC [22] is a nice tool that aggregates the results from supported bioinformatics tools such as FastQC and Trimmomatic across many samples into a single report.
6. The methods to trim and filter the reads further before assembly, such as digital normalization and partitioning, are optional. Generally, the purpose of these methods is to decrease assembler memory requirements. Whether or not to use those methods depends on the size and characteristics of the sequence datasets such as the sequencing error, and sequencing depth as well as the assembler being used. Some new assemblers such as MEGAHIT introduced in this chapter are more efficient in memory usage and can better handle the variance of read coverage. Consequently, these methods may be less useful for assemblers such as MEGAHIT relative to other assemblers that are likely to require more memory.
7. De Bruijn graph-based approaches are commonly used in modern sequence assemblers, especially for assembling short-read data. We first define the k-mer spectrum of a genome as a list of all oligomers of a given length k present in the sequenced genome. The de Bruijn graph approach starts by generating the k-mer spectrum for the set of reads used in an assembly. The k-mer spectrum also includes abundance information which can be used to solve the repeats problem later. Next a de Bruijn graph can be constructed by using the k-mer spectrum. The de Bruijn graph contains the $k-1$ length prefixes and suffixes of the original k-mers as nodes, and if the adjacent $k-1$ -mers have an exact overlap of length $k-2$, the two nodes are linked by an edge. The assembly problem becomes the problem to find a path through every edge in the graph. The de Bruijn graph approach is highly suited for the assembly of short sequencing reads, however since they have split each read into short k-mers, some context information inherent in the reads is missing in the de Bruijn graph.
8. Many assemblers exist for assembling metagenomic data. The older assemblers initially designed for single-genome assembly such as Velvet [23] and SPAdes [24] can be used for metagenomic data, though they may not be the best choice for datasets derived from samples containing many taxa that share

similar genomic signatures (e.g., that have similar k-mer profiles). Recently more assemblers were developed specifically for metagenomic data, some of which have evolved from single-genome assemblers such as MetaVelvet [25], metaSPAdes [26], IDBA-UD [27], and Ray Meta [28], while others have been designed and developed specifically for metagenomic datasets (e.g., MEGAHIT and DISCO). Each of these assemblers has distinct advantages and disadvantages. It is still difficult to predict which assembler will provide the best performance for any specific metagenomic dataset. Recently several groups of researchers have published their work benchmarking or comparing the assemblers on metagenomic data, which are helpful in guiding choice of assembler [6, 7].

9. Generally to assemble a metagenome adequately, the sequencing coverage should be around $20\times$ [6]. However, reads from high-abundance taxa will often have much higher coverage than $20\times$. In these situations, redundant reads may be removed to reduce the memory requirement of the assembling step and minimize assembly errors. This approach, known as *digital normalization*, also implemented as the “normalize-by-media.py” script in the khmer package, evaluates the coverage of reads by the median frequency of k-mers on the reads and discards the reads if the coverage is above a specified threshold [29]. The parameters of the size of k-mer k and the coverage threshold C can be supplied to the script. After using digital normalization to remove the redundant reads, the “filter-abund.py” script in khmer can be used to remove the reads with low coverage that cannot be used for assembly.
10. For large metagenomic datasets (>50 million reads), it may be useful to use a method called “partitioning” to separate the reads that do not connect to each other into different subgroups and then assemble the subgroups separately. This reduces computing requirements and memory usage [1]. Several scripts are included in the khmer package for this purpose. These preassembly filtering approaches, including partitioning and digital normalization, have been applied to successfully assemble two large soil metagenomes. Please refer to khmer documentation and [30] for more details.
11. Some examples of binning tools that can assign taxonomic labels to contigs include Kraken [31], PhyloPythiaS+ [32], taxator-tk [33], and many others. Mostly these taxonomic binning tools use the compositional features such as G+C content, the abundance profile of k-mers in a sequence, and/or the similarity between the sequence and a reference database of known genes and/or genomes. Some of these taxonomic binning tools were developed for reads rather than contigs (e.g., MEGAN [34]). There are also integrated annotation pipelines like IMG/M

[35] and MG-RAST [36] that perform taxonomic binning for the input reads with a user-friendly web interface.

12. As with assemblers, there have been many binning tools developed recently. Several tools have been developed with the strategy of combining both coverage profile and k-mer frequency distributions, such as CONCOCT [4], Maxbin2 [37], GroopM [38], and MetaBAT [14]. Like the various assemblers, each of these binning tools has their respective advantages and disadvantages. For example, GroopM requires at least three samples to run [39]. It is difficult to predict which binning tool will yield the best performance on any given dataset. The CAMI (Critical Assessment of Metagenome Interpretation) project is a comprehensive benchmark of most of the widely used computational software on metagenomics assembly, genome binning, taxonomic profiling and binning that can serve as a good reference for readers to make the decision [7]. The other strategy is to run different binning tools, and summarize those results from these tools in order to get better results than from any single binning tool. DAS Tool is a new software package applying a strategy that attempts to exploit the strengths of individual binning tools by combining their results [5]. In DAS Tool, a scoring function is built to estimate the quality and completeness of the resulting bins from different binning tools based on the abundance of duplicated single-copy genes in a bin. Then DAS Tool can select the highest scoring bins and put them in the resulting bin set.
13. Other than Bowtie 2, there are alternative tools to map sequence reads to assembled contigs to generate the coverage profile required for many of the binning tools mentioned above, such as BWA [40] and BBMap. Of recent interest is BWA-mem [41], a new tool based on BWA, which promises improved performance.

References

1. Pell J, Hintze A, Canino-Koning R et al (2012) Scaling metagenome sequence assembly with probabilistic de Bruijn graphs. Proc Natl Acad Sci U S A 109:13272–13277. <https://doi.org/10.1073/pnas.1121464109>
2. Sangwan N, Xia F, Gilbert JA (2016) Recovering complete and draft population genomes from metagenome datasets. Microbiome 4:8. <https://doi.org/10.1186/s40168-016-0154-5>
3. Kang DD, Froula J, Egan R, Wang Z (2014) MetaBAT: Metagenome binning based on abundance and tetranucleotide frequency. No. LBNL-7106E. Ernest Orlando Lawrence Berkeley National Laboratory, Berkeley, CA
4. Alneberg J, Bjarnason BS, de Bruijn I, et al (2014) Binning metagenomic contigs by coverage and composition. Nat Methods 11:1144–1146. doi: <https://doi.org/10.1038/nmeth.3103>
5. Sieber CMK, Probst AJ, Sharrar A et al (2017) Recovery of genomes from metagenomes via a dereplication, aggregation, and scoring strategy. bioRxiv:107789
6. Vollmers J, Wiegand S, Kaster AK (2017) Comparing and evaluating metagenome assembly tools from a microbiologist’s perspective—not only size matters! PLoS One 12: e0169662

7. Sczyrba A, Hofmann P, Belmann P et al (2017) Critical Assessment of Metagenome Interpretation—a benchmark of computational metagenomics software. bioRxiv:99127. <https://doi.org/10.1101/099127>
8. Awad S, Irber L, Brown CT (2017) Evaluating metagenome assembly on a simple defined community with many strain variants. bioRxiv:155358
9. Bolger AM, Lohse M, Usadel B (2014) Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics 30:2114–2120. <https://doi.org/10.1093/bioinformatics/btu170>
10. Li D, Liu C-M, Luo R et al (2015) MEGA-HIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. Bioinformatics 31:1674–1676. <https://doi.org/10.1093/bioinformatics/btv033>
11. Li R, Zhu H, Ruan J et al (2010) De novo assembly of human genomes with massively parallel short read sequencing. Genome Res 20:265–272. <https://doi.org/10.1101/gr.097261.109>
12. Luo R, Liu B, Xie Y et al (2012) SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. Gigascience 1:18. <https://doi.org/10.1186/2047-217X-1-18>
13. Mikheenko A, Saveliev V, Gurevich A (2016) MetaQUAST: evaluation of metagenome assemblies. Bioinformatics 32:1088–1090. <https://doi.org/10.1093/bioinformatics/btv697>
14. Kang DD, Froula J, Egan R, Wang Z (2015) MetaBAT, an efficient tool for accurately reconstructing single genomes from complex microbial communities. PeerJ 3:e1165. <https://doi.org/10.7717/peerj.1165>
15. Langmead B, Salzberg SL (2012) Fast gapped-read alignment with Bowtie 2. Nat Methods 9:357–359. <https://doi.org/10.1038/nmeth.1923>
16. Li H, Handsaker B, Wysoker A et al (2009) The sequence alignment/map format and SAMtools. Bioinformatics 25:2078–2079. <https://doi.org/10.1093/bioinformatics/btp352>
17. Parks DH, Imelfort M, Skennerton CT et al (2015) CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome Res 25:1043–1055. <https://doi.org/10.1101/GR.186072.114> gr.186072.114
18. Martin M (2011) Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnetJ 17:10. <https://doi.org/10.14806/ej.17.1.200>
19. Zhang Q, Awad S, Brown CT (2015) Crossing the streams: a framework for streaming analysis of short DNA sequencing reads. PeerJ Preprints. <https://doi.org/10.7287/peerj.preprints.890v1>
20. Crusoe MR, Alameldin HF, Awad S et al (2015) The khmer software package: enabling efficient nucleotide sequence analysis. F1000Res 4:900. <https://doi.org/10.12688/f1000research.6924.1>
21. Zhang Q, Pell J, Canino-Koning R et al (2014) These are not the k-mers you are looking for: efficient online k-mer counting using a probabilistic data structure. PLoS One 9:e101271. <https://doi.org/10.1371/journal.pone.0101271>
22. Ewels P, Magnusson M, Lundin S, Käller M (2016) MultiQC: summarize analysis results for multiple tools and samples in a single report. Bioinformatics 32:3047–3048. <https://doi.org/10.1093/bioinformatics/btw354>
23. Zerbino DR, Birney E (2008) Velvet: algorithms for de novo short read assembly using de Bruijn graphs. Genome Res 18:821–829. <https://doi.org/10.1101/gr.074492.107>
24. Bankevich A, Nurk S, Antipov D et al (2012) SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. J Comput Biol 19:455–477. <https://doi.org/10.1089/cmb.2012.0021>
25. Namiki T, Hachiya T, Tanaka H, Sakakibara Y (2012) MetaVelvet: an extension of Velvet assembler to de novo metagenome assembly from short sequence reads. Nucleic Acids Res 40:e155–e155. <https://doi.org/10.1093/nar/gks678>
26. Nurk S, Meleshko D, Korobeynikov A, Pevzner PA (2017) metaSPAdes: a new versatile metagenomic assembler. Genome Res 27:824–834. <https://doi.org/10.1101/gr.213959.116>
27. Peng Y, Leung HCM, Yiu SM, Chin FYL (2012) IDBA-UD: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. Bioinformatics 28:1420–1428. <https://doi.org/10.1093/bioinformatics/bts174>
28. Boisvert S, Raymond F, Godzaridis É et al (2012) Ray Meta: scalable de novo metagenome assembly and profiling. Genome Biol 13:R122. <https://doi.org/10.1186/gb-2012-13-12-r122>
29. Brown CT, Howe A, Zhang Q et al (2012) A reference-free algorithm for computational

- normalization of shotgun sequencing data. arXiv preprint arXiv 1203:4802
30. Howe AC, Jansson JK, Malfatti SA et al (2014) Tackling soil diversity with the assembly of large, complex metagenomes. *Proc Natl Acad Sci U S A* 111:4904–4909. <https://doi.org/10.1073/pnas.1402564111>
 31. Wood DE, Salzberg SL (2014) Kraken: ultra-fast metagenomic sequence classification using exact alignments. *Genome Biol* 15:R46. <https://doi.org/10.1186/gb-2014-15-3-r46>
 32. Gregor I, Dröge J, Schirmer M et al (2016) PhyloPythiaS+: a self-training method for the rapid reconstruction of low-ranking taxonomic bins from metagenomes. *PeerJ* 4:e1603. <https://doi.org/10.7717/peerj.1603>
 33. Dröge J, Gregor I, McHardy AC (2015) Taxator-tk: precise taxonomic assignment of metagenomes by fast approximation of evolutionary neighborhoods. *Bioinformatics* 31:817–824. <https://doi.org/10.1093/bioinformatics/btu745>
 34. Huson DH, Auch AF, Qi J, Schuster SC (2007) MEGAN analysis of metagenomic data. *Genome Res* 17:377–386. <https://doi.org/10.1101/gr.5969107>
 35. Markowitz VM, Chen IMA, Chu K et al (2013) IMG/M 4 version of the integrated metagenome comparative analysis system. *Nucleic Acids Res* 42(D1):D568–D573
 36. Wilke A, Bischof J, Gerlach W et al (2015) The MG-RAST metagenomics database and portal in 2015. *Nucleic Acids Res*. <https://doi.org/10.1093/nar/gkv1322>
 37. Wu Y, Simmons BA, Singer SW (2015) MaxBin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets. *Bioinformatics*:1–2. <https://doi.org/10.1093/bioinformatics/btv638>
 38. Imelfort M, Parks D, Woodcroft BJ et al (2014) GroopM: an automated tool for the recovery of population genomes from related metagenomes. *PeerJ* 2:e603. <https://doi.org/10.7717/peerj.603>
 39. Lin H-H, Liao Y-C (2016) Accurate binning of metagenomic contigs via automated clustering sequences using information of genomic signatures and marker genes. *Sci Rep*. <https://doi.org/10.1038/srep24175>
 40. Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* 25:1754–1760. <https://doi.org/10.1093/bioinformatics/btp324>
 41. Li H (2013) Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM arXiv Preprint arXiv:1303.3997



Chapter 13

From RNA-seq to Biological Inference: Using Compositional Data Analysis in Meta-Transcriptomics

Jean M. Macklaim and Gregory B. Gloor

Abstract

The proper analysis of high-throughput sequencing datasets of mixed microbial communities (meta-transcriptomics) is substantially more complex than for datasets composed of single organisms. Adapting commonly used RNA-seq methods to the analysis of meta-transcriptome datasets can be misleading and not use all the available information in a consistent manner. However, meta-transcriptomic experiments can be investigated in a principled manner using Bayesian probabilistic modeling of the data at a functional level coupled with analysis under a compositional data analysis paradigm. We present a worked example for the differential functional evaluation of mixed-species microbial communities obtained from human clinical samples that were sequenced on an Illumina platform. We demonstrate methods to functionally map reads directly, conduct a compositionally appropriate exploratory data analysis, evaluate differential relative abundance, and finally identify compositionally associated (constant ratio) functions. Using these approaches we have found that meta-transcriptomic functional analyses are highly reproducible and convey significant information regarding the ecosystem.

Key words Compositional data, Probability distribution, Transcriptomics, Microbiome, Bayesian inference, Standardized effect, Meta-transcriptome

1 Introduction

High-throughput omics technologies have changed our ability to probe and understand complex microbial ecosystems. Although the most popular approach is targeted sequencing of the bacterial 16S rRNA gene, this only aims to answer “who is there” and not “what are they doing.” Increasingly more functional omics approaches are being used including sequencing the transcribed RNA in a community; this approach is termed meta-RNAseq, or meta-transcriptomics. The underlying goal of this approach is to classify mRNA transcripts into functional predictions (e.g., a metabolic process or pathway) that are novel descriptors of a community/individual organism, or that are differentially expressed or regulated between conditions (e.g., a disease and healthy state).

Sequencing the RNA of entire ecosystems is extremely challenging for several reasons. First, prior to sequencing, it can be challenging to coordinate the collection and processing of samples in order to retain a semblance of their *in situ* expression levels. This includes optimizing “wet lab” protocols for optimal yield and quality, and reducing background contamination from rRNA and host RNA. Second, post-sequencing, it can be difficult to identify and create an appropriate reference library to map all the generated sequenced reads for all the organisms in the samples. In some cases, the mapping problem can be alleviated by assembling RNA-seq reads into putative contigs [1]. However, this becomes less practical when read depth is low causing sparse datasets. In practice, reads from the majority of genes expressed in rare or low abundance species would have very low coverage making assembly difficult if not impossible. On top of inadequate reference information, repetitive sequencing mapping between or within genomes can make resolving nonunique read mapping difficult. Third, both the abundance of the organisms and the expression levels of their genes can exhibit variation in transcript levels; this change can be in the same direction, or in different directions. Thus, the mean expression level–variance relationships can be decidedly difficult to model, making the use of traditional tools that depend on this relationship unreliable [2]. In other words, approaches designed for a single-organism comparative transcriptomics cannot capture the variance between changing expression and gene content/organism abundance. Fourth, there is increasing appreciation that the ecosystem can contain different strains, or even different species, yet can maintain a similar functional profile [3]. This functional redundancy must be specifically addressed in the analysis approach compared to analysis of changing transcript abundance within the same organism under two conditions.

These complexities make the use of many traditional tools problematic, and this chapter provides protocols for the downstream processing of high-throughput sequencing reads from the Illumina sequencing platforms into useful tables, and how to perform exploratory data analysis, differential abundance, and strength of association using a toolkit adapted from those used for compositional data analysis in other domains of science [4–6].

2 Materials

2.1 Computing Requirements

Most of the workflow can be performed on a standard UNIX or MacOS laptop with 4–8Gb of RAM. Mapping and assembly is recommended on a high-powered server with at least 16Gb of RAM and 40Gb of available disk space for output (however this will vary depending on the amount of starting data). Once a counts table is generated, the analysis and visual exploration can be

performed on a standard laptop (tested on: 2013 MacBook Pro 15”, and 2016 MacBook Pro 13”).

The user should be familiar and comfortable with running command line tools and base R as well as installing R packages and modifying scripts as needed.

2.2 Software Installations

- R Statistical Programming Language (at least version 3.3)
 - <https://www.r-project.org>.
- R package libraries required (Source)
 - ALDEx2 (Bioconductor).
 - CoDaSeq (CRAN).
 - igraph (CRAN).
 - car (CRAN).
 - zCompositions (CRAN).
- DIAMOND (<https://github.com/bbuchfink/diamond>)
- Perl version 5
 - <https://www.perl.org>.
- A BASH Shell
 - If you are working on UNIX/MacOS this should be pre-installed.

2.3 Scripts and Workflows

- Sample collection and RNA isolation from vaginal swabs
 - https://github.com/ggloor/MIMB_2018/blob/master/RNA%20isolation_vagswabs.pdf.
- Using DIAMOND to map SEED
 - https://github.com/ggloor/MIMB_2018/tree/master/diamond_to_seed.
- SEED reference database
 - DOI for the dataset: 10.6084/m9.figshare.5836065
- The sample R code needed to reproduce the Compositional Data analysis steps
 - https://github.com/ggloor/MIMB_2018/blob/master/CoDa_code.r.
- ALDEx2 for differential abundance analysis (R Bioconductor package)
 - <https://bioconductor.org/packages/release/bioc/html/ALDEx2.html>.
- CoDaSeq functions for compositional analysis of high-throughput sequencing (R code)
 - <https://github.com/ggloor/CoDaSeq>.

3 Methods

3.1 Sample Collection and Processing for Sequencing

Collection and processing of RNA samples involves more critical timing and consideration compared to collection of DNA samples as the half-life of bacterial mRNA is minutes [7]. Generally, samples should be placed immediately in an RNA preservation buffer (e.g., Bacterial RNAProtect by Ambion, or RNAlater by QIagen) according to the preservation protocol, and then frozen at -80°C within a reasonable timeframe (usually 1–4 h). Depending on the amount of starting material, there are commercial RNA extraction kits available (e.g., PowerSoil Total RNA by Mo Bio), or more traditional phenol-chloroform (e.g., TRIzol) extraction procedures can be used which, in our experience, can produce higher quality and quantity of RNA from limited samples. Quantity of the RNA is best assessed using a Qubit Fluorometer, and quality is best assessed on an Agilent Bioanalyzer.

Since >95% of extracted bacterial RNA is noncoding ribosomal RNA and therefore not informative for functional expression, a step is required in the sample processing to remove these uninformative molecules (termed rRNA depletion, or mRNA enrichment). This step can be done prior to the sequencing library preparation (e.g., with MICROBExpress from Ambion), or with more recent kits available, at the same time as the library generation (e.g., ScriptSeq Complete Kit from Illumina). The choice here should depend on knowledge of the community structure, i.e., a gram-negative targeting kit will be a better choice if the sample is composed of gram-negatives bacteria. As many of these kits rely on a number of probes to target specific sequence variants, it is best to check the manufacturer specifications for whether the probes will be appropriate for the organisms contained in your sample.

In some samples (particularly host-associated samples with low bacterial load such as skin, tissue, or urine) there are potentially large amounts of host RNA contamination. Like the bacterial rRNA, if not removed prior to sequencing the reads from this proportion of RNA will have to be removed computationally and thus will reduce the number reads available for sequencing functional mRNA and thereby reduce your power to detect or differentiate these products. There are various kits available for host RNA removal. If host contamination is minimal, one option is to ensure adequate sequencing so that *in silico* removal of these reads does not have a large impact on coverage for the reads of interest. There is no “rule of thumb” for how much read coverage is required per sample as it depends on a number of things: the complexity of the sample (how many organisms), the relative abundance of the particular organism(s) or transcript(s) you want to assess, whether the reads will be used for assembly of *de novo* transcripts, and the amount of expected “background contamination” (rRNA, host,

poor quality, or other uninformative reads) that will need to be removed in silico. In our experience, about half the reads acquired will map sufficiently, and only ~30% of those will be easily assigned to a functional annotation [3, 8]. This of course will vary greatly with the factors mentioned above, and with choices for downstream analysis including the reference databases. We were able to acquire meaningful information in relatively uncomplex vaginal microbiome samples with ~50 million reads per sample. With this in mind, the choice of Illumina platform for sequencing becomes critical due to read output limits. At this time, the Illumina HiSeq allows for the largest number of samples to be multiplexed on a single run (Illumina HiSeq 2500: 300 million to 4 billion per run) [9]. Libraries are best generated by the sequencing center and samples multiplexed with Illumina indices to make downstream processing easier.

The detailed protocol used for [3] to collect clinical swabs for RNA sequencing is linked at the beginning of this chapter as an example.

3.2 QC and Filtering Reads, Generating a Counts Table

Ideally the sequencing data will be received as demultiplexed FASTQ files, meaning each sample will have its own FASTQ or two FASTQ files if paired reads are generated. Currently, Illumina data is released into BaseSpace where the user can evaluate the quality of the run. We also recommend the tool FastQC (<http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>) for a visual check of read quality and overrepresented sequences in the reads. This will help identify points to trim reads (for assembly) and whether a large number of rRNA reads are in the output data. Once reads are sufficiently trimmed and filtered by your preferred method, the next steps will be either (or a combination of) (1) mapping reads to a set of reference sequences, (2) de novo assembly of reads into transcripts, (3) assigning a functional annotation to the reads or mapped components. Finally, a “counts table” will be generated for comparative analysis using compositional data analysis (CoDa) tools.

3.3 Mapping

If at all possible, we recommend first mapping reads to a known reference. This is more computationally efficient than BLAST to a larger database, or assembling reads *de novo*, and it identifies reads that have a previously annotated function from another source. The choice of a reference database depends on the goals of the analysis, and described below are two approaches. The first approach was used in [3], but the second approach is recommended/preferred.

1. Map to predicted ORFs from reference genomes

https://github.com/ggloor/MIMB_2018/tree/master/build_clustered_database

Full or partial high-quality genomes that have been previously annotated can be merged into a reference database. This takes some consideration for the size of the database, and the overlap in sequence information between similar genes or genomes which would have to be reduced upstream of mapping, or merged after. The strategy employed below is to collect only annotated open reading frames (ORFs) and merge redundant ORFs (based on sequence identity) prior to mapping.

1. From NCBI, acquire the .ffn files for the genomes of interest to build your reference set. It's recommended you use a command line tool such as wget via the ftp link (<ftp://ftp.ncbi.nih.gov/genomes/refseq/bacteria/>).
2. Concatenate the files into one: cat *.ffn>all.ffn.
3. To remove sequence redundancy for mapping, collapse the sequences into clusters by percent identity (e.g., 95% ID) using USEARCH (<https://www.drive5.com/usearch/>).
4. Use the reference sequence for each cluster (refseq) to build a bowtie2 reference library. Use bowtie2 to map reads to the reference library.
5. Generate a counts per refseq table using HTSeq (<https://htseq.readthedocs.io>).

At this point, the mapped refseqs can be annotated functionally with whichever method/database preferred, for example, SEED (<http://www.theseed.org/>) [10, 11] or KEGG (<http://www.kegg.jp>) [12] for pathway predictions. Additionally, the clustering information from USEARCH can be used to determine ORFs unique to a particular organism, or those shared between genomes of different organisms. Note: The interpretation of “unique” ORFs depends greatly on what you choose to include in your database. For example, including one species of a particular genus will make those ORFs appear unique compared to including multiple species of the same genus.

2. Map to the SEED database of protein sequences (preferred)
https://github.com/ggloor/MIMB_2018/tree/master/diamond_to_seed

Often in meta-transcriptome analysis we are not necessarily most interested in changes of individual gene expression levels, but rather the proportion of transcripts dedicated to particular functions and pathways, of which there could be many genes performing the same function. Therefore, rather than mapping individual ORFs (as above), the preferred approach is to aggregate the reads by function, under the hypothesis that the function is the basic unit of ecosystem gene expression. A function would be one enzymatic or structural role. So for example, a function would be the sum of all reads mapping to

a single EC number or mapping to a single SEED subsystem level 4 category which define unique cellular functions [11]. We have seen that the assumption of ecosystem equivalence can be valid, since one taxon can be seen to provide the majority of the expression for a given function, even though that taxon is rare, and other taxa in the system have the ability to express that function but choose not to [3, 10].

Although this could be accomplished after the ORF mapping and then annotation ORFs to functional roles, a more direct approach is to use a database of proteins with defined roles. This circumvents two issues: (1) the reads assigned will already have a putative role and there isn't a need for a second annotation step and (2) by mapping translated nucleotide reads to protein sequences a smaller database of proteins can be mapped since protein sequences are far more conserved than nucleotide. The SEED (<http://www.theseed.org>) is a curated database of accurate genome annotations across thousands of genomes [10]. Below is the protocol for rapidly assigning FASTQ reads to a provided SEED database of reference protein sequences:

Required to run

1. UNIX or MacOS system. It's highly recommended you use a high-memory (>16Gb), high-performance server. You will also need at least 40Gb of disk space for output (this will vary depending on the amount of data you are processing).
2. DIAMOND (<https://github.com/bbuchfink/diamond>) is a super fast sequencing aligner for protein or DNA translation (nt->protein) searches.
3. The SEED database (DOI: 10.6084/m9.figshare.5836065). The database was custom curated from SEED (<http://www.theseed.org/>) to contain non-redundant protein sequences with assigned SEED subsystem functions. Briefly, each fig.peg sequence comes from an individual genome and has a subsys4 (enzymatic) functional assignment. There are many fig. pegs in a subsys4 category. From there, subsys4 are hierarchically organized into broader functional groups (subsys3, subsys2, subsys1) which are nonunique to a subsys4 (i.e., a subsys4 can belong to multiple subsys3).
4. Workflow scripts (from GitHub):
 - (a) diamond_to_seed.sh
 - (b) blast_to_counts.pl
 - (c) merge_counts.pl

Running a DIAMOND search

There is a shell script `diamond_to_seed.sh`. To run, do the following:

1. You should have a data directory (usually called “data”) containing the demultiplexed read files in `.fastq.gz` format/extension. The file names should look like: `F8G-2_S43_R1_001.fastq.gz` where everything before the file extension `.fastq.gz` will be taken as the sample name for downstream output
2. Make a working directory, e.g., `mkdir map_seed`. Copy `diamond_to_seed.sh` and the `bin` directory containing two perl scripts `blast_to_counts.pl` and `merge_counts.pl` to this directory
3. Before running `diamond_to_seed.sh` define the paths at the top of script. Example directory structure:
 - (a) `../project_name/data/reads.fastq.gz`
 - (b) `../project_name/map_seed/diamond_to_seed.sh`
 - (c) `../project_name/map_seed/bin/blast_to_counts.pl`
4. Running: `nohup ./diamond_to_seed.sh &` Note: Since this program takes a while to run, using `nohup` will push it to the background and push any output to terminal to `nohup.out`

Output:

A `diamond_output` directory will be created in the directory you run the script with the `.daa` files, the converted `.m8` files, and the total counts table `all_counts.txt` will be in your working directory. The counts table contains all samples (columns) and the associated read count per subsys4 (rows).

It is often the case that only a minority of the reads can be mapped to specific genes in the library, and that only a minority of the remainder of the reads can be included in assembled open reading frames. The reason for the incomplete annotation of reads is unknown, but is related to the large proportion of sequenced genomes that cannot be assigned by homology to known gene families. It is possible that many reads do not map simply because the reference library is incomplete, and that the remainder of the reads are obtained at a read depth that is too low for assembly. However, many of the unmapped reads cannot be assigned even upon exhaustive BLAST comparisons [1].

3.4 Compositional Data Analysis (CoDa), Overview

The analyses are conducted using a compositional data (CoDa) approach, where the ratios between abundances are examined, rather than the abundances directly [4]. High-throughput sequencing is inherently compositional [2, 13–16] since the total number of counts observed is determined by the machine and not by the actual counts observed in the underlying ecological sample [13].

Usual methods that rely on normalized counts or relative abundance for ordination and separation of multivariate sequencing datasets are driven by the most abundant features [17], while differential abundance analysis identifies as the most significantly different those that are relatively rare [14]. See ref. [13] for a full discussion of these points.

A CoDa approach allows the variance of features to be examined both during the ordination and the differential abundance phases of analysis [14, 18]. The CoDa approach offers several other important advantages. These include: that the CoDa approach is relatively scale invariant, meaning that sequence depth normalization is not required prior to data transformation [2, 4]; that the CoDa approach is sub-compositionally coherent, meaning that the analysis of the features in common is consistent between subsets of the data [4, 5, 13, 16]; and that the negative correlation bias among the features in CoDa is acknowledged [15, 19].

One complication of a CoDa approach is that the commonly used transformations cannot be performed if any of the features contain a value of 0 since the method relies on logarithmic transformations of the data. This shortcoming is not severe when conducting descriptive exploratory data analysis [13, 14], and 0 values can be imputed using the zCompositions R package [20] when performing analyses that rely on point estimates of the data.

However, univariate comparisons can be very unreliable when features with 0 values are included [2, 14]. Thus, univariate analyses are performed in a Bayesian framework where the actual table of counts is used as the prior information to estimate a posterior distribution where each instance in the distribution is consistent with the observed data [2, 21], under the assumption that a value of 0 for a feature is often derived from an underlying distribution where the 0 count arises because of under sampling [21]. Using such a probabilistic framework features with 0 counts are observed to have non-0 probabilities with wide uncertainty, while features with progressively more counts are observed to have progressively less probabilistic uncertainty in the observation of their “true” frequency.

These posterior probabilities are used as the input for CoDa-based transformations and standard statistical tests and effect-size measures. Thus, with this approach, the uncertainty associated with feature counts translates into uncertainty when determining the test-statistics, and reporting the expected value of the statistics results in a robust estimate of the “true” statistic that would have been observed with a large number of technical replications and are more consistent [22].

The use of these methods is for both exploratory multivariate and explanatory univariate approaches and are described in detail in the Subheadings 3 and 4; these approaches are found to be generally useful when analyzing a wide variety of high-throughput

sequencing experimental designs, with little or no modification from the methods described here [3, 23–26].

There are three main steps in any analysis.

1. Exploratory data analysis, where the analyst explores the overall shape of the data, looks for potential confounding factors and determines if the experiment is likely to provide insights. The CoDa approach for this is the compositional biplot which displays the linear relationships between the samples and the genes or functions on one plot [27]. This is done using point estimates of the data, and the initial transformations used here can be used for distance based clustering and partitioning methods [5, 6].
2. Differential abundance analysis whereby the investigator determines which, if any genes/transcripts or functions are different between the conditions. This is often the major goal and the approaches provided here are intended to replace others that are commonly used, but are known to have a substantial false positive problem [22, 28].
3. Correlation analysis, where the investigator determines which genes or functions may be coordinately regulated. Correlation is especially problematic in high-throughput sequencing datasets, and the vast majority of correlation analyses performed and published are “just plain wrong” [15] because they are prone to both false positive and false negative inference [19]. The approaches shown here are internally consistent and fully compliant with the nature of the data [13]. Interested readers are encouraged to explore the “propr” R package [29] which gives a thorough treatment of correlation in high-throughput sequencing datasets, and alternative use case [30].

3.5 Exploratory Data Analysis

The first analysis is exploratory data analysis using compositional PCA plots of the center-log-ratio (clr) transformed values. PCA plots show linear relationships between samples and features, and are a form of dimension reduction. The clr transformation takes the logarithm of the ratio of the count of each feature and the geometric mean count of all features in the sample [4]. That is, the clr is the log of the ratio between the abundance of a gene and the average abundance of all genes. This is similar to qPCR, except the “internal standard” is not a single housekeeping gene but the average of all genes. All subsequent interpretations of the data are thus “relative to the geometric mean abundance,” rather than count. This informs the user as to whether there is a difference between groups, and provides a qualitative overview of the structure of the data. All code given below can be obtained from https://github.com/ggloor/MIMB_2018/tree/master/CoDa_code.r, along with the example dataset. All code there is commented more fully.

Required to run. The entire script below has been developed on an early 2013 MacBook Pro with 16Gb of RAM with a 2.7GHz Intel Core i7 processor. The R script takes approximately 1 minute to run with a memory footprint under 2Gb. The workflow has been used in workshops and courses and has been run successfully on a wide variety of student laptops. The source for R packages required for the script is documented inside the file CoDa_code.r; all are publicly available and actively maintained.

1. Navigate to the directory that contains the data table generated as above. Open a terminal or R console window (*see Note 1*). Load the required R packages for the analysis, these are not part of base R and must be installed from either the CRAN (<https://www.r-project.org>) or Bioconductor. Load the following packages by typing:
 - (a) `library(ALDEx2)`.
 - (b) `library(CoDaSeq)`.
 - (c) `library(igraph)`.
 - (d) `library(car)`.
2. The entire script is in a plain text document and can be rerun simply by cutting and pasting it into the terminal window after making any desired modifications to any of the parameters (*see Note 2*).
3. Read in the data table by typing the following command
 - (a) `d <- read.table("example_data/all_counts.txt", header=T, row.names=1, check.names=F, sep="\t", comment.char="", stringsAsFactors=FALSE, quote="", na.strings = "")` (*see Note 3*).
4. It is important to ensure that features that are absent or exceedingly rare are removed prior to analysis. This can be accomplished with one of the filter functions from the CoDaSeq R package. To filter out features that have 0 counts in all samples type the following into the terminal,
 - (a) `d.f <- codaSeq.filter(d.g, min.count = 10, samples.by.row=FALSE)` (*see Note 4*).
5. The remaining 0 values in the data table need to be replaced by an estimate of the actual 0 value. This is done with the zCompositions R package [20], by typing the following into the terminal,
 - (a) `d.n0 <- cmultRepl(t(d.f), label=0, method='CZM')`
6. Convert the read counts to point estimates centered log-ratio values. This can be accomplished in CoDaSeq by typing the following into the terminal,
 - (a) `d.clr <- codaSeq.clr(d.n0, samples.by.row=TRUE)`

7. Find linear combinations of the ratios between features that best describe the data by performing a singular value decomposition of the data. This is the raw data for the PCA exploration. This can be done by typing into the terminal,
 - (a) `d.pcx <- prcomp(d.clr)` (*see Note 5*).
8. Plot the feature loadings and the sample relationships on two plots. First, we set up a list of groups. In this case samples 1-3 are in group 1, samples 4-8 are in group 2. Type the following three commands into the terminal,
 - (a) `"grps=list(c(1:3), c(4:7))"`, `"par(mfrow=c(1,2)`
 - (b) `codaSeq.PCAplot(d.pcx, plot.groups=TRUE, grp=grps, grp.col=c("blue", "red"), plot.circles=FALSE, plot.loadings=TRUE)"` (*see Note 6*).

Figure 1 shows an example PCA and loadings plot. Rules for interpreting these plots can be found in [27, 31], but essentially, samples that are close together have similar compositions, and features that are close together are found to have ratios that are nearly constant across all samples: i.e., they are positively correlated. The samples in group 1 are close together as are the samples in group 2. Thus, we can see at a glance that the two groups are very different. All features that are far apart are negatively correlated, and no information regarding negative correlation can be obtained from compositional data [15, 19].

9. It can be useful to identify samples that contribute more variance to the data than expected. This can be accomplished using the following command,
 - (a) `codaSeq.outlier(d.clr[grps[[1]],], plot.me=FALSE)`.
 - (b) substitute `grps[[2]]` for a test of the red group (*see Note 7*).

3.6 Differential Abundance Analysis Using ALDEx2

Differential relative abundance analysis provides a numerical description of the qualitative results observed in the compositional PCA plots. Essentially, this approach draws a line through the multivariate space to separate the two predefined groups, and computes statistical tests on the features that find themselves on each side of that line. All summary statistics and statistical tests are expected values of a posterior distribution of the data generated by Monte-Carlo replicates [2, 13, 14].

1. Set up the comparison groups by making a vector of offsets for the two groups. The first group is the first three samples, and the second group is the last five samples. Type the following into the terminal
 - (a) `conds <- c(rep("A", 3), rep("B", 5))`

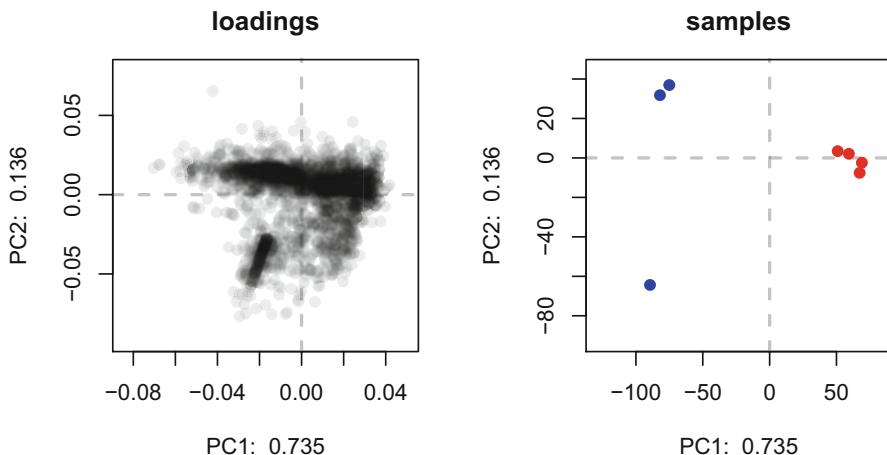


Fig 1 The codaSeq.PCAplot function plots the loadings and the principle components for the samples. The major differences between plots based on center log-ratio transformed values and numeric values is that these plots must be interpreted in the context of ratio, and not absolute, abundances [27]. The positions of the points on the “loadings” plot represent the standard deviation of the features in the dataset. Thus, their position indicates the contribution of each feature to the separation between samples. Points that are close together, in general, will be found to have near constant ratios across all the samples. Points that are distant can be negatively associated ratios or unassociated ratios. Only further exploration can determine which case is true [27]. The points running out in a straight line represent features that are 0 in one group and not 0 in the other. Points on the “samples” plot represent the multivariate Aitchison distance between points. Points are colored red or blue to indicate group membership. The group represented by the blue group can be seen to have greater variance than the group represented by the red group. Applying the test for outliers shows that the sample on the bottom left may be an outlier

2. Generate the distribution of clr values that are consistent with the read counts by typing the following into the terminal
 - (a) `x <- aldex.clr(d,g,conds)` (*see Note 8*).
3. Calculate summary descriptive statistics on the data by typing
 - (a) `x.e <- aldex.effect(x,conds)`
4. Calculate expected *p*-values and expected Benjamini-Hochberg corrected *p*-values by typing
 - (a) `x.t <- aldex.ttest(x,conds)` (*see Note 9*).
5. Group the data into one data frame
 - (a) `x.all <- data.frame(x.e,x.t,stringsAsFactors=FALSE)`
 - (b) This table contains all the summary information for the pairwise comparison (*see Note 10*).
6. Check to see if there are differences between the two groups by plotting. The most informative plot is the effect size plot [32] that can be made by typing,
 - (a) `aldex.plot(x.all)` (*see Note 11*), and example output is given in Fig. 2.

