UNIVERSITY OF CALIFORNIA, SAN DIEGO

Making pipelines and large processing microbial community studies available to any user, any time, any place

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Computer Science

by

Jose Antonio Navas Molina

Committee in charge:

Professor Rob Knight, Chair
Professor Nuno Bandeira
Professor Vineet Bafna
Professor Pieter Dorrestein
Professor Larry Smarr

2018

The dissertation of Jose Antonio Navas Molina is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2018

DEDICATION

To my family, and specially to my wife, Embriette Hyde.

EPIGRAPH

First, solve the problem. Then, write the code

—John Johnson

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AGP	American Gut Project
AMI	Amazon Machine Image
AWS	Amazon Web Services
BIOM	Biological Observation Matrix
BLAST	Basic Local Alignment Search Tool
bp	base pairs
CLI	Command Line Interface
EBI ENA	European Bioinformatics Institute's European Nucleotide Archive
EC2	Elastic Compute Cloud
EMP	Earth Microbiome Project
EMPO	Earth Microbiome Project Ontology
ENVO	Environment Ontology
GNPS	Global Natural Products Social Molecular Networking
GUI	Graphical User Interface
HMP	Human Microbiome Project
IaaS	Infrastructure as a Service
IBD	Inflamatory Bowel Disease

ITS	Internal Transcribed Spacer
LKM	Loadable Kernel Module
MDS	Multidimensional Scaling
MIMARKs	Minimum information about a marker gene sequence
NCBI	National Center for Biotechnology Information
NGS	Next Generation Sequencing
NIH	National Institutes of Health
ONT	Oxford Nanopore Technologies
OOM	Out-Of-Memory
OTU	Operational Taxonomic Unit
PCoA	Principal Coordinate Analysis
PCR	Polymerase Chain Reaction
PD	Phylogenetic Diversity
QIIME	Quantitative Insights into Microbial Ecology
rRNA	ribosomal ribonucleic acid
SCFGs	stochastic context-free grammars
SOP	Standard Operating Procedures
SVG	Scalable Vector Graphics
VAMPS	Visualization and Analysis of Microbial Population Structures

VM	Virtual Machines

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

The work presented in this thesis would not have been possible without the incredible people that have been supporting me.

First of all, I would like to thank my advisor, Rob Knight. His vision on the future and his excitement about it has been contagious, and was an incredible source of motivation. His massive network of collaborators and his willingness to introduce young researchers to it gave me the opportunity to work on amazing projects with brilliant people. His support through all these years has been critical for the success of my thesis.

I would also like to thank my committee, for their comments, feedback, incredible advice and their willingness to participate in this thesis (Nuno Bandeira, Vineet Bafna, Larry Smarr and Pieter Dorrestein).

I would like to thank each and every one of my lab mates, because they have all helped me at one point or another during my time in the Knight Lab. However, a few members (or ex-members) deserve a speacial shout-out. Jose Carlos Clemente Litran opened the doors of his house for me when I just landed for first time in the United States. His support, help, conversations and kindness during such a difficult time of my life gave me enough energy to overcome such challenge. Antonio Gonzalez and Yoshiki Vazquez Baeza, for sharing so many happy hours, sports events and other festivities so we can wash out some of the stress with beer or "fruity drinks". Jeff DeReus, for being a great support during the last two years

of the this hard process, and for introducing me to CrossFit, my current way of turning my brain off from work.

I am extremely grateful to my parents, Jose Navas Garcia and Maria Josefa Molina Alonso, who taught me all the values that make me the person that I am nowadays; and they didn't doubt for a second to support me on this american adventure even though it meant I was going to be far from them.

My sister Susana Navas Molina and her husband Xavier Garcia Jané have shown an incredible amount of support and love during this incredible experience. They made sure that I had an awesome time everytime that I got back home, and prepared the most amazing wedding that I could imagine.

Last but not least, I would like to thank a special member of the Qiime Forum that her name caught my eye since the very beginning: Embriette Hyde. I have been so lucky that you decided to join the Knight Lab and opened your soul and your heart to me. It is incredible how our frienship evolved so quickly into a beautiful romance and now I am the happiest man in the world because I can call you my wife. Embriette Hyde, I love you from the deepest part of my heart and I am very thankful for all the inconditional support during my PhD. Thanks for taking such a good care of me on those long nights and making sure that I get my butt of the chair to go to CrossFit, or run a half marathon, or a simple outdoors hike so that I don't go insane. I love you!

Section 1.1, in full, reproduces the material as it appears in "The microbiome and big data". J. A. Navas-Molina, E. R. Hyde, J. G. Sanders and R. Knight.

Current Opinion in Systems Biology, 2017, DOI: 10.1016/j.coisb.2017.07.003.

Section 2.1, in full, reproduces the material as it appears in "Advancing our understanding of the human microbiome using QIIME". J. A. Navas-Molina, J. M. Peralta-Sanchez, A. Gonzalez, P. J. McMurdie, Y. Vazquez-Baeza, Z. Xu, L. K. Ursell, C. Lauber, H. Zhou, S. J. Song, J. Huntley, G. L. Ackermann, D. Berk-Lyons, S. Holmes, J. G. Caporaso and R. Knight. *Methods in Enzymology*, 2013, DOI: 10.1016/B978-0-12-407863-5.00019-8.

Section 2.2.1 has been adapted from the original publication in "Subsampled open-reference clustering creates consistent, comprehensive OTU definitions and scales to billions of sequences". J. R. Rideout, Y. He, J. A. Navas-Molina, W.A. Walters, L. K. Ursell, S. M. Gibbons, J. Chase, D. McDonald, A. Gonzalez, A. Robbins-Pianka, J. C. Clemente, J. A. Gilbert, S. M. Huse, H. W. Zhou, R. Knight and J. G. Caporaso. *PeerJ*, 2014, DOI: 10.7717/peerj.545.

Section 2.2.2 has been adapted from the original publication in "Open-source sequence clustering methods improve the State of the Art". E. Kopylova, J. A. Navas-Molina, C. Mercier, Z. Z. Xu, F. Mahe, Y. He, H. Zhou, T. Rognes, J. G. Caporaso, R. Knight *mSystems*, 2016, DOI: 10.1128/mSystems.00003-15

Section 2.2.3 has been adapted from the original publication in "Deblur rapidly resolves single-nucleotide community sequence patterns". A. Amir, D. McDonald, J. A. Navas-Molina, E. Kopylova, J. T. Morton, Z. Z. Xu, E. P. Kightley, L. R. Thompson, E. R. Hyde, A. Gonzalez, R. Knight *mSystems*, 2017, DOI: 10.1128/mSystems.00191-16

Section 2.3.1 has been adapted from the original publication in "The oral and skin microbiomes of captive Komodo dragons are significantly shared with their habitat". E.R. Hyde, J. A. Navas-Molina, S. J. Song, J. G. Kueneman, G. Ackermann, C. Cardona, G. Humphrey, D. Boyer, T. Weaver, J. R. Mendelson, V. J. McKenzie, J. A. Gilbert, R. Knight *mSystems*, 2016. DOI: 10.1128/mSystems.00046-16

Section 2.3.2 has been adapted from the original publication in "A communal catalogue reveals Earth's multiscale microbial diversity". *Nature* L. R. Thompson, J. G. Sanders, D. McDonald, A. Amir, J. Ladau, K. J. Locey, R. J. Prill, A. Tripathi, S. M. Gibbons, G. Ackermann, J. A. Navas-Molina, S. Janssen, E. Kopylova, Y. Vazquez-Baeza, A. Gonzalez, J. T. Morton, S. Mirarab, Z. Z. Xu, L. Jiang, M. F. Haroon, J. Kanbar, Q. Zhu, S. J. Song, T. Kosciolek, N. A. Bokulich, J. Lefler, C. J. Brislawn, G. Humphrey, S. M. Owens, J. Hampton-Marcell, D. Berg-Lyons, V. McKenzie, N. Fierer, J. A. Fuhrman, A. Clauset, R. L. Stevens, A. Shade, K. S. Pollard, K. D. Goodwin, J. K. Jansson, J. A. Gilbert, R. Knight, The Earth Microbiome Project Consortium, 2017. DOI: 10.0.4.14/nature24621

Section 2.3.3, in part, has been submitted for publication of the material as it may appear in Science, 2018, D. McDonald, E. R. Hyde, J. W. Debelius, J. T. Morton, A. Gonzalez, G. Ackermann, A. A. Aksenov, B. Behsaz, C. Brennan, Y. Chen, L. DeRight Goldasich, P. C. Dorrestein, R. R. Dunn, A. K. Fahimipour, J. Gaffney, J. A Gilbert, G. Gogul, J. L. Green, P. Hugenholtz, G. Humphrey, C. Huttenhower, M. A. Jackson, S. Janssen, D. V. Jeste, L. Jiang, S. T. Kelley, D.

Knights, T. Kosciolek, J. Ladau, J. Leach, C. Marotz, D. Meleshko, A. V. Melnik, J. L. Metcalf, H. Mohimani, E. Montassier, J. A. Navas-Molina, T. T. Nguyen, S. Peddada, P. Pevzner, K. S. Pollard, G. Rahnavard, A. Robbins-Pianka, N. Sangwan, J. Shorenstein, L. Smarr, S. J. Song, T. Spector, A. D. Swafford, V. G. Thackray, L. R. Thompson, Y. Vazquez-Baeza, A. Vrbanac, P. Wischmeyer, E. Wolfe, Q. Zhu, The American Gut Consortium, R. Knight.

Section 2.3.4 has been adapted from the original publication in "Correcting for microbial blooms in fecal samples during room-temperature shipping". *mSystems* A. Amir, D. McDonald, J. A. Navas-Molina, J. Debelius, J. T. Morton, E. R. Hyde, A. Robbins-Pianka, R. Knight 2017. DOI: 10.1128/mSystems.00199-16

Section 3.1, in full, reproduces the material as it appears in "Addressing memory exhaustion failures in Virtual Machines in a cloud environment". J. A. Navas-Molina, S. Mishra. *43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2013, DOI: 10.1109/DSN.2013.6575330.

Section 3.2, in full, reproduces the material as it appears in "CUDSwap: Tolerating Memory exhaustion failures in cloud computing". J. A. Navas-Molina, S. Mishra. *International Conference on Cloud and Autonomic Computing (ICCAC)*, 2014, DOI: 10.1109/ICCAC.2014.12.

Section 4.1, in part, has been submitted for publication of the material as it may appear in Nature Methods, 2018, A. Gonzalez, J. A. Navas-Molina, T. Kosciolek, D. McDonald, Y. Vazquez-Baeza, S. Janssen, A. D. Swafford, S. B. Orchanian, J. G. Sanders, J. Shorenstein, H. Holste, S. Petrus, A. Robbins-Pianka,

C. J. Brislawn, M. Wang, J. R. Rideout, E. Bolyen, M. Dillon, J. G. Caporaso, P. C. Dorrestein, R. Knight.

Section 5.1, in full, reproduces the material as it appears in "From sample to multi-omics conclusions in under 48 hours". R. A. Quinn, J. A. Navas-Molina, E. R. Hyde, S. J. Song, Y. Vazquez-Baeza, G. Humphrey, J. Gaffney, J. J. Minich, A. V. Melnik, J. Herschend, J. DeReus, A. Durant, R. J. Dutton, M. Khosroheidari, C. Green, R. da Silva, P. C. Dorrestein, R. Knight *mSystems*, 2016, DOI: 10.1128/mSystems.00038-16

VITA

2012	B. S. in Informatics Engineering, Universitat Politècnia de Catalunya, Barcelona
2013	M. Sc. in Computer Science, University of Colorado at Boulder, Boulder
2018	Ph. D. in Computer Science, University of California, San Diego

PUBLICATIONS

Author names marked with † indicate shared first co-authorship.

J. A. Navas-Molina, E. R. Hyde, J. G. Sanders, R. Knight. "The microbiome and big data", *Current Opinion in Systems Biology*, 2017, DOI: 10.1016/j.coisb.2017.07.003.

J. A. Navas-Molina, J. M. Peralta-Sánchez, A. González, P. J. McMurdie, Y. Vázquez-Baeza, Z. Xu, L. K. Ursell, C. Lauber, H. Zhou, S. J. Song, J. Huntley, G. L. Ackermann, D. Berg-Lyons, S. Holmes, J. G. Caporaso, R. Knight. "Advancing our understanding of the human microbiome using QIIME", *Methods in Enzymology*, 2013, DOI: 10.1016/B978-0-12-407863-5.00019-8.

E. Kopylova, **J. A. Navas-Molina**, C. Mercier, Z. Xu, F. Mahé, Y. He, H. Zhou, T. Rognes, J. G. Caporaso, R. Knight. "Open-source sequence clustering methods improve the state of the art", *mSystems*, 2017, DOI: 10.1128/mSystems.00003-15.

J. A. Navas-Molina, S. Mishra. "Addressing memory exhaustion failures in Virtual Machines in a cloud environment", *43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2013, DOI: 10.1109/DSN.2013.6575330.

J. A. Navas-Molina, S. Mishra. "CUDSwap: Tolerating Memory exhaustion failures in cloud computing", *International Conference on Cloud and Autonomic Computing*, 2014, DOI: 10.1109/ICCAC.2014.12.

A. Amir, D. McDonald, **J. A. Navas-Molina**, J. Debelius, J. T. Morton, E. R. Hyde, A. Robbins-Pianka, R. Knight. "Correcting for microbial blooms in fecal samples during room-temperature shipping", *mSystems*, 2017, DOI: 10.1128/mSystems.00199-16.

A. Amir, D. McDonald, **J. A. Navas-Molina**, E. Kopylova, J. T. Morton, Z. Xu, E. P. Kightley, L. R. Thompson, E. R. Hyde, A. González, R. Knight. "Deblur rapidly resolves single-nucleotide community sequence patterns", *mSystems*, 2017, DOI: 10.1128/mSystems.00191-16

L. R. Thompson, J. G. Sanders, D. McDonald, A. Amir, J. Ladau, K. J. Locey, R. J. Prill, A. Tripathi, S. M. Gibbons, G. Ackermann, **J. A. Navas-Molina**, S. Janssen, E. Kopylova, Y. Vázquez-Baeza, A. González, J. T. Morton, S. Mirarab, Z. Xu, L. Jiang, M. F. Haroon, J. Kanbar, Q. Zhu, S. J. Song, T. Kosciolek, N. A. Bokulich, J. Lefler, C. J. Brislawn, G. Humphrey, S. W. Owens, J. Hampton-Marcell, D. Berg-Lyons, V. McKenzie, N. Fierer, J. A. Fuhrman, A. Clauset, R. L. Stevens, A. Shade, K. S. Pollard, K. D. Goodwin, J K. Jansson, J. A. Gilbert, R. Knight, The Earth Microbiome Project Consortium. "A communal catalogue reveals Earthś multiscale microbial diversity". *Nature*, 2017, DOI: http://dx.doi.org/10.1038/nature24621.

†R. A Quinn, †**J. A. Navas-Molina**, †E. R Hyde, S. Jin Song, Y. Vázquez-Baeza, G. Humphrey, J. Gaffney, J. J Minich, A. V Melnik, J. Herschend, J. DeReus, A. Durant, R. J Dutton, M. Khosroheidari, C. Green, R. da Silva, P. C Dorrestein, R. Knight "From sample to Multi-Omics conclusions in under 48 Hours", *mSystems*, 2016, DOI: 10.1128/mSystems.00038-16.

E. R. Hyde, **J. A. Navas-Molina**, S. J. Song, J. Kueneman, G. Ackerman, C. Cardona, G. Humphrey, D. Boyer, T. Weaver, J. Mendelson, V. McKenzie, J. Gilbert, R. Knight. "The oral and skin microbiomes of captive Komodo dragons are significantly shared with their habitat", *mSystems*, 2016, DOI: 10.1128/mSystems.00046-16.

J. R. Rideout, Y. He, , **J. A. Navas-Molina**, W. A. Walters, L. K. Ursell, S. M. Gibbons, J. Chase, D. McDonald, A. González, A. Robbins-Pianka, J. C. Clemente, J. A. Gilber, S. M. Huse, H. W. Zhou, R. Knight, J. G. Caporaso. "Subsampled open-reference clustering creates consistent, comprehensive OTU definitions and scales to billions of sequences". *PeerJ*, 2014, DOI: 10.7717/peerj.545.

The following publications were not included as part of this dissertation, but were also significant byproducts of my doctoral training.

Y. Vázquez-Baeza, A. Gonzalez, L. Smarr, D. McDonald, J. T. Morton, **J. A. Navas-Molina**, R. Knight. "Bringing the Dynamic Microbiome to Life with Animations", *Cell Host and Microbe*, 2017, DOI: 10.1016/j.chom.2016.12.009.

J. T Morton, J. Sanders, R. A Quinn, D. McDonald, A. Gonzalez, Y. Vázquez-Baeza, **J. A. Navas-Molina**, S. Jin Song, J. L Metcalf, E. R Hyde, M. Lladser,

P. C Dorrestein, R. Knight. "Balance trees reveal microbial niche differentiation". *mSystems*, 2017, DOI: 10.1128/mSystems.00162-16.

S. Shalapour, X.J. Lin, I. N. Bastian, J. Brain, A. D. Burt, A. A. Aksenov, A. F. Vrbanac, W. Li, A. Perkins, T. Matsutani, Z. Zhong, D. Dhar, **J. A. Navas-Molina**, J. Xu, R. Loomba, M. Downes, R. T. Yu, R. M. Evans, P. C. Dorrestein, R. Knight, C. Benner, Q. M. Anstee, M. Karin. "Inflammation-induced IgA+ cells dismantle anti-liver cancer immunity", *Nature*, 2017, DOI: doi:10.1038/nature24302

ABSTRACT OF THE DISSERTATION

Making pipelines and large processing microbial community studies available to any user, any time, any place

by

Jose Antonio Navas Molina

Doctor of Philosophy in Computer Science

University of California, San Diego, 2018

Professor Rob Knight, Chair

Advances in 'omics technologies are producing vast amounts of data, bringing microbiome research to a whole new level. This increase in data is pushing the limits of existing analysis tools, creating a rapidly-changing environment in which new tools are constantly being released. This presents a challenge to researchers, who need to constantly learn new analytical tools, expose themselves to new environments such as cloud computing or supercomputers, and deal with the problems resulting from a heterogeneous environment lacking the enforcement of standards.

This thesis demonstrates how computational optimizations, enforcement of standards, and minimizing the learning curve for analytical tools and computational environments empower researchers to push the microbiome field forward.

Chapter 1 motivates and contextualizes the thesis, exposing the challenges and opportunities that current microbiome research faces as it presents itself as a big data field. Next, Chapter 2 presents the first gold standard approach for analyzing microbiome data, improvements in analytical tools, and examples of how these improvements move microbiome research forward. Chapter 3 describes a system that lowers the access barrier to cloud computing that researchers without a computational background face. Chapter 4 exposes the importance of meta-analyses to increase researchers' ability to discover new findings and how much effort is currently spent to perform such meta-analyses. This chapter also presents Qiita, a web-based system focused on facilitating meta-analyses by enforcing standards, normalizing data representation and processing, and providing a common interface to current state-of-the-art analysis tools. Chapter 5 describes how using the tool improvements and data standardizations presented in Chapters 2 and 4, respectively, speed up the process of analyzing microbiome samples to levels never reached before. Finally, the concluding chapter of this thesis discusses the results and the opportunities opened due to these advances, paying special attention to precision medicine, a topic in which the microbiome is becoming key.

Chapter 1

What is the microbiome and why is it important?

The microbiome is the group of microorganisms that live in, on, and around us. Understanding the interactions between these microorganisms and their niche can lead to advances in a wide variety of research areas, including, but not limited to human and pet health [31, 57, 205], forensics [50, 80], climate change [195] and pharmaceuticals [149, 128].

Researchers have at their disposal multiple technologies that allow them to make different inquiries about the microbial system that they are investigating. Target gene sequencing (such as 16S/18S ribosomal ribonucleic acid (rRNA) or Internal Transcribed Spacer (ITS) marker gene sequencing surveys) allows researchers to survey the different bacteria and Archaea present in a sample. Shotgun metagenomics provides a view of the functional potential of the microbial commu-

nity, and metatranscriptomics confirms gene expression at a specific time point. Metabolomics surveys provide a view of the small molecules contained in a sample, either from microbial or host origin. The combination of the results of these technologies is crucial to establish correlations between the microorganisms and their environment. The vast amount of data generated by these technologies requires fast and efficient resources and tools for effective data analysis. The work presented in this thesis is motivated by the challenges that microbial ecologists are facing when analyzing such large microbiome datasets, some of which are summarized in the following section.

Section 1.1, in full, reproduces the material as it appears in "The microbiome and big data". J. A. Navas-Molina, E. R. Hyde, J. G. Sanders and R. Knight. *Current Opinion in Systems Biology*, 2017, DOI: 10.1016/j.coisb.2017.07.003. The dissertation/thesis author was the primary author of this paper.

1.1 The microbiome and big data

Microbiome datasets have expanded rapidly in recent years. Advances in DNA sequencing, as well as the rise of shotgun metagenomics and metabolomics, are producing datasets that exceed the ability of researchers to analyze them on their personal computers. Here we describe what Big Data is in the context of microbiome research, how this data can be transformed into knowledge about microbes and their functions in their environments, and how the knowledge can be applied to move microbiome research forward. In particular, the development of new high-resolution tools to assess strain-level variability (moving away from Operational Taxonomic Unit (OTU)), the advent of cloud computing and centralized analysis resources such as Qiita (for sequences) and Global Natural Products Social Molecular Networking (GNPS) (for mass spectrometry), and better methods for curating and describing "metadata" (contextual information about the sequence or chemical information) are rapidly assisting the use of microbiome data in fields ranging from human health to environmental studies.

1.1.1 From cells to bits: what is big data in microbiome research?

Since the term "microbiome" was coined by Joshua Lederberg in 2001 [107], the microbiome research field has exploded both in terms of the heterogeneity of the data produced and in the amount of data generated. Early approaches to

characterizing the microbiome were based on targeted detection techniques in the laboratory, such as culturing and assays based on the Polymerase Chain Reaction (PCR), and assessed limited numbers of subjects (on the order of tens) [17]. The introduction of sequencing technologies revolutionized the field, enabling investigators to characterize microbial communities directly from primary samples. Historically, the 16S ribosomal ribonucleic acid (rRNA) gene, a marker gene that exists in all bacteria and archaea as an essential part of the ribosome, has been targeted for these sequence-based profiling efforts. Its ubiquity among bacteria and archaea and the low cost of the approach has made it the most widely used for microbiome profiling of samples. Similarly, amplification and sequencing of the 18S rRNA gene and the Internal Transcribed Spacer (ITS) permit investigators to profile the eukaryotic and fungal communities present in a sample using similar techniques. Since the introduction of Next Generation Sequencing (NGS), technologies have evolved from generating a few hundred thousand reads per run (454 GS) to tens of million reads (Illumina MiSeq) or even a few billion reads per run (Illumina HiSeq) [64]. Benchmarked protocols, such as those used by the Earth Microbiome Project (EMP) and widely adopted by researchers around the globe, facilitate meta-analyses of unprecedented size-investigators can combine studies, each with hundreds to thousands of samples, into a single large analysis effort. The precipitous drop in sample processing and sequencing costs associated with new technology development is enabling researchers to move beyond simple taxonomy and abundance-based work to species and strain level profiling as well as

descriptions of functional pathways through whole genome shotgun metagenomics sequencing. As a result, researchers are able to ask more critical questions of their samples and are utilizing other technologies, such as detection of small molecules via mass spectrometry, to confirm or refute hypotheses driven by functional pathway and gene abundance information obtained from shotgun sequencing data.

The rate at which these technologies are increasing their data output is faster than our computational power is growing [217], effectively shifting the costs of a research study from the sequencing pipeline to the data analysis pipeline. Additionally, as researchers utilize larger and larger datasets, they are able to design large-scale studies to ask (and answer) complex questions. The metadata associated with samples, therefore, is becoming an increasingly large contributor to microbiome big data and the challenges associated with streamlining data analysis. Standards such as Minimum information about a marker gene sequence (MIMARKs) [225] have helped investigators format their metadata to facilitate data analysis and data upload to repositories such as the European Bioinformatics Institute's European Nucleotide

Archive (EBI ENA). Nevertheless, as samples are increasingly processed in parallel with multiple different protocols (i.e., 16S, 18S, ITS, shotgun, metabolomics, etc.), correct formatting of metadata to capture this information and facilitate multi-omics correlative analyses will require careful attention and appropriate implementation of tools capable of handling hundreds to thousands of columns of

data for hundreds to thousands of samples. Tools such as Qiita [1] are being developed to address the challenges associated with analyzing large numbers of samples, processed via multiple different protocols, and with complex metadata-and these tools rely on both the availability and effective usage of large-scale compute resources. The ability to apply tools such as Quantitative Insights into Microbial Ecology (QIIME) in the cloud; e.g., using Amazon Web Services (AWS) [162], has broadened these capabilities far beyond the original user base, and enabled users in developing countries such as Bangladesh to use these tools without operating their own large-scale compute infrastructure. These techniques are now being applied in the United States through Illumina's BaseSpace [2] and NIH's Cloud Pilot [3].

1.1.2 From bits to knowledge: how is big data moving microbiome research forward?

Initial efforts to characterize and understand the healthy human microbiome using NGS techniques [196, 29] raised more questions than answers, and led to the explosion of microbiome research that has identified associations between the microbiome and diseases as varied as obesity, inflammatory bowel disease, cardiovascular disease, and autism (among many others). Most of these studies have simply identified associations and the question of causation or simple association remains unknown. Key studies, such as the obesity work done by Jeffrey

[1] `http://qiita.microbio.me`
[2] `https://basespace.illumina.com/home/index`
[3] `https://commonfund.nih.gov/bd2k/commons`

I. Gordon and his team at Washington University [201, 202, 169] and the personalized nutrition work done by Eran Segal of the Weizmann Institute [226] are coming closer to answering the question of causality versus association. However, it is becoming increasingly clear that integrating DNA sequence data with other omics techniques such as metatranscriptomics (sequencing the RNA), proteomics (sequencing the proteins), and metabolomics (characterizing the metabolites) will be key for advancing microbiome research. An example of the power of combining multiple techniques for assessing the microbiome is the National Institutes of Health (NIH) Human Microbiome Project (HMP), the largest human microbiome sequencing effort at the time of its publication in 2012. 16S rRNA gene amplicons were generated from total of 4788 samples collected from 242 healthy adults [196] and sequenced using 454 pyrosequencing. Additionally, a whole genome shotgun sequencing on the paired-end Illumina platform was performed on a subset of 681 samples, generating 2.9 Gigabases per sample (close to 2 terabytes of data for the entire dataset).

The HMP shotgun metagenomics data revealed a key observation: while no taxon was observed in all individuals (i.e., no core healthy microbiome was identified), the functional pathways inferred from the shotgun data were evenly distributed across individuals and body sites. While this was an important observation, the addition of other data types, such as RNA-seq or metabolomics would have provided precise information regarding the actual activity of the microbial community and which small molecules were present, respectively, further

exemplifying importance of combining different -omics techniques for generating hypotheses that ultimately lead to studies designed to obtain a more complete picture of a given microbial community (and the significance of its presence). For example, as reported by Bouslimani et al. [15], using a paired sequencing-mass spectrometry approach allowed the investigators to identify correlations between Propionibacterium genera and the presence of oleic acid, palmitic acid, mono-oleic, and palmitic acylated glycerols on human skin. Hypothesizing that Propionibacterium mediates the hydrolyzation of triacylglycerides or diacylglycerides from human acylated glycerols, Bouslimani et al. cultured Propionibacterium acnes in a medium supplemented with the triglyceride triolein and examined the resulting metabolic products, ultimately confirming their hypothesis.

Microbiome citizen science initiatives such as the American Gut Project (AGP) [4] have made significant contributions to the field by democratizing microbiome research and thus providing large-scale datasets that can be used as comparative frameworks for other studies. Citizens support the science by sending samples from their bodies, their pets, or their environment as well as the necessary funds to cover the sample processing. These projects face the challenge of dealing with large numbers of samples; while most current microbiome studies contain hundreds or a few thousand samples, these citizen science efforts contain a continually growing number of samples that in some cases are on the order of over ten thousand samples, pushing the limits of the current computational tools.

[4] `http://americangut.org`

Furthermore, this democratization is not free: subject data is self-reported, and at times, significant amounts of data are necessary to correctly characterize the sample source. The AGP currently collects up to 400 variables about study participants, including detailed dietary information proffered through a standardized food frequency questionnaire (VioScreen). Analyzing all these variables is a challenge, and one solution is crowd sourcing the data analysis itself. All de-identified AGP data are made public as soon as they are available, allowing researchers and clinicians around the world to use the data to identify correlations between those variables and the microbiome data which can generate new hypotheses, or to contextualize their own studies with the largest open source human microbiome dataset that currently exists. The power of meta-analyses is apparent from early work by Lozupone and Knight [120], in which 21,752 16S rRNA sequences from diverse environments sampled across 111 studies were analyzed together to find that the main environmental driver differentiating microbial communities was salinity, rather than temperature, humidity, or a number of other environmental factors. However, when we restrict the analysis to the human gut microbiome, technical factors that differ between studies, such as DNA extraction, PCR primers, and sequencing platform are often larger than the biological effects we seek to discover [122]. Performing similar large-scale meta-analyses with the AGP data and the hundreds of other publicly available human microbiome datasets will be critical for identifying universal microbiome signatures associated with different health and disease states, and for understanding which technical variables have

larger effect sizes than biological variables. Big Data has also proven critical in the context of microbial epidemiology. Using Mycobacterium tuberculosis as an example, Guthrie and Gardy [68] describe the utility of using NGS techniques for understanding disease outbreaks. Whole genome sequencing of a specific pathogen can reveal the infection path (including patient 0) of the outbreak by allowing investigators to follow mutations from several strains isolated from infected individuals. Whole genome sequencing can also be used to diagnose disease. For example, determination of antibiotic resistance of M. tuberculosis is a notoriously difficult clinical problem; current gold-standard diagnostic techniques are culture-based and can take up to 8 weeks to generate results. Whole genome sequencing can reduce this time to a few days when the mutations responsible for drug resistance are well characterized and the reference databases are high quality. As a byproduct, the usage of whole genome sequencing for outbreak tracking and rapid diagnostics generates a genome catalogue that can be used for new drug development as well as better disease characterization. Clinical sequencing and diagnostic timeframes are becoming even faster with the advent of nanopore sequencing technology, currently commercialized by Oxford Nanopore Technologies (ONT) through the MinION sequencer. The reads produced by ONT devices are longer but comparatively less accurate compared to other sequencing technologies; however, they are generated extremely rapidly and portably. Similar in size and price to a high-end smartphone, the MinION sequencer facilitates near-immediate data acquisition, meaning sequences can be generated much closer to the biological

source. Nanopore sequencers have been used to perform same-day diagnosis of tuberculosis [210] as well as in situ monitoring of an Ebola outbreak [159]. The speed and portability can also benefit non-epidemiological microbiome work by making field-based work where sample transit and storage are difficult to impossible more obtainable. The MinION has already been used for on-site microbiological surveys in Antarctica [82] and produced the first sequences generated in space aboard the International Space Station [25].

1.1.3 Looking to the future: opportunities and challenges

The tools and technologies that have enabled microbiome research thus far continue to improve at breakneck pace. Increased usage of fast, portable sequencers such as the MinION and of multi-omics techniques means that the amount of data collected by microbiome researchers will quickly reach never before seen sizes, which will pose challenges for data storage and analysis. This wealth of information also will facilitate the understanding of bacterial community mechanics and interactions like never before, leading to groundbreaking developments not only in human health [204, 31, 114], but also in agriculture [178], biofuels [75], and many other applications. One of the biggest challenges facing the field as investigators aim to achieve these goals is the ability to integrate and correlate the massive amounts of data produced by these protocols and to identify biologically relevant information that can be used to formulate testable hypotheses. As investigators begin to utilize and combine multi-omics technologies, they are faced with tools

and protocols that are at different stages of development. For example, one of the difficulties associated with mass spectrometry analysis of small molecules is that in many cases we are unable to determine whether molecules are microbial or host-derived due to lack of annotation, and if indeed derived from the microbiome, which specific group(s) of bacteria generated the chemical signature. Applying mass spectrometry techniques to more and more microbiome datasets will enable researchers to build the existing databases. Even among sequence data, biases exist towards well studied environments, such as the human gut, while less studied environments, such as coral reefs, are not represented accurately (Earth Microbiome Project, in review). Developing tools to cross-compare sequence and small molecule data is also a key challenge; many of the techniques to assess sequence data are phylogeny based and cannot be applied to mass spectrometry outputs. Additionally statistical approaches for assessing microbiome sequence data [138, 126] will need to be validated on mass spectrometry data, or new, appropriate tools will need to be developed. Finally, visualizing multi-omics data together in a clear, meaningful way poses an interesting challenge, particularly given that such tools will need to be able to process thousands of data points from thousands of samples. Large-scale meta-analyses, such as those described in the previous section, also pose a unique challenge. Current 16S rRNA studies contain tens of millions of reads, and the amount of data utilized in meta-analyses is likely to be orders of magnitude larger as shotgun sequence and metabolomics data become a routine part of microbiome studies. The largest known meta-analysis in existence,

performed on the first 27,742 samples from 91 different studies in the EMP [5] exposed key problems. First, the current tools utilized to analyze the data cannot handle more than 30,000 samples at a single time. Additionally, the importance of standardizing metadata also became crystal clear. Although standard metadata definitions exist, data repositories currently do not enforce their compliance, and the metadata normalization effort is shifted to the researcher performing the meta-analysis. New tools as well as more accurate documentation will be key to facilitate the adoption of the standards in the community.

Last but not least, one of the most important challenges that will face microbiome research in the near future is the translation of results from the laboratory to everyday life. The human body is a supra-organism containing a wide variety of microorganisms that provide up to 99% of the genetic material present in our bodies. Ignoring this part of the system when assessing the well-being of a person is akin to performing a routine physical but only checking the blood pressure of the patient. Although the ultimate goal of human microbiome research is to implement clinical microbiome surveys, there is much work to be done before this goal can be realized. First, and most importantly, more data need to be collected and analyzed. Well-designed studies on clinical cohorts will be key for identifying meaningful host-microbiome associations and how these associations can be leveraged to improve human health. Universal Standard Operating Procedures (SOP) will also be critical to minimize lab to lab variation [181], including protocols for

[5] `http://earthmicrobiome.org`

sample collection, handling, storage and processing, as well as standardizing analysis tools. Clinician education will also be critical to enable health care providers to understand the limits of microbiome research as well as the advantages, and easy to understand microbiome analysis reports will be a key part of clinician education. Finally, sample processing and analysis times and costs need to be reduced. While in some cases genomic analysis is more rapid than gold standard diagnostics, in many cases, the processing time and costs outweigh the advantages of these techniques. For example, RNA-seq remains a lengthy, complex approach. The MinION may be useful for addressing this issue as it is able to directly accept RNA without the requirement for cDNA generation; however, widespread use of this tool will likely be closely tied to a reduction in the current error rate suffered by the system.

Microbiome research is currently on the precipice of producing orders of magnitude more data than ever before. To accurately assess and utilize this data, investigators will rely on the development of tools, pipelines, and SOPs able to effectively handle big data. Together, researchers, clinicians, and computer scientists are poised to revolutionize microbiome research and its applications in human health, agriculture, food science, and a number of other critical fields.

Chapter 2

Analyzing large scale microbial community cohorts

Chapter 1 exposed some of the challenges that researchers face when analyzing the big datasets resulting from microbial community studies. These challenges are exacerbated in massive iniatives like the Earth Microbiome Project (EMP) [59, 58, 197] and the American Gut Project (AGP) [1].

The EMP is a collaborative effort of more than 500 investigators, and aiming to characterize the Earth's microbial life using amplicon sequencing, metagenomics and metabolomics. The goal of the EMP is to process up to 200,000 samples for 16S ribosomal ribonucleic acid (rRNA) marker gene sequencing, releasing the data as the data are generated prior to publication and crowd sourcing the data analysis. More than 60 publications have already resulted from data generated by the EMP,

[1] `http://americangut.org`

and these average 17 citations per paper per year [2]. Currently, the EMP is in its second phase, in which the researchers aim to generate the shotgun metagenomic assemblies and metabolomic profiles of 500 samples.

The AGP is the largest crowd sourced, crowd funded, citizen science project. It aims to open microbiome analysis to anyone, and to create a vast, free, microbiome dataset. In the AGP, the participants donate a sample (typically a stool sample, although the project is not limited to stool) as well as the financial contribution that supports processing, sequencing and analysis of the sample. AGP participants complete a questionnaire with dietary and lifestyle questions. The answers to the questionnaire are then stored in a standard set of columns, and are made publicly available together with the sequences, both de-identified.

Both of these initiatives are releasing a powerful framework to the research community. On the one hand, researchers around the globe can mine the datasets to find new trends and generate new hypotheses. On the other hand, they can use it to contextualize their own samples and increase the power of their own analyses by comparing them against these cohorts. The success of these type of initiatives, however, relies on the use of standardized practices for sample processing and analysis, to minimize technical differences between the samples. The EMP released standard protocols [3] for sample collection and preparation, which have been widely adopted by the community and cited around 2000 times [2]. Normalizing the data

[2] http://www.earthmicrobiome.org/publications/

[3] http://www.earthmicrobiome.org/protocols-and-standards/

analysis is as important as normalizing the data processing to minimize technical differences. Quantitative Insights into Microbial Ecology (QIIME) [20], an open-source pipeline for analyzing microbiome data developed in the Knight lab, is one of the most popular [4] tools for those analyses. With 155 scripts and over 1000 parameters, the risk of introducing technical differences during the data analysis step is high. Section 2.1, published in the journal *Methods in Enzymology, 2013*, contains the first gold standard approach for the analysis of microbial community datasets, as well as providing to the researchers with suggestions about how to minimize the introduction of technical differences while analyzing the data. As the first author of this publication, I co-wrote the text, generated the majority of the figures and wrote the IPython notebook [153] attached to the publication.

Section 2.1, in full, reproduces the material as it appears in "Advancing our understanding of the human microbiome using QIIME". J. A. Navas-Molina, J. M. Peralta-Sanchez, A. Gonzalez, P. J. McMurdie, Y. Vazquez-Baeza, Z. Xu, L. K. Ursell, C. Lauber, H. Zhou, S. J. Song, J. Huntley, G. L. Ackermann, D. Berk-Lyons, S. Holmes, J. G. Caporaso and R. Knight. *Methods in Enzymology*, 2013, DOI: 10.1016/B978-0-12-407863-5.00019-8.

[4] The original paper has been cited over 8,700 times according to Google Scholar at the moment of this writing

2.1 Advancing our understanding of the human microbiome using QIIME

High-throughput DNA sequencing technologies, coupled with advanced bioinformatics tools, have enabled rapid advances in microbial ecology and our understanding of the human microbiome. Quantitative Insights into Microbial Ecology (QIIME) is an open-source bioinformatics software package designed for microbial community analysis based on DNA sequence data, which provides a single analysis framework for analysis of raw sequence data through publication quality statistical analyses and interactive visualizations. In this paper, we demonstrate the use of the QIIME pipeline to analyze microbial communities obtained from several sites on the bodies of transgenic and wild-type mice, as assessed using 16S ribosomal ribonucleic acid (rRNA) gene sequences generated on the Illumina MiSeq platform. We present our recommended pipeline for performing microbial community analysis, and provide guidelines for making critical choices in the process. We present examples of some of the types of analyses that are enabled by QIIME, and discuss how other tools, such as phyloseq and R, can be applied to expand upon these analyses.

2.1.1 Introduction

Advances in DNA sequencing technologies, together with the availability of culture-independent sequencing methods and software for analyzing the mas-

sive quantities of data resulting from these technologies, have vastly improved our ability to characterize microbial communities in many diverse environments. The human microbiota, the collection of microbes living in or on the human body, is of considerable interest: microbial cells outnumber human cells in our bodies by a ratio of up to 10 to 1 [174]. These microbial communities contribute to healthy human physiology [35, 38, 191] and development [39, 90], and dysbiosis (or imbalance in these communities) is now known to be associated with disease, including obesity [199] and Crohn's disease [41]. More recently, evidence from transplants into germ-free mice suggests that some of these associations may be causal, because certain phenotypes can be transmitted by transmitting the microbiota [23, 133, 202], even including transmission of human phenotypes into mice [74, 93, 184].

Illumina's MiSeq and HiSeq DNA sequencing instruments respectively sequence tens of millions, or billions, of DNA fragments in a single sequencing run [96]. The rapidly increasing data volumes typical of recent studies drives a need for more efficient and scalable tools to study the human microbiome [61]. QIIME [20] is an open-source pipeline designed to provide self-contained microbial community analyses, from interacting with raw sequence data through publication-quality statistical analyses and visualizations.

QIIME integrates commonly used third-party tools, and implements many diversity metrics, statistical methods, and visualization tools for analyzing microbial data. Consequently, most individual steps in the microbial community analysis can be performed in multiple ways. Here, we describe how samples are prepared

for an Illumina MiSeq run, the QIIME pipeline, and our view of the current best practices for analyzing microbial communities with QIIME. Although there are other pipelines available, including mothur [177], the RDP tools [150, 151], ARB [123], Visualization and Analysis of Microbial Population Structures (VAMPS) [5], and other platforms, in this review we focus on analysis with the MiSeq platform and QIIME as this combination is increasingly popular as a method for analyzing microbial communities and a detailed comparison of other available pipelines and sequencing platforms is beyond the scope of the present work.

2.1.2 QIIME as integrated pipeline of third party tools

An early barrier to adoption of QIIME was that it was difficult to install, in part because of the large number of software dependencies (third party packages that need to be installed before QIIME is operational). The large number of dependencies was, however, a deliberate choice made during QIIME development. To build a pipeline for sequence analysis that encompasses the many steps from sequence collection, curation, and statistical analysis, the user must consider many existing tools that have been developed to perform specific functions, and extensively benchmarked on their ability to perform these functions, such as the UCLUST program for clustering sequences into Operational Taxonomic Unit (OTU) [43]. A pipeline thus has two options: either re-implement the algorithm, or use the existing software (by creating a wrapper that allows its input and output

[5] `http://vamps.mbl.edu`

to be incorporated into the pipeline). The QIIME developers choose to wrap all the algorithms rather than re-implement them. This choice preserves the integrity of the programs that make up the pipeline, as there is no doubt that the tool being used is the one designed, created, and tested by the original authors, and, in most cases, peer-reviewed by the scientific community. The reuse of existing software also allows the QIIME pipeline to include and distribute newly developed and improved algorithms more rapidly than would be possible if each algorithm had to be re-implemented and re-tested to check that it matched the original. Thus QIIME users can be sure that they have the most up-to-date tools for their analysis, and can credit the authors of the component software packages appropriately.

One important, but sometimes poorly understood, aspect of the QIIME pipeline is that it wraps algorithms and tools produced by other researchers into a single pipeline for sequence analysis. It is therefore important to cite the individual tools that you use as well as QIIME itself. For example, an analysis using the default QIIME parameters [20] would use UCLUST [43] to cluster the sequences against the GreenGenes database [37], assign taxonomy using the RDP classifier [213], and build Principal Coordinate Analysis (PCoA) beta diversity plots using UniFrac [118]. It is important for researchers who are considering contributing to the QIIME pipeline to recognize that their contributions will be cited, so that they can continue to expand upon their work. For example, the pick_otus.py script alone offers a choice of nine different clustering algorithms, each developed by researchers who should be acknowledged if their particular algorithm is used.

For taxonomy databases and other reference databases, including GreenGenes, it is also important to cite the release version that you are using [37], not least because the results will change depending on which release you used, and others may not be able to reproduce your results without this information. For GreenGenes, the default taxonomy database in QIIME, the version is named after the release date, such as the 12_10 release. The latest version of GreenGenes can always be downloaded from the qiime.org website. Using the same GreenGenes reference database version is critical for comparisons of taxonomy assignments and OTU across different studies. For this reason, all the studies in the QIIME database are always processed against the same release version of GreenGenes.

An overview of some of the key tools used by the default QIIME pipeline follows:

- UCLUST [43]. Used for OTU picking.
- USEARCH [43]. Used for OTU picking and chimera checking.
- RDP Classifier [213]. Used for taxonomy assignment.
- GreenGenes Database [37]. Used as a reference database for taxonomy assignment and reference-based OTU picking (see below).
- PyNAST [19]. Used for multiple sequence alignment.
- UniFrac [118]. Used as a phylogenetic metric for beta diversity analysis.

2.1.3 PCR and sequencing on Illumina MiSeq

Microbial community analysis typically begins with the extraction of DNA from primary samples (note that although most of this DNA comes from cells in the sample, some may consist of dead cells or extracellular DNA, so the representation of the active community from these sources is not perfect). Although methods for DNA extraction vary, several large initiatives such as the Earth Microbiome Project (EMP) [59] and the Human Microbiome Project (HMP) [196, 29, 200] have standardized on the MOBIO PowerSoil DNA extraction kit [6] to efficiently recover DNA from a wide range of sample types. After extraction, samples are Polymerase Chain Reaction (PCR) amplified under permissive conditions with primers containing the MiSeq sequencing adapters, a 12-nucleotide Golay barcode (first introduced in [49]) on the forward primer, followed by the bases matching the 16S rRNA gene; the reverse primer is not barcoded [22]. The annealing temperature is set to 50°C, which in our hands minimizes PCR artifacts (both primer dimer and background smear) while encouraging the primers to anneal to the largest diversity of sequences possible. Similarly, we believe that including sequencing adaptors and barcodes in the PCR step has advantages over multiple enzymatic treatments of the 16S amplicon that are otherwise needed to introduce adaptors and barcodes after PCR. The first, and most important consideration is the reduction of sample handling, which lowers the chance of contamination, mislabeling and loss of small-volume samples during preparation. Combining the adapters and barcodes

[6] `http://www.mobio.com`

in the PCR step allows for exact well-to-well mapping of samples to primers, providing a standardized way to track sample-barcode combinations through library preparation, an important consideration when sequencing hundreds to thousands of samples using 96- or 384-well sample preparation formats.

Because the MiSeq can generate a large number of sequences per run, many samples can be multiplexed on each single sequencing run. The choice of barcodes thus deserves some attention. For instance, homebrew barcodes can be as simple as using an arbitrary sequence of known nucleotides placed at the front of the amplicon and fed into an informatics pipeline for detection. Although simple, this approach has limited ability to detect sequencing error [22], and increases the risk of misassignment of a sequence to the wrong sample. The use of error correcting barcodes, such as Hamming [73] or Golay codes [22], allows the user to detect and correct errors in the barcode, decreasing the chances that a sequence is assigned to the wrong sample. Error-correcting barcodes also allow the user to retain more sequences, because 8-nucleotide Hamming codes can detect and correct 2 and 1 bit errors, respectively [73], and 12-nucleotide Golay codes can detect and correct 4 and 3 bit errors, respectively [72]. With the unique Golay codes described in [22], up to 2167 samples could be multiplexed on a single MiSeq run at a depth of 4600 per sample, certainly sufficient to detect the effects of many biological phenomena of interest [95, 97]. As the QIIME default settings detect Golay barcodes, we encourage the use of these codes when possible to maximize sequence retention and assignment accuracy.

Detailed instructions for loading the MiSeq for amplicon runs with custom barcodes can be found on the EMP website [7]. Briefly, pooled libraries are analyzed by Bioanalyzer (Agilent Technologies) and diluted to 2 μM quantitated by use of a Qubit Fluorometer (Life Technologies, High Sensitivity reagents). The phiX spike-in library (Illumina Inc.) is also diluted to 2 μM prior to use. Denaturation of the pooled 16S rRNA gene amplicon libraries and the phiX control is performed according to manufacturer's instructions (Illumina Inc.), giving a denatured template concentration of 20 ρM. Denatured templates are further diluted to 5 ρM (using Illumina HT1 buffer) and subsequently combined to give an 85% 16S rRNA gene amplicon library and 15% phiX control pool (1000 μL total volume). Improvements in the Illumina analysis software may allow reduction of this phiX spike-in, allowing more of the sequences to be used for 16S rRNA gene amplicons.

MiSeq reagent cartridges are prepared according to the manufacturer's instructions (Illumina Inc.). The sample pool (1000 μL total volume) is loaded in to cartridge position 17. Custom 16S rRNA gene Read 1, Index Read, and Read 2 sequencing primers are added directly to cartridge wells containing the standard Illumina Read 1, Index Read, and Read 2 sequencing primers (wells 12, 13 and 14 respectively, 5 μL each primer at 100 μM concentration [22]). Primers are added to wells using a long gel loading tip, and gently mixed using a plastic Pasteur pipette. Care must be taken to assure that reagents in the cartridge are localized to the bottom of the wells, and that no bubbles are present.

[7] `http://www.earthmicrobiome.org`

The spike-in of PhiX, at least at low levels, is still critical for obtaining usable amplicon data because the optics require diversity at each nucleotide position, which is not possible with absolutely conserved nucleotides within the 16S rRNA gene (or most other genes of interest). Many users have had difficulty mixing this protocol for custom amplicons with Illumina's own indexing protocol, which allows a maximum of 96 samples to be multiplexed per run at the time of writing. It is critical to use either this protocol exactly (allowing arbitrary numbers of custom barcodes) or to use Illumina's barcoding protocol, but not to mix and match steps and reagents.

2.1.4 QIIME workflow for conducting microbial community analysis

The Illumina MiSeq technology can generate up to 107 sequences in a single run [96]. QIIME takes the instrument output, and generates useful information about the community represented in each sample. At a coarse-grained level, we divide this process into upstream and downstream stages (Figure 2.1). The upstream step includes all the processing of the raw data (sequencing output), and generating the key files (OTU table and phylogenetic tree) for microbial analysis. The downstream step uses the OTU table and phylogenetic tree generated in the upstream step to perform diversity analysis, statistics and interactive visualizations of the data. Additionally, QIIME increasingly interfaces with other packages

such as IPython and R, allowing additional analyses to be conducted.

To illustrate some of the main features of QIIME, together with some of the analyses that can be performed outside QIIME, we use an example dataset consisting of samples from different body sites of 12 mice: the oral cavity, ileum, cecum, colon, fecal pellet, skin and whole mouse sample by homogenizing the mouse carcass. 7 mice were wild type genotype (WT from here so on), while the 5 remaining mice were transgenic (TG from here so on). The samples were collected by students during the IQ-Bio course taught by Manuel Lladser and Rob Knight during Spring 2013 at University of Colorado at Boulder (course identifiers: APPM5720-001-2013, CHEM4751-001-2013, CHEM5751-001-2013, CSCI4830-006-2013, CSCI7000-006-2013, MCDB6440-001-2013).

Upstream analysis steps

The QIIME analysis workflow starts with the sequencing output (fastq files), and a user-generated mapping file. The mapping file contains information for understanding what is in each sample and is therefore critical for performing the rest of the analyses; it is in tab-delimited text format. The main information in this file is a unique identifier for each sample, the barcode used for each sample, the primer sequence used, and a description for each sample, together with additional user-defined information that is necessary for understanding the results such as which species the sample was taken from, which site on the body is being studied, clinical variables relevant to the study, etc. The sample identifier,

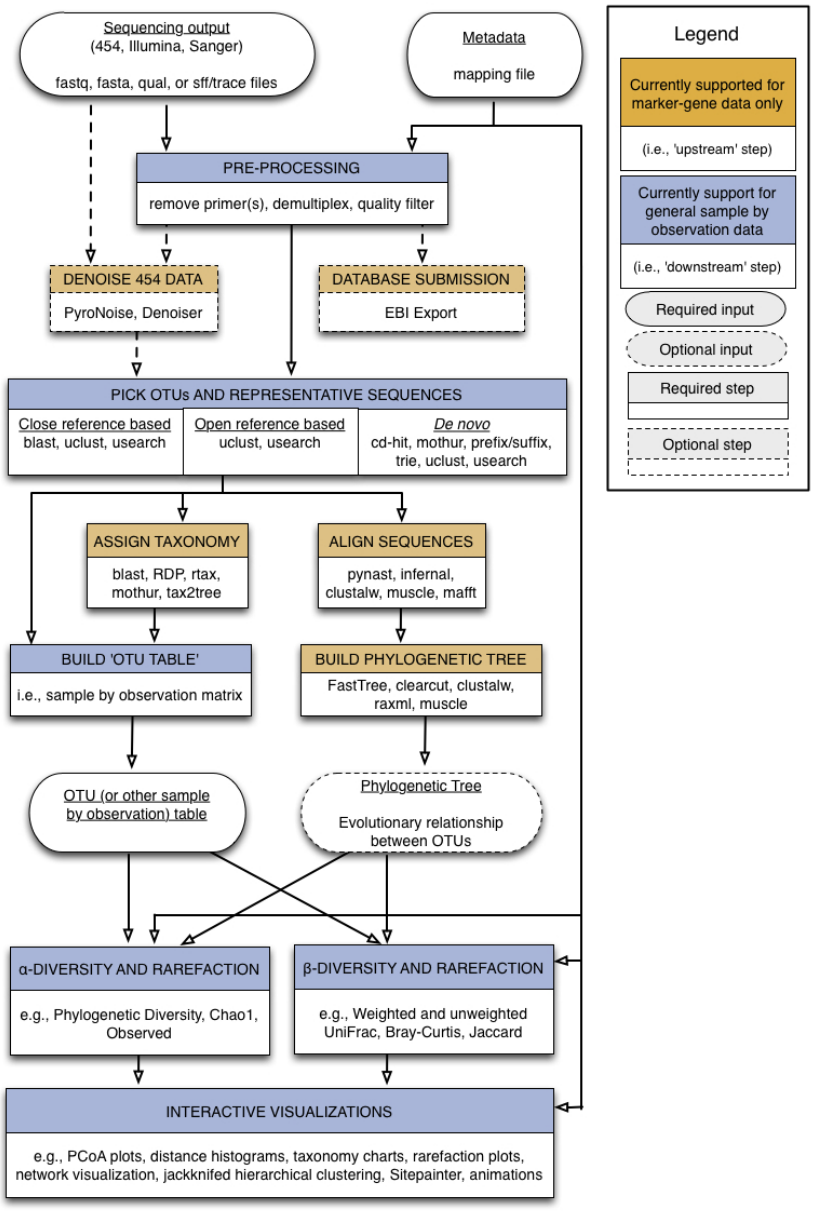


Figure 2.1: **QIIME workflow overview.** The Upstream process (brown boxes) includes all the steps that generate the OTU table and the phylogenetic tree. This step starts by preprocessing the sequence reads and ends by building the OTU table and the phylogenetic tree. The Downstream process (blue boxes) includes steps involved in analysis and interpretation of the results, starting with the OTU table and the phylogenetic tree and ending with alpha and beta diversity analyses, visualizations and statistics.

barcode and primer sequence information are required for the first step of the QIIME workflow. This preprocessing step combines sample demultiplexing, primer removal and quality-filtering. Additional information provided about the samples in the mapping file is helpful for later steps, especially for analyses that aggregate the samples by these fields (for example, comparing lean to obese subjects). We therefore recommend including as much additional data about the samples as possible (often called sample metadata). This auxiliary information is also very useful for identifying contaminated samples. For example, SourceTracker [88] is a package included in QIIME that identifies the proportion of different community sources, including contamination, in each sample based on a database of samples from known communities.

De-multiplexing and quality filtering. As mentioned above, high-throughput sequencing allows multiple samples to be combined in a single sequencing run [96]. However, each sequence must then be linked back to the individual sample that it came from via a DNA barcode. The barcodes, which are short DNA sequences unique to each sample, are incorporated into each sequence from a given sample during PCR. QIIME uses the barcodes in the mapping file to demultiplex, i.e. to assign the sequences back to the samples they are derived from, using error-correcting codes where available (as noted above). QIIME is also able to demultiplex variable-length barcodes such as those used in the HMP: see Variable-length barcodes in Other features below.

During demultiplexing, QIIME removes the barcodes and primer sequences

because they are not needed in later steps. Thus, the result after demultiplexing is a sequence matching the amplified 16S rRNA gene.

The third part of preprocessing is quality-filtering. Quality-filtering improves diversity estimates with Illumina sequencing substantially [14]. Illumina instruments, like most sequencing instruments, generate a quality score for each nucleotide (Phred), related to the probability that each nucleotide was read incorrectly. QIIME uses the Phred score and user-defined parameters to remove sequence reads that do not meet the desired quality. These user-defined parameters are: the percentage of consecutive high quality base calls (p), the maximum number of consecutive low quality base calls (r), the maximum number of ambiguous bases (typically coded as N) (n) and the minimum Phred quality score (q). For a detailed discussion of how these parameters affect diversity results, see [14]. This study recommends standard values for these parameters as $r = 3$, $p = 75\%$, $q = 3$ and $n = 0$, which are the default values in the QIIME pipeline. However, the optimal values for these parameters can vary both for individual sequencing runs and for different downstream analyses: for example, analyses such as machine learning benefit from larger numbers of low-quality sequences, whereas accurate counts of OTUs from a mock community require fewer, higher-quality sequences. Table 2.1 contains an overview of the guidelines presented in [14] for tuning these parameters to a given dataset.

The Illumina quality filtering approach differs in its fundamental principles from 454 denoising [160, 168]. 454 denoising is based on flowgram clustering [160,

Table 2.1: Overview of the guidelines to tune up the quality filtering parameters.

Dataset characteristics	q	p	r	Results
Majority of high-quality, full-length sequences	increase	increase	-	Retrieving full-length sequences with low error rates, increasing the discovery rate of rare OTUs
Short reads, or reads truncated by early low-quality base calls	-	lower	increase	Retain lower-quality but taxonomic usefult reads
Maximize read count for machine-learning tools, cross-metadata OTU counts comparison, etc	-	lower	-	Increased sample size

161] and is primarily targeted at reducing homopolymer runs, which are not a problem on the Illumina platform to the same extent. In contrast, the Illumina quality filtering is based on a per-base Phred quality score and does not target indels.

The QIIME quality filtering process works as follows. Starting at the beginning of the sequence, QIIME checks that the next r Phred values exceed the user-defined quality threshold q. If the test is positive, it continues slicing the window of r bases until the test fails, or the end of the sequence is reached. The sequence is then trimmed to the last base that met the quality threshold. The next test determines whether the length of the trimmed sequence exceeds p, expressed as the percentage of length of the raw sequence. If this check fails, the sequence is excluded. Otherwise, QIIME performs the last check on the sequence, which counts the number of ambiguous characters (N) in the trimmed sequence and checks that it is less than n. If the test fails, the sequence is rejected. QIIME combines the de-multiplexing, primer removal and quality filtering processes in a single script, split_libraries_fastq.py:

```
split_libraries_fastq.py
  -i $PWD/IQ_Bio_16sV4_L001_sequences.fastq.gz \
  -b $PWD/IQ_Bio_16sV4_L001_sequences_barcodes.fastq.gz \
  -m $PWD/IQ_Bio_16sV4_L001_map.txt -o $PWD/slout \
  --rev_comp_mapping_barcodes
```

In our example dataset, we used the –rev_comp_mapping_barcodes option in order to indicate that the barcodes present in the mapping file are reverse complements of Golay 12 barcodes. We used the recommended default parameters for quality filtering on this dataset. However, to change the values for the r, p, n and q quality filtering parameters, we can use the $-r$, $-p$, $-n$ and $-q$ options to the script. This command will write a FASTA-formatted file in the slout folder, called seqs.fna, which contains the demultiplexed sequences that pass the quality filter. Each sequence contains the information about which sample it came from encoded in the name of the sequence.

Multiple lanes of Illumina fastq data can be processed together in a single call to the script, just by passing the sequence files, the barcode files and the mapping files in the same order to the $-i$, $-b$ and $-m$ options, respectively. For example, with two lanes, the command would look like:

```
split_libraries_fastq.py
  -i sequences1.fastq,sequences2.fastq
  -b sequences1_barcodes.fastq,sequences2_barcodes.fastq
  –m mapping1.txt,mapping2.txt -o slout
```

The user can check how many sequences have been demultiplexed and passed quality-filtering by using the count_seqs.py command. This command also shows the mean and standard deviation of the sequence length:

```
count_seqs.py -i $PWD/slout/seqs.fna
```

```
12687021 :slout/seqs.fna (Sequence lengths (mean +/- std):
150.9989 +/- 0.1715)

12687021 : Total
```

OTU picking. The next step is clustering the preprocessed sequences into OTU, which in traditional taxonomy represent groups of organisms defined by intrinsic phenotypic similarity that constitute candidate taxa [186, 188]. For DNA sequence data, these clusters, and hence the OTUs, are formed based on sequence identity. In other words, sequences are clustered together if they are more similar than a user-defined identity threshold, presented as a percentage (s). This level of threshold is traditionally set at 97% of sequence similarity, conventionally assumed to represent bacterial species [40]; other levels approximately represent other taxa, although the fit between molecular data and traditional taxonomy varies for different taxa. QIIME supports three approaches for OTU picking (de novo, closed-reference and open-reference), and multiple algorithms for each of these approaches (Table 2.2). The de novo approach (Figure 2.2a) groups sequences based on sequence identity. The closed-reference approach (Figure 2.2b) matches sequences to an existing database of reference sequences. If a sequence fails to match the database, it is discarded. The open-reference approach (Figure 2.2c) also starts with an existing database and tries to match the sequences

against them. However, if a sequence does not match the database, it is added to the database as a new reference sequence.

Table 2.2: Supported OTU picking methods in QIIME, with a brief description of the algorithm employed and in which OTU picking approach can be used.

Method	Picking approach			Description	Reference
	de novo	closed reference	open reference		
cd-hit	Yes	-	-	Applies a "longest-sequence-first list removal algorithm" to cluster sequences	[112, 111]
Mothur	Yes	-	-	Takes an aligned set of sequences and clusters them using a nearest-neighbor, furthest-neighbor or average-neighbor algorithm.	[177]
prefix/suffix	Yes	-	-	Clusters sequences which are identical in their first and/or last bases.	QIIME team, unpublished
Trie	Yes	-	-	Clusters sequences which are identical sequences and sequences which are subsequences of other sequences.	QIIME team, unpublished
blast	-	Yes	-	Compares and clusters each sequence against a reference database of sequences.	[3]
uclust	Yes	Yes	Yes	Creates seed sequences which generate clusters based on percent identity.	[43]
usearch	Yes	Yes	Yes	Creates seed sequences which generate clusters based on percent identity, filters low abundance clusters and performs de novo and reference based chimera detection.	[43]

The OTU picking strategies shown in Figure 2.2 are built on top of algorithms for de novo clustering. Of the various algorithms available, the furthest-neighbor, average-neighbor or nearest neighbor in mothur [175, 177] (also named complete linkage, average linkage, and single linkage respectively) are the most widely used. Furthest-neighbor requires that each sequence is closer than the distance threshold to every other sequence already in the OTU (Figure 2.3). Average-neighbor requires that the average pairwise distance of all sequences in the OTU is closer than the distance threshold. Nearest-neighbor requires that each sequence is closer than the distance threshold to any sequence already in the OTU. Because these three algorithms are variants on hierarchical clustering, they

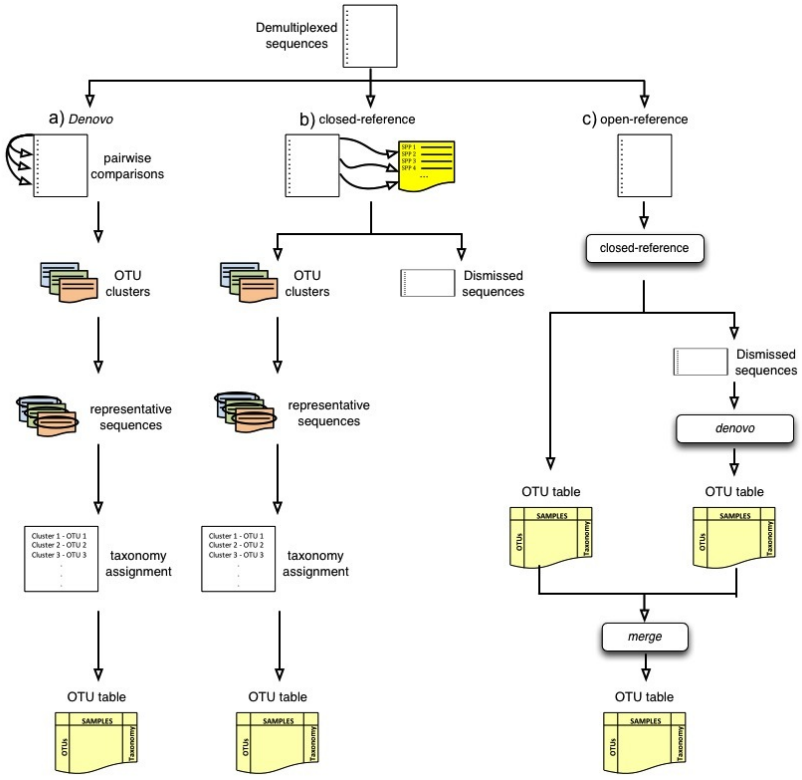


Figure 2.2: **Cartoon representation of the OTU picking approaches.** (a) de novo, (b) closed-reference and (c) open reference OTU picking respectively. In the de novo method, sequences are compared to each other and then clusters are formed. In the closed-reference method, sequences are compared directly to a reference dataset (e.g. GreenGenes). Sequences that match a reference sequence are clustered; the remaining sequences are discarded. I both OTU picking methods, once clusters are formed, a representative sequence is selected and then taxonomy is assigned to that sequence (and applied to the rest of the sequences that make up the OTU). Open-reference combines the closed-reference and open-reference methods. The first step is identical to closed-reference, sequences discarded in the first step are clustered into OTUs by the de novo method, and both OTU tables are merged into a single final OTU table. De novo and open-reference cluster all the sequences, but closed-reference allows better comparisons between studies, especially those using different primers, because all OTUs occur in a common reference space.

require loading the distance matrix (proportional to the square of the number of dereplicated sequences) into memory, and are therefore challenging to apply to large datasets (e.g., larger than 105 sequences). The OTUs yield by these three algorithms also change their memberships at different sequencing depths (i.e. the number of sequences chosen for clustering), which can be a problem for estimates of total OTU numbers [171].

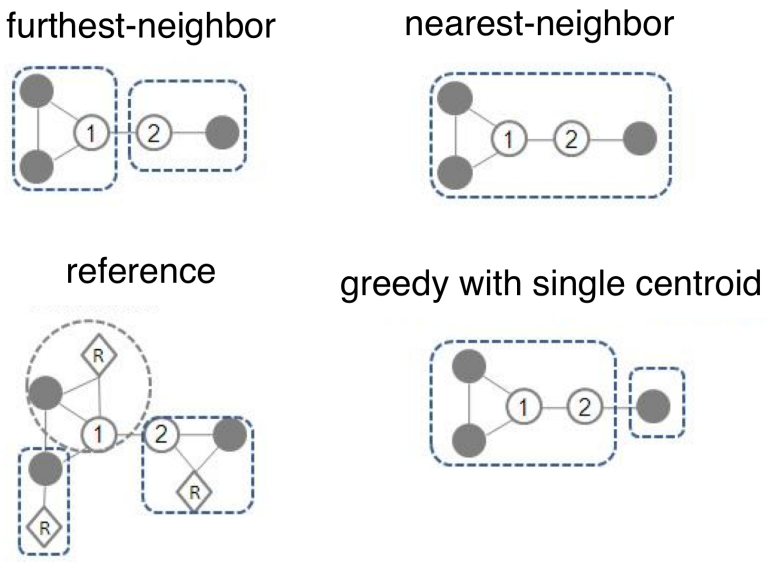


Figure 2.3: **Cartoon demonstrating different clustering algorithms.** Circles representing sequences linked with lines are within the distance threshold. The two numbered sequences are the first and second sequences in order in the file. The reference algorithms only consider the distance between reference (R) and sequences.

A solution to the distance matrix problem comes from uclust and usearch, which are greedy algorithms based on using a single centroid in each OTU [43]. The centroid could be either from a reference database (usearch) or identified de novo from the sequence dataset (both uclust and usearch) (Figure 2.3). Sequences

are serially compared to centroids in a user-defined order (usually decreasing abundance). If a sequence falls within the distance threshold of more than one centroid, the new sequence can either be grouped with the first centroid encountered, or the one with the closest distance. Both uclust and usearch are much more efficient than the hierarchical methods, and they do not need to load a large distance matrix into memory (although recent versions of mothur also avoid the constraint of loading the full distance matrix). usearch is the default de novo OTU picking method in QIIME. Note that it is essential to note both your OTU picking strategy, and, if de novo OTU picking is used, which algorithm you used to do it: it is not sufficient simply to state that you used a 97% threshold.

Because the OTU picking approach selection is a critical point in microbial community analysis, the QIIME team has produced a detailed document that describes the OTU picking protocols, their advantages and limitations [8]. Table 2.3 compares the different OTU picking approaches and gives guidelines for choosing an appropriate OTU picking strategy.

The recommended OTU picking approach is open-reference OTU picking, because this approach provides the best trade-off between the time taken to complete the analysis and the ability to discover novel diversity.

Once the sequences have been clustered into OTUs, a representative sequence is picked for each OTU. The entire cluster will thus be represented by a single sequence, speeding up subsequent steps (because redundant sequences need

[8] `https://github.com/qiime/qiime/blob/master/doc/tutorials/otu_picking.rst`

Table 2.3: OTU picking approaches comparison. The table shows when each of the OTU picking approaches should be used and when they cannot be applied. It briefly describes the advantages and disadvantages of using each of the OTU picking approaches.

	de novo	closed reference	open reference
Must use if	There is no reference sequence collection to cluster against (e.g. infrequently used marker gene)	Comparing non-overlapping. amplicons. The reference set of sequences must span both of the regions being sequenced	-
Cannot use if	Comparing non-overlapping amplicons (e.g. V2 and V4 regions of 16S rRNA)	There is no reference sequence collection to cluster against (e.g. infrequently used marker gene)	Comparing non-overlapping amplicons (e.g. V2 and V4 regions of 16S rRNA). There is no reference sequence collectionto cluster against (e.g. infrequently used marker gene)
Pros	All reads are clustered	Fast, as it is fully parallelizable (useful for extremely large datasets). Better tree and taxonomy quality since the OTUs are already defined on the reference set.	All reads are clustered. Fast, as is partially run on parallel.
Cons	Time consuming since it runs in serial respect to the reference set because the reads that dont hit the reference sequence collection are discarded, so the analysis focus on the already known diversity. If the studied environment is not well-characterized, a large fraction of the reads can be thrown away	Inability to detect novel diversity with respect to the reference set because the reads that dont hit the reference sequence collection are discarded, so the analysis focus on the already known diversity. If the studied environment is not well- characterized, a large fraction of thereads can be thrown away	There are still some steps performed in serial. If the data set contains a lot of novel diversity with respect to the reference set, this can still be slow.

not be considered). QIIME allows the representative sequence to be selected using several techniques: choosing a sequence at random, choosing the longest sequence, the most abundant sequence or the first sequence. If using uclust or usearch [43], the cluster seed will be used as the representative sequence. The default behavior in QIIME is to use the most abundant sequence in each OTU as the representative sequence, because these sequences are least likely to represent sequencing errors (for other applications, such as clustering with near-full-length Sanger sequences, it may be more desirable to pick the longest sequence instead). In case of closed-reference OTU picking, sequences from the reference collection should be used as the representative sequences, which is the default behavior when the closed-reference approach is selected.

Identify chimeric sequences. During the PCR amplification process, some of the amplified sequences can be produced from multiple parent sequences, generating sequences known as chimeras. Although these sequences are technical artifacts rather than representing actual members of the community, chimeric sequences are important for alpha diversity estimates (although they are less important for cross-sample comparisons, because each chimera is relatively rare and the same chimera is unlikely to be generated systematically in different samples [108]. However, the same chimera can sometimes be generated in multiple PCR reactions: for example, Haas et al. [69] reported that chimeric sequences formed from Streptococcus and Staphylococcus occurred multiple times independently, so presence of the same sequence in multiple PCR does not mean that it is not chimeric.

QIIME currently supports three different methods for detecting chimeras: blast fragments, a taxonomy-assignment-based approach using BLAST [3]; ChimeraSlayer [69], which uses BLAST to identify potential chimera parents; and usearch 6.1 [43], which can perform de novo chimera detection based on abundances as well as reference-based chimera detection. The recommended method for identifying chimeric sequences is uchime [46], which is integrated in the usearch 6.1 [43] pipeline. Uchime is the fastest method for detecting chimeric sequences and it is executed by default if the usearch method is selected for picking OTU.

Taxonomy assignment. The next step in the QIIME workflow is to assign the taxonomy to each sequence of the representative set. This step connects

the OTUs to named organism, which is useful for inferring likely functional roles for members of the community. When using a closed-reference approach for OTU picking, the taxonomy of the sequences can be pulled out from the reference set. In case of the open-reference and de novo approaches, because the clusters are not created from any reference database (as a reminder, in the open-reference approach, sequences that fail to cluster to the reference database form new clusters), the taxonomy should be assigned using a reference dataset. We recommend the GreenGenes database [37, 131] as the default reference data set for assigning taxonomy, although the RDP [28] and Silva [158] databases also have strengths and weaknesses relative to GreenGenes and should be considered for some analyses. Silva includes microbial eukaryotes and has invested substantial effort in cleaning up marine taxa; RDP has close links to formally recognized names in taxonomy, which can be especially useful for medical microbiology. QIIME can assign taxonomy against any of the given databases, or against a custom database, using several methods: BLAST [3], RDP Classifier [213], rtax [187], mothur [177] and tax2tree [131]. The QIIME team recommends the RDP classifier method [213] with a confidence value of 0.8. However, if the user has paired-end reads, the best method to use is the rtax [187], and the user should provide the fasta files with both the first and second read from the paired-end sequencing. Note that the taxonomy assignment method and the reference database must both be described in order for an analysis to be reproducible, and that these methods can have a larger effect on taxonomy than the underlying biological sample, so it is important to be

consistent [115].

Sequence alignment. The next step in the QIIME workflow is to align the sequences. The sequences must be aligned to infer a phylogenetic tree, which is used for diversity analyses and to understand the relationships among the sequences in the sample. Currently, QIIME supports the following methods for performing sequence alignment: PyNAST [19], Infernal [144], clustalw [102], muscle [42] and mafft [84]. The recommended (and default) method is PyNAST [19]. This method aligns the sequences against a template sequence alignment, for which we recommend the GreenGenes core set [37].

When sequences do not align well using PyNAST, the Infernal package [144] should be used. Like PyNAST, it requires a template alignment, but unlike PyNAST, it uses stochastic context-free grammars (SCFGs) to align incorporating secondary structure. Although this method is slow compared to other methods, it does takes advantage of RNA secondary structure (provided in a Stockholm-format file) and can be useful for aligning more variable rRNAs. For marker genes other than rRNA genes, the best strategy for building phylogenetic trees is to align the protein sequences (if available) using MUSCLE.

Phylogeny construction. This step in the QIIME workflow infers a phylogenetic tree from the multiple sequence alignment generated by the previous step. The phylogenetic tree represents the relationships among sequences in terms of the amount of sequence evolution from a common ancestor. This phylogenetic tree is used in many downstream analyses, such as the UniFrac metric [118] for

beta diversity.

The current methods supported for inferring the phylogenetic tree in QIIME are FastTree [157], clearcut [47], clustalw [102], raxml [193] and muscle [42]. The default and recommended method in QIIME is the FastTree [157] method because it shows the best trade-off between run time and reliability of the inferred tree.

Make OTU table The last part of the upstream stage in QIIME is to construct the OTU table. The OTU table is a sample by observation matrix that also includes the taxonomic prediction for each OTU. For the OTU table representation, QIIME uses the Genomics Standards Consortium candidate standard Biological Observation Matrix (BIOM) format [130]. The OTU table, the mapping file and the phylogenetic tree, are the main files for performing the downstream analysis.

QIIME can perform all the steps for generating the OTU table and the phylogenetic tree from the preprocessed data in a single command. There is a separate command for each OTU picking approach. In the following commands, we assume that the GreenGenes reference files [37] are located in the current directory. As a remainder, our seqs.fna has 12.687.021 sequences of length 150.9989 +/- 0.1715:

- For de novo (run time 80 hours on 1 processor (not parallelizable)):

```
pick_de_novo_otus.py -i $PWD/slout/seqs.fna \
-o $PWD/denovo_otus
```

- For closed-reference (run time 2 hours on 20 processors):

```
pick_closed_reference_otus.py
  -i $PWD/slout/seqs.fna \
  -o $PWD/closed_ref_otus \
  -r $PWD/gg_12_10_otus/rep_set/97_otus.fasta \
  -t $PWD/gg_12_10_otus/taxonomy/\
     97_otu_taxonomy.txt \
  -a –O 20
```

- For open-reference (run time 27 hours on 20 processors):

```
pick_open_reference_otus.py \
 -o $PWD/open_ref_otus \
 -i $PWD/slout/seqs.fna \
 -r $PWD/gg_12_10_otus/rep_set/97_otus.fasta \
 -a –O 20
```

Because the closed-reference and open-reference OTU picking approaches can be run in parallel, we use the $-a$ and $-O$ 20 options in order to run them using 20 processors.

Downstream analysis steps

Once we have generated the OTU table and the phylogenetic tree, we can start the downstream analysis. At this point, we strongly recommend performing

a second level of quality-filtering, based on OTU abundance. The recommended procedure is to discard those OTUs with a number of sequences less than 0.005% of the total number of sequences (see Bokulich et al. [14] for a detailed description of the effect of this parameter in further downstream analyses). QIIME executes the OTU abundance quality-filtering step through the script filter_otus_from_otu_table.py:

```
filter_otus_from_otu_table.py \
 -i $PWD/open_ref_otus/\
     otu_table_mc2_w_tax_no_pynast_failures.biom \
 -o $PWD/open_ref_otus/otu_table_filtered.biom \
 --min_count_fraction 0.00005
```

This step greatly reduces the problem of spurious OTUs, most of which are present at very low abundance.

QIIME 1.7.0 allows a first-pass view of common diversity analyses using a single command: core_diversity_analysis.py. One of the parameters required by this command is the sampling depth, the number of sequences that should be included in each sample for diversity analyses. This is required, because many of the commonly used diversity metrics are very sensitive to the number of sequences obtained per sample, such that samples that are similar in the number of sequences that were obtained appear similar to one another. This is bad because the number of sequences per sample is typically a methodological artifact, not reflective of

biological reality. The sampling depth defines the size of the random subset of sequences that will be selected for each sample for all subsequent diversity analyses.

The optimal sampling depth is data-dependent. There is no universal way of choosing a rarefaction level, although heuristics can be applied. For example, if most samples have more than 10,000 sequences and the rest range from 500 to 50 sequences per sample, it would be recommended to use 10,000 as the rarefaction level. Although many studies show marked variation in sequence depth with only a few bad samples, it is not always easy to choose the rarefaction level. We strongly recommend rarefying over 1000 sequences/sample for Illumina MiSeq, because samples below this level often suffer from other quality issues as well.

The information needed to choose the rarefaction level can be obtained from the script print_biom_table_summary.py, which shows summary information on the OTU table such as the number of sequences, the number of OTUs, the number of samples and the number of counts per sample, among others:

```
print_biom_table_summary.py \
 -i $PWD/open_ref_otus/otu_table_filtered.biom


Num samples: 90
Num observations: 783
Total count: 10637688.0
Table density (fraction of non-zero values): 0.4289
```

```
Table md5 (unzipped): eb0f1d7fbb50bc31695dade31db1e198
Counts/sample summary:
 Min: 1.0
 Max: 493427.0
 Median: 99111.0
 Mean: 118196.533333
 Std. dev.: 94277.5956531
 Sample Metadata Categories: None provided
 Observation Metadata Categories: taxonomy
Counts/sample detail:
BLANK4.732555: 1.0
BLANK5.732537: 1.0
 Joshua.Jose.WTAbd.732533: 1.0
 Nick.Krishna.TG.Fec.732513: 2.0
 TH.CVA.WT.Oral.732491: 2.0
BLANK2.732552: 3.0
BLANK3.732479: 5.0
BLANK6.732470: 7.0
 Elizabeth.Chris.WT.Abd.732490: 10.0
 Uri.Jake.TGAbd.732468: 10.0
 TH.CVA.WT.Abd.732477: 13.0
```

```
 BLANK10.732524: 812.0
  Elizabeth.Chris.WT.Oral.732520: 7410.0
  Elizabeth.Chris.WT.Col.732481: 21746.0
Jordan.Lisette.TG.Ile.732463: 27149.0
  ...
  TH.CVA.WT.Fec.732553: 372327.0
  Wang.TG.Cec.732527: 396391.0
  TH.CVA.WT.Ile.732517: 493427.0
```

In the above output we can see the information contained in the OTU table resulting from applying the open-reference OTU picking. Some of the relevant information contained in this output is the total number of samples (90), the total number of OTUs (783), the number of reads (10637688) and the number of OTUs per sample. Applying the above heuristic, we could select a subsampling depth of 7410 sequences. However, because we have run three different OTU picking approaches and we want to compare them, we must search for the rarefaction level that best fits the three OTU tables. Below are the summarized information for the de novo OTU table and the closed reference OTU table, respectively:

```
print_biom_table_summary.py \
 -i $PWD/denovo_otus/otu_table_filtered.biom
Num samples: 93
Num observations: 600
```

```
Total count: 11122386.0

Table density (fraction of non-zero values): 0.4344

Table md5 (unzipped): b002dd85c93fd9d0571ff23b05d21dde

Counts/sample summary:

 Min: 0.0

 Max: 497234.0

 Median: 108322.0

 Mean: 119595.548387

 Std. dev.: 93487.3335598

 Sample Metadata Categories: None provided

 Observation Metadata Categories: taxonomy

Counts/sample detail:

BLANK7.732497: 0.0

BLANK8.732522: 0.0

 Jordan.Lisette.TG.Abd.732467: 0.0

BLANK4.732555: 1.0

BLANK5.732537: 1.0

 Joshua.Jose.WTAbd.732533: 1.0

BLANK2.732552: 3.0

 Nick.Krishna.TG.Fec.732513: 3.0

 TH.CVA.WT.Oral.732491: 3.0
```

```
 BLANK3.732479: 5.0
 BLANK6.732470: 9.0
  Elizabeth.Chris.WT.Abd.732490: 10.0
  Uri.Jake.TGAbd.732468: 10.0
  TH.CVA.WT.Abd.732477: 13.0
 BLANK10.732524: 825.0
  Elizabeth.Chris.WT.Oral.732520: 7376.0
  Joey.Aaron.Kyle.WT.Abd.732541: 35655.0
  ...
  Wang.TG.Cec.732527: 394351.0
  TH.CVA.WT.Ile.732517: 497234.0

print_biom_table_summary.py \
 -i $PWD/closed_ref_otus/otu_table_filtered.biom
Num samples: 90
Num observations: 673
Total count: 9434459.0
Table density (fraction of non-zero values): 0.4250
Table md5 (unzipped): 257b528478a2700c72f979ce8d9a9a1c
Counts/sample summary:
  Min: 1.0
```

```
 Max: 347785.0

 Median: 90092.0

 Mean: 104827.322222

 Std. dev.: 78560.4683831

 Sample Metadata Categories: None provided

 Observation Metadata Categories: taxonomy

Counts/sample detail:

 BLANK4.732555: 1.0

 BLANK5.732537: 1.0

 Joshua.Jose.WTAbd.732533: 1.0

 BLANK3.732479: 2.0

 Nick.Krishna.TG.Fec.732513: 2.0

 TH.CVA.WT.Oral.732491: 2.0

 BLANK2.732552: 3.0

 Uri.Jake.TGAbd.732468: 5.0

 BLANK6.732470: 7.0

 Elizabeth.Chris.WT.Abd.732490: 10.0

 TH.CVA.WT.Abd.732477: 12.0

 BLANK10.732524: 710.0

 Elizabeth.Chris.WT.Oral.732520: 7205.0

 Elizabeth.Chris.WT.Col.732481: 22652.0
```

```
...
TH.CVA.WT.Fec.732553: 329988.0
TH.CVA.WT.Ile.732517: 347785.0
```

From the above output, we see that a reasonable rarefaction level for the three tables is 7205 counts per sample, derived from the closed reference OTU picking.

Once the subsampling depth is chosen, we can execute the core_diversity_analyses.py command over the three OTU tables. We provide the subsampling depth via the $-e$ parameter, the OTU table via the $-i$ parameter, the mapping file through the $-m$ parameter and the metadata categories to use in categorical analyses through the $-c$ parameter. The $-o$ parameter is used to provide the output directory and the $-a -O$ 64 are used to run the command in parallel using 64 processes.

```
mkdir $PWD/diversity_analysis


core_diversity_analyses.py \
-i $PWD/open_ref_otus/otu_table_filtered.biom \
-m $PWD/IQ_Bio_16sV4_L001_map.txt \
-t $PWD/open_ref_otus/rep_set.tre \
-e 7205 -c GENOTYPE,BODY_SITE \
-o $PWD/diversity_analysis/open_ref -a -O 64
```

```
core_diversity_analyses.py \
 -i $PWD/denovo_otus/otu_table_filtered.biom \
 -m $PWD/IQ_Bio_16sV4_L001_map.txt \
 -t $PWD/denovo_otus/rep_set.tre -e 7205 \
 -c GENOTYPE,BODY_SITE \
 -o $PWD/diversity_analysis/denovo -a -O 64


core_diversity_analyses.py \
 -i $PWD/closed_ref_otus/otu_table_filtered.biom \
 -m $PWD/IQ_Bio_16sV4_L001_map.txt \
 -t $PWD/gg_12_10_otus/trees/97_otus.tree \
 -e 7205 -c GENOTYPE,BODY_SITE \
 -o $PWD/diversity_analysis/closed_ref -a -O 64
```

The core_diversity_analyses.py command filters the OTU table before executing the diversity analyses. The filter removes samples from the OTU table that do not have at least the user-defined subsampling depth (7205 in our case). This filtering removes low-coverage samples from the OTU table, because they are not informative enough to be included in the study. After these samples have been filtered, the script performs the rarefaction step at the given subsampling depth.

The output of this script is an HTML file that can be opened in a web

browser (Figure 2.4). This HTML file gives access to the results of the different diversity analysis performed (taxa summaries, α-diversity, β-diversity and category significance) which will be explained in further sections.

For the following downstream analysis we have used the OTU table and phylogenetic tree resulting from the open-reference OTU picking approach. In cases where we are performing comparisons between OTU picking approaches, we will specify which approaches we have used.