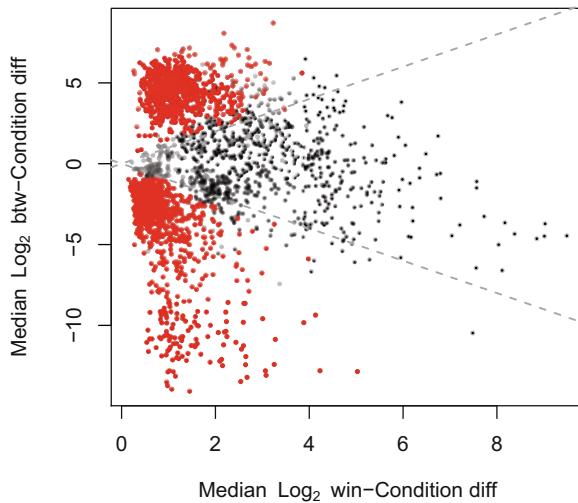


Fig 2 The `aldex.plot` function plots features with differential abundance between groups. The y -axis shows the between group difference, the x -axis plots the within group difference (dispersion). Variables are colored red if they have an expected Benjamini-Hochberg false discovery rate from a Welch's t -test less than 0.1. The dashed lines show where the difference and dispersion are equal, and so represents an effect size of 1. Effect size plots are useful to determine which features are distinguishable between conditions and why [32]. All values are \log_2 , and the alphabetically lower group name is by definition changing in the positive direction

7. The features that are significantly different, or that have large effect sizes can be determined by simple subsetting in R (*see Note 12*).

3.7 Compositional Association Using the phi-Metric

All correlation metrics are prone to false positive associations because correlated features must maintain a constant ratio in all samples [15]. Association between features in compositional data is thus very problematic because investigators do not fully understand the nature of compositional data [15, 19]. See the supplement to [13] for an extended treatment on this topic written for the non-specialist. The phi-metric is a strength of association measure that can be used to identify those features that maintain nearly constant ratios in all samples [15, 33]. The phi-metric is computed from the Monte-Carlo instances of the dataset in step 3.6.2.a above, rather than from point estimates in step 3.5.6. Expected values calculated from the Monte-Carlo instances are much more robust than are point estimates for high-dimensional data [2].

1. Calculate the expected phi values using the `CoDaSeq` function
 - (a) `x.phi <- codaSeq.phi(x)` (*see Note 13*).
2. The returned data can be visualized using graphical analysis software, but for a quick analysis of which features are strongly

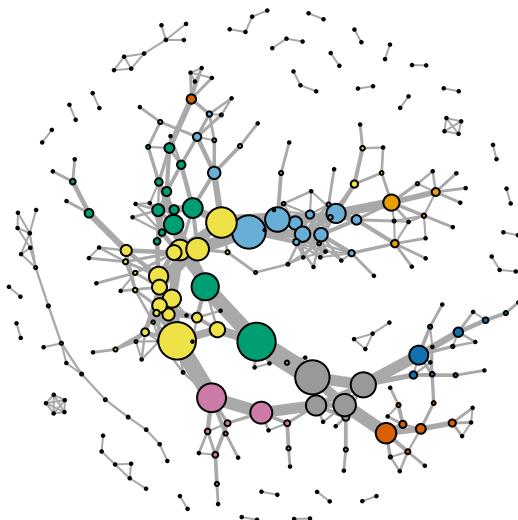


Fig 3 The codaSeq.phi function takes as input the output of the aldex.clr function. The results can be deconstructed and visualized using igraph (<http://igraph.org/r/>) or other graphical analysis software that the user is familiar with. Functions displaying strong associations can be plotted per group. This graph shows nodes sized by their node-centrality property, with edges sized by edge-centrality

associated, a dataframe can be produced with the feature names and cluster groups. Type the following commands sequentially into the terminal to generate a set of features connected by low phi values (highly correlated)

- (a) `x.lo.phi <- subset(x.phi, phi <=0.002)", "g <- graph.data.frame(x.lo.phi, directed=FALSE)", "g.clust <- clusters(g).`
- (b) `g.df.u <- data.frame(Name=V(g)$name, cluster=g.clust $membership, stringsAsFactors=FALSE).`
- (c) `g.df <- g.df.u[order(g.df.u[, "cluster"]),]` (*see Note 14*).

3. The graph can be broken down into individual connected sets as follows,

- (a) `dg <- decompose.graph(g).`

and individual sets can be plotted by typing,

- (b) `plot(dg[[1]]).`

- (c) where the numerical value reflects the individual graph. Figure 3 shows the result of plotting the betweenness, and the code is given in the supplementary script since it is rather involved (*see Note 15*).

3.8 Plotting Differences at a Functional Level

It is generally useful to observe what functional groups, or pathways are differentially abundant between the two groups. One simple way to do this is to plot the differences or effect sizes for each function when grouped to a pathway or more general function [3]. The starting points for this plot are the x.all dataframe from ALDEx2 and a reference lookup table containing the mappings between different KEGG or SEED hierarchies, subsystems2roleNA.txt, that is supplied in the example_data folder. For example, the first hierarchy entry in order from SEED 1 to SEED 4 is: Stress Response : Oxidative Stress : Glutathione:_Biosynthesis_and_-gamma-glutamyl_cycle : 5-oxoprolinase (EC 3.5.2.9)

1. Function level stripcharts are implemented in the CoDaSeq package. These graphs show the SEED functions for grouped into each higher SEED category. For example, there are many individual functions that are involved in potassium metabolism and we can see that some are relatively more abundant in each group. The starting point for the plot is the data from Sub-heading 3.6.5.a, together with a mapping file that contains the hierarchical information for each lower level SEED category. This is supplied in the example_data folder. The plot can be generated using the following commands
 - (a) `e<-read.table("example_data/subsystems2roleNA.txt", sep="\t", comment.char=" ", stringsAsFactors=FALSE, quote="")`
 - (b) `colnames(e) <- c("SEED3","SEED1","SEED2", "SEED4")`
 - (c) `codaSeq.stripchart(aldex.out=x.all, group.table=e, group.label="SEED1", heir=TRUE, heir.base="-SEED4", mar=c(4,22,4,0.5), x.axis="diff.btw", sig.method="we.eBH", sig.cutoff=.05)`
 - (d) (*see Note 16*). Figure 4 shows example output for SEED level 4 functions plotted by SEED level 1 groups, the most general functional grouping. In this plot each SEED level 4 function is plotted in the appropriate SEED1 grouping. Note that the given SEED4 function may appear in more than one SEED1 group.

4 Notes

1. A basic ability to manipulate data tables using R is assumed. Useful resources can be found at: https://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf.
2. The user is encouraged to examine the R commands in the CoDa_code.r file which should be opened in a plain text editor. This contains all commands used for the data analysis. All

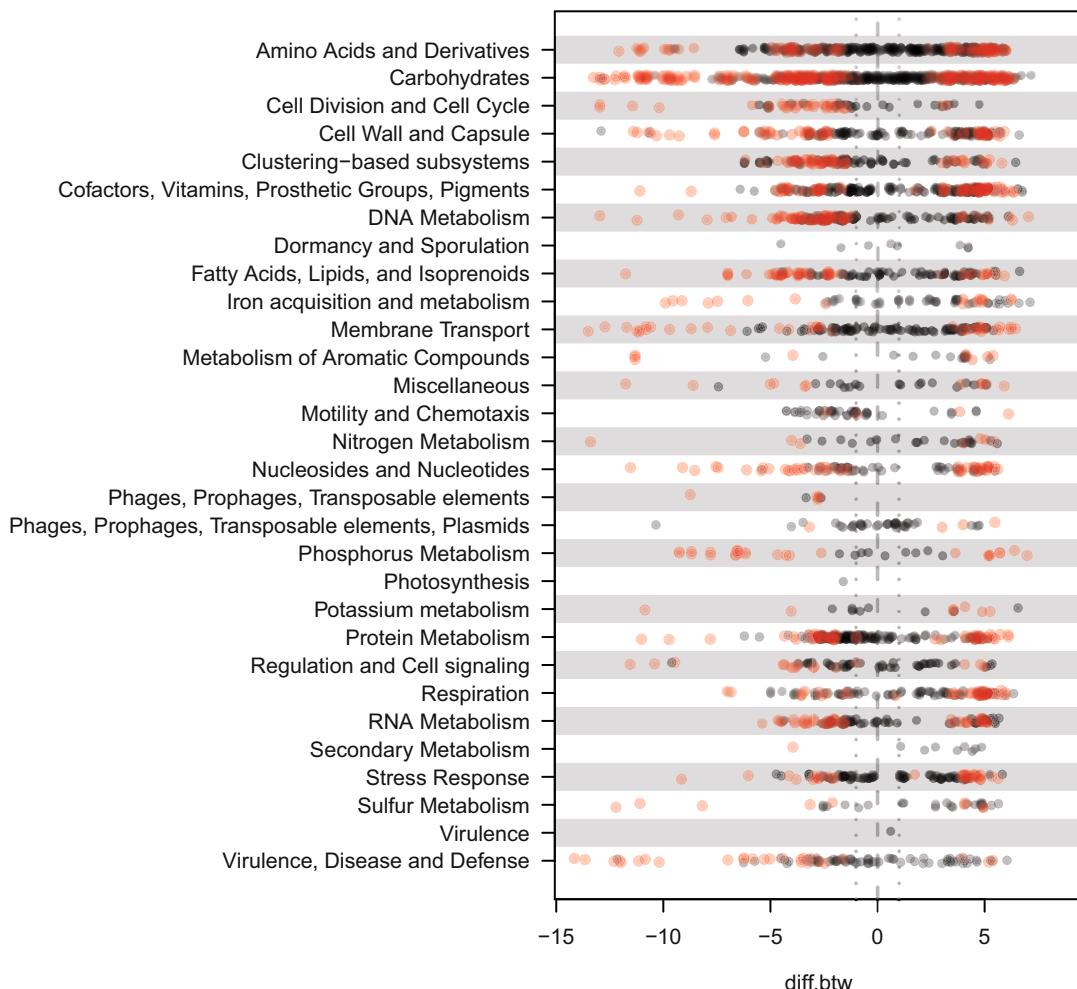


Fig 4 The codaSeq.stripchart function plots the lowest level features by their higher level groupings. In the example here the lowest level function, SEED 4, is plotted in groups of the most general SEED functional level, SEED 1 with the between group difference on the x-axis. Points are colored red if they have a Benjamini-Hochberg adjusted p -value less than 0.01, and gray otherwise. Any of the columns in the aldex output file can be plotted on the x-axis. The plot is useful to observe at a glance which functional groups change in general, and which functional groups contain large differences in both directions. So for example above, the SEED 4 functions for Virulence, Disease and Defence are more abundant in Group A than Group B

characters must be plain ASCII text, and the use of a text editor such as Atom (<https://atom.io>) or TextWrangler or BBEdit for Mac, Notepad++ for Windows, or RStudio for either.

3. It is important that the user understands the basic file system on their computer, that the user keep all the files together in one directory, and that the user maintains a structured approach. We find it useful to keep files in sub-directories of each project, where the sub-directories contain: raw and intermediate data, figures, Perl, bash or R code. For the purposes of

this analysis these directories are called: data, fig, and chunk. The input table is a table of mapped reads per feature with the first column being the feature name. For example, the first feature in the table is “(3R)-hydroxymyristoyl-[acyl carrier protein] dehydratase (EC 4.2.1.-),” the subsequent columns being the read counts per sample. The column names are the sample IDs.

4. There are many other filtering options, and the user is encouraged to try several of them to ensure that the results are not being overly influenced by rare, abundant or low variance features. See [23] for an example of the robustness of the CoDa approach.
5. It is important to use the “prcomp” function. The “princomp” function in R is deprecated, and should not be used.
6. The grps list contains the indices of the samples in each group. So in this instance the first group is columns 1–3 of the input variable (d.g), and the second group is columns 4–7 of the variable. If only the sample relationships are desired to be shown, then set the mfrow parameter to c(1,1), and plot.loadings=FALSE. An alternative approach is to use the base R biplot command, or the colored biplot command from the compositions R package. The axes are labeled according to the percent variation explained by the features. In the example shown, we can see a clear split between the red and blue groups. We can further observe that this split is driven by a small number of features on the left-hand side of the loadings plot, and a larger number of features on the right-hand side of the loadings plot.
7. The outlier method is a modification of the one in (35), which identifies samples that contribute excessive variance to the datasets. It must be applied per group.
8. The number of random instances of the data can be controlled with the mc.samples parameter. By default it is set to 128. For more precision with small sample sets (under 5 per group), set this to at least 1000. When dealing with large sample sets (over 100 per group) this parameter can be as low as 16. In addition, when dealing with asymmetric datasets as is often found in meta-transcriptomic and metagenomic datasets, the denominator used for the clr value can be set manually. See the ALDEx2 vignette released with the R package for examples.
9. All values reported by ALDEx2 are expected values of the number of Monte-Carlo replicates performed in **step 2**. Thus, the expected *p*-value for no effect is 0.5 (the expected value of a random uniform distribution).
10. All abundance values are log2 and are expected values relative to the geometric mean abundance of the feature. The

information in the table is as follows: rab.all—the abundance of the feature relative to the geometric mean abundance; rab.win.A/B—the abundance of the feature in set A relative to the geometric mean abundance in set A or set B; diff.btw—the difference in abundance between set A and B for the feature; diff.win—the maximum dispersion of the feature in set A or B (negative values indicate that the A feature is more abundant than the B feature); effect—the expected value of the ratio between diff.btw and diff.win; overlap—the overlap between the distributions of the A and B feature abundances; we/wi.ep—the expected *p*-value from Welch's *t*-test or Wilcoxon rank test; we/wi.eBH—the expected false discovery rate for the two tests.

11. The effect plot colors in red those features that are significantly different at the chosen false discovery rate derived from a Welch's *t*-test, and the features that are outside the diagonal dashed lines are those that have an expected effect size greater than 1 [32]. ALDEEx2 can also plot an MA plot with the command, "aldex.plot(x.all, type="MA")."
12. When examining transcriptomes or meta-transcriptomes an effect size of 1.5–2 is often useful, since effect sizes are much more reproducible than *p*-values [34]. To subset on effect size of >2 type, "effect2 <- x.all[which(abs(x.all\$effect) > 2),]" for all descriptive statistics, or type "effect2 <- rownames(x.all)[which(abs(x.all\$effect) > 2)]" to extract only the names of the features with at least that absolute effect size. Similar subsetting can be used to identify *p*-values or false discovery rates below a chosen threshold by changing the column name and comparison that the subset is performed on.
13. The phi returns the lower triangle of the expected values of a symmetrical phi function [15].
14. There are a number of excellent introductions to graph manipulation with R. See <http://igraph.org/r/doc/>, or <http://www.shizukalab.com/toolkits/sna/igraph-vs-statnet>, or <http://michael.hahsler.net/SMU/LearnROnYourOwn/code/igraph.html>, or <https://users.dimi.uniud.it/~massimo.franceschet/R/communities.html>.
15. The approach demonstrated has been superseded by the release of the propo R package on CRAN [29] <https://cran.r-project.org/web/packages/propo/index.html>, which incorporates this ad hoc method into a fully functional R package. There is a comprehensive tutorial, and worked example vignettes for gene expression analysis. Readers are encouraged to use this package when possible.
16. Grouped stripcharts can be made for both one to many mapping hierarchies (such as the SEED hierarchy [10]), or

one to one mapping hierarchies (such as a taxonomic mapping). The user indicates a one-to-many grouping by the heir=TRUE parameter, and must supply the column that is the base of the hierarchy with the heir.base parameter, and the name of the hierarchical group with the group.label parameter. The format of the input data table for a one-to-many mapping must include one column that contains the exact same feature names as is found in the input count table (the hier.base column), and one or more columns with the group that the feature belongs to (the group.label column). Duplicate feature names are permitted if the feature belongs to more than one group. Files used as input can be in the same format and can be called with the same flags, or can be in an alternate format where the rownames correspond to the feature names in the count tables. In this case, only the group.label flag should be given.

References

1. Jiang Y, Xiong X, Danska J, Parkinson J (2016) Metatranscriptomic analysis of diverse microbial communities reveals core metabolic pathways and microbiome-specific functionality. *Microbiome* 4:2. <https://doi.org/10.1186/s40168-015-0146-x>
2. Fernandes AD, Macklaim JM, Linn TG, Reid G, Gloor GB (2013) ANOVA-like differential expression (aldex) analysis for mixed population rna-seq. *PLoS One* 8:e67019. <https://doi.org/10.1371/journal.pone.0067019>
3. Macklaim MJ, Fernandes DA, Di Bella MJ, Hammond J-A, Reid G, Gloor GB (2013) Comparative meta-RNA-seq of the vaginal microbiota and differential expression by Lactobacillus iners in health and dysbiosis. *Microbiome* 1:15. doi: <https://doi.org/10.1186/2049-2618-1-12>
4. Aitchison J (1986) The statistical analysis of compositional data. Chapman & Hall, London, England
5. van den Boogaart KG, Tolosana-Delgado R (2008) “Compositions”: a unified R package to analyze compositional data. *Comput Geosci* 34:320–338. <https://doi.org/10.1016/j.cageo.2006.11.017>
6. Pawlowsky-Glahn V, Egozcue JJ, Tolosana-Delgado R (2015) Modeling and analysis of compositional data. John Wiley & Sons
7. Bernstein JA, Khodursky AB, Lin P-H, Lin-Chao S, Cohen SN (2002) Global analysis of mRNA decay and abundance in escherichia coli at single-gene resolution using two-color fluorescent dna microarrays. *Proc Natl Acad Sci* 99:9697–9702
8. Macklaim JM, Gloor GB, Anukam KC, Cribby S, Reid G (2011) At the crossroads of vaginal health and disease, the genome sequence of Lactobacillus iners AB-1. *Proc Natl Acad Sci U S A* 108(Suppl 1):4688–4695. <https://doi.org/10.1073/pnas.1000086108>
9. Besser J, Carleton HA, Gerner-Smidt P, Lindsey RL, Trees E (2017) Next-generation sequencing technologies and their application to the study and control of bacterial infections. *Clin Microbiol Infect*. <https://doi.org/10.1016/j.cmi.2017.10.013>
10. Overbeek R, Olson R, Pusch GD, Olsen GJ, Davis JJ, Disz T, Edwards RA, Gerdes S, Parrello B, Shukla M, Vonstein V, Wattam AR, Xia F, Stevens R (2014) The seed and the rapid annotation of microbial genomes using subsystems technology (rast). *Nucleic Acids Res* 42:D206–D214. <https://doi.org/10.1093/nar/gkt1226>
11. Mitra S, Rupek P, Richter DC, Urich T, Gilbert JA, Meyer F, Wilke A, Huson DH (2011) Functional analysis of metagenomes and metatranscriptomes using SEED and KEGG. *BMC Bioinform* 12 Suppl 1:S21. <https://doi.org/10.1186/1471-2105-12-S1-S21>
12. Kanehisa M, Goto S, Furumichi M, Tanabe M, Hirakawa M (2010) KEGG for representation and analysis of molecular networks involving diseases and drugs. *Nucleic Acids Res* 38: D355–D360. <https://doi.org/10.1093/nar/gkp896>

13. Gloor GB, Macklaim JM, Pawlowsky-Glahn V, Egozcue JJ (2017) Microbiome datasets are compositional: and this is not optional. *Front Microbiol* 8:2224. <https://doi.org/10.3389/fmicb.2017.02224>
14. Gloor GB, Macklaim JM, Vu M, Fernandes AD (2016) Compositional uncertainty should not be ignored in high-throughput sequencing data analysis. *Aust J Stat* 45:73–87. <https://doi.org/10.17713/ajs.v45i4.122>
15. Lovell D, Pawlowsky-Glahn V, Egozcue JJ, Marguerat S, Bähler J (2015) Proportionality: a valid alternative to correlation for relative data. *PLoS Comput Biol* 11:e1004075. <https://doi.org/10.1371/journal.pcbi.1004075>
16. Quinn TP, Erb I, Richardson MF, Crowley TM (2017) Understanding sequencing data as compositions: an outlook and review. *bioRxiv*. <https://doi.org/10.1101/206425>
17. Aitchison J (1983) Principal component analysis of compositional data. *Biometrika* 70:57–65
18. Egozcue JJ, Pawlowsky-Glahn V, Gloor GB (2018) Linear association in compositional data analysis. *Aust J Stat* 47:3–31
19. Palarea-Albaladejo J, Martín-Fernández JA (2015) ZCompositions—R package for multivariate imputation of left-censored data under a compositional approach. *Chemom Intel Lab Syst* 143, 85–96. <https://doi.org/10.1016/j.chemolab.2015.02.019>
20. Jaynes ET, Bretthorst GL (2003) Probability theory: the logic of science. Cambridge University Press, Cambridge
21. Thorsen J, Brejnrod A, Mortensen M, Rasmussen MA, Stokholm J, Al-Soud WA, Sørensen S, Bisgaard H, Waage J (2016) Large-scale benchmarking reveals false discoveries and count transformation sensitivity in 16S rRNA gene amplicon data analysis methods used in microbiome studies. *Microbiome* 4:62. <https://doi.org/10.1186/s40168-016-0208-8>
22. Bian G, Gloor GB, Gong A, Jia C, Zhang W, Hu J, Zhang H, Zhang Y, Zhou Z, Zhang J, Burton JP, Reid G, Xiao Y, Zeng Q, Yang K, Li J The gut microbiota of healthy aged Chinese is similar to that of the healthy young. *mSphere* 2:e00327–e00317. <https://doi.org/10.1128/mSphere.00327-17>
23. Goneau LW, Hannan TJ, MacPhee RA, Schwartz DJ, Macklaim JM, Gloor GB, Razvi H, Reid G, Hultgren SJ, Burton JP (2015) Subinhibitory antibiotic therapy alters recurrent urinary tract infection pathogenesis through modulation of bacterial virulence and host immunity. *MBio* 6. <https://doi.org/10.1128/mBio.00356-15>
24. McMillan A, Rulisa S, Sumarah M, Macklaim JM, Renaud J, Bisanz JE, Gloor GB, Reid G (2015) A multi-platform metabolomics approach identifies highly specific biomarkers of bacterial diversity in the vagina of pregnant and non-pregnant women. *Sci Rep* 5:14174. <https://doi.org/10.1038/srep14174>
25. McMurrough TA, Dickson RJ, Thibert SMF, Gloor GB, Edgell DR (2014) Control of catalytic efficiency by a coevolving network of catalytic and noncatalytic residues. *Proc Natl Acad Sci U S A* 111:E2376–E2383. <https://doi.org/10.1073/pnas.1322352111>
26. Aitchison J, Greenacre M (2002) Biplots of compositional data. *J Royal Stat Soc Ser C (Appl Stat)* 51:375–392
27. Hawinkel S, Mattiello F, Bijnens L, Thas O (2017) A broken promise: microbiome differential abundance methods do not control the false discovery rate. *Brief Bioinform* bbx104
28. Quinn T, Richardson MF, Lovell D, Crowley T (2017) Propr: an R-package for identifying proportionally abundant features using compositional data analysis. *bioRxiv*. <https://doi.org/10.1101/104935>
29. Erb I, Quinn T, Lovell D, Notredame C (2017) Differential proportionality—a normalization-free approach to differential gene expression. *bioRxiv*. <https://doi.org/10.1101/134536>
30. Gloor GB, Reid G (2016) Compositional analysis: A valid approach to analyze microbiome high-throughput sequencing data. *Can J Microbiol* 62:692–703. <https://doi.org/10.1139/cjm-2015-0821>
31. Gloor GB, Macklaim JM, Fernandes AD (2016) Displaying variation in large datasets: Plotting a visual summary of effect sizes. *J Comput Graph Stat* 25:971–979. <https://doi.org/10.1080/10618600.2015.1131161>
32. Erb I, Notredame C (2016) How should we measure proportionality on relative gene expression data? *Theory Biosci* 135:21–36
33. Gierliński M, Cole C, Schofield P, Schurch NJ, Sherstnev A, Singh V, Wrobel N, Gharbi K, Simpson G, Owen-Hughes T, Blaxter M, Barton GJ (2015) Statistical models for rna-seq data derived from a two-condition 48-replicate experiment. *Bioinformatics* 31:3625–3630. <https://doi.org/10.1093/bioinformatics/btv425>
34. Halsey LG, Curran-Everett D, Vowler SL, Drummond GB (2015) The fickle p value generates irreproducible results. *Nat Methods* 12:179–185. <https://doi.org/10.1038/nmeth.3288>



Chapter 14

Subsampled Assemblies and Hybrid Nucleotide Composition/Differential Coverage Binning for Genome-Resolved Metagenomics

Laura A. Hug

Abstract

Metagenomic analyses for reconstruction of genomes from mixed microbial community datasets now routinely allow rapid, accurate genome recovery for tens to hundreds of organisms from environmental samples. This chapter provides a step-by-step protocol for reconstructing genomes from metagenomic datasets, with a focus on the most abundant community members. Subsampling assembly approaches are implemented to improve assembly of abundant genome sequences, an iterative process that targets progressively less abundant populations and improves total community representation in the final merged assembly. A hybrid approach to genome binning is described, combining differential coverage information from a series of metagenomic samples with nucleotide composition information. This approach strengthens binning through application of multiple independent variables for contig clustering. Genome curation through error correction and gap closure leads to high-quality draft genomes, and, for some community members, closed and complete genome sequences reconstructed directly from environmental samples.

Key words Metagenomics, Binning, Microbial communities, Candidate phyla, Nucleotide composition, Abundance profiles

1 Introduction

Advances in sequencing technologies and bioinformatic methods for data handling have allowed access to microbial communities from a wide variety of environments and hosts, providing culture-independent information on the taxonomic composition and metabolic potential of these communities. Metagenomics, where total microbial DNA is shotgun sequenced, is a powerful tool for examining microbial communities, and is a popular method to access genomic information for organisms that have resisted cultivation efforts. While initial metagenomic surveys focused on environments hosting a low-diversity microbial community to reduce the complexity of the analysis [1, 2], current techniques allow in-depth genome reconstruction for low-abundance, high-diversity

microbial communities from environments such as soil and sediment [3–6]. Where a metagenomic dataset provides a blueprint or overview of the total microbial diversity and predicted functions in an environment, the additional analysis step of binning the recovered genomic fragments into draft or closed complete genomes [7, 8] allows identification of the specific contributions of individual members to the community, including prediction of interactions with other community members, roles in biogeochemical cycling, and association of markers with specific populations [5, 6, 9, 10].

The complexity of the microbial communities assayed can lead to technical difficulties. Closely related strains can confound assembly algorithms, lowering recovery of these organisms' genomes [10]. Highly abundant organisms' genomes often assemble poorly, even when using assembly algorithms designed to accommodate the variation in sequence abundance inherent in mixed microbial community DNA [11]. Long-read technology such as PacBio and Moleculo can address these issues [11, 12], but are not yet routinely used for metagenomic datasets due to considerations of high error rates and cost, respectively.

This chapter will describe a method and relevant variations for genome-resolved metagenomics, targeting the most abundant organisms within a dataset and leading to high-quality draft genomes. The method includes suggested programs and scripts, but with the rapid increase in metagenomic datasets and accompanying analysis software packages, there are many alternative programs that can be substituted for those discussed here (alternatives are, where possible, identified in the notes for each step). Major procedures are divided into three subheadings: Subheading 3.1—subsample-based iterative metagenomic assembly; Subheading 3.2—hybrid differential coverage and nucleotide composition binning; and Subheading 3.3—genome curation through error correction and gap closure. The three subheadings are presented as modular units of a combined workflow, but can be used independently of each other with no conflicts.

2 Materials

The following methods require a paired-end metagenomic sequencing dataset. All methods have been validated solely with Illumina sequence datasets.

Subheading 3.2: “Hybrid differential coverage and nucleotide composition binning” requires, at minimum, two metagenomic datasets from a time series, depth series, or other kind of connected sampling (*see Note 1*).

No specific materials are necessary other than sufficient computational resources to conduct the described analyses. Alternative software options are provided in the notes where possible.

3 Methods

DNA extraction from environmental samples is a highly tailored process (*see Note 2*). The following methods begin at the stage of completed metagenomic sequencing and assume Illumina sequencing datasets, but can be adapted for alternative sequencing technologies.

3.1 Subsampling Assemblies

This section works with a single metagenomic sequence dataset. The following protocol was implemented in [5].

1. From fastq formatted file(s), quality trim reads using Sickle [13] (*see Note 3*).
2. If not already interleaved, interleave the trimmed paired-end data files (singleton reads will be ignored from here on in). There are a number of scripts available for interleaving. “Interleave-fastq.py” from the developers of the Ray assembler [14, 15] is an easy one to implement (<https://sourceforge.net/p/denovoassembler/ray-testsuite/ci/master/tree/scripts/interleave-fastq.py>), as is fq2fa in the idba package, which will merge paired files and convert to fasta at the same time (useful if later running idba_ud as an assembler) ([16], <https://github.com/loneknightpy/idba>).
3. Begin iterative subsampling. Identify the minimum amount of information that will still lead to some assembly (contigs > 10,000 base pairs (bp), at least some contig coverages of 10–20×). The number of reads required for this first step will vary with depth of sequencing and community complexity. If community composition information is available, estimate the proportion of the metagenomic reads required to bring the most abundant organism’s coverage to 20× (*see Note 4* for sample calculation). If no community composition information is available, start by targeting either 5% or 10% of the total data depending on expected community complexity (enriched or dominated community = 5%, highly complex/low abundance = 10%).
4. Pull reads from interleaved metagenome reads file to the desired percentage.

If total number of reads in the trimmed interleaved dataset is not known, use the UNIX command wc to get line counts:

```
$ wc -l sequencefile.fastq
```

If fastq format, divide by four to get total read number. If fasta format, divide by two to get total read number. Determine the number of reads required for the current percentage.

These reads can be randomly sampled or, given that sequence data is output in a random order, simply taken as a segment of the total file using unix head/tail commands:

```
$ head -n [line number of start position] sequencefile.fastq | tail -n [line number of end position] > outputfile_sizefraction.fastq
```

If pulling reads from a fastq file, ensure the total lines selected as well as the head position in the file (an optional burnin) are both divisible by eight (each read uses four lines, this ensures only complete pairs are used). If pulling reads from a fasta file, ensure the total lines and burnin values are both divisible by four (each read uses two lines). The burnin is optional—taking the first n reads should be just as random a sequence distribution as taking the second n reads and so on.

5. Assemble read subset. Idba_ud is a relatively fast, accurate assembler that takes into account the neven depth (“ud”) found in mixed community metagenomes [16]. Idba_ud requires interleaved fasta file format, which can be converted from fastq using the fq2fa.py script included in the idba package (<https://github.com/loneknightpy/idba>) (see Note 5 for alternative assembler options).
6. Map subsampled reads to the assembly. Bowtie2 [17] is fast and efficient, and can be coupled with shrinksam (<https://github.com/bcthomas/shrinksam>) to reduce output file sizes considerably. Determine contig coverage for each contig in the assembly (BEDtools is an excellent option for this calculation [18]). (see Note 6 for alternatives for read mapping and for links to tutorials for Bowtie2 and BEDtools.)
7. Assess the coverage statistics to determine if initial subsampling level is appropriate for the first subsample. If there are numerous scaffolds above 50× coverage, consider trying a smaller subsample (e.g., 2% when the original subsample was 5% of the reads). If there are no or very few scaffolds with coverage at or above 20× coverage, consider discarding this subsample and moving instead to a larger subsample (e.g., up to 10% from 5%). Coverage values between 15 and 35× are an ideal level for assembly.
8. Iterate through steps 3–5, selecting a larger subsample each time (e.g., 5%, 10%, 25%, 33%, 66%, and 100%). This can be done with or without replacement of reads. For simplicity, with replacement is favored (see Note 7 for sampling without replacement).

9. Merge all assemblies. Minimus2 is an excellent program for this [19]. Increase overlap size (-D OVERLAP=n; default is 42 bp, recommendation for assembled scaffolds is 200+ bp). Increase minimum percent identity for overlap (-D MINID=n; default is 94, recommendation for assembled scaffolds is 97+, approximately equivalent to strain level differentiation).

3.2 Differential Coverage Binning

This section requires multiple metagenomic sequence datasets from a time-series, depth transect, or similar set of related samples.

1. Use Bowtie2 or equivalent fast read mapper to map reads from each metagenomic dataset to the assembly of interest (*see Subheading 3.1, step 6*). Save .sam files in a single directory.
2. Bin the assembled scaffolds, or a subset of abundant scaffolds (*see Note 8*) based on the coverage and nucleotide composition table(s). There are a number of reasonable binning algorithms available, and more being developed each year. Emergent self-organizing maps (ESOM, [3, 7]), ABAWACA (<https://github.com/CK7/abawaca>), CONCOCT [20], MetaBat [21], and Anvi'o [22] are all popular programs able to incorporate coverage and nucleotide composition information. Kang *et al.* provide a reasonable comparison of these methods [23].
- (a) Generate a table of differential coverage values for each scaffold across each metagenome. This table can also include nucleotide composition information, a recommended additional level of information for binning.

For ESOM, ABAWACA, CONCOCT:

prepare_esom_files.pl (https://github.com/CK7/esom/blob/master/prepare_esom_files.pl) is an open-source option for generating this table. To apply multiple sam files, use the -sa flag, providing an id and the SAM file name for each dataset (e.g., -sa data1 data1.sam -sa data2 data2.sam etc.). Use the -k flag to set nucleotide composition, where -k 0 means no nucleotide composition, -k 1 means mononucleotide, -k 2 dinucleotide frequencies, and so on (*see Notes 9–11*).

For MetaBat and Anvi'o:

The programs will automatically generate this table from BAM files. To generate BAM files from Bowtie2 SAM files using samtools:

```
samtools view -bS test.sam > test.bam
```

There is an excellent tutorial for the Anvi'o workflow here: <http://merenlab.org/2016/06/22/anvio-tutorial-v2/>

- (b) Bin genomes according to the binning program of choice. For ESOM, following map construction, this is a manual process of selecting regions of the map to identify grouped contigs. For ABAWACA, CONCOCT, MetaBat, and Anvi'o, bins are automatically generated and output.
3. Examine bin quality and completeness.
- (a) CheckM [24] is a robust tool for this step, identifying single-copy gene presence/absence and duplications to determine statistics for each genome bin.
 - (b) Additional metrics that can be examined include simple GC plots to identify outlier contigs and/or examination of the phylogenetic affiliation of the genes within the genome bin for consistent signatures (*see Note 12*).
- If satisfied with the bins, move to Subheading 3.3, **step 1**. If not, or if certain bins appear problematic, continue to **step 4** in this section (*see Note 13*).
4. Rebinning can address certain issues in initial binning outputs, including resolving “megabins” that are composed of multiple genomes (identified by high levels of contamination in CheckM output). If a large number of bins do not pass quality standards, binning can be redone using a different binning program or a different balance of nucleotide composition to series abundance data (**steps 2 and 3**) (*see Note 9*). Alternatively, specific bins containing multiple genomes can be rebinned separately from the larger dataset, reducing computational complexity and enhancing resolution between organism abundance and genome signature patterns. In this case, either generate a subset table of the relevant contigs’ information (abundance, nucleotide composition), or remap the metagenomic reads to the contents of the megabin prior to binning the smaller dataset using the binning program of choice. Repeat **step 3**, bin quality and completion, for all new bins.

An alternative currently in beta development but showing promise is RefineM from the developers of CheckM (hosted on github: <https://github.com/dparks1134/RefineM>), which identifies partial bins that should be merged, contigs that should be included in bins, divergent contigs that should be removed from bins, and eukaryotic contamination.

This process can be iterative, and largely directed by the quality of the data, complexity of the community surveyed, and specific research questions. An end point for binning will vary depending on the project goals.

3.3 Genome Curation Through Single-Genome Assembly, Error Correction, and Gap Closure

This section works with genome bins rather than complete metagenomic datasets. Each step can be applied separately, in isolation or in combination. If in combination, follow the order suggested here.

1. Genome refinement through single-genome assembly.
 - (a) Generate a sequence read file containing all reads and their pairs for a given genome bin. Map metagenomic reads to a genome bin of interest, or subset the Bowtie2 results from initial contig mapping (Subheading 3.2, step 1) to identify the relevant subset from the assembly. When pulling sequence reads, pull both forward and reverse reads, regardless of whether each of the pair mapped (use pull-seq, *see Note 7*). This will aid in scaffold extension compared to the original assembly.
 - (b) Reassemble the single-genome sequence read file. As this represents a single genome, an isolate-optimized assembler such as Velvet [25] is appropriate.
 2. Genome refinement through error correction and gap closure is available through several genome-finishing scripts and programs. Approaches that are specific to metagenome-derived genomes include a scaffold curation script ra2.py (maintained on github at https://github.com/christophertbrown/fix_assembly), which conducts simultaneous error correction and scaffold internal gap closure through read-mapping based metrics.
- Alternatively, FinishM, an alpha version of a genome finisher from metagenomic data, is available on github at <https://github.com/wwood/finishm>. The absence of a curated, published pipeline for genome curation and finishing is a recognized need.

4 Notes

1. Ideally, the metagenomic sequencing dataset will contain six or more samples distributed across a time series, depth series, or alternative type of gradient within your experimental system. The larger the number of independent metagenomes with some overlapping community members, the stronger the signal for binning.
2. We recommend reading Lever et al. [26] for guidelines in nucleic acid extractions from a wide variety of environments.
3. Sickle has excellent documentation on github (<https://github.com/ucdavis-bioinformatics/sickle/blob/master/README.md>). There are options for trimming interleaved or separate paired-end files. Note also, if Illumina reads were quality scored

using CASAVA 1.8 or later, counterintuitively, the read type flag is `-t sanger`, not `-t illumina`. This is because the scoring ranges are larger in Sanger quality scoring, and the newer Illumina scoring shares that range.

4. For example, take a community with a most abundant member at 30% of the total, and a metagenome that has been sequenced with 25,000,000 150 bp reads (a typical Illumina MiSeq run). Assuming an average genome size of 3 Mbp (adjust as needed if genome size is known), then expected coverage is:

$$(\# \text{ reads} \times \text{read length}) \times \text{proportion of community/genome size} = (25,000,000 \times 150) \times 0.3/3,000,000 = 375\times.$$

To get to 20 \times requires division by 18.75, which is 5.33% of the data (100/18.75). In this scenario, an initial subsampling of 5% of the sequence reads will lower the expected coverage of the dominant organism to approximately 20 \times .

5. Other popular metagenomic sequence assemblers include MegaHit [27, 28], Meta-Velvet [29], and Ray Meta [14]; the list continues to grow. For the purposes of this protocol, the assembler chosen does not matter provided sequences are being assembled—testing a few to identify the one with best results on the dataset may be a worthwhile use of time. Here is a list of current metagenomic assembly options: <https://omictools.com/metagenomic-assembly-category>.
6. There are many short-read alignment programs, including the widely used Bowtie2, BWA ([30], <http://bio-bwa.sourceforge.net/>), and BBmap (<https://sourceforge.net/projects/bbmap/>) software packages. Shrinksam was written for Bowtie2 output and may not be compatible with other short-read aligners. An excellent step-by-step tutorial on running a Bowtie2 read mapping coupled to determining assembly coverage statistics can be found here: http://metagenomics-workshop.readthedocs.io/en/latest/functional-annotation/read_mapping.html.
7. There may be an advantage to without-replacement methods in datasets with highly dominant organisms (e.g., where a 5% subsample does not lower coverage values below 50 \times). In this case, removing reads that map to the assembled subsample may significantly reduce the size of the remaining metagenome dataset and subsequent computational complexity. To conduct without-replacement iterative subsampling, map all reads from the dataset to the current subsampled assembly using Bowtie2. Remove mapping reads and their pairs from the dataset prior to

taking the next subsample. This can be done using awk to identify reads mapping to the assembly:

```
$ awk '$3 != "*" readmapped.sam > hits.txt
```

followed by pullseq (<https://github.com/bcthomas/pullseq>) to exclude these reads from the original read dataset. With pullseq, always direct output to a new file, to keep track of iterations. Then go back to Subheading 3.1, step 3 and take the next subsample, but from the reduced read file generated here.

8. If the microbial community is highly complex, or there is specific interest in the most abundant organisms from the assembled sample, consider simplifying the dataset by binning only the most abundant scaffolds above a threshold (e.g., 20× coverage using the complete dataset) to reduce the complexity of the dataset.
9. Combining nucleotide and coverage information can add significant power to binning, but the two data types need to be balanced. If the metagenomic series data is from eight samples, those will contribute eight columns to the data table used for binning. Combining this with tetranucleotide information (256 columns) will swamp the coverage signal in the downstream clustering analyses, where each column is typically weighted the same. Better in this case to use dinucleotide (16 columns) composition information. It is often useful to generate several tables combining coverage information with different nucleotide-composition frequency calculations, and testing each to see which provides the strongest signal for binning. It is difficult to know *a priori* which component of your dataset will provide the strongest information.
10. Normalization of the coverage information can be applied at this step. Normalization is recommended if the series datasets are different sizes—log transformation is the normalization available at this step (-log-transform), which normalizes across rows (abundance information across a scaffold, rather than within a sample). Alternatively, within-sample proportional normalization can be implemented in R or even excel once the data table has been generated. The recommendation is to work from raw abundance values initially, and examine binning power prior to normalization.
11. CONCOCT requires slight alterations of this protocol. It requires separate coverage and nucleotide tables, so prepare_e-som_files.pl will have to be run twice: once with the coverage information and -k 0 for no nucleotide information, and once with no coverage information and with -k [1-n] for nucleotide composition.

12. Phylogenetic affiliation of genes can identify mixed bins containing unrelated organisms, but must be applied with caution: genomes for organisms with no close relatives in the databases used for gene annotation will typically display a mixed signal of “best hits” that can erroneously appear as contamination. If single-copy gene numbers suggest no contamination and phylogenetic signal is split between major phyla (e.g., Firmicutes, Proteobacteria, Cyanobacteria), this may indicate a novel lineage genome rather than a mis-binning.
13. Unacceptable bins include bins with less than a given threshold % completion (usually 70%) and bins with greater than a given threshold of contamination (usually 5–10%). One notable exception are DNA virus and plasmid bins, which will present as small (usually less than 100,000 bp) and with no or very few single copy marker genes. Gene annotations can be used to identify these, with hallmark genes for viruses (e.g., capsid and tail proteins) and plasmids (e.g., plasmid replication domain) encoded.

References

1. Venter JC, Remington K, Heidelberg JF et al (2004) Environmental genome shotgun sequencing of the Sargasso Sea. *Science* 304:66–74. <https://doi.org/10.1126/science.1093857>
2. Tyson GW, Chapman J, Hugenholtz P et al (2004) Community structure and metabolism through reconstruction of microbial genomes from the environment. *Nature* 428:37–43. <https://doi.org/10.1038/nature02340>
3. Brown CT, Hug LA, Thomas BC et al (2015) Unusual biology across a group comprising more than 15% of domain Bacteria. *Nature* 523:208–211. <https://doi.org/10.1038/nature14486>
4. Castelle CJ, Wrighton KC, Thomas BC et al (2015) Genomic expansion of domain archaea highlights roles for organisms from new phyla in anaerobic carbon cycling. *Curr Biol* 25:690–701. <https://doi.org/10.1016/j.cub.2015.01.014>
5. Hug LA, Thomas BC, Sharon I et al (2016) Critical biogeochemical functions in the subsurface are associated with bacteria from new phyla and little studied lineages. *Environ Microbiol* 18:159–173. <https://doi.org/10.1111/1462-2920.12930>
6. Albertsen M, Hugenholtz P, Skarszewski A et al (2013) Genome sequences of rare, uncultured bacteria obtained by differential coverage binning of multiple metagenomes. *Nat Biotechnol* 31:533–538. <https://doi.org/10.1038/nbt.2579>
7. Dick GJ, Andersson AF, Baker BJ et al (2009) Community-wide analysis of microbial genome sequence signatures. *Genome Biol* 10:R85. <https://doi.org/10.1186/gb-2009-10-8-r85>
8. Kantor RS, Wrighton KC, Handley KM et al (2013) Small genomes and sparse metabolisms of sediment-associated bacteria from four candidate phyla. *MBio* 4:e00708–e00713. <https://doi.org/10.1128/mBio.00708-13>
9. Wrighton KC, Castelle CJ, Wilkins MJ et al (2014) Metabolic interdependencies between phylogenetically novel fermenters and respiratory organisms in an unconfined aquifer. *ISME J* 8:1452–1463. <https://doi.org/10.1038/ismej.2013.249>
10. Sharon I, Morowitz MJ, Thomas BC et al (2013) Time series community genomics analysis reveals rapid shifts in bacterial species, strains, and phage during infant gut colonization. *Genome Res* 23:111–120. <https://doi.org/10.1101/gr.142315.112>
11. Sharon I, Kertesz M, Hug LA et al (2015) Accurate, multi-kb reads resolve complex populations and detect rare microorganisms. *Genome Res* 25:1830–1831. <https://doi.org/10.1101/gr.183012.114>
12. Frank JA, Pan Y, Tooming-Klunderud A et al (2016) Improved metagenome assemblies and taxonomic binning using long-read circular

- consensus sequence data. *Sci Rep* 6:25373. <https://doi.org/10.1038/srep25373>
13. Joshi N, Fass J (2011) Sickle: A sliding-window, adaptive, quality-based trimming tool for FastQ files. (Version 1.33) [Software]. <https://github.com/najoshi/sickle>
14. Boisvert S, Raymond F, Godzardis E et al (2012) Ray Meta: scalable de novo metagenome assembly and profiling. *Genome Biol* 13: R122. <https://doi.org/10.1186/gb-2012-13-12-r122>
15. Boisvert S, Laviotte F, Corbeil J (2010) Ray: simultaneous assembly of reads from a mix of high-throughput sequencing technologies. *J Comput Biol* 17:1519–1533. <https://doi.org/10.1089/cmb.2009.0238>
16. Peng Y, Leung HCM, Yiu SM, Chin FYL (2012) IDBA-UD: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. *Bioinformatics* 28:1420–1428. <https://doi.org/10.1093/bioinformatics/bts174>
17. Langmead B, Salzberg SL (2012) Fast gapped-read alignment with Bowtie 2. *Nat Methods* 9:357–359. <https://doi.org/10.1038/nmeth.1923>
18. Quinlan AR (2014) BEDTools: the Swiss-Army tool for genome feature analysis. *Curr Protoc Bioinformatics* 47:11.12.1–34. <https://doi.org/10.1002/0471250953.bt112s47>
19. Sommer DD, Delcher AL, Salzberg SL, Pop M (2007) Minimus: a fast, lightweight genome assembler. *BMC Bioinformatics* 8:64. <https://doi.org/10.1186/1471-2105-8-64>
20. Alneberg J, Bjarnason BS, de Bruijn I et al (2014) Binning metagenomic contigs by coverage and composition. *Nat Methods* 11:1144–1146. <https://doi.org/10.1038/nmeth.3103>
21. Kang DD, Froula J, Egan R, Wang Z (2015) MetaBAT, an efficient tool for accurately reconstructing single genomes from complex microbial communities. *PeerJ* 3:e1165. <https://doi.org/10.7717/peerj.1165>
22. Eren AM, Esen ÖC, Quince C et al (2015) Anvi'o: an advanced analysis and visualization platform for 'omics' data. *PeerJ* e1319:3. <https://doi.org/10.7717/peerj.1319>
23. Kang DD, Rubin EM, Wang Z (2016) Reconstructing single genomes from complex microbial communities. *Inf Technol* 58:133–139. <https://doi.org/10.1515/itit-2016-0011>
24. Parks DH, Imelfort M, Skennerton CT et al (2015) CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. *Genome Res* 25:1043–1055. <https://doi.org/10.1101/gr.186072.114>
25. Zerbino DR, Birney E (2008) Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res* 18:821–829. <https://doi.org/10.1101/gr.074492.107>
26. Lever MA, Torti A, Eickenbusch P et al (2015) A modular method for the extraction of DNA and RNA, and the separation of DNA pools from diverse environmental sample types. *Front Microbiol* 6:476. <https://doi.org/10.3389/fmicb.2015.00476>
27. Li D, Liu C-M, Luo R et al (2015) MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics* 31:1674–1676. <https://doi.org/10.1093/bioinformatics/btv033>
28. Li D, Luo R, Liu C-M et al (2016) MEGAHIT v1.0: A fast and scalable metagenome assembler driven by advanced methodologies and community practices. *Methods* 102:3–11. <https://doi.org/10.1016/j.ymeth.2016.02.020>
29. Namiki T, Hachiya T, Tanaka H, Sakakibara Y (2012) MetaVelvet: an extension of Velvet assembler to *de novo* metagenome assembly from short sequence reads. *Nucleic Acids Res* 40:e155. <https://doi.org/10.1093/nar/gks678>
30. Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* 25:1754–1760. <https://doi.org/10.1093/bioinformatics/btp324>



Chapter 15

Transkingdom Networks: A Systems Biology Approach to Identify Causal Members of Host–Microbiota Interactions

Richard R. Rodrigues, Natalia Shulzhenko, and Andrey Morgun

Abstract

Improvements in sequencing technologies and reduced experimental costs have resulted in a vast number of studies generating high-throughput data. Although the number of methods to analyze these “omics” data has also increased, computational complexity and lack of documentation hinder researchers from analyzing their high-throughput data to its true potential. In this chapter we detail our data-driven, transkingdom network (TransNet) analysis protocol to integrate and interrogate multi-omics data. This systems biology approach has allowed us to successfully identify important causal relationships between different taxonomic kingdoms (e.g., mammals and microbes) using diverse types of data.

Key words Omics, Transkingdom, Network analysis, Causal relationships

1 Introduction

Over the last decade assessing eukaryotic and prokaryotic genomes and transcriptomes have become extremely easy. With technologies like microarrays and next-generation sequencing, investigators now have faster and cheaper access to high-throughput “-omics” data [1]. This in turn has increased the number of analysis methods [2] and allows for the exploration of new and different biological questions to provide insights and better understanding of host, host–microbial systems, and diseases [3–5].

Studies usually focus on identifying differences between “groups” (e.g., healthy versus diseased or treatment versus control) or changes across a time course (e.g., development of an organism or progression of a disease). Depending on the biological questions, such studies generate one or more types of omics data [6–8], e.g., host gene expression and gut microbial abundance. Typically, studies analyze these omics data separately, comparing gene

Electronic supplementary material: The online version of this chapter (https://doi.org/10.1007/978-1-4939-8728-3_15) contains supplementary material, which is available to authorized users.

expression and microbial abundance between groups or across stages. Although such analysis methods have been very useful, they do not directly answer the most critical questions of host–microbiota interactions, i.e., which microbes affect specific pathways in the host and which host pathways/genes control specific members of the microbial community? Therefore, to answer those questions, these analyses are usually followed by literature searches to identify relationships between host genes and microbes.

Different algorithms and methods have been proposed to integrate multi-omics data [9–13]. More recently, a few published studies have not only integrated microbiome and host data, but have also been able to successfully test their computational predictions in the laboratory [14–19]. In this chapter we describe our data-driven, transkingdom network (TransNet) analysis pipeline (Fig. 1) that has allowed us to make validatable computational inferences. We construct networks using correlations between differentially expressed elements (e.g., genes, microbes) and integration of high-throughput data from different taxonomic kingdoms (e.g., human and bacteria). In fact, TransNet analysis can be applied to integrate any “Transomics” data, between as well as within taxonomic kingdoms, e.g., miRNA and gene expression, protein and metabolite, bacterial and host gene expression, or copy number, methylation, and gene expression, provided the different data are obtained from the same samples. Interrogation of this network allows us to pinpoint important causal relationships between data. For example, using this method we inferred and validated: (1) microbes and microbial genes controlling a specific mammalian pathway [15]; (2) a microbe that mediates effect of one host pathway on another [14]; (3) a host gene that mediates control of gut microbe through an upstream master regulator gene [14]. Below we show how TransNet analysis can be used to integrate host gene expression with microbial abundance to create transkingdom networks.

2 Materials

2.1 Program Availability

Our transkingdom network analysis pipeline is independent of programming language or software. However, for ease of access and usage simplicity, we have provided our pipeline as a convenient R package `TransNetDemo` (<https://github.com/richrr/TransNetDemo>) and supplementary document (File S1) in addition to the description provided. Although the user can choose to perform the following steps in a programming language or software of their choice, we suggest using our R package.

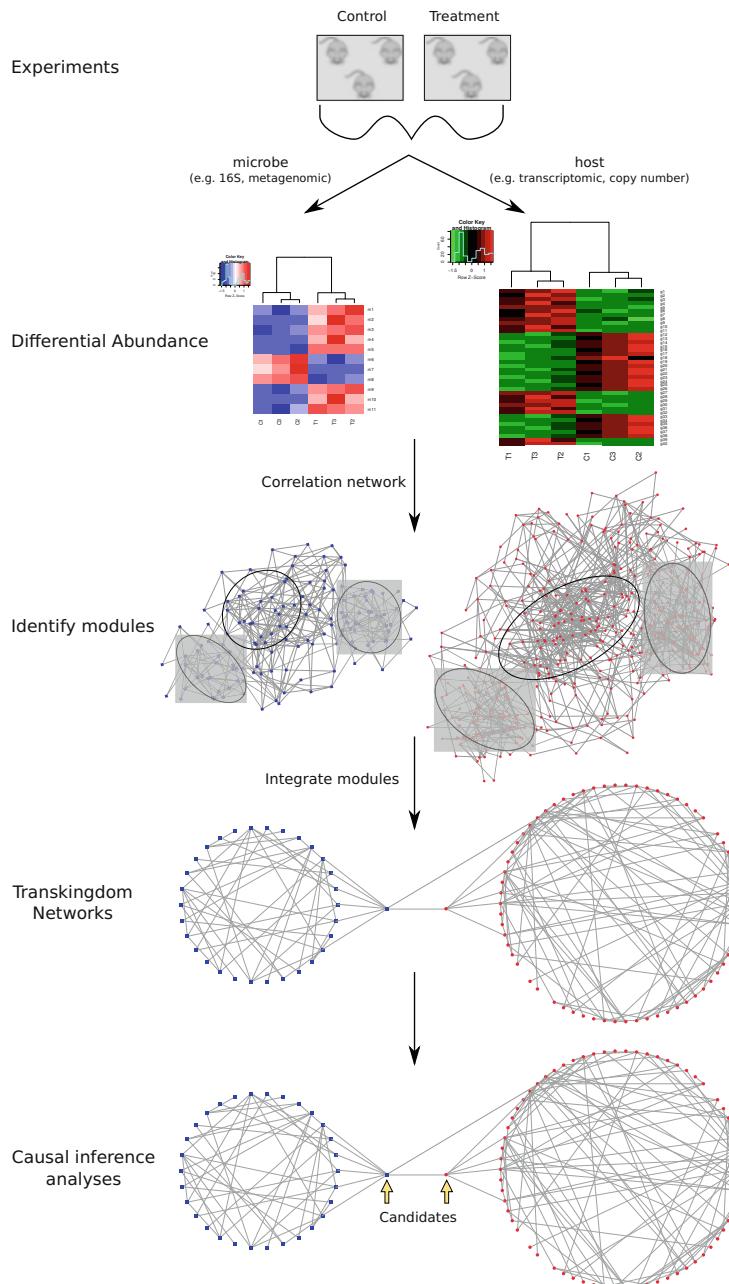


Fig. 1 Overview of transkingdom network analysis. Omics data for multiple data types (e.g., microbial, gene expression) are analyzed to identify differentially abundant elements (e.g., microbes, genes). For each group (e.g., treatment or control) co-expression networks are constructed for each data type followed by the identification of dense subnetworks (modules). Calculating correlations between module elements of the different data types creates the “transkingdom” network. Network interrogation of the transkingdom network allows identification of causal members and regulatory relationships

2.1.1 Required**R Packages**

Install the following packages along with their dependencies: stringr, ProNet, igraph, ggplot2, gplots from CRAN (<https://cran.r-project.org/>). The following commands will automatically install the required version of the packages (<https://github.com/richrr/TransNetDemo/blob/master/DESCRIPTION>).

2.1.2 Installing**TransNetDemo**

- library(devtools)

- install_github("richrr/TransNetDemo")

- library(TransNetDemo)

2.1.3 Code Referenced in**the Chapter**

The following scripts (available at <https://github.com/richrr/TransNetDemo/tree/master/inst/demo> and File S1) guide users in running TransNet:

- GeneDemo.R
- MicrobeDemo.R
- GeneMicrobeDemo.R
- Heatmaps.R

The following functions (available at <https://github.com/richrr/TransNetDemo/tree/master/R> and File S1) are used in the pipeline:

- Apply_sign_cutoffs.R
- Calc_bipartite_betweenness_centrality.R
- Calc_combined.R
- Calc_cor.R
- Calc_median_val.R
- Check_consistency.R
- Compare_groups.R
- Correlation_in_group.R
- Diff_abundance.R
- Get_shortest_paths.R
- Get_template_matrix.R
- Identify_subnetworks.R
- Puc_compatiable_network.R

2.2 Data Sources

Due to a variety of data generation technologies, biological questions, and software, description of every possible analysis is beyond the scope of this chapter. We expect that the user has access to tab-delimited file(s) containing the measurements of biological data type, e.g., gene expression, copy number, methylation, miRNA, or microbial abundances across samples. Depending on the data type the user can find reviews and protocol papers describing the analysis needed to produce “abundance” tables [20–24].

The transkingdom network analysis method can be applied to any experimental design (e.g., treated/untreated, control/disease). As an example we will use simulated data from a simple experimental design, where 25 mice each are fed either high fat high sucrose (HFHS) or normal chow diet (NCD) for 8 weeks, to investigate the effects of diet on host–microbial interactions. At the end of the experiment, among other phenotypic measurements (e.g., body weight, enzyme levels, hormone levels), the gene expression levels and microbial abundance in the gut (e.g., ileum) of the samples were measured. Depending on resource availability, high confidence and consistent results can be achieved by increasing the number of samples per group and/or repeating the above experiment multiple times. In this example data, we have two such experiments. A brief description of how to generate the abundance tables is mentioned below. Information about how the network analysis protocol can be adapted to answer some other biological questions have been mentioned in Subheading 4 of this chapter.

2.3 Gene Expression Analysis

Several different technologies, each with their own pros and cons, allow for the measuring of transcriptome levels in an organism. Although microarrays were extensively used over the last two decades, the availability of cheap and efficient library preparation kits and sequencing methods allow for the expression measurements of known and novel genes using RNA-Seq technologies [25].

In case of RNA-Seq data, the sequencing facilities usually provide fastq files that contain raw reads per sample (demultiplexed) (*see Note 1*). Here, the number of reads corresponding to a particular gene is proportional to that gene’s expression level. Software like FASTX-Toolkit (http://hannonlab.cshl.edu/fastx_toolkit/), FastQC (<http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>), PRINSEQ [26], or cutadapt [27] can be used for adapter removal and quality control. Depending on the availability of a gold-standard reference host genome sequence, gene expression abundance can be measured using the Tuxedo [21] or Trinity [28] pipeline. Both of these pipelines permit the analysis of single or paired reads and different read lengths (*see Note 2*) while outputting a file containing the expression levels (number of reads) of genes (rows) present in each sample (columns). The obtained read counts can be normalized by simple (e.g., quantile normalization, counts per million (CPM), reads per kilobase per million mapped reads (RPKM) [29]) or more sophisticated methods (e.g., DESEQ [30], edgeR [31]) (*see Notes 3 and 4*).

2.4 Microbial Abundance Analysis

The advent of next-generation technologies has helped in the study of microbial richness and diversity. Scientists no longer need to rely on cultivation methods and can directly sequence the microbiome, helping explore previously unknown microbes. The amplicon-based sequencing technologies rely on using a gene marker (16S

[32, 33] ribosomal RNA gene, Internal Transcribed Spacer [34, 35], etc.) to identify microbial presence and abundance. Although relatively cheaper than shotgun metagenomics, they rely on databases of known genomic markers to identify microbes and rarely provide taxonomy at the species or strain levels. The shotgun metagenomics sequencing approach does a better job at surveying the entire genome of microbes since it does not focus on amplifying specific genes. Consequently, it provides fine-grained taxonomic information along with a more accurate representation of the microbial structure and function, including the previously unknown “dark matter” microbes [36].

Software like QIIME [37], MOTHUR [38], etc. provide all-in-one toolkits that can demultiplex, perform quality control, and analyze the amplicon-based sequences. Similar to RNA-Seq data, the fastq files obtained from the sequencing facility need to be processed for the removal of barcodes, adapter, and primers followed by filtering to retain high quality sequences. The reads are grouped (binned) per sequence similarity (usually at 97% threshold) into operational taxonomic units (OTUs). The taxonomy of a known microbe (or the ancestor taxonomy of the top matches) closest to the representative sequence of the OTU is assigned to all the reads in that OTU. The tools output a file containing the abundance (number of reads) of OTUs (rows) present in each sample (columns). The obtained read counts can be relativized or cumulative sum scaling (CSS) [39] normalized.

Shotgun metagenomic data can be analyzed [36] using tools such as MG-RAST [40], MEGAN [41], MetaPhlAn [42], and HUMAnN [43]. Although most of these software packages provide taxonomic and functional analyses, they are not standalone. Demultiplexing and quality control need to be done before the reads are imported in the software. Especially in case of host-microbe systems, PuMA (<http://blogs.oregonstate.edu/morganshulzhenkolabs/softwares/puma>) provides an all-inclusive software pipeline that can be more user-friendly. PuMA uses cutadapt for quality control and Bowtie [44] to identify reads that match the host genome and discards these “contaminating” reads from downstream analysis. The remaining microbial reads are aligned to a database of known protein sequences using DIAMOND [45], followed by taxonomic and functional (e.g., SEED, COG, KEGG) assignments using MEGAN. PuMA outputs a file containing the abundance of microbes and pathways (rows) in each sample (columns). The appropriate normalization techniques from the RNA-Seq or amplicon sequencing methods can be performed on the abundance table.

In summary, the user needs at least one of each of the following files before starting network analysis:

- Mapping file: tab-delimited file containing the group (e.g., treated/untreated, control/disease) affiliation for each sample with “Factor” and “SampleID” as column headers, respectively.
- Data files: tab-delimited files containing the abundance of elements (host genes and microbes) per sample, where the elements and samples are rows and columns, respectively. Importantly, each sample must have both types of data available.
 - Normalized gene expression file: the column “IdSymbol” contains the unique genes while the remaining columns contain their expression levels across different samples.
 - Normalized otu abundance file: the column “IdSymbol” contains the unique microbes while the remaining columns contain their abundance across different samples.

3 Methods

The following steps will help to identify key elements of a system from high-confidence modules of a multi-omics network. We show the first few steps with the gene abundance file(s) using the code from the GeneDemo.R (GD) file available in our package. It is straightforward to run similar steps on the microbe abundance file (s); however, we have also provided the code in MicrobeDemo.R (MD) file for ease of use.

- Start by setting defaults for variables that you will use in the analysis, such as significance thresholds (GD: lines 7–9), groups to be compared (GD: lines 11–13), and headers of relevant columns from the mapping (GD: lines 14–15) and abundance files (GD: line 16).
- Next you want to identify the differentially expressed elements (GD: line 29). The network analysis can be performed using all (differentially and non-differentially expressed) elements (genes, microbes, etc.). However, we suggest identifying the elements that show differential abundance between groups (*see Notes 5 and 6*), using code from Compare_groups.R (Cg) file, to focus on the important elements and make the analyses computationally efficient.
 - Read from the mapping file to extract the samples from each group (Cg: lines 11–20).
 - Read from the gene abundance file (Cg: lines 22–25).
 - Then perform test for differential abundance using code from Diff_abundance.R (Da) file. This function returns the mean and median for each group along with the fold change and *p*-value (Da: lines 8–28).

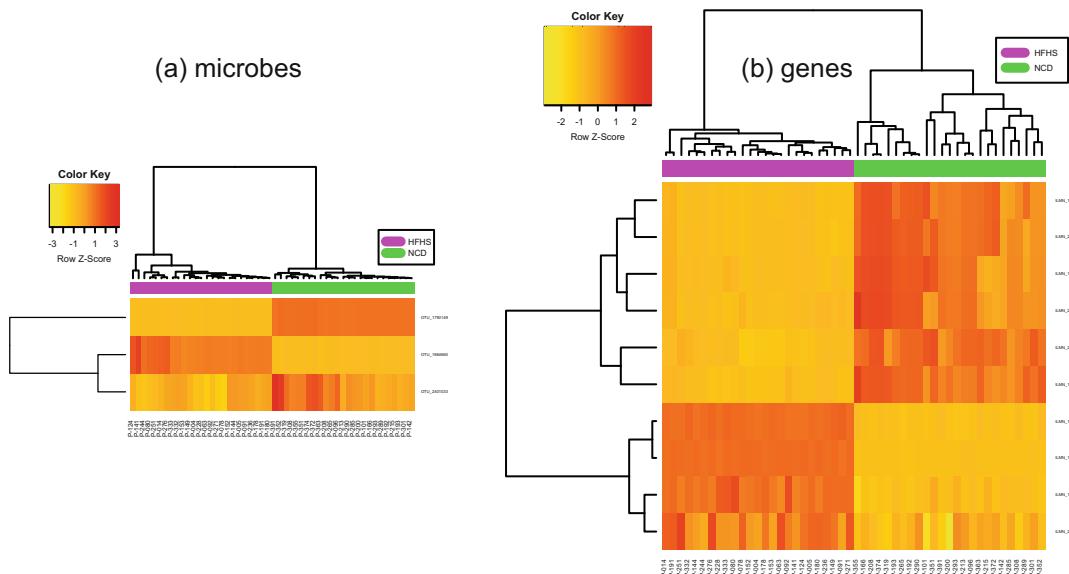


Fig. 2 Heat map from hierarchical clustering of differentially abundant elements. Rows indicate **(a)** microbes and **(b)** genes, while columns indicate samples. The purple and green colors indicate samples belonging to the groups A (HFHS) and B (NCD), respectively. The yellow and orange colors indicate decrease and increase, respectively, in expression or abundance, and color intensity corresponds to the level of fold change

- Next, account for multiple testing using Benjamini-Hochberg's FDR calculation (Cg: lines 38–41).
 - Finally, select the differentially expressed genes using appropriate FDR cutoff (GD: line 33) (<0.05) (Fig. 2b).
 - We highly recommend that if you have datasets obtained from replicate experiments or in different sample cohorts that you perform the above steps for each experiment and do meta-analysis [16, 18, 46, 47] (GD: lines 39–47).
 - First, the meta-analysis selects for genes that show fold change direction consistency across datasets (Check_consistency.R), e.g., upregulated (or downregulated) across all experiments.
 - Second, for the genes showing consistent fold change direction use Fisher's method to calculate a combined *p*-value (Calc_combined.R) from the individual *p*-values (from comparison test) across multiple experiments.
 - Then apply appropriate significance thresholds (Apply_sign_cutoffs.R) based on individual *p*-value (<0.3) in each dataset, combined (Fisher's) *p*-value across datasets (<0.05), and FDR (<0.1) across the combined *p*-values to identify consistently differentially abundant elements.
 - Ensuring the same direction of regulation in all datasets and restricting individual *p*-values at each individual dataset allows

controlling of heterogeneity between datasets. Note that mere calculation of Fisher *p*-value for meta-analysis followed by application of FDR is not sufficient for accurate identification of differential abundance/expression.

- Determining associations between elements (e.g., genes and/or microbes) is central for network reconstruction. Defining strength and sign of correlation (GD: line 56) can help to determine whether two elements (i.e., biological entities represented by nodes in a network) have a positive or negative interaction. Such information about potential relationships (*see Note 7*), using code from Correlation_in_group.R (Cig) file, is crucial for interrogating and understanding the regulatory mechanisms between elements. Note, correlations are calculated using data from samples representing one group (phenotypic class), never pooling samples from all groups for estimation of correlation. Therefore, the following steps should be performed for each group separately.
 - Read from the mapping file to extract the samples from a group (Cig: lines 11–18).
 - Read from the gene abundance file (Cig: lines 20–23).
 - Then create pairs (Cig: lines 25–32) from the consistent genes obtained in the previous step.
 - Next perform test for correlation on gene pairs using code from Calc_cor.R (Ccr) file. This function returns the correlation and *p*-value (Ccr: lines 8–17).
 - Next, account for multiple testing using Benjamini-Hochberg's FDR calculation (Cig: lines 48–50).
 - Finally, select the significantly correlated gene pairs using appropriate FDR cutoff (GD: line 60) < 0.1.
- We highly recommend that if you have datasets obtained from replicate experiments or different sample cohorts that you perform the above steps for each experiment and do meta-analysis (GD: lines 65–72).
 - First, the meta-analysis selects for gene pairs that show correlation direction consistency across datasets (Check_consistency.R), e.g., positive (or negative) across all experiments.
 - The next steps of combining the individual *p*-values (from correlation test) and applying multiple significance cutoffs are similar to those in the meta-analysis of genes.
- At this point you have a network for a single group where nodes are genes and edges indicate significant correlation. Next, we identify the proportion of unexpected correlations (PUC) [48] (GD: line 83). Edges in a network where the sign of correlations does not correspond to the direction of change are unexpected (*see Note 8*), are not likely to contribute to the process under

investigation, and hence discarded using code from Puc_compatible_network.R (Pcn) file.

- First, for each gene pair identify the sign of correlation (Pcn: lines 47–53).
- Second, calculate if each gene in the pair has the same direction of regulation (i.e., fold change) (Pcn: lines 56–65).
- Pairs are expected and kept (Pcn: lines 70–80) if they satisfy either of these conditions:
 - Positively correlated genes have the same fold change direction.
 - Negatively correlated genes have different fold change direction.
- At this point you have a network consisting of regulatory relationships. Next, the obtained network can be systematically studied to answer different biological questions (*see Note 9*). Most often, network interrogation relies on identifying highly inter-connected sets of nodes. Such a subnetwork is called a module (or cluster). Identify clusters (GD: line 89) using the MCODE method (*see Note 10*) from the Identify_subnetworks.R file (Fig. 3b).
- Repeat the above steps for the microbial (or any other data type) abundance file(s) to obtain heat map (Fig. 2a) and clusters (Fig. 3a) per biological data type (e.g., genes, microbes). Refer to the code in MicrobeDemo.R file.

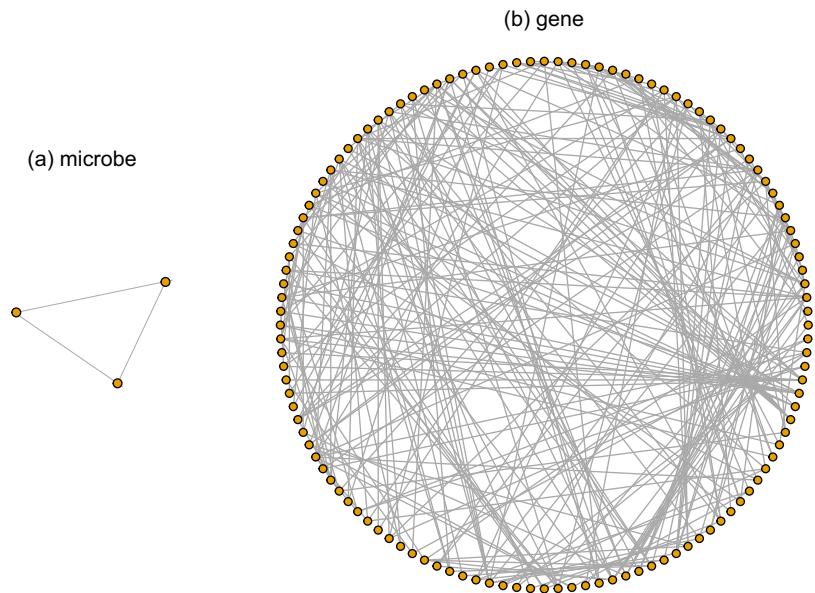


Fig. 3 Clusters obtained from the correlation networks. The PUC compatible (a) microbe and (b) gene networks for an individual group (HFHS) are mined to identify densely connected subnetworks. Edges indicate significant correlation between elements

- The next step is to integrate subnetworks to create transkingdom networks using code from the GeneMicrobeDemo.R (GMD) file. Note that at this point you have already identified modules from the gene and the microbe networks. Similar to the above steps, create pairs between nodes from the different modules (GMD: line 29) (*see Note 11*), calculate correlations within a group (GMD: line 32), and identify significant pairs based on single (GMD: line 36) or meta (GMD: lines 41–49) analysis. Next, apply PUC analysis and remove unexpected edges from this transkingdom (gene–microbe) network as it is done for regular gene expression (and microbial abundance) network (GMD: line 58).
- Combining the gene–gene correlations (edges from the gene subnetworks) (GMD: line 74), microbe–microbe correlations (edges from the microbe subnetworks) (GMD: line 77), and the gene–microbe correlations (GMD: line 80) creates the full transkingdom network (GMD: line 83) (Fig. 4).

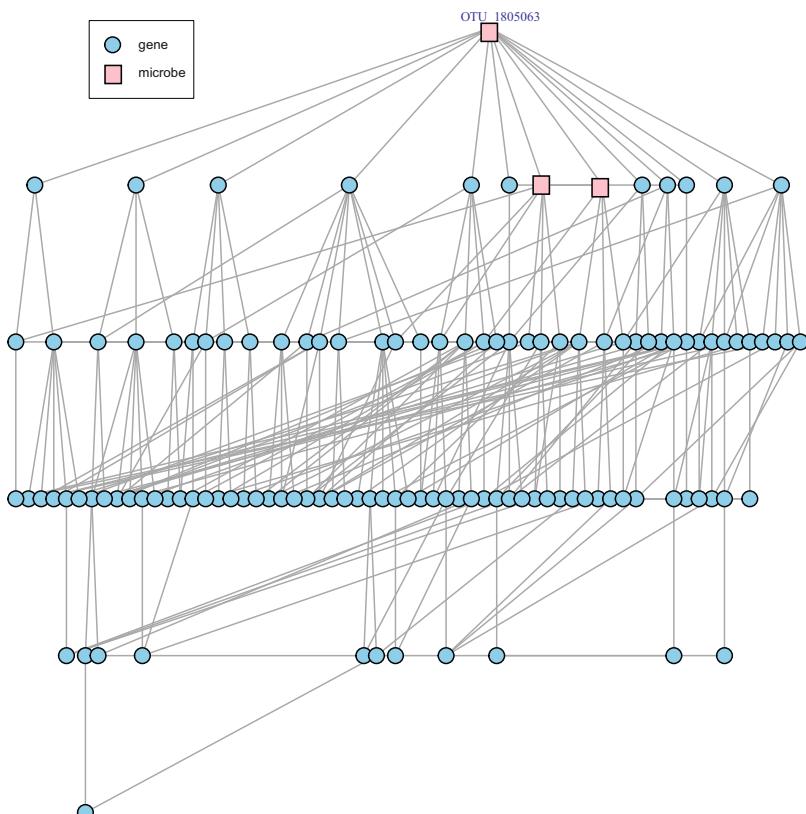


Fig. 4 Transkingdom network. A full network, for the HFHS group, contains gene–gene, microbe–microbe, and gene–microbe edges. Edges indicate significant correlation between elements. The blue circle and pink square indicate gene and microbe nodes, respectively. The labeled node has the highest BiBC measurement among microbes and is therefore considered to be important and a potential causal player in the experiment

- Finally, identify elements that are crucial for crosstalk between the different modules in a network using bipartite betweenness centrality (BiBC) (GMD: lines 92–119) (*see Note 12*). This approach involves calculating the shortest paths between nodes from different modules using code from Get_shortest_paths.R file. The elements with the highest BiBC measurement (GMD: lines 123–128) are more likely to be critical in mediating the transfer of signals between the different modules of a network and candidates for further experimentation.

4 Notes

The above protocol was written for a step-by-step introduction to transkingdom network analysis. Although the above experimental setup and analyses should suffice in most cases please see the following suggestions for other alternatives to the analysis.

1. Attaching unique barcodes to samples in the amplification step of sequencing library preparation allows multiple samples to be pooled in a single sequencing run. This process is termed as multiplexing. After sequencing the barcodes can be used to separate reads per sample, a process termed as demultiplexing.
2. The DNA fragment (template) can be sequenced in single or both directions and is referred to as single-end or paired-end sequencing. Read length refers to the number of bases sequenced per DNA fragment. For example, 250-bp paired-end sequencing means that 250 bases were sequenced from each end of the DNA giving one fastq file per sample for each end of the read.
3. Data normalization is a crucial step in analysis and network reconstruction [49]; hence, choose the appropriate normalization method for your biological data [50, 51]. Normalization methods differ in how they account for the sequencing depth (in next-generation sequencing data), gene or transcript length, estimation of data variability; however, no normalization method universally outperforms other methods. However, if unsure about which normalization to use we recommend quantile normalization since, in our experience, it works well for most biological data.
4. In the case of microarray data, hybridization facilities usually provide scan files (Affymetrix CEL, Illumina IDAT, or GenePix GPR) that contain the intensity of probes per sample. Here, the probe intensity is proportional to the corresponding gene expression level. Software like Affymetrix® Expression Console™, Illumina’s GenomeStudio, and GenePix® Pro, as well as packages like affy [52] and limma [53], allow for background

correction, normalization, and summarized probe intensities while outputting a file containing the expression levels of genes (rows) present in each sample (columns).