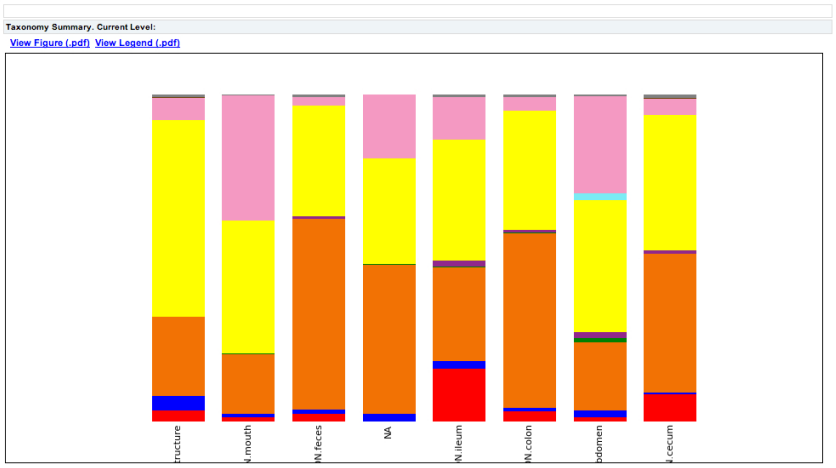
Taxa summaries. One way to visualize the OTUs in each sample is a taxa summary, which summarizes the relative abundance of the taxa present in a set of samples on multiple taxonomic levels (e.g. phylum, order, etc.) (see Figure 2.5). This provides a quick way to identify samples that may be drastically different from others (i.e. outliers), and visually identify expected patterns and differences between and among samples. For example, this tool can be used to identify patterns such as differences in the relative abundance of Firmicutes and Bacteroidetes in the gut microbiomes of lean versus obese mice, e.g. Ley, Backhed, Turnbaugh, Lozupone, Knight, and Gordon [108]. These patterns can then be statistically tested using other methods, either within QIIME or elsewhere. QIIME contains a workflow called summarize_taxa_through_plots.py that generates user-specified plot types, including bar, pie, and area graphs. These graphs provide a way to visually compare the composition of each sample, or of groups of samples. An OTU table with assigned taxonomies is the only required input file, and options allow the user summarize across categories (using the metadata file), at different

Run summary data	
Master run log	log_20130607115410.txt
BIOM table statistics	biom_table_summary.txt
Filtered BIOM table (minimum sequence count: 7205)	table_mc7205.biom.gz
Beta diversity results (even sampling: 7205)	
Distance boxplots (weighted_unifrac)	GENOTYPE_Distances.pdf
Distance boxplots statistics (weighted_unifrac)	GENOTYPE_Stats.txt
Distance boxplots (weighted_unifrac)	BODY_SITE_Distances.pdf
Distance boxplots statistics (weighted_unifrac)	BODY_SITE_Stats.txt
3D plot (weighted_unifrac, continuous coloring)	weighted_unifrac_pc_3D_PCoA_plots.html
3D plot (weighted_unifrac, discrete coloring)	weighted_unifrac_pc_3D_PCoA_plots.html
2D plot (weighted_unifrac, continuous coloring)	weighted_unifrac_pc_2D_PCoA_plots.html
2D plot (weighted_unifrac, discrete coloring)	weighted_unifrac_pc_2D_PCoA_plots.html
Distance matrix (weighted_unifrac)	weighted_unifrac_dm.txt
Principal coordinate matrix (weighted_unifrac)	weighted_unifrac_pc.txt
Distance boxplots (unweighted_unifrac)	GENOTYPE_Distances.pdf
Distance boxplots statistics (unweighted_unifrac)	GENOTYPE_Stats.txt
Distance boxplots (unweighted_unifrac)	BODY_SITE_Distances.pdf
Distance boxplots statistics (unweighted_unifrac)	BODY_SITE_Stats.txt
3D plot (unweighted_unifrac, continuous coloring)	unweighted_unifrac_pc_3D_PCoA_plots.html
3D plot (unweighted_unifrac, discrete coloring)	unweighted_unifrac_pc_3D_PCoA_plots.html
2D plot (unweighted_unifrac, continuous coloring)	unweighted_unifrac_pc_2D_PCoA_plots.html
2D plot (unweighted_unifrac, discrete coloring)	unweighted_unifrac_pc_2D_PCoA_plots.html
Distance matrix (unweighted_unifrac)	unweighted_unifrac_dm.txt
Principal coordinate matrix (unweighted_unifrac)	unweighted_unifrac_pc.txt
Taxonomic summary results	
Taxa summary bar plots	bar_charts.html
Taxa summary area plots	area_charts.html
Taxonomic summary results (by BODY_SITE)	
Taxa summary bar plots	bar_charts.html
Taxa summary area plots	area_charts.html
Taxonomic summary results (by GENOTYPE)	
Taxa summary bar plots	bar_charts.html
Taxa summary area plots	area_charts.html
Category results	
Category significance (GENOTYPE)	category_significance_GENOTYPE.txt
Category significance (BODY_SITE)	category_significance_BODY_SITE.txt
Alpha diversity results	
Alpha rarefaction plots	rarefaction_plots.html
Alpha diversity statistics (GENOTYPE, PD_whole_tree)	GENOTYPE_PD_whole_tree.txt
Alpha diversity statistics (GENOTYPE, observed_species)	GENOTYPE_observed_species.txt
Alpha diversity statistics (GENOTYPE, chao1)	GENOTYPE_chao1.txt
Alpha diversity statistics (BODY_SITE, PD_whole_tree)	BODY_SITE_PD_whole_tree.txt
Alpha diversity statistics (BODY_SITE, observed_species)	BODY_SITE_observed_species.txt
Alpha diversity statistics (BODY_SITE, chao1)	BODY_SITE_chao1.txt

Need help?
You can get answers to your questions on the QIIME Forum.
See the QIIME tutorials for examples of additional analyses that can be run.
You can find documentation of the QIIME scripts in the QIIME script index.

Figure 2.4: **HTML result from core_diversity_analyses.py.** This HTML file summarizes and gives access to the results of the diversity analyses conducted on the given OTU table

taxonomic levels, or only using OTUs that are present at abundances higher or lower than user-defined thresholds. The web interface allows a scroll-over feature that identifies the taxonomy of the separate taxa. Additional output files include image files of the charts and legends, and tab-delimited files of the calculated abundances, which can then be further filtered and manipulated for downstream statistical analyses.



Legend	Taxonomy	Total count	Total %	UBERON.multi.tissue.structure %	UBERON.mouth %	UBERON.feces %	NA %	UBERON.ileum %	UBERON.colon %	UBERON.skin.of.abdomen %	UBERON.cecum %
	k__Bacteria;Other	0	4.5%	3.3%	1.3%	2.5%	0.0%	16.2%	3.0%	1.3%	8.2%
	k__Bacteria;p__Actinobacteria	0	1.9%	4.5%	1.1%	1.1%	2.3%	1.2%	2.2%	0.7%	
	k__Bacteria;p__Bacteroidetes	3	36.5%	24.2%	18.2%	58.4%	45.8%	28.7%	53.4%	20.9%	42.4%
	k__Bacteria;p__Cyanobacteria	0	0.3%	0.1%	0.1%	0.0%	0.1%	0.3%	0.2%	1.3%	0.0%
	k__Bacteria;p__Deferribacteres	0	0.8%	0.0%	0.2%	0.7%	0.0%	1.7%	0.7%	1.7%	1.0%
	k__Bacteria;p__Firmicutes	3	40.2%	60.0%	40.4%	33.9%	32.4%	37.1%	36.5%	40.3%	41.4%
	k__Bacteria;p__Fusobacteria	0	0.3%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.2%	0.0%
	k__Bacteria;p__Proteobacteria	1	14.9%	6.9%	38.4%	2.5%	19.5%	12.9%	4.1%	29.7%	5.1%
	k__Bacteria;p__Spirochaetes	0	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	k__Bacteria;p__TM7	0	0.1%	0.3%	0.0%	0.1%	0.0%	0.1%	0.1%	0.0%	0.1%
	k__Bacteria;p__Tenericutes	0	0.5%	0.6%	0.2%	0.8%	0.0%	0.7%	0.7%	0.4%	1.0%
	k__Bacteria;p__Verrucomicrobia	0	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.1%

Figure 2.5: **Taxa summary of the example dataset.** Samples have been grouped and averaged by body site, and taxonomic composition is shown on the phylum level. Each column in the plot represents a body site, and each color in the column represents the percentage of the total sample contributed by each taxon group at phylum level. The taxa summaries plot help us to see which taxon groups are more prevalent in a sample. For example, the fecal samples are dominated by Bacteroidetes, while mouth and skin samples are dominated by Proteobacteria. We can also see that Fusobacteria is only present at appreciable levels in the skin samples.

Diversity analysis. Microbial ecology studies the diversity of microor-

ganisms by characterizing bacterial communities in different environments, and determining the factors that drive diversity in these communities [10]. Whittaker (1960) and Whittaker (1972) define different types of measurements of diversity as alpha, beta and gamma diversities. Alpha diversity is defined as the diversity of organisms in one sample or environment. Beta-diversity is the difference in diversities across samples or environments. Finally, gamma-diversity (γ-diversity) measures the diversity at a broader scale, such as a province or region. Several different metrics of alpha- and beta-diversity are implemented in in QIIME pipeline. Diversity measurements and their applications in microbial have been discussed in detail elsewhere [83, 97, 121], and here we focus on examples of their application.

Alpha diversity analysis. QIIME can generate plots showing the results of alpha diversity, allowing the user to choose the diversity metric and different rarefaction levels (Figure 2.6). These images are often used to estimate the true species richness of a community.

QIIME implements dozens of the most widely used alpha diversity indices, including both phylogenetic indices (which require a phylogenetic tree) and non-phylogenetic indices. Users can obtain a list of the alpha diversity indices implemented in QIIME by passing the parameter *s* to the alpha_diversity.py script. Phylogenetic metrics have been especially useful in our experience because they provide additional power by accounting for the degrees of phylogenetic divergence between sequences within each sample. Thus, for alpha diversity, we recommend Phylogenetic Distance (PD) [48] over OTU counts; however, the choice of metric

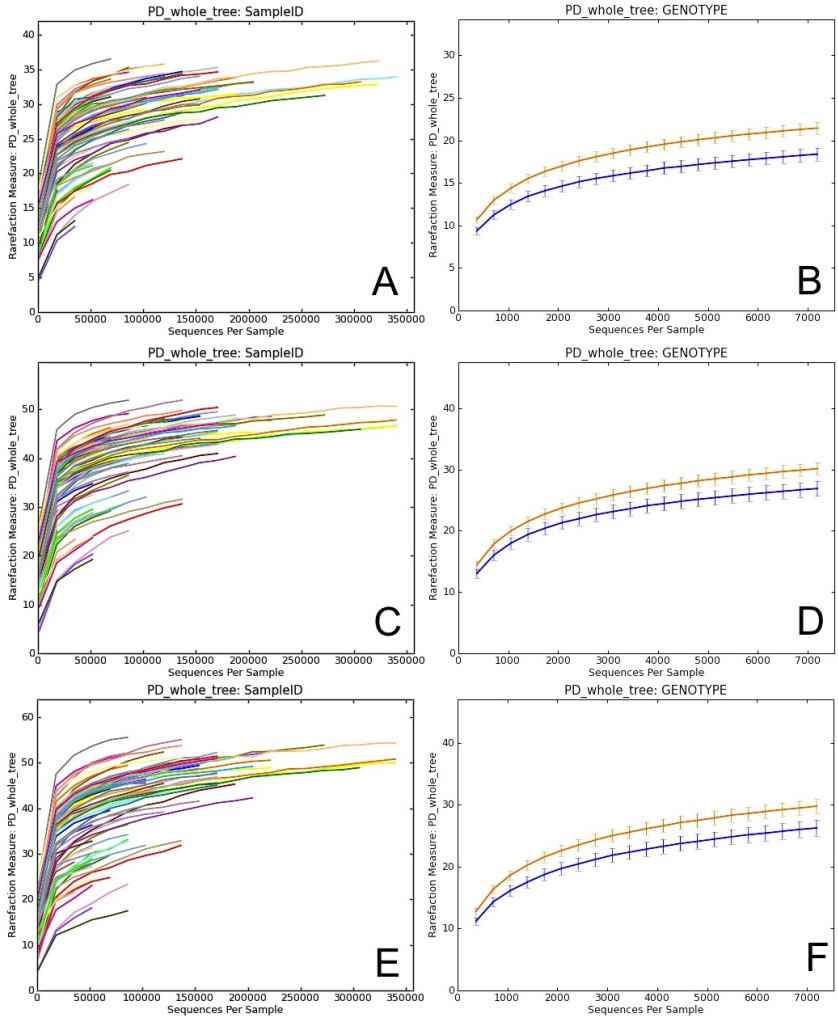


Figure 2.6: **Alpha diversity curves at different rarefaction depths using different OTU picking methods.** Each line represents the results of the alpha diversity phylogenetic diversity whole tree metric (PD Whole Tree in QIIME). A, C and E represent alpha diversity of each sample at a different sequence depth in each of the OTU picking protocols (closed-, open-reference and de novo). In closed-reference, the diversity plateaus (reaches an asymptote) because only OTUs in the reference database already can be considered, greatly reducing the OTU number over what is possible if the sequences are clustered de novo. Comparing these curves is difficult because the sequencing depth differs among samples. B, D and F show differences in alpha diversity between the two mouse genotypes, wild type (WT - orange) and transgenic (TG - blue), using the different OTU picking approaches. Both curves show the same rarefaction levels, allowing easier comparisons between categories. The curves again level off, showing that the sequencing effort is sufficient to detect most of the OTUs (this saturation can be confirmed using Good's coverage, or conditional uncovered probability, or other formal coverage statistics). The error bars show the standard error of the mean diversity at each rarefaction level across the multiple iterations.

will depend on the question. In particular, one might be interested in pure estimates of community richness (such as the observed number of OTUs, or the Chao1 estimator of the total number that would be observed with infinite sampling), in pure estimates of evenness, or of measures that combine richness and evenness such as the Shannon entropy (if there is no hypothesis in advance about which richness measure is appropriate, remember to correct for multiple comparisons if many are applied to the same dataset). Here is an example of how to compute rarefaction curves for three different alpha diversity metrics using a QIIME parameters file:

```
echo  alpha_diversity:metrics shannon,\
      PD_whole_tree,observed_species  \
 > alpha_params.txt


alpha_rarefaction.py \
 -i $PWD/open_ref_otus/otu_table_filtered.biom \
 -m $PWD/IQ_Bio_16sV4_L001_map.txt \
 -o $PWD/diversity_analysis/alpha_rare_open_ref_uneven \
 -a -O 64 -n 20 --min_rare_depth 1000 -e 340000 \
 -p $PWD/alpha_params.txt \
 -t $PWD/open_ref_otus/rep_set.tre
```

This step generates an interactive HTML document with figures showing the results for each alpha diversity metric and for each group of samples. Curves reach

asymptotes when the sequencing effort (sequencing depth) does not contribute additional OTUs. In this sense, curves would differ in their shape as a function of the selected OTU picking method.

Comparisons should be adjusted to the same depth of sequencing. Rarefaction curves can be useful for assessing the sequencing effort sufficient for representing and comparing the microbial communities (Figure 2.6). However, although rarefaction curves were widely used during the era of Sanger sequencing, when most communities were undersampled, it is often more useful today to report the coverage and the estimated and observed numbers of OTUs at one rarefaction depth rather than to use a figure for rarefaction curves.

Beta diversity analysis. Beta diversity can also be calculated from the rarefied OTU tables, comparing the microbial communities based on their compositional structures. As with alpha diversity, QIIME can compute many phylogenetic and non-phylogenetic beta diversity metrics (shown by the command beta_diversity.py $-s$).Of these, we have found UniFrac to be most generally useful in revealing biologically meaningful patterns. Unifrac measures the amount of unique evolution within each community with respect to another by calculating the fraction of branch length of the phylogenetic tree that is unique to either one of a pair of communities [118]. QIIME implements several variants of Unifrac, including weighted and unweighted Unifrac. The weighted Unifrac metric is weighted by the difference in probability mass of OTUs from each community for each branch, whereas unweighted Unifrac only consider the absence/presence of the OTUs [120].

Weighted Unifrac is thus recommended for detecting community differences that arise from differences in relative abundance of taxa, rather than in which taxa are present. Like other metrics considering taxon abundance, weighted Unifrac is sensitive to the bias from DNA extraction efficiency, PCR amplification, etc.; this may explain why, in our hands at least, unweighted UniFrac has often provided results that correlate better with clinical or environmental variables than does weighted UniFrac. The choice of metrics is critical in beta diversity analysis as metrics differ substantially in their ability to detect clustering or gradient patterns among microbial communities on the same dataset [9, 166, 176]. See Kuczynski et al. [97] for a detailed discussion of the performance of different nonphylogenetic metrics.

QIIME calculates the beta diversities between each pairs of input samples, forming a distance matrix. The distance matrix then can be visualized with methods such as PCoA and hierarchical clustering, both of which have been widely used for data visualization for decades. PCoA transforms the original multidimensional matrix to a new set of orthogonal axes that explain the maximum amount of inertia in the dataset and the current implementation in QIIME scales to thousands of samples. We are currently evaluating approximate methods that will allow scaling to millions of samples [61]. QIIME allows the PCoA plots to be visualized interactively in 3-dimensions, currently using the KiNG viewer [27]. To assess the stability of the PCoA plot, jackknife resampling can be performed on the OTU table, repeating the PCoA procedure for each resampled table and plotting the aggregate results as confidence ellipsoids around the sample points (Figure 2.7).

Jackknifing is recommended because many diversity metrics, including UniFrac, are sensitive to the number of sequences per sample [119].

Taxonomic information can be displayed on top of the PCoA using biplots (Figure 2.8) (this analysis requires the output file from previous taxon summary step). The coordinates of a given taxon are computed as the weighted average of the coordinates of all samples, where the weights are the relative abundances of the given taxon in the set of samples. This plot is particularly suited for identifying taxa that drive the differentiation between groups of microbial communities.

Another popular method for finding relationships among samples is hierarchical clustering, which groups samples together into a tree. Although hierarchical clustering can be effective in some cases, it should be used with caution because the eye can easily be drawn to incorrect relationships (such as samples that are adjacent in terms of the order of their labels but are topologically far apart in the tree). In general, we recommend using PCoA as a method of detecting grouping in the data, but demonstrate hierarchical clustering here as an example. Here we analyze the beta diversity distance matrix using UPGMA, which forces the samples into an ultrametric tree (i.e. a tree in which the distance from the roots to the tips is the same for every tip) (Figure 2.9). The resulting tree file is in Newick format, and can be visualized by programs including TopiaryExplorer [154], the R package ape [152] and the package distory [26]. UPGMA can also be applied to the jackknifed subsamples to provide an estimate of the statistical confidence in the clustering, by showing the frequency of each nodes in the original full data set

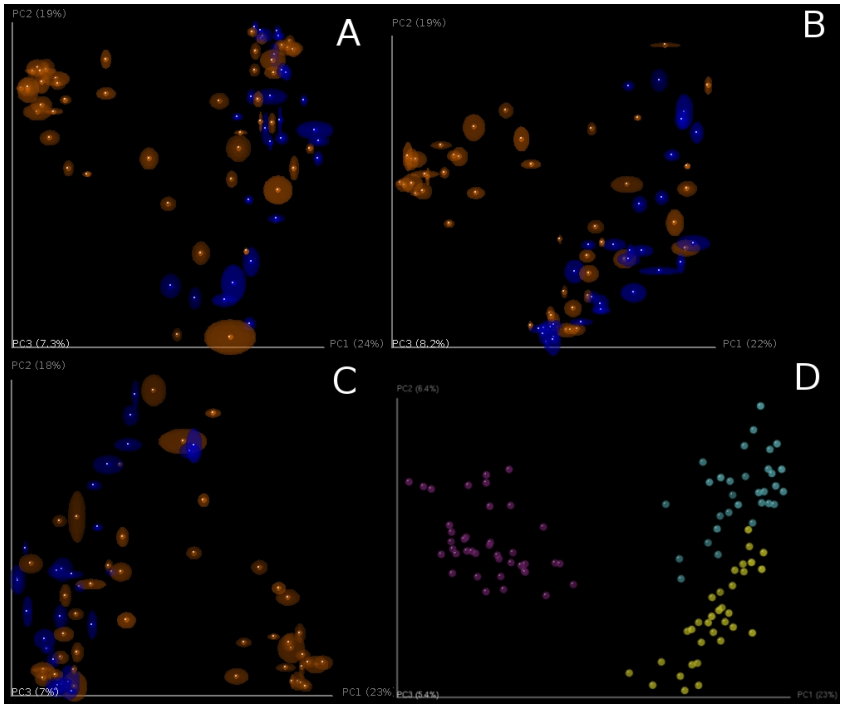


Figure 2.7: **PCoA plots of unweighted Unifrac beta diversity.** Panels A-C shows jackknifed replicate results for the example data set using de novo OTU picking, closed-reference OTU picking and open-reference OTU picking, illustrating different results from the three OTU picking approaches (Table 2.3). Each dot represents a sample, either from a WT mouse (orange) or TG mouse (blue). The two groups are not clearly separated, probably because the data set is contaminated (recall that this is a class project and different participants varied in their dissection skills). The size of the ellipsoids show the variation for each sample calculated from jackknife analysis. These plots are generated by the command jackknifed_beta_diversity.py -i $PWD/denovo_otus/otu_table_filtered.biom -t $PWD/denovo_otus/rep_set.tre -m $PWD/IQ_Bio_16sV4_L001_map.txt -o $PWD/diversity_analysis/jk_denovo -e 7205 -a -O 64 (the input parameters should be adapted for using the OTU tables from different OTU picking approaches). Panel D shows the beta diversity PCoA plot of a data set from the keyboard data set [50] which links individuals to their computer keyboard through microbial community similarity. Each dot represents a microbial community sampled from either fingertips or keyboard keys from three individuals, annotated by the three colors shown in the plot. In contrast to panels A-C, Panel D shows the microbial communities well-separated by individual in the PCoA plot.

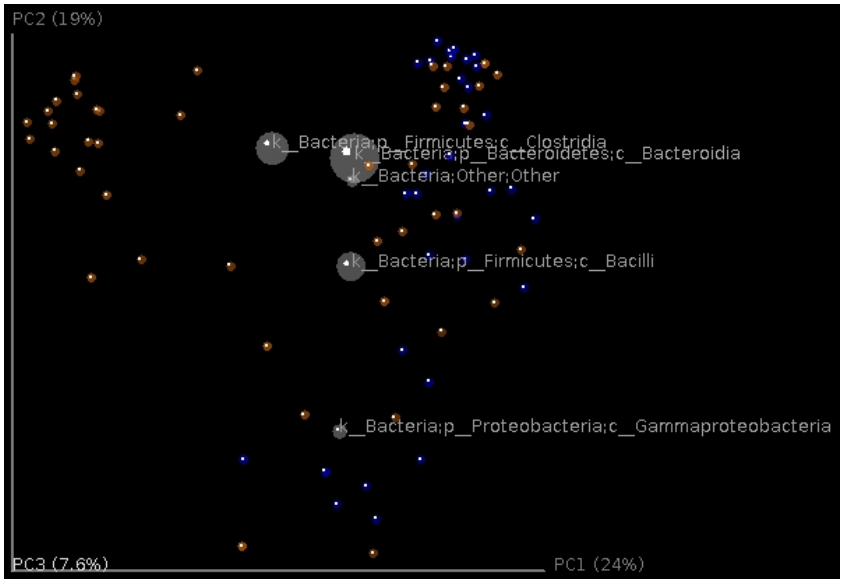


Figure 2.8: **Biplot of the example data set.** This is the unweighted Unifrac beta diversity plot, similar to Figure 2.7, with labels for the most 5 abundant phylum-level taxa added. The size of the sphere for each taxon is proportional to the mean relative abundance of that taxon across all samples. This plot is created by the command make_3d_plots.py -i $PWD/diversity_analysis/open_ref/bdiv_even7205/unweighted_unifrac_pc.txt -m $PWD/IQ_Bio_16sV4_L001_map.txt -t $PWD/diversity_analysis/open_ref/taxa_plots/table_mc7205_sorted_L3.txt –n_taxa_keep 5 -o $PWD/diversity_analysis/3d_biplot

cluster that are supported by the jackknife replicates. We generally recommend against the use of hierarchical clustering as a method for identifying and visualizing sample groupings, so have not invested as much effort in enabling this technique in QIIME as has been invested in other visualizations. However, if you do plan to use hierarchical clustering, it is important to be aware that substantial work has been done on more effective visualization methods, e.g. in distory [26], and performing additional analyses outside QIIME may allow improvements over the default visualizations.

Statistical significance of differences in alpha and beta diversity. Which statistical tests should be applied depends on the particular hypotheses and predictions defined *a priori* in a given research study. QIIME implements several scripts that perform a broad range of statistical tests between samples or groups of samples using both alpha and beta diversity measurements. For alpha diversity, the compare_alpha_diversity.py script performs comparisons between groups of samples. The script uses the alpha diversity measurements of samples standardized to a given number of sequences per sample, and performs nonparametric two-sample t-tests (i.e. using Monte Carlo permutations to calculate the p-value) comparing each pair of groups of samples. Rarefaction is a critical step in these analyses, as noted above, because typically diversity estimates depend on the number of sequences per sample. At the maximum rarefaction depth, wild type and transgenic mice did not show differences in alpha diversity as measured by PD metric (wild type: (mean +/- sd) = 45.19 +/- 10.6; transgenic: 40.01 +/- 9.5; t =

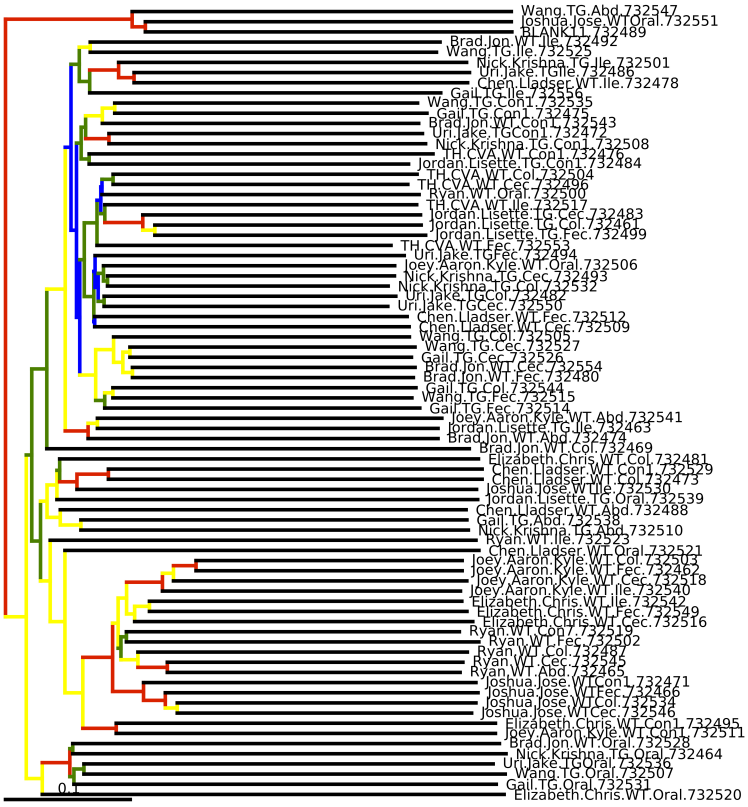


Figure 2.9: **Bootstrapped UPGMA clustering on the example data set.** The tree is shown with the internal nodes colored by bootstrap support (red: 75-100%, yellow: 50-75%, green: 25-50% and blue: $< 25\%$). Although this visualization is popular in the literature, we generally recommend alternatives such as PCoA.

-2.17, p = 0.102). We also tested for differences in alpha diversity between body sites. We found differences between cecum and ileum (cecum (mean +/- s.d.) = 51.1 +/- 3.6; ileum: 36.72 +/- 8.2; t = 5.35, p = 0.028), cecum and mouth (mouth: 29.54 +/- 10.1; t = 6.62, p = 0.028) and feces and mouth (feces: 48.4 +/- 4.0; t = 5.47, p = 0.028). None of the other pairs of comparisons between body sites showed significant differences in alpha diversity (colon: 46.0 +/- 9.2; multi-tissue: 46.26 +/- 9.1; skin: 42.13+/- 7.4; all p-values $>$ 0.056).

The appropriate statistical tests of beta diversity also depend on the research question being asked. These tests compare sets of distances between samples in the distance matrix. Careful attention must be paid both to Type I error (rejecting the null hypothesis when it is actually true), and to Type II error (accepting the null hypothesis when it is actually false, i.e. lack of statistical power). Type I error is more likely when variance is unequal between groups, and when many comparisons are performed on the same data (although multiple comparisons corrections correct for the increased Type I error, they often raise the Type II error rate instead). As always, results should be interpreted with caution and common sense. A highly statistically significant result stemming from data with a low correlation coefficient may indicate that a relationship has little biological meaning, and examining the scatterplot to see if the result is driven by a few outliers would be prudent. Further theoretical validation (especially of the multivariate statistical tests) is also needed, especially because the distributions underlying microbial community data have in general not yet been well characterized.

Comparisons between distance matrices are performed in QIIME using the compare_distance_matrices.py script. This script can perform analyses including the Mantel test, the Partial Mantel test, and the Mantel Correlogram. The Mantel test is a non-parametric test that compares two distance matrices, and calculates a correlation coefficient and a significant p-value using permutations that preserve the rows and columns. For the purpose of showing some examples (because the mouse data does not include a time series component), we will use the sequence dataset published by Caporaso et al. [21], where the authors studied variation in the bacterial community in the human gut over time series. We will compare the Unifrac distance matrix and a distance matrix as differences in days since the treatment started. Both distance matrices showed a significant correlation (Mantel test: $p = 0.035$), showing that bacterial communities were more similar as they were close in sampling. The Mantel test measures the overall correlation between distance matrices, but Mantel Correlograms measure this effect when taking into account the distances between samples marked by specific metadata variables. Essentially, the second distance matrix (in our case, days since the treatment started) is divided into classes. The classes into which the second distance matrix (days after experiment started) is determined by Sturge's rule, a method for determining the width of bars in a histogram based on the binomial formula. Then Mantel tests are run between these distance classes and the beta diversity distance matrix. We found that none of the distance classes were significantly related to the bacterial community (Figure 2.10: all comparisons $p > 0.120$, after Bonferroni correction

for multiple comparisons). The Mantel test showed us that there is an overall correlation between bacterial community and days after the experiment started (samples collected closer in time had more similar bacterial communities), and Mantel Correlogram showed that there is no significant correlation between the bacterial community and any of the classes into which the days after the experiment started matrix was divided. In other words, in this case, discretization of the data into a few timepoint classes led to an undetectable pattern; in contrast, use of the whole time series yielded an interpretable result. However, in other datasets, the reverse is often true, especially if the variation is not monotonic (e.g. in the case of seasonal variation).

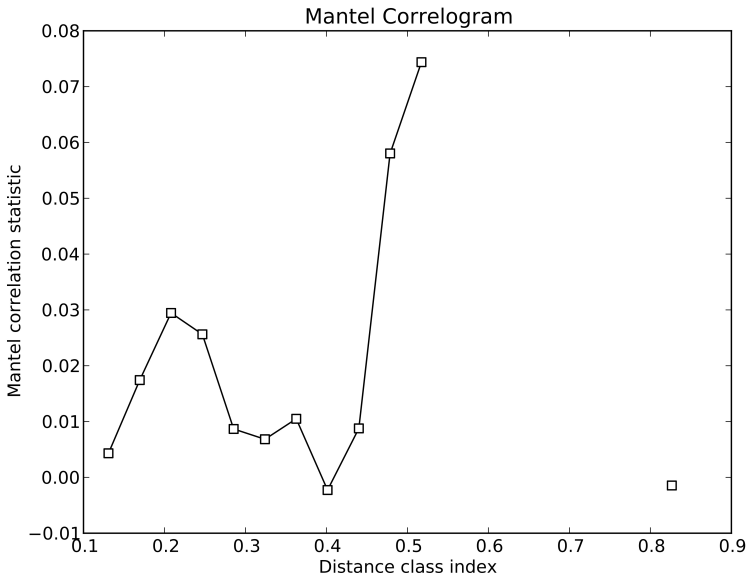


Figure 2.10: **Mantel Correlogram showing the Mantel correlation statistics between unweighted Unifrac distance matrix and each class in the days after experiment started distance matrix.** Classes in the second distance matrix are determined by Sturge's rule. White dots show non-significant relationship since black dots show significant ones.

The partial Mantel test is similar to the Mantel test, except that the analysis is controlled by a third variable. When we compare the beta diversity distance matrix with days after the experiment started by controlling by sampling date, we find the same trend noted before (Partial Mantel test: $p = 0.010$). Samples collected close in time have similar bacterial communities and this effect is independent of the date of collection.

Several visual and statistical tests have been implemented in QIIME in order to compare between and within beta-diversity distances. Distance histograms are an easy way to compare both types of distances graphically (make_distance_histograms.py). The output is an html file that shows as many histograms as categories. It is very useful to compare all-within category against all-between category, or the distribution of distances within each group (Figure 2.11). Probably a more useful tool to compare these beta-diversity distances is by means of box-plots (make_distance_boxplots.py, Figure 2.12). The box-plot script generates a box-plot graph and performs a t-test. Box-plots showed that there were no differences between the distances within mouse type and between types. However, the statistical test shows highly significant differences ($p < 0.001$) when comparing within and between distances. Once again, we recommend caution and common sense when the p-values are interpreted. It is likely to get a significant p-value, although a close inspection of the box-plot reveals that standard error bars overlap. Basically this result is due to the large number of comparisons: a small Student t-statistic (obtained when differences between two data set are small) and these

large degrees of freedom may be highly significant (i.e. the two data set are very different) even with conservative multiple test corrections (as Bonferroni).

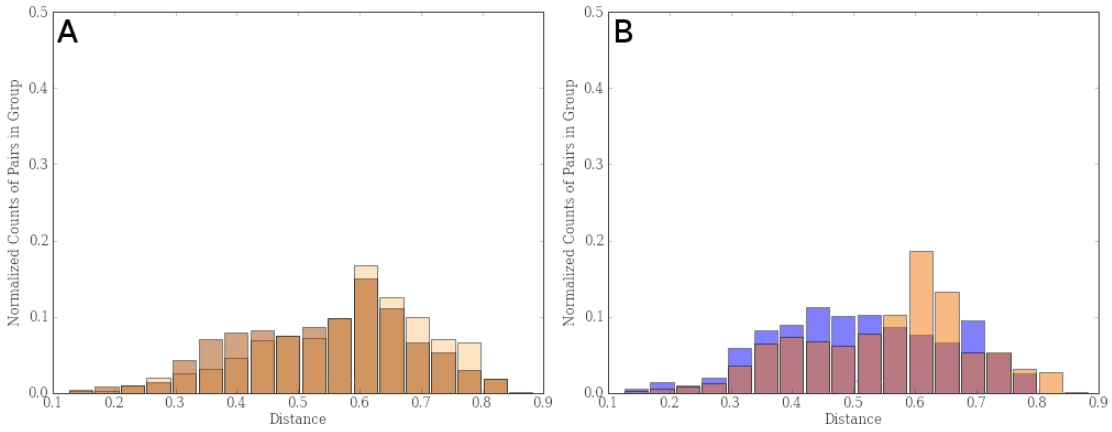


Figure 2.11: **Histograms of the example data set.** (A) Histogram showing distribution of distances between (light brown) and within (dark brown) mice gut microbiota taking into account both wild type and transgenic mouse groups. (B) Distribution of within distances in gut bacterial community of wild type mice (light orange) and transgenic ones (blue).

Other multivariate analyses provide additional powerful tools for exploring significant relationships between the beta diversity distance matrix and factors or covariates. compare_categories.py offer different statistical tests, where ANOSIM and adonis are usually employed. ANOSIM is a non-parametric statistical test that compares ranked beta-diversity distances between different groups and calculates a p-value and a correlation coefficient by permutation. Adonis partitions the variance in a similar way to the ANOVA family of tests, specifically testing variation within a category is smaller or greater than variation between categories. It calculates a pseudo F-value, a p-value and a correlation coefficient (R2). Significant p-values must be interpreted together with their R2 values to infer biological

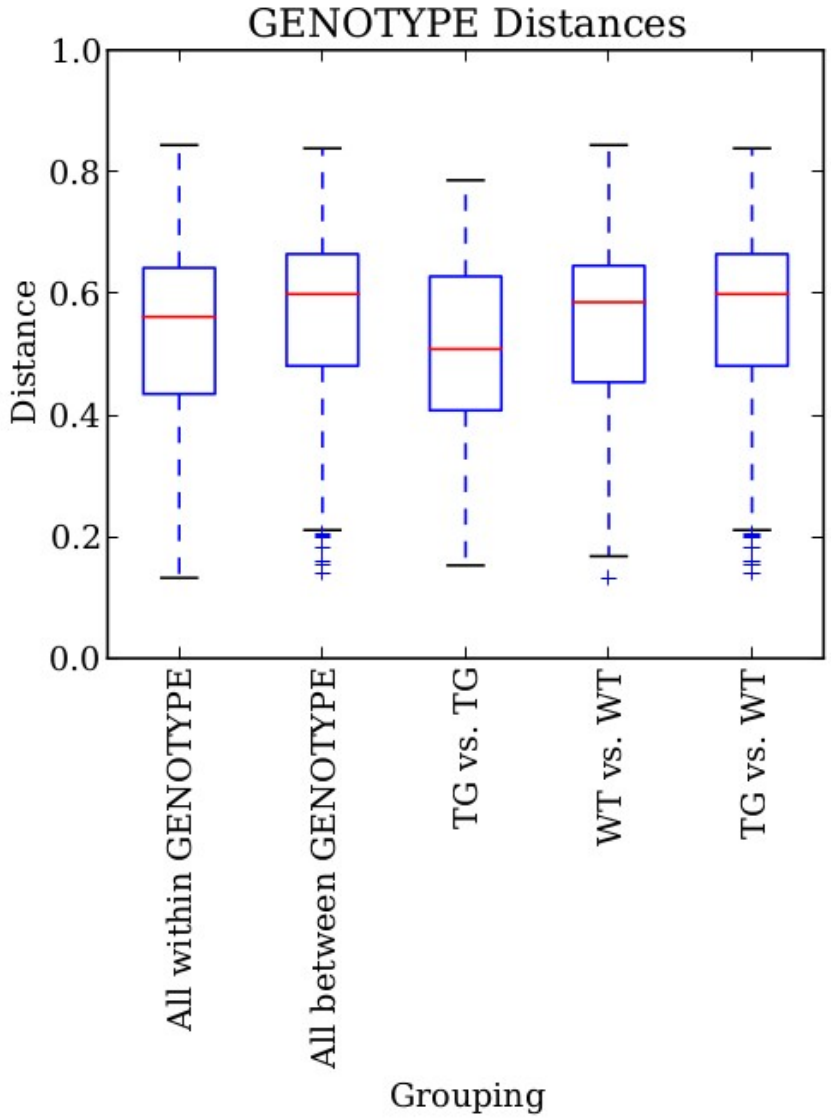


Figure 2.12: **Box-plots of the unweighted UniFrac distances for bacterial gut microbiota in both mouse type (WT: wild type; TG: transgenic).** Within distances represent distances within any of the two groups since between distances show distances between both groups. TG vs. TG and WT vs. WT represent within distances in transgenic and wild type groups respectively. Although averages are different, standard error overlaps in all cases.

meanings from the results. It is worth to mentioning here that PERMANOVA and adonis are similar statistical methods, and usually provide equivalent results. However, PERMANOVA only allows categorical factors, whereas both categorical and continuous variables may be used in adonis. Both ANOSIM and adonis analyses indicate that bacterial communities in wild-type and transgenic mice significantly differ from one another (ANOSIM: $r = 0.134$, $p < 0.001$; adonis, $r2 = 0.046$, $p < 0.001$). However, the correlation coefficients are low, so the significant p-values need to be interpreted cautiously because this result may not be biologically relevant.

OTU networks. Network-based analysis can sometimes be very useful for displaying how OTUs are partitioned between samples, and how samples are related each other, although we have found that this analysis only works well for datasets in which the samples are not all equally connected. Networks are therefore a powerful way for visually displaying certain large and complex datasets to emphasize similarities and differences among samples. Network analyses are implemented in QIIME through the script make_otu_network.py. This script generates the OTU network files to be passed into Cytoscape [180] and statistics for those networks (specifically, a bipartite graph in which nodes represent either OTUs or samples, and edges represent a connection between an OTU and a sample (Ley et al., 2008)). Cytoscape is not wrapped in the QIIME pipeline and it is run as a separate program. The files used by Cytoscape 2.8.2 are: the real edge table (real_edge_table.txt) which contains the columns from, to, eweight and consensus_-

lin, among others dictated by the headers in the mapping file; and the real node file (real_node_table.txt) which contains a node for each OTU and each sample in the study. It uses the OTU file and the user metadata mapping file.

The visual output of this analysis is a clustering of samples according to their shared OTUs (i.e. samples that share more OTUs cluster closer together, as do OTUs shared by more samples): samples and OTUs are represented as dots in the space (nodes) and connected by lines (edges). The degree to which samples cluster is based on the number of OTUs shared between samples, and this is weighted according to the number of sequences within an OTU.

In the network diagram, both types of nodes, OTU nodes and sample nodes, can be easily modified using Cytoscape's graphical user interface, with symbols such as filled circles for OTUs and filled squares for samples. If an OTU is found within a sample, both nodes are connected with a line (an edge). The nodes and edges can then be colored to emphasize certain aspects of the data.

This method is not simply used for descriptive visualizations: the connections within the network can also be analyzed statistically to provide support for the clustering patterns displayed in the network. A G-test for independence is used to test whether sample nodes within categories (such as within a genotype, in our example mouse study) are more connected within than a group than expected by chance. Each pair of samples is classified according to whether its members shared at least one OTU, and whether they share a category. Pairs are then tested for independence in these categories (this asks whether pairs that share a category

also are equally likely to share an OTU). This statistical test can also provide support for an apparent lack of clustering when it appears that a parameter is not contributing to the clustering.

In our example dataset, mouse samples show some degree of clustering in the space depending on whether the genotype is wild-type or transgenic (Figure 2.13). These clusters in the network were significant different (G-test: $p < 0.001$). Surprisingly, bacterial communities of mice did not visually cluster by body site, although the statistical test shows highly significant differences in samples from different body sites. These results must be interpreted cautiously. The degrees of freedom in the statistical test depend on the number of comparisons so, highly significant results might be obtained even when differences between clusters are slight. In other cases, these differences are obvious and easy to interpret. In the first application of this analysis in microbial ecology, the gut bacteria of a variety of mammals was surveyed, and the network diagrams were colored according to the diets of the animals, which highlighted the clustering of hosts by diet category (herbivores, carnivores, omnivores). In a later meta-analysis of bacterial surveys across habitat types, the networks were colored in such a way that the phylogenetic classification of the OTUs was highlighted: this analysis revealed the dominance of shared Firmicutes in vertebrate gut samples versus a much higher diversity of phyla represented amongst OTUs shared among environmental samples [108].