5. Depending on the experimental design and biological question apply appropriate parametric (paired or unpaired *t*-test, analysis of variance (ANOVA), multivariate ANOVA (MANOVA), etc.) and nonparametric (Man-Whitney, Wilcoxon rank sum test, Multi-response Permutation Procedures (MRPP), etc.) tests to identify differential abundance.
6. It is common practice to visualize the levels of differentially abundant elements. The code from Heatmaps.R file can help to visualize the significant genes and microbes from our example.
7. Pearson or Spearman correlation analyses between two elements from the same samples should suffice. However, use partial correlation [54] or other methods [55] to detect correlations and reduce indirect interactions.
8. If two elements have a regulatory relationship we expect them to behave in certain ways. For example, consider two groups. Two positively correlated genes in a group should have the same direction of fold change between two groups. On the other hand, two negatively correlated genes should have the opposite direction of fold change [48].
9. The network analysis can be extended to identify differentially correlated genes in co-expression networks obtained for the different groups and uncover regulatory mechanisms in phenotypic transitions [56, 57].
10. Cfinder and graph clustering (MCL) [19] are some other tools to help identify modules in networks.
11. In our example, gene expression was correlated with taxon abundance to identify genes and microbes with similar or opposite variation across samples within a group. Such pairs indicate potential associations between the nodes. Correlations between other data types are possible, provided that the measurements are available from the same set of samples.
12. In our example, we used the bipartite betweenness centrality measure to identify elements that are important for crosstalk between the different modules of the network. This is because the nodes with high BiBC scores lie on the largest number of shortest paths taken by nodes between the different modules to communicate with each other and therefore have more control over information passing in the network. BiBC assumes that every pair of node pass equally important messages and that nodes always communicate using the shortest paths which may not be true in each case. Therefore, depending on the biological question, the user can inspect multiple network

topology properties such as the degree, eccentricity, and centrality measures using *NetworkAnalyzer* in Cytoscape to identify important elements in the full transkingdom network.

Acknowledgments

The authors thank Karen N. D’Souza, Khiem Lam, and Dr. Xiaoxi Dong for their help in writing the book chapter. This work was supported by the NIH U01 AI109695 (AM) and R01 DK103761 (NS).

References

1. Schuster SC (2008) Next-generation sequencing transforms today’s biology. *Nat Methods* 5(1):16–18
2. Metzker ML (2010) Sequencing technologies—the next generation. *Nat Rev Genet* 11(1):31–46
3. Goodwin S, McPherson JD, McCombie WR (2016) Coming of age: ten years of next-generation sequencing technologies. *Nat Rev Genet* 17(6):333–351
4. Mardis ER (2008) The impact of next-generation sequencing technology on genetics. *Trends Genet* 24(3):133–141
5. Morozova O, Marra MA (2008) Applications of next-generation sequencing technologies in functional genomics. *Genomics* 92(5):255–264
6. Erickson AR et al (2012) Integrated metagenomics/metaproteomics reveals human host-microbiota signatures of Crohn’s disease. *PLoS One* 7(11):e49138
7. Moreno-Risueno MA, Busch W, Benfey PN (2010) Omics meet networks—using systems approaches to infer regulatory networks in plants. *Curr Opin Plant Biol* 13(2):126–131
8. Imhann F et al (2016) Interplay of host genetics and gut microbiota underlying the onset and clinical presentation of inflammatory bowel disease. In: *Gut*
9. Joyce AR, Palsson BO (2006) The model organism as a system: integrating ‘omics’ data sets. *Nat Rev Mol Cell Biol* 7(3):198–210
10. Gehlenborg N et al (2010) Visualization of omics data for systems biology. *Nat Methods* 7(3 Suppl):S56–S68
11. Poirel CL et al (2013) Reconciling differential gene expression data with molecular interaction networks. *Bioinformatics* 29(5):622–629
12. Zhang W, Li F, Nie L (2010) Integrating multiple ‘omics’ analysis for microbial biology: application and methodologies. *Microbiology* 156(Pt 2):287–301
13. Greer R et al (2016) Investigating a holobiont: Microbiota perturbations and transkingdom networks. *Gut Microbes* 7(2):126–135
14. Greer RL et al (2016) Akkermansia muciniphila mediates negative effects of IFNgamma on glucose metabolism. *Nat Commun* 7:13329
15. Morgan A et al (2015) Uncovering effects of antibiotics on the host and microbiota using transkingdom gene networks. *Gut* 64(11):1732–1743
16. Mine KL et al (2013) Gene network reconstruction reveals cell cycle and antiviral genes as major drivers of cervical cancer. *Nat Commun* 4:1806
17. Schirmer M et al (2016) Linking the Human Gut Microbiome to Inflammatory Cytokine Production Capacity. *Cell* 167(4):1125–1136 e8
18. Shulzhenko N et al (2011) Crosstalk between B lymphocytes, microbiota and the intestinal epithelium governs immunity versus metabolism in the gut. *Nat Med* 17(12):1585–1593
19. Dong X et al (2015) Reverse enGENEering of Regulatory Networks from Big Data: A Roadmap for Biologists. *Bioinform Biol Insights* 9:61–74

20. Caporaso JG et al (2010) QIIME allows analysis of high-throughput community sequencing data. *Nat Methods* 7(5):335–336
21. Trapnell C et al (2012) Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nat Protoc* 7(3):562–578
22. Laird PW (2010) Principles and challenges of genomewide DNA methylation analysis. *Nat Rev Genet* 11(3):191–203
23. Krumm N et al (2012) Copy number variation detection and genotyping from exome sequence data. *Genome Res* 22(8):1525–1532
24. Perez-Diez A, Morgan A, Shulzhenko N (2007) Microarrays for cancer diagnosis and classification. *Adv Exp Med Biol* 593:74–85
25. Zhao S et al (2014) Comparison of RNA-Seq and microarray in transcriptome profiling of activated T cells. *PLoS One* 9(1):e78644
26. Schmieder R, Edwards R (2011) Quality control and preprocessing of metagenomic datasets. *Bioinformatics* 27(6):863–864
27. Martin M (2011) Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnetJ* 17(1):10
28. Haas BJ et al (2013) De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. *Nat Protoc* 8(8):1494–1512
29. Mortazavi A et al (2008) Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat Methods* 5(7):621–628
30. Anders S, Huber W (2010) Differential expression analysis for sequence count data. *Genome Biol* 11(10):R106
31. McCarthy DJ, Chen Y, Smyth GK (2012) Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. *Nucleic Acids Res* 40(10):4288–4297
32. Stackebrandt E, Goebel BM (1994) Taxonomic Note: A Place for DNA-DNA Reassociation and 16S rRNA Sequence Analysis in the Present Species Definition in Bacteriology. *Int J Syst Evol Microbiol* 44(4):846–849
33. Lane DJ et al (1985) Rapid determination of 16S ribosomal RNA sequences for phylogenetic analyses. *Proc Natl Acad Sci U S A* 82(20):6955–6959
34. Brookman JL et al (2000) Identification and characterization of anaerobic gut fungi using molecular methodologies based on ribosomal ITS1 and 18S rRNA. *Microbiology* 146(Pt 2):393–403
35. Schoch CL et al (2012) Nuclear ribosomal internal transcribed spacer (ITS) region as a universal DNA barcode marker for Fungi. *Proc Natl Acad Sci U S A* 109(16):6241–6246
36. Sharpton TJ (2014) An introduction to the analysis of shotgun metagenomic data. *Front Plant Sci* 5:209
37. Kuczynski J et al (2011) Using QIIME to analyze 16S rRNA gene sequences from microbial communities. *Curr Protoc Bioinformatics* 10:7 Chapter 10. Unit
38. Schloss PD et al (2009) Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Appl Environ Microbiol* 75(23):7537–7541
39. Paulson JN et al (2013) Differential abundance analysis for microbial marker-gene surveys. *Nat Methods* 10(12):1200–1202
40. Meyer F et al (2008) The metagenomics RAST server - a public resource for the automatic phylogenetic and functional analysis of metagenomes. *BMC Bioinform* 9:386
41. Huson DH, Weber N (2013) Microbial community analysis using MEGAN. *Methods Enzymol* 531:465–485
42. Segata N et al (2012) Metagenomic microbial community profiling using unique clade-specific marker genes. *Nat Methods* 9(8):811–814
43. Lindgreen S, Adair KL, Gardner PP (2016) An evaluation of the accuracy and speed of metagenome analysis tools. *Sci Rep* 6:19233
44. Langmead B et al (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol* 10(3):R25
45. Buchfink B, Xie C, Huson DH (2015) Fast and sensitive protein alignment using DIAMOND. *Nat Methods* 12(1):59–60
46. Rodrigues RR, Barry CT (2011) Gene pathway analysis of hepatocellular carcinoma genomic expression datasets. *J Surg Res* 170(1):e85–e92
47. Morgan A et al (2006) Molecular profiling improves diagnoses of rejection and infection in transplanted organs. *Circ Res* 98(12):e74–e83
48. Yambartsev A et al (2016) Unexpected links reflect the noise in networks. *Biol Direct* 11(1):52
49. Saccenti E (2017) Correlation patterns in experimental data are affected by normalization procedures: consequences for data analysis and

- network inference. *J Proteome Res* 16(2):619–634
50. Hua YJ et al (2008) Comparison of normalization methods with microRNA microarray. *Genomics* 92(2):122–128
51. Li P et al (2015) Comparing the normalization methods for the differential analysis of Illumina high-throughput RNA-Seq data. *BMC Bioinform* 16:347
52. Gautier (2004) L., et al., affy--analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics* 20(3):307–315
53. Ritchie (2015) M.E., et al., limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res* 43(7):e47
54. de la Fuente A et al (2004) Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics* 20(18):3565–3574
55. Weiss S et al (2016) Correlation detection strategies in microbial data sets vary widely in sensitivity and precision. *ISME J* 10(7):1669–1681
56. Thomas LD et al (2016) Differentially correlated genes in co-expression networks control phenotype transitions. *F1000Res* 5:2740
57. Skinner J et al (2011) Construct and Compare Gene Coexpression Networks with DAPfinder and DAPview. *BMC Bioinform* 12:286



Chapter 16

Constructing and Analyzing Microbiome Networks in R

Mehdi Layeghifard, David M. Hwang, and David S. Guttman

Abstract

Microbiomes are complex microbial communities whose structure and function are heavily influenced by microbe–microbe and microbe–host interactions mediated by a range of mechanisms, all of which have been implicated in the modulation of disease progression and clinical outcome. Therefore, understanding the microbiome as a whole, including both the complex interplay among microbial taxa and interactions with their hosts, is essential for understanding the spectrum of roles played by microbiomes in host health, development, dysbiosis, and polymicrobial infections. Network theory, in the form of systems-oriented, graph-theoretical approaches, is an exciting holistic methodology that can facilitate microbiome analysis and enhance our understanding of the complex ecological and evolutionary processes involved. Using network theory, one can model and analyze a microbiome and all its complex interactions in a single network. Here, we describe in detail and step by step, the process of building, analyzing and visualizing microbiome networks from operational taxonomic unit (OTU) tables in R and RStudio, using several different approaches and extensively commented code snippets.

Key words Graph theory, igraph, Microbial co-occurrence, Microbiome, Network, OTU table, R, RStudio

1 Introduction

Microbiomes are complex microbial communities whose structure and function are heavily influenced by microbe–microbe and microbe–host interactions. These interactions are mediated by a range of mechanisms, encompassing direct cell-to-cell contact and interspecies signaling, to indirect metabolite sensing, all of which have been implicated in the modulation of disease progression and clinical outcome [1]. An example of complex microbial interactions involved in disease aggravation is polymicrobial synergism, which describes cases where infections by multiple interacting species of bacteria are more severe than single-agent infections. Polymicrobial synergism is reported to result in increased levels of antibiotic resistance, biofilm development, tissue damage and adaptation to the environment [2, 3]. Therefore, understanding the microbiome

as a whole, including both the complex interplay among microbial taxa and interactions with their hosts, is essential for understanding the spectrum of roles played by microbiomes in host health, development, dysbiosis, and polymicrobial infections.

While the widespread adoption of next-generation sequencing technologies has dramatically increased the scope and scale of microbiome studies, the analytical methodologies used to study microbe–microbe and host–microbe interactions are surprisingly limited [4]. Network theory, in the form of systems-oriented, graph-theoretical approaches, is an exciting holistic methodology that can facilitate microbiome analysis and enhance our understanding of the complex ecological and evolutionary processes involved. Using network theory, one can model and analyze a microbiome and all its complex interactions in a single network. An interesting aspect of network theory is that the architectural features of networks appear to be universal to most complex systems, such as microbiomes, molecular interaction networks, computer networks, microcircuits, and social networks [5]. This universality paves the way for using expertise developed in well-studied nonbiological systems to unravel the interwoven relationships that shape microbial interactions.

The first network model described mathematically is the random network introduced in 1960 by Paul Erdős and Alfred Rényi [6]. This model assumes a network of randomly interconnected nodes, in which nodes' degrees will follow a Poisson distribution and most nodes have a number of connections comparable to the network's average degree. Most natural or artificial networks, however, show a power-law degree distribution, where a few nodes have a very large number of connections, while other nodes have no or few connections. These networks are often called scale-free networks [7]. There are also small-world networks, which describe a model in which most nodes are accessible to every other node through a relatively short path. Finally, regular networks are highly ordered nonrandom networks, where all the nodes have exactly the same degree [8].

A wide range of methods, with varying levels of efficiency and accuracy, have been used to construct networks based on microbiome data. The simplest methods are (dis)similarity- or distance-based techniques. The most popular methods, however, are correlation-based techniques, where significant pairwise associations between operational taxonomic units (OTUs, a grouping of organisms circumscribed by a specified level of DNA sequence similarity at a marker gene) are detected using a correlation coefficient such as Pearson's correlation coefficient or Spearman's non-parametric rank correlation coefficient. However, the use of correlation coefficients to detect dependencies between members

of a microbiome suffers from limitations such as detecting spurious correlations due to compositionality [9], and being severely under-powered owing to the relatively low number of samples.

The concerns over correlation-based analyses have led to the development of methods that are robust to compositionality. SparCC (Sparse Correlations for Compositional data), for example, is a new technique that uses linear Pearson's correlations between the log-transformed components to infer associations in compositional data [10]. SPIEC-EASI (SParse InversE Covariance Estimation for Ecological Association Inference) is another statistical method for the inference of microbial ecological networks that combines data transformations developed for compositional data analysis with a graphical model inference framework with the assumption that the underlying ecological association network is sparse [11].

An alternative approach uses probabilistic graphical models (PGMs), which provides a probability theory framework based on discrete data structures in computer science, to measure uncertainty in high dimensional data. PGMs use probability theory and graph theory in combination to tackle both uncertainty and complexity in data, simultaneously. PGMs use graphs as the foundation for both measuring joint probability distributions and representing sets of conditional dependence within data in a compact fashion (see [12] for a detailed review of various network construction methods applicable to microbiome data). One method, *EBIC-glasso*, estimates sparse undirected graphical models for continuous data with multivariate Gaussian distribution through the use of L1 (*lasso*) regularization before using an extended Bayesian information criteria (EBIC) to select the most fitting model. *lasso* regularization is a regression-analysis method that enforces a sparsity constraint on the data that can lead to simpler and more interpretable models less prone to overfitting. *lasso* essentially makes the data smaller (i.e., sparse) and performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model by selecting only a subset of the provided covariates for use in the final model [13]. Given a collection of graphical models for the data, information criteria enable us to estimate the relative quality of each model or tune parameters of penalization methods such as the graphical lasso. Thus, EBIC provides a means for model selection and optimization.

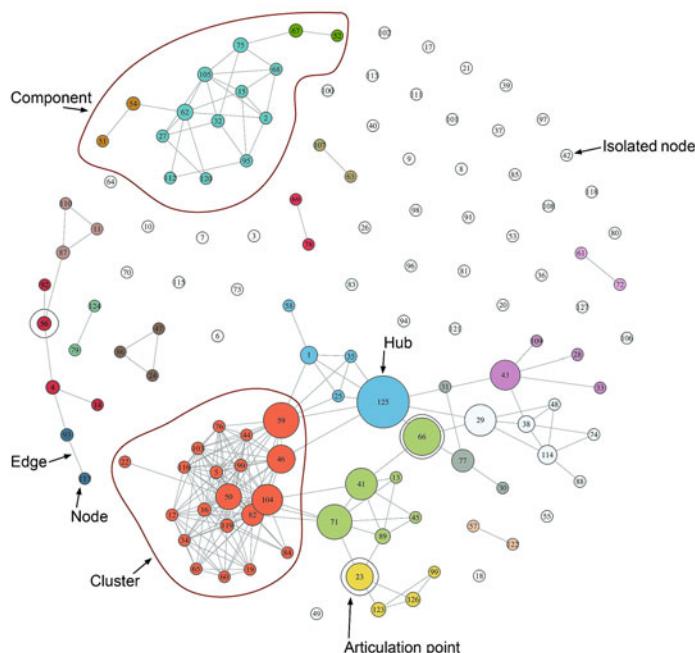
A microbiome network usually consists of clusters (also known as modules) of closely associated microbes (i.e., groups of coexisting or co-evolving microbes), as well as individual microbes that are central to the network's structure (i.e., keystone taxa). Clusters not only represent the local interaction patterns in the network, but also contribute to the network's structure and connectivity, and can

have biological, taxonomic, evolutionary, or functional importance. Topological clustering methods are the most popular techniques used to detect clusters in the networks. For example, the “walktrap” algorithm is a bottom-up approach that assumes short random walks of three to five steps through interconnected nodes are more likely to stay within a cluster due to the higher level of interconnectedness within clusters [14]. The Markov clustering (MCL) algorithm, on the other hand, is a stochastic approach that tries to simulate a flow within the network structure, strengthening the flow where nodes are highly interconnected and weakening it in other regions (van [15]). The inherent clustering structure of the network, which influences the flow process, will be eventually revealed when the flow process stabilizes. MCL has found great popularity in network biology and has been used to identify families within protein networks, detect orthologous groups, predict protein complexes from protein interaction networks, and find gene clusters based on expression profiles.

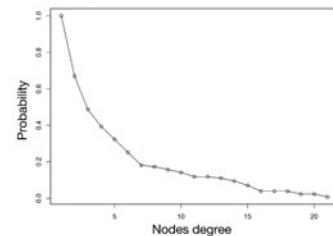
Two popular approaches have been used to detect keystone taxa from microbiome networks: centrality indices and link-analysis methods. Keystone taxa found by node centrality indices are expected to have more links, can reach all the other taxa more quickly, and control the flow between other taxa. Examples of node centrality indices are degree centrality [16], node- and edge-betweenness [16, 17], and closeness [16]. Link-analysis algorithms, on the other hand, are iterative and interactive data-analysis techniques used to evaluate connections between nodes. The PageRank algorithm [18] is a well-known link-analysis method that is based on the assumption that keystone taxa are likely to be more connected to other taxa when compared to non-keystone taxa.

Here, we present a comprehensive protocol for performing network analysis on microbiome data. We will begin with a brief description of how to obtain and set up R and RStudio software and the required packages, followed by explaining how to import and preprocess a microbiome OTU table. Next, we will explain how to build microbiome co-occurrence networks using several different approaches. Then, we will show how to infer clusters of coexisting or co-evolving microbes within the microbiome network before explaining how to find most central microbes (i.e., keystone taxa) within communities. Finally, we will describe how to simulate different types of network as well as how to estimate various network metrics, before discussing network visualization. A microbiome network with clusters, hubs, and other features highlighted is represented in Fig. 1a. More detailed foundational information and descriptions of the microbiome network methods used here are available in [12].

a) Network



b) Degree distribution



c) Nodes centrality

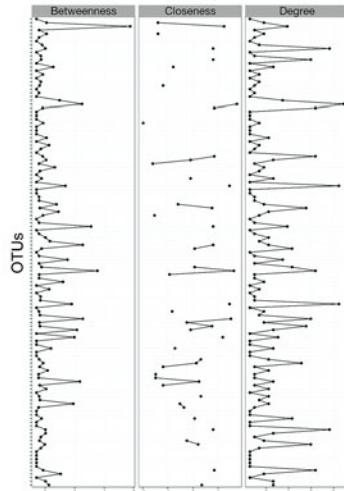


Fig. 1 A microbiome network and its properties. (a) the SparCC network constructed from American Gut microbiome project OTU table is shown here. Various network properties discussed in this method are highlighted in this figure. Clusters are shown with different colors, while the isolated nodes are shown in white. The size of the nodes corresponds to their betweenness centrality score. (b) The degree distribution of the same network is plotted using the code snippet provided in Subheading 3.9, step 10. (c) Centrality measures of the network are plotted using the *centralityPlot* function of qgraph package (Subheading 3.9, step 10)

2 Materials

2.1 OTU Table

Operational taxonomic unit (OTU) tables are usually in the form of a $n \times m$ matrix, where n is number of samples and m is the number of OTUs, or vice versa. OTU tables can usually be resolved at different taxonomic levels using microbiome-processing tools such as QIIME. Here, the OTU table from the American Gut project [19], which is included in the SpiecEasi package, will be used to construct microbiome networks. The American Gut data come from two sampling rounds (containing 304 and 254 human samples, respectively) and comprise several thousand OTUs. The data used here is a smaller subsample (containing 289 samples and 127 OTUs) created by Kurtz et al. [11] for demo purposes after several preprocessing and filtering steps. These steps are detailed here [11].

2.2 Obtaining R and RStudio

The method described here relies upon R, a free software environment for statistical computing and graphics, and RStudio, an integrated development environment (IDE) for R. RStudio includes a console and a syntax-highlighting editor that supports direct code execution, as well as tools for plotting, command history, debugging, and workspace management. R and RStudio are available for all major operating systems, and can be downloaded from <https://www.r-project.org/> and <https://www.rstudio.com/products/rstudio/>, respectively. In the code snippets throughout this chapter, lines starting with “#” are comments provided to make the code more legible to the user. Moreover, independent lines of code or commands are accommodated on single lines. Blocks of code (i.e., lines of code encapsulated within round or curly brackets), however, are naturally broken down into several lines for improved readability.

2.3 Obtaining Required Libraries

igraph and qgraph are two collections of open source and free network analysis tools that can be used in R to construct, simulate, analyze, and visualize networks. Moreover, the vegan package will be used to calculate pairwise (dis)similarity/distance between OTUs as well as to permute the OTU table. The MCL package is needed for cluster detection and SpiecEasi will be used to construct microbiome networks. These packages can be installed in R by copying the following commands into RStudio, or by entering the package names in the dropdown menu *Tools / Install Packages*.

```
# Install Required packages
install.packages("igraph")
install.packages("qgraph")
install.packages("vegan")
install.packages("MCL")
# Install SpiecEasi package
install.packages("devtools")
library(devtools)
install_github("zdk123/SpiecEasi")
```

Here, R version 3.4.2 was used on Ubuntu Linux (Ubuntu 17.10; Platform: x86_64-pc-linux-gnu) with SpiecEasi_0.1.3, devtools_1.13.4, MCL_1.0, vegan_2.4-5, qgraph_1.4.4, igraph_1.1.2, and BiocInstaller_1.28.0 as installed packages.

Depending on the operating-system setup and the software already installed on the system, some packages might not be installed due to lacking dependencies. In that case, we should call the *install.packages* function with an extra argument to make sure the dependencies are also installed.

```
install.packages("igraph", dependencies = TRUE)
```

All the packages discussed in this method come with manual or tutorials, which we encourage the users to read carefully for more information about the available methods, their applications and detailed real-life or toy examples. While typing a command in RStudio or R console, pressing the “tab” key will suggest options. This is especially useful, if user forgets the arguments that could be supplied to a function.

3 Methods

3.1 Load and Process OTU Table

1. First, we need to load the required packages into the R environment.

```
library(igraph)
library(qgraph)
library(vegan)
library(MCL)
library(SpecEasi)
```

2. We will use an OTU table from the American Gut project as an example dataset. First, load this file into R, and then reformat the row and column names to OTU_x and Sample_x, respectively, where x is the number of that row or column, which will help with readability. You can view the data in RStudio after loading the file and changing the row and column names by selecting the file name (amgut1.filt) in the **Data** window.

```
# Load OTU table
data("amgut1.filt")
# Change row and column names to a more readable format
colnames(amgut1.filt) <- sapply(1:ncol(amgut1.filt),
                                function(x) paste("OTU", x, sep = "_"))
rownames(amgut1.filt) <- sapply(1:nrow(amgut1.filt),
                                 function(x) paste("Sample", x, sep = "_"))
```

3. Alternatively, any user-provided OTU table can be loaded in R environment using the following command, given that the file includes header and row names and entries are separated by “,”. See **Note 1** to learn how to get help on built-in functions.

```
user.table <- read.table(infile,
                           header = T,
                           row.names = 1,
                           sep = ",")
```

4. Values in an OTU table are usually reported as absolute abundances. For some of the downstream processing, however, we will need to make samples comparable to each other by converting the absolute abundance values to relative abundances.

See Notes 2 and 3 for more information on how to deal with missing data and filter poor samples, respectively.

```
otu.relative <- amgut1.filt / rowSums(amgut1.filt)
```

3.2 Dissimilarity-Based Network

- To build a microbiome network using dissimilarity-based approaches, we will need to first compute a pairwise dissimilarity matrix from the OTU table. This can be performed by calling the *vegdist* function from vegan package using a variety of distance and dissimilarity indices, including “manhattan,” “euclidean,” “bray,” and “jaccard.” Here, we use the Bray–Curtis dissimilarity index, which is a statistic used to quantify the compositional dissimilarity between two different samples, based on the counts in each sample. The Bray–Curtis dissimilarity is bounded between 0 and 1, where 0 means identical composition of OTUs, and 1 means lack of any shared OTUs. Given that *vegdist* by default calculates distances between the rows while we want distances between the columns (i.e., OTUs), we need to transpose our input OTU table (*t(otu.relative)*; *see Note 4*).

```
# Create dissimilarity matrix
distances <- vegdist(t(otu.relative),
                      method = "bray")
```

- In order to build a network, the dissimilarity matrix needs to be converted to an adjacency matrix (via first converting the distance object to a matrix using *as.matrix* function). An adjacency matrix A is an $n \times n$ binary matrix (i.e., only containing 1s and 0s), where n is the number of OTUs and entry $A_{i,j}$ (i.e., entry at row i and column j) indicates a link between OTU $_i$ and OTU $_j$, only if the value at that entry is 1. This conversion can be performed by specifying a threshold. For example, the following code snippet will convert the dissimilarity matrix to an adjacency matrix with threshold set at 0.6, so that two OTUs with a dissimilarity index smaller than or equal to 0.6 will be connected to each other. *See Notes 5 and 6* to learn how to import and export networks. By setting mode and diagonal parameters (*mode* and *diag* below) to “undirected” and FALSE, respectively, we make sure to get an undirected network without loops.

```
# Convert distance object to a matrix
diss.mat <- as.matrix(distances)
diss.cutoff <- 0.6
diss.adj <- ifelse(diss.mat <= diss.cutoff, 1, 0)
# Construct microbiome network from adjacency matrix
diss.net <- graph.adjacency(diss.adj,
                             mode = "undirected",
                             diag = FALSE)
```

3.3 Correlation-Based Network

- An alternative approach to construct a microbiome network is to use pairwise correlation coefficients (i.e., Pearson, Kendall, or Spearman) calculated between the OTUs. The *cor* function, by default, computes the correlations between the columns of the input matrix.

```
cor.matrix <- cor(otu.relative, method = "pearson")
```

- Similar to dissimilarity-based methods, we need to convert the correlation matrix into an adjacency matrix. Unlike dissimilarity-based methods, however, we compare the absolute values to a threshold (0.3 in the example below), because we are interested in both positive and negative correlations.

```
# Convert correlation matrix to binary adjacency matrix
cor.cutoff <- 0.3
cor.adj <- ifelse(abs(cor.matrix) >= cor.cutoff, 1, 0)
# Construct microbiome network from adjacency matrix
cor.net <- graph.adjacency(cor.adj,
                             mode = "undirected",
                             diag = FALSE)
```

- In order to avoid including false positives (i.e., spurious correlations) in the network, one can permute the OTU table many times (100 times in the example below) and calculate a *p*-value for each possible pairwise interaction to test the validity of the detected interaction. A small *p*-value (e.g., ≤ 0.01 used below) indicates strong evidence against the null hypothesis that the detected interactions are spurious or random. The following function will take the OTU table (designated as MB within the function), correlation method, number of permutations and the desired significance level as input to generate the underlying microbiome network. After applying the Benjamini-Hochberg correction for multiple tests, it will then convert the corrected *p*-values to an adjacency matrix, from which it will construct the microbiome network. The permutation parameters can be adjusted by the user via changing the arguments of the *permfull* function.

```
# Execute this command after running Function 1
cor.net.2 <- build.cor.net(amgut1.filt,
                            method = 'pearson',
                            num_perms = 100,
                            sig_level = 0.01)

##### Function 1: Construct microbiome network using
# permutation
build.cor.net <- function(MB, method, num_perms, sig_level) {
  taxa <- dim(MB)[2]
  MB.mat <- array(0, dim = c(taxa, taxa, num_perms + 1))
  # Perform permutation
```

```

MBperm <- permatswap(MB, "quasiswap", times = num_perms)
# Convert to relative abundance
MB.relative <- MB / rowSums(MB)
MB.mat[, , 1] <- as.matrix(cor(MB.relative, method=method))
for(p in 2:num_perms) {
  MBperm.relative <- MBperm$perm[[p-1]]
  / rowSums(MBperm$perm[[p-1]])
  MB.mat[, , p] <- as.matrix(cor(MBperm.relative, method =
method))
}
# Get p-values
pvals <- sapply(1:taxa,
  function(i) sapply(1:taxa, function(j)
    sum(MB.mat[i, j, 1] > MB.mat[i, j, 2:num_perms])))
pvals <- pvals / num_perms
# p-value correction
pvals_BH <- array(p.adjust(pvals, method = "BH"),
  dim=c(nrow(pvals), ncol(pvals)))
# Build adjacency matrix
adj.mat <- ifelse(pvals_BH >= (1 - sig_level), 1, 0)
# Add names to rows & cols
rownames(adj.mat) <- colnames(MB)
colnames(adj.mat) <- colnames(MB)
# Build and return the network
graph <- graph.adjacency(adj.mat, mode = "undirected", diag =
FALSE)
}

```

3.4 Graphical Model Networks

1. We can use the *EBICglasso* function of qgraph to build a microbiome network by computing the underlying sparse Gaussian graphical model of our OTU table, using graphical lasso based on extended Bayesian information criterion (EBIC). We first compute a correlation or partial correlation matrix from the OTU table. The graphical lasso technique will then be used to find the graph with the best EBIC. Next, we will build the qgraph network and convert it to an igraph network (*see Note 7*).

```

# Compute (partial) correlations
ebic.cor <- cor_auto(amgut1.filt)
# Identify graph with the best EBIC
ebic.graph <- EBICglasso(ebic.cor, ncol(amgut1.filt), 0.5)
# Build the network
ebic.qgnet <- qgraph(ebic.graph, DoNotPlot = TRUE)
# Convert to igraph network
ebic.net <- as.igraph(ebic.qgnet, attributes = TRUE)

```

2. We can also use the *FDRnetwork* function of qgraph to find the OTU table's underlying graphical model using local false discovery rate. Similar to *EBICglasso*, we first compute a correlation or partial correlation matrix. Next, we build the graphical model using one of the three available methods: “lfdr” for the local false discovery rate, “pval” for the p-value, and “qval” for the q-value. Q-values are p-values that have been adjusted for the False Discovery Rate (FDR; the proportion of false positives expected to result from a test). For comparison, while a *p*-value ≤ 0.01 means that less than or equal to 1% of all tests will result in false positives, a *q*-value $\leq 1\%$ indicates that less than or equal to 1% of only significant results will lead to false positives. Finally, we will build the qgraph network and convert it to an igraph network.

```
# Compute (partial) correlations
fdr.cor <- cor_auto(amgut1.filt)
# Identify graphical model
fdr.graph <- FDRnetwork(fdr.cor, cutoff = 0.01, method =
"pval")
# Build the network
fdr.qgnet <- qgraph(fdr.graph, DoNotPlot = TRUE)
# Convert to igraph network
fdr.net <- as.igraph(fdr.qgnet, attributes = TRUE)
```

3.5 SparCC and SPIEC-EASI Networks

1. SparCC networks can be constructed by feeding the OTU table (with absolute abundance values) to the *sparcc* function of the SpiecEasi package followed by converting the correlation matrix to an adjacency matrix using a threshold (*sparcc.cutoff* $<- 0.3$ below).

```
# SparCC network
sparcc.matrix <- sparcc(amgut1.filt)
sparcc.cutoff <- 0.3
sparcc.adj <- ifelse(abs(sparcc.matrix$Cor) >= sparcc.
cutoff, 1, 0)
# Add OTU names to rows and columns
rownames(sparcc.adj) <- colnames(amgut1.filt)
colnames(sparcc.adj) <- colnames(amgut1.filt)
# Build network from adjacency
sparcc.net <- graph.adjacency(sparcc.adj,
                                mode = "undirected",
                                diag = FALSE)
```

2. SPIEC-EASI networks are constructed using the *spiec.easi* function of the SpiecEasi package. The resulting object contains a matrix called *refit*, which is a sparse adjacency matrix that can be directly used to build the microbiome network.

```

# SPIEC-EASI network
SpiecEasi.matrix <- spiec.easi(amgut1.filt,
                                method = 'glasso',
                                lambda.min.ratio = 1e-2,
                                nlambda = 20,
                                icov.select.params = list(rep.num=50))

# Add OTU names to rows and columns
rownames(SpiecEasi.matrix$refit) <- colnames(amgut1.filt)

# Build network from adjacency
SpiecEasi.net <- graph.adjacency(SpiecEasi.matrix$refit,
                                    mode = "undirected",
                                    diag = FALSE)

```

3.6 Hub Detection

1. Hubs are nodes in the network that have a significantly larger number of links compared to the other nodes in the network. A hub in a microbiome network can be considered as an equivalent to a keystone species in the microbial community. Using centrality indices (closeness and betweenness below) to find keystone species will output a vector containing values between 0 and 1 for every node in the network. Link-analysis methods (page_rank and hub_score below), on the other hand, will output an object in which there is a vector containing the values for the nodes.

```

# Use sparcc.net for the rest of the method
net <- sparcc.net

# Hub detection
net.cn <- closeness(net)
net.bn <- betweenness(net)
net.pr <- page_rank(net)$vector
net.hs <- hub_score(net)$vector

```

2. These centrality vectors can be sorted to select taxa with highest probability of being keystone species. In the following, nodes are sorted based on their *hub_score* measurements (*net.hs* below) and the top 5 ($n = 5$ below) are chosen.

```

# Sort the species based on hubbiness score
net.hs.sort <- sort(net.hs, decreasing = TRUE)
# Choose the top 5 keystone species
net.hs.top5 <- head(net.hs.sort, n = 5)

```

3.7 Cluster Detection

1. Two cluster-detection methods from the igraph package and one from the MCL package are used here. It should be noted that igraph methods output an object containing various information on the detected clusters, including but not limited to the membership of each cluster (*membership* function below).

While igraph methods work directly on the network object, MCL should be applied to the adjacency matrix (*adj* below). See Note 8 for more detail on alternative methods.

```
# Get clusters
wt <- walktrap.community(net)
ml <- multilevel.community(net)
# Get membership of walktrap clusters
membership(wt)
# Get clusters using MCL method
adj <- as_adjacency_matrix(net)
mc <- mcl(adj, addLoops = TRUE)
```

2. The clusters detected by various methods can be compared to each other using igraph's *compare* function. In addition, customized vectors of known or expected cluster memberships can be created in order to compare with the results of the clustering methods. Here, we divided the nodes into five clusters by random sampling (*sample* function below). This, however, could be replaced by a user-provided list. In case of identical clusters, the output will be 0. Identified clusters (*wt* below) can be plotted as a dendrogram.

```
# Compare clusters detected by different methods
compare(membership(wt), membership(ml))
compare(membership(wt), mc$Cluster)
# Create customized membership for comparison
expected.cls <- sample(1:5, vcount(net), replace = T) %>%
  as_membership
compare(expected.cls, membership(wt))
# Plot clusters as dendrogram
plot_dendrogram(wt)
```

3. One measure of the strength of division of a network into clusters or modules is network modularity. High modularity indicates that the network has dense connections within certain groups of nodes and sparse connections between these groups. The modularity of a graph with respect to a given membership vector can be used to estimate how separated different clusters of taxa are from each other.

```
# Calculate modularity
modularity(net, membership(wt))
```

3.8 Network Simulation

Network simulation is usually used for various comparative or analytical reasons. The code snippet below will generate regular, random, small-world, and scale-free networks, respectively. The variable *num.nodes* indicates the number of nodes in the simulated

network. k represents the degree of each node in the regular network. p stands for the probability of drawing an edge between two arbitrary nodes in the random network and the rewiring probability in the small-world network. dim and nei in the small-world network represent, respectively, the dimensions of the starting lattice and the neighborhood within which the node of the lattice will be connected. All these functions default to undirected networks. See Note 9 on how to simulate fictional OTU tables.

```
# Simulate networks
Num.nodes <- 50
regular.net <- k.regular.game(num.nodes, k = 4)
random.net <- erdos.renyi.game(num.nodes, p = 0.037)
smallworld.net <- sample_smallworld(dim = 1, num.nodes, nei =
2, p = 0.2)
scalefree.net <- barabasi.game(num.nodes)
```

3.9 Network Features

1. Basic features of a network, such as vectors of nodes or edges, names of the nodes, and number of nodes or edges can be extracted using the following commands.

```
# Network features
nodes <- V(net)
edges <- E(net)
node.names <- V(net)$name
num.nodes <- vcount(net)
num.edges <- ecount(net)
```

2. Transitivity, also known as the clustering coefficient, measures the probability that the adjacent nodes of a certain node are themselves connected. Using local type will generate a score for every node in the network, whereas using global type will produce one transitivity score for the whole network.

```
clustering_coeff <- transitivity(net, type = "global")
```

3. If we are interested in knowing which nodes are directly connected to any given node in the network, we can use the *neighbors* function. Moreover, we can see if two nodes share any neighboring nodes using the *intersection* function. OTU nodes 1 and 25 are labeled as 1 and 25 in Fig. 1a.

```
# Obtain the neighbors of nodes 1 and 25
otu1_neighbors <- neighbors(net, "OTU_1")
otu25_neighbors <- neighbors(net, "OTU_25")
# Find neighbors shared by nodes 1 and 25
intersection(otu1_neighbors, otu25_neighbors)
```

4. All the edges incident to one or multiple nodes (i.e., all edges connecting that node or nodes to other nodes) can be obtained using *incident* and *incident_edges* functions, respectively. The number of links connecting any given node to the network is that node's degree. Indegree of a node is the number of links ending at that node and outdegree is the number of links originating from the node. In undirected networks, indegree and outdegree are the same, hence the mode is set to "all."

```
# Edges incident to OTU_1
otu1.edges <- incident(net, "OTU_1", mode = "all")
# Edges incident to OTU_1 and OTU_25
otus.edges <- incident_edges(net, c("OTU_1", "OTU_25"),
mode = "all")
# Extracting/printing the incident edges separately
otus.edges$"OTU_1"
otus.edges$"OTU_25"
```

5. The average nearest neighbor degree (ANND) of a given node (or a set of nodes) can be calculated using the *knn* function. ANND is a measure of the dependencies between degrees of neighbor nodes. This allows us to test if the correlation between degrees of neighbor nodes is positive and the nodes of high degree have a preference to connect to other nodes of high degree or the correlation is negative and the nodes of high degree have a connection preference for nodes of low degree. The following code snippet calculates and prints the average nearest neighbor degree for all the nodes in the network (hence, *vids* = $V(\text{net})$).

```
net.knn <- knn(net, vids = V(net))
net.knn$knn
```

6. To find all nodes reachable, directly or indirectly, from a given node (e.g., OTU_1 below) we can use the *subcomponent* function which outputs a list of connections.

```
sub.node1 <- subcomponent(net, v = "OTU_1", mode = "all")
```

7. Isolated nodes that are not connected to any other node in the network can be removed as follows.

```
clean.net <- delete.vertices(net, which(degree(net, mode = "all") == 0))
```

8. Sometimes, a network is consisted of multiple disconnected components. These components can be obtained and printed as follows.

```
# Network components
net.comps <- components(net)
```

```

# Print components membership
net.comps$membership
# Print components sizes
net.comps$csize
# Print number of components
net.comps$no

```

9. Then, the largest or any other component can be used to induce (i.e., extract) a subnetwork from the microbiome network using *induced_subgraph* function. In fact, any set of nodes can be selected via R's standard subsetting techniques (shown below to extract components) and used to induce a subnetwork. All the methods applicable to a network can also be applied to the subnetworks.

```

# Largest component
largest.comp <- V(net) [which.max(net.comps$csize) == net.
comps$membership]
# Second component
second.comp <- V(net) [net.comps$membership == 2]
# The component containing OTU_1
otu1.comp <- V(net) [net.comps$membership ==
which(names(net.comps$membership) ==
"OTU_1")]
# Largest component subnetwork
largest.subnet <- induced_subgraph(net, largest.comp)
# Subnetwork for the component containing OTU_1
otu1.subnet <- induced_subgraph(net, otu1.comp)

```

10. The degree distribution of a network's nodes can be obtained and plotted to gain a better grasp of node connectivity (Fig. 1b). The *centralityPlot* function of the qgraph package can also be used to plot nodes' degree, closeness and betweenness measures for a network (or several networks), side by side, for comparison purposes (Fig. 1c). It should be noted that qgraph functions can be applied to networks produced by igraph without any modification. Here, the degree distribution is plotted as the cumulative sum of degrees (*cumulative = T*). To disable it, one should set the cumulative parameter to *False* (*cumulative = F*).

```

# Degrees
deg <- degree(net, mode = "all")
# Degree distribution
deg.dist <- degree_distribution(net, mode = "all", cumulative = T)
# Plot degree distribution
plot(deg.dist, xlab = "Nodes degree", ylab = "Probability")
lines(deg.dist)
# qgraph method
centralityPlot(net)

```

11. Real-life random or regular networks are rare. Small-world and scale-free networks, on the other hand, are quite common and, sometimes, it is important to know the type of the network at hand. The `fit_power_law` of `igraph` package tries to fit a power law function to the degree distribution and outputs a p-value, among other statistics, to indicate if the test rejects the hypothesis ($p\text{-value} < 0.05$) that network's degree distribution is drawn from the fitted power-law distribution. One notable member of scale-free family of network models is the hierarchical network model. Unlike other scale-free networks that predict an inverse relationship between the average clustering coefficient and the number of nodes, hierarchical networks show no relationship between the size of the network and its average clustering coefficient. Usually, the power-law behavior starts showing only above a threshold value, which if provided ($xmin = 10$, below), allows to fit only the tail of the distribution. The `smallworldness` function of `qgraph` package outputs a vector of statistics, first of which is a *smallworldness* score. This function measures the transitivity and the average shortest path length of the input network before computing the average of the same measures on a number of random networks (10 random networks here, $B = 10$). The small-worldness score is then computed as the transitivity of the input network over its average shortest path length after both being normalized by the same measures obtained from the random networks. If this score is higher than 1 (or higher than 3, to be more stringent), the network will be considered of small-world type.

```
# Scalefreeness: Fit a power_law to the network
deg <- degree(net, mode = "in")
pl <- fit_power_law(deg, xmin = 10)
pl$KS.p
# Smallworldness
sw <- smallworldness(net, B = 10)
sw[1]
```

12. The intersection and union of edges of two networks can be obtained, in order to compare two networks constructed by two different methods from the same data or two networks built from different time points. Here, we use the subnetwork containing OTU_1 (see above) as the second network.

```
intersect.edges <- intersection(net, otu1.subnet)
union.edges <- union(net, otu1.subnet)
```

13. A pairwise Jaccard similarity matrix can be calculated for some or all network nodes in order to know how similar some taxa

are in terms of their connection patterns in the network. Here, a similarity matrix is calculated for all the nodes.

```
node.similarity <- similarity(net, vids = V(net), mode = "all",
method = "jaccard")
```

14. Some nodes in the network are so central to the whole or a part of network that their removal will break the network into more components. These nodes are called articulation points or cut vertices and can be identified as follows.

```
# Find articulation points
AP <- articulation.points(net)
```

3.10 Network Visualization

1. The simplest way to visualize the network is to use igraph's built-in plot function with the network object as either the only parameter or together with the object containing the identified clusters (*wt* below). *See Note 10* on how to use igraph demo function.

```
# Simple plotting
plot(net)
plot(wt, net)
```

2. The visualization can be customized by feeding various parameters to the plot function. Here, for example, we customized nodes color, size, shape, frame color, label size, and label color. Similar customization can be applied to edge attributes as shown below with edge color. We also specified a layout for the network (Fig. 1a; **Note 11**).

```
# Customized plotting
plot(net,
      main = "Microbiome Network",
      vertex.color = "white",
      vertex.size = 12,
      vertex.shape = "circle",
      vertex.frame.color = "green",
      Vertex.label.size = 1,
      Vertex.label.color = "black",
      edge.color = "grey",
      layout = layout.fruchterman.reingold)
```

3. When the node names are long or there is a possibility of node labels overlapping due to the larger number of nodes, we can plot the network with node numbers and provide the node labels in the legend (*see Note 12* on how to handle large datasets). The following function, in addition to moving the node labels to the legend, will also scale the size of the nodes

based on their hub score (either calculated by one of the igraph hub detection methods or provided by user) and saves the plot as an image. The size or quality of the image can be customized by modifying the line that defines the image properties. This function takes four arguments; microbiome network, the hub scores, a name for the output file, and a title for the plot. Similar to previous section, node and edge attributes can be customized by changing the arguments fed to the *plot* function within Function 2.

```
# Function 2: Plot network with node size scaled to hubbiness
plot.net <- function(net, scores, outfile, title) {
  # Convert node label from names to numerical IDs.
  features <- V(net)$name
  col_ids <- seq(1, length(features))
  V(net)$name <- col_ids
  node.names <- features[V(net)]

  # Nodes' color.
  V(net)$color <- "white"

  # Define output image file.
  outfile <- paste(outfile, "jpg", sep = ".")
  # Image properties.
  jpeg(outfile, width = 4800, height = 9200, res = 300, quality = 100)
  par(oma = c(4, 1, 1, 1))

  # Main plot function.
  plot(net, vertex.size = (scores * 5) + 4, vertex.label.cex = 1)
  title(title, cex.main = 4)

  # Plot legend containing OTU names.
  labels = paste(as.character(V(net)), node.names, sep = " ")
  legend("bottom", legend = labels, xpd = TRUE, ncol = 5, cex = 1.2)
  dev.off()
}

# Execute this command after running Function 2
plot.net(net, net.hs, outfile = "network1", title = "My Network")
```

4. The next function will take two more arguments: membership of the detected clusters and articulation points (*cls* and *AP* below, respectively). The nodes will be colored based on their cluster membership and articulation points will be highlighted with a halo. Isolated nodes that are not member of any cluster will be colored white. We also added a custom layout using qgraph package in order to separate the overlapping nodes.

This layout can be further customized by changing the parameters fed into the pertinent function below (i.e., *qgraph.layout.fruchtermanreingold* function).

```
# Function 3: Plot network with clusters and node size scaled to
# hubbiness
plot.net.cls <- function(net, scores, cls, AP, outfile, title) {
  # Get size of clusters to find isolated nodes.
  cls_sizes <- sapply(groups(cls), length)
  # Randomly choosing node colors. Users can provide their own
  # vector of colors.
  colors <- sample(colours(), length(cls))
  # Nodes in clusters will be color coded. Isolated nodes will be
  # white.
  V(net)$color <- sapply(membership(cls),
    function(x) {ifelse(cls_sizes[x]>1,
      colors[x], "white")})

  # Convert node label from names to numerical IDs.
  node.names <- V(net)$name
  col_ids <- seq(1, length(node.names))
  V(net)$name <- col_ids

  # To draw a halo around articulation points.
  AP <- lapply(names(AP), function(x) x)
  marks <- lapply(1:length(AP), function(x) which(node.names ==
  AP[[x]]))

  # Define output image file.
  outfile <- paste(outfile, "jpg", sep=".")
  # Image properties.
  jpeg(outfile, width = 4800, height = 9200, res = 300, quality =
  100)
  par(oma = c(4, 1, 1, 1))

  # Customized layout to avoid nodes overlapping.
  e <- get.edgelist(net)
  class(e) <- "numeric"
  l <- qgraph.layout.fruchtermanreingold(e, vcount=vcount(net),

  area=8* (vcount(net)^2),

  repulse.rad=(vcount(net)^3.1))
  # Main plot function.
  plot(net, vertex.size = (scores*5)+4, vertex.label.cex=0.9,
  vertex.label.color = "black",
  mark.border="black",
  mark.groups = marks,
```

```

mark.col = "white",
mark.expand = 10,
mark.shape = 1,
layout=1)
title(title, cex.main=4)

# Plot legend containing OTU names.
labels = paste(as.character(V(net)), node.names, sep =
") ")
legend("bottom", legend = labels, xpd = TRUE, ncol =
5, cex = 1.2)
dev.off()
}

# Execute this command after running Function 3
plot.net.cls(net, net.hs, wt, AP,
outfile = "network2", title = "My Network")

```

4 Notes

1. The full description of a given function, the parameters it accepts, and the output it produces can be obtained by typing the question mark before the function's name. For example, to get more information on *walktrap.community* function, we can run the following command.

?walktrap.community

2. Many functions can handle missing values in data automatically. Yet, it is always a good practice to take care of missing values explicitly. For example, the following command removes all taxa with missing values, provided that the missing values are represented as NAs.

otu.table <- otu.table[, colSums(is.na(otu.table)) == 0]

3. Although optional, it is usually considered good practice to drop poor samples with fewer of observations. This line of code drops samples that have less than 1% of the observations of the largest sample in the OTU table.

otu.table <- otu.table[, colSums(abs(otu.table)) >
floor(max(colSums(otu.table)) / 100)]

4. Among R functions that accept matrices as input, some apply the operation in a column-wise fashion (i.e., *cor* function) while others work in a row-wise fashion (i.e., *vegdist* function). Therefore, it is critical to provide the input matrix in the correct

format by transposing the input accordingly, prior to or while feeding the input to the desired function.

```
otu.t <- t(otu)
```

5. In cases where user already has access to a microbiome network, it could be imported into R, if the file format is known. The correct file format should be selected from the list below (e.g., *format = "graphml"*).

```
imported_net <- read_graph(infile,
                           format = c("edgelist", "pajek",
                                     "ncol", "lgl", "graphml", "dimacs",
                                     "graphdb", "gml", "dl"))
```

6. Similarly, the microbiome network constructed in R can be exported into a file to be imported and used in other software. The desired file format can be selected from the list below (e.g., *format = "graphml"*).

```
write_graph(net, outfile,
            format = c("edgelist", "pajek", "ncol", "lgl",
                      "graphml", "dimacs", "gml", "dot",
                      "leda"))
```

7. Even though the qgraph network could be easily plotted or analyzed using available methods in the qgraph package, for the sake of consistency as well as the greater repertoire of analytical tools in igraph package, we recommend converting to an igraph network. It should be noted that sometimes qgraph to igraph object conversion might throw some warnings due to the differences between graph attributes that could be transferred between the two objects. The network topology, however, will be accurately transferred.
8. The igraph package comes with several built-in methods to detect clusters within microbiome networks including edge betweenness, fast greedy, leading eigenvectors, Louvain multi-level, spin-glass, and label propagation. Although these methods use various algorithms, their main function, which is cluster detection, remains the same. igraph's manual provides detailed information on how to use each of these methods.
9. Sometimes one needs to create and use a toy OTU table for testing purposes. The following code snippet produces an OTU table with 20 OTUs and 50 samples with abundance values randomly chosen from between 1 and 100, before row and column names are assigned.

```
# Create a simulated OTU table followed by adding row and
# column names
otu.table <- matrix(sample(1:100, 1000, replace = TRUE),
                      nrow = 20,
                      ncol = 50)
rownames(otu.table) <- paste("OTU", 1:nrow(otu.table), sep =
= " ")
colnames(otu.table) <- paste("Sample", 1:ncol(otu.table),
sep = " ")
```

10. igraph comes with a demo function that can be used to get a solid grasp of methods available in the package (e.g., *centrality* function below). The *igraph_demo* function can be called without any arguments to see what demos are available, before choosing one of the available demos. Moreover, this function can be run interactively.

```
# Get available demos
igraph_demo()
igraph_demo("centrality")
# Run interactively
if (interactive()) {
  igraph_demo("centrality")
}
```

11. Every time we plot a network, the positions of the nodes are recalculated, even when using the same layout. Therefore, to have a more visually stable and comparable representation, we should either use a fixed set of coordinates for all the nodes or choose a desired layout and assign it as a fixed attribute of the network. The fixed set of coordinates should be in the form of a $n \times m$ matrix, where n is the number of nodes and m is the x and y coordinates for each node.

```
# Assign custom (random) coordinates to layout
net$layout <- array(1:40, dim = c(20, 2))
# Assign a layout as a fixed attribute of the network
net$layout <- layout.fruchterman.reingold(net)
```

12. When working with large datasets containing thousands of OTUs, we can convert the adjacency matrix to a sparse matrix using *Matrix* function for a faster and more memory-efficient analysis.

```
sparse.adj <- Matrix(adj, sparse=TRUE)
```

References

1. Magalhaes AP, Azevedo NF, Pereira MO, Lopes SP (2016) The cystic fibrosis microbiome in an ecological perspective and its impact in antibiotic therapy. *Appl Microbiol Biotechnol* 100:1163–1181. <https://doi.org/10.1007/s00253-015-7177-x>
2. Dalton T, Dowd SE, Wolcott RD, Sun Y, Watters C, Griswold JA, Rumbaugh KP (2011) An *in vivo* polymicrobial biofilm wound infection model to study interspecies interactions. *PLoS One* 6:e27317. <https://doi.org/10.1371/journal.pone.0027317>
3. Murray JL, Connell JL, Stacy A, Turner KH, Whiteley M (2014) Mechanisms of synergy in polymicrobial infections. *J Microbiol* 52:188–199. <https://doi.org/10.1007/s12275-014-4067-3>
4. Legendre P, Legendre L (2012) Numerical ecology. Developments in environmental modelling, vol 24, 3rd English ed. Elsevier, Amsterdam
5. Barabási AL, Oltvai ZN (2004) Network biology: understanding the cell's functional organization. *Nat Rev Genet* 5:101–113. <https://doi.org/10.1038/nrg1272>
6. P. Erdős AR On the evolution of random graphs. In: Publication of the Mathematical Institute of the Hungarian Academy of Sciences; 1960
7. Barabasi AL, Albert R (1999) Emergence of scaling in random networks. *Science* 286:509–512
8. Watts DJ, Strogatz SH (1998) Collective dynamics of 'small-world' networks. *Nature* 393:440–442. <https://doi.org/10.1038/30918>
9. Chen EZ, Li H (2016) A two-part mixed-effects model for analyzing longitudinal microbiome compositional data. *Bioinformatics* (Oxford, England) 32:2611–2617. <https://doi.org/10.1093/bioinformatics/btw308>
10. Friedman J, Alm EJ (2012) Inferring correlation networks from genomic survey data. *PLoS Comput Biol* 8:e1002687. <https://doi.org/10.1371/journal.pcbi.1002687>
11. Kurtz ZD, Muller CL, Miraldi ER, Littman DR, Blaser MJ, Bonneau RA (2015) Sparse and compositionally robust inference of microbial ecological networks. *PLoS Comput Biol* 11:e1004226. <https://doi.org/10.1371/journal.pcbi.1004226>
12. Layeghifard M, Hwang DM, Guttman DS (2017) Disentangling Interactions in the Microbiome: A Network Perspective. *Trends Microbiol* 25:217–228. <https://doi.org/10.1016/j.tim.2016.11.008>
13. Tibshirani R (2018) Regression shrinkage and selection via the lasso: a retrospective. *J Roy Stat Soc Ser B (Stat Method)* 73:273–282. <https://doi.org/10.1111/j.1467-9868.2011.00771.x>
14. Pons P, Latapy M (2018) Computing communities in large networks using random walks. In: Yolum GT, Gürgen F, Özturan C (eds) Computer and information sciences—ISCIS 2005. Lect notes comput sci, vol 3733. SpringerLink, Berlin, Heidelberg, pp 284–293. https://doi.org/10.1007/11569596_31
15. Dongen SV (2008) Graph Clustering Via a Discrete Uncoupling Process. *SIAM J Matrix Anal Appl* 30:121–141. <https://doi.org/10.1137/040608635>
16. Freeman LC (1978) Centrality in social networks conceptual clarification. *Soc Netw* 1:215–239. [https://doi.org/10.1016/0378-8733\(78\)90021-7](https://doi.org/10.1016/0378-8733(78)90021-7)
17. Brandes U (2001) A faster algorithm for betweenness centrality. *J Math Sociol* 25:163–177. <https://doi.org/10.1080/0022250X.2001.9990249>
18. Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. In: Comput netw ISDN syst, vol 1–7. Elsevier Science Publishers, Brisbane, Australia, pp 107–117. [https://doi.org/10.1016/S0169-7552\(98\)00110-X](https://doi.org/10.1016/S0169-7552(98)00110-X)
19. McDonald D, Birmingham A, Knight R (2015) Context and the human microbiome. *Microbiome* 3:52. <https://doi.org/10.1186/s40168-015-0117-2>



Chapter 17

Bayesian Inference of Microbial Community Structure from Metagenomic Data Using BioMiCo

Katherine A. Dunn, Katelyn Andrews, Rana O. Bashwih,
and Joseph P. Bielawski

Abstract

Microbial samples taken from an environment often represent mixtures of communities, where each community is composed of overlapping assemblages of species. Such data represent a serious analytical challenge, as the community structures will be present as complex mixtures, there will be very large numbers of component species, and the species abundance will often be sparse over samples. The structure and complexity of these samples will vary according to both biotic and abiotic factors, and classical methods of data analysis will have a limited value in this setting. A novel Bayesian modeling framework, called BioMiCo, was developed to meet this challenge. BioMiCo takes abundance data derived from environmental DNA, and models each sample by a two-level mixture, where environmental OTUs contribute community structures, and those structures are related to the known biotic and abiotic features of each sample. The model is constrained by Dirichlet priors, which induces compact structures, minimizes variance, and maximizes model interpretability. BioMiCo is trained on a portion of the data, and once trained a BioMiCo model can be employed to make predictions about the features of new samples. This chapter provides a set of protocols that illustrate the application of BioMiCo to real inference problems. Each protocol is designed around the analysis of a real dataset, which was carefully chosen to illustrate specific aspects of real data analysis. With these protocols, users of BioMiCo will be able to undertake basic research into the properties of complex microbial systems, as well as develop predictive models for applied microbiomics.

Key words Microbial community structure, Hierarchical Bayesian model, MCMC, OTU abundance, Supervised learning, Cross-validation, Predictive model, Prediction error, Microbiome

1 Introduction

High-throughput sequencing technology allows us to investigate the composition of microbial communities that are often comprised of large numbers of uncultivated species. In addition to potentially high diversity, microbial communities can have highly complex structures [1, 2]. Such communities are often organized into ecologically coherent assemblies of species (i.e., individuals that tend to co-occur, and putatively interact), with different microbial assemblages having unique spatial and temporal variability that is a

function of both biotic and abiotic interactions [3–6]. However, the species and strains typically associated with one assemblage can interact with individuals from a different assemblage; thus, rather than being discrete entities, assemblages represent central tendencies in the association between individuals and their response to abiotic factors [7, 8]. The sensitivity of microbial abundance to both biotic and abiotic factors means that species and strains often have highly sparse distributions among community samples (i.e., high abundance in a few samples, and low or absent from most others). This poses a serious analytical challenge. Classical testing methods have limited capacity to resolve complex features within data where the number of variables is huge (hundreds, and sometimes thousands of species or strains), the interactions are complex, and abundance information is sparse [9–11]. A novel analytical framework, called BioMiCo [12], was developed for learning the structure of microbial communities from samples of environmental DNA sequence data, and for learning how mixtures of multiple assemblages are related to known features of each sample.

BioMiCo is a hierarchical mixed-membership modeling framework. It takes as input abundance information for either taxonomic units (e.g., 16S derived OTUs) or functional units (e.g., metagenome-derived KEGG Orthology classes), and it can be applied to either cross-sectional or serially sampled data. Under BioMiCo, an environmental DNA sample is viewed as being comprised of one, or more, communities corresponding to one or more known features of the sample’s environment (feature indexed by K). The structure of the sample is then modeled as a hierarchical mixture of multinomial distributions, where each community is comprised of a unique mixture of L different assemblages, and each assemblage is comprised of a unique mixture of T different OTUs. The hierarchical structure of the model allows “sharing” of OTUs between assemblages, and “sharing” of the assemblages between samples; thus, all samples are used to resolve the relationship between the known features of the data and the different components of community structure. BioMiCo is a supervised Bayesian method, using a “training dataset” to learn posterior mixture of (a) OTUs within assemblages, and (b) assemblages within communities that correspond to the known features of each sample. Thus a trained model will have “learned” how to explain and differentiate a set of K different feature labels according to microbial community structure, and can be used to predict the feature labels of “future” samples with an appropriate probability. Features can be generalized factors (e.g., location, season of year, health status) or a specific attribute (e.g., in the case of a human microbiome, the identity of the human host). “Test data” are any data with known features that were not used in the training of the model, and reserved for use in a procedure called cross-

validation [9]. Such test data are used to quantify the accuracy of a trained model to correctly predict features that are withheld from the model.

In this chapter we describe how to download and install BioMiCo. We then present three protocols for using the BioMiCo modeling framework for real data analysis. The first illustrates the basic process of training a BioMiCo model on one set of data, and then using that trained model to make predictions for an independent set of data. The second illustrates how to use the technique of cross-validation to assess the accuracy of BioMiCo-based predictions. The third illustrates how users can identify and set parameter values for running the MCMC algorithm that yield stable estimates of posterior probabilities. By training on carefully chosen biotic and abiotic features users of BioMiCo will have the capacity to address basic questions about the properties of complex microbial systems [13] as well as develop predictive models for applied microbiomics [14].

2 Materials: Software Download and Installation

This section describes how to obtain BioMiCo, and provides an installation procedure for OS X, FreeBSD, GNU/Linux, and other Unix-like operating systems. BioMiCo can be installed on Windows; however, all protocols presented in this chapter are intended for Unix-like operating systems (including OS X).

2.1 Prerequisites

BioMiCo is an R package; therefore, R must be installed in order to run the installation of BioMiCo. The Rcpp package, Boost C++ libraries and the GNU scientific library (GSL), must also be installed on your system, as BioMiCo depends on these. Perl scripts are used to automate the analytical steps required for cross-validation. BioMiCo prerequisites are open source and available for free download.

```
R package: https://cran.r-project.org/
Rcpp package: http://www.rcpp.org/
Boost: http://www.boost.org/
GNU Scientific Library: https://www.gnu.org/software/gsl/
Perl: https://www.perl.org/
```

The details of the installation may differ depending on your system. Although not required, we recommend that Mac users download MacPorts (<https://www.macports.org/>), and use it to compile, install and update R, Boost and GSL. Mac users may need to install Xcode before installing GSL or Boost. Xcode is a suite of development tools available free of charge from the Mac App Store.

For all systems, make a note of where the boost directory is located (typically /opt/local/include), as the location is required by the installation of BioMiCo.

2.2 Installation of BioMiCo

1. Download the latest version of the BioMiCo package and unpack the archive to a local folder.

BioMiCo archive: <https://sourceforge.net/projects/biomico/files>

2. Navigate within your file system to the local folder that contains the BioMiCo archive, and familiarize yourself with the files and the directory structure. The archive includes source code, R scripts, example data files, and a user guide. Check the user guide for a list of the files and directories that should be included within the package, and verify that they have been unpacked to the local folder.
3. If your boost directory is not located in /opt/local/include you will need to edit the Makevars file. The Makevars file is located in the src directory within the BioMiCo_R_Package directory. Edit the line in the Makevars file that starts with PKG_CPPFLAGS and change the default path within that line (/opt/local/include) to indicate the location of the boost directory on your system.
4. Navigate within your file system to the BioMiCo directory within the BioMiCo_R_Package directory. BioMiCo is built from this location. To build and install BioMiCo, type the following at the command line:

```
R CMD build BioMiCo_R_Package  
R CMD INSTALL BioMiCo_x.x.x.tar.gz
```

These commands are case sensitive, and the x's should be replaced with the current version number.

5. Open and familiarize yourself with the formats of the three types of data files that BioMiCo takes as input. Example input files (train.ix, env, and test.ix) are provided with the BioMiCo archive.

train.ix: This file contains the OTU counts (decimal are not allowed) for the samples you want to use to learn how OTUs contribute to assemblages, and how assemblages correspond to the known features of each sample. The contents of this file are structured as a matrix. The first row contains unique OTU labels (*see Note 1*), and the first column contains unique sample IDs. The remaining columns contain the counts of unique OTUs within each sample.

env: This file contains unique labels for the known features of each sample. The content of this file is comprised of at least two columns. The first column lists the unique sample IDs

(these must exactly match those within `train.ix`, and occur in the same order), and the second column gives the unique feature labels for each sample.

`test.ix`: This file contains the OTU counts for those samples that you will use in BioMiCo to predict the features of a sample according to its microbial community structure.

3 Methods

BioMiCo is a Bayesian framework, and inferences are based upon statistical estimates of posterior-probability distributions. This is achieved through a Markov Chain Monte Carlo (MCMC) algorithm, which is the basis of each of the following analytical protocols. Further description of Bayesian inference, and details of the MCMC algorithm employed by BioMiCo, are available in Shafiei et al. [12], its supporting information, and the publications cited therein.

Input files required to run all protocols are included in the protocol folders within the BioMiCo archive.

3.1 Training and Testing on Independent Data

In the following protocol we use a time series for coastal ocean bacterial communities to illustrate how to apply a trained predictive model to independent data. A BioMiCo model is first trained to resolve how pelagic bacterial OTUs (derived from 16S data) contribute to assemblages, and how community structure differs according to season. The training data represent four time points (spring equinox, summer solstice, autumn equinox, winter solstice) collected over six-years from a coastal inlet in the temperate northwest Atlantic Ocean [3]. The OTU abundance training data are provided in the `train.ix` file in the BioMiCo archive. For this analysis the known feature of each sample is the season of the year, and the feature labels are defined in `env` file provided in the BioMiCo archive. Training on these two files leads to a model that can distinguish seasons according to their characteristic community structures.

The ultimate objective of the analysis is to apply the trained model to an independent dataset (i.e., a test dataset) to investigate the seasonal process of succession within the pelagic community over a full annual cycle. The test data in this case are comprised of 25 independent samples collected bi-weekly over a single year. Those data are provided in the `test.ix` file in the BioMiCo archive. The trained model is applied to the test data to estimate mixing probabilities for assemblages at each of the 25 time points. The idea is that the composition and timing of seasonal shifts in community structures will be evident in the temporal change in the mixture of assemblages. Graphical analyses included in this

protocol are used to visualize the pattern and timing of the seasonal cycle of community succession.

1. Inspect the contents of your three input files and verify that they contain the correct data, and that the contents are correctly formatted. For this exercise use the three input files provided within the BioMiCo archive for protocol 1 (`train.ix`, `env`, and `test.ix`).
2. Use a plain text editor to open and configure the R script called `trainbyalltestbyone.R`. Program variables within this file are used to set the name and location of your data files, and to specify parameters associated with running the MCMC algorithm. Those program variables that are commonly adjusted as part of a data analysis are summarized below.
3. Use `setwd` to specify the location of your input files when you are not running your BioMiCo analysis within the directory that contains those files. If running from within the directory place a # in front of this line.
4. Use `TRAIN.Mat=`: to set several program variables associated with your training dataset.

Use the `file=` variable to set the filename and location of the training data. For this analysis use the file provided in protocol 1 and set `file=train.ix`.

Use `header=TRUE` to indicate that the first row of your training data provides the name of the OTUs.

5. Use `ENVS=`: to set several program variables associated with the unique feature labels for your samples.

Use the `file=` to set the filename and location of the feature labels. For this analysis use the file provided in protocol 1 folder and set `file=env`.

Use `header=TRUE` when the file contains a first line that identifies the columns

6. Use `TEST.test=`: to set several program variables associated with your test dataset.

Use `file=` to set the filename and location of your test data. For this analysis use the file provided in protocol 1 folder and set `file=test.ix`.

As the OTUs must be the same in the training and test datasets, the labels are not required but the order of OTUs must be the same as in the training data file. Use `header=TRUE` if you include the OTU labels on the first line otherwise you can exclude the OTU labels within that file and denote `header=FALSE`.

7. Use `source:` to specify the location of your `BioMiCo-Scripts.R` file supplied in the BioMiCo download folder.

8. Use `num.communities=`: to set the maximum number of assemblages (L). It is acceptable to use the default value of 25 because most analyses require fewer than 25, and a sparse symmetric Dirichlet prior is employed by BioMiCo to minimize the number of assemblages used to explain the data and thereby improve model interpretability.
9. Use `train.results=`: to set several parameters associated with analysis of the training data.

`burnin=`: Use this to set the number of initial MCMC iterations to discard from the analysis. For this analysis use `burnin=1000`.

`nrestarts=`: Use this to permit multiple MCMC runs. While we recommend this, using this option will lead to data structures that can be confusing. For this analysis use the default `nrestarts=1`.

Use `ndraws.per.restart=` to set the total number of samples from the MCMC run that will be retained for analysis. For this analysis set `ndraws.per.restart=20`.

Use `delay=` to set the number of iterations of the MCMC run between those samples that will be retained for analysis. For this analysis set `delay= 1000` (*see Note 2*).

10. Use `test.results=`: to set several parameters associated with the analysis of the test data.

```
Set burnin=100
Set nrestarts=1
Set ndraws.per.restart=20
Set delay=50
```

11. Run the R script called `trainbyalltestbyone.R` by typing the following at the command line:

```
R CMD BATCH trainbyalltestbyone.R
```

This script runs the analyses in two phases. In the first phase, the model is “trained” to distinguish feature labels provided as input within `env` (seasons) according to signature patterns of community structure. The second phase is the “test phase,” whereby the trained model is used to estimate the posterior contribution of the season-specific communities to each sample. These posterior probabilities are interpreted as “membership weights” for each sample, with the seasonal transitions in community structures inferred from the temporal changes in the membership weights. All this information is contained within two R binary images: one for the training phase and one for the testing phase.

12. Run the R script called `train_analysis.R` from within the folder containing the `.RData` files generated above to analyze

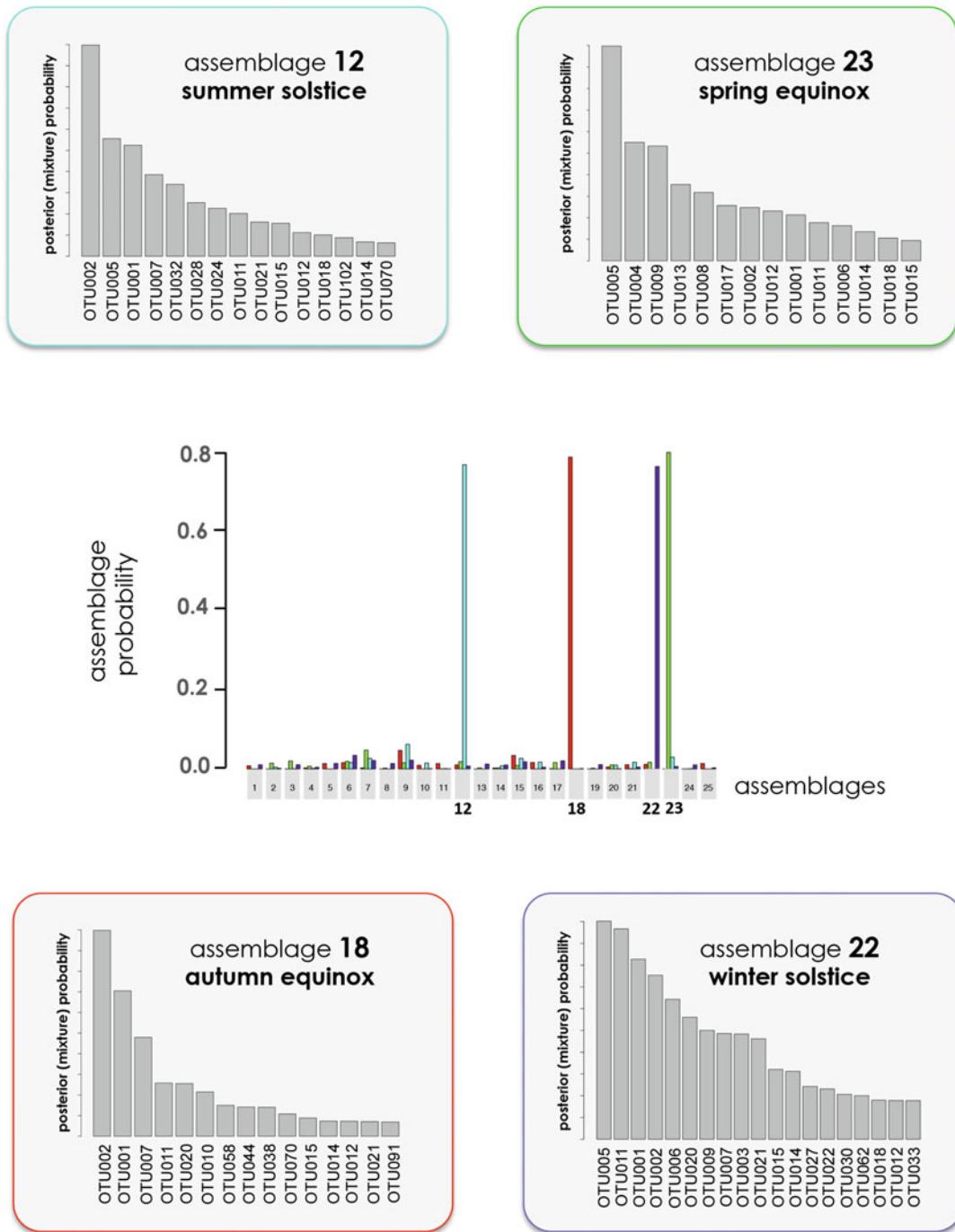


Fig. 1 Predominant OTUs of the assemblage structures associated with the spring equinox, summer solstice, autumn equinox, and winter solstice samples. Training samples were collected at a coastal ocean site in the Northwest Atlantic (Bedford Basin, NS, Canada) during the spring equinox, summer solstice, autumn equinox, and winter solstice from 2005 to 2010. The central plot shows the posterior probabilities (PP) of each assemblage within the model with respect to time points (spring equinox, summer solstice, autumn equinox, and winter solstice). Although the model permits up to 25 assemblages, the posterior probability distribution supports just 4 major structures; one for each of the 4 seasonal sampling points. Assemblage No. 12 has PP of

the *training data*. Run the script by typing the following at the command line:

```
R CMD BATCH train_analysis.R
```

This will generate three text files (`assemblage_pp.txt`, `OTU_pp.txt`, and `OTUs_from_dominant_assemblages.txt`). The first two files contain the posterior probability (PP) of each assemblage in each feature label (season), and the PP of each OTU in each assemblage. The last file provides information about the distribution of the predominant OTUs (see Note 3) within each assemblage. In addition to these text files, this script also generates a number of plots. The file `assemblage_plot.pdf` is informative about community structure, as it presents the PP (membership weight) of each assemblage relative to the features of interest. Plots are also produced showing how the predominant OTUs ($OTU\ PP > 0.01$) contribute to the predominant assembles (assemblage PP > 0.1).

13. Inspect the graphical output produced in step 12 above. An example plot of how OTUs contribute to assemblages, and how assemblages are related to feature labels is provided in Fig. 1. After reviewing Fig. 1, inspect the plots that you produced in step 4 with the objective of learning how the model-based estimates of posterior probabilities (mixture weights) are interpreted in terms of community composition.
14. Run the R script called `test_analysis.R` from within the folder containing the `.RData` files generated above to analyze the *test data*. Run the script by typing the following at the command line:

```
R CMD BATCH test_analysis.R
```

This creates a text file (`predictions.txt`) containing the membership weight of each feature for a given sample. These results are plotted within the file `prediction_plots.png`. Note that when the objective is to classify unknown samples according to community structure, each sample within the test dataset is then classified according to the feature label having the maximum posterior probability.

Fig. 1 (continued) 77% for the summer solstice (turquoise). Assemblage No. 23 has PP of 80% for the spring equinox (green). Assemblage No. 18 has PP of 78% for the autumn equinox (red). Assemblage No. 22 has PP of 76% for the winter solstice (purple). Note the assemblage numbering system used by BioMiCo is arbitrary, and the assemblage numbers will differ from run to run. The corner plots show the posterior distribution of the predominant OTUs ($PP > 0.01$) within each assemblage. If the MCMC has reached its stationary distribution, the PPs can be interpreted as estimates of the mixture weights of OTUs within microbial assemblages, and how assemblages are mixed to explain the full community structure of a given sample

15. Inspect the graphical output produced in **step 14** above. Compare your output to Fig. 2 to understand (a) how posterior probabilities are interpreted as assemblage mixture weights in order to explain the full community structure of independent samples and (b) how temporal data can be used to infer community transitions and relate those transitions to independent biotic and abiotic variables.

3.2 Cross-Validation to Assess Predictive Accuracy

The training phase of the analysis used in Subheading 3.1 above produced a model that was subsequently applied to an independent dataset to infer seasonal shifts in community composition. This was possible because the trained model was able to *predict* the relative contribution of different seasonal assemblages even though the model was provided no information about when the subsequent samples had been collected. Thus the trained model can be viewed as a *predictive model*, with the reliability of the inferences being a function of the *predictive accuracy* of the trained model. The following protocol employs a technique called cross-validation to formally assess the predictive accuracy of a trained BioMiCo model. Note that this protocol assumes that the user is already familiar with basic principles of analyses covered in Subheading 3.1, and this protocol uses the same training dataset. Users are strongly encouraged to master protocol 1 before moving on to this protocol.

Cross-validation is a two-phase analysis that is intended to “mimic” the two-phase inferential procedure employed in Subheading 3.1 above. Cross-validation differs from the above procedure in that (a) the training data with known feature labels (in this example, six years of equinox and solstice samples) are divided into two subsets, with replication, (b) the data are trained on only one subset of the labeled samples, and (c) the remaining samples are input to BioMiCo with the labels “hidden” and the trained model is used to predict the hidden labels. The frequency of correct prediction of the hidden labels, taken over all replicates, provides a measure of the predictive accuracy of the model. The following protocol illustrates two alternative strategies: leave-one-out cross-validation, and cross-validation of random partitions into 2/3 training and 1/3 testing subsets of the data. The former is recommended for datasets having small numbers of samples and the later for larger datasets.