This OTU-based approach to comparisons between samples provides a counterpoint to the tree-based PCoA graphs derived from the UniFrac analyses. In

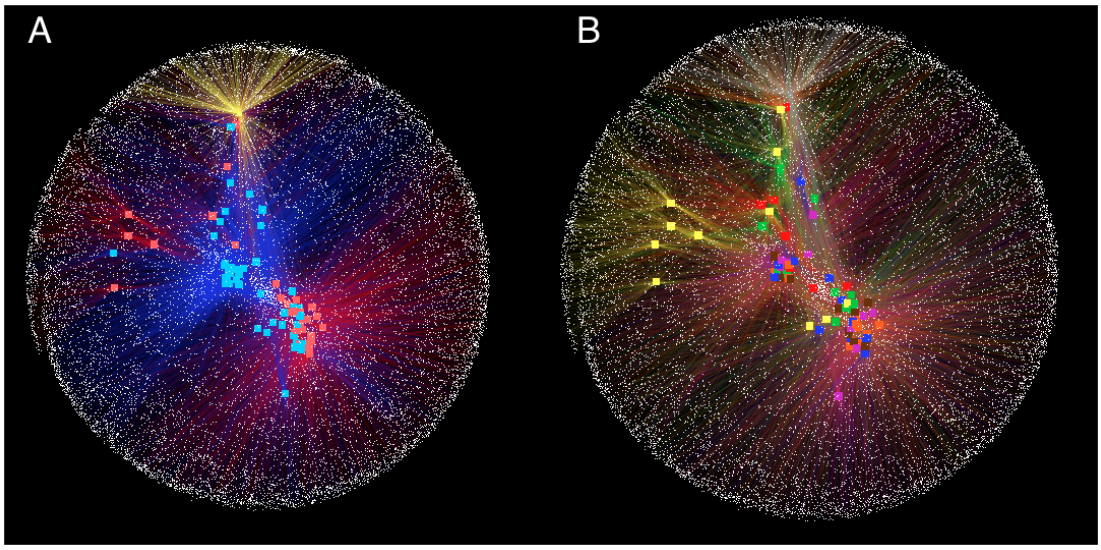


Figure 2.13: **OTU-Network bacterial community analysis applied in wild type and transgenic mice.** (A) Network colored by genotype (wild type: blue; transgenic: red). Control sample (yellow dot) is external in the network and several OTU are not shared with mice. Although we can see some degree of clustering, discrimination by genotypes is difficult to assess. (B) Network colored by body site (mouth: yellow; skin: in red; ileum: in blue; colon: in pink; cecum: in orange; feces: in brown; and multi-tissue samples: in green). A control sample is colored in grey. There is no clear sample clustering by body site, suggesting that there is not a core set of OTUs that differentiates one site from another.

most studies, the two approaches reveal the same patterns. They can, however, reveal different aspects of the data. The network analysis can provide taxonomic connections among samples in a visual manner, whereas PCoA-UniFrac clustering can reveal sub-clusters that may be obscured in the network. The principal coordinates can be pulled out individually and regressed against other metadata; the network analysis can provide a visual display of shared versus unique OTUs. Thus, together these tools can be used to draw attention to different aspects of a dataset.

OTU heatmaps. Another method to visualize the relationships between OTUs and samples is the heatmap, which is widely used for other applications in molecular biology [219]. This method was initially developed by Loua [117] to visualize population characteristics of 20 districts of Paris.

In our case, heatmaps can be used for exploratory analysis of microbiomes by mapping abundance values to a color scale in a condensed, pattern-rich format, in which each row corresponds to an OTU and each column corresponds to a sample. A good heatmap graphic can generate hypotheses about sample and/or OTU clustering in the data, which can then be followed up with additional more formal analyses. Two key structural aspects of a heatmap graphic greatly affect whether it will reveal interpretable patterns: (1) the ordering of the axes, and (2) the color scaling.

QIIME can create OTU heatmaps using two different scripts: make_otu_heatmap.py and make_otu_heatmap_html.py. The first script generates a heatmap

in which OTUs are represented in rows and samples in columns. OTUs and samples can be sorted and clustered by the phylogenetic tree and by the UPGMA hierarchical clustering, respectively. However, the visualizations of both trees (phylogenetic and hierarchical) in the final heatmap are not currently implemented directly in QIIME, and these hierarchical displays must be prepared using external software such as R. QIIME also supports sample clustering by a metadata category if the user provides a mapping file. The samples will be clustered within each category level using Euclidean UPGMA. The script sort_otu_table.py allows sorting the OTU table by a category in the mapping file, allowing defining the order of the samples in the heatmap. Figure 2.14 shows the output of make_otu_heatmap.py. There we can see a drawback to heatmaps: when the number of samples or OTUs included in the graphic is too high, the density of the graphic can be overwhelming. Thus, we recommend that the OTU table be filtered to a smaller number of samples (or categories) and taxa to identify the most important patterns, as we will show later in this section.

The second script (make_otu_heatmap_html.py) creates an interactive OTU heatmap from an OTU table (Figure 2.15). This script parses the OTU count table and filters the table by counts per OTU (user-specified). It then converts the table into a javascript array, which can be loaded into a web browser. The OTU heatmap displays raw OTU counts per sample, where the counts are colored based on the contribution of each OTU to the total OTU count present in the sample (blue: contributes low percentage of OTUs to sample; red: contributes

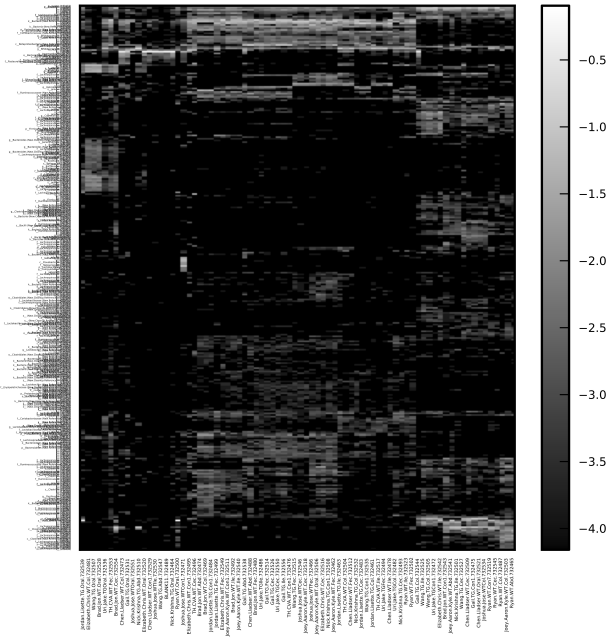


Figure 2.14: **Heatmap of OTUs present in the different samples from transgenic and wild type mice.** The intensity of black shows the abundance of certain OTU in each sample. Both samples and OTUs are sorted by UPGMA tree and the OTU phylogenetic tree, respectively.

high percentage of OTUs). This web application allows the user to filter the OTU table by number of counts per OTU. The user also has the ability to view the table based on taxonomy assignment. Additional features include: the ability to drag rows up and down by clicking and dragging on the row headers; and the ability to zoom in on parts of the heatmap by clicking on the counts within the heatmap.

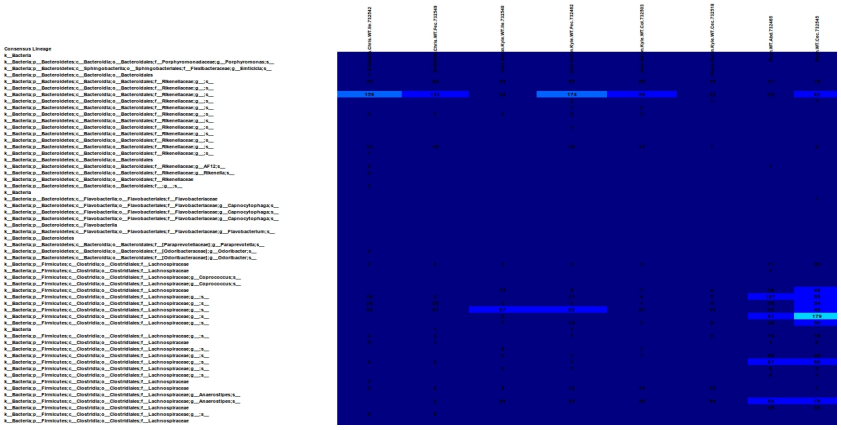


Figure 2.15: **Interactive heatmap of OTUs present in the different samples from transgenic and wild type mice.** This visualization is a result of an HTML file that can be opened in any web browser. The advantage of this heatmap is that it is easy to manipulate the abundance level for coloring, or transpose samples and OTUs between columns and rows.

Improved OTU heatmap visualizations can be generated using the plot_heatmap() command in the phyloseq package for R [134]. This package takes a similar approach to NeatMap [164], in that it uses ordination results rather than hierarchical clustering to determine the index order of each axis. For plot_heatmap, the default color scaling maps a particular shade of blue to a log transformation of abundance that generally works well for microbiome data, although the user can select alternative transformations.

In this example, a key step was proper filtering of the data. We removed OTUs that appear in only a few samples. The possible contribution to the graphic of these infrequent OTUs is limited, more often contributing to noise that causes the heatmap to look dark, empty, and uninterpretable (see Figure 2.14). We used a non-metric multidimensional scaling of the Bray-Curtis distance to determine the order of the OTUs and samples. From this representation, it is possible to distinguish high-level patterns and simultaneously note the samples and OTUs involved. For instance, all but a few of the mouth samples are in a cluster toward the middle of the heatmap. One of the key features of this group is an obvious relative overabundance of three Firmicutes OTUs, which are among the most abundant in this subset of the data. Similarly, another clear pattern is a distinction between a group of wild type samples from various body sites on the left of the heatmap that appear to have higher proportions of a number of different Firmicutes OTUs, as well as a few specific Bacteroidetes OTUs. This is distinct from the largest cluster of samples on the right-hand side of the heatmap, in which many of the most-abundant OTUs are a different subset of Bacteroidetes and Firmicutes OTUs. We also found it helpful to further pursue these high-level patterns by splitting the data into Firmicutes-only and Bacteroidetes-only subsets, and then plotting new heatmaps with finer-scale taxonomic labels. This required essentially the same commands and limited additional effort, well-tailored for exploratory interactive analysis, much of which we have documented in Supplemental File 1.

Although heatmaps have been deployed widely in molecular biology, espe-

cially in protein expression studies, some of the other displays we have discussed such as principal coordinates plots and taxonomy plots often provide more easily interpretable results. However, summarizing relations between taxa through ordination plots or network analyses have been shown to be powerful tools for highlighting similarities and differences among samples and taxa in our OTU table, and a carefully constructed heatmap (though not, in most cases, the default output) can be a useful guide to understanding and hypothesis generation.

OTU category significance. The experimental design of a microbial study will often involve comparing two or more groups for differences in the abundance of OTUs; for example, are there taxa that significantly differ between the control group and the experimental group? One way to assess this question is to compare the relative abundances of each microbial member between the two groups. This functionality is built into a script called otu_category_significance.py. We can test if there are significant differences in OTU abundance between mouse genotypes either wild type (WT) or transgenic (TG). We can assess differences between these groups using the following command:

```
otu_category_significance.py \
 -i $PWD/diversity_analysis/open_ref/table_mc7205.biom \
 -m $PWD/IQ_Bio_16sV4_L001_map.txt \
 -o $PWD/open_ref_otu_categ_sig_output -c GENOTYPE \
 -s ANOVA
```

Here we run an ANOVA to assess the relative abundance of each taxon in the OTU table between our two genotype groups. The output will be written to a user-specified file called otu_cat_sig.txt. This document will list the OTU ID, the raw p-value, the Bonferonni corrected p-value, the False Discovery Rate (FDR) p-value, as well as the relative average abundance for each of the groups in the selected category (genotype in our case), and the OTU taxonomy string (if provided in the initial OTU table). While many of these taxa may be significantly different between groups according to the raw p-value, it is extremely important that only p-values that have been corrected for against multiple comparisons, using either Bonferroni or FDR, be considered as significant. Many times a user's OTU table will contain hundreds or thousands of OTUs, and thus a p-value is likely to reach significance based solely on the large number of statistical comparisons being computed (for a probability threshold of 0.05, 1 of 20 comparisons results significant just by chance). It is often very helpful to open the .txt files produced by otu_category_significance.py in a spreadsheet so that columns can be sorted according to p-values.

The otu_category_significance.py script also contains several other statistics for comparing groups. The g-test can be used to determine if the presence or absence of a given taxa is significantly different between groups, and can be specified by passing the option -s g_test in the command. The user can also run a paired t-test to determine whether there are taxa that significantly differ between two paired points. For example, imagine the experimental design sampled a

group of mice before and after a dietary intervention. Using the paired-t statistic in otu_category_significance.py would then compare each mouse's after timepoint to the before timepoint, and test for differences that were consistent across mice, rather than grouping all the before and after timepoints together. For continuous variables, QIIME can calculate the Pearson correlations of OTU abundance with those variables. QIIME is also capable of longitudinal data analysis, which is suitable for the samples tracking the same subjects at multiple points in time, e.g., the oral microbiota of 6 persons after meals in a day. Specifically, longitudinal Pearson correlation can be calculated, accounting for intra-subject correlation of measurements.

Machine learning. QIIME can also take advantage of several machine learning algorithms to solve two important issues in high-throughput metagenomic studies: correction of mislabeling, and quantifying sample contamination.

This mislabeling problem is an increasing issue as the number of processed and pooled sequences increases [89]. This mislabeling can be addressed using supervised classifiers, a machine learning technique that is able to fix incorrect metadata. QIIME uses the random forest [16] supervised classifier implemented in R [113] to recover the mislabeled samples by training the classifier with the relative abundance taxa [87]. Knights et al. [89] shows that this approach can even recover up to 30-40% mislabeled samples when the biological patterns are especially clear.

This same technique can be also applied to find taxa that play a key role in differentiating groups of samples, as is done in OTU category significance. How-

ever, the difference between OTU category significance and the machine learning technique is the type of model the construct. While the OTU category significance creates an explanatory model (i.e. it gives a model that best fits the current dataset), the machine learning technique creates a predictive model [87].That is, it creates a model that is able to generalize future data, minimizing the expected prediction error.

Since the supervised learning trains a classifier, it is important to provide useful predictors (OTUs in our case). Thus, it is highly recommended to filter the input OTU table to remove those OTUs that are present in few samples (e.g. < 10 samples). As in previous analyses, a rarified OTU table should be used, so that artificial diversity induced due to different sampling effort is removed. In our example dataset, we can use the subsampled OTU table generated for previous analyses and remove the low-abundance OTUs:

```
filter_otus_from_otu_table.py \
 -i $PWD/diversity_analysis/open_ref/table_mc7205.biom \
 -o $PWD/diversity_analysis/open_ref/\
    otu_table_filtered10.biom \
 -s 10
```

Running the following command, will run the supervised learning algorithm using the GENOTYPE category and 10-fold cross-validation, providing mean and standard deviation of errors:

```
supervised_learning.py \
 -i $PWD/diversity_analysis/open_ref/\
     otu_table_filtered10.biom \
 -m $PWD/IQ_Bio_16sV4_L001_map.txt -c GENOTYPE \
 -o $PWD/open_ref_supervised_learning_output -e cv10
```

This script will store several files on the output folder. The most important file is summary.txt:

```
cat $PWD/open_ref_supervised_learning_output/summary.txt
Model Random Forest
Error type 10-fold cross validation
Estimated error (mean +/- s.d.) 0.23373 +/- 0.15058
Baseline error (for random guessing) 0.42308
Ratio baseline error to observed error 1.81011
Number of trees 500
```

The important information in this file is the Ratio baseline error to observed error, which shows the ratio between the expected error of the random forest classifier and the expected error of a classifier that always guesses the most abundant class (Baseline error). Our recommendation is that a ratio of at least 2 shows a good classification. In our example data set, this value is 1.81011, which is close to 2 but not enough to be considered a good classification.

The contamination quantification problem is addressed in QIIME using

SourceTracker [88]. Given a list of known source environments and a sink (or set of sinks) environment(s), SourceTracker uses a Bayesian approach jointly with Gibbs sampling to predict the quantity of taxa that each source, or an unknown source, contributes to the taxa that makes up the sink environment. For a more detailed description of the algorithm, see Knights et al. [88].

The first step to use SourceTracker in QIIME is to modify the mapping file of our example dataset and add two columns: SourceSink and Env. The SourceSink column tells SourceTracker which sample is a source and which sample is a sink, while the Env column provides the environment. In our example, we have defined samples from mouth, ileum, cecum, colon, fecal pellet and skin as sources and the whole mouse homogenization as a sink. In the Env column we have defined the environments as the body site (mouth, ileum, cecum, colon, feces, skin and homogenization).

As a machine learning algorithm, SourceTracker needs useful OTUs (predictors) as inputs for training the algorithm. Here, we will use the same OTU table as used for the supervised_learning.py script. However, SourceTracker does not yet accept BIOM tables, so we have to transform them into to a tab-delimited OTU table (note that this table can also be opened in Excel or other popular tools):

```
convert_biom.py \
 -i $PWD/diversity_analysis/open_ref/\
    otu_table_filtered10.biom \
```

```
 -o $PWD/diversity_analysis/open_ref/\
     otu_table_filtered10.txt -b
```

Then, we can call SourceTracker using the following command (the $SOURCETRACKER_PATH variable should be defined if you have successfully install SourceTracker):

```
R --slave --vanilla --args \
 -i $PWD/diversity_analysis/open_ref/\
     otu_table_filtered10.txt \
 -m $PWD/IQ_Bio_16sV4_L001_map_ST.txt \
 -o $PWD/open_ref_sourcetracker_output \
 < $SOURCETRACKER_PATH/sourcetracker_for_qiime.r
```

The output from the SourceTracker algorithm is a set of pdf files that shows the mixture of the sources that makes up the sink (see Figure 2.16).

Procrustes analysis. When we want to compare samples in PCoA space that were processed in different ways, such as: different ribosomal RNA subunits, primer sets, or algorithmic choices for processing, we can use Procrustes analysis [66, 140, 208]. Procrustes analysis is a statistical shape algorithm that allows us to compare different distributions by rescaling and applying a rotation matrix; this is, if the group of samples we are have the same shape but in different size or orientation the algorithm will resize and rotate them to make the shapes fit. As an example, we present the results of comparing the different OTU picking

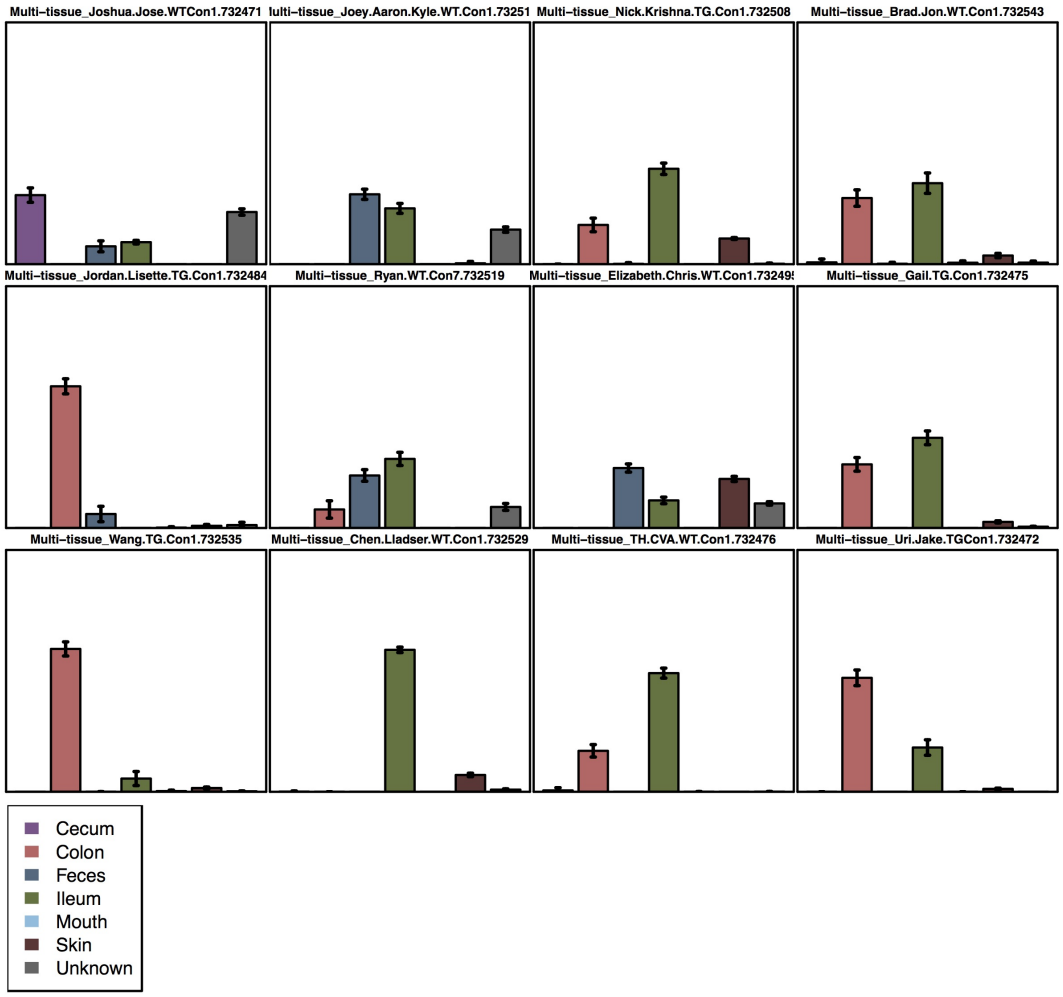


Figure 2.16: **SourceTracker output showing a bar plot for each sink (mouse) present in the dataset.** Each bar is a potential source (body site) and the height of each bar represents the percentage of taxa the source contributes to the taxa in the sink. The advantage of this visualization over the other two (area and pie chart) is that it shows error bars that allow to see the variance of the prediction.

algorithms, where we can see that even as the number of OTU clusters change the distribution described is similar with a confidence of MC p-value: 0.00 and M2: 0.097 for closed-reference vs. de novo, and MC p-value: 0.00 and M2: 0.035 for closed-reference vs. open reference. Both cases used the first three axes (i.e. the axes displayed in the plot), and 100 repetitions, Figure 2.17. To generate these plots we ran these commands:

```
transform_coordinate_matrices.py \
 -i $PWD/diversity_analysis/closed_ref/bdiv_even7205/\
 unweighted_unifrac_pc.txt,$PWD/diversity_analysis/denovo/\
 bdiv_even7205/unweighted_unifrac_pc.txt \
 -r 100 -o $PWD/procrustes/closed_ref-denovo

compare_3d_plots.py \
 -i $PWD/procrustes/closed_ref-denovo/pc1_transformed.txt,\
 $PWD/procrustes/closed_ref-denovo/pc2_transformed.txt \
 -o $PWD/procrustes/closed_ref-denovo/plot \
 -m $PWD/IQ_Bio_16sV4_L001_map.txt

transform_coordinate_matrices.py \
 -i $PWD/diversity_analysis/closed_ref/bdiv_even7205/\
 unweighted_unifrac_pc.txt,$PWD/diversity_analysis/\
```

```
 open_ref/bdiv_even7205/unweighted_unifrac_pc.txt \
 -r 100 -o $PWD/procrustes/closed_ref-open_ref

compare_3d_plots.py \
 -i $PWD/procrustes/closed_ref-open_ref/\
 pc1_transformed.txt,\
$PWD/procrustes/closed_ref-open_ref/\
 pc2_transformed.txt \
 -o $PWD/procrustes/closed_ref-open_ref/plot \
 -m $PWD/IQ_Bio_16sV4_L001_map.txt
```

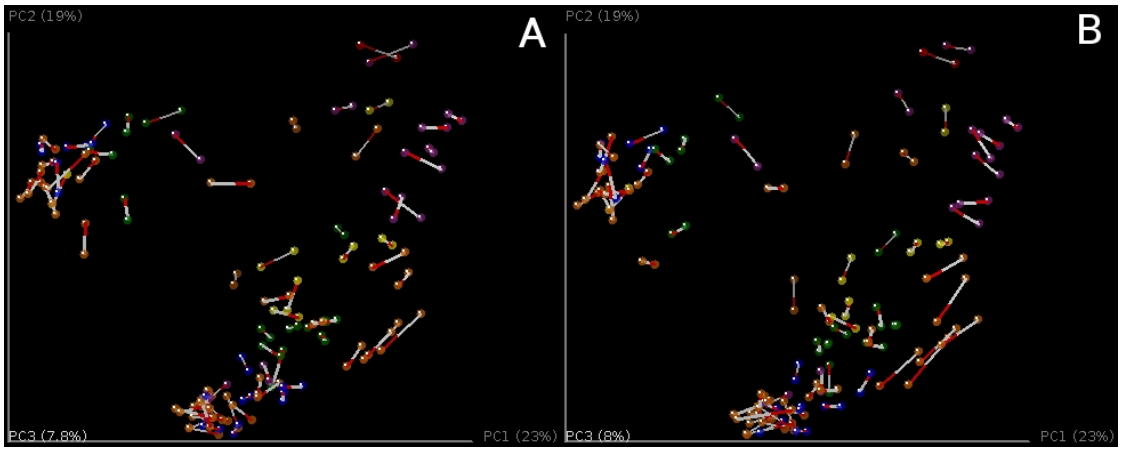


Figure 2.17: **Procrustes analysis of different picking algorithms, where we can see that different OTU clustering methods yield similar PCoA distributions.** PCoA plots are colored by BODY_HABITAT. A) Comparing samples with clusters picked using the de novo picking protocol against the closed-reference. B) Comparing samples with clusters picked using the open-reference picking protocol against the closed-reference.

SitePainter. Spatial data poses unique challenges, and the types of statis-

tical analyses described above often obscure spatial patterns [56, 76]. SitePainter [62] is a web-based tool that creates images representing the geographical (spatial) distribution of our samples, and then color them based on taxonomy summaries (defining which taxa occur where), and PCoA axes (defining how similar the patches are along the principal axes).

To create a new image we suggest using Adobe Illustrator, Inkscape or SitePainter. This list is in descending order of usability. In any of these tools, we need to create a Scalable Vector Graphics (SVG) image that has closed paths, ellipsoids and rectangles for any path that we want to color; and open paths, lines or text for those that we want SitePainter to ignore. The latter are useful for static images and give a nice background for the image. Note that SVG images are text files, so they can be opened in any graphics program in the list above, or in any text editor. The difference between an open and closed paths is that the element in has a letter z at the end of the definition of the lines of the path, so, for example, `<path d=M 10 10 L 30 10 L 20 30 z>` is a closed path but `<path d=M 10 10 L 30 10 L 20 30>` is an open one.

There are two main QIIME-generated inputs that should be loaded into SitePainter: taxa summaries and Multidimensional Scaling (MDS) technique results, including NMDS and PCoA. To exemplify the creation and usage of images in SitePainter, we will filter the OTU table and the beta diversity file to only have one mouse. Filtering and summarizing the OTU table:

```
filter_samples_from_otu_table.py
 -i $PWD/diversity_analysis/open_ref/bdiv_even7205\
 /table_mc7205_even7205.biom \
 -m $PWD/IQ_Bio_16sV4_L001_map.txt \
 -o $PWD/forSitePainter/otu_table_Gail.biom \
 -s  GROUP:Gail


summarize_taxa.py \
 -i $PWD/forSitePainter/otu_table_Gail.biom \
 -o $PWD/forSitePainter/taxa_sum -t
```

Filtering the beta diversity file and then recalculating PCoA is necessary every time we add or remove samples of our analyses, because PCoA results depend on the samples included in the analysis. Thus it is not sufficient to simply remove samples from PCoA results calculated on a larger set of samples:

```
filter_distance_matrix.py \
 -i $PWD/diversity_analysis/open_ref/bdiv_even7205\
 /unweighted_unifrac_dm.txt \
 -m IQ_Bio_16sV4_L001_map.txt \
 -o $PWD/forSitePainter/unweighted_unifrac_dm.txt \
 -s  GROUP:Gail
```

```
principal_coordinates.py \
 -i $PWD/forSitePainter/unweighted_unifrac_dm.txt \
 -o $PWD/forSitePainter/unweighted_unifrac_pc.txt
```

Then we create an image in Adobe Illustrator that represents the mice and its gastrointestinal tract, Figure 2.18-A. Once this figure is created and saved in SVG format (this example uses version 1.1 of SVG), we open the image in any text editor and replace any letter z with nothing; this will destroy all the closed paths and will facilitate manipulation in SitePainter.

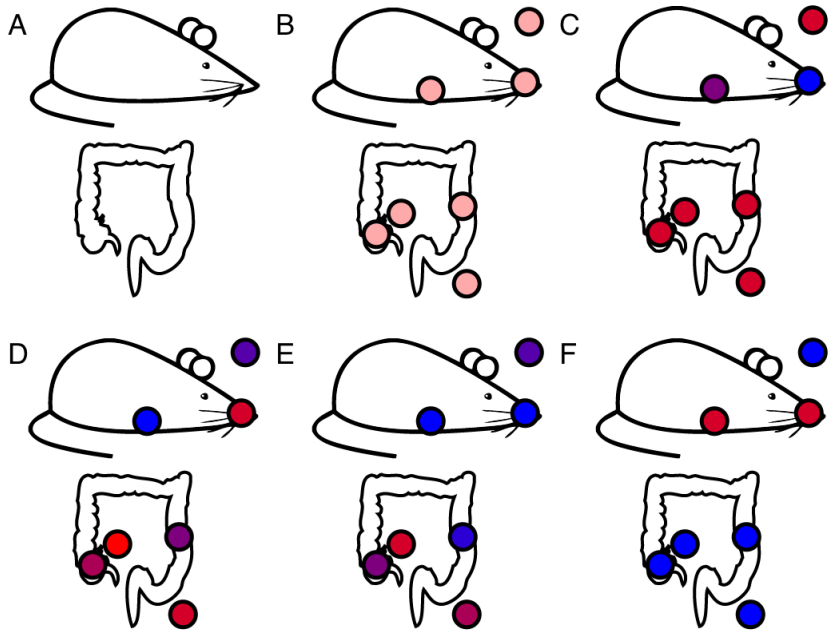


Figure 2.18: **Image representing the mouse and its gastrointestinal tract.** A) Raw image without samples. B) Image in SitePainter with samples. C-D) PCoA axis 1 and 2, in red high values, in blue low values, similar colors represent similar communities. E-F) Taxonomic distributions of (E) Betaproteobacteria and (F) Gammaproteobacteria, in red high abundance, in blue low abundance.

Now, we can open this image in SitePainter by clicking on the pencil/flower

image on the right corner, choosing Open Image, and select our file. Then we add the places that we want to color using the rectangle or ellipsoid tool, Figure 2.18-B. Now we need to make our samples in the image match the names of the sample names from our files; for this we need to click on `Elem. -> Click to update` on the right menu, this will show us the current sample names in the image; then, we double click on each one and change the name to make it match the sample name in the mapping file. Note that SitePainter does not accept sample names with dots (.), so if the sample name has this character, we need to replace it with an underscore (_). We do not need to change the QIIME files, as this will happen automatically in SitePainter. When we hover over each name, the sample will change color, facilitating the identification of the image we are selecting. If different sites have the same name, they will be colored with the same value from the QIIME output files.

The final step is to load the resulting QIIME files. To do this, we use the Metadata loader on the top left of the menu. This opens the file. We then move the right menu to the Meta. tab. Here we can select which column we want to use for coloring, and then click Color elements, to select more, Figure 2.18-C-F. For detailed instructions about changing colors and other details visit SitePainter's website [9].

[9] `http://sitepainter.sourceforge.net/tutorials/index.html`

2.1.5 Other features

Testing linear gradients, including time series analysis

Recent microbiome surveys have started integrating gradients (commonly over time) in their study design. We will discuss a first and general approach for those cases, using the Moving Pictures of the Human Microbiome Dataset [21], where two subjects were sampled daily for up to 396 days in three different body sites (sebum, saliva and feces). Note that the mouse dataset that we use as a primary example lacks a natural temporal ordering in the study design, so we can not use it as an example for this analysis.

PCoA plots provide a snapshot about the relative communities of many samples condensed in a single figure. However, coloring the points in PCoA space according to a color gradient can be very difficult to understand. A first approach in this case is to connect the samples belonging to the same subject/treatment subsequently sorted using the values in the gradient, i.e. one timepoint after the other (see Figure 2.19 b). An interactive plot like this can be generated using the following command:

```
make_3d_plots.py \
 -i $PWD/moving_pictures/unweighted_unifrac_pc.txt \
 -m $PWD/moving_pictures/merged_columns_mapping_file.txt \
 -o $PWD/moving_pictures/vectors \
 --add_vectors=BODY_SITEHOST_SUBJECT_ID,DAYS_SINCE_EPOCH
```

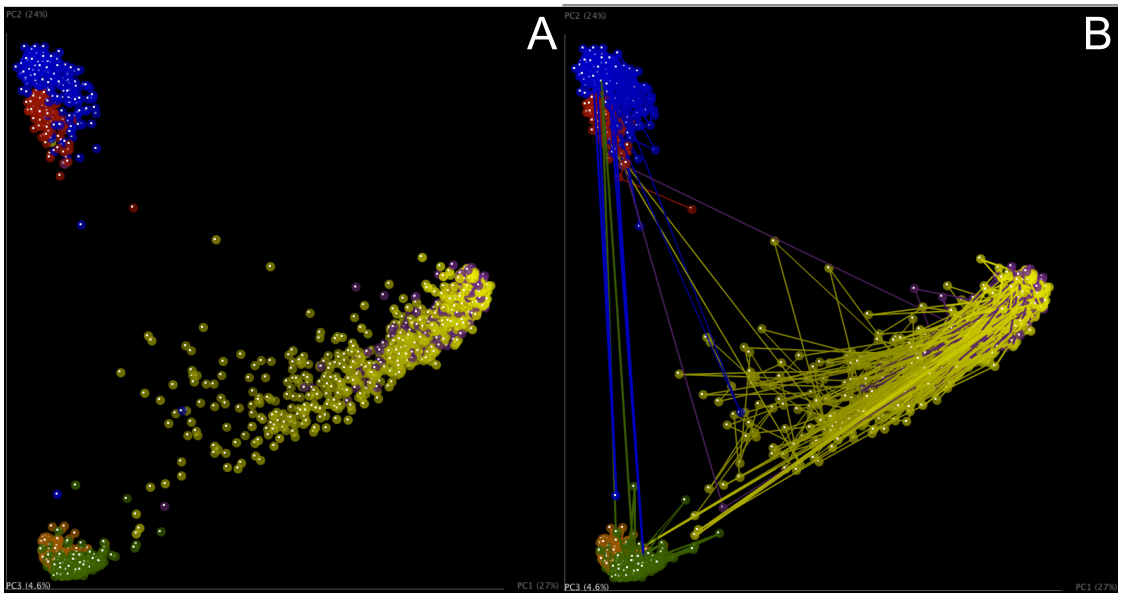


Figure 2.19: **Beta diversity plots for the moving pictures dataset using unweighted UniFrac as the dissimilarity metric.** (a) PCoA plot colored by the body site and subject. (b) PCoA plot colored by the body site and subject with connecting lines between samples. Note in (b) that these lines allow us to track the individual body sites with a different approach.

An important thing to note here is that because we want to track each of the three body-sites (SampleTypes) for the two subjects (Subject), we need a column in our mapping file that allows us to make that distinction. Hence we need to concatenate those two columns in our metadata mapping file using an external spreadsheet editor or another tool. Also note that the gradient used is a category named DAYS_SINCE_EPOCH (i.e. the number of days since January 1, 1970). The idea here is to have a common reference for the collection date of each of the samples.

Although a visualization like the one created in the previous example is often sufficient, replacing one of the axes in the PCoA plot with the data explaining

the gradient provides a different insight into the analyzed data (See Figure 2.20).

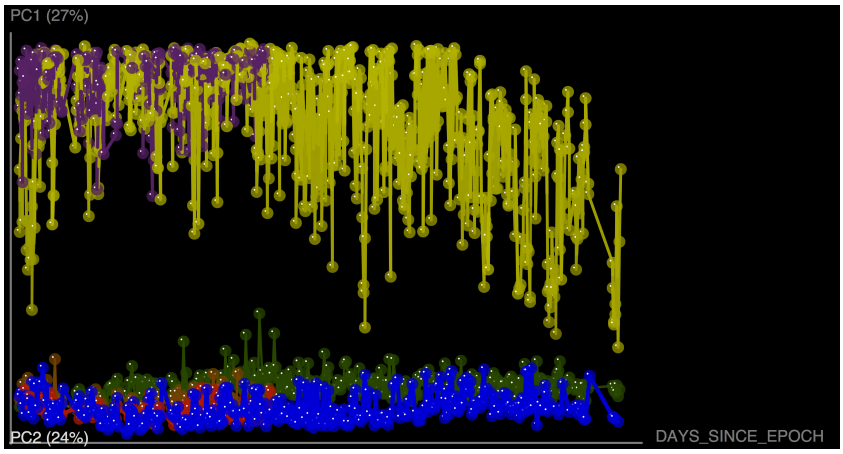


Figure 2.20: **Three dimensional plots in which two of the axes are PC1 and PC2 and the other is the day when that sample was collected in reference to the epoch time.** Although this is not explicitly a beta diversity plot, this representation allows differentiation of the individual trajectories over time.

```
make_3d_plots.py \
 -i $PWD/moving_pictures/unweighted_unifrac_pc.txt \
 -m $PWD/moving_pictures/merged_columns_mapping_file.txt \
 -o $PWD/moving_pictures/vectors \
 --add_vectors=BODY_SITEHOST_SUBJECT_ID,DAYS_SINCE_EPOCH \
 -a DAYS_SINCE_EPOCH
```

These visual representations can often identify meaningful patterns. To statistically support these assertions, one-way analysis of variance (ANOVA) can be used over the values grouped by a category of interest. In a case where user wants to test for independence between the variation of one group of trajectories and another, this command could be used:

```
make_3d_plots.py \
    i unweighted_unifrac_pc.txt \
   m mapping_file.txt    o vectors \
  add_vectors=SampleTypeAndSubject,days_since_epoch \
    a days_since_epoch --vectors_algorithm avg \
--vectors_path anova_stats.txt
```

Processing 454 data

We have described the recommended workflow for conducting microbial community analysis on an Illumina MiSeq dataset. However, QIIME can also perform microbial community analysis on the 454 platform. The main advantage of 454 over Illumina is that 454 generates longer sequences, which can allow a better taxonomy assignment. However, the 454 technology produces fewer reads per dollar, or per sequencing run [96].

The 454 processing workflow differs from the Illumina workflow in the sequence preprocessing. In this case, the output file from the sequencing facility is a fasta file containing the reads, and a quality score file which contains the score for each base in each sequence included in the FASTA file. In this case, the command used for the 454 preprocessing is split_libraries.py:

```
split_libraries.py \
-m Fasting_map.txt -f Fasting_Example.fna \
```

```
-q Fasting_Example.qual -o slout
```

Similarly to the Illumina processing, this script also performs a quality filtering. In this case, the quality filtering is based on cut-offs for sequence length, end-trimming or minimum quality score. However, to successfully remove the read artifacts, a denoising process has to be performed [168] to reduce the impact of homopolymer runs (runs of the same base). The 454 denoising process is a slow, computationally intensive problem that does not scale to large datasets, as it is based on flowgram clustering [161].

Variable length barcodes Variable-length barcodes are used for two reasons: to make the number of flows (rather than the number of bases) constant [51], or to stagger the reads to reduce bad signal from low complexity at a given position in the set of amplicons being sequenced. This approach is not recommended today because such samples are not easily demultiplexed, and there is checksum, like Hamming or Golay, that allows error-correction and improved sample assignment [73]. However, the HMP used variable length barcodes to identify their samples within sequencing runs. Thus, QIIME allows demultiplexing such files by using the parameter -b in split_libraries.py, as follows:

```
split_libraries.py \
-m map_file_with_variable_length_barcodes.txt \
-f your_fna.fna -q your_qual.qual \
-o split_library_output_variable_length/ \
```

```
-b variable_length
```

18S rRNA gene sequencing

QIIME can be also used to perform analysis on 18S rRNA gene sequence data (in eukaryotes), as well as other markers such as Internal Transcribed Spacer (ITS). The main difference between performing analyses with 18S rRNA gene data instead of 16S rRNA gene data (or ITS data) is the reference database used for OTU picking, the taxonomic assignments and the template-based alignment building, since it must contain eukaryotic sequences.

The recommended database to use as a reference for 18S rRNA sequences is the Silva database [158]. At the time of writing, the most recent QIIME-compatible Silva database is the 108 release. Since this database contains the three domains of life, it can be used as a reference for 18S rRNA data sets.

When conducting studies mixing 18S rRNA data and 16S rRNA data, you should take into account that picking OTUs against the Silva database will assign taxa to all three domains of life. In this case, it is recommended to split the OTU table by domain, generating an OTU for each domain (Archaea, Bacteria and Eukarya). At this point, each of these tables can be used in downstream analysis in the same way as performed for 16S rRNA data.

Shotgun metagenomics

Shotgun metagenomics is also supported in QIIME, although it is still experimental and it should be used at the user's own risk. Currently, the QIIME team recommends the blat method [85] for searching nucleic acid sequence reads in a reference database, although usearch [43] is also supported. The main reason for preferring blat against usearch is that protein reference database often require 64-bit applications, and blat is free of charge, while the 64 bit version of usearch is not.

There are many reference databases (IMG, KEGG, M5nr, among others), and they all supported by QIIME, since the user only needs to supply a single fasta file containing the sequence records. The command that QIIME provides for mapping reads against the reference database is map_reads_to_reference.py, and it can be performed in parallel using the parallel_map_reads_to_reference.py script.

Support for QIIME in R

First published in 1996, R is an integrated software application and programming language designed for interactive data analysis (R Core Team). It is available for Linux, Mac OS, and Windows free of charge under an open-source license (GPL2). Since its inception, R has found a niche as a tool for interactive statistical analysis through functional programming. Primary investigation and inference are performed by writing a series of repeatable commands as scripts that can be recorded and published. This paradigm lends itself well to reproducible

research, and is enhanced substantially by R's integration with tools for literate programming such as Sweave [53], knitr [222], and R markdown [10], as well as data graphics. There are thousands of free and open-source extensions to R (packages) available from the main R repository, CRAN, further organized by volunteer experts into 31 task views (which are in fact workflow inventories). Among these are dedicated package lists relevant to microbiome data, including phylogenetics, clustering, environmetrics, machine learning, multivariate and spatial statistics, as well as a separate reviewed and curated repository dedicated to biological statistics called Bioconductor (over 600 packages).

At present, support for QIIME in R is predominantly achieved through a package called phyloseq [134] dedicated to the reproducible analysis of microbiome census data in R. phyloseq defines an object-oriented data class for the consistent representation of related (heterogenous) microbiome census data that is independent of the sequencing- or OTU-clustering method (storing OTU abundance, taxonomy classification, phylogenetic relationships, representative biological sequences and sample covariates). The package supports QIIME by including functions for importing data from biom-format files derived from more recent versions of QIIME (import_biom) as well as legacy OTU-taxonomy delimited files (import_qiime and related user accessible subfunctions). Later editions of phyloseq (>1.5.15) also include an API for importing data directly from the microbio.me/qiime data repository. In all cases, these API functions return an instance

[10] `http://CRAN.R-project.org/package=markdown`

of the phyloseq class that contains the available heterogenous components in native R classes. phyloseq includes a number of tools for connecting with other microbiome analysis functions available in other R packages, as well as its own functions for flexible graphics production built using ggplot2 [218], demonstrated in supplemental files and online tutorials. For researchers interested in developing or using methods not directly supported by phyloseq, nor its data infrastructure, the biom-format specific core functions in phyloseq have been migrated to an official API in the biom-format project as an installable R package called biom, now released on CRAN. This also includes some biom-format specific functionality that is beyond the scope of phyloseq, though support for QIIME is still likely best achieved using phyloseq.