3.2.1 Leave-One-Out Cross-Validation

Leave-one out cross-validation divides the set of n different labeled samples into all possible subsets having just one sample “left out” of the training subset. The objective is to train on all possible subsets for $n-1$ data, and use each of the obtained trained models to predict the label for the one sample that was left out of the training phase. In this way, a prediction is obtained for each of the n different labeled samples that are available for training (*see Note 4*). The

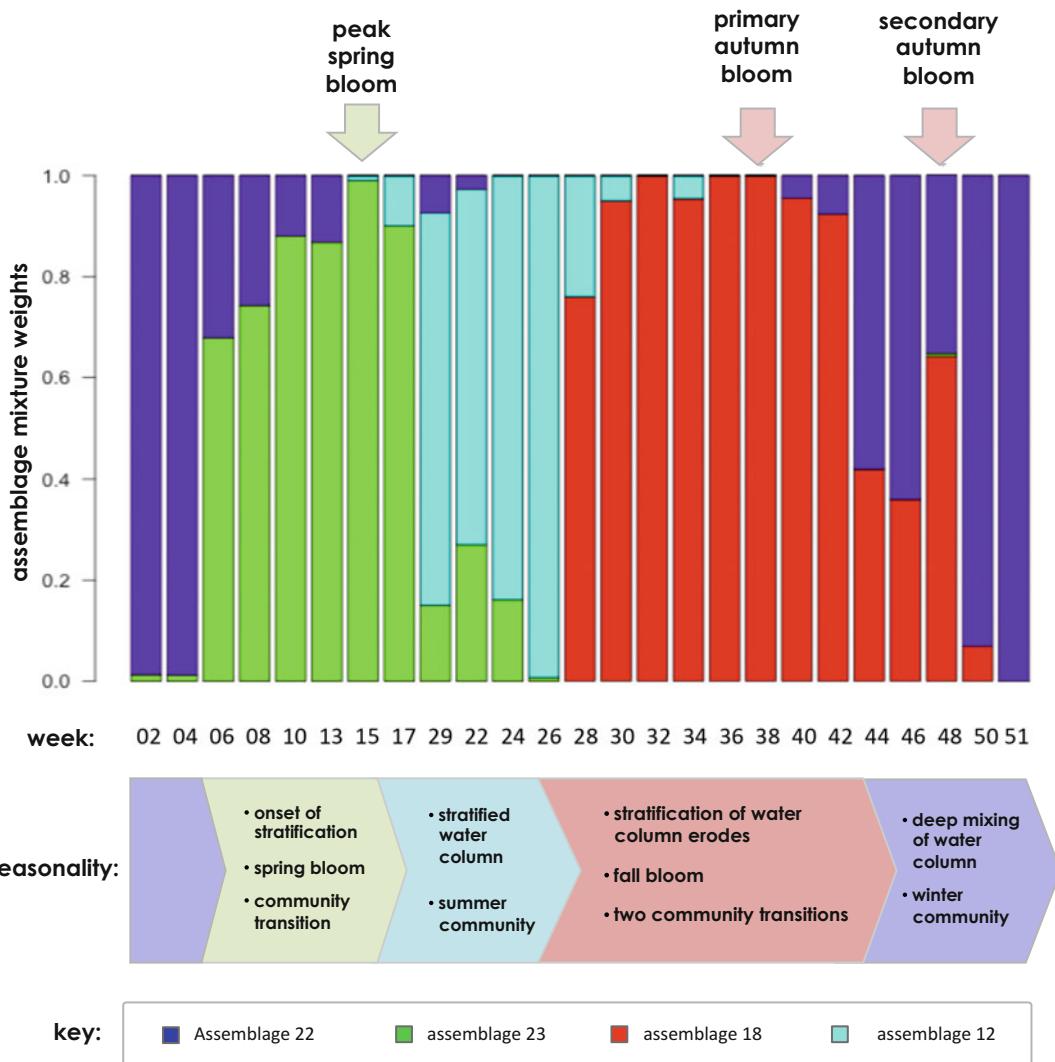


Fig. 2 Seasonal transitions of bacterioplankton community assemblages. Mixtures of seasonal bacterioplankton assemblages (spring equinox, summer solstice, autumn equinox, and winter solstice) are inferred for each of 25 bi-weekly water column samples at a coastal Northwest Atlantic Ocean site (Bedford Basin, N.S., Canada). The community structure at each time point is inferred by treating the posterior probabilities as mixture weights. The BioMiCo model was trained on only the spring equinox, summer solstice, autumn equinox and winter solstice samples from 2005 to 2010, and the plot shows the inferred mixtures of communities for an independent set of samples collected from January to December of 2009. As expected, the equinox and solstice samples from 2009 are dominated by the community structures learned from the training dataset. Mixtures of these structures at the other time points in 2009 reflect transitions in community structures, and these temporal dynamics closely match the biotic (e.g., timing and composition of phytoplankton blooms) and abiotic (e.g., changes in temperature and stratification of the water column) transitions that were independently observed at this sampling site. Particularly noteworthy is the secondary autumn bloom, which was an event observed at week 48 involving profound changes in both biotic and abiotic variables.

following protocol uses a single Perl script to create from the single matrix of training data (`train.ix`) and their associated labels (`env`) all the input files required for leave-one-out cross-validation. The same Perl script can be used to further automate the serial analysis of those subsets of data.

1. Inspect the contents of the input files `train.ix` and `env` provided in the archive for protocol 2.1, and verify that they contain the correct data, and that the contents are correctly formatted (*see Note 5*).
2. Make a note of the total number of labeled samples (n). For the matrix of training data (`train.ix`) used here as an example, $n = 24$.
3. Create a new analytical directory for your cross-validation analyses, and copy the data files `train.ix` and `env`, and the Perl scripts `sample_leavelout_runner.pl` and `batchRfile-maker.pl` to that analytical directory.
4. Save a text-based record of the absolute path to the analytical directory created in **step 3** above. In the series of commands below `<path1>` is used to denote this path in your file system.
5. Save a text-based record of the absolute path to the directory that contains `BioMiCoScripts.R`. In the series of commands below `<path2>` is used to denote this path in your file system.
6. Decide if you want to run the leave-one-out analyses in series or in parallel. The example data for this protocol is small enough to run in series, and we recommend you start by using the supplied Perl script to run the entire analysis in series (*see Note 6*).

To run in series: Navigate within your file system to your analytic directory and type the following at the command line:

```
perl sample_leavelout_runner.pl <path1> 24 <path2> yes
```

To run in parallel: Navigate within your file system to your analytic directory and typing the following at the command line:

```
perl sample_leavelout_runner.pl <path1> 24 <path2> no
```

List the contents of your analytical directly using the `ls` command and note the numbered subdirectories that were created by running the analysis in parallel. For the example data there should be $n = 24$ subdirectories, one for each leave-one-out analysis.

Navigate to each of the numbered subdirectories and type the following at the command line (*see Note 7*):

```
R CMD BATCH sample_leaveloutrun.r
```

7. To automatically extract and summarize results (*see Note 8*) obtained by running analyses in series, or in parallel, you

navigate within your file system to your analytical directory and type the following at the command line:

```
perl batchRfilemaker.pl <path1> 24
```

This will create a file called `batchRfile.r`. This file can be used to analyze all of the runs by typing the following at the command line:

```
R CMD BATCH batchRfile.r
```

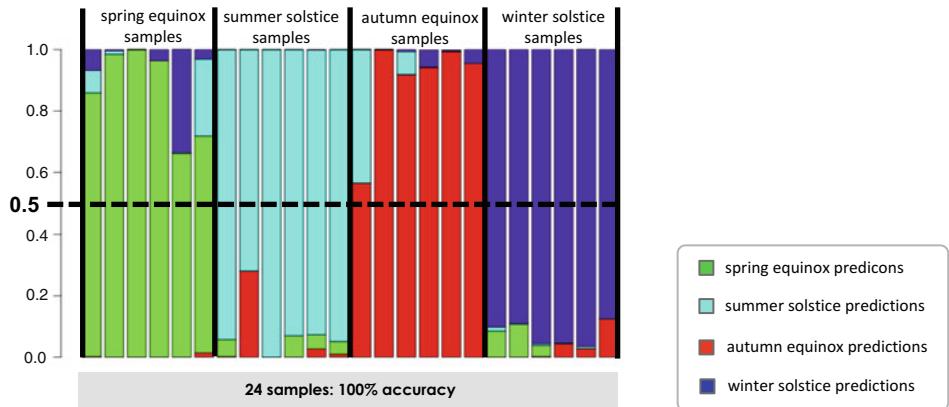
8. Inspect graphical output produced in **step 7** above. Here, the relationship of the contribution of OTUs to assemblages, and assemblages to feature labels, was inferred for each of the 24 leave-one-out analyses. These results should be very similar to those shown in Fig. 1, which were obtained from the complete data. Figure 3a shows aggregation of leave-one-out *predictions* for the spring equinox, summer solstice, autumn equinox, and winter solstice samples collected from 2005 through 2010. Note that in each case the model predicted the “left out” sample label correctly, as that label always had the highest posterior probability.

3.2.2 Cross-Validation by 2/3 Training and 1/3 Testing with Replication

If you have a large number of samples it may not be feasible to do a leave-one-out analysis. In these cases you can analyze the data by withholding a randomly sampled portion of the data for testing, and train on the remaining data. In this example we will employ 2/3 of the data for training, and withhold the remaining 1/3 of the data for testing. In this protocol, samples are randomly assigned to either the training or testing dataset in 10 replicates. The random assignment is replicated multiple times to allow for testing of all samples in at least one of the replicate runs (*see Note 9*). The following protocol uses a single Perl script to create from the original matrix of training data (`train.ix`) and their associated labels (`env`) all of the input files required for multiple replicate runs of 2/3 training and 1/3 testing. The replicate data are analyzed as previously described.

1. Inspect the contents of the input files `train.ix` and `env` provided in the archive for protocol 2.2, and verify that they contain the correct data, and that the contents are correctly formatted.
2. Create a new analytical directory for this cross-validation analysis, and copy the data files `train.ix` and `env`, and the Perl scripts `replicate_maker.pl` to that analytical directory.
3. Save a text-based record of the absolute path to your new analytical directory. Here `<path1>` is used to denote this path in your file system.

A. predictions for leave-one-out cross validation



B. predictions for 2/3 cross validation aggregated over 10 replicates

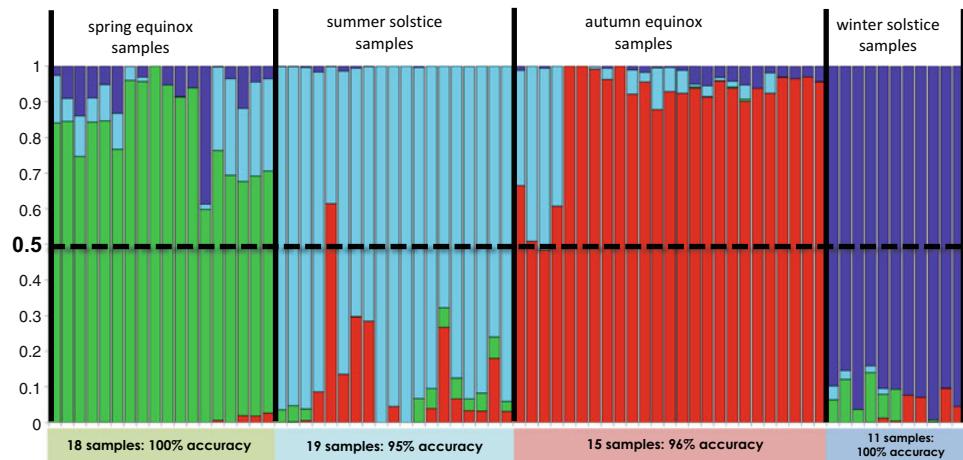


Fig. 3 (a) Predictions for leave-one-out cross-validation. Plot of posterior probabilities for each of the features (spring equinox, summer solstice, autumn equinox, and winter solstice) for each sample derived from 24 separate leave-one-out analysis of the Bedford Basin training dataset. The feature label with the largest contribution is taken as the best prediction for that sample. In all cases the correct prediction had the largest posterior probability (>0.50). Samples are ordered within the plot as follows: spring equinox (2005–2010), summer solstice (2005–2010), autumn equinox (2005–2010), and winter solstice (2005–2010). **(b)** Predictions for 2/3 cross-validation aggregated over 10 replicates. A plot of the predictions from 10 random subsamplings of the data where 2/3 of the samples served as the training subset and 1/3 served as the test subset. Some samples are represented more than once across the test subsets, as samples were assigned randomly within each replicate. The samples are ordered as in Panel (a). The feature label with the largest contribution is taken as the best prediction for that sample. Note that some classification errors are made here, and not in the leave-one-out cross-validation above, because the model was trained using less data (just 2/3 of the total). Interestingly, the small number of classification errors made for the summer solstice and autumn equinox samples involved samples that happened to be included in the test subsets multiple times, and in other replicates the same samples were classified correctly. This result is consistent with classification errors arising from the sampling errors associated with the smaller training subset used here

4. Save a text-based record of the absolute path to the directory that contains BioMiCoScripts.R. Here <path2> is used to denote this path in your file system.
5. Determine how many replicates you wish to perform on the data. We will run 10 for this analysis (*see Note 10*).
6. Navigate within your file system to your analytical directory and type the following at the command line:

```
perl replicate_maker.pl <path1> <path2> 10
```

List the contents of your analytical directly using the `ls` command and note the replicate subdirectories created by running the script in the previous step. For this example there should be 10 replicate subdirectories. Each replicate directory contains a training file (`train.txt`), a testing file (`test.txt`), a feature labels file (`env`), and the R script to run BioMiCo (`trainbyalltestbyone.R`). The analytical directory will also contain files (`rep_#.r`); this file contains the information used to generate the random replicates, thereby allowing for reproducibility of your result.

7. Run BioMiCo on each replicate by navigating to each of the numbered directories and type the following at the command line (*see Note 11*):

```
R CMD BATCH trainbyalltestbyone.R
```

8. To extract and summarize results, navigate within your file system to each replicate directory and run the two analysis R scripts from exercise 1 (*see Note 12*).

To analyze the *training data* for a replicate within a directory, type the following at the command line:

```
R CMD BATCH train_analysis.R
```

To analyze the *test data* for a replicate within a directory, type the following at the command line:

```
R CMD BATCH test_analysis.R
```

9. Inspect graphical output produced in **step 8** above. In this exercise, **step 8** will produce plots for the contribution of OTUs to assemblages, and how assemblages are related to feature labels, for each of the 10 cross-validation replicates. These will be similar to those produced from the leave-one-out cross-validation, and also similar to the full dataset results (Fig. 1), but with more variation due to training on just 2/3 of the full dataset. Likewise the predictions will have more error associated with them. However pooling *predictions* across all 10 replicates (Fig. 3b) yields results very similar to those obtained by using leave-one-out cross-validation (Fig. 3a).

The same sample can be in multiple replicates, and these should generally give very similar results, however, they can vary due to differences in the makeup of the training dataset. For this reason we observe a small number of mis-classifications in the summer solstice and autumn equinox samples (Fig. 3b) when using the 2/3 training and 1/3 testing approach. The important result is that sample classification is correct in aggregate (e.g., summer solstice 2006 was tested in 5 replicates, and correctly classified in 4 of those five tests).

3.3 Setting Parameter Values for the MCMC Algorithm and Assessing the Outcome

Statistical inference of posterior probabilities under BioMiCo is based on Markov Chain Monte Carlo (MCMC). While MCMC is a powerful tool for sampling probability distributions, it is a complex process and it can sometimes be challenging to run it sufficiently long enough to get a good result. The goal is to run the MCMC long enough such that the stationary distribution of the MCMC is equal to the target posterior distribution. For some datasets the target distribution can be achieved quickly, but for others it can take a very long time. Because of this, users must pay attention to several factors associated with running the MCMC. Chief among these are: (a) length of the MCMC run, (b) the amount of “burn in” to discard (*see Note 13*), and (c) the variability of results among separate runs due to the errors associated with estimating the posterior probability distribution via MCMC (*see Note 14*).

The example datasets used in Subheadings 3.1 and 3.2 above represent an “easy” inference problem, in so far as there is a very strong relationship between community structure and season, and it does not take very long for the MCMC to reach the target distribution. While this is ideal for exercises that have relatively quick run times, analyses of real data can be more challenging and require longer runs. For this reason, we have chosen a different dataset for this final protocol, based on a comprehensive study of temporal dynamics in vaginal microbiomes of asymptomatic women [15]. This dataset is not as “easy,” and is used to highlight the sensitivity of an analysis to inadequate choices for the burn-in and the total length of the MCMC run. The results are used to illustrate how users can show, for their own data, that they have identified values for both burn-in and length of the MCMC run to yield relatively stable estimate of posterior probabilities.

This dataset [15] is challenging because (a) there is very low OTU diversity, and (b) the research question involves distinguishing among many feature labels (13; $k = 10$). We employ these data to explore how the number of burn-in samples discarded and the overall length of MCMC impact the estimates of assemblage probabilities, and ultimately predictive accuracy. Our example dataset is for a subset of 10 asymptomatic women from [15] having high Nugent scores [16]. The data are provided within the BioMiCo archive as a single training matrix (`vagtrain.ix`) and an

associated set of labels (`vagenv`). Here we train only on a single feature label, patient identity for 10 women, so that the impact of burn-in and overall length of run can be efficiently illustrated.

1. Inspect the contents of the input files `vagtrain.ix` and `vagenv` provided within the BioMiCo archive for protocol 3, and verify that they contain the correct data, and that the contents are correctly formatted.
2. Create a new analytical directory for this analysis, and copy the data files `vagtrain.ix` and `vagenv`, and the R analysis script `parameter_analysis.r` to that directory.
3. Copy the Perl script `parameter_testing.pl` to your analytical directory.

This script automates the process of setting alternative values for the parameters associated with running the MCMC algorithm. Specifically, they are set by modifying the program variables within the R script called `parameter_testing.R`. This exercise is focused on the *burn-in* and the *delay* variables. The *burn-in* variable sets the number of initial MCMC iterations to discard from the analysis. These are discarded because they do not reflect the equilibrium state of the MCMC and thus can contribute considerable variability to the posterior distribution. The *delay* variable sets the number of iterations between the samples from the MCMC run that are retained for analysis. Thus, the combination of the number of samples from the MCMC run that will be retained for analysis (set by the `ndraws.per.restart` variable) and the *delay* variable sets the length of the MCMC run. Since we will be retaining the same number of samples from the MCMC for each analysis (see Note 15), we change the length of the MCMC run by increasing the *delay* between samples from the MCMC.

The script modifies the `train.results=` variable, which sets several parameters associated with the MCMC run. For this exercise the variables will be set as follows:

```
burnin=500, or =1000, or =1500, or =2000
nrestarts=1
ndraws.per.restart=20
delay=500, or =1000, or =1500, or =2000
```

Users can use the Perl script `parameter_testing.pl` to carry out an initial investigation for their own data by editing the `TRAIN.Mat` and `ENVS` variables within this script to point to their own files. However, values of *burn-in* and *delay* variable in this script will likely be too small for most real datasets. Alternatively, users can carry out their own investigation (recommended) by editing the values for `burnin=` and

`delay=` variables in `trainbyalltestbyone.R` to fully assess the challenge posed by their own data.

4. Save a text-based record of the absolute path to your new analytical directory. Here `<path1>` is used to denote this path in your file system.
5. Save a copy of the complete path to your directory that contains `BioMiCoScripts.R`. Here `<path2>` is used to denote this path in your file system.
6. Navigate within your file system to your analytical directory and type the following at the command line:

```
perl parameter_testing.pl <path1> <path2>
```

List the contents of your analytical directly using the `ls` command and note the subdirectories created by running the script in this step. Subdirectories for *burn-in* values of 500–2000 and *delay* values of 500–2000 were generated. Each directory contains a single R script `parameter_testing.r`. These scripts use the original training file (`vagtrain.ix`) and feature labels file (`vagenv`) from your analytical directory, so do not move or alter those files.

7. Run BioMiCo by navigating to each subdirectory and type the following at the at the command line:

```
R CMD BATCH parameter_testing.r
```

8. Extract and summarize results obtained under different settings for the MCMC algorithm as follows:

Copy the `parameter_analysis.r` script into each subdirectory.

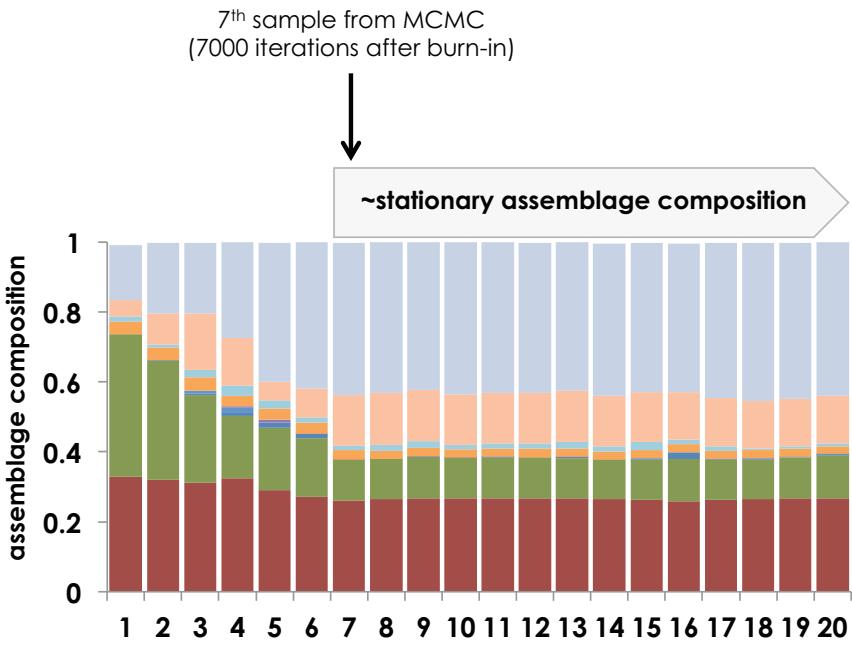
Open each copy of the `parameter_analysis.r` file and change the first line to load the name of the `.RData` file for that run.

Type the following at the command line:

```
R CMD BATCH parameter_analysis.r
```

This will yield a plot of cumulative estimates of the assemblage posterior probabilities; the plotted estimates are cumulative over the MCMC sampling process as it progresses (see **Note 15**). In this exercise, program variables were set to retain a total of 20 samples from the MCMC process. Thus, the plots show how estimates of the assemblage probabilities change as the MCMC process approaches the 20th sample from the chain. The assemblage probabilities should stabilize once the MCMC reaches its stationary distribution, and presumably, the target posterior distribution.

9. Inspect the graphical output produced in **step 8** above. Figure 4 illustrates how the estimates of assemblage probabilities



Assemblages

Key: 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2

total: L=25 assemblages permitted within model

predominant: 4 predominant assemblages (PP>0.1) at stationary

minor: 21 assemblages with PP<0.1 at stationary.
16 are not shown because PPs are trivially small throughout MCMC run.

MCMC

burn-in: 1st 500 iterations discarded

sampling Interval: 1000 iterations between samples retained for analysis

posterior estimates: cumulative

total length: 20000 iterations post burn-in

Fig. 4 Parameters of the MCMC run and convergence to relatively stable estimates of posterior probabilities. The plot shows the posterior probabilities (PPs) of assemblages within a single patient (No. 5 of a vaginal microbiome study [15]). The model was set to allow a maximum of 25 assemblages, with a sparse and symmetric Dirichlet prior employed to ensure that the data were explained using the minimum number of assemblage structures. Since this yields 16 assemblages with trivially small probabilities for visualization purposes only the 9 assemblages having PPs > 0.01 are shown in the plot. Note that even after discarding 500 iterations as burn-in, the PPs are not stationary, and a further 7000 MCMC iterations are required before a stable assemblage distribution is observed within the samples retained for analysis. Once the MCMC has reached its stationary distribution, the PPs can be interpreted as estimates of the mixture weights of microbial assemblages within patient 5's vaginal microbiome

stabilize as the MCMC progresses. Note that after the assemblage probabilities have stabilized, there are 5 assemblages having relatively large probabilities, with the remaining being trivially small (*see Note 16*). This result is achieved, despite the model permitting up to a maximum of $L = 25$ assemblages, because the Dirichlet prior minimizes the number of assemblages used to explain these data. Examine all the plots produced in **step 8** to investigate how burn-in and length of MCMC affect the point where assemblage probabilities stabilize.

4 Notes

1. Beware of characters within your OTU labels that R might interpret as math operators. If you cannot avoid using such characters, then place the entire OTU label within double quotes.
2. The total length of your MCMC run is determined by your choice of values for the `ndraws.per.restart` and the `delay=` variables.
3. The predominant OTUs within an assemblage are determined according to their posterior probability (PP) distribution. There are several alternative methods for inferring the predominant OTUs: (a) $PP \geq 0.01$ [3]; (b) the OTUs with the highest PPs that sum to 95% posterior density [17]; (c) the OTUs with the highest PPs that sum to 50% posterior density [17]; (d) the OTUs above the inflection point in the posterior distribution [18]. The provided R scripts use the first criterion; $PP \geq 0.01$.
4. Leave-one-out cross-validation ensures that predictive error will be assessed for each labeled sample in the dataset; however, it is computationally costly for large datasets. When datasets are so large that it is computationally prohibitive to carry out the leave-one-out approach, we recommend random division of data into 2/3 training and 1/3 testing. A protocol for cross-validation based on 2/3 training and 1/3 testing is presented in Subheading 3.2.2.
5. It is best practice to visually verify the contents of input files, whenever it is practical. Typical data analyses often involve adding, or removing, samples from the training datasets, as well as training on alternative sets of feature labels. In such cases, it is easy to lose track of the input data files, or introduce errors, thereby running the MCMC with incorrect data and/or labels. As MCMC can be computationally costly, a visual check of the contents of your input files is an important step in your data analysis protocol.

6. Running the individual replicates of a cross-validation analysis in parallel is recommended when multiple processors are available, as it will greatly reduce time it takes to obtain results.
7. Running the cross-validation replicates in parallel requires manually starting the MCMC for each replicate from within separate analytical directories (created automatically by running the Perl script). While it will take more time to individually start the MCMC runs, as compared to using a script to run the replicate analyses in series, the total time required to obtain results will be shorter when multiple CPU cores are available.
8. The Perl script `batchRfilemaker.pl` is employed to create an R script that can be used to automatically extract and summarize results of a cross-validation analysis regardless of how the replicates were run (in series or in parallel). The script produces the R script `batchRfile.r` which can then be run in R.
9. We have found that the apportionment of samples into 2/3 training and 1/3 testing works well for most datasets. With very large datasets, apportionment into 1/2 training and 1/2 testing can be used. As the training phase is more computationally costly than the testing phase, reducing the training data from 2/3 to 1/2 of the full dataset can improve the practicality of cross-validation for large datasets.
10. You want to run enough replicates so that each sample is included in a test dataset at least once. Note that 10, as was used here, is typically too small for most real-datasets, which tend to have more data samples than was used here for demonstration purposes.
11. This is the same R script that you ran in **step 11** in protocol. This R script was automatically placed in each of the replicate directories when they were created in an earlier step by using the Perl script.
12. The scripts required to extract and summarize results of a 2/3 training and 1/3 testing cross-validation analysis are not the same as those used for the leave-one-out cross-validation; for the 2/3 training and 1/3 testing cross-validation the scripts are the same as from protocol 1 **steps 12** and **14**. Running these scripts will produce figures similar to Figs. 1 and 2.
13. Because the start-point of the MCMC involves random initialization of the model variables, posterior probabilities derived from the beginning iterations of the MCMC do not represent samples from the stationary distribution. This is why the starting point of the MCMC contributes considerable error to the estimate of the target distribution, and is typically discarded.

The discarded iterations are referred to as “burn-in”. The user determines the number of burn-in iterations to discard, and sets the value within the R script called `trainbyalltest-byone.R`.

14. MCMC is employed to estimate the posterior probability distribution of both the OTU mixture weights and the assemblage mixture weights. The MCMC has to be run long enough to get a good result for each. Finite runs will be associated with some level of estimation error, and these errors will manifest as variability of results among separate MCMC runs. When you believe you have run the MCMC long enough to obtain the stationary assemblage distribution, you can carry out one or more separate MCMC runs of the same length to assess the level of estimation error for both the assemblage and OTU mixture weights.
15. The MCMC algorithm of BioMiCo is designed to generate random draws from the Bayesian posterior distribution. Thus, running the MCMC initiates a sampling process. Following the burn-in period (where all samples from the MCMC process are discarded), some samples are retained for further analysis. Because the MCMC is sampling conditional distributions, the successive iterations, or “steps,” of the sampling process are not independent. This means that successive MCMC samples will be auto-correlated with some tendency to produce “clumpy” results over the short term. The *delay* variable is used to set the number of MCMC iterations to run between those samples that will be retained from the MCMC for analysis; by “thinning” the sampling in this way the short-term effects of auto-correlation are minimized. It is the set of retained samples from the MCMC that will be used to estimate the posterior probabilities for the model. If the MCMC has been run long enough to converge to its stationary distribution, then it will have sampled OTUs and assemblages in proportion to their posterior probabilities.
16. BioMiCo assigns arbitrary ID numbers to the assemblage structures. Hence, the ID numbers (and the colors in the plots automatically produced by the provided R scripts) will differ for the same assemblage from one run to the next. Assemblages can be compared between runs according to their posterior distribution for OTUs, although this can be challenging in some cases [19].

Acknowledgments

This work was supported by NSERC Discovery Grant (DG3645-2015) and a Schulich Joint Research Project (JRP 48677) to JPB. We thank Joseph R. Migrone for helpful discussions, and for direct assistance with the computational resources. We thank Noor Youssef, Christopher Jones and Hong Gu for helpful discussions.

References

1. Shade A, Caporaso JG, Handelsman J et al (2013) A meta-analysis of changes in bacterial and archaeal communities with time. *ISME J* 7(8):1493–1506
2. Boon E, Meehan CJ, Whidden C et al (2014) Interactions in the microbiome: communities of organisms and communities of genes. *FEMS Microbiol Rev* 38(1):90–118
3. El-Swaiss H, Dunn KA, Bielawski JP et al (2015) Seasonal assemblages and short-lived blooms in coastal north-west Atlantic Ocean bacterioplankton. *Environ Microbiol* 17(10):3642–3661
4. Dunn KA, Moore-Connors J, MacIntyre B et al (2016) Early changes in microbial community structure are associated with sustained remission after nutritional treatment of pediatric Crohn's disease. *Inflamm Bowel Dis* 22(12):2853–2862
5. Dunn KA, Moore-Connors J, MacIntyre B et al (2016) The gut microbiome of pediatric Crohn's disease patients differs from healthy controls in genes that can influence the balance between a healthy and dysregulated immune response. *Inflamm Bowel Dis* 22(11):2607–2618
6. Tomas N, Fortin N, Bedrani L et al (2017) Characterising and predicting cyanobacterial blooms in an 8-year amplicon sequencing time course. *ISME J* 11(8):1746–1763
7. Leibold MA, Holyoak M, Mouquet N et al (2004) The metacommunity concept: a framework for multi-scale community ecology. *Ecol Lett* 7(7):601–613
8. Burke C, Steinberg P, Rusch D et al (2011) Bacterial community assembly based on functional genes rather than species. *Proc Natl Acad Sci U S A* 108(34):14288–14293
9. Friedman J, Hastie T, Tibshirani R (2001) The elements of statistical learning. Data mining, inference, and prediction. In: Springer series in statistics, New York
10. Knights D, Costello EK, Knight R (2011) Supervised classification of human microbiota. *FEMS Microbiol Rev* 35(2):343–359
11. Weiss S, Xu ZZ, Peddada S et al (2017) Normalization and microbial differential abundance strategies depend upon data characteristics. *Microbiome* 5(1):27
12. Shafiei M, Dunn KA, Boon E et al (2015) BioMiCo: a supervised Bayesian model for inference of microbial community structure. *Microbiome* 3(1):8
13. Zarraonaindia I, Smith DP, Gilbert JA (2013) Beyond the genome: community-level analysis of the microbial world. *Biol Philos* 28(2):261
14. Moore-Connors JM, Dunn KA, Bielawski JP et al (2016) Novel strategies for applied metagenomics. *Inflamm Bowel Dis* 22(3):709–718
15. Gajer P, Brotman RM, Bai G et al (2012) Temporal dynamics of the human vaginal microbiota. *Sci Transl Med* 4(132):132ra52
16. Nugent RP, Krohn MA, Hillier SL (1991) Reliability of diagnosing bacterial vaginosis is improved by a standardized method of gram stain interpretation. *J Clin Microbiol* 29(2):297–301
17. Bashwih RO (2016) Inference and investigation of marine microbial community structures in the global oceans (Masters Thesis). Dalhousie University Thesis database (<http://hdl.handle.net/10222/72144>)
18. Shafiei M, Dunn KA, Chipman H et al (2014) BiomeNet: A Bayesian model for inference of metabolic divergence among microbial communities. *PLoS Comput Biol* 10(11):e1003918
19. Tang C (2016) Statistical approaches for matching the components of complex microbial communities (Masters Thesis). Dalhousie University Thesis database (<http://hdl.handle.net/10222/71708>)



Chapter 18

Analyzing Metabolic Pathways in Microbiomes

Mobolaji Adeolu, John Parkinson, and Xuejian Xiong

Abstract

Understanding the metabolic activity of a microbial community, at both the level of the individual microbe and the whole microbiome, provides fundamental biological, biochemical, and clinical insights into the nature of the microbial community and interactions with their hosts in health and disease. Here, we discuss a method to examine the expression of metabolic pathways in microbial communities using data from metatranscriptomic next-generation sequencing data. The methodology described here encompasses enzyme function annotation, differential enzyme expression and pathway enrichment analyses, and visualization of metabolic networks with differential enzyme expression levels.

Key words Metabolism, Metatranscriptomics, Enzyme activity, Metabolic network, Enzyme annotation, Differential expression, Pathway enrichment, Network visualization

1 Introduction

Cellular metabolism comprises the chemical reactions which drive the synthesis and degradation of the essential building blocks of life. These reactions underlie essential cell processes spanning energy production, cell growth, cell proliferation, and apoptosis. Thus, an understanding of relationships and interactions between the component parts of cellular metabolism, namely enzyme-catalyzed reactions and their metabolites, can provide fundamental biological, biochemical, physiological, and ecological insights into the nature of the organism or community in which these metabolic components are found [1, 2]. Furthermore, a comprehensive understanding of the complex interactions between metabolic enzymes and metabolites and an understanding of their physiological role can identify novel therapeutic targets for the design and development of drugs targeting pathogens and/or aiding in the understanding of the contribution of metabolism to community changes such as dysbiosis [3–6].

The enzyme-catalyzed reactions, which form the basis of cellular metabolism, are organized into a hierachal classification system

developed by the Enzyme Commission (EC) [7]. Each enzyme-catalyzed reaction is assigned a four-field numerical designation known as an EC number, in which the first three fields represent the reaction type and the fourth field represents the substrate of the enzyme-catalyzed reaction. For example, an enzyme with the EC number 3.1.4.16 would be classified as a hydrolase (3.1.4.16), acting on ester bonds (3.1.4.16), specifically, the phosphodiester bond (3.1.4.16) found on the 2' carbon of a 2',3'-cyclic-nucleotide (3.1.4.16). Thus, the annotation of the correct EC classification to metabolic enzymes provides a means of building generalizable network representations of enzyme-catalyzed reactions and substrates representing metabolic pathways.

EC numbers are formally assigned by the Joint Commission on Biochemical Nomenclature (JCBN) of the International Union of Biochemistry and Molecular Biology (IUBMB) and the International Union of Pure and Applied Chemistry (IUPAC) on the basis of published experimental results characterizing individual enzyme-catalyzed reactions [8]. However, this process is costly and laborious, limiting its applicability to the growing abundance of genetic sequencing data produced by next-generation sequencing. Computational methods for the annotation of EC numbers to enzymes provide a scalable alternative to experimentally determined enzyme classification. The most common method of computational enzyme annotation is sequence similarity-based annotation [1, 2, 9–11]. Sequence similarity-based enzyme annotation uses sequence similarity search algorithms, such as BLAST [9], to identify enzymes which may be potentially homologous to a novel enzyme from a database of known and annotated enzymes, such as the manually curated Swiss-Prot [12] database or the Kyoto Encyclopedia of Genes and Genomes(KEGG) (1) which largely relies on automated orthology assignments. The EC annotation from the enzyme in the curated database is then considered to be the EC annotation for the novel enzyme. However, determining enzyme orthology based on sequence similarity is highly unreliable at lower sequence identities and is confounded by functionally diverged paralogous sequences and reaction diversity within orthologous sequences [13–15]. A second class of enzyme-annotation approaches utilizes sequence profiles, often based on hidden Markov models, to represent different enzyme classes [16–18]. These profiles rely on the presence of specific sequence patterns, motifs, and functionally discriminating residues in a novel enzyme sequence to annotate the novel enzyme with an EC classification from an annotated enzyme family. Profile-based enzyme annotation methods have higher sensitivity for members of an enzyme family at low overall sequence identity levels than sequence similarity-based annotation tools [16, 17]. A third class of enzyme-annotation methods utilize ensemble methodology encompassing multiple

heterogeneous enzyme-annotation strategies to improve the accuracy of enzyme annotations [19–21].

The enzymatic function annotations generated by the tools described above can subsequently be used to reconstruct metabolic network graphs of the microbial community. Enzyme-substrate relationships can be inferred from curated databases such as KEGG (1) and BioCyc (2). Then one or more protein–protein interaction networks, where nodes representing enzymes are linked through edges representing shared substrates, can be constructed by the homology mapping of the identified bacterial transcripts to *E. coli* homologs and subsequent layering of expression data onto a previously generated network for *E. coli* [22]. The networks can then be visualized using generalized network visualization tools such as Cytoscape [23] or tools specifically designed to visualize metabolic networks such as iPath [24]. The use of metabolic network graph visualizations allows us to obtain a global view of metabolism in the microbial community and enables the identification of biochemically related enzymes sharing similar taxonomic profiles.

In this chapter, we discuss a method to examine the expression of metabolic pathways in microbial communities using data from metatranscriptomic sequencing. First, in Subheading 2, we will describe a consensus ensemble enzyme annotation strategy utilizing both sequence similarity-based annotation produced using DETECT [11] and DIAMOND [25] and sequence profile-based annotation produced using PRIAM [16]. Then, in Subheading 3, we describe the statistical analyses required to evaluate differential expression of enzymes using DESeq2 [26] followed by the mapping of enzymes to metabolic pathways from KEGG (1) and BioCyc (2) and detection of differentially enriched pathways. Lastly, in Subheading 4, we describe a method for visualizing the differential expression of metabolic pathways analyzed in the previous sections. The methods we describe here include suggested programs and the scripts to run them. Additionally, we describe alternative tools which can be substituted for our suggestions. All custom scripts described in this chapter can be obtained from github (<https://github.com/ParkinsonLab/Metabolic-pathways-in-microbiomes>).