As with some of the earlier examples of QIIME commands with corresponding output and figures, in this section we have included some key R commands potentially useful during interactive analysis in the R environment. For simplicity, show only results related to the open-reference OTU data, stored in an object in our examples named open, and imported into R using the phyloseq command import_biom.

```
open = import_biom( path-to-file.biom , )
```

Additional input data files can also be provided to import_biom, or merged with open after its instantiation. For clarity, subsets and transformations of the data in open are stored in objects having names that begin with open. As with the

remainder of the examples highlighted in this section, the complete code sufficient for reproducing all results and figures are included in the R Markdown originated document, Supplemental File 1, which includes several additional examples not shown here, and is available with supporting files on GitHub[11].

Although not always very illuminating, a comparison of OTU-richness between samples or groups of samples can easily be achieved with the plot_richness command. For the most precise estimates of richness for most samples, this should be performed before random subsampling or other transformations of the abundance data. Here open contains data that has already been randomly subsampled. In figure 2.21 we can see that the wild type samples are generally more diverse (higher richness) and somewhat more variable than the transgenic samples for essentially all body sites, though the differences between the two mice genotypes are small.

```
plot_richness(
  open, x= BODY_SITE ,
  color = GENOTYPE ) + geom_boxplot()
```

This plot command also illustrates the use of a function in ggplot2, geom_boxplot, that instructs the ggplot2 graphics engine to add an additional graphical element in this case a boxplot for each of the natural groups in the graphic. These available additional graphical instructions (called layers in the grammar of graphics nomenclature) are embedded with the returned plot object for subsequent

[11] https://github.com/joey711/navasetal

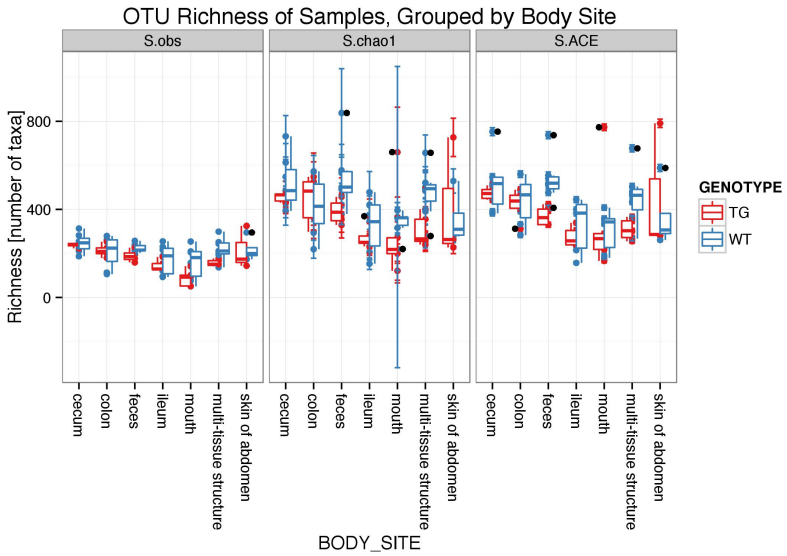


Figure 2.21: **Categorically summarized OTU richness estimates using the plot_richness function.** Samples are grouped on the horizontal axis according to body site, and color shading indicates the mouse genotype. The vertical axis indicates the richness estimates in number of distinct OTUs, and a separate boxplot is overlaid on the points for each combination of genotype and body site. The S.obs, S.chao1, and S.ACE panels show the rarefied observed richness, Chao-1 richness, and ACE richness estimates, respectively

rendering, inspection, or further modification, allowing for powerfully customized representations of the data.

Here is an example leveraging the abundance bar plot function from phyloseq, plot_barr, in order to compare the relative abundances of key phyla between the wild type and transgenic mice across body sites. The first step was actually some additional data transformations (not shown, see Supplemental File 1) in order to subset the data to only major expected phyla (subset_taxa), merge OTUs from the same phyla as one entry (merge_taxa), and merge samples from the same body site and mouse genotype (merge_samples).

```
p2 = plot_bar(
  openphyab, bodysite ,
  fill = phyla , title = title)
p2 + facet_gird(GENOTYPE)
```

From this first bar plot it is clear that all body sites from the average wild type mouse have Firmicutes as their phylum of largest cumulative proportion, except for the feces, where it is anyway a close call between Firmicutes and Bacteroidetes. By contrast, some of the average transgenic mice samples have a much higher proportion of Proteobacteria or Bacteroidetes than the corresponding wild type samples. One drawback to this type of stacked bar representation is that it is difficult to compare any of the sub-bars except for those at the bottom. If needed, this can be alleviated by changing the facet_grid call such that a separate panel is

made for each phyla in the dataset, as follows.

```
p2 + facet_grid(
  phyla ~ GENOTYPE) + ylim(0, 100)
```

With essentially the same effort to produce, the 14 panels of this second bar plot graphic allow an easy and quantitative comparison of the relative abundances of each phylum across body sites and genotype.

Microbiome datasets can be highly multivariate in nature, and dimensional reduction (ordination) methods can be a useful form of exploratory analysis to better understand some of the largest patterns in the data. Many ordination methods are wrapped in phyloseq by the ordinate function, and many more are offered in available R packages. Here we show an example performing multidimensional scaling (MDS) on the precomputed unweighted UniFrac distance matrix for the open-reference dataset. The ordination result (openUUFMDS) is first passed to plot_scree in order to explore the scree plot representing the relative proportions of variability represented by each successive axis. Both the ordination result and the original data are then passed to plot_ordination with sufficient parameters to shade the sample points by genotype, and create separate panels for each body site.

```
openUUFMDS = ordinate(
 open, "MDS",
 distance = UniFrac[["unweighted"]][["open"]])
```

```
plot_scree(openUUFMDS, "Unweighted Unifrac MDS")
plot_ordination(open, openUUFMDS, color = "GENOTYPE")
+ geom_point(size = 5) + facet_wrap( ~BODY_SITE)
```

It appears that a subset of the wild-type samples from all but the mouth and abdomen-skin body sites cluster toward the left of the plot. This appears to be the major pattern along the axis that also comprises the greatest proportion of variability in the dataset. At this stage of analysis it seems worthwhile to try to identify which OTU abundances are most different between these groups, and then perform some formal validation/testing of these differences.

2.1.6 Recommendations

Here, we highlight some of the main aspects to take into account when performing microbial community analysis:

- Use the open-reference OTU picking approach if your data allows it. It will reduce the running time and will recover all the diversity in your samples.

- Perform an OTU quality filtering based on abundance, by removing singletons, for instance. See [14] for further discussion on how to tune this quality filtering and its effects on downstream analysis. Quality filtering is critical for obtaining reasonable numbers of OTUs from a sample.

- Consider whether you need to remove specific taxa from your study, such

chloroplast or host DNA sequences when analyzing microbial datasets.

- Remove samples from your study that have low coverage (i.e. low OTU counts). They are likely uninformative and usually indicate low-quality reads.

- Rarefy your OTU table in order to mitigate the differences on the sequencing effort, so the downstream diversity analyses wont be biased by the artificial diversity generated due to the difference in sequencing depth.

2.1.7 Conclusions

QIIME is a powerful tool for the analysis of bacterial community allowing researchers to recapitulate the necessary steps in the processing of sequences from the raw data to the visualizations and interpretation of the results. Two advantages make QIIME very useful: fidelity to the algorithms used, and consistency in the analysis. Fidelity is obtained because QIIME wraps existing software, preserving the integrity of the original programs and algorithms designed, created, and tested by the original authors. Consistency is obtained because QIIME can be applied to sequences from different platforms, and once the upstream process is done; the analysis (downstream) process is the same independent of the sequencing platform used. These characteristics, together with the fact that QIIME is open-source software with continuous support to users via QIIME forum, have promoted the rapid increase in the QIIME user community since its publication [20].

Downstream and upstream processes are implemented in QIIME in a way

that offers several options to perform the analyses. In this review, we discuss and demonstrate the principles for each step, what the scripts do and how to choose between options. Independent of the use of QIIME, this review also provides an overview of many of the typical steps in a microbial community analysis based on analysis of 16S rRNA sequences produced by high-throughput sequencing. Some of these tools are well developed with a long history in general ecology, whereas others are still in rapid development; we encourage microbial ecologists and bioinformaticians to work together to create, develop and implement new strategies and tools that allow further exploration of this fascinating field.

2.1.8 Acknowledgments

We thank William A Walters and Jessica Metcalf for productive discussion and their useful comments about QIIME. We also acknowledge Manuel Lladser for helping collect the dataset and allowing us to use it, and the IQBio IGERT grant for funding data collection. JANM is supported by a graduate scholarship funded jointly by the Balsells Foundation and by the University of Colorado at Boulder. SH is partially supported by NIH grant R01 GM086884. This work was partially supported by the Howard Hughes Medical Institute.

2.2 Bottlenecks in large scale microbial studies: sequence clustering

Section 2.1 described a microbial community analysis pipeline in depth. In that pipeline, one of the most time-consuming steps is performing sequence clustering (also known as OTU picking). Sequence clustering is the processing step that groups sequences into OTUs based on sequence similarity. The OTUs found in a sample are used as an approximation of the species richness in the given niche. Sequence similarity is computed using pairwise sequence alignment [145, 185], an expensive computational task that is quadratic in the length of the input sequences and the number of input sequences. With datasets containing from a few hundred thousand reads to a few billion reads [64], performing all pairwise sequence alginments is too computationally expensive to be performed in a timely manner. The sequence clustering problem shares characteristics with the biological sequence database search problem, which has been studied for more than 30 years. Sections 2.2.1, 2.2.2 and 2.2.3, contain a summary of my contributions to optimize the sequence clustering step of microbial community datasets analysis.

Section 2.2.1 has been adapted from the original publication in "Subsampled open-reference clustering creates consistent, comprehensive OTU definitions and scales to billions of sequences". J. R. Rideout, Y. He, J. A. Navas-Molina, W.A. Walters, L. K. Ursell, S. M. Gibbons, J. Chase, D. McDonald, A. Gonzalez, A. Robbins-Pianka, J. C. Clemente, J. A. Gilbert, S. M. Huse, H. W. Zhou, R. Knight

and J. G. Caporaso. *PeerJ*, 2014, DOI: 10.7717/peerj.545.

Section 2.2.2 has been adapted from the original publication in "Open-source sequence clustering methods improve the State of the Art". E. Kopylova, J. A. Navas-Molina, C. Mercier, Z. Z. Xu, F. Mahe, Y. He, H. Zhou, T. Rognes, J. G. Caporaso, R. Knight *mSystems*, 2016, DOI: 10.1128/mSystems.00003-15

Section 2.2.3 has been adapted from the original publication in "Deblur rapidly resolves single-nucleotide community sequence patterns". A. Amir, D. McDonald, J. A. Navas-Molina, E. Kopylova, J. T. Morton, Z. Z. Xu, E. P. Kightley, L. R. Thompson, E. R. Hyde, A. Gonzalez, R. Knight *mSystems*, 2017, DOI: 10.1128/mSystems.00191-16

2.2.1 Subsampled open-reference clustering creates consistent, comprehensive OTU definitions and scales to billions of sequences

Section 2.1 described three different OTU picking approaches: closed-reference, *de-novo* and open-reference. The open-reference approach was the recommended approach because it offers benefits over the other two approaches. The open-reference approach run time is shorter than the *de-novo* approach because it includes a parallel closed-reference step. Additionally, the open-reference approach doesn't discard any sequences from the input dataset because it contains a *de-novo* step. However, if the microbial organisms present in an environment have not been

previously characterized and included in the reference database, many sequences will fail to cluster during the closed-reference step, generating long running times on the *de-novo* step. To further reduce the running time of the open-reference approach, we presented a new approach: the subsampled open-reference approach [170].

The following text has been adapted from the original publication in *PeerJ, 2014*. As a contributor to this manuscript, I was involved in the design of the subsampled open-reference pipeline, contributed to the source code, performed some of its evaluations, wrote sections of the manuscript and reviewed drafts of the manuscript.

A detailed description of the workflow is illustrated in Figure 2.22. It is implemented using UCLUST v1.2.22q [43] for clustering in QIIME-1.6.0 [20] and later, though any sequence clustering software that provides support for *de-novo* and closed-reference clustering could be substituted for UCLUST. The inputs provided to this method are demultiplexed, quality-filtered sequences, and a reference sequence collection (for example, the Greengenes 13 8 97% OTU representative sequences [37, 131]). First, sequences are clustered in parallel using a closed-reference OTU picking workflow, where sequences are queried against the reference database at percent identity s (default 97%). If a read matches a reference sequence at greater than or equal to s% identity, it is assigned to the OTU defined by that reference sequence. These are referred to as the reference OTUs. Next, a random subsample of n% (n should be small, the default value in QIIME 1.8.0-dev and

earlier is 0.1%) of the sequences that failed to match the reference sequence collection are clustered *de-novo*, and the cluster centroids for all resulting OTUs are used to define a new reference sequence collection. Those OTUs are referred to as the new reference OTUs. The sequences that were not included in the random subsample that was clustered *de-novo* then go through an additional round of parallel closed-reference OTU picking, this time where they are clustered against the new reference OTUs based on matching a sequence in the new reference sequence collection at greater than or equal to s% identity. This creation of a new reference database allows us to harness the parallelization of our closed-reference OTU picking pipeline, greatly decreasing the time it takes for sequences that fail to hit the initial reference database to be clustered into OTUs. In the final clustering step, sequences that fail to hit a reference sequence during this final closed-reference OTU picking step are clustered *de-novo*. These are referred to as the clean-up OTUs. Finally, the reference OTUs, new reference OTUs, and clean-up OTUs are combined into a single OTU table (i.e., table of counts of OTUs on a per-sample basis, as described in [130]), and this table, as well as a filtered table excluding OTUs with counts less than or equal to a user-defined threshold c, are provided to the user. By default, $c = 2$, so each OTU is observed at least twice (i.e., singleton OTUs are excluded). Because many more of the sequences can be clustered using closed-reference OTU picking in this workflow, it can run in far less time than classic open-reference OTU picking.

We validated the subsampled open-reference OTU picking workflow by com-

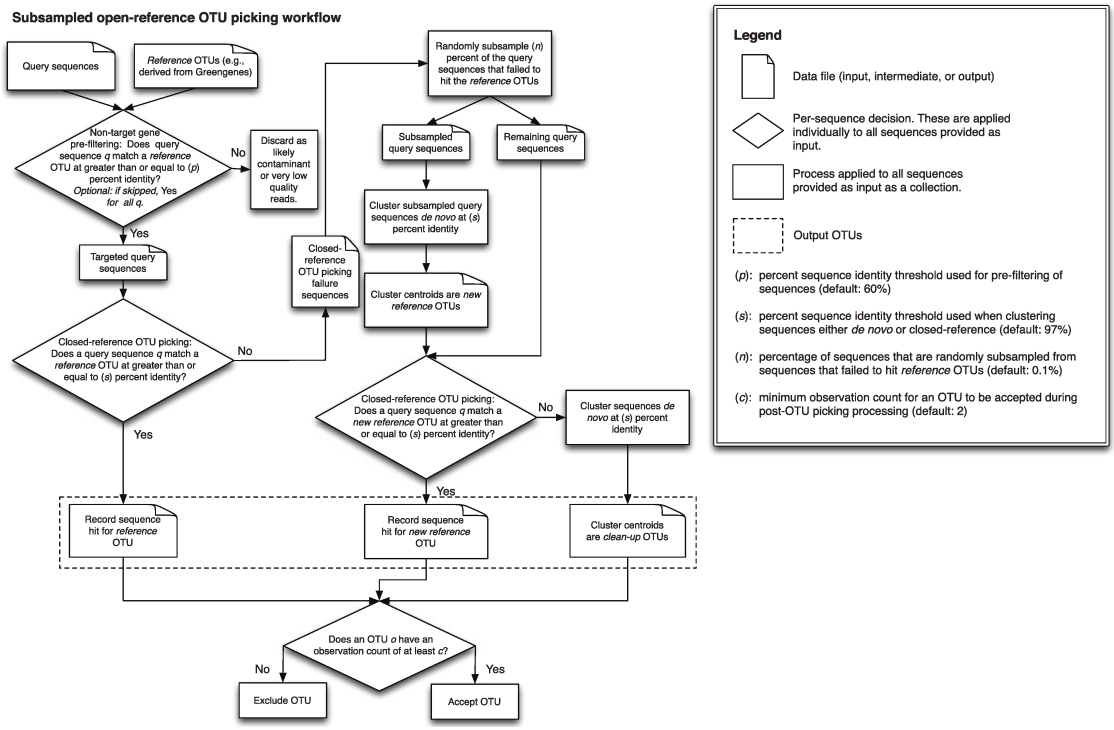


Figure 2.22: **Schematic of the subsampled open-reference OTU picking algorithm.**

paring it to the classic (i.e. non subsampled) open-reference clustering methods on three different datasets: the Lauber "88 Soils" study [103] (referred to as *88-soils* here), the Caporaso "Moving Pictures" study [21] (referred to as *moving-pictures* here), and the Costello "Whole Body" study [30] (referred to as *whole-body* here) using three metrics. First, we tested the correlation between sample alpha diversities (OTU counts, i.e. QIIME's *observed species* metric and Phylogenetic Diversity (PD) [48]) based on subsampled open-reference OTU picking and the classic open-reference clustering. Next, we tested whether beta diversity patterns (as determined by weighted and unweighted UniFrac [118] distances between samples) were consistent across OTU picking protocols, based on Mantel tests [127] with 1,000 Monte Carlo iterations. Finally, we tested whether the same taxonomic profiles were obtained on a per-sample basis using each of the OTU picking methods. It is important to note that we are not trying to assess whether one method is better than another using these metrics. Instead we are testing whether the methods give highly correlated results.

Alpha diversity (whole-body PD Pearson $r = 0.989$; 88-soils PD Pearson $r = 0.930$; moving-pictures PD Pearson $r = 0.996$), beta diversity (whole-body unweighted UniFrac Mantel $r = 0.948$; 88-soils unweighted UniFrac Mantel $r = 0.939$; moving-pictures unweighted UniFrac Mantel $r = 0.991$) and taxonomic summaries (whole-body: $r = 0.999$ at phylum level, 0.999 at species level; 88-soils $r = 0.999$ at phylum level, $r = 0.999$ at species level; moving-pictures $r = 0.999$ at phylum level, $r = 0.999$ at species level) were highly correlated between

classic and subsampled open-reference OTU picking. Minor differences likely arise from the non-deterministic step of rarefying all samples to even sampling depth before comparing samples. These results suggest that subsampled open-reference picking yields the same results as classic open-reference OTU picking, including identical numbers of sequences failing to hit the reference database, and therefore is a suitable replacement.

2.2.2 Open-source sequence clustering methods improve the State of the Art

Section 2.2.1 described a faster approach to perform open-reference OTU picking. The OTUs quality and the final running time are dependant on the actual underlying tool being used to perform the OTU picking. In the previous section, UCLUST v1.2.22q [43] was used, which was developed in 2010, and had become the default option for researchers performing micorbial community analysis. Since then, new tools have been published in the literature and a comprehensive benchmark of those tools was needed to evaluate if a new, faster, more accurate tool was available.

The following material has been adapted from the original publication in *mSystems, 2016*. As a contributor to this manuscript, I was involved in the integration of the new tools in QIIME, contributed to the experimental design, provided input about the compatible OTU definitions, performed some of the benchmarks,

wrote sections of the manuscript and reviewed drafts of the manuscript.

Between 2012 and 2015, four new sequence-clustering tools have emerged: OTUCLUST from the Micca package [2], Swarm [124, 125], SUMACLUST (C. Mercier, F. Boyer, E. Kopylova, P. Taberlet, A. Bonin, and E. Coissac, submitted for publication), and SortMeRNA [92]. These tools include open-source implementation, and the latter three implement multilevel parallelization, providing excellent potential alternatives to UCLUST [43]. In this study, we evaluated these new open-source tools and compared them against UCLUST and USEARCH, two commonly used options available in QIIME, UPARSE [44], the latest USEARCH amplicon analysis pipeline, and the three hierarchical clustering algorithms available in mothur [177].

A variety of datasets were chosen to evaluate the performance of these open-source OTU clustering approaches relative to QIIMEs UCLUST/USEARCH-based OTU clustering approaches as well as UPARSE. Two 16S rRNA gene simulated datasets were generated as FASTQ files. The first one (*sim_even*) represents an even distribution of 1,076 species, randomly subsampled from the Greengenes 97% [37, 131] database and computationally amplified at the same depth (100 reads/amplicon) and length (150 base pairs (bp)) using PrimerProspector [211] for extracting the V4 region and the ART [79] simulator for amplification and sequencing simulation. The second data set (*sim_staggered*) represents the same 1,076 species as the *sim_even* data set but amplified at different (random) species abundance levels. We used four different previously published mock community data sets:

three 16S rRNA gene mock community data sets (*Bokulich_2*, *Bokulich_3*, and *Bokulich_6*) from Bokulich et al. [14] and an 18S gene (*mock_nematodes*) data set from Porazinska et al. [155]. Finally, we also used three previously published natural data sets: a 16S rRNA gene soil data set (*canadian_soil*) from Neufeld et al. [147], a 16S rRNA gene human data set (*body_sites*) from Costello et al. [30], and an 18S rRNA gene soil data set (*global_soil*) from Ramirez et al. [165].

Performance was evaluated using a variety of metrics, including the accuracy of OTU and taxonomic assignments, alpha diversity (within-sample diversity), beta diversity (between-sample diversity), and taxonomic correlation. All tools showed increased precision after the removal of singleton OTUs (OTUs consisting of only one sequence), so all results presented here have had singleton OTUs removed. Table 2.4 summarizes basic performance results for all software.

We found that Swarm, SUMACLUST, UCLUST, and UPARSE (with relaxed parameters) performed equally well on simulated datasets where the ground truth was well established, with *mothur_average* and OTUCLUST closely behind. Despite this controlled chimera-free environment, UPARSE with recommended parameters reported the lowest accuracy for the *sim_staggered* data set, implying that stringent quality filtering can cause a significant underestimation of species abundance and diversity and lead to incorrect biological results. For the mock communities, most tools were able to correctly detect the expected number and identity of genera, but only UPARSE reported significantly fewer false-positive taxa (followed by OTUCLUST and USEARCH). For UPARSE, this was expected,

Table 2.4: **Benchmark summary.** OTU counts do not include singletons. F measure (F1) is for assigned taxonomies at the genus level. The PD whole-tree column for *Bokulich_2* and *Bokulick_3* represent PD intervals across various sampling depths. Procrustes M^2 (the sum of squared deviations or the dissimilarity of two datasets for UniFrac PCoA) and rho (Pearson's correlation coefficient for taxonomies at genus level) values are with repect to UCLUST (default for QIIME versions 1.0.0 to 1.9.1). Monte Carlo P values were not included, since all values were < 0.05 except for *de novo* usearch52 versus uclust ($P = 0.09$). The darkest blue shades represent the highest F1 scores, while the darkest red shades represent results closest to those obtained with UCLUST

			Data set																						
			Simulated						Mock										Genuine						
			sim_even (V4)			sim_staggered (V4)			Bokulich_2 (V4)			Bokulich_3 (V4)			Bokulich_6 (V4)			body_sites (V2)			canadian_soil (V4)			global_soil (V9, 18S)		
	Software		OTUs	PD	F_1	OTUs	PD	F_1	OTUs	PD	F_1	OTUs	PD	F_1	OTUs	PD	F_1	OTUs	M^2	ρ	OTUs	M^2	ρ	OTUs	M^2	ρ
de_novo	swarm		1,042	101.50	0.84	1,035	104.00	0.83	7,084	[4-50]	0.48	6,349	[4-35]	0.50	1,223	39.41	0.54	14,184	0.19	0.96	59,688	0.16	0.94	80,321	0.87	0.98
	sumaclust		1,031	104.06	0.83	1,022	109.92	0.83	9,575	[4-157]	0.38	13,982	[4-190]	0.41	3,317	90.80	0.52	7,103	0.18	0.99	74,284	0.14	0.87	60,781	0.50	0.96
	uparse_q3		1,013	104.02	0.84	997	110.57	0.84							199	9.22	0.59	156	0.38	0.29	11,259	0.03	0.85			
	uparse_q16		972	100.74	0.84	806	93.28	0.78				57	[2-3]	0.79	31	3.53	0.45	108	0.36	0.26	6,275	0.06	0.75			
	uclust		1,045	105.37	0.83	1,035	110.42	0.83	20,084	[5-234]	0.40	21,929	[5-236]	0.40	4,397	105.37	0.52	11,204	0.00	1.00	91,143	0.00	1.00	82,642	0.00	1.00
	usearch52		1,035	106.09	0.83	1,015	110.76	0.81	1,522	[3-22]	0.50	2,602	[4-28]	0.55	798	22.86	0.55	3,903	0.17	0.94	47,679	0.05	0.94	41,668	0.93	0.98
	usearch61		1,049	104.85	0.84	1,034	110.68	0.83	22,987	[7-313]	0.39	24,704	[7-292]	0.41	4,635	123.04	0.51	14,483	0.18	0.99	102,435	0.06	0.99	102,211	0.48	0.98
	mothur_near		957	110.09	0.82	949	110.45	0.81				1,600	[2-51]	0.44	447	23.63	0.54	806	0.45	0.12	31,546	0.06	0.76	11,440	0.53	0.76
	mothur_fur		978	109.22	0.82	970	109.86	0.81				28,808	[5-263]	0.40	5,159	75.05	0.51	3,358	0.22	0.23	92,887	0.03	0.86	32,378	0.56	0.78
	mothur_avg		963	109.99	0.82	959	110.98	0.82				13,255	[4-176]	0.41	2,314	55.90	0.51	2,491	0.26	0.11	83,664	0.05	0.86	20,809	0.49	0.72
closed_ref	usearch61	F_1 *tax*	1,275	129.19	0.83	1,267	127.50	0.82	1,027	[5-26]	0.53	614	[4-18]	0.59	631	26.02	0.61	5,982	0.06	0.96	13,808	0.06	0.96	3,784	0.50	0.55
		F_1 *OTUs*			0.68			0.69																		
	uclust	F_1 *tax*	1,238	127.59	0.83	1,225	126.02	0.84	1,053	[5-27]	0.53	557	[5-18]	0.57	547	25.03	0.60	5,446	0.00	1.00	13,659	0.00	1.00	305	0.00	1.00
		F_1 *OTUs*			0.69			0.70																		
	sortmerna	F_1 *tax*	1,072	122.75	0.82	1,067	121.89	0.81	396	[4-15]	0.53	290	[4-13]	0.61	382	19.47	0.57	6,174	0.06	0.99	13,281	0.06	0.98	255	0.34	0.75
		F_1 *OTUs*			0.80			0.80																		
	usearch52	F_1 *tax*	1,001	115.38	0.80	980	113.39	0.78	571	[5-30]	0.54	331	[5-22]	0.64	315	18.24	0.59	3,355	0.08	0.97	4,121	0.04	0.79	5,763	0.48	0.19
		F_1 *OTUs*			0.70			0.68																		
open_ref	uclust		1,262	106.12	0.83	1,245	111.29	0.83	10,169	[3-97]	0.40	4,170	[3-104]	0.42	4,109	93.67	0.48	12,442	0.00	1.00	87,936	0.00	1.00	37,380	0.00	1.00
	sortmerna_ sumaclust		1,072	104.77	0.82	1,085	111.80	0.81	9,272	[3-132]	0.39	2,649	[3-140]	0.41	2,727	88.56	0.51	10,242	0.06	0.98	79,363	0.03	0.82	35,345	0.12	0.92
	usearch61		1,304	106.04	0.83	1,293	112.36	0.83	9,414	[3-108]	0.40	3,966	[3-126]	0.41	3,421	80.89	0.53	12,807	0.06	0.97	87,300	0.06	0.80	43,175	0.10	0.94

as a large proportion of reads was filtered out prior to clustering, leaving evidence of only the most abundant taxa (OTUs comprised of hundreds of thousands of reads). The majority of false-positive taxa reported by other tools were low-abundance OTUs that could be mapped to Basic Local Alignment Search Tool (BLAST)s NT database with very high similarity (E value, $< 1e50$). If the users primary goal is to focus on the most abundant microbial profiles, low-abundance OTUs may be filtered out postclustering, but care should be taken, because such low-abundance OTUs can be important members of communities [179].

Although most open-source tools report an increased run time in comparison to UCLUST and USEARCH 2.23, they provide the benefit of finding significantly fewer OTUs. In the case of SortMeRNA, longer reads (150 bp) are quicker to align than the same number of shorter reads (100 bp) due to many fewer high-scoring candidate reference sequences to analyze. Moreover, all of these tools support multilevel multithreading and can easily scale to modern big-data processing demands. An alternative to reducing run time is to filter out a substantial number of reads, as done by UPARSE; unfortunately, the filtering parameters are sensitive to different data, and choosing them manually by trial and error can be a time-consuming task with unpredictable outcomes in diversity.

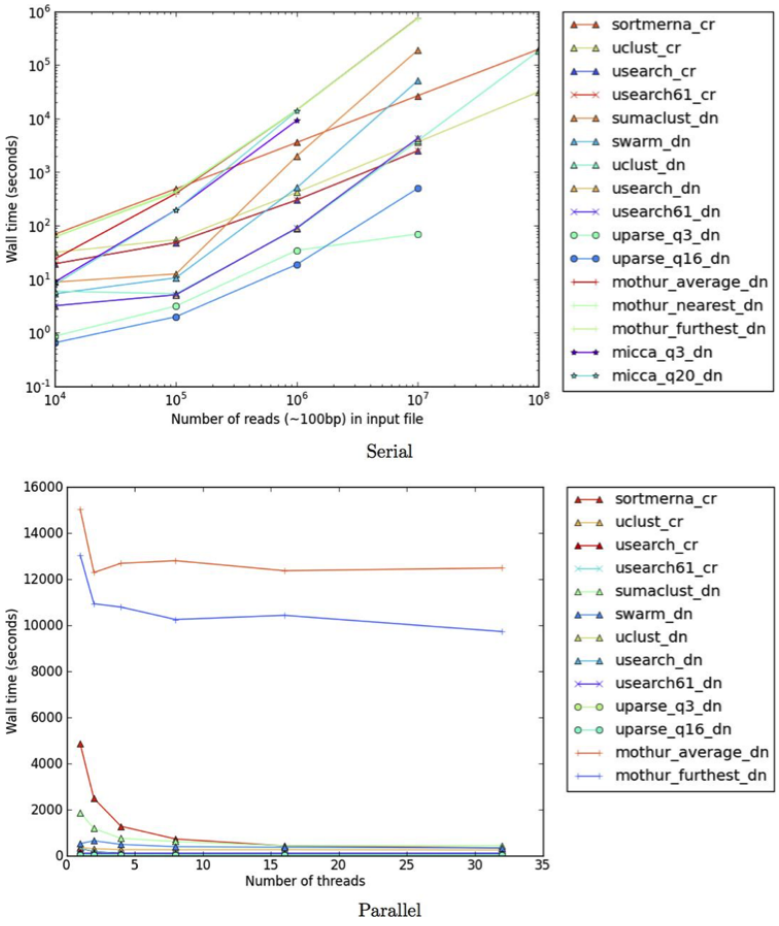


Figure 2.23: **Runtime performance of all benchmarked software.** All tests were performed using 1 to 32 coers on Intel Xeon CPU E5-2640 v3 at 2.60 GHz. Input files contained reads subsampled from the Global Gut [224]. For serial performance, some tools do not show results for 10^8 reads due to exceeding wall time limit (230 hours) or failed memory allocation. For parallel performance, a single file containing 1 million Illumina sequences was used over multiple threads

2.2.3 Deblur rapidly resolves single-nucleotide community sequence patterns

Sections 2.2.1 and 2.2.2 were focused on improving the performance of the OTU picking process for analyzing microbial community datasets. OTUs are defined to approximate the species richness of a sample, and reduce the effects of the sequencing error from Next Generation Sequencing (NGS) technologies. However, OTUs are based on an arbitrary sequence identity threshold (typically 97%), which reduces the phylogenetic resolution, because two sequences that are more similar than the identity threshold can't be differentiated. To assess this problem, we presented a new method, Deblur, that instead of grouping sequences based on an arbitrary sequence identy threshold, uses statistical methods to find the underlying true sequence and remove erroneus sequences [6].

The following material has been adapted from the original publication in *mSystems, 2016*. As a contributor to this manuscript, I was involved in the discussions of the Deblur pipeline, contributed to the source code, generated figures for the manuscript and reviewed drafts of the manuscript.

Similar in concept to AmpliconNoise [161], a denoising method for pyrosequencing, Deblur, like DADA2 [18] and UNOISE2 [45], attempts to obtain single-nucleotide resolution from Illumina data with statistical methods to infer the putative true sequences within a sample that give rise to the distribution of observed error-prone sequences. Unlike DADA2 and UNOISE2, Deblur operates on

each sample independently. It compares sequence-to-sequence Hamming distances within a sample to an upper-bound error profile combined with a greedy algorithm to obtain single-nucleotide resolution. The Deblur algorithm is implemented as follows (see Figure 2.24). First, sequences are sorted by abundance. Second, from the most to least abundant sequence, the number of predicted error-derived reads is subtracted from neighboring reads based on their Hamming distance, using an upper bound on the error probability. A parameterized maximal probability for indels (defaulting to 0.01) and a parameterized mean read error rate for normalization (defaulting to 0.5%) are included. Finally, any sequence whose abundance drops to 0 after a subtraction is removed from the list of valid sequences. Sequences not considered to be valid (i.e., noise) are removed. After applying Deblur, only reads likely to have been presented to the sequencer are retained. However, it is possible that the reads would still contain chimeras originating from PCR. Reads are filtered for de novo chimeras using UCHIME [46] as implemented by VSEARCH [172] using modified parameters.

Stability (i.e., obtaining the same sOTU across different samples) is becoming critical as more study designs exploit existing samples from resources like the Earth Microbiome Project [58] or require integration of sequence data collected over time such as the American Gut Project [12]. We compared the levels of stability of Deblur and DADA2 using technical replicates from a data set consisting of 40 individuals, each with one fecal sample sequenced twice on two separate MiSeq

[12] `http://americangut.org`

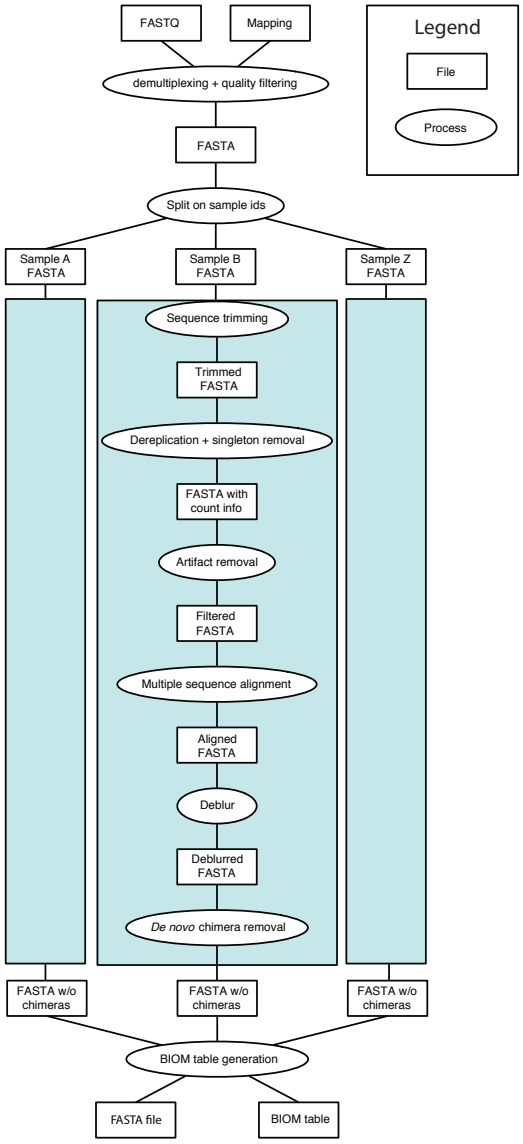


Figure 2.24: **The deblur pipeline.** A demultiplexed and quality filtered fasta/-fastq file (or a directory of per-sample fasta/fastq files) is used as the input to the pipeline. Following initial splitting to per-sample fasta files, all processing is done independently on each sample. Sequences are trimmed and dereplicated with singletons removed. Reads are then depleted from sequencing artifacts either using a set of known sequencing artifacts (such as PhiX) (negative filtering) or using a set of known 16S sequences (positive filtering). Resulting nonartifact reads are then aligned for easy indel detection. This multiple sequence alignment is then used as the input for the Deblur algorithm. Each Deblurred sample is then checked for de novo chimeras, and the resulting sOTUs from all samples are combined into a single BIOM [130] table (with sequences labeled as the sOTU IDs)

runs [77]. sOTUs for each run were assessed separately, and we compared the fractions of sOTUs from one run to those present in the second run, as a function of the minimal sOTU frequency. Deblur showed greater stability than DADA2 at a higher frequency cutoff (Figure 2.25-A), indicating that a larger fraction of sOTUs from the first run were also identified in the second run.

Next, we compared DADA2 and Deblur using a complex natural community and a previously published data set of fecal samples from two species of howler monkeys [4]. Deblur and DADA2 detected 1,938 and 1,636 sOTUs, respectively, after removal of sOTUs with fewer than 10 total reads from each method. Following filtering, about 70% of the sOTUs were identical between the methods. As expected, both methods identified differential sOTUs (permutation-based rank mean test; 0.1 false-discovery rateBenjamini-Hochberg method [FDR-BH] control value) with 61% of Deblur sOTUs differentiating between primate species (1,193/1,938), compared to 55% of DADA2 sOTUs (891/1,636). To assess whether the sOTUs unique to either method were from increased numbers of artifacts, we used BLAST [3] to compare each unique sequence against nt/nr and plotted the fraction of sOTUs with zero, one, or two mismatches. We observed that sOTUs unique to Deblur showed fewer mismatches than those unique to DADA2 (Figure 2.25-B). The distribution of sOTUs over the monkey samples suggests that the sOTUs unique to Deblur are more plausible because they show a pattern similar to those identified by both methods, whereas the sOTUs unique to DADA2 have markedly different patterns of clusters of unique sOTUs within single samples

(Figure 2.25-C).

Finally, to explore performance characteristics, we used a MiSeq run from the stability analysis in order to assess computational space and time demands of DADA2, Deblur, and UNOISE2 (where possible) over an increasing number of samples. UNOISE2 was an order of magnitude faster than Deblur, while Deblur was an order of magnitude faster than DADA2 (Figure 2.25-D).

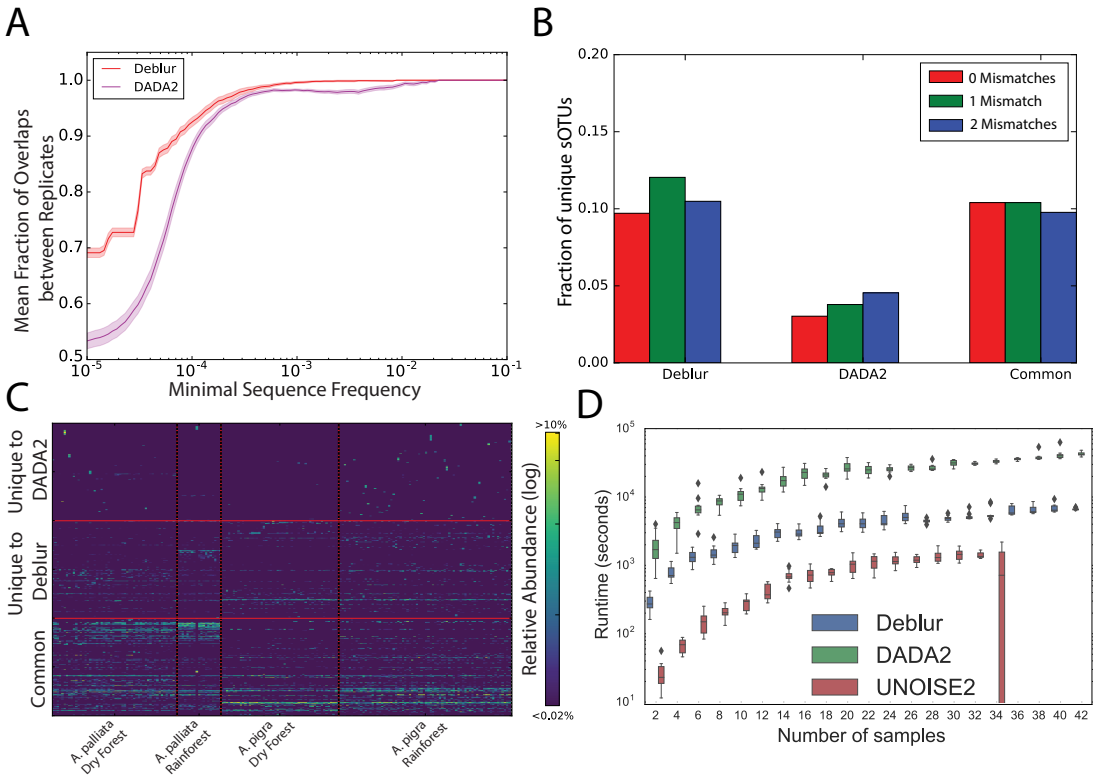


Figure 2.25: **Benchmarks of OTU picking tools on natural communities.** (A) Stability analysis on experimental technical repeats. Data indicate fractions of overlapping sOTUs from two technical replicates in all OTUs as a function of the minimal frequency threshold present in one of the repeats. (B and C) Application of Deblur in the howler monkey data set. (B) Fraction of sequences matching entries in the NCBI nr/nt database (as of 1 December 2016) with 0.1 or 2 mismatches (red, green, or blue, respectively) from sOTUs unique to Deblur or to DADA2 or present in both (left to right). (C) Heat maps showing sOTUs (rows) in common with Deblur and DADA2, as well as those unique to Deblur and DADA2 (bottom, middle, and top rows, respectively). Samples (columns) are sorted by species and habitat. A total of 200 sOTUs per group (i.e., common, unique to Deblur, or unique to DADA2) were randomly selected for visualization purposes. (D) Single-threaded runtime comparison of Deblur, DADA2, and UNOISE2 against one of the stability MiSeq runs at increasing numbers of samples.

2.3 Applying these tools to advance microbiome science

Sections 2.1 and 2.2 presented my work on standardizing and improving the pipelines to analyze microbial community data. Sections 2.3.1, 2.3.2, 2.3.3 and 2.3.4 provide examples of manuscripts that I have been involved in that have taken advantage of my work presented so far.

Section 2.3.1 has been adapted from the original publication in "The oral and skin microbiomes of captive Komodo dragons are significantly shared with their habitat". E.R. Hyde, J. A. Navas-Molina, S. J. Song, J. G. Kueneman, G. Ackermann, C. Cardona, G. Humphrey, D. Boyer, T. Weaver, J. R. Mendelson, V. J. McKenzie, J. A. Gilbert, R. Knight *mSystems*, 2016. DOI: 10.1128/mSystems.00046-16

Section 2.3.2 has been adapted from the original publication in "A communal catalogue reveals Earth's multiscale microbial diversity". *Nature* L. R. Thompson, J. G. Sanders, D. McDonald, A. Amir, J. Ladau, K. J. Locey, R. J. Prill, A. Tripathi, S. M. Gibbons, G. Ackermann, J. A. Navas-Molina, S. Janssen, E. Kopylova, Y. Vazquez-Baeza, A. Gonzalez, J. T. Morton, S. Mirarab, Z. Z. Xu, L. Jiang, M. F. Haroon, J. Kanbar, Q. Zhu, S. J. Song, T. Kosciolek, N. A. Bokulich, J. Lefler, C. J. Brislawn, G. Humphrey, S. M. Owens, J. Hampton-Marcell, D. Berg-Lyons, V. McKenzie, N. Fierer, J. A. Fuhrman, A. Clauset, R. L. Stevens, A. Shade, K. S. Pollard, K. D. Goodwin, J. K. Jansson, J. A. Gilbert, R. Knight,

The Earth Microbiome Project Consortium, 2017. DOI: 10.0.4.14/nature24621

Section 2.3.3, in part, has been submitted for publication of the material as it may appear in Science, 2018, D. McDonald, E. R. Hyde, J. W. Debelius, J. T. Morton, A. Gonzalez, G. Ackermann, A. A. Aksenov, B. Behsaz, C. Brennan, Y. Chen, L. DeRight Goldasich, P. C. Dorrestein, R. R. Dunn, A. K. Fahimipour, J. Gaffney, J. A Gilbert, G. Gogul, J. L. Green, P. Hugenholtz, G. Humphrey, C. Huttenhower, M. A. Jackson, S. Janssen, D. V. Jeste, L. Jiang, S. T. Kelley, D. Knights, T. Kosciolek, J. Ladau, J. Leach, C. Marotz, D. Meleshko, A. V. Melnik, J. L. Metcalf, H. Mohimani, E. Montassier, J. A. Navas-Molina, T. T. Nguyen, S. Peddada, P. Pevzner, K. S. Pollard, G. Rahnavard, A. Robbins-Pianka, N. Sangwan, J. Shorenstein, L. Smarr, S. J. Song, T. Spector, A. D. Swafford, V. G. Thackray, L. R. Thompson, Y. Vazquez-Baeza, A. Vrbanac, P. Wischmeyer, E. Wolfe, Q. Zhu, The American Gut Consortium, R. Knight.

Section 2.3.4 has been adapted from the original publication in "Correcting for microbial blooms in fecal samples during room-temperature shipping". *mSystems* A. Amir, D. McDonald, J. A. Navas-Molina, J. Debelius, J. T. Morton, E. R. Hyde, A. Robbins-Pianka, R. Knight 2017. DOI: 10.1128/mSystems.00199-16

2.3.1 The oral and skin microbiomes of captive Komodo dragons are significantly shared with their habitat

The following text has been adapted from the original publication in *mSystems, 16*. As a contributor to this manuscript, I performed the data analysis included in the manuscript, wrote the IPython notebok [153] attached to the publication, generated figures and reviewed drafts of the manuscript.

The evidence for both vertebrate animals and humans indicates that closed environments not only limit exposure to complex microbial diversity but also promote microbial transfer from the host to the environment, rather than from the environment to the host. Fully characterizing the effects of captivity on host-environment microbial sharing will be key for future studies of vertebrate microbial ecology and may prove instrumental in improving animal husbandry practices. To more thoroughly describe the effects of captivity on host-environment microbiome sharing and how this may affect vertebrate ecology studies, there is a need to examine the microbial ecology of the host-environment interaction in a number of vertebrate species, both in the wild and in captivity. Here we use as a model the captive Komodo dragon (*Varanus komodoensis*), applying 16S rRNA amplicon sequencing to characterize the oral, fecal, skin, and environment-associated microbiomes to answer two main questions: first, is the extent of host-environment microbiome sharing observed for captive Komodo dragons typical of that observed among other vertebrates living in closed environments, and second, is the host-

environment microbiome sharing observed among captive Komodo dragons characteristically different from that observed among wild vertebrates? To answer these questions, we explored whether host-environment microbiome sharing in captive Komodo dragons was similar to the pattern observed for humans and pets living in homes [106] and dissimilar to the pattern observed among wild amphibians living in open ecosystems [98]. Together with existing studies, the data suggest that living in closed environments is associated with extensive host-environment microbial sharing. This sharing is likely to be circular in naturethe host contributes microbes to its environment and then, in the absence of significant exposure to microbes from external sources, reacquires those microbes from its environment, only to share them with the environment once again (or vice versa). This may be a radical departure from the microbial communities and exposures which vertebrates cohabitate with and have evolved alongside in the wild, and could have significant effects on health and disease [105].

To determine how much of the Komodo dragon's microbiome is shared with its environment (or vice versa) and whether and how specific the environment is to the dragon, we obtained matched dragon-environment samples from the Denver and Honolulu zoos. In terms of taxonomic composition and abundance, environmental microbiomes appeared most similar to salivary and skin microbiomes from the phylum down to the genus level in both the Denver and Honolulu zoo cohorts (see Figure 2.26 A).

We applied SourceTracker [88] to samples from the Denver Zoo Komodo

dragons to determine which dragon microbiome sources (saliva, feces, and skin) contributed to the dragon environment. The microbiomes of items in the Denver Komodo dragons enclosures were largely sourced from Komodo dragon salivary, skin, and fecal samples (Figure 2.26 B), with unknown sources comprising less than 50% of the microbial communities of most environmental sample types. Additionally, skin, saliva, and fecal communities were distinct from one another in a SourceTracker independence test (Figure 2.26 B), suggesting that any skin, saliva, or fecal communities detected on environmental materials actually came from the dragons skin, mouth, or feces. Further supporting this pointat least in the context of salivaseveral bacterial taxa found in the mouths of the Komodo dragons studied here, including *Staphylococcus*, *Corynebacterium*, *Pseudomonas*, and *Bacteroides*, have previously been reported in the mouths of captive Komodo dragons [137, 60]. This suggests that environmental microbes designated as sourced from the Komodo dragons oral cavity likely actually do come from the mouth and not any other source. The nature and extent of host-microbiome transfer to environmental objects varied with sample type; for example, Komodo dragon saliva was the main source of the microbial communities detected in soil and on rock and glass, while Komodo dragon skin was the main source of the microbial communities detected on metal (Figure 2.26 B). Performing SourceTracker analyses with Komodo dragon samples designated as sinks and environmental samples designated as sources revealed that the microbial communities of Komodo dragon fecal, saliva, and skin samples are sourced from a variety of environmental materials, each contributing

30% or less of the microbial community (Figure 2.26 C and D). There is no one environmental material that contributes more than any other material to Komodo dragon feces or saliva; however, Komodo skin microbial communities are sourced majorly from glass and unknown sources (each 40%).

To further assess host-environment microbiome sharing, in both closed/-captive and open/wild environments, we additionally performed SourceTracker analyses on two previously published data setsa wild amphibian skin-environment microbiome data set [98] and a human-pet-house microbiome data set [106]and compared them to the Komodo dragon data set. As previously shown, humans and their pets contribute a large amount of their microbiomes to their living environments [106], similarly to the patterns we observed with captive Komodo dragons. However, while Komodo dragon microbiome sources (skin, saliva, and feces) were found to be distinct sources, we did not observe this level of source independence when applying the SourceTracker independence test to the human/pet data set.

Host-environment microbiome sharing between amphibians and their living environment was not as extensive as that observed among captive Komodo dragons and their enclosures or humans and pets and their homes. More than 75% of soil and sediment microbial communities were obtained from unknown microbiome sources; however, the identified source for 75% of water microbial communities was amphibian skin (Figure 2.27 A). Each source (here defined as individual amphibian species) was highly independent from each other source (Figure 2.27 B). Defining amphibian skin as a sink and environmental samples as sources, water

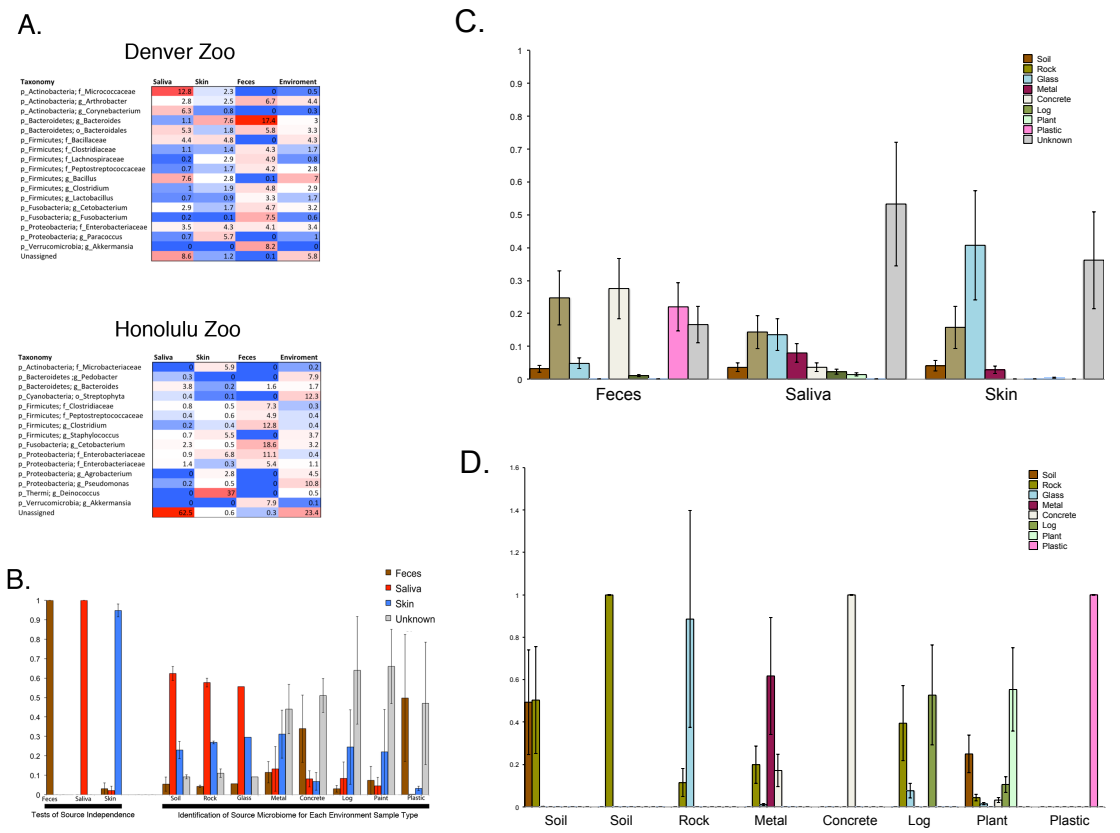


Figure 2.26: **Taxonomy and SourceTracker results for the Komodo dataset.** (A) Heat maps illustrate the percent abundances of the most abundant genera (all OTUs taxonomically classified to the same genus were collapsed into a single genus summary) present in saliva, skin, feces, and environmental (Env) samples collected from the Denver and Honolulu zoos. The deepest taxonomic classification achieved is listed for each genus. The heat map colors indicate percent abundance (red [high abundance] to blue [low abundance]). (B) Komodo dragon SourceTracker analysis reveals that the microbial communities of many environmental sample types are sourced from skin, saliva, and feces rather than unknown sources (i.e., not from Komodo dragon skin, saliva, or feces). Data are plotted as the means ± standard errors of the means (error bars) of samples from Denver and Honolulu zoo Komodo dragons. (C) SourceTracker analyses with Komodo dragon fecal, salivary, and skin samples designated as sinks and environmental samples designated as sources reveals that a variety of environmental sources, rather than a single environmental source, contribute to the microbial communities of Komodo dragon feces, saliva, and skin. Unknown sources (i.e., not the environments sampled from the Komodo dragon enclosures) also contribute about 40% or more of the microbial community of saliva and skin samples (only 20% of fecal samples). (D) Independence tests reveal that about half of the environmental samples are not independent from other environmental samples. Data are the means ± standard errors of the means of Denver and Honolulu Komodo dragon and environmental samples.

was identified as a major source of the microbes on the skin of most species; nevertheless, at least 20% of the microbial community on the skin of all species was contributed by unknown sources (Figure 2.27 C). Soil, sediment, and water were all confirmed to be independent sources (Figure 2.27 D).

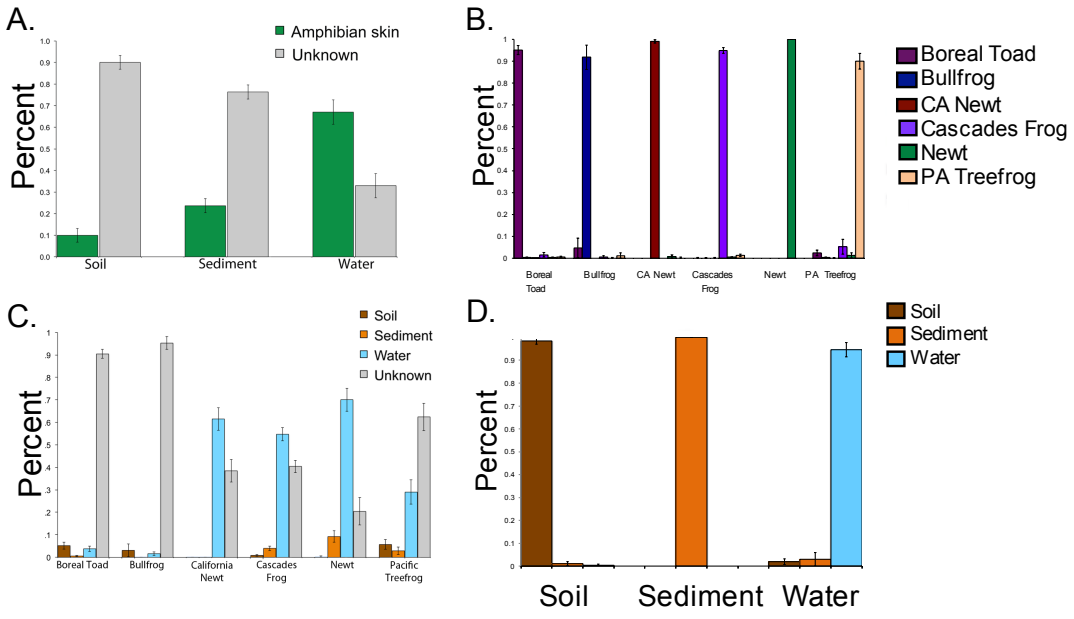


Figure 2.27: **SourceTracker results for the amphibians dataset.** (A) Amphibian SourceTracker analysis reveals that water is the only sample type that obtains a notable amount of its microbial community from amphibian skin; unknown sources (i.e., not amphibian skin) are the main microbiome contributors to soil and sediment. (B) Independence tests reveal that amphibian skin is independently specific to species. (C) Designating environment the source and amphibian skin the sink reveals that water is the only environmental type that contributes largely to the microbial communities on amphibian skin, with unknown sources also largely contributing to the amphibian skin microbiome. (D) Independence tests reveal that each environment type is also independent from each other environment type. Data are the means ± standard errors of the means.

2.3.2 A communal catalogue reveals Earth's multiscale microbial diversity

The following material has been adapted from the original publication in *Nature, 2017*. As a contributor to this manuscript, I quality filtered the 97 independent studies included in the manuscript, generated the OTU-based closed and open reference tables and the open reference phylogenetic tree and provided scripts to reproduce those steps.

The EMP was founded in 2010 to sample the Earth's microbial communities at an unprecedented scale in order to advance our understanding of the organizing biogeographic principles that govern microbial community structure [59, 58, 197]. The EMP asked the global scientific community for environmental samples and associated metadata spanning diverse environments and capturing spatial, temporal, and/or physicochemical covariation. The first 27,751 samples from 97 independent studies represent diverse environment types (Figure 2.28 A), geographies (Figure 2.28 B), and chemistries. The EMP encompasses studies of bacterial, archaeal, and eukaryotic microbial diversity. The analysis here focuses exclusively on the bacterial and archaeal components of the overall database (for concision, use of microbial will hereafter refer to bacteria and archaea only). Associated metadata included environment type, location information, host taxonomy (if relevant), and physicochemical measurements. Physicochemical measurements were made in situ at the time of sampling. Investigators were encouraged to measure temperature

and pH at minimum. Salinity, oxygen, and inorganic nutrients were measured when possible, and investigators collected additional metadata pertinent to their particular investigations.

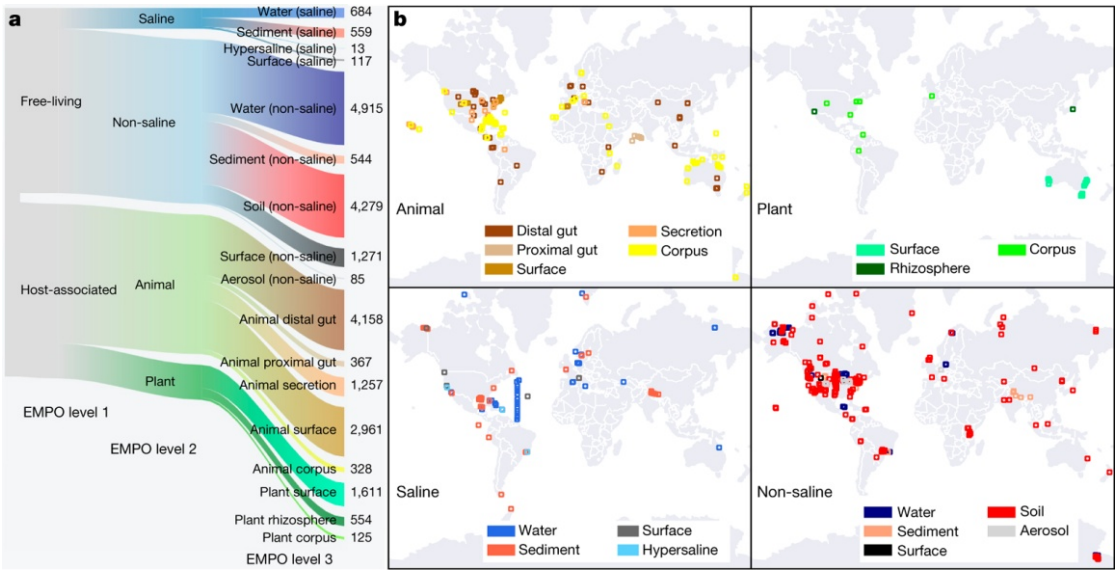


Figure 2.28: **Environment type and provenance of samples.** (A) The EMPO classifies microbial environments (level 3) as free-living or host-associated (level 1) and saline or non-saline (if free-living) or animal or plant (if host-associated) (level 2). The number out of 23,828 samples in the QC-filtered subset in each environment is provided. EMPO is described with examples at http://www.earthmicrobiome.org/protocols-and-standards/empo. (B) Global scope of sample provenance: samples come from 7 continents, 43 countries, 21 biomes (ENVO), 92 environmental features (ENVO), and 17 environments (EMPO).

Beyond measured physical covariates, the breadth of environments in the EMP catalogue allows a detailed exploration of how microbial diversity is distributed across environments. Diversity among communities (beta-diversity) is driven by turnover (replacement of taxa) and nestedness (gain or loss of taxa resulting in differences in richness) [24]. If turnover dominates, then disparate communities will harbour unique taxa. If nestedness dominates, then communities

with fewer taxa will be subsets of communities with more taxa. We tested for nestedness using a 2,000-sample subset with even representation across environments and studies. Given the contrasting environments and geographic separation among the many studies in the EMP, we expected different environments to contain unique sets of taxa and to show little nestedness. However, we found that communities across environments were significantly nested (Figure 2.29 A, B; P¡0.05) in comparison to null models (Figure 2.29 C), accounting for the observed patterns of richness. At coarse taxonomic levels, an average of 84% of phyla, 73% of classes, and 58% of orders that occurred in less diverse samples also occurred in more diverse samples. These patterns could have resulted from several mechanisms, including ordered extinctions [190] and the filtering of complex communities over time [11], differential dispersal abilities [116] and cascading sourcesink colonization processes that assemble nested subsets from more complex communities, or by the tendency of larger habitat patches to support more rare taxa with lower prevalence [55]. Notably, finer taxonomic groupings showed less nestedness (Figure 2.29 C), indicating that the processes that underlie nested patterns of turnover are likely to reflect conserved aspects of microbial biology, and not to result from the interplay of diversification and dispersal on short timescales.

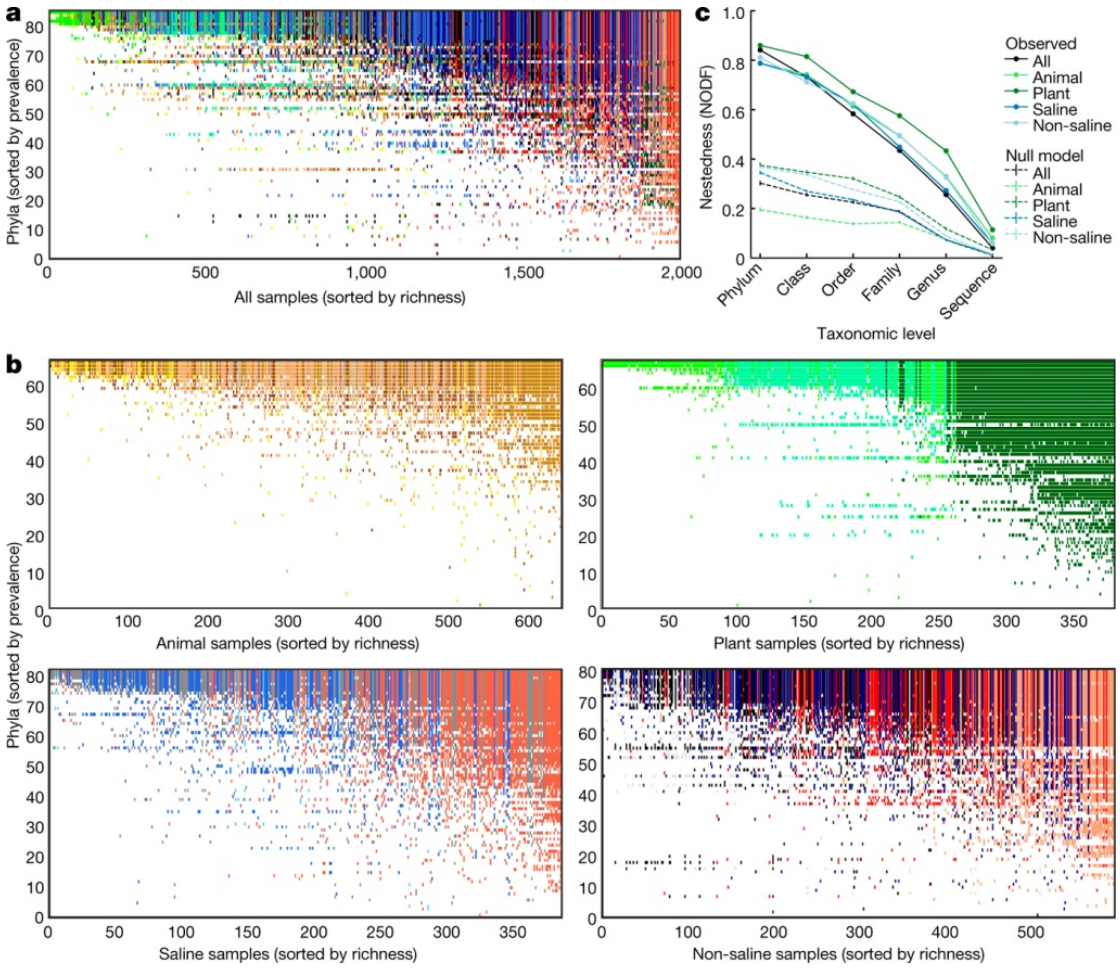


Figure 2.29: **Nestedness of community composition.** (A) Presenceabsence of phyla across samples, with phyla (rows) sorted by prevalence and samples (columns) sorted by richness. Shown is a subset of the EMP consisting of n=2,000 biologically independent samples with even representation across environments and studies. With increasing sample richness (left to right), phyla tended to be gained but not lost (P¡0.0001 versus null model; NODF (nestedness measure based on overlap and decreasing fills) statistic and 95% confidence interval=0.841±0.018). (B) As in A but separated into non-saline, saline, animal, and plant environments (P¡0.0001, respective NODF=0.811 ±0.013, 0.787±0.015, 0.788±0.018 and 0.860±0.021). (C) Nestedness as a function of taxonomic level, from phylum to tag sequence, across all samples and within environment types. Also shown are median null model NODF scores (± s.d.). NODF measures the average fraction of taxa from less diverse communities that occur in more diverse communities. All environments at all taxonomic levels were more nested than expected randomly, with nestedness higher at higher taxonomic levels (for example, phyla).

2.3.3 American Gut: An open platform for citizen-science microbiome research

The following material has been adapted from the original publication submitted in *Science, 2018*. As a contributor to this manuscript, I generated the per participant results, provided support maintaining the software and reviewed drafts of the manuscript.

We therefore launched the American Gut Project (AGP) [13], now the largest crowdfunded and crowdsourced microbiome citizen-science project to date, with the goal to discover the kinds of microbes and microbiomes "in the wild. Our project informs participants about their own microbiomes by providing them with a standard report that places them in context of the full AGP and Human Microbiome Project (HMP) datasets, and provides a broad set of resources to support research about the human microbiome, including an online course. Unlike many other large microbiome studies, the AGP deposits all de-identified data into the public domain on an ongoing basis without access restrictions. This reference database has allowed us to characterize the diversity of the industrialized human gut microbiome at an unprecedented scale, to explore novel relationships with health, lifestyle, and dietary factors, and to establish the AGP resource and infrastructure as a living platform for discovery (e.g., through targeted sub-populations and through the application of multi-omics techniques).

[13] `http://americangut.org`

Key variables that we found to have the greatest effects on the composition of gut microbes - plant consumption, antibiotic use, and even age - are in flux in the global population. Our lifespans are increasing, we are traveling more, our diets are becoming homogenized, and we are consuming more antibiotics. In each case, these trends are likely to favor more homogenization and less diverse gut microbes. Ongoing efforts, such as the AGP, will allow researchers to document and potentially mitigate the effects of such change. They will also afford insights into our past through collections from more diverse subpopulations, which will allow us to better understand the context of our choices in the future.

A unique aspect of the AGP is the open community process of assembling the Research Network and analyzing these data. Because participants fund the project, no funding agency mandates restrictions on data analysis to a specific group of investigators. Thus, these data are released into the International Nucleotide Sequence Database Collaboration (INSDC) (and GNPS for metabolomics data) as soon as initial quality control and anonymization steps have been applied. Analysis details are shared through a public forum [14]. Scientific contributions to the project were made through a geographically diverse Research Network represented herein as the American Gut Consortium (including explicitly named authors). This network was established prior to project launch and has continued to grow over time. The analyses described were performed through an open contribution model in which pre-computed forms of these data were publicly pro-

[14]GitHub, `https://github.com/knightlab-analyses/american-gut-analyses`

vided with encouragement to the American Gut Consortium to explore the dataset. This model allows the project to use a living analysis approach, embracing new researchers and analytical tools on an ongoing basis (e.g., Qiita and Global Natural Products Social Molecular Networking (GNPS)). Additionally, because the AGP is a subproject of the Earth Microbiome Project (EMP) [197]), all samples were processed using the publicly available and widely used EMP 16S rRNA gene amplification, sequencing, and data analysis protocols to facilitate meta-analyses. For example, we combined the AGP with fecal samples collected from a fecal transplant study and an infant microbiome time series, the latter using different DNA sequencing technology, to highlight how this context can provide insight.

2.3.4 Correcting for microbial blooms in fecal samples during room-temperature shipping

The following material has been adapted from the original publication in *mSystems, 2016*. As a contributor to this manuscript, I was involved in the discussion for establishing the criteria to choose blooming bacteria, provided input on the figures design and reviewed drafts of the manuscript.

The use of sterile swabs is a convenient way to collect samples for microbiome studies, but in some cases, it is not feasible to immediately freeze or utilize a preservative. For example, the AGP allows members of the general public to send samples for 16S rRNA gene amplicon sequencing through domestic post without

a preservative. This is because proven preservation methods can be cumbersome, dangerous, expensive, or sample type specific, complicating participation in microbiome citizen science. Although some studies have demonstrated that the effects of room-temperature storage are secondary to physiologically relevant differences between comparison groups [189, 183, 104], certain bacterial taxa, particularly those in the class *Gammaproteobacteria*, grow well at room temperature. This is problematic, as some *Gammaproteobacteria* species have been associated with disease, such as Inflamatory Bowel Disease (IBD) [57]. Therefore, to identify meaningful patterns in microbiome studies that do not utilize sample preservation, it is crucial to remove at high specificity the taxa that thrive at room temperature (i.e., blooming bacteria).

Without filtering candidate blooms, there were notable differences (as observed using Bray-Curtis PCoA) between AGP fecal samples and the fresh-frozen fecal samples; filtering the bloom sequences from all samples removed these differences (Figure 2.30 A versus B). In the PCoA space corresponding to the data determined without filtering, the primary separation is explained by the presence of a large percentage of bloom sequences (Figure 2.30 A); the sizes of the spheres are scaled by the percentage of bloom sequences in the respective sample. Following the removal of the blooms, this dominant effect was abolished and samples with high levels of blooms clustered with samples from the other studies (Figure 2.30 B). Similar results were observed in assessing class-level taxonomy abundances (Figure 2.30 C versus D): prior to filtering, a high relative abundance

of *Gammaproteobacteria* (27%) was present in the AGP samples compared to the fresh-frozen samples (1.5% to 3.5%), while the AGP profile seen after filtering more closely resembled that of the fresh-frozen samples. Importantly, applying the filter minimally changed the taxonomic profiles of fresh-frozen samples (Figure 2.30 D). The filtering procedure is available in a Jupyter Notebook [153] at https://github.com/knightlab-analyses/bloom-analyses.

There is a balance between type 1 and type 2 errors that must be considered in applying this filter. The cost of removing a sequence is that it becomes invisible in the analysis, and it is possible that real sequences are lost. Conversely, retaining a bloom sequence increases noise caused by shipment conditions, which can artificially alter biological conclusions. Therefore, a balance between loss of data and inaccurate, noisy data must be obtained. To select an appropriate number of blooming bacterial sequences to subtract from the AGP data set to maximize the amount of data retained while reducing inaccuracies caused by blooms, we tested the effect of nested filtering levels on the ability to detect the well-known effect of age on alpha diversity [224, 90]. As can be seen in Figure 2.30 E and F, this effect was undetected by a Kruskal-Wallis test when none of the candidate blooms were removed. However, filtering the top four candidate blooms restored the ability to detect a significant difference in diversity by age. Critically, the identification of the bloom sOTUs was done independently of this positive control. For analysis of the AGP cohort, we recommend removal of the sequences of the top 10 candidate blooming bacterial taxa, as this maximizes the expected age effect (Figure 2.30

E). Different studies may need to remove a different subset of bloom sequences, as retaining some of these sequences might be critical, depending on the study characteristics. With meta-analysis, if this filter is applied, it must be applied identically to all samples represented to avoid introduction of a systematic bias.

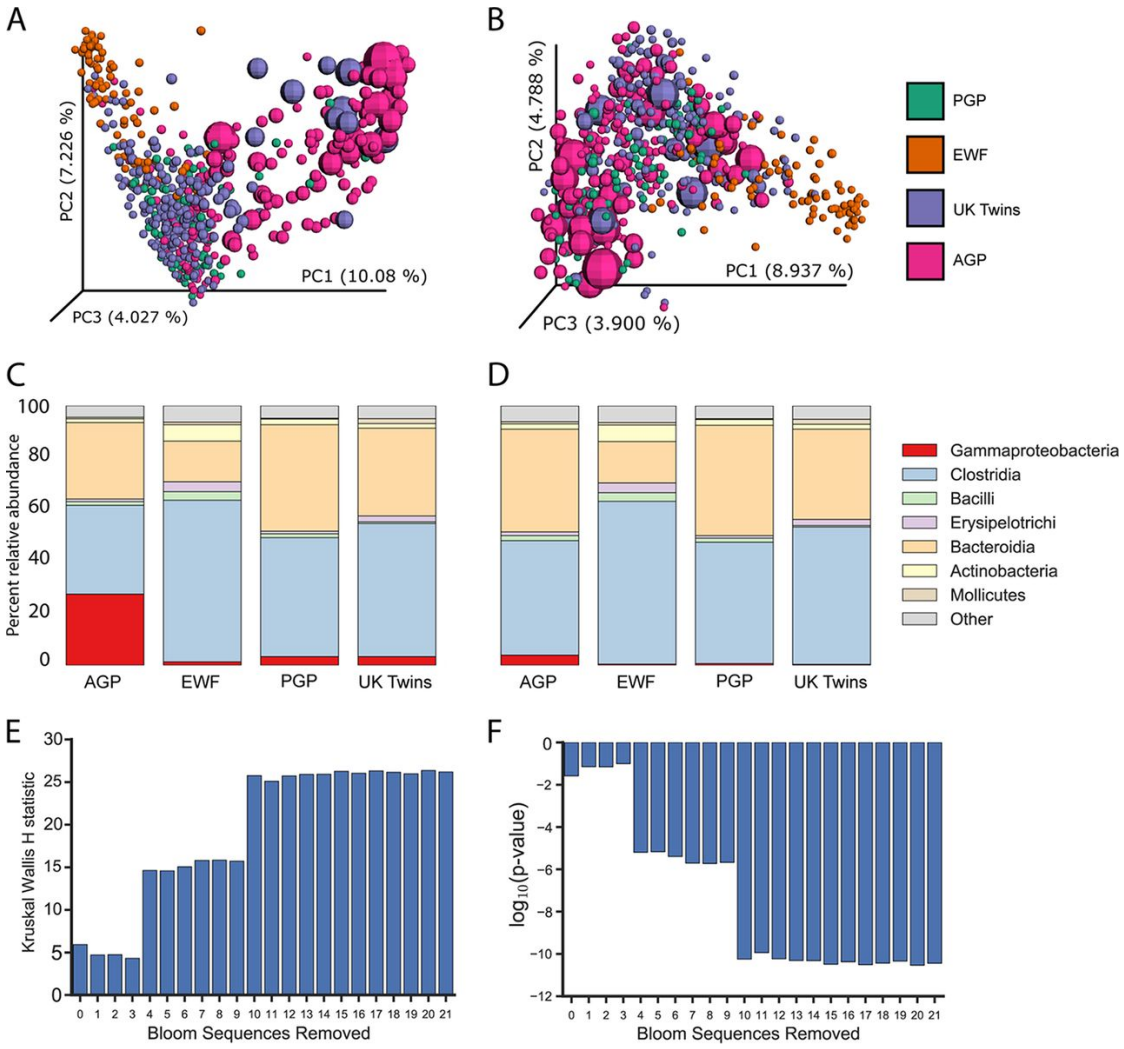


Figure 2.30: **Effect of bloom filtering on American Gut data.** (A and B) PCoA of Bray-Curtis distances from a random subset of 200 American Gut Project samples (colored pink) compared to 3 studies containing fresh-frozen fecal samples: Personal Genome Project (colored green); whole-grain feces [209] (colored orange); and UK Twins [63] (colored purple), respectively. The PCoA data shown represent results obtained before (A) and after (B) applying the filter for blooms to all samples. The size of a sphere is scaled by the amount of candidate bloom bacteria in a sample prior to filtering. (C and D) Mean taxonomy distribution for the same studies before (C) and after (D) filtering for blooms. (E and F) The well-known effect of age on alpha diversity and how the effect is observed only after the removal of bloom reads. The Kruskal-Wallis test statistic (E) and corresponding log(P value) (F) are shown for different numbers of bacteria used for the filtering before applying the test. A value of 0 on the x axis indicates no filtering. The x axis is ordered by decreasing severity score of the bloom where bloom 1 represents greater severity than bloom 2, and each point on the x axis includes the prior blooms (e.g., position 5 includes bloom sOTUs 1 through 5).

Chapter 3

Better memory management in the cloud

Chapter 2 described the first gold standard for analyzing microbial community datasets, exposed and alleviated some of the computational bottlenecks in the pipeline and showed some examples of the application of such pipeline. Some of the steps of the presented pipeline are too computationally expensive to be run in a personal laptop, and access to a supercomputer is needed to complete those steps. However, microbiologists do not necessarily have access to supercomputers and they have to rely on cloud computing. Tools such Quantitative Insights into Microbial Ecology (QIIME) [20] and IPython [153] provide Amazon Machine Image (AMI)s that enable biologists to run their analysis in Amazon's Elastic Compute Cloud (EC2) infrastructure [163]. This facilitates microbial biologists' work by avoiding the often complex task of installing the required software to run

their analyses as well as providing an environment suitable to support the large datasets that then currently face. But running these analyses in the cloud presents new challenges to microbiologists. One of the first decisions that microbial biologists face when running these analyses in the cloud is choosing the resources that their virtual machine in the cloud should contain. Usually microbiologists are unaware of the resources required by the analysis tools that they are going to use, and often the requirements of these tools is highly dependant on the nature of the data. In these cases, the biologists are left with a trial and error procedure until they have enough resources for their analysis or they have to choose a virtual machine with more resources than needed that they go underutilized. In both of these cases, the microbiologists end up spending more money (and time) than needed, which can be critical for those scientists running on a budget.

Sections 3.1 and 3.2 expose that the most critical resource on Amazon's EC2 is memory, and they describe and implement a solution that mitigates the impact of the trial and error procedure, by allowing the user's task to be completed at a small expense on running time. The material in sections 3.1 and 3.2 was published in the *43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2013* and the *International Conference on Cloud and Autonomic Computing, 2014*, respectivelly. As the first author of these publications, I conceived the idea, designed and implemented the software, designed and executed the benchmarks and wrote the text.

Section 3.1, in full, reproduces the material as it appears in "Addressing

memory exhaustion failures in Virtual Machines in a cloud environment". J. A. Navas-Molina, S. Mishra. *43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2013, DOI: 10.1109/DSN.2013.6575330.

Section 3.2, in full, reproduces the material as it appears in "CUDSwap: Tolerating Memory exhaustion failures in cloud computing". J. A. Navas-Molina, S. Mishra. *International Conference on Cloud and Autonomic Computing (ICCAC)*, 2014, DOI: 10.1109/ICCAC.2014.12.

3.1 Addressing memory exhaustion failures in Virtual Machines in a cloud environment

With the expansion of the cloud computing usage over a wide range of areas and different kinds of users, the cloud providers are taking full advantage of all their resources as much as they can. Memory is the most expensive resource in terms of oversubscribing and this has resulted in high price to the end user. Furthermore, performing swapping in Virtual Machines (VM) is expensive, so the cloud provider usually do not offer any swapping space for its systems. As a consequence, when a VM runs out of memory, user processes are killed. This scenario in the cloud environment is especially critical, since the user loses all of his/her execution time and, by extension, the money invested in this computation. This paper addresses this critical problem by providing a kernel extension that monitors the memory requirements of a VM and prevents the out of memory state by creating swapping space dynamically. The paper describes the design and implementation of a preliminary prototype of this kernel extensions and evaluates its performance.

3.1.1 Introduction

Cloud computing is increasingly being used for a wide range of applications and services mainly due to its elasticity and new application opportunities [8]. Because of this expansion, cloud providers are facing a lot of pressure to make their

physical resources handle high user demands. This has resulted in oversubscribing resources by the cloud providers. The most problematic resource to oversubscribe is memory, and indeed a cloud user is charged significantly for memory. Memory unit cost is higher than any other cloud resource ($0.028/GB versus $0.012/CPU). Cloud providers usually dont provide any swap space in their instances due to high impact on the performance of their systems. As a result, when a users VM doesnt have enough memory to execute all the running applications, user processes are killed in order to keep the VM alive. This memory exhaustion failure results in a large financial loss for the user, since all the work done by the killed processes is lost and all the money invested in these processes is wasted. Furthermore, the user has to start a new, larger VM, increasing the total cost for the user. In Linux systems, processes are killed using the Out-Of-Memory (OOM) Killer, a kernel module that prevents the Out Of Memory machine state in the VM. In this paper, we address this memory exhaustion problem by introducing a kernel module called CUDSwap. This module is designed to avoid the OOM Killer calls by adding more virtual memory to the system, i.e. adding more swapping space, when needed. CUDSwap is a dynamic kernel module that monitors the amount of free system memory, and adds swap space when the amount of free memory falls below a threshold. We have implemented a prototype of this kernel module. Through some preliminary evaluation, we show that CUDSwap prevents memory exhaustion failures. The paper describes the design, implementation and preliminary evaluation of this prototype. In addition, we provide a cost benefit analysis of using CUDSwap.

By using a dynamic approach, CUDSwap uses the storage space only when it is strictly needed. Furthermore, a lot of cloud users do not have enough computer knowledge to create the swap space before running their program, so CUDSwap creates the swap space for them. Another advantage of CUDSwap is in the case where a user unable to accurately predict her program memory requirement. In some applications, it is hard to predict the amount of memory they will need and the user may make an incorrect approximation that may result in provisioning insufficient memory to its processes. CUDSwap enables such processes to complete their execution. The remainder of this paper is organized as follows. Section 3.1.2 provides a brief review of some important related work. Section 3.1.3 describes the details of how OOM Killer management with respect to how it is invoked. Section 3.1.4 describes the design approach of the system. Section 3.1.4 describes the Linux implementation details of CUDSwap. Section 3.1.5 discusses the performance results. Section 3.1.6 provides a cost benefit analysis. Finally, Section 3.1.7 discusses future directions and concludes the paper.

3.1.2 Related Work

Memory oversubscription has been extensively studied from the cloud provider point of view, i.e. the impact on the physical machine as a result of running several VMs on it. A wide range of systems has been developed to face this challenge. In general, these systems fall into two large categories depending on their approach: VM migration or Network Memory. Systems using the VM

migration approach are tailored to support sustained periods of memory oversubscription. These systems provides support for reconfiguring a VM in a new physical machine with enough resources to fulfill VMs requirements. The main disadvantage of this approach is the VM downtime. In order to be able to migrate the VM from one physical machine to another, it has to be suspended in the old machine and resumed in the new one. Although live migration techniques allow VM migration with minimal downtime, they still have to face the network link saturation. Some examples of these systems are Xen [13], VMWares VMotion [146] or SnowFlock [99]. On the other hand, systems using the Network Memory approach are designed to support short memory overloads. These systems create a new memory hierarchy by adding a new level of memory cache between the main memory and the disk, locating it across the network. A large number of these systems use the concept of cooperative memory, which consists of performing memory swapping across the network. The swapped out pages are stored in remote page repositories. Earlier research has shown that cooperative memory has better performance that disk swap [7]. However, the performance of these systems degrades significantly when the duration of the overload increases due to network bottleneck. Examples of such systems are Cellular Disco [65], Cooperative Caching [33] or Nswap [148]. Recently, hybrid systems have been proposed in order to take advantage of the VM migration and Network Memory benefits. One such system is Overdriver [220], which monitors the memory overload and creates adaptive thresholds. Based on these thresholds, the system decides between performing Cooperative Swap or

VM migration. Our CUDSwap work differs from these earlier approaches from the memory overload point of view. While earlier approaches try to overcome the challenge of memory exhaustion failure by managing the physical memory of the host machine, we analyze the problem from the guest VM point of view. This way, we are giving the opportunity to the end user to choose between different VM configurations knowing that her applications will be completed and she can decide based on the performance-costs tradeoffs.