2 Enzyme Annotation

Preface: The following section assumes that you have the amino-acid sequences of the full-length proteins from your metatranscriptomic sequencing project available as fasta file(s) and taxonomically partitioned expression data for those proteins, in the format shown in Table 1 with protein names in column 1 rather than EC numbers, available in a comma-separated file. The generation of protein sequences and taxonomic annotations of metatranscriptomic sequencing reads have been covered in previous sections. Translation of nucleotide sequences to amino acid and file format

Table 1 The format of the enzyme table. The first column lists the annotated enzymes by their EC numbers, and the second column gives the RPKM value of enzymes. The remaining columns show the RPKM values of enzymes under different user-specified taxonomic categories. For each enzyme, the RPKM (second column) is the sum of RPKMs of all genes mapped to the enzyme, while its RPKM of a taxonomic category is the sum of RPKMs of mapped genes belonging to the category

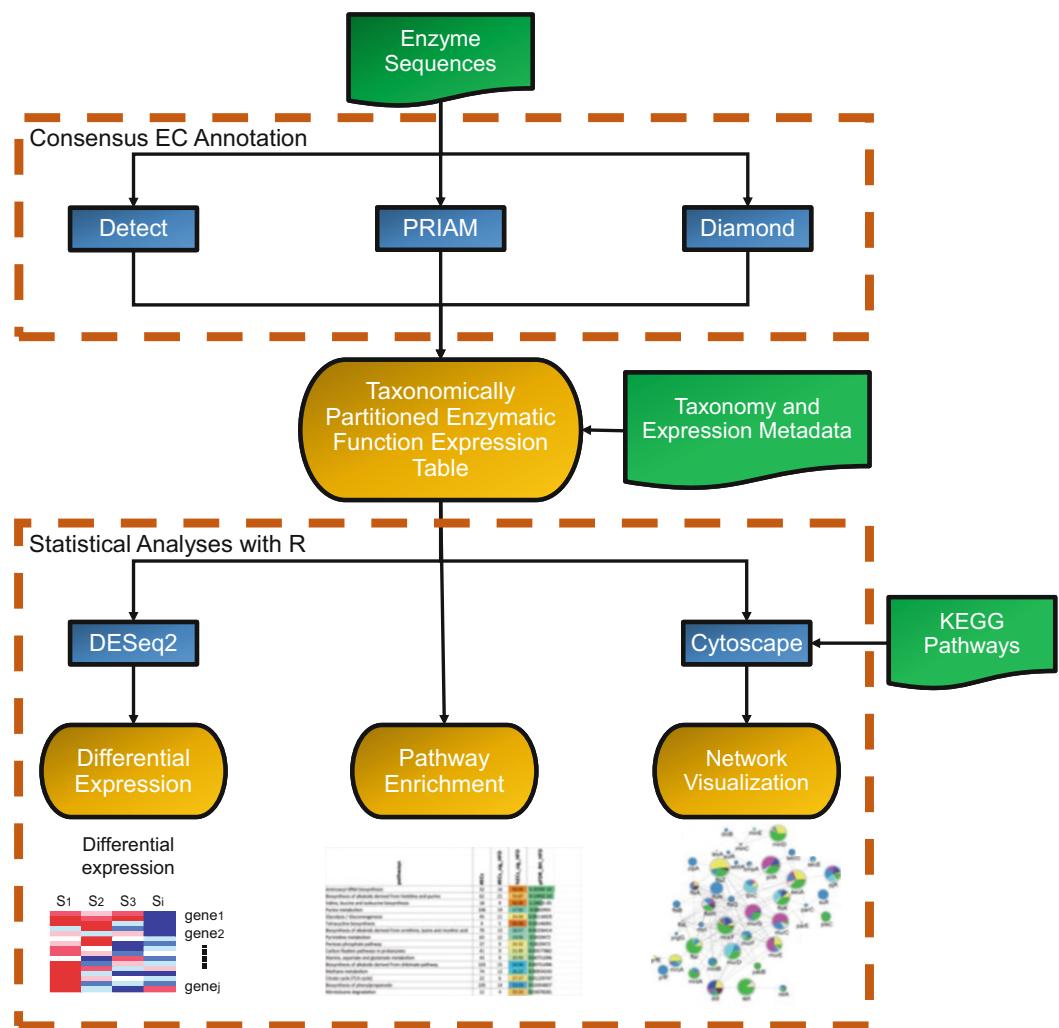


Fig. 1 Overview of the enzyme annotation and visualization pipeline

conversion from any other popular sequence file format to fasta can be accomplished using EMBOSS [27]. Lastly, the instructions below assume that you are running commands in a POSIX compliant shell (e.g., Linux, macOS, Cygwin, or the Windows Subsystem for Linux in Windows 10) and that you have both java and python in your path variable.

Here we describe a consensus ensemble enzyme annotation strategy utilizing DETECT [11], DIAMOND [25], and PRIAM [16] (Fig. 1). In the methodology described below, we run each of the three tools independently to generate enzymatic function predictions for the genes/proteins in the metatranscriptome and then combine their predictions using a simple heuristic (e.g., DETECT || (DIAMOND && PRIAM)) through a Python script (*see Note 1*).

2.1 DETECT-Based EC Annotation

DETECT [11] is a high-precision sequence similarity-based enzymatic function annotation tool which is differentiated from other sequence similarity based annotation tools by DETECT’s utilization of enzyme family-specific models for sequence diversity among enzyme families. These models of sequence diversity enable DETECT to calculate a robust integrated likelihood score for each enzyme function prediction allowing us to select only high-confidence enzymatic function predictions.

1. Install DETECT (compsysbio.org/projects/DETECT):

```
$ wget
http://compsysbio.org/projects/DETECT/detect_1.0.tar.gz
$ tar xvzf detect_1.0.tar.gz
```

2. Download and extract the EMBOSS package, compile the programs, then add them to user’s local path (required for DETECT).

```
$ wget ftp://emboss.open-bio.org/pub/EMBOSS/emboss-latest.tar.gz
$ tar xvzf emboss-latest.tar.gz
$ cd ./EMBOSS-*
$ ./configure
$ make
$ export PATH=$PATH:./emboss/
```

3. Run DETECT command on the selected protein fasta file (*see Note 2*).

```
$ python Detect.py input.fasta_file --output_file output_file --top_predictions_file top_predictions_file --num_threads N
```

where:

`input.fasta_file`—the filename of input protein fasta file.

`--output_file output_file`—the filename of output results.

`--top_predictions_file top_predictions_file`—the filename of results that enumerates predictions with probability over 0.2.

`--num_threads N`—number of CPU threads.

4. Extract high-confidence predictions from “`top_predictions_file`” which is generated from the above DETECT step (*see Note 3*).

```
$ python extract_toppred.py top_predictions_file detect_EC_file
```

where:

`top_predictions_file`—the filename of results that enumerates predictions with probability over 0.2 from the above DETECT step.

`detect_EC_file`—the filename of DETECT-annotated EC results where predictions are with the high confidence.

2.2 DIAMOND-Based EC Annotation

DIAMOND [25] is a sequence aligner for protein and translated DNA searches. It designed for high performance analysis of large sequence datasets (*see Note 4*).

1. Install DIAMOND (github.com/bbuchfink/diamond):

```
$ wget
https://github.com/bbuchfink/diamond/releases/download/
v0.8.38/diamond-linux64.tar.gz
$ tar xvzf diamond-linux64.tar.gz
```

2. Use the following Python script to download the Swiss-Prot and Enzyme reference databases and generate a tab-separated sequence to EC mapping file.

```
$ python Download_EC_Annotated_SwissProt.py
```

3. Index the EC-annotated Swiss-Prot database for use with DIAMOND.

```
$ diamond makedb -p N --in db_file -d db_name
```

where:

`-p N`—number of CPU threads.

`--in db_file`—the filename of the input protein reference database in FASTA format (may be gzip compressed).

`--d db_name`—the filename of the output DIAMOND database.

4. Use DIAMOND to search for matches to your proteins in the EC-annotated subset of the Swiss-Prot database.

```
$ diamond blastp --query input_file --db db_name --outfmt
"6 qseqid sseqid qstart qend sstart send evalue bitscore
qcovhsp slen pident" --out output_file --evalue
0.0000000001 --max-target-seqs 1
```

where:

`--query input_file`—the filename of the query input file in FASTA format (may be gzip compressed).

`--db`—the filename of the DIAMOND database.

`--outfmt`—format of the output files (*see Note 5*).

--out output_file—the filename of output results.
 --eval 0.0000000001—maximum expected value to report an alignment is 0.0000000001.
 -max-target-seqs 1—the first 1 match that pass the eval threshold.

5. Extract hits and generate a tab-separated EC annotation file.

```
$ python extract_diamond.py output_file diamond_EC_file
"SwissProt_EC_Mapping.tsv"
```

where:

output_file—the filename of DIAMOND output results from the above DIAMOND step.

diamond_EC_file—the filename of a tab-separated EC annotation table.

"SwissProt_EC_Mapping.tsv"—the filename of a mapping table between SwissProt protein IDs and EC numbers.

2.3 PRIAM-Based EC Annotation

PRIAM [16] is a profile-based enzyme function annotation tool which utilizes sequence profiles produced from multiple sequence alignments of enzyme families. Additionally, the authors of PRIAM provide profile hidden Markov models for each of the enzyme family alignments used in their PRIAM profile generation, allowing for the substitution of HMMer [28] based profile annotation for PRIAM-based annotation (*see Note 6*).

1. Install PRIAM (priam.prabi.fr):

```
$ wget http://priam.prabi.fr/utilities/PRIAM_search.jar
$ wget http://priam.prabi.fr/REL_MAR15/Distribution.zip
$ unzip Distribution.zip, mkdir PRIAM_MAR15/PROFILES/LIBRARY,
ls PRIAM_MAR15/PROFILES/*chk > PRIAM_MAR15/PROFILES/LIBRARY/profiles.list
```

2. Install the NCBI BLAST stand-alone applications (required for PRIAM):

```
$ wget ftp://ftp.ncbi.nih.gov/blast/executables/legacy.
NOTSUPPORTED/2.2.26/blast-2.2.26-x64-linux.tar.gz
$ tar xvzf blast-2.2.26-x64-linux.tar.gz
```

3. Use PRIAM to search for matches to your proteins in the PRIAM EC profile database.

```
$ java -jar PRIAM_search.jar -np number_of_threads -n
Session_name -i input_file -p Distribution/PRIAM_MAR15
-od output_directory -e T -pt 0.5 -mo -l -mp 70 -cc T -bd
blast_directory/bin
```

where:

- np—number of processors to use.
- n—the name of the PRIAM searching session.
- i—the name of input file containing the protein sequences (Fasta format).
- p—the directory containing the release of PRIAM.
- od—the output directory containing all intermediates and results files.
- e T—if the job successfully complete, the intermediate files will be erased.
- pt 0.5—the threshold of probability is 0.5 (*see Note 7*).
- mo -1—the maximum overlap length between the matches of two profiles. Setting it to -1 means this filter is inactivated.
- mp 70—minimal length proportion of a matched profile is set as 70.
- cc T—check catalytic residues patterns.
- db—the location of the installed ncbi blast tool (*see Note 8*).

4. Extract top hits and generate a tab-separated EC annotation file using the following Python script:

```
$ python extract_PRIAM.py
output_directory/RESULTS/seqsECs.tab priam_EC_file
```

where:

seqsECs.tab—the results from the above step and is located in the “RESULTS” folder inside the PRIAM output folder, i.e., output_directory/RESULTS/.

priam_EC_file—the PRIAM-annotated EC table (tab-separated).

2.4 Consensus EC Annotation

Lastly, we will generate consensus predictions from the output of the three previously run tools using a Python script that generates a list of EC annotations that were predicted by either the high-precision annotation tool (DETECT) or by both our sequence similarity and profile-based annotation tools (DIAMOND and PRIAM).

```
$ python Consensus_ECs.py detect_EC_file diamond_EC_file
priam_EC_file output_dir
$ python EC_RPKM_table.py "Consensus.ECs_All"
taxonomically_partitioned_protein_expression_data result_table
```

3 Statistical analysis

After enzyme annotation, we may perform statistical analyses to identify, for example, enzymes exhibiting differential expression. Please note that in order to obtain meaningful results, multiple samples are required (either technical or biological replicates [29]). For metatranscriptomic analysis, we recommend that the number of sample replicates should be at least 3 or more (*see Note 9*).

3.1 Differential Expression of Enzymes

To identify differentially expressed enzymes between different conditions, we can apply one of two widely used tools, DESeq [26, 30] and edgeR [31]. The tools apply similar strategies, for example, implementing general differential analyses on the basis of the negative binomial (NB) model for count data, however they differ in how they estimate data dispersion [32].

In this section, all statistical analyses are done using R version 3.3.2 [33], and both DESeq and edgeR have available R packages [32]. Here we focus on DESeq2 [26] (an improved version of DESeq). It has been recently demonstrated that DESeq2 and edgeR outperform other tools when there are <12 sample replicates per condition and DESeq outperforms others when there are >12 replicates per condition [34].

1. As a result from the previous annotation step, an annotated enzyme table (in a comma-delimited csv file) and a metadata file (in a csv file) are fed into this step.
2. The enzyme table includes counts number for enzymes (in rows) under different conditions (in columns). The metadata includes at least two columns, in which different condition names are in the first column, while the second one specifies the condition factor related to each condition name. Please make sure that the header of the second column is “condition.”
3. Before running R scripts, four R packages (DESeq2, RColorBrewer, gplots and calibrate) are required to be installed properly. If not, the packages can be installed through Bioconductor as follows:

```
source("https://bioconductor.org/biocLite.R")
biocLite("DESeq2"); biocLite("RColorBrewer");
biocLite("gplots"); biocLite("calibrate")
```

4. Use the following command to call DESeq2 package and check its version.

```
library("DESeq2"); packageVersion("DESeq2")
```

5. Call the functions “my_functions.R” and “main_run_DeSeq2.R,” and then run the DESeq pipeline with proper input parameters (*see Note 10*):

```
source('my_functions.R')
source('main_run_DeSeq2.R')
R1 <- main_run_DESeq2(input_count_file, input_meta_file,
condition1, condition2, pvalue_cutoff, foldchange_cutoff,
outputfile)
```

where:

input_count_file—the filename of the EC table.
input_meta_file—the filename of the metadata.
condition1,2—the pair-wise conditions to be analyzed.
pvalue_cutoff—the threshold of adjusted p values.
foldchange_cutoff—the threshold of fold changes (log2).
outputfile—the filename of output results (*see Note 11*).
R1—the results from DESeq2 analysis (*see Note 12*).

3.2 Pathway Enrichment Analysis

We use the hypergeometric test (*see Note 13*) to detect pathways enriched with significantly expressed enzymes, sig. enzymes in short.

1. Following the steps from the above subsection, run pathway enrichment analysis with proper input parameters:

```
R2 <- main_enrichment_analysis(input_pathway_file, input_ec_file, pvalue_cutoff, foldchange_cutoff, outputfile)
```

where:

input_pathway_file—the filename of pathway information.
input_ec_file—the filename of enzymes table generated from the above subsection.
pvalue_cutoff—the threshold of adjusted p values.
foldchange_cutoff—the threshold of fold changes (log2).
outputfile—the filename of output results (*see Note 14*).
R2—the pathway table with pathway information and statistics from enrichment analysis.

4 Visualization

To visualize enzyme expression under different conditions in the context of a metabolic pathway, we need a new type of enzyme table (also in a tab-delimited format). For each condition, in addition to being summed across all taxa, the relative expression of an annotated enzyme, provided as reads per kilobase per million mapped

reads (RPKM), can also be defined and subsequently visualized, for a set of user-specified taxonomic categories. The taxonomic categories are predefined by users based on their expectation and interests. For example, in Table 1, the enzyme table under one condition has 19 columns, listing 17 predefined taxonomic categories.

Once generated this enzyme table is imported into the Cytoscape [23] software platform to visualize enzyme expression in the context of different metabolic pathways, where not only differentially expressed enzymes but also the taxonomic breakdown for each expressed enzyme is provided. In this section, metabolic pathway information is obtained from the KEGG (1) resource, although BioCyc (2) represents a viable alternative. Here we illustrate pathway visualization through the Pantothenate and CoA Biosynthesis pathway (see Fig. 2).

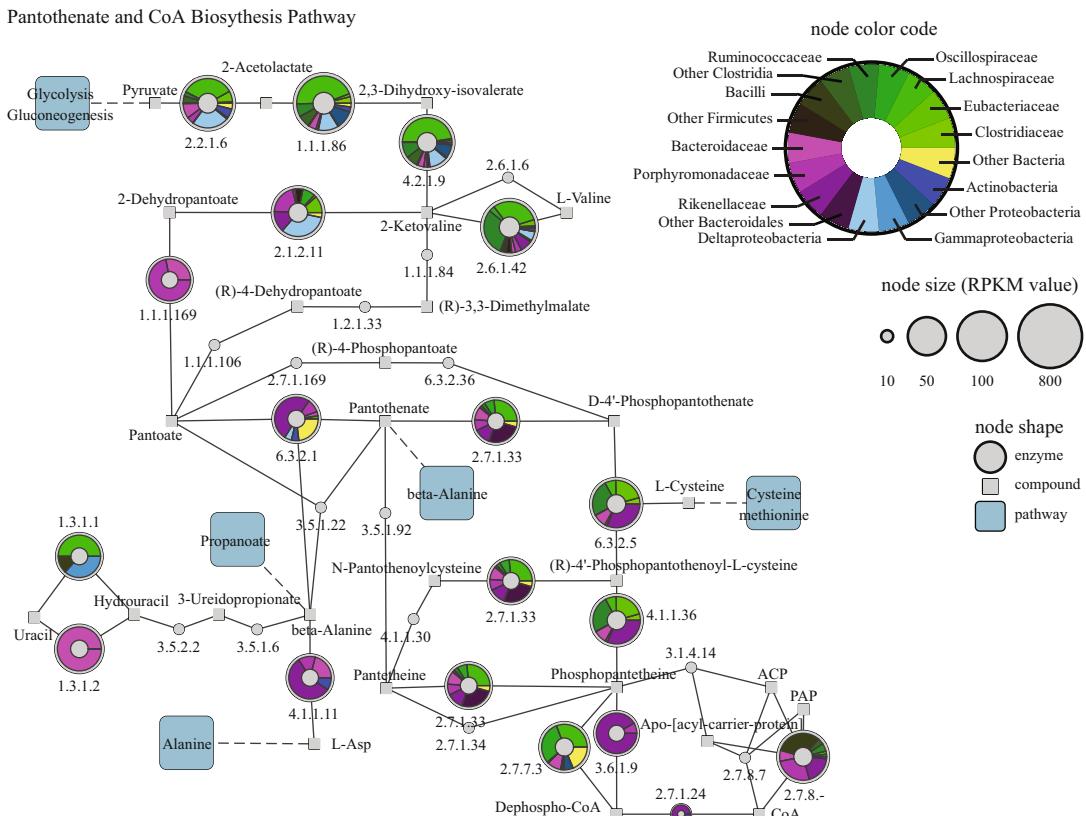


Fig. 2 The Pantothenate and CoA Biosynthesis pathway. Enzymes involved in the pathway are represented as circular nodes. The size of nodes shows the expression (RPKM) value of enzymes. The taxonomic composition of enzymes is shown in a donut chart graphic, where shades of red, green, and blue indicate taxonomic categories from Firmicutes, Bacteroides, and Proteobacteria, respectively. Other taxonomic groups are indicated in yellow

1. Install Cytoscape (<http://www.cytoscape.org/>). In this section, we use Cytoscape v3.4.0.
2. Download a reference pathway (EC) in KGML format from KEGG (*see Note 15*).
3. Import the KGML file into Cytoscape. Go to **File** → **Import** → **Network** → **File...** → select “ec00770.xml” → **Open** → check “**Import pathway details from KEGG Database**” → **OK** (*see Note 16*).
4. Import the enzyme table into the pantothenate pathway. Go to **File** → **Import** → **Table** → **File...** → select the enzyme table file → **Open**. From the pop-up **Import Columns From Table** window, select **KEGG_NODE_LABEL** from the drop-down menu of **Key Column for Network:** → **OK**.
5. Remove/Change some properties of nodes defined by the KEGG style. Go to **Control Panel** → select **Style** tab → select **Node** tab.
 - (a) Remove the oval shape of nodes: check the **Lock node width and height** box.
 - (b) Make the nodes re-locatable: delete the **X Location** and **Y Location** properties. That is, open the subpanel of **X Location** or **Y Location** by clicking arrow → click **Remove Mapping** symbol.
 - (c) Let the size of nodes vary according to the RPKM value of enzymes. That is, open the subpanel of **Size** by clicking arrow → select **RPKM** from the drop-down menu of **Column**, and **Continuous Mapping** of **Mapping Type** → set the proper node size by double clicking the mapping window.
6. Set donut graphics to nodes in order to visualize the taxonomic distribution of each enzyme. Go to **Image/Chart 1** → select **Charts** tab → select **Ring** symbol.
 - (a) Add taxonomic categories to donuts. In the **Data** tab, add the taxonomic categories from **Available Columns** to **Selected Columns** orderly.
 - (b) Set the color scheme of each taxonomic category orderly in the **Options** tab.
 - (c) Click **Apply**.
7. To customize visualization, Cytoscape offers the ability to alter a number of network properties such as Label Font Size, Label Position, Fill Color, node location, as well as edge properties (*see Note 17*).

5 Notes

1. There are other methods that users can use to annotate enzymes such as EFICAz (Enzyme Function Inference by a Combined Approach) [19]. Here we rely on DETECT as it has proven to result in more accurate annotations.
2. Due to the size of metatranscriptomic datasets, to speed up enzyme annotation, users can split input sequence files into smaller files and then perform DETECT in parallel on a compute cluster. The user can split the file using a custom Python script, e.g.,

```
$ python split_fasta.py number_of_sequences  
input_fasta_file out_directory
```

where:

number_of_sequences—the predefined number of sequences in each smaller file.

input_fasta_file—the filename of input sequence data.

out_directory—the directory containing split sequence files.

3. If the input fasta file is split into multiple files, users will need to concatenate the resultant DETECT output:

```
$ cat output1 output2 ... outputN > output_file
```

4. BLAST is an alternative way to perform sequence similarity-based database searching. We use DIAMOND here because DIAMOND is significantly faster than BLAST while maintaining comparable sensitivity.
5. Users can get details from the DIAMOND manual (https://github.com/bbuchfink/diamond/blob/master/diamond_manual.pdf).
6. We have found PRIAM to usefully complement BLAST-based annotations, which together can be used to identify additional enzymes that DETECT is unable to capture. Other profile-based methods such as Prosite [18] might also be considered to increase enzyme coverage. However, the user should be aware that these methods can increase the number of false-positive assignments.
7. For matching profiles, PRIAM provides a Bayesian-based probability score associated with the annotation. Above a given threshold of probability, a set of matching profiles can be considered as true positive annotations.
8. PRIAM needs the NCBI BLAST tool to be installed and may also need its location specified. To test if users have to explicitly

state the location of BLAST, users can type “blastpgp –” in the console. If the computer responds with the blastpgp help menu BLAST is present in the users default path and no further action is required. Otherwise, users will need to specify the “bd” option. This option normally needs to be used only the first time users use PRIAM or if they change the location of BLAST from its default.

9. Replication of samples or conditions lends statistical power that increases the confidence of experimental findings. Replicates can be used to measure variation, increase precision and detect outliers in the experiments. However, the high cost of meta-transcriptomic studies has constrained the number of replicates. Here, to meet the statistical requirement and reduce the cost of experiments, we recommend the minimum number of sample replicates is 3.
10. Here we recommend setting pvalue_cutoff = 0.05 and fold-change_cutoff = 1.
11. The expressed enzymes with statistics, such as log2Fold-Change (fold change in log2 format), padj (adjusted p values), and counts are output into a csv-formatted file.
12. The DESeq2 results are returned into R1 which is a list containing the following four variables.

R1\$dds—the input data set formatted for the DESeq2 function.

R1\$res—the results from the DESeq2 analysis.

R1\$summary—the information about which variables and tests were used.

R1\$resSig—the significantly differentially expressed enzymes (sig. enzymes in short) with the statistics and original counts.

13. In statistics, hypergeometric tests use the hypergeometric distribution to calculate the statistical significance of obtaining k successes in n draws without replacement where the sample population is N and consists of exactly K successes. The hypergeometric test is often used for enrichment analysis and is equivalent to one-tailed Fisher’s exact test.
14. The pathways with enrichment statistics, such as padj (adjusted p values), are output into a csv-formatted file.
15. For example, for the Pantothenate and CoA Biosynthesis pathway, users can go to KEGG pathway website http://www.genome.jp/kegg-bin/show_pathway?map00770, select **Reference pathway (EC)** → **GO** → **Download KGML**. Then a file named “ec00770.xml” is downloaded onto users’ local computer.

16. The pantothenate pathway is visualized in a style analogous to those presented on the KEGG website. Each oval blue node represents an enzyme, small white dots are compounds, and cyan rectangles refer to other pathways.
17. Please refer to the Cytoscape tutorial (http://www.cytoscape.org/manual/Cytoscape3_4_0Manual.pdf) for more details.

Acknowledgments

This work was supported through funding from the Natural Sciences and Engineering Research Council (RGPIN-2014-06664), and the University of Toronto's Medicine by Design initiative which receives funding from the Canada First Research Excellence Fund. Computing resources were provided by the SciNet HPC Consortium. SciNet is funded by the Canada Foundation for Innovation under the auspices of Compute Canada, the Government of Ontario, Ontario Research Fund—Research Excellence, and the University of Toronto.

References

1. Kanehisa M, Furumichi M, Tanabe M et al (2017) KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res* 45:D353–D361
2. Caspi R, Billington R, Ferrer L et al (2015) The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Res* 44:D471–D480
3. Nicholson JK, Holmes E, Kinross J et al (2012) Host-gut microbiota metabolic interactions. *Science* 336:1262–1267
4. Lee DS, Burd H, Liu J (2009) Comparative genome-scale metabolic reconstruction and flux balance analysis of multiple *Staphylococcus aureus* genomes identify novel antimicrobial drug targets. *J Bacteriol* 191:4015–4024
5. Holmes E, Kinross J, Gibson GR (2012) Therapeutic modulation of microbiota-host metabolic interactions. *Sci Transl Med* 4:137rv136
6. Wacker SA, Houghtaling BR, Elemento O et al (2012) Using transcriptome sequencing to identify mechanisms of drug action and resistance. *Nat Chem Biol* 8:235–237
7. Webb OF (1992) Enzyme nomenclature. Academic Press
8. Moss GP (2017) Recommendations of the nomenclature committee of the international union of biochemistry and molecular biology on the nomenclature and classification of enzymes by the reactions they catalyse, <http://www.chem.qmul.ac.uk/iubmb/enzyme/>.
9. Altschul SF, Madden TL, Schäffer AA (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 25:3389–3402
10. Moriya Y, Itoh M, Okuda S et al (2007) KAAS: an automatic genome annotation and pathway reconstruction server. *Nucleic Acids Res* 35:W182–W185
11. Hung SS, Wasmuth J, Sanford C et al (2010) DETECT—a density estimation tool for enzyme classification and its application to *Plasmodium falciparum*. *Bioinformatics* 26:1690–1698
12. Boekmann B, Blatter MC, Famiglietti L et al (2005) Protein variety and functional diversity: Swiss-Prot annotation in its biological context. *C R Biol* 328:882–899
13. Devos D, Valencia A (2001) Intrinsic errors in genome annotation. *Trends Genet* 17:429–431
14. Schnoes AM, Brown SD, Dodevski I et al (2009) Annotation error in public databases: misannotation of molecular function in enzyme superfamilies. *PLoS Comput Biol* 5: e1000605

15. Gerlt JA, Allen KN, Almo SC et al (2011) The Enzyme Function Initiative. *Biochemistry* 50:9950–9962
16. Claudel-Renard C, Chevalet C, Faraut T et al (2003) Enzyme-specific profiles for genome annotation: PRIAM. *Nucleic Acids Res* 31:6633–6639
17. Eddy SR (1998) Profile hidden Markov models. *Bioinformatics* 14:755–763
18. Sigrist CJ, De Castro E, Cerutti L (2012) New and continuing developments at PROSITE. *Nucleic Acids Res* 41:D334–D347
19. Kumar N, Skolnick J (2012) EFICAz2.5: application of a high-precision enzyme function predictor to 396 proteomes. *Bioinformatics* 28:2687–2688
20. Mohammed A, Guda C (2011) Computational Approaches for Automated Classification of Enzyme Sequences. *J Proteomics Bioinform* 4:147–152
21. Mostafavi S, Morris Q (2010) Fast integration of heterogeneous data sources for predicting gene function with limited annotation. *Bioinformatics* 26:1759–1765
22. Peregrin-Alvarez JM, Xiong X, Su C et al (2009) The Modular Organization of Protein Interactions in *Escherichia coli*. *PLoS Comput Biol* 5:e1000523
23. Shannon P, Markiel A, Ozier O (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res* 13:2498–2504
24. Yamada T, Letunic I, Okuda S et al (2011) iPath2.0: interactive pathway explorer. *Nucleic Acids Res* 39:W412–W415
25. Buchfink B, Xie C, Huson DH (2015) Fast and sensitive protein alignment using DIAMOND. *Nat Meth* 12:59–60
26. Love MI, Huber W, Anders S (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol* 15:550
27. Li W, Cowley A, Uludag M (2015) The EMBL-EBI bioinformatics web and programmatic tools framework. *Nucleic Acids Res* 43: W580–W584
28. Eddy SR (2011) Accelerated Profile HMM Searches. *PLoS Comput Biol* 7:e1002195
29. Prosser JI (2010) Replicate or lie. *Environ Microbiol* 12:1806–1810
30. Anders S, Huber W (2010) Differential expression analysis for sequence count data. *Genome Biol* 11:R106
31. Robinson MD, McCarthy DJ, Smyth GK (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 26: 139–140
32. Anders S, McCarthy DJ, Chen Y (2013) Count-based differential expression analysis of RNA sequencing data using R and Bioconductor. *Nature protocols* 8:1765–1786
33. R Core Team (2016) R: A language and environment for statistical computing.
34. Schurch NJ, Schofield P, Gierliński M (2016) How many biological replicates are needed in an RNA-seq experiment and which differential expression tool should you use? *RNA* 22:839–851



Chapter 19

Sparse Treatment-Effect Model for Taxon Identification with High-Dimensional Metagenomic Data

Zhenqiu Liu and Shili Lin

Abstract

To identify disease-associated taxa is an important task in metagenomics. To date, many methods have been proposed for feature selection and prediction. However, those proposed methods are either using univariate (generalized) regression approaches to get the corresponding P -values without considering the interactions among taxa, or using lasso or L_0 type sparse modeling approaches to identify taxa with best predictions without providing P -values. To the best of our knowledge, there are no available methods that consider taxon interactions and also generate P -values.

In this paper, we propose a treatment-effect model for identifying taxa (STEMIT) and performing statistical inference with high-dimensional metagenomic data. STEMIT will provide a P -value for a taxon through a two-step treatment-effect maximization. It will provide causal inference if the study is a clinical trial. We first identify taxa associated with the treatment-effect variable and the targeting feature with sparse modeling, and then estimate the P -value of the targeting gene with ordinary least square (OLS) regression. We demonstrate that the proposed method is efficient and can identify biologically important taxa with a real metagenomic data set. The software for L_0 sparse modeling can be downloaded at <https://cran.r-project.org/web/packages/l0ara/>.

Key words Treatment effect, Sparse modeling, Taxon identification, Metagenomics, Statistical inference

1 Introduction

With advances in next-generation sequencing technologies, massive metagenomics data have been generated and made available in the public domain. Large sequencing efforts, including the Human Microbiome Project [5, 24] and the Earth Microbiome Project [3], have recently generated data sets consisting of billions of reads corresponding to trillions of nucleotides (base pairs). In medicine, metagenomics has been utilized to link changes in the microbial communities living inside and on human bodies to common dis-

eases such as type II diabetes [6], Crohn's disease [13], and cancer. Efficient mining of such metagenomic data will facilitate the understanding of the microbiome–host interactions and the study of the biological mechanisms of a disease.

An essential aspect of metagenomic analysis is to detect the changes of relative taxon abundances across different clinical or environmental conditions. Differentially abundant taxa can be detected by statistically assessing the difference in relative abundances between communities. However, there are several challenges related to analyzing the metagenomic data, due to discrete data type, high dimensionality, small sample size, under sampling, and high levels of biological and technical variability. Because taxa are quantified by counting the number of reads matching specific genes, metagenomic data are discrete and asymmetric, and there are dependencies between expected values and variances. Standard statistical methods that rely on normality assumptions are not appropriate and may lose statistical power for detecting differences [7, 9]. Data normalization and transformation must be performed before applying standard statistical tools. Usually, the total sum of counts in each sample is used to correct the difference in sequencing depth across different samples. Further, arcsin and log functions can be used to transform the data into normal distribution [9, 14]. Two primary approaches for detecting differential taxa associated with different clinical (experimental) conditions are currently in use. The first type is composed of statistical tests. Fisher's exact and binomial tests have been used in metagenomics for pairwise comparisons of samples [12, 21]. Other methods are either based on linear model or generalized linear model (GLM), including methods based on zero-inflated normal distribution, negative binomial model, zero-inflated beta regression, and zero-inflated negative binomial model [2, 15, 16, 25]. However, these approaches all analyze one gene at a time without considering the interactions among them. More importantly, since the statistics for different genes are dependent on one another, there is no guarantee that the Benjamini-Hochberg procedure can control the false discovery rate (FDR). On the other hand, our group has developed sparse modeling and machine-learning approaches for multivariate taxon selection and prediction [9–11]. Such approaches have performed well in selecting genes for predictions, but one drawback of such types of approaches is that they cannot provide *P*-values and correct statistical inferences for the selected taxa.

In recent years, both statistical inferences for high-dimensional data and treatment-effect models have been studied in different research fields including economics and biostatistics [1] (Gruber et al. 2013). In this manuscript, we propose a sparse treatment-effect model to adjust the effect of other taxa, and then estimate the *P*-values of each gene with the identified taxa for that gene and an ordinary least square (OLS) methods. For data collected from a

randomized trial, such an approach will reveal possible causal relations among taxa. It will provide more reliable P -values even when dealing with case-control or other observational data.

2 Methods

2.1 Treatment-Effect Model

Treatment-effect models were first proposed in the context of Rubin's model of causality [17, 18]. The model was proposed in terms of the potential outcomes under treatment alternatives, in which only one of the outcomes is observed for each patient sample (subject). The causal effect of a treatment on a subject is defined as the difference between an observed outcome and its counterfactual. Let the upper-case letters be the random variables, and lower-case letters be the values of the random variables. Mathematically, assuming that a sample of subjects is randomly assigned to two treatment arms, treatment and control, denoted as $T \in \{0, 1\}$. Let $\Upsilon(t)$ be the potential outcome of a subject if assigned to treatment $T = t$, $t = \{0, 1\}$. Given the baseline covariates of a p -dimensional vector $X = \mathbf{x}$, where $\mathbf{x} = [x_1, x_2, \dots, x_p]'$, and n independent and identically distributed realization of (Υ, T, X) , $\{(\Upsilon_i, T_i, X_i), i = 1, \dots, n\}$, the treatment effect (ATE) given $X_i = \mathbf{x}$ can be defined as

$$\begin{aligned}\tau(\mathbf{x}) &= E[\Upsilon_i(1) - \Upsilon_i(0)|X_i = \mathbf{x}] \\ &= E[\Upsilon_i|X_i = \mathbf{x}, T_i = 1] - E[\Upsilon_i|X_i = \mathbf{x}, T_i = 0],\end{aligned}$$

where $X_i = \mathbf{x}$ accounts for the heterogeneity in response to treatment with respect to individual baseline covariates. In a high-dimensional setting, there are many baseline covariates related to a subject; however, it is unknown as to which covariates are associated with the outcome Υ . Sparse modeling such as L_1 or L_0 -based regularized regression can be used to identify outcome-associated covariates.

2.2 Sparse Modeling with ADMM Method

Suppose we have the following linear model:

$$E[\Upsilon(t)|X] = \alpha + \tau t + X\beta + \epsilon, \quad t = 0, 1,$$

where X is the baseline covariate matrix that is potentially associated with the outcome Υ . Then the average treatment effect is $E[\Upsilon(1)|X] - E[\Upsilon(0)|X] = \tau$. In principle, the parameters α and β can be estimated with regularized linear regression and then the P -values of the corresponding parameters can be evaluated with ordinary least squares (OLS) linear regression. This two-step outcome regression approach can be estimated with an alternating direction method of multipliers (ADMM) efficiently.

Let $\theta = [\alpha, \tau, \beta]'$ be the collection of model parameters. We further let \mathbf{y} be the gene being considered, t be the class information (1/0), and X be the rest of the genes that we need to adjust. Therefore, we have $Z = [\mathbf{1}, t, X]$, and we will find the solution for $E[\mathbf{y} | Z] = Z\theta + \epsilon$ by solving the following optimization problem:

$$\operatorname{argmin}_{\theta} E(\theta) = \operatorname{argmin}_{\theta} \left\{ \frac{1}{2} (\mathbf{y} - Z\theta)'(\mathbf{y} - Z\theta) + \frac{\lambda}{2} \|\theta\|_0 \right\},$$

where $\|\theta\|_0 = \sum_i I(\theta_i \neq 0)$ is the number of nonzero elements. The optimization problem is NP hard. We have proposed the following iterative system to approximate the zero norm optimization problem [11]:

$$\operatorname{argmin}_{\theta} E(\theta) = \operatorname{argmin}_{\theta} \left\{ \frac{1}{2} (\mathbf{y} - Z\theta)'(\mathbf{y} - Z\theta) + \frac{\lambda}{2} \sum_i \frac{\theta_i^2}{\psi_i^2} \right\}, \quad (1)$$

$$\psi = \theta. \quad (2)$$

Given ψ , we utilize the alternative direction method of multiplier (ADMM) to solve Eq. 1. Let

$$f(\theta) = \frac{1}{2} (\mathbf{y} - Z\theta)'(\mathbf{y} - z\theta),$$

$$\mathcal{G}(\phi, \psi) = \frac{\lambda}{2} \sum_i \frac{\phi_i^2}{\psi_i^2}.$$

The ADMM of Eq. 1 is

$$\operatorname{argmin}_{\theta, \phi} f(\theta) + \mathcal{G}(\phi), \quad \text{subject to } \theta = \phi. \quad (3)$$

The augmented Lagrangian is

$$\begin{aligned} L_{\psi}(\theta, \phi, \mu) &= f(\theta) + \mathcal{G}(\phi, \psi) + \mu'(\theta - \phi) + \frac{\lambda}{2} \|\theta - \phi\|_2^2 \\ &= f(\theta) + \mathcal{G}(\phi, \psi) + \frac{\lambda}{2} \|\theta - \phi + \mu\|_2^2 + \frac{\lambda}{2} \|\mu\|_2^2, \end{aligned} \quad (4)$$

where μ is the dual variable. Under the ADMM framework, Eqs. 1 and 2 become

$$\operatorname{argmin}_{\theta, \phi, \psi} \left\{ f(\theta) + \mathcal{G}(\phi, \psi) + \frac{\lambda}{2} \|\theta - \phi + \mu\|_2^2 \right\}, \quad \text{subject to : } \psi = \theta. \quad (5)$$

The ADMM problem can be solved by the following procedures:

The ADMM procedures for L_0 approximation:

Given $\theta^0 = \psi^0 = \phi^0$ and small ε
for $k = 1, 2, \dots$ do,

$$\theta^{k+1} = \arg \min_{\theta} \{f(\theta) + \frac{\lambda}{2} \|\theta - \phi^k + \mu^k\|_2^2\}$$

$$\psi^{k+1} = \theta^{k+1}$$

$$\phi^{k+1} = \arg \min_{\phi} \{g(\phi, \psi^{k+1}) + \frac{\lambda}{2} \|\theta^{k+1} - \phi + \mu^k\|_2^2\}$$

$$\mu^{k+1} = \mu^k + (\theta^{k+1} - \phi^{k+1})$$

Stop if $\|\theta^{k+1} - \theta^k\| < \varepsilon$, $\|\mu^{k+1} - \mu^k\| < \varepsilon$, and $\|\mu^{k+1} - \theta^{k+1}\| < \varepsilon$
end

2.2.1 θ Updates

The suboptimal problem for θ can be solved analytically, given that the other parameters are fixed.

$$E_\theta = f(\theta) + \frac{\lambda}{2} \|\theta - \phi^k + \mu^k\|_2^2.$$

$$\frac{\partial E_\theta}{\partial \theta} = -Z'(\mathbf{y} - Z\theta) + \lambda(\theta - \phi^k + \mu^k) = 0$$

\Rightarrow

$$\theta^{k+1} = (Z'Z + \lambda I)^{-1}[Z'\mathbf{y} + \lambda(\phi^k - \mu^k)], \quad (6)$$

where the second term measures the differences between the primal solution ϕ^k and the dual solution μ^k .