3.1.3 Out-Of-Memory Linux Management

The OOM machine state is an undesired state where the Kernel is not able to allocate more memory because there isnt sufficient virtual memory available, i.e. the main memory space and the swap space (in case of its existence) are full. In this scenario, the Linux kernel tries to free up memory using the OOM Killer [1]. The OOM Killer is a kernel system tailored to free up memory by killing processes. The OOM Killer is the last resource used by the kernel to free up memory, since the kernel always tries to maintain all the user processes alive. Killing processes is a critical operation, so the OOM Killer has to decide which process is the most appropriate to be killed. The OOM Killer is designed in a way that it tries to free up as much memory as possible by killing as few processes as possible (only one if it is possible), and lose as little work done (by killed processes) as possible. In order to do so, the OOM Killer assigns a rank for each process following a set

[1] `http://linux-mm.org/OOM_Killer`

of rules. The rank for each process is computed in a cumulative manner. Each process is continuously assigned points and the process that has more points is more likely to be a candidate for termination. The process rank is initialized with the amount of resident memory allocated by the process. The independent allocated memory of each child process (excluding kernel threads) is then added to the parent rank. After this, the process rank is decreased regularly by the CPU and run times. This way, processes running for a long time are more likely to be kept alive, fulfilling the premise of losing the minimum amount of work done. The rank of niced processes is doubled because they are likely less important. Next, processes with the CAP_SYS_ADMIN or CAP_SYS_RAWIO capabilities have their ranks reduced, since these processes have rights to perform system administration operations and input/output operations, respectively. They may leave the system in an inconsistent state if killed. Finally, the process rank is shifted by the value in `/proc/<pid>/oom_adj`, which is a user-defined value and set to its parent value by default. The final result of following this procedure to determine which process to kill when needed is that the processes that are killed are less important (niced), use lots of memory, have not so far executed for long, and are not performing any input/output operations.

OOM Checklist

Before calling the OOM Killer, the out of memory manager should go through a checklist in order to ensure that the OOM Killer is called if and only if

it is necessary. This checklist performs the following steps:

1. Is there enough swap space left? If yes, do not call OOM.

2. Has it been more than five seconds since last failure? If yes, do not call OOM.

3. Have we failed within the last second? If no, do not call OOM.

4. Has it been ten failures at least in the last five seconds? If no, do not call OOM.

5. Has a process been killed within the last five seconds? If yes, do not call OOM.

This checklist ensures that the system is really out of memory and it is not, for example, waiting for I/O to complete for pages swapped to disk.

3.1.4 Preventing Out-Of-Memory State

Design Overview

CUDSwaps main goal is to avoid the calls to the OOM Killer by adding virtual memory dynamically. In order to do that, CUDSwap is divided into three blocks. The first block (mod_hack_brk module) is tailored to monitor the free memory of the system and suspend the current process when there is a likelihood of memory exhaustion failure. The second block (swap_creator process) is responsible

for creating a file, format it as a swap space and activate it in order to allow the kernel to use it. Finally, the third block (wake_up module) is tailored to wake up the suspended processes. Figure 3.1 shows the overall behavior of the CUDSwap system. Each time a process requests more memory to the kernel, the mod_hack_brk module intercepts the requests and checks if the amount of free memory is below a system-dependent threshold. If it is below the threshold, the module suspends the current process, stores the process identifier in a file and notifies to the swap_creator module that a new swap space is needed. This module creates a new swap space and, when it finishes, reads the process identifiers from the file created by the mod_hack_brk and sends them to the wake_up module, which wakes up those processes and allows them to continue their execution. This division in three modules is convenient because it matches the three different steps carried out during the VM checking and creation:

1) mod_hack_brk Module: The mod_hack_brk module is a kernel module that checks the amount of free memory present in the system and checks if it is below a system-dependent threshold. By default, the kernel sets a threshold in order to know if the machine is in the out of memory state. This threshold is placed at 3% of the total amount of virtual memory present in the system. This way, the kernel has enough memory to run the OOM Killer, if needed. The mod_hack_brk is more conservative and places the threshold at 7% of the total amount of systems virtual memory. Hence, the swap_creator process will have enough memory to avoid the OOM Killer and create the swap space. In order to check the amount

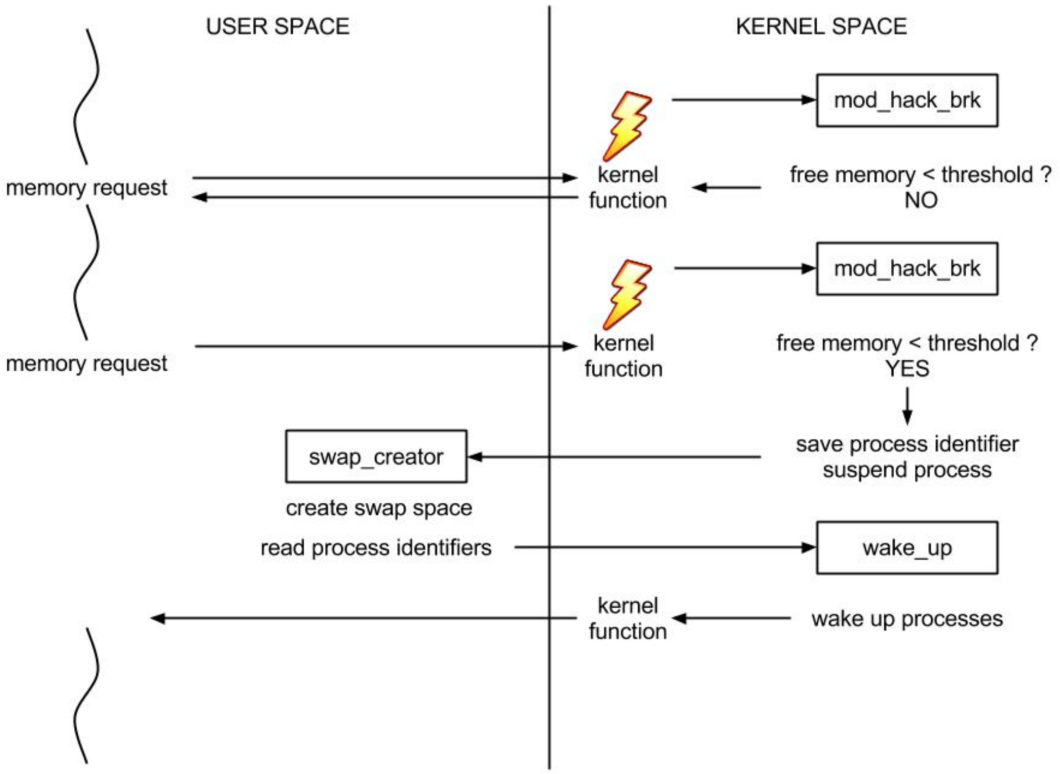


Figure 3.1: **Design overview.**

of free memory we had to decide between two main directions: perform a periodic poll or check each time a process requests more memory. The former has the main problem that it will introduce an overhead in the system even if the system is idle. Another important drawback of this solution is that we have to manage a trade-off between the time between polls and the introduced overhead: if the polls are not sufficiently frequent, we may miss an out of memory state and a user process will be killed. On the other hand, if the polls are too frequent, the introduced overhead will be prohibitive. Thus, we decided to check the amount of free memory each time a process requests more memory. When a process requests more memory to the kernel, the the mod_hack_brk module intercepts this requests and, before any kernel function is executed, checks the available memory in the system. This solution has the main drawback that it introduces and overhead each time a process requests memory, but it avoids the trade-off described above regarding the polling solution. In case the amount of free memory is below the threshold, the mod_-hack_brk module suspends the current process in order to prevent it from trying to allocate more memory. Then, the module stores the process id in a configuration file and notifies to the swap_creator process that more swap space (i.e. virtual memory) is needed.

2) swap_creator Process: The swap_creator process is a process that runs with root privileges in user space and its main function is to create a new swap space whenever needed. During most of its lifetime, this process is sleeping and it is woken up only when a new swap space is needed. The main reason why this is a

process and this functionality is not integrated within the mod_hack_brk module is that it is a bad idea in general to open files from the kernel space. I/O operations are the source of a large number of errors, and one of these errors in kernel space will cause the entire system to crash. Hence, having all these operations in user space makes CUDSwap more robust. Once this process is woken up, it performs four important steps. First, it creates a 2GB file with no holes (i.e. it is not a sparse file and it is zeroed). Second, it creates a child process that formats the created file as a swap space. Third, the swap creator process mounts the created file as a swap space, increasing the amount of virtual memory. Finally, this process reads the configuration file where the mod_hack_brk module had stored the sleeping process ids and sends them to the wake_up module.

3) wake_up Module: The wake_up module is a kernel module tailored to receive process ids of sleeping processes and wake them up. The reason why this is a kernel module is because the processes have been changed to sleeping mode in kernel space and, in order to wake them up in a reliable manner, they should be woken up in kernel space.

Implementation

1) Intercepting memory requests: In the Linux kernel, there are two main system calls that can be used by a process in order to modify its data segment: do_mmap and do_brk. The most common system call used is do_brk. In order to intercept the do_brk calls, we take advantage of Kprobes [94]. Kprobes is a kernel

debugger system that allows the module programmer to add functions before and after a certain system call is executed. This way, we can introduce a function before the do_brk system call is executed and the mod_hack_brk module can perform the needed checks to ensure the minimum free virtual memory to avoid the OOM Killer calls.

2) Getting Memory Information: The easiest way to get the memory information is by reading the `/proc/meminfo` file. Although reading a file from kernel space is not recommended, the fact that this is a well-known file reduces the probabilities of an I/O failure. However, there isnt any easy way to manage files from the kernel space because the standard libraries for manage files are not exported in kernel space. Furthermore, in the newer kernel versions, the system calls to manage files (`sys_open/read/write/close`) are not exported. Hence, the only way to manage file in kernel space is to go one step below and use low-level kernel functions, using `linux/fs.h`. Since it is hard to manage files directly using this API, we have created a simple library on top of this API that simplifies its usage and it is as similar as possible to the standard C library API. This library is a bit hackish because the functions defined in `linux/fs.h` expect that the buffer address passed as parameter belongs to the user space. The addresses that we are using are from kernel space, so these functions will fail. Our library fixes this issue by marking our addresses as safe. This hack is done using the set_fs function, as shown below.

```
old_fs = get_fs();
set_fs(KERNEL_DS);
/* File operations */
set_fs(old_fs);
```

Once we get our library for file management, we can parse the `/proc/meminfo` file and get all the needed information. We have created a structure, mem info struct, which stores all the relevant memory information and facilitates the out of memory state detection. The contents of this structure are shown below.

```
typedef struct{
        unsigned long ram;
        unsigned long swap;
        unsigned long ram_free;
        unsigned long swap_free;
        unsigned long cached;
        unsigned long buffers;
        unsigned long total_vm;
        unsigned long free_vm;
        unsigned long sys_threshold;
        unsigned long committed_vm;
        unsigned int oc_ratio;
```

```
} mem_info_struct;
```

3) mod_hack_brk - swap_creator Communication: The mod_hack_brk-swap_-creator communication is challenging because we have to perform communication from the kernel space to the user space (on the other direction, it is straightforward: system calls). Furthermore, the swap creator process is a root process and the active process during the execution of mod_hack_brk may be a non-root process without privileges to send a signal to a root process. However, both problems are solved because we have access to the signal primitives. Using the signal primitives, we can provide the entire task struct of the receiving process, and our signals do not pass the privileges checks.

4) swap_creator - wake_up Communication: The swap_creator-wake_up communication is easier since the sender is in user space and the receiver in kernel space. In this case, we have decided to use the procfs API in order to send each process id to the wake up module. The wake up module creates a new entry in the `/proc` filesystem, and the swap creator process simply opens it and writes the process identifier of the sleeping process.

3.1.5 Performance

CUDSwap is a set of modules that is always running in the system, so it will be interesting to see how it affects the overall performance of the system. Since CUDSwap mainly affects the performance of the do_brk call, we have coded

a simple benchmark that stresses this situation.

Our benchmark consists of a program that allocates a large amount of memory 2GB in our tests in chunks of 1KB. It then randomizes the character present in this memory and counts the number of characters between 0 and 9. Finally, it frees the allocated memory. These operations are performed 20 times. This simple benchmark executes a bunch of do_brk, so we can notice the overhead introduced by CUDSwap. It also accesses to all the positions in the array multiple times, so we can also notice the performance degradation due to swapping.

We performed our evaluation in Amazon EC2 [2]. We have selected an M1 Medium instance with the following characteristics: 2ECU, 3.75GB memory, 410GB storage, and Linux kernel v3.0.14. Then, we ran our test, first without CUDSwap and next with CUDSwap running on it. The total time to run our benchmark without CUDSwap was 1183.24 seconds, and with CUDSwap, it was 1223.27 seconds. Thus the overhead introduced by CUDSwap is quite low, only about 1.03X slower.

The second interesting performance comparison is running our benchmark in a smaller instance with not enough memory, creating a new swapping space and allowing it to finish. We chose an M1 Small instance with the following characteristics: 1ECU, 1.7GB memory, 160GB storage, and Linux kernel v3.0.14. The total run time in the small instance that included adding swap space through CUDSwap was 8845.32 seconds. If a medium instance is chosen instead, the run

[2] `http://www.amazon.com/ec2`

time is 1223.27 seconds. Thus, the benchmark in the smaller instance is 7.23X slower due to low performance of swapping.

Finally, we also measured the time spent to create the swap space. This time was 418.43 seconds, which is much larger than we expected. This low performance is induced by the fact that the storage in Amazon EC2 is in EBS volumes [3], which are attached to an instance through the network.

3.1.6 Cost Analysis

Since CUDSwap is tailored for the cloud environment, its evaluation has to be based on its derived costs too. First of all, we should derive the cost per resource unit (ECU per hour, GB of memory per hour and GB of storage per hour). In order to do that, we will solve systems of three equations of the type:

$$C_{cpu}P_{cpu} + C_{mem}P_{mem} + C_{disk}P_{disk} = P_{instance}$$

Here C_i and P_i are the configuration and price for the resource i and $P_{instance}$ is the price of the instance.

Instead of picking only three instance types and solving only one system of equations, we pick a subset of available instances in Amazon EC2 and solve all systems of three equations resulting from all possible permutations. Table 3.1 shows the selected configurations and their price. We used a simple Python script to generate all systems, their solutions and average them.

[3] `http://www.amazon.com/ebs`

Table 3.1: Selected instance configurations.

Name	C_{cpu}	C_{mem}	C_{disk}	$C_{instance}$
M1 Medium	2	3.75	410	0.139
M1 Large	4	7.5	850	0.260
M1 Extra-large	8	15	1690	0.520
M3 Extra-large	13	15	0	0.580
M3 Double extra-large	26	30	0	1.160

Our calculations show that the cost for each ECU per hour is $0.012, for each GB of memory per hour is $0.028 and for each GB of storage per hour is $0. We can see that the user is charged more for memory usage than for the CPU usage. Hence, one possible way to save users money is to use a smaller instance with swap space enabled.

If we pick as an example the test run from the previous section, where the benchmark took 8845.32 seconds (2-3 hours) in the small instance and 1223.27 seconds (<1 hour) in the medium instance, we can see that the user is charged $0.195 (3 * 0.065) in the small instance case and $0.130 in the medium instance. Hence, if the user knows in advance that his memory requirements will exceed that of small instance, it is cheaper to run the code in the medium instance than in the smaller one with swap space.

However, the common case is one where the user doesnt know in advance

the total memory requirements. In this case, the user typically runs his code in the smaller instance and, when it gets killed, it terminates the small instance and runs his code in the medium instance. Using the same example as above, we note that the user would have spent a total amount of $0.195 (1 hour for the small instance $0.065 and 1 hour for the medium one $0.130). Then, we can see that the user would have spent the same amount of money as running the code on a small instance with swap space. In such a case, it is better for the user to use the small instance with swap space as that avoids the hassle of moving the application from one instance to another. Although in this case CUDSwap dont save users money, it improves the users experience because she doesnt feel that she is wasting the money on the small instance when it gets killed.

3.1.7 Conclusion

In this paper we have described CUDSwap, a set of kernel modules that prevents the memory exhaustion failure in virtual machines in cloud computing environment. We demonstrated that the memory oversubscription is the most expensive resource in a cloud environment and this cost is shifted to end user. Finally, we showed how CUDSwap could improve users experience in a cloud environment.

The current prototype of CUDSwap is only a preliminary prototype and has a large scope for improvement. The first source of improvement will be to obtain the memory information directly from the kernel routines instead of reading the `/proc/info` file. This will improve the out of memory state detection. With the

current implementation, we may miss an out of memory state if a process tries to allocate more than 4% of total memory with a single call. This is because we check for memory threshold before a memory allocation takes place, but we do not take into account how much memory the process is going to allocate.

The second source of improvement will be the way process identifiers are shared between the mod_hack_brk and the swap_creator process. Currently, this is done through a configuration file, but one possible approach will be using the procfs as in the wake_up module case. This change will completely avoid the necessity to open files in kernel space, making CUDSwap more robust and compliant with Linux kernel development standards.

Finally, our current performance testing of CUDSwap is quite preliminary and limited. We need to perform an extensive evaluation of CUDSwap with a wide variety of applications having varied memory requirements. In particular, we need to test UDSwap with standard memory benchmarks, and perform a cost benefit analysis. This future tests can show cases where a smaller instance with swap space is cheaper than a bigger instance with enough memory, as well as they will allow to characterize the situations where CUDSwap is highly useful.

3.2 CUDSwap: Tolerating Memory exhaustion failures in cloud computing

Cloud computing is now being used by a wide variety of users, ranging from expert programmers and system administrators to scientists and laymen. Cloud providers are taking full advantage of all their resources as much as they can. Memory is the most expensive resource in terms of oversubscription and this has resulted in high price to the end user. Furthermore, performing swapping in Virtual Machines (VM) is expensive, so the cloud providers usually do not offer any swapping space for their systems. As a consequence, when a VM runs out of memory, user processes are killed. This scenario in cloud environment is especially critical, since the user loses all of his/her execution time and, by extension, the money invested in this computation. For cloud users such as life scientists with varying memory requirements, this is a critical problem. This paper presents CUDSwap, a kernel extension module designed to detect memory exhaustion in cloud instances, add more swap space, and thus prevent process failures. CUDSwap has been implemented in Linux kernel and has been evaluated over a variety of workloads as well as real-world life science applications. The paper describes CUDSwap design and implementation details, and reports performance details from the evaluation.

3.2.1 Introduction

Cloud computing is being used in a wide variety of fields, including web hosting, media content, scientific computing, and many more. This wide range implies that the cloud users are not necessarily computing experts, especially in the world of scientific computing that includes biologists, physicists, chemists, etc. From our past experience working with scientists, we have found that their experience with the cloud is often quite poor, and sometimes they feel that they are wasting money using the cloud. Usually, scientists do not have an a priori idea of the amount of various computing resources they will need to complete their computing tasks. Typically, they perform a rough estimation of the resources they will need and launch the cloud instance that they think will be enough to run their applications. An incorrect estimation of required resources can lead to poor performance and even complete failure. In the case of an incorrect CPU estimation, the application will simply take longer to finish, but the work is not lost. In the case of incorrect storage estimation, the user can dynamically add more storage to the instance, so that the application can continue working without any problem.

However, the most critical resource is memory, because when it is exhausted, the application is aborted and all the work done by the application is lost. This is because cloud providers usually don't provide any swap space in their instances due to high impact on the performance of their systems. As a result, when a user's virtual machine (VM) doesn't have enough memory to execute all the running

applications, user processes are killed in order to keep the VM alive. In Linux systems, processes are killed using the Out Of Memory Killer (OOM Killer), a kernel module that prevents the Out Of Memory machine state in the VM. This memory exhaustion failure not only results in poor user experience, but also results in a large financial loss for the user. All the work done before a process is killed is lost and all the money invested in these processes is wasted. Furthermore, the user has to start a new, larger VM, increasing the total cost for the user.

In this paper, we address this memory exhaustion failure problem in VM instances by introducing CUDSwap, an elegant kernel module that requires minimal changes to the current Linux kernel. This module is designed to avoid the OOM Killer calls by increasing the amount of virtual memory on the fly. CUDSwap is a dynamic kernel module that monitors the amount of free system memory, and adds swap space whenever needed, so that the application process is not killed. We have implemented a prototype of CUDSwap for Linux kernel and evaluated it extensively for a variety of applications ranging from artificial workloads to real-world, life science applications on Amazon EC2. This evaluation demonstrates that CUDSwap prevents memory exhaustion failures of applications running on cloud. In addition, CUDSwap improves user experience and even reduces the total cost of running applications on cloud. We provide a detailed cost benefit analysis of using CUDSwap.

By using a dynamic approach, CUDSwap uses the storage space only when it is strictly needed. Furthermore, with increasing use of cloud in areas beyond

computer science, a large number of users do not have enough computing background to create swap space before running their program. CUDSwap creates swap space for them. Another advantage of CUDSwap is in the case where a user is unable to accurately predict her program memory requirement. In some applications, it is hard to predict the amount of memory they will need and the user may make an incorrect approximation that may result in provisioning insufficient memory to its processes. CUDSwap enables such processes to complete their execution.

The remainder of this paper is organized as follows. Section 3.2.2 provides a brief review of some important related work. Section 3.2.3 describes the details of how VMs manage memory exhaustion at present. Section 3.2.4 describes the design approach of the system. Section 3.2.5 describes the Linux implementation details of CUDSwap. Section 3.2.6 discusses the performance results, as well as a cost benefit analysis. Finally, Section 3.2.7 discusses future directions and concludes the paper.

3.2.2 Related Work

Earlier work on memory exhaustion failures in the cloud has not been focused on the users point of view. They are targeted to provide solutions to the memory oversubscription problem from the provider's point of view, i.e. the impact on the physical machine as a result of running several VMs on it. There are mainly two approaches to this problem: VM migration and Network Memory.

Systems that provide VM migration are Xen [13], VMWare's VMotion [146]

and SnowFlock [99], among others. These systems move a VM that is hosted in a machine with not enough memory and re-deploys it in a new machine. They provide support for reconfiguring a VM in a new physical machine with enough resources to fulfill VM's requirements. They are tailored to support sustained periods of memory oversubscription. The main disadvantage of this approach is the VM downtime. In order to be able to migrate the VM from one physical machine to another, it has to be suspended in the old machine and resumed in the new one. Although live migration techniques allow VM migration with minimal downtime, they still have to face the problem of network link saturation.

Systems using the Network Memory approach include Cellular Disco [65], Cooperative Caching [33] and Nswap [148], among others. These systems are designed to support short memory overloads. They create a new level on memory hierarchy by adding a new level of memory cache between the main memory and the disk, located across the network. A large number of these systems use the concept of cooperative memory, which consists of performing memory swapping across the network. The swapped out pages are stored in remote page repositories. Earlier research has shown that cooperative memory has better performance that disk swap [7]. However, these systems do not support long periods of memory oversubscription and are tailored for short bursts of memory overloads. The performance of these systems degrades significantly when the duration of the overload increases due to network bottleneck.

Recently, hybrid systems have been proposed in order to take advantage of

the VM migration and Network Memory benefits. One such system is Overdriver [220], which monitors the memory overload and creates adaptive thresholds. Based on these thresholds, the system decides between performing Cooperative Swap or VM migration.

However, all the above-described systems do not solve the problem of the memory exhaustion on the client VM instance. CUDSwap differs from these earlier approaches from the memory overload point of view. While earlier approaches tried to overcome the challenge of memory exhaustion failure by managing the physical memory of the host machine, we address the problem from the guest VM's point of view. This way, we are giving the opportunity to the end user to choose between different VM configurations knowing that her applications will be completed and she can decide based on performance-cost tradeoffs. In [136], we provided an analysis of this problem and proposed a preliminary solution for it. However, that solution relies on a heuristic that can potentially miss an Out Of Memory state, resulting in the OOM Killer being called. The heuristic is based on the default memory threshold (3% of the total memory) that Linux uses to kill a process if the system is hitting the OOM state. Specifically, CUDSwap uses a 7% threshold, which is checked at the time a process requests more memory. However, it does not take into account the amount of memory requested by the process, so if it requests more than $7\% - 3\% = 4\%$ of the total amount of memory, CUDSwap will miss the OOM state and the process will be killed. Here, we will describe the improvements we have made in the design and implementation of CUDSwap based

on our experience in [136].

3.2.3 Memory Exhaustion in VMs

The Out Of Memory (OOM) machine state is an undesired state where the kernel is not able to allocate more memory because there isn't sufficient virtual memory available, i.e. the main memory space and the swap space (in case of its existence) are full. In this scenario, the Linux kernel tries to free up memory using the OOM Killer [4]. The OOM Killer is a kernel system tailored to free up memory by killing processes.

The OOM Killer is the last resource used by the kernel to free up memory, since the kernel always tries to keep all the user processes alive. Killing processes is a critical operation, so the OOM Killer has to decide which process is the most appropriate to be killed. The OOM Killer is designed in a way that it tries to free up as much memory as possible by killing as few processes as possible (only one if it is possible), and lose as little work done (by killed processes) as possible. In order to do so, the OOM Killer assigns a *rank* for each process following a set of rules. The rank for each process is computed in a cumulative manner. Each process is continuously assigned points and the process that has more points is more likely to be a candidate for termination. The process rank is initialized with the amount of resident memory allocated by the process. The independent allocated memory of each child process (excluding kernel threads) is then added to the parent rank.

[4] `http://linux-mm.org/OOM_Killer`

After this, the process rank is decreased regularly by the CPU and run times. This way, processes running for a long time are more likely to be kept alive, fulfilling the premise of losing the minimum amount of work done.

The rank of *niced* processes is doubled because they are likely less important. Next, processes with the *CAP_SYS_ADMIN* or *CAP_SYS_RAWIO* capabilities have their ranks reduced, since these processes have rights to perform system administration operations and input/output operations, respectively. They may leave the system in an inconsistent state if killed. Finally, the process rank is shifted by the value in */proc/<pid>/oom_adj*, which is a user-defined value and set to its parent value by default.

The final result of following this procedure to determine which process to kill when needed is that the processes that are killed are less important (niced), use lots of memory, have not so far executed for long, and are not performing any input/output operations.

OOM Checklist

Before calling the OOM Killer, the out of memory manager should go through a checklist in order to ensure that the OOM Killer is called if and only if it is necessary. This checklist performs the following steps:

1. Is there enough swap space left? If yes, do not call OOM.

2. Has it been more than five seconds since last failure? If yes, do not call

OOM.

3. Have we failed within the last second? If no, do not call OOM.

4. Has it been ten failures at least in the last five seconds? If no, do not call OOM.

5. Has a process been killed within the last five seconds? If yes, do not call OOM.

This checklist ensures that the system is really out of memory and it is not, for example, waiting for I/O to complete for pages swapped to disk.

3.2.4 CUDSwap Design

In order to prevent calls to the OOM killer, CUDSwap needs to perform three important functions. First, it needs to monitor the available free space in memory. Second, it needs to suspend a process requesting new memory whenever the available free space goes below some threshold and create additional swap space. Finally, it needs to wake the suspended process after additional swap space has been created. CUDSwap consists of two main modules: *mod_hack_brk* and *swap_creator* (See Figure 3.2).

mod_hack_brk Module

The *mod_hack_brk* module is tailored to monitor the free memory of the system and suspend the current process if its current memory request will produce

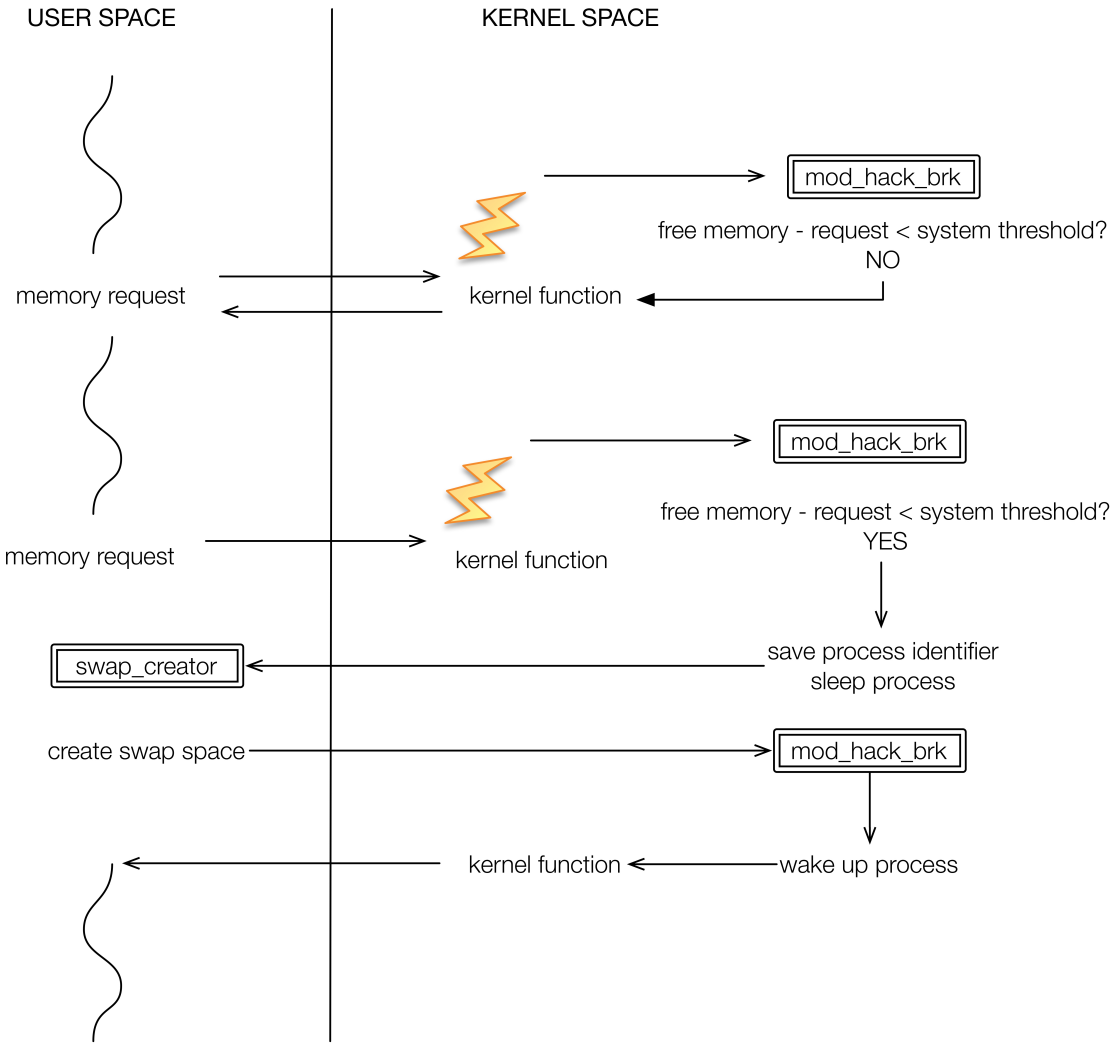


Figure 3.2: Design Overview.

a memory exhaustion failure.. By default, the Linux kernel sets a threshold in order to know if the machine is in the out of memory state. This threshold is placed at 3% of the total amount of virtual memory present in the system. This way, the kernel has enough memory to run the OOM Killer, if needed. The *mod_hack_brk* module essentially checks if the free memory will go below this threshold if more memory is allocated to a process.

A key question is when should the *mod_hack_brk* module check whether the amount of free memory has fallen below the threshold at 3% of the total amount of system's virtual memory. There are two options: perform a periodic poll or check each time a process requests more memory. The former has the problem that it will introduce an overhead in the system even if the system is idle. Another important drawback of this solution is that we have to manage a trade-off between the polling time period and the overhead introduced due to polling: if the polls are not sufficiently frequent, we may miss an out of memory state, which may result in a process being killed. On the other hand, if the polls are too frequent, the introduced polling overhead will be prohibitive.

For these reasons, we decided to check the amount of free memory each time a process requests more memory. Each time a process requests more memory from the kernel, the *mod_hack_brk* module intercepts the requests and checks if the amount of free memory in the system minus the amount of memory requested by the process is below the system threshold. If so, it puts the requester process to sleep, saves the process id of that process, and then wakes up the *swap_creator*

module (described below) to create new swap space. After new swap space has been created, the *swap_creator* process notifies the *mod_hack_brk* module, which wakes up all the processes that were put to sleep. This solution ensures that only processes that have requested more memory than available in the system are put in sleep mode, while memory requests of processes requesting smaller amounts of memory are satisfied.

There are two main system calls that can be used by a process to modify its data segment: *do_mmap* and *do_brk*. The *mod_hack_brk* module intercepts these system calls and, before they are executed, checks the available memory in the system. The main drawback of this solution is that it introduces an overhead each time the *do_brk* or *do_mmap* system call is executed, but it avoids the trade-off described above regarding the polling solution.

swap_creator Module

The *swap_creator* module is a process that runs with root privileges in user space and its main function is to create new swap space whenever needed. During most of its lifetime, this process is sleeping and it is woken up by the *mod_hack_brk* module only when new swap space is needed.

Once this process is woken up, it performs three important steps. First, it creates a 2GB file with no holes (i.e. it is not a sparse file and it is zeroed). Second, it creates a child process that executes the *mkswap* command on the created file. Finally, the *swap_creator* process mounts the created file as a swap

space, increasing the amount of virtual memory. After this final step, this process notifies the *mod_hack_brk* module, and goes back to sleep.

Design Discussion

The functionality of creating new swap space is implemented as a separate module (*swap_creator* module) running in the user space. The main reason why this functionality is not integrated within the *mod_hack_brk* module is that it is a bad idea in general to open files from the kernel space. I/O operations are the source of a large number of errors, and one of these errors in the kernel space will cause the entire system to crash. Hence, having these operations in user space makes CUDSwap more robust.

We highlight the fact that in this design, new swap space is created strictly if it is needed, i.e. if the amount of free memory in the system minus the amount of memory requested by the process is below the system threshold. This is different from our earlier design [136] where new swap space was created if there was a likelihood of memory exhaustion. After performing several experiments, we concluded that this higher threshold is too conservative. We observed that it resulted in wasting system resources most of the time, i.e. new swap space was created when the available free memory would not have fallen below 3% and hence OOM Killer process wouldn't have been invoked.

In addition, the system design has been simplified from our earlier design. The *wake_up* module from the earlier implementation has been removed and its

functionality has been included in the *mod_hack_brk* module. The main reason for this change was to improve security and reliability of the system. The process identifiers are no longer stored in a configuration file, they are stored in memory in the *mod_hack_brk* module. This way, the process identifiers are no longer exposed to the file system, which can be targeted by third party applications that can modify it, adding or removing process ids, resulting in an undesired behavior in the system. This change also simplifies the functionality of the *swap_creator* module. It no longer needs to manage the process ids of the processes that need to be woken up.

3.2.5 CUDSwap Implementation

Intercepting do_brk and do_mmap

We need to not only intercept *do_brk* or *do_mmap* calls, but also have access to the amount of memory that the process is requesting. This information can be obtained using Jprobes [129], another flavor of Kprobes [94] that gives access to the function call parameters. Jprobes is a kernel debugger system that allows the module programmers to add functions before and after a certain system call is executed. This way, we can introduce a function before the *do_brk* system call is executed, and the *mod_hack_brk* module can perform the needed checks to ensure the minimum free virtual memory to avoid the OOM Killer calls. Using the parameters of the system call, we can compute the amount of memory that the

process is requesting, so we can deterministically decide whether the system has enough memory to handle the request.

Getting Memory Information

We obtain memory information directly from kernel routines. The Linux kernel provides two different routines for getting the memory information: *si_meminfo* and *si_swapinfo*. The former provides the information of the RAM usage, while the latter provides the information of the swap space. However, the *si_swapinfo* routine is not exported to the Loadable Kernel Module (LKM) space. We address this problem by recognizing that the two routines have the same signature and their usage in the LKM does not incur any security issue. Thus, we have modified the Linux kernel source to export the *si_swapinfo* routine and then use it in our *mod_hack_brk* module. We should highlight that this is the the only change needed in the current Linux kernel.

This process of obtaining memory information directly from the kernel routines is a significant change from our earlier implementation that read memory information from the */proc/meminfo* file. This change completely removes the interaction of the kernel with the file system, removing any source of kernel failure due to this interaction.

mod_hack_brk - swap_creator Communication

The *mod_hack_brk* and *swap_creator* modules have to communicate in both directions. The *mod_hack_brk* module needs to wake up the *swap_creator* module whenever new swap space is needed, and the *swap_creator* module needs to communicate with the *mod_hack_brk* module after the new swap space has been created. The former communication is challenging because we have to perform communication from the kernel space to the user space. Furthermore, the *swap_creator* process is a root process and the active process during the execution of *mod_hack_brk* may be a non-root process without privileges to send a signal to a root process. However, both problems are solved because we have access to the signal primitives. Using the signal primitives, we can provide the entire *task_struct* of the receiving process, and our signals do not go through the privilege checks.

For communication from the *swap_creator* process to the *mod_hack_brk* module, the *mod_hack_brk* module creates a new entry on the /proc virtual file system and the *swap_creator* process only writes a single value to notify that the swap space has been successfully created. This implementation is much more secure than our earlier implementation, since the process ids are no longer provided through the /proc file system and the *mod_hack_brk* module can ensure that the process ids that it has are correct and secure to use.

3.2.6 CUDSwap Evaluation

To evaluate the performance of CUDSwap, we have used four different workloads:

1. Workload 1: An artificial application that executes ten times a large chunk of memory allocation and performs a sequential write followed by a sequential read on it.

2. Workload 2: An artificial application that allocates a large chunk of memory and performs a random write followed by a random read on the entire chunk of memory.

3. Workload 3: An artificial application that executes Workload 1, and in parallel also executes a process that continuously writes a large file to disk, in order to stress the I/O system.

4. Workload 4: A real-world, bioinformatics application. We have used the sequence clustering step on the QIIME pipeline [20] using the UCLUST algorithm [43]. This step takes a sequence file with the input sequence and a reference file with the cluster sequence seeds. It then parses the input file and tries to group the input sequence with reference seeds such that they are similar above some user-defined threshold.

In order to conduct our experiments, we have used three different instances of Amazon EC2 [5]: Micro, Small and Medium instances. Table 3.2 shows the

[5] `http://www.amazon.com/ec2`

characteristics of these instances. We decided to use a real cloud to conduct our experiments in order to remove the potential infrastructure differences present in a controlled lab environment. Instead, our controlled set of applications that exhibits different memory behaviors (including a real-world scientific application) provides much more realistic performance measures.

Table 3.2: Selected instance configurations.

Instance	CPU	Memory	Storage	Price
Micro	1 or 2 ECU	615 MB	EBS only	$0.02 per hr
Small	1 ECU	1.7 GB	160 GB	$0.06 per hr
Medium	2 ECU	3.75 GB	410 GB	$0.12 per hr

Micro vs Small Instance

In our first experiment, we compare the performance of running the four workloads on Micro instance versus running them on Small instance. For the Workloads 1, 2 and 3, we have used 1 GB of memory, which is considerably larger than the amount of memory available on the Micro instance. For the Workload 4, we have used a subset of 1,000,000 input sequences from Yatsunenko human gut microbiome study [224] and the 94% representative sequence set from the GreenGenes database [37]. With these parameters, Workload 4 uses about 0.7 GB of memory, which is again larger than the amount of memory available on the

Micro instance. First, we ran all four workloads on the Micro instance running standard VM without the CUDSwap kernel extension. In all four cases, our jobs were killed after a few minutes of execution.

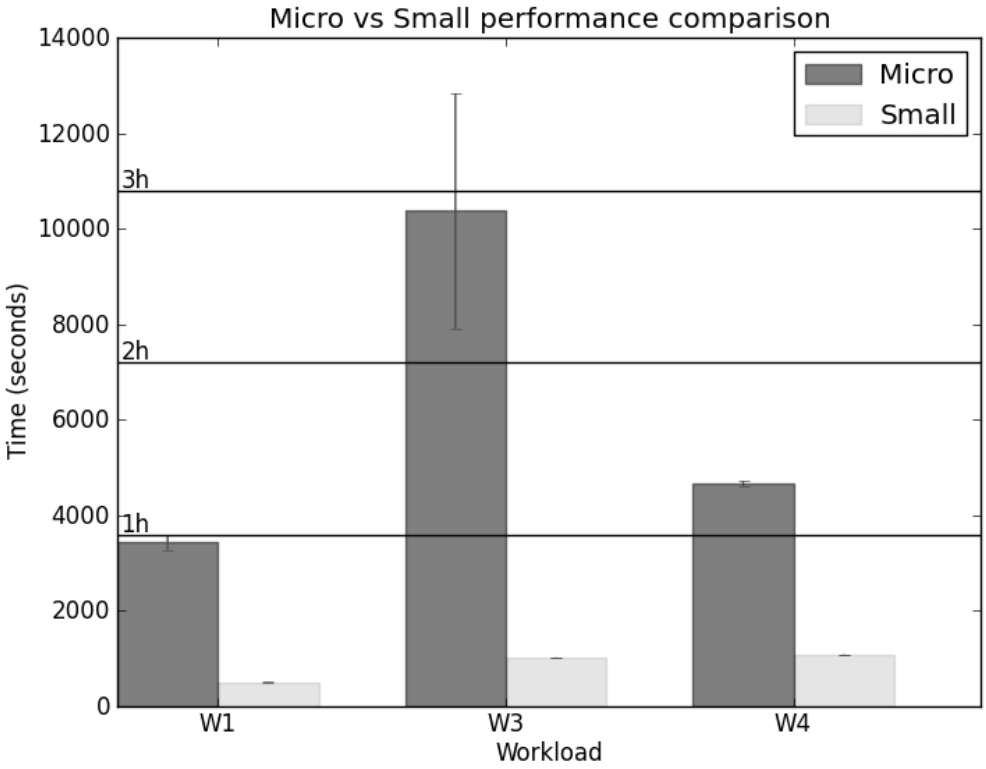


Figure 3.3: **Comparison of the different workload performance between the Micro instance and the Small instance.**

Next, we ran the four workloads on the Micro instance and the Small instance, in which the Micro instance VM incorporates the CUDSwap kernel extension. Figure 3.3 shows the results obtained for different workloads. The results of the Workload 2 are not shown in the figure for clarity. While it took only about 8 minutes for Workload 2 to be executed on the small instance, it did not finish even after 20 hours in the micro instance. This huge difference is caused by the fact

that Workload 2 is designed to remove memory locality completely. As a result, each time the process tries to access a memory location, it is almost always located on the swap space, causing the system to swap pages in and out aggressively.

Our first observation is that none of our jobs on Micro instance were killed, indicating that CUDSwap successfully created new swap space and prevented calls to OOM killer. Of course, in each case, the execution time on the Micro instance is considerably larger than the execution time on the Small instance. Nevertheless, it is important to note that CUDSwap enables completion of a job despite an incorrect estimation of memory requirements.

Performance on the Micro instance was slower than the performance on the Small instance by 6.73X for Workload 1, 10.14X for Workload 3, and 4.32X for Workload 4. However, since Amazon EC2 works as a pay-as-you-go service, we should take into account the cost of instances in our evaluation too, in addition to the performance. From the cost information in Table 3.2, we notice that the execution times of Workloads 1 and 4 are less than two hours, and so, using a Micro instance with CUDSwap will be cheaper than using a Small instance ($0.02 versus $0.06 for Workload 1, and $0.04 versus $0.06 for Workload 4). For Workload 3, however, there is no benefit of using a Micro instance with CUDSwap instead of using a Small instance. Most of the time they will have the same cost ($0.06), and sometimes the Micro instance may be more expensive if ends up taking more than 3 hours to complete the job ($0.08).

So, based on these workloads, we notice that CUDSwap may result in re-

ducing the cost to a user in some cases, but will result in higher execution times. However, it is important to note that CUDSwap is designed for situations where a user makes an incorrect estimation of his/her memory requirements. In such cases, the user will start his/her job on a Micro instance, incur the cost of Micro instance, notice that the job has been killed, and then start the job again on a Small instance and thus incur the cost of Small instance in addition. In this situation, the cost of running Workload 3 on Micro instance is also cheaper than the cost of running it first on Micro instance and then later on the Small instance ($0.06 in the first case versus $0.08 in the second case). Furthermore, while considering the performance in the second case, we should also take into account the time lost due to first running the job partially on Micro instance and then setting up a Small instance and restarting the job. This time could be several minutes or even more depending on when the job on the Micro instance is killed. So, even in terms of performance, CUDSwap may result in saving time. Finally, CUDSwap provides a better user experience. Users (especially non-computer scientists) will typically get annoyed or frustrated when they see their jobs being killed and losing all the work after running for some amount time. CUDSwap avoids this situation.