2.2.2 ϕ Updates

Given θ^{k+1} , ψ^{k+1} , and μ^k , we need to optimize the following suboptimal problem:

$$\begin{aligned} E_\phi &= g(\phi, \psi^{k+1}) + \frac{\lambda}{2} \|\theta^{k+1} - \phi + \mu^k\|_2^2, \\ &= \frac{\lambda}{2} \sum_i \frac{\phi_i^2}{(\psi_i^{k+1})^2} + \frac{\lambda}{2} \|\theta^{k+1} - \phi + \mu^k\|_2^2. \end{aligned} \quad (7)$$

$$\frac{\partial E_\phi}{\partial \phi} = \lambda \phi / (\psi^{k+1})_o^2 - \lambda(\theta^{k+1} - \phi + \mu^k) = 0,$$

where $(\psi^{k+1})^2 = (\psi^{k+1})_o \circ \psi^{k+1}$ is the element-wise multiplication.

\Rightarrow

$$\phi^{k+1} = \left(\frac{(\theta^{k+1})_o^2}{(\theta^{k+1})_o^2 + 1} \right) (\theta^{k+1} + \mu^k). \quad (8)$$

Equation 8 demonstrates that $\phi_i^{k+1} = 0$ if $\theta_i^{k+1} = 0$ and ϕ is a weighted average between the primal variable θ^{k+1} and dual variable μ^k .

2.2.3 λ Determination

The regularized parameter λ controls the sparsity of the model. It can be determined by cross-validation or by popular information criteria, including Akaike information criterion (AIC), Bayesian information criterion (BIC), and risk inflation criteria (RIC) [11].

2.2.4 Treatment Effect and P-value Estimation

Because of the oracle property of L_0 optimization, the genes identified should be the independent variables that predict the output gene. We fit the ordinary least square (OLS) model with the identified genes for treatment effect and P -value estimation. The post double-selection method [1] is used for such two-step estimation. In this approach, the binary class is regarded as the treatment effect (t), each gene is in turn considering as an output y , and the rest of the genes are regarded as adjusting for confounding. We first fit two models separately with L_0 -based sparse modeling:

$$\mathbf{y} = \alpha_1 + X\beta_1 + \varepsilon_1, \quad (9)$$

$$t = \alpha_2 + X\beta_2 + \varepsilon_2, \quad (10)$$

Then, we run the OLS model $\mathbf{y} = \alpha + \tau t + X_s \beta + \varepsilon$, with t and the union of the selected genes X_s from Eqs. 9 and 10 together for p -value and treatment-effect estimation.

3 Results

3.1 Simulation Data

We generate the count data with the following known correlation structure:

$$\Sigma = \begin{bmatrix} 1 & r & r^2 & r^3 & \dots \\ r & 1 & r & r^2 & \dots \\ r^2 & r & 1 & r & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \dots & r^3 & r^2 & r & 1 \end{bmatrix},$$

and Poisson distributions [10], where the count data has the Poisson distribution with mean c_{ij} , and $\log c_{ij}$ has the normal distribution $N(\mu, \Sigma)$ with mean μ and covariance matrix Σ . We generate the data with several values of correlations $r = 0, 1, 0, 3, 0, 6$, and $0, 9$. The sample size is fixed at $n = 100$, and the number of features is $p = 200$. The two groups have the same correlation structure but different means only for the first 5 features. The first group has the mean vector $\mu_1 = [2, 2, \dots, 2]$, while the means of the first 5 features in the second group increased by $\Delta\mu$, which are set at five different levels: $0.5, 0.75, 1.0, 1.25$, and 1.5 . The means of the rest of the features in group 2 are all set at 2 as in the first group. The simulations are performed 100 times for each of the mean differences ($\Delta\mu = 0.5, 0.75, 1.0, 1.25, 1.5$) and correlation ($r =$

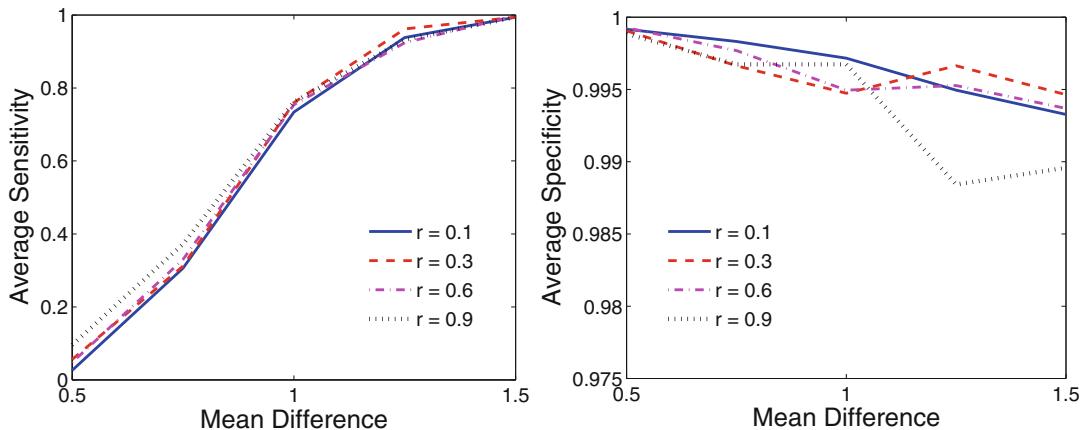


Fig. 1 Average sensitivity and specificity for different $\Delta\mu$'s and different correlations. Left panel: average sensitivity; Right panel: average specificity

0. 1, 0. 3, 0. 6, 0. 9). The average sensitivity and specificity for all combinations are shown in Fig. 1.

Figure 1 shows that the average sensitivity over 100 simulations increases as $\Delta\mu$ increases from 0.5 to 1.5. When $\Delta\mu = 1.5$, the average sensitivity is close to 1. In addition, the average sensitivities with different r 's are quite similar, indicating that the proposed method is robust with different correlations. Moreover, as shown in the right panel of Fig. 1, the average specificity decreases slightly, when $\Delta\mu$ increases from 0.5 to 1.5. Nevertheless, the average specificities are greater than 0.995, and the false positive rates ($1 - \text{specificity}$) are controlled at 0.005 for $r = 0.1, 0.3, 0.6$. The false positive rate for very high correlation ($r = 0.9$) can also be controlled at less than 0.015, indicating the proposed approach can control the false positive rate very well. The average treatment effect (TE) for the first 5 features over 100 simulations is presented in Fig. 2.

Figure 2 shows that the treatment effects have a strong positive linear relation with the mean difference for the first 5 features. There are no significant differences for different correlations, indicating that the proposed approach can detect the mean difference efficiently.

3.2 IBD Metagenomic Data

The metagenomic data that will be used in this investigation were generated by Tong et al. [23]. There are a total of 299 samples with 76 inflammatory bowel disease (IBD) and 223 non-IBD subjects. Two hundred eighty five of the samples (72 IBDs and 213 controls) have metagenomic data available. Overall, there are 5648 operational taxonomic units (OTUs) identified. We merge the OTUs at the genus and species levels and discard the taxa with less than 5 reads on average. Then we apply the sparse treatment-effect model for identifying taxa (STEMIT) to the data. The identified

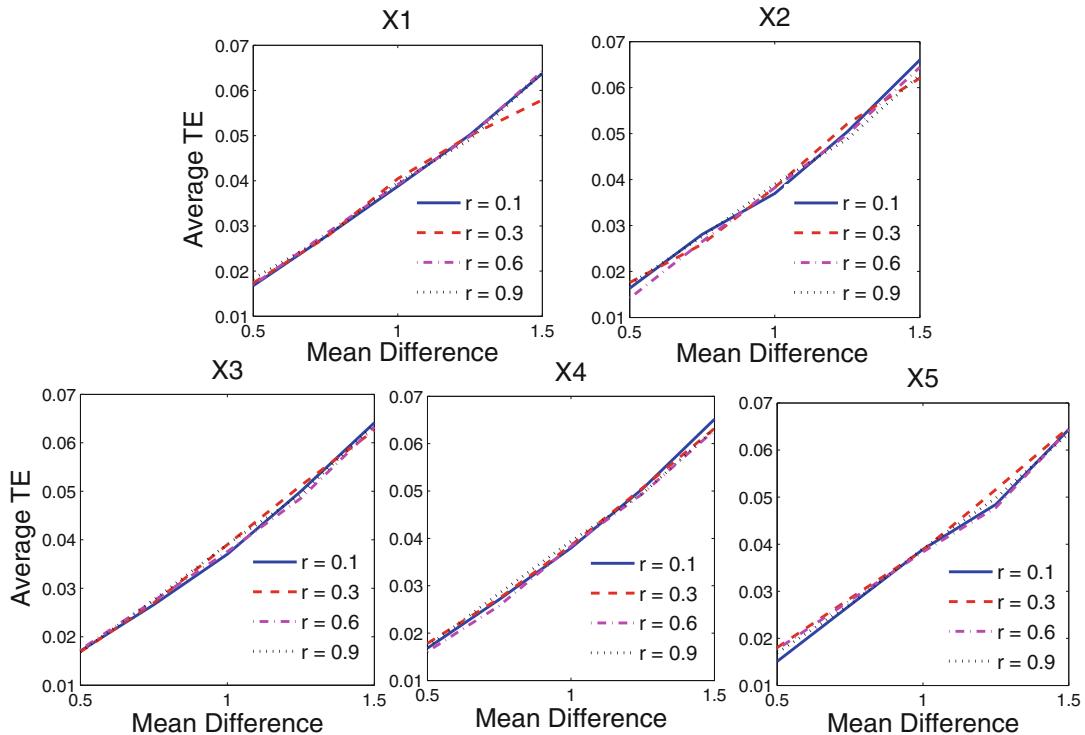


Fig. 2 The treatment effect with different mean differences $\Delta\mu$ for the first 5 features

Table 1
Identified taxa at the genus and species level

Genus level			Species level		
Selected taxa	ATE	FDR	Selected taxa	ATE	FDR
<i>Coprococcus</i>	-0.024	0.034	<i>Roseburia faecis</i>	-0.002	0.001
<i>Escherichia</i>	0.022	0.03	<i>Ruminococcus obeum</i>	-0.003	0.001
<i>Eubacterium</i>	-0.016	0.0008	<i>Serratia marcescens</i>	0.0033	7.2e-06
<i>Faecalibacterium</i>	-0.140	1.0e-10			
<i>Holdemania</i>	-0.0053	0.0001			

ATE average treatment effect, FDR false discovery rate

taxa are presented in Table 1. The table shows that five genera are identified to be associated with IBD all with a $FDR < 0.05$. Genera *Coprococcus*, *Eubacterium*, *Faecalibacterium*, and *Holdemania* have lower relative abundance in IBD with a negative treatment effect, while genus *Escherichia* has a higher relative abundance in IBD. Among the four genera with lower abundance in IBD, *Coprococcus* and *Faecalibacterium* are known to be associated with IBD [19],

while *Eubacterium* has been reported to be significantly decreased in Crohn's disease (CD) patients as compared with healthy controls [22]. While *Holdemani*a has not been well studied in IBD, it is demonstrated to be associated with glucose metabolism disorders and the metabolic syndrome in older adults [8]. *Escherichia*, on the other hand, has higher relative abundance in IBD. It is known that *Escherichia* may cause intestinal epithelial barrier dysfunction, and is a potential gut pathogen [20]. Finally, the three associated species are *Roseburia faecis*, *Ruminococcus obeum*, and *Serratia marcescens*. While *S. marcescens* has a higher relative abundance, both *R. faecis* and *R. obeum* have a lower relative abundance with a negative treatment effect.

4 Conclusions

We propose a two-step treatment effect approach for identifying disease-associated taxa and making statistical inferences with high-dimensional metagenomic data. STEMIT can adjust the effect of other taxa and obtain a reliable *P*-value for each gene. When the taxa are highly correlated, there is no guarantee that the Benjamini-Hochberg procedure can control the false discovery rate (FDR). As demonstrated with simulations, the proposed method can control false positive rates with different sensitivities and correlations efficiently. When applied to an IBD metagenomic data set, it can identify biologically important taxa for IBD.

References

- Belloni A, Chernozhukov V, Hansen C (2014) High-dimensional methods and inference on structural and treatment effects. *J Econ Perspect* 28(2):29–50
- Fang R, Wagner B, Harris J, Fillon S (2016) Zero-inflated negative binomial mixed model: an application to two microbial organisms important in oesophagitis. *Epidemiol Infect* 1:1–9
- Gilbert JA, Jansson JK, Knight R (2014) The earth microbiome project: successes and aspirations. *BMC Biol* 12(1):1
- Gruber S, van der Laan MJ (2010) A targeted maximum likelihood estimator of a causal effect on a bounded continuous outcome. *Int J Biostat* 6(1):26. <http://doi.org/10.2202/1557-4679.1260>.
- Human Microbiome Project Consortium (2012) Structure, function and diversity of the healthy human microbiome. *Nature* 486 (7402):207–214
- Karlsson F, Tremaroli V, Nookaew I, Bergström G, Behre C, Fagerberg B, Nielsen J, Bäckhed F (2013) Gut metagenome in European women with normal, impaired and diabetic glucose control. *Nature* 498:99–103
- Law C, Chen Y, Shi W, Smyth G (2014) Voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol* 15(2):R29
- Lippert K, Kedenko L, Antonielli L, Kedenko I, Gemeier C, Leitner M, Kautzky-Willer A, Paulweber B, Hackl E (2017) Gut microbiota dysbiosis associated with glucose metabolism disorders and the metabolic syndrome in older adults. *Benef Microbes* 13:1–12. <http://doi.org/10.3920/BM2016.0184>
- Liu Z, Hsiao W, Cantarel BL, Drábek EF, Fraser-Liggett C (2011) Sparse distance-based learning for simultaneous multiclass classification and feature selection of metagenomic data. *Bioinformatics* 27(23):3242–3249
- Liu Z, Sun F, Braun J, McGovern D, Piantadosi S (2015) Multilevel regularized regression

- for simultaneous taxa selection and network construction with metagenomic count data. *Bioinformatics* 31(7):1067–1074
11. Liu Z, Li G (2016) Efficient regularized regression with L_0 penalty for variable selection and network construction. *Comput Math Methods Med* 2016:3456153
 12. Mackelprang R, Waldrop MP, DeAngelis KM, David MM, Chavarria KL, Blazewicz SJ, Rubin EM, Jansson JK (2011) Metagenomic analysis of a permafrost microbial community reveals a rapid response to thaw. *Nature* 480 (7377):368–371
 13. Manichanh C, Rigottier-Gois L, Bonnaud E, Gloux K, Pelleter E, Frangeul L, Nalin R, Jarrin C, Chardon P, Marteau P et al (2006). Reduced diversity of faecal microbiota in Crohn's disease revealed by a metagenomic approach. *Gut* 55(2):205–211
 14. Nayfach S, Pollard KS (2016) Toward accurate and quantitative comparative metagenomics. *Cell* 166(5):1103–1116
 15. Paulson JN, Stine OC, Bravo HC, Pop M (2013) Differential abundance analysis for microbial marker-gene surveys. *Nat Methods* 10(12):1200–1202
 16. Peng X, Li G, Liu Z (2016) Zero-inflated beta regression for differential abundance analysis with metagenomics data. *J Comput Biol* 23 (2):102–110
 17. Rubin DB (1974) Estimating causal effects of treatment in randomized and nonrandomized studies. *J Educational Pschol* 66:688–701
 18. Rubin DB (2005) Causal inference using potential outcomes: design, modeling, decisions. *J Am Stat Assoc* 100:322–331
 19. Shaw KA, Bertha M, Hofmekler T, Chopra P, Vatanen T, Srivatsa A, Prince J, Kumar A, Sauer C, Zwick ME, Satten GA, Kostic AD, Mulle JG, Xavier RJ, Kugathasan S (2016) Dysbiosis, inflammation, and response to treatment: a longitudinal study of pediatric subjects with newly diagnosed inflammatory bowel disease. *Genome Med* 8(1):75
 20. Shawki A, McCole DF (2016) Mechanisms of intestinal epithelial barrier dysfunction by adherent-invasive *Escherichia coli*. *Cell Mol Gastroenterol Hepatol* 3(1):41–50
 21. Smith RJ, Jeffries TC, Roudnew B, Fitch AJ, Seymour JR, Delpin MW, Newton K, Brown MH, Mitchell JG (2012) Metagenomic comparison of microbial communities inhabiting confined and unconfined aquifer ecosystems. *Environ Microbiol* 14(1):240–253
 22. Takahashi K, Nishida A, Fujimoto T, Fujii M, Shioya M, Imaeda H, Inatomi O, Bamba S, Sugimoto M, Andoh A (2016) Reduced abundance of butyrate-producing bacteria species in the fecal microbial community in Crohn's disease. *Digestion* 93(1): 59–65
 23. Tong M et al (2013) A modular organization of the human intestinal mucosal microbiota and its association with inflammatory bowel disease. *PLoS One* 8:e80702
 24. Turnbaugh P, Ley R, Hamady M, Liggett C, Knight R, Gordon J (2007) The human microbiome project: exploring the microbial part of ourselves in a changing world. *Nature* 449:804–810
 25. Zhang X, Mallick H, Tang Z, Zhang L, Cui X, Benson AK, Yi N (2017) Negative binomial mixed models for analyzing microbiome count data. *BMC Bioinf* 18(1):4

INDEX

A

- ABAWACA 219, 220
Abiotic variables/factors 268, 276, 277
Abundance 17, 31, 38, 43,
 45, 50, 58, 66, 67, 69, 78, 82, 83, 89, 91, 97, 113,
 118, 123, 124, 139, 144, 147, 156–159,
 161–164, 170, 171, 173, 175, 176, 183,
 185–190, 194–196, 200–202, 204–206, 210,
 211, 215, 216, 220, 223, 227–237, 239, 249,
 252, 253, 264, 268, 271, 292, 310, 316, 317
ALDEx2 151, 159–161,
 195, 203–206, 208, 210, 211
Alpha diversity 42, 58, 68,
 79, 82, 83, 119, 124, 126, 134
Alternating direction method of multipliers
 (ADMM) 311, 312
American Gut 247, 249
Analysis of variance (ANOVA) 159, 165, 239
Ancestral states/ancestral state
 reconstruction 172, 173, 176
Annotation 176,
 180, 189, 197, 199, 200, 224, 292, 293,
 295–300, 304
ANOSIM 119, 125
Antibiotics 50, 72, 243
Anvi'o 219, 220
Archaea 64, 100, 144
Asthma 51, 64

B

- Bacteria 29–31, 35, 46, 50,
 52–54, 56, 58, 64, 66, 69, 70, 72, 73, 79, 80, 88,
 100, 144, 196, 198, 228, 243, 294
Bacteriophage 18
BAM files 186, 219
Barcode 30, 33, 57,
 59, 60, 75, 79–81, 232, 238
Bayesian methods 268
BBMap 133, 135,
 190, 222
BEDtools 218
Beta diversity 43, 58, 68,
 78, 79, 118, 119, 124, 125, 139, 175
Betweenness 207, 238,
 239, 246, 247, 254, 264

- Bifidobacterium* 66, 80
Binning 99, 180, 185,
 186, 189, 190, 215–224
Bioconductor 151, 152,
 156, 159, 195, 203, 300
BioCyc 293, 302
Biofilm 100, 101, 103, 104, 243
Biogeochemical cycling 1, 216
Biological observation matrix (BIOM)
 format 78, 175
Biomarkers 31, 89–95
BioMiCo 267–284, 288
Biotic variables/factors 268, 276, 277
BLAST 22, 23, 45,
 197, 200, 292, 298, 304, 305
Blastocystis 30, 31
Bombus pascuorum 120
Boost C++ libraries 269
Bowtie2 38, 181,
 186, 190, 198, 218, 219, 221, 222
Bray–Curtis index 250
Breastmilk 63–83
BWA 190, 222

C

- Canadian Bioinformatics Workshop 171
Candidate phyla 99
CASAVA 128, 222
CbiA 88, 91, 94, 96
Centered log-ratio (CLR) 159, 160,
 163–166, 202
Centrality 207, 238,
 239, 246, 247, 254, 265
CheckM 181, 186, 187, 220
Chimera/chimeric 31, 38,
 40, 58, 75, 78, 117, 122, 133, 136, 139, 140
Chronic obstructive pulmonary disease (COPD) 51
Clustering 31, 44, 45,
 58, 83, 117, 125, 133, 137, 138, 141, 147, 198,
 202, 223, 234, 239, 246, 255, 256, 259
Clustering coefficient 256, 259
CoDaSeq 195, 203–209
COG 174, 176, 232
Colony forming units (CFUs) 49
Colostrum 64–66, 68, 70

- Community structure 145, 196, 267–288
 Compositional data 164, 193–212, 245
 CONCOCT 190, 219, 220, 223
 Confounding variables 71, 72
 CONSeQuence 90, 144
 Contamination 1, 2, 4–9, 11,
 25, 59, 72, 73, 76, 81, 109, 194, 196, 220, 224
 Contig clustering 186
 Contigs 20–25, 144,
 179–190, 194, 217, 218, 220, 221
 Controls 2, 4–8, 11,
 19, 20, 22, 25, 35, 50, 58, 69, 71, 72, 76, 80–82,
 90, 103, 105, 106, 109, 133, 134, 139, 145, 173,
 210, 227–229, 231–233, 235, 239, 246, 303,
 310, 311, 314, 315, 317
 Co-occurrence 246
 Copy number 171, 173, 230
 Core samples 1–8, 12
 Coverage 20, 88, 100,
 137, 186, 188–190, 194, 196, 215–224, 304
 CRAN 195, 203, 211, 230
 Cross validation 269, 276–281, 286, 287, 314
 Culture 49–60, 88, 150, 215
 Culture-dependent 50
 Culture-independent 49, 50, 143
 Cumulative sum scaling (CSS) 156, 232
 Cutadapt 33, 35, 39, 44, 181, 187, 231
 Cystic fibrosis (CF) 51, 58
 Cytoscape 240, 293, 303, 306
- D**
- DADA2 116, 117, 121–122, 128, 147, 175
 De Bruijn graph 184, 188
 Degree 99, 240,
 244, 246, 247, 256–259
 Dendritic cells (DCs) 70
 Denoising 114, 116,
 117, 121, 122, 147, 175
 Dereplication 78
 DESeq2 151, 153–154,
 164, 293, 300, 301, 305
 DETECT 78, 80, 196,
 244, 246, 264, 293, 295–297, 299, 301, 304, 315
 DIAMOND 23, 195, 199,
 200, 293, 295, 297–299, 304
 Differential abundance 58, 147,
 150–159, 161, 194, 195, 201, 202, 204–206,
 229, 233, 235, 239
 Differential expression 156, 159, 293, 300–301
 DISCO 181, 189
 DNA
 extraction 2–4, 6, 8–11, 74–76, 79, 217
 quantification 4
 Draft genomes 100

E

- Earth Microbiome Project (EMP) 30, 33, 34
 edgeR 151, 153–155, 231, 300
 EFICAz 304
 EMBOSS 295, 296
 Emergent self organizing maps (ESOM) 219, 220
 EMPeror 79, 134
 Enzyme Commission (EC) numbers/
 classifications 199, 292, 294, 298
 Eukaryote/eukaryotic microbes 30–32, 34, 45
 Exploratory data analysis (EDA) 165, 194, 201–204
 Extended Bayesian information
 criterion (EBIC) 245, 252

F

- Faith's phylogenetic diversity 118, 124
 False-discovery correction (including
 Benjamini-Hochberg) 310
 False discovery rate (FDR) 167, 206,
 211, 234, 235, 253, 257, 310, 316, 317
 FASTA 25, 37–39, 78,
 136, 137, 184, 187, 217, 218, 293, 295–297, 299
 FASTQ 20, 21,
 35–37, 58, 114–116, 120, 128, 131, 134–136,
 183, 197, 199, 217, 218, 231, 232, 238
 FastQC 19, 21, 33,
 35, 37, 133–135, 181–183, 187, 188, 197, 231
 Fasttree 78, 118, 124, 128
 FASTX 44, 133, 135, 181, 187, 231
 FASTX-Toolkit 133, 181, 187, 231
 Filtration 17–19, 24, 104
 FinishM 221
 Firmicutes 67, 68, 70, 224, 294, 302
 Fisher's exact test 305
 Fluorescence-activated cell sorting
 (FACS) 100–102, 105, 108, 109
 454 FLX Titanium 152
 Function 1, 22, 23, 30,
 34, 64, 70, 87, 91, 97, 118, 119, 152–161, 163,
 164, 167, 169–176, 190, 193–199, 202, 203,
 205–211, 216, 230, 232, 233, 235, 243,
 246–265, 268, 276, 292, 293, 295, 296, 298,
 301, 304, 305, 310, 317
 Functional annotation 197
 Functional prediction 22, 174, 175, 193
 Fungi 30, 64, 88, 140

G

- Gammaproteobacteria 88, 294
 Gastrointestinal tract 70, 170
 GC content 180
 Gene expression 198, 211,
 227–231, 233, 237, 239

GeneMarkS 19, 23
 Generalized linear mixed models (GLMM) 79
 Generalized linear model (GLM) 160, 167, 310
 Genome-resolved metagenomics 215–224
 Genomes 1, 11, 18,
 20–25, 35, 45, 46, 91, 99–103, 106–109, 144,
 169, 171–176, 180, 183–187, 190, 194,
 198–200, 215, 216, 220–222, 224, 227, 231,
 232, 292, 305
 GetORF 25
 Github 41, 114, 115,
 120, 132–133, 199, 220, 221, 293
 GNU scientific library (GSL) 269
 Graph clustering 239
 Graph theory 245
 Greengenes 44, 108, 118,
 122, 133, 137, 171, 172, 175

H

Hidden Markov models 298
 HLGLVQAHEVR 88, 94
 Homology 22, 23, 200, 293
 Host contamination 196
 Host interactions (microbe–microbe,
 host–microbe) 50, 243, 244, 310
 HPLC 90
 Human immunodeficiency virus (HIV) 72
 Human microbiome 29, 30, 81
 HUMAnN 232
 Hypergeometric distribution/
 hypergeometric test 301, 305

I

IDBA_UD 189, 217, 218
 igraph 195, 203, 207,
 211, 230, 248, 249, 252–255, 258–261, 264, 265
 Illumina 4, 19–21, 25,
 30–33, 35, 37, 43–45, 54, 56–60, 65–69, 73, 74,
 77, 80, 82, 100, 114, 117, 120, 121, 128, 138,
 183, 194, 196, 197, 217, 221, 222, 316
 Illumina HiSeq 20, 66, 197
 Illumina MiSeq 19, 20,
 30, 32, 58, 65, 67, 69, 77, 120, 121, 222
 Illumina Nextera XT 100
 Inflammatory Bowel Disease (IBD) 315–317
 Internal transcribed spacer (ITS) 31, 140, 232
 International Union of Biochemistry and
 Molecular Biology (IUBMB) 292
 International Union of Pure and Applied
 Chemistry (IUPAC) 292
 iPath 293
 Isotope label 89, 91, 95
 iVirus 22

J

Jaccard index 119, 124, 250
 Joint Genome Institute Integrated Microbial
 Genomes (JGI IMG) 171

K

KEGG Orthology (KO) 171, 268
 Keystone taxa 245, 246
 Khmer 181, 187, 189
 k-mers 180, 184, 187–190
 Kruskal-Wallace 160
 Kyoto Encyclopedia of Genes and Genomes
 (KEGG) 171,
 174–176, 198, 208, 232, 268, 292, 293, 302,
 303, 305, 306

L

Lachnospiraceae 50, 294
 Lactation 65, 71–73
 LC/MS 89, 93
 Library preparation 2, 4, 5, 8,
 10–12, 19, 35, 100, 183, 196, 231, 238
 Limma-voom 151, 157–159
 Linux 33, 35, 75, 132,
 170, 248, 269, 295
 Long-read sequencing 216
 Low biomass samples 1, 8, 10
 Lyophilization 73
 Lysis 2, 3, 9, 75,
 80, 100, 102, 103, 106, 107, 109
 Lysis buffer 3, 9, 75, 80, 106, 107, 109

M

MacPorts 269
 MacQIIME 75
 MAFFT 118, 123
 MANOVA 239
 Mann-Whitney 239
 Marker gene survey 175
 Markov chain Monte Carlo (MCMC) 269,
 271–273, 275, 282–288
 Markov clustering (MCL) algorithm 246
 Mass spectrometry 87–97
 Maximum likelihood 173, 176
 Maximum parsimony 176
 Media 50–55, 58, 59, 189
 MEGAHIT 25, 181,
 183–185, 188, 189, 222
 MetaBat 181, 185–187,
 190, 219, 220
 Metabolic pathways 291–306
 Metabolomics 291–306

- Metadata 40–42, 79, 115, 116, 118–120, 123–126, 128, 300, 301
- Metagenome analyzer (MEGAN) 22, 23, 189, 232
- MetagenomeSeq 144–145, 151, 156–158
- Metagenomics 1, 2, 4–6, 10, 11, 17, 21, 22, 24, 99, 131, 144, 147, 163, 165, 166, 169, 170, 179, 180, 185, 186, 188–190, 210, 215–224, 232, 267–317
- MetaPhlAn 232
- Metaproteomics 90
- MetaQuast 181, 185
- Metatranscriptomics 193–212, 293, 300, 304
- Meta-Velvet 222
- Metavir 22
- MetE 88, 91, 94, 96, 97
- MetH 88, 90, 91, 94, 96, 97
- MG-RAST 22, 23, 190, 232
- Microarray 156–158, 227, 231, 238
- Microbiome helper 131–141, 174
- Microspheres 2–7, 13
- Minimus2 219
- Mitochondria 31, 77, 81, 82
- Mock community 77, 82
- Moleculo 216
- Monte-Carlo 204–206, 210
- Morphotype 55
- Mothur 43–45, 58, 131, 146, 171, 232
- Multidimensional scaling 163
- Multinomial distribution 268
- Multiple displacement amplification (MDA) 4, 11–12, 25, 100–103, 106–110
- MultiQC 181, 188
- Multi-response Permutation Procedures (MRPP) 239

N

- N50 185
- Nasal swab 49, 50, 59
- NCBI 22, 96, 198, 298, 304
- Nearest-neighbor degree 257
- Nearest sequenced taxon index (NSTI) 174, 176
- Negative binomial 150, 153, 154, 164, 310
- Networks 227–240, 243–265, 292, 293, 303
- Newick format 172
- Nextera XT kit 20
- Non-metric multidimensional scaling 163
- Normal distribution 310
- Normalization 143–167, 173, 188, 189, 201, 223, 231, 232, 238, 239, 310
- npSeq 147, 151, 156, 157
- Nugent score 282

O

- Open reading frames (ORFs) 23, 25, 198, 199
- Operational taxonomic units (OTUs) 38–43, 45, 46, 48, 58, 77–79, 82, 83, 117, 119, 124, 126, 128, 131, 133, 134, 136–141, 147, 161, 162, 171–176, 232, 233, 246–254, 256–259, 261, 264, 265, 270–275, 279, 281, 282, 286, 288, 315
- Optimization 81, 82, 93, 245, 312, 314

P

- PacBio 20, 216
- Paired-End reAd mergeR (PEAR) 133, 135
- Paired-end sequences/sequencing 30, 77, 114, 183
- Pantothenate/CoA pathways 302, 303, 305, 306
- Parasite 29
- Pathway enrichment analysis 301
- Pathways 193, 198, 208, 228, 301–306
- Pearson correlation 244, 245
- Pelagic community 271
- Peptides 22, 88–97
- Perl 195, 200, 209, 269, 278, 279, 281, 284, 287
- PERMANOVA 119, 125, 165
- Permutation 239, 251
- Phenotype 55, 231, 235
- PHRED 128
- Phylogenetic diversity (PD) 118, 124–126
- Phylogenetic screening 101–103, 107–108, 110
- Phylogeny/phylogenetic tree 17, 41, 42, 57, 78, 118, 119, 123–124, 126, 170, 172, 173, 176
- Phyloseq 34, 41, 42, 58, 151, 153, 154, 157, 163
- PICRUSt 169–176
- Plasmids 209, 224
- Poisson distribution 314
- Polymerase 33, 43, 54, 56, 57, 80, 81, 107
- Polymerase chain reaction (PCR) 2, 3, 5, 12, 24, 29, 30, 33, 35, 54–59, 65–69, 74, 76, 77, 80, 81, 100, 102, 103, 107, 108, 113, 135, 136, 144, 152, 166, 167
- barcodes 57
- bias 30
- chimeric sequences 58
- primers 69, 144, 167
- Polymicrobial synergism 243
- Posterior probability (PP) 271, 274, 275, 279, 280, 282, 286, 288
- Power-law distribution 259
- Powersoil Kit 34
- pplacer 172
- PRIAM 293, 295, 298, 299, 304, 305
- Principal component analysis (PCA) 163, 202, 204

- Principal coordinate analysis (PCoA) 69, 79, 118, 119, 124, 125, 139, 140, 163, 175
- PRINSEQ 231
- Probabilistic graphical models (PGMs) 245
- Prodigal 25, 187
- Prokaryotic Virus Orthologous Groups (pVOGs) 23
- Proportion of unexpected correlations (PUC) 235–237
- Protein sequences
- Proteobacteria 67, 68, 70, 123, 224, 294, 302
- Proteomics 87, 89
- Protist 45, 100, 140
- Pseudomonas aeruginosa* 58
- PuMA 232
- p-values 152, 154, 161, 205, 210, 211, 234, 235, 251–253, 259, 301, 305, 310, 311, 314
- Python 23, 78, 118, 122, 174, 295–299, 304
- Q**
- qgraph 247–249, 252, 253, 258, 259, 261, 262, 264
- Quantitative Insights into Microbial Ecology (QIIME) 34, 40, 43–46, 58, 78, 79, 114–120, 122–126, 128, 131, 133–139, 146, 171, 174, 175, 232
- Quantitative Insights Into Microbial Ecology version 2 (QIIME2) 113–128
- Quantitative PCR (qPCR) 68, 166, 202
- QUAST 181, 185
- Qubit 4, 10, 12, 19, 20, 75, 81, 196
- Q-values 253
- R**
- Random network 244, 256, 259
- Rarefaction/rarefying 34, 41, 42, 46, 78, 83, 119, 126, 128, 138–140, 146, 147, 159, 175
- Ray Meta 25, 189, 222
- Rcpp 269
- Read mapping 194, 218, 221, 222
- Reference genome 169–172, 187, 197
- RefineM 220
- Regression 165, 245, 310, 311
- Regular networks 244, 256, 259
- Replication 114, 201, 224, 276, 279–282, 305
- Resampling 146–147, 156
- Respiratory tract 49–60
- Rfam 176
- Ribosomal Database Project (RDP) 44, 45, 75, 78, 136
- RNA extraction 196
- RNA preservation 196
- RNA-Seq 147, 150, 156, 157, 193–212, 231, 232
- rRNA depletion/mRNA enrichment 196
- R statistical programming language 195
- RStudio 34, 41, 151, 209, 246, 248, 249
- S**
- Sample metadata 41, 42, 115, 120
- Sampling/sequencing depth 5, 6, 17–25, 34, 50, 51, 72, 73, 83, 119, 122, 124, 126, 138, 140, 147, 150, 169, 179, 183, 185, 188, 201, 218, 238, 247, 255, 282, 284, 285, 288, 310
- SAMseq 147, 151, 156
- Scale-free networks 244, 255, 259
- Scythe 20
- Sediment 13, 101, 103–104, 109
- SEED 195, 198–200, 208, 209, 211, 232
- SEED reference database 195
- Selected reaction monitoring (SRM) 87–97
- Selective media 50, 53
- Sequence alignment 45, 113, 118
- Sequence assembly 77, 188, 222
- Sequence reads 36, 117, 180, 183, 186, 188, 190, 221, 222
- Sequencing depth 83, 119, 122, 124, 126, 138, 140, 167, 179, 183, 185, 188, 238
- Shannon diversity 119, 126
- Short read archive 120
- Shotgun sequencing 99, 101, 144
- Sickle 19–21, 217, 221
- SILVA database 31, 34, 46, 140, 170
- Simulation 147–150, 255–256, 314–315
- Single-cell genomics 99–110
- Skyline 90–95, 97
- Sludge 103, 104
- Small-world networks 244
- SOAPdenovo 184
- SortMeRNA 133, 134, 137
- SPAdes/metaSPAdes 19, 21, 188, 189
- Sparse Correlations for Compositional data (SparCC) 245, 247, 253–254
- Sparse treatment-effect model for identifying taxa (STEMIT) 315, 317
- Spearman correlation 239
- SpiecEasi 245, 247, 248, 253–254

Sputum	51, 54, 55, 58, 59	Tryptic peptide	89–91, 95
16S rRNA	1, 2, 5, 17, 24, 29–31, 50, 54, 56–57, 59, 74–77, 79, 80, 99, 101, 102, 108, 113–129, 131–141, 144, 152, 172	Tuxedo	231
18S rRNA	30, 31, 34, 133, 139	U	
STAMP	134, 139, 140, 174, 175	UCLUST	45
Statistical analysis of microarrays (SAM) file	186, 219	UniFrac	78, 118, 119, 124, 125, 127, 134, 139, 140
Stool	34, 35, 64–69, 79, 140, 150	Unipept	90, 91
Storage	6, 8, 73–74, 96, 186	UNITE database	140
Subsurface	1–13	UNIX	35, 134, 194, 195, 199, 217, 218, 269
Succinicolasticum	161, 162	UPARSE	75, 77, 83
SUMACLUST	45, 133		
Supervised learning	268	V	
Swiss-Prot	292, 297, 298	Variable regions	30, 31, 34, 39, 45, 56, 57, 135
Systems biology	227–240	Viral metagenomes (Virome)	17–25
T		Viral particles	19
Tax4Fun	170	VirtualBox (VBox)	33, 132, 140
Taxonomic assignment/taxonomic classification	22, 23, 25, 30–32, 35, 39–41, 45, 78, 107, 113, 114, 118, 122, 123	Visualization	13, 32, 34, 116, 117, 119, 120, 122–126, 128, 134, 138–139, 246, 260–263, 293, 295, 301–303
Taxonomic classification	22, 23, 25, 30–32, 35, 39–41, 45, 107, 113, 114, 118, 122, 123	Vitamin B ₁₂	87–97
Test data	268, 269, 271–273, 275, 281, 287	VSEARCH	34, 35, 37–39, 44, 133
Time series	73, 216, 219, 221, 271	W	
Training data	45, 46, 268, 271–273, 275–282, 286, 287	Welch's t-test	206, 211
Transkingdom network (TransNet)	228, 230	Whole-genome amplification (WGA)	11, 100–103, 106, 107, 109, 110
Transkingdom networks	227–240	Wilcoxon	160, 161, 211, 239
Treatment-effect model	309–317	X	
Trichomonas	30, 31	Xcode	269
Trimmomatic	20, 33, 35, 44, 181, 183, 187, 188	Z	
Trinity	231	Zero-inflated Gaussian (ZIG)	156, 157