Small vs Medium Instance

In order to compare the Small instance versus the Medium instance we have changed the parameters of our workloads. In this case, for Workloads 1 and 3, we have used a chunk of memory of 2GB. Due to the results obtained in

our previous test, we have not used Workload 2 here, as it will have very poor performance on the Small instance and we are not going to get any benefit. For Workload 4, we have used the same subset of 1,000,000 input sequences, but we have used the 99% representative sequence set from the GreenGenes database. With these parameters, Workload 4 uses about 2.28 GB of memory. Again, we first ran all three workloads on the Small instance running standard VM without the CUDSwap kernel extension. In all three cases, our jobs were killed after a few minutes of execution.

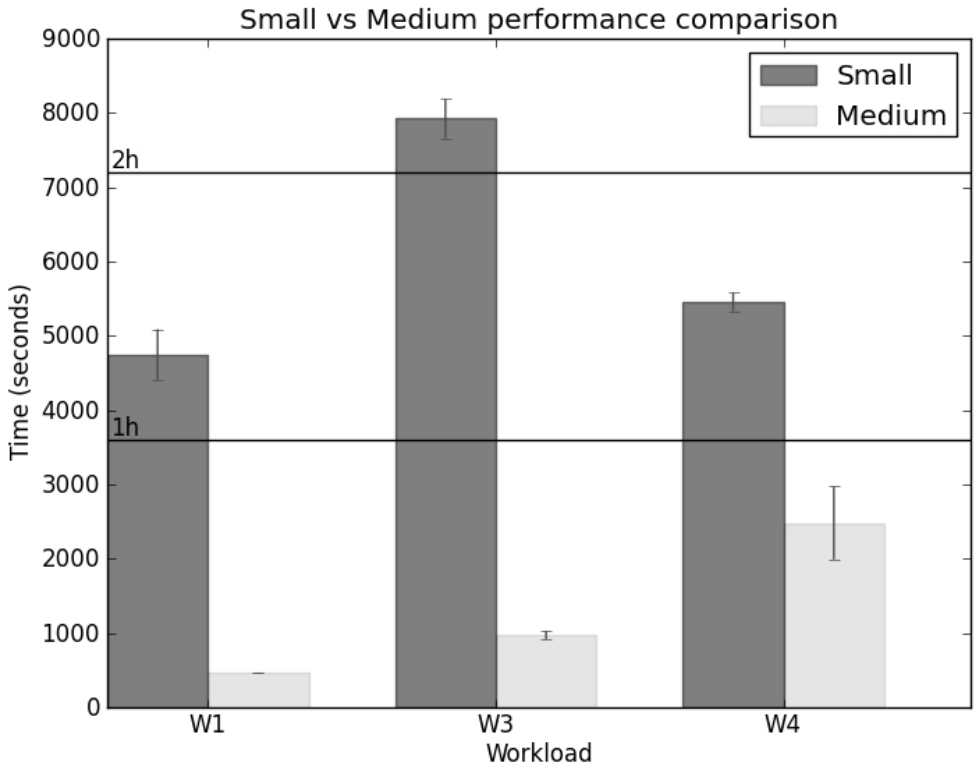


Figure 3.4: **Comparison of the different workload performance between the Small instance and the Medium instance.**

Next, we ran the three workloads on the Small instance and the Medium

instance, in which the Small instance VM incorporates the CUDSwap kernel extension. Figure 3.4 shows the results obtained for the three different workloads. Again, our first observation is that none of our jobs on Small instance were killed, indicating that CUDSwap successfully created new swap space and prevented calls to OOM killer. Of course, in each case, the execution time on the Small instance is considerably larger than the execution time on the Medium instance.

Performance degradation on the Small instance in comparison to the Medium instance is 10.01X for Workload 1, 8.14X for Workload 3 and 2.20X for Workload 4. In terms of cost incrurred in using a Small instance with CUDSwap versus using a Medium instance, we see that there is no advantage of using CUDSwap. The cost is same for Workloads 1 and 4 ($0.12) and more expensive for workload 3 ($0.18 for Small instance versus $0.12 for Medium instance). However, considering the scenario where a user first starts his/her job on a Small instance, notices that the job is killed, and then restarts the job on Medium instance, we see a cost advantage of using the Small instance with CUDSwap. The cost is $0.12 for Workloads 1 and 4 running on Small instance with CUDSwap versus $0.18 or $0.24 for the Small/Medium instance scenario outlined above. Similarly, for Workload 3, the cost is $0.18 for Small instance and $0.18, $0.24 or $0.30 for the Small/Medium instance scenario.

In addition, the last two observations we made in Subsection 3.2.6 are relevant here as well. When we take into account the time lost due to first running the job partially on Small instance and then setting up a Medium instance and

restarting the job, CUDSwap may result in saving time as well. Finally, CUDSwap provides a better user experience by preventing calls to OOM killer and ensuring that the user job is not killed even when the user incorrectly estimates his/her memory requirements.

Performance Overhead of CUDSwap

Since CUDSwap is a monitoring module that is always running on the system, an application incurs the monitoring overhead even if it never needs additional swap space. Thus, it is important to evaluate the performance overhead incurred due to CUDSwap usage. Monitoring overhead of CUDSwap comes from the inception of every *do_brk* and *do_mmap* calls that the application makes. These calls are made whenever the application requests new memory.

To estimate this overhead, we ran Workload 1 and Workload 4 on a Medium instance under two different configurations, on a standard Linux kernel without CUDSwap kernel extensions and on a Linux Kernel with CUDSwap kernel extension. For Workload 1, we used 2GB of memory, and for Workload 4, we used the same configuration as the one we used in our Small versus Medium experiment. Workloads 2 and 3 have the same memory allocation (memory request) patterns as Workload 1 and since CUDSwap only interferes during the allocation process, their overhead will be same as that in Workload 1.

Figure 3.5 shows the results of the experiment. As we can see in the plot, performance overhead incurred by CUDSwap is negligible (less than 1%). This

means there is no downside to using CUDSwap continuously in the system even when the memory requirements of the jobs have been correctly estimated by the user and no new swap space is needed.

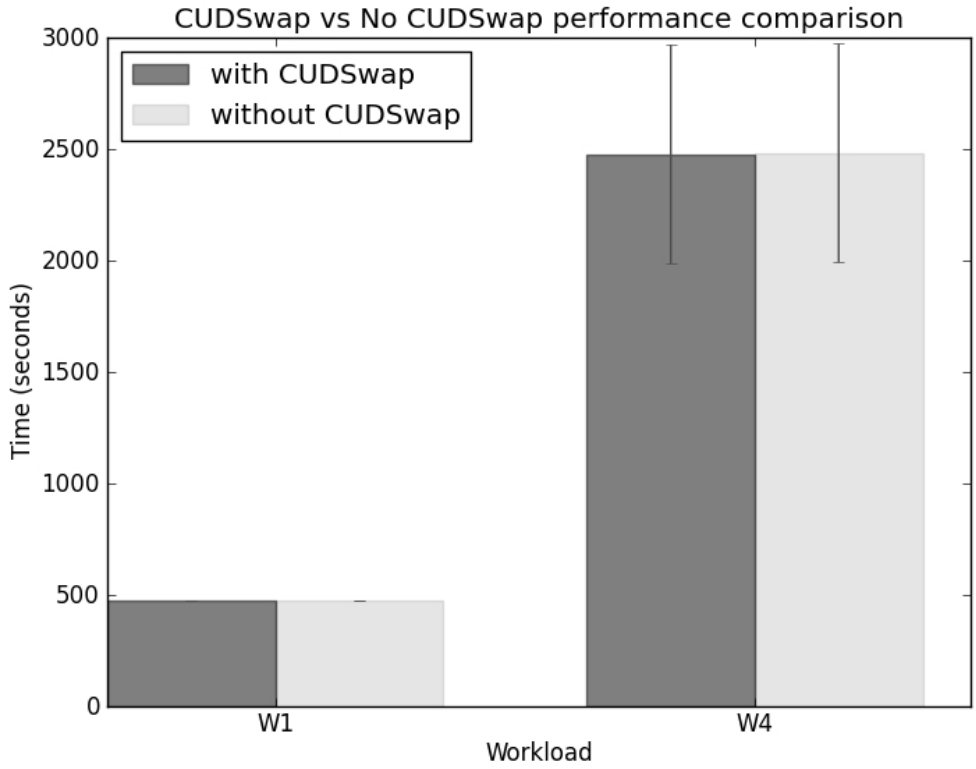


Figure 3.5: **Comparison of Workloads 1 and 4 performance on a Medium instance with and without CUDSwap.**

3.2.7 Conclusion

We have presented and evaluated CUDSwap, a memory monitor for guest operating systems that automatically adds virtual memory to the system creating a swap space when the VM is running out of memory. Memory is the most expensive resource in a cloud environment and any memory oversubscription cost is shifted to

the end user. Our evaluation demonstrates that CUDSwap prevents calls to OOM killer when the VM is short on memory by detecting this situation and adding more swap space. Furthermore, overhead of CUDSwap is negligible under normal circumstances when the VM has sufficient memory.

CUDSwap enables completion of a job despite an incorrect estimation of memory requirements. This is an important functionality because it is difficult for most cloud users to estimate accurately their application's memory requirements. In such circumstances, users may either oversubscribe by renting larger instances than needed, or undersubscribe by renting a smaller instance than needed and incurring job failure. In both cases, users end up spending more money. CUDSwap enables users to save money in situations where users may undersubscribe. In fact, our evaluation shows that CUDSwap also enables overall time saving for the user in case of undersubscription, if we consider the entire duration of staring a job on smaller instance, incurring job failure after some partial execution and then subscribing a larger instance. So, overall CUDSwap is very useful in terms of saving both time and money for a user who undersubscribes due to an incorrect estimation of his/her memory requirements.

While computing with the cloud, multiple evaluation criteria should be taken into account. Specifically, Cloud computing costs incurred by the user are as important as the running time of the computing job. Actual tradeoff should be left to the user. A user may choose to incur high runtime cost on a lower budget, while another user may choose to incur low runtime cost on a high budget. Thus,

although applications may be running up to 10X slower due to process thrashing, the results presented on section 3.2.6 show that a user can save money. Thus, a user may choose to undersubscribe his/her instance in favor of reducing the costs of using the cloud.

There is another subtle advantage of using CUDSwap in the form of better user experience. In a scenario where a user encounters a failure and restarts his/her job on a larger instance, he/she may get annoyed or frustrated feeling that he/she has wasted time and money. CUDSwap avoids this scenario.

In general, CUDSwap is useful only when there is difficulty in estimating the memory requirements of a job. If a user can accurately predict his/her memory requirements, he/she should certainly subscribe the instance that provides sufficient memory for job completion, and not undersubscribe and depend on CUDSwap.

CUDSwap is being used by several graduate students in the BioFrontiers Institute at our university. Jose Antonio Navas-Molina is a graduate student of the BioFrontiers Institute and started working on CUDSwap after encountering the memory undersubscription problem. Our current implementation of CUDSwap is quite stable. Nevertheless, there are some additional future directions that we are addressing. At present, CUDSwap intercepts *do_brk* and *do_mmap* systems calls. Another way of consuming memory in the system is by a forking new process, which creates a new process and a new chunk of memory needs to be allocated for the new process. This situation can also be detected using the Jprobes system and we plan to incorporate it in CUDSwap in the future.

Another area that we plan to investigate is the utility of CUDSwap for applications that have widely varying memory requirements with very short periods of peak requirements. In the absence of CUDSwap, a user will have to subscribe a larger instance to satisfy these peak requirements. With CUDSwap, it may be more optimal to subscribe a smaller instance, since the time in actual swapping will be short.

3.2.8 Acknowledgments

Jose Antonio Navas-Molina is supported by the Balsells fellowship.

Chapter 4

Meta-analyses: importance, challenges and solutions

A unifying theme in the work presented in section 2.3 is that it takes advantage of previously published datasets to increase the power of the findings. The Komodo Dragon paper [81] compared new findings on captive Komodo dragons with wild amphibians [98] and humans and pets living in homes [106] to hypothesize that the lack of interactions with an open environment can negatively affect human and animal health. The Earth Microbiome Project (EMP) [59, 58, 197] combined samples from 97 independent studies to answer spatial, temporal and evolutionary microbial community questions at a global scale. The American Gut Project (AGP) used previous studies to show that the findings resulted from citizen-science microbiome research can replicate previous results and, move research forward by creating a massive dataset of human samples that can be used to generate new

hypotheses. Finally, the manuscript about correcting microbial blooms [5] used previously published datasets to describe a new technique used to reduce technical differences caused by sample shipping at room temperature.

This technique of using multiple datasets (published or not) to improve the findings of a study is known as a meta-analyses. As shown by the previous examples, meta-analyses is a powerful tool that is becoming increasingly common on microbiome research [120, 108, 182]. However, this extra power comes with its own set of challenges that can delay the publication of the results from months (as in the Komodo Draogn manuscript) to years (as in the EMP manuscript). Namely, this challenges can be grouped in three topics: (1) technical differences, (2) data availability, and (3) data standardization.

Technical differences are a result of differences on handling the samples, and they can originate in any step of the proces: from decisions about sample collection and preservation [189], the Polymerase Chain Reaction (PCR) primers or sequencing platform of choice [96, 198], or even the laboratory in which the samples are processed [182]. These technical differences can overpower the underlying biological differences, making meta-analyses almost impossible. Thus, it is critical that this information is captured and made available at publication time, so other researchers can reproduce the results and/or decide if they can use the published data in a meta-analyses with their own samples.

The second challenge presented to a researcher wanting to perform a meta-analysis is collecting the data of the previously published results. Although current

publishers usually require the data to be deposited in a long-term repository such as the European Bioinformatics Institute's European Nucleotide Archive (EBI ENA), there are no mechanisms that ensure that the data deposited is valid or complete. Besides the DNA sequence data, the researcher must also track down the metadata describing the samples. Even when the metadata are publicly available, it may be incomplete and/or the specific encoding may not be clear for other users, requiring communication with the original authors to decode the information.

Finally, the researcher wanting to perform meta-analyses needs to combine and normalize his/her data with the previously published data to perform such meta-analyses. Although standards exist to represent sample metadata, such as the Minimum information about a marker gene sequence (MIMARKs) standard [225], they are not enforced or validated by the long-term repositories. This process of normalizing metadata is tedious and hard to automate, increasing the risk of introducing errors in the metadata which can alter the results of the meta-analyses. Apart from the metadata, the DNA sequence data itself may not be normalized. Sequence data available in the long-term repositories is not ensured to be the raw data. Preprocessing performed on those sequences, such as quality control, can introduce technical differences that affect the results of the meta-analyses and reduce ability to find biological differences.

Section 4.1 presents Qiita, a web-based service designed to alleviate the meta-analysis challenges by enforcing standards, requiring sample handling in-

formation, normalizing raw data representation and processing, and hosting over 150,000 samples publicly available from around the world. The material in section 4.1 is submitted for publication in *Nature Methods*. As co-first author of this publication, I have been involved in the design and implementation of the database, graphical user interface and plugin system of Qiita and contributed to writing the text.

Section 4.1, in part, has been submitted for publication of the material as it may appear in Nature Methods, 2018, A. Gonzalez, J. A. Navas-Molina, T. Kosciolek, D. McDonald, Y. Vazquez-Baeza, S. Janssen, A. D. Swafford, S. B. Orchanian, J. G. Sanders, J. Shorenstein, H. Holste, S. Petrus, A. Robbins-Pianka, C. J. Brislawn, M. Wang, J. R. Rideout, E. Bolyen, M. Dillon, J. G. Caporaso, P. C. Dorrestein, R. Knight.

4.1 Qiitas web-enabled platform accelerates microbiome meta-analyses from months to minutes

Multi-omic advances provide new insights into the function and composition of the microbial world one study at a time. However, to understand relationships across studies we must aggregate them into meta-analyses to identify features reproducible across biospecimens and data layers, and generate new hypothesis. Qiita dramatically accelerates such integration tasks in a web-based platform for the analysis and comparison of microbiome studies, demonstrated using the Human Microbiome Project and iHMP.

Recent years have seen exponential growth in studies that generate large quantities of microbiome and metabolome data, enabled by advances in high-throughput techniques [22]. Further advances in bioinformatics tools allow us to put these samples in the context of other studies, revolutionizing our picture of microbial diversity [197], and enabling useful insight into dysbiotic states relevant to human health [71]. In principle, the vast increase in available data should enable broader and more accurate insights into the diversity and functional impacts of the microbial world. However, these tools require increasing investments of time and effort by highly trained individuals: we now generate data faster than the few skilled experts can process it. Furthermore, the often idiosyncratic methods

employed by trained analysts can create new confounding variables that limit the power of meta-analyses, which would have ideally gained insight and statistical power by combining samples from many studies. Despite these challenges, meta-analyses of microbiomes have a rich history of success, identifying the major global drivers of diversity in microbial communities [120], characterizing the evolution of the vertebrate gut microbiome [109], and surveying specialized fields such as the built environment [1]. Meta-analyses also enable scientists to identify important biases such as DNA extraction, primers, or analytical pipelines [36, 122], which need to be controlled to generate biological discoveries.

To address these challenges, we developed Qiita, an open-source web-based platform that enables non-bioinformaticians to perform their own analyses and meta-analyses easily using standardized pipelines such as such as QIIME2 [20] and GNPS [212], accessed within a simple graphical user interface, starting with primary data and ending with statistical analyses and publication-quality figures. Not only does Qiita make curated high-quality data processing accessible to vastly more researchers, it ensures that the resulting information can be directly queried and compared, enabling rapid meta-analysis at otherwise impossible scales.

Meta-analyses typically involve tremendous effort, primarily due to three common issues. First, raw data (e.g., sequence data, spectra, study covariates, etc.) are frequently not open or completely accessible. In practice, a researcher must typically expend months of effort tracking down the data and covariates necessary for a meta-analysis [101]. Second, while there are common standards for sample

metadata (i.e., study covariates), such as Minimum Information about any (x) Sequence (MIxS) standards [225], the major sequence repositories do not enforce them, leading to varying degrees of use. Third, when authors of an existing work do make processed data (e.g., quality filtered sequence data, BIOM files, etc.) available, the files rarely contain details about the processing itself. Differences in sample or data processing can lead to technical differences that outweigh and obscure the biological differences in the data [36, 182]. Practically, this creates a high barrier to entry for novice and skilled researchers alike to analyze information that is ostensibly publically available.

Qiita alleviates these issues using the following strategies. First, users create studies that contain a description of the work; relevant publications; detailed metadata describing the collection and processing parameters for each sample; and relevant covariates, based on the MIxS standards [225], ensuring only administrator-reviewed standards-compliant metadata are loaded as public into the system. Users can thus keep data organized into discrete packages for comprehension and access by other users once they make their study public. Second, users must upload the rawest form of the data possible, typically multiplexed or demultiplexed FASTQ files generated from common sequencing platforms. Qiita can thus store and re-access the raw data as new pipelines and databases are adopted. Third, users select from a constrained set of processing parameters, which are subsequently retained with the data. This tracking and standardization ensures that newly processed data can be immediately compared to hundreds of thousands of samples already

in the database, and enables streamlined data deposition into ENA-EBI by automatically generating and submitting the necessary files (as has been performed now for 102,292 samples; Figure 4.1A). Finally, relevant samples for comparison in a meta-analysis can be quickly discovered via search of study title, metadata values, or even sequence data through the redbiom plugin [1], and quickly combined for analysis using a Qiime 2-based analysis plugin. When more specialized analyses are required, combined feature tables, metadata, and analytical artifacts (e.g. distance matrices, filtered subsets of samples, etc.) can be downloaded for use in other pipelines.

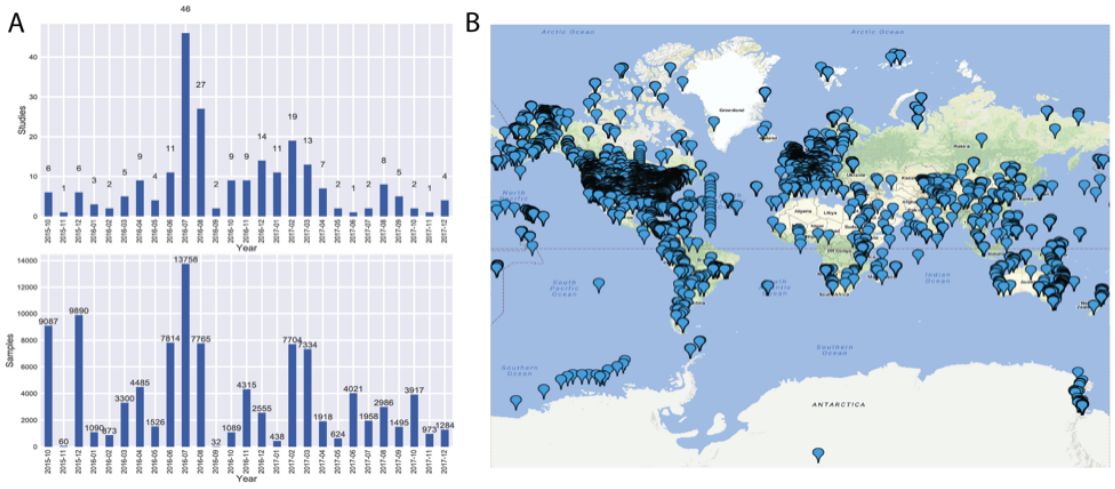


Figure 4.1: **Data loaded in Qiita and uploaded to EBI.** A. Monthly studies and sample depositions to EBI-ENA via Qiita. B. Geographical distribution of the samples present in Qiita.

To exemplify Qiitas meta-analysis utility, we tested the reproducibility of a study of how microbiomes of Inflamatory Bowel Disease (IBD) subtypes relate to those of healthy individuals [71]. We combined the 16S data from three stud-

[1] `https://github.com/biocore/redbiom`

ies of IBD-affected cohorts [71, 57] and iHMP, with the HMP1 study of healthy individuals [29] and another study samples of Clostridium difficile-affected patients that underwent a Fecal Matter Transplant (FMT) [216]. Using the web interface, we computed Unweighted UniFrac [118] and performed Principal Coordinates Analysis (PCoA). The plot shows the expected clustering via the body site of the sample (Figure 4.2A). However, when we examine only fecal samples (UBERON:feces category), we observe a pattern explained by sequencing platform as previously observed [122], Figure 4.2B. Restricting analysis to samples using the same sequencing platform (all but the HMP1 study), we observe clustering of the different IBD subtypes as previously reported [71, 57], Figure 4.2C. Using the feces-only distance matrix generated via the Qiita interface, we used QIIME2 to calculate the distance from each sample to a healthy plane [71], replicating the PCoA result across these independent studies. The samples from the C. difficile patients are also further from the healthy plane than those from the IBD subtypes, yet are much closer to the healthy plane after restoration of the microbiome via FMT, Figure 4.2D. This entire analysis took less than 5 minutes of person-time to perform, and did not require manual intervention once the processing pipeline was initiated until the use of the files offline in a Jupyter Notebook [2]. As this example demonstrates, Qiita is a powerful tool to combine and compare studies for meta-analyses and represents a significant advance for promoting facile data analysis within the microbiome research community.

[2] `https://github.com/knightlab-analyses/qiita-paper`

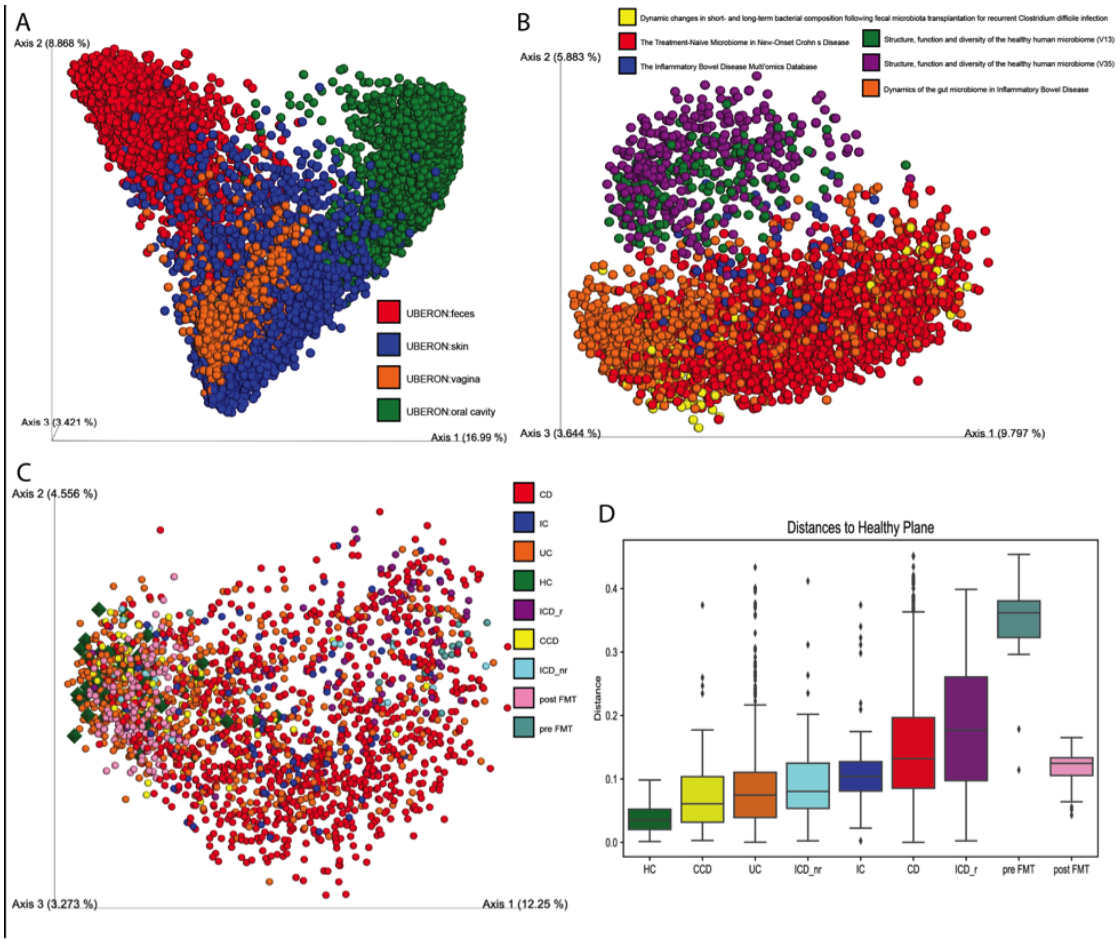


Figure 4.2: **Example Meta-Analysis in Qiita.** A. Unweighted UniFrac PCoA Meta-Analysis of three studies examining different IBD sub-types, C. difficile patients that underwent FMT, and the HMP1 and iHMP target gene data, where we can see strong clustering by body habitat. B. Only fecal samples from the same studies, showing separation due to wet-lab processing; in purple, yellow and orange HMP1 and iHMP samples. C. Removal of samples with different processing reproduces the PCoA resembling published results on the distribution of IBD sub-types (healthy samples as dark green diamonds). D. Calculating distances from a healthy plane [71], we can reproduce the results (which took several months to compile and compute originally), and even see how the distances from the patients with a C. difficile infection have larger distances before FMT and smaller afterwards.

By establishing an accessible path from annotated data to consistent and interoperable results, Qiita provides a practical way to harness the growth of sequence data for continuing value beyond its initial use by applying the living data concept [212] of ongoing reprocessing and annotation. Centralizing data storage and computation alleviates the substantial burden of independently maintaining compute resources. The web-based interface for Qiita allows users to avoid operating-system restrictions and obviates the need to train users to install, configure, and troubleshoot software via the command-line. To date, this resource hosts over 50TB of omics data from over 460,000 samples originating from studies that span the world, Figure 4.1B. More than 168,000 of these samples, including the entire recently released Earth Microbiome Project (EMP) [197] are public and immediately available for meta-analyses. As this collection grows, it will become increasingly important to improve the quality of associated metadata. Currently, Qiita requires MIxS-compliant metadata prior to making sequences public, ensuring that samples available for meta-analysis meet a minimum standard; additionally, expert-curated Gold studies with exceptional metadata are highlighted both as common references and to promote better practices in the community.

The power of this configuration has already been demonstrated through publications during the developmental stage of the platform and in our Qiita workshops, carried out regularly since early 2017 at UC San Diego Center for Microbiome Innovation [3]. Though not yet officially released, Qiita has already re-

[3] `http://cmi-workshop.readthedocs.io/en/latest/`

ceived over 100 citations in Google Scholar, and numerous publications have used samples from multiple studies in Qiita to perform meta-analyses. Additionally, custom instances of Qiita can be easily set up on virtual or physical machines to host specific datasets, as we have exemplified for the IBDMDB in the iHMP at http://ihmp.ucsd.edu/.

Qiita thus provides a unique resource allowing researchers to contextualize their data, perform meta-analyses across hundreds of studies and thousands of samples, and seamlessly deposit data into standards-compliant databases. Critically, a model of easy input, consistent output assures that time and effort spent analyzing each new study incrementally and usefully adds to the total resource. Qiita will thus revolutionize the pace of microbiome analyses and meta-analyses.

4.1.1 Online methods

Code design and availability

Qiita is designed using a three layer pattern: storage, logic, and interface. We describe each layer individually.

The storage layer design is a combination of a PostgreSQL 9.3.17 database and a structured filesystem. This approach allows Qiita to maintain referential integrity within and between studies, sample metadata, the analysis pipeline(s), and the commands executed over the different data types. However, the data volume is such that it can encumber a relational database, so the data (e.g., sequence

files, contingency tables etc.) are stored in standard formats (e.g FASTA, FASTQ, BIOM). The database maintains file path locations using indirection to allow files to reside on any number of filesystems. Additionally, this layer also stores the covariates (metadata) of each sample split in two main tables: a sample and a preparation information. The sample information are the covariates pertinent to the sample, while the preparation is how the sample was processed in the wet-lab and data generation (target gene sequencing, shotgun, metabolomic, etc).

The Qiita logic layer is written in Python using Object Oriented Programming, defining an object for each important element of the system. All data in Qiita are represented by an artifact object. An artifact represents a collection of files which reside on the filesystem, the logical types associated with each file, and a logical type of the artifact itself. Commands can specify which type of artifacts they accept as input and which type of artifacts they generate as output. The type of artifacts and the commands used to analyze artifacts are defined by Qiita plugins, which encapsulate the compute logic. Qiita defines two types of plugins: Qiita Type Plugins and Qiita Plugins. The Qiita Type Plugins define new artifact types, and is how data are imported into Qiita. A Qiita Type Plugin must define only two operations: Validate and Generate HTML summary. The Validate operation receives as input the set of files, and user associated types, for a new artifact and the preparation information and determines if the set of files defines a valid artifact for the given preparation. For example, in the case of a set of per-sample FASTQ files, the validator checks that each of the samples has a unique

file, and that the names of these files match those in the run_prefix column in the preparation information. The Generate HTML summary obtains the contents of an artifact and generates an HTML file summarizing the contents of such artifact. This summary provides a user-interpretable overview of the artifact, usually helpful enough to determine if something went wrong with the processing of the artifact. In contrast, the Qiita Plugin represents a collection of logically related commands (e.g., methods for constructing distance matrices). Each command within a Qiita Plugin accepts one or more artifacts as input, runtime parameters, and produces one or more artifacts as output. Each command execution is logged in the Qiita relational database, specifically, Qiita stores the plugin used, the command executed within the plugin, the artifacts provided as inputs, the parameters specified, and the artifacts generated.

The motivation for a modular plugin system is separation of concerns and encapsulation as each plugin runs in its own discrete environment and communicates with Qiita through an internal communication layer. This approach allows the plugins to be written in any programming language, with plugin specific dependencies, without introducing dependency conflicts with other plugins in the system. These environments are managed using plugin-specific conda environments. To facilitate the development of new Qiita plugins by external developers, we have created a Qiita client library [4] and two Cookiecutter (Qiita Type Plugin [5]

[4] `https://github.com/qiita-spots/qiita_client`
[5] `https://github.com/qiita-spots/qtp-template-cookiecutter`

and Qiita Plugin [6]) templates that set up the boilerplate code needed for an initial plugin repository and communication with Qiita.

The interface layer is a web-based interface accessible via Google Chrome, and that is powered from the server side via Tornado 3.1.1 [7]. The interface design and implementation has gone through multiple rounds of review, utilizing feedback kindly provided by users attending Qiita workshops.

The source code, and comprehensive test suite, for the Qiita package can be found in https://github.com/biocore/qiita. The source code for the officially supported Qiita plugins can be found under the qiita-spots GitHub organization at https://github.com/qiita-spots. All source code in the qiita repository and qiita-spots organization are BSD-licensed.

Data analysis

One of the most important items for a successful meta-analysis is consistency during the data processing. To achieve this consistency, Qiita processes all raw data with one of several standard parameter sets, based on the recommendations published in the literature. The parameters for demultiplexing and quality control the 16S rRNA gene sequences are based on the assessment performed Bokulich et al. [14], while the parameters for OTU picking are based on the recommendations provided in Navas-Molina et al [143]. In addition to OTU picking, Qiita also permits sub-OTU sequence clustering with Deblur [6]. In the

[6] `https://github.com/qiita-spots/qp-template-cookiecutter`
[7] `http://www.tornadoweb.org/`

deblur manuscript, the authors used more stringent quality control parameters from those outlined by Bokulich et al. [14].

Data availability

All data used is available via Qiita and EBI (where applicable). The Human Microbiome Project (HMP) and Integrative Human Microbiome Project (iHMP) data is available via the HMP Data Analysis and Coordination Center (DACC) [8]. Analytical steps for this paper can be found in [9]. Additionally, the Qiita Analysis can be found here [10], you must be log in to Qiita to access it.

[8] https://hmpdacc.org/
[9] https://github.com/knightlab-analyses/qiita-paper
[10] https://qiita.ucsd.edu/analysis/description/15093/

Chapter 5

Making meta-analysis accessible to the clinician

Section 1.1 introduced one of the most important challenges in microbiome research: translating the research results from the laboratory to everyday life, and in particular to human health. The human body is a complex ecosystem, hosting a wide variety of microorganisms that play key roles in our well-being. However, this aspect of the human body is generally ignored during routine doctor's visits.

One of the reasons why microbiome analyses are not routinely used in the clinic is because microbiome research is still in its infancy, and additional better designed studies on clinical cohorts are needed to identify microbiome-host interactions that can directly be applied to human health. Reproduction of these results, integration of multiple datasets and standardization of the data are key to achieving the quality results needed for clinical applications. Sections 2.1 and 4

presented techniques that address these challenges.

Another challenge for bringing microbiome analysis to the clinic is the time that it takes to perform a microbial community analysis. Typical microbiome analyses take from months to years to complete. However, the time to perform these analyses can be reduced by following Standard Operating Procedures (SOP) for sample handling and processing, using a standard analysis framework, optimizing analysis bottlenecks and employing an interdisciplinary team of experts to analyze the data. Section 5.1 shows how, using the work presented so far in this dissertation and the interaction of an interdisciplinary team of analysis experts, microbial community analyses can be performed in under 48 hours, a time frame short enough to provide useful information in a clinical setting.

The material in section 5.1 was published in *mSystems, 2013*. As a co-first author of this publication, I performed the fast initial 16S analysis, generated the first pass analysis for more in-depth analyses of the 16S data, directed the 16S analysis team, generated figures for the publication, and contributed to writing the text.

Section 5.1, in full, reproduces the material as it appears in "From sample to multi-omics conclusions in under 48 hours". R. A. Quinn, J. A. Navas-Molina, E. R. Hyde, S. J. Song, Y. Vazquez-Baeza, G. Humphrey, J. Gaffney, J. J. Minich, A. V. Melnik, J. Herschend, J. DeReus, A. Durant, R. J. Dutton, M. Khosroheidari, C. Green, R. da Silva, P. C. Dorrestein, R. Knight *mSystems*, 2016, DOI: 10.1128/mSystems.00038-16

5.1 From sample to multi-omics conclusions in under 48 hours

Multi-omics methods have greatly advanced our understanding of the biological organism and its microbial associates. However, they are not routinely used in clinical or industrial applications, due to the length of time required to generate and analyze omics data. Here, we applied a novel integrated omics pipeline for the analysis of human and environmental samples in under 48 h. Human subjects that ferment their own foods provided swab samples from skin, feces, oral cavity, fermented foods, and household surfaces to assess the impact of home food fermentation on their microbial and chemical ecology. These samples were analyzed with 16S rRNA gene sequencing, inferred gene function profiles, and liquid chromatography-tandem mass spectrometry (LC-MS/MS) metabolomics through the Qiita, PICRUSt, and GNPS pipelines, respectively. The human sample microbiomes clustered with the corresponding sample types in the American Gut Project [1], and the fermented food samples produced a separate cluster. The microbial communities of the household surfaces were primarily sourced from the fermented foods, and their consumption was associated with increased gut microbial diversity. Untargeted metabolomics revealed that human skin and fermented food samples had separate chemical ecologies and that stool was more similar to fermented foods than to other sample types. Metabolites from the fermented foods,

[1] `http://www.americangut.org`

including plant products such as procyanidin and pheophytin, were present in the skin and stool samples of the individuals consuming the foods. Some food metabolites were modified during digestion, and others were detected in stool intact. This study represents a first-of-its-kind analysis of multi-omics data that achieved time intervals matching those of classic microbiological culturing.

Importance Polymicrobial infections are difficult to diagnose due to the challenge in comprehensively cultivating the microbes present. Omics methods, such as 16S rRNA sequencing, metagenomics, and metabolomics, can provide a more complete picture of a microbial community and its metabolite production, without the biases and selectivity of microbial culture. However, these advanced methods have not been applied to clinical or industrial microbiology or other areas where complex microbial dysbioses require immediate intervention. The reason for this is the length of time required to generate and analyze omics data. Here, we describe the development and application of a pipeline for multi-omics data analysis in time frames matching those of the culture-based approaches often used for these applications. This study applied multi-omics methods effectively in clinically relevant time frames and sets a precedent toward their implementation in clinical medicine and industrial microbiology.

5.1.1 Introduction

The omics field is expanding rapidly, driven by the plummeting cost of DNA sequencing, the widespread availability of DNA sequencers and mass spectrome-

ters, and the seemingly unlimited breadth of its applications. However, generating, processing, analyzing, and interpreting the data typically takes months and requires substantial technical expertise in large multidisciplinary teams, in part, due to the rapidly evolving nature of the component techniques. The speed of mass spectrometry and nucleic acid sequencing (the tools required to generate omics data) has increased rapidly in the last decade, and they have separately been applied to clinical diagnostics in a targeted fashion. For example, high-throughput sequencing for the detection and typing of single pathogens in complex samples has achieved turnaround times of hours to days [167, 135, 142, 67, 159], and mass spectrometry analysis of metabolites has been performed in the clinic and laboratory in essentially real-time [12, 78]. However, the integration of multi-omics technologies and their application to the microbiome field have not yet achieved time frames compatible with clinical needs in human health, industrial microbiology, or routine laboratory experiments.

Multi-omics studies of the human microbiome can have enormous impact, providing a more comprehensive picture of a microbial community than a single omics approach on its own [54, 52]. These studies have led to an understanding of how microbial communities in our bodies produce metabolites that affect our health and transform the drugs we consume [214, 91, 70, 128]. One of the first integrated omics analysis related to the human microbiome was by Li et al. [110], who revealed an association between the gut microbiota and host metabolites in a cohort of Chinese subjects by using clone library sequencing and nuclear magnetic

resonance. This, and more recent multi-omics studies [184, 169], had multiyear gestation times. Today, when considering the time between receipt of samples with informed consent and statistical conclusions from integrated omics data, these studies still require months to years to complete.

In order to develop rapid multi-omics pipelines with broad applicability, they must first be tested using subjects and samples that are strongly influenced by their exposure to microbes and microbial chemical products. The subjects in this study are tightly linked to their microbial partners through their active involvement with fermented foods. This mutualistic relationship is believed to have existed since the Paleolithic era [132] and continues around the globe today. Modern human evolution is intertwined with the influence of microbial fermentation processes in the foods we eat and within our own bodies. Depending on the type of food and conditions used during fermentation, different types of microbial communities form, composed of various bacterial and fungal species [221], and the metabolic products of these communities can impact human health [203]. Previous studies found that species originating from microbially diverse fermented foods, such as cheese and salami, are able to colonize the gastrointestinal tract [203]. Furthermore, with the significant effects of antibiotics and a processed food-based diet on our microbiomes [128, 34, 202], there is an interest in the health benefits of fermented foods as alternatives. Here, we present the results from a simple, robust multi-omics platform integrating analyses of human, environmental, and animal samples in the clinically relevant time frame of less than 48 h. This pipeline is now

possible because of rapid advances in the development of software for the analysis and integration of omics data and standardized protocols that allow streamlined insertion of matched samples into multi-omics pipelines. We demonstrate how individuals commonly exposed to fermented foods show influences of these microbes on and in their bodies.

5.1.2 Results and Discussion

General description of the 48-h analysis and multi-omics pipeline. Samples were collected by seven volunteers (two families and two individuals, designated households 1 to 4) who regularly prepare and eat fermented foods and who were recruited to the American Gut Project (AGP) [2] via word-of-mouth through the Second Annual San Diego Fermentation Festival in San Diego, CA. The AGP is an IRB-approved citizen science project comprising more than 7,000 samples from more than 6,500 individuals. Consenting participants received an AGP sampling kit after they gave consent and took a survey online, and the data were stored in a secure database. The deidentified metadata were then immediately downloaded into a file formatted for use in Qiita [3]. Due to the infrastructure surrounding the process, participant consent and sample-associated metadata were obtained before the samples arrived in the laboratory, facilitating immediate preparation for sample processing upon arrival. Notably, the metadata can be used for both

[2] http://www.americangut.org
[3] https://qiita.ucsd.edu/

16S rRNA gene sequencing and metabolomics analyses, further streamlining the multi-omics approach. Samples were collected by cotton swab and subjected to DNA and metabolite extraction to describe the composition and activity of the corresponding microbial communities. Samples were subjected to a streamlined, high-throughput process involving preparation for 16S rRNA gene (variable region 4 [V4]) sequencing via the Earth Microbiome Project protocols [22, 86] and for liquid chromatography-tandem mass spectrometry (LC-MS/MS) [15]. The first description of both the microbial communities and molecules, including alpha and beta diversity, and specific effects of fermented foods on the microbial and chemical ecology of the subjects, occurred within 48 h after samples were delivered to the laboratory (Figure 5.1). Computational resources, including the Barnacle cluster available through the UCSD center for microbiome innovation connected to the Comet supercomputer located at the San Diego Supercomputer Center, allowed >50 central processing unit (CPU) h of processing in <11 h of wall time (note that some of the component steps are not parallelized), giving results back to the researchers fast enough to interpret the data in a timely manner.

There are four main components that enabled the development of this rapid multi-omics pipeline and its implementation in less than 48 h (Figure 5.1). First, subjects easily and efficiently enrolled themselves as part of an already existing, IRB-approved project (the AGP), enabling the use of on-the-spot informed consent and standardized metadata collection. Second, the protocols used to collect meta-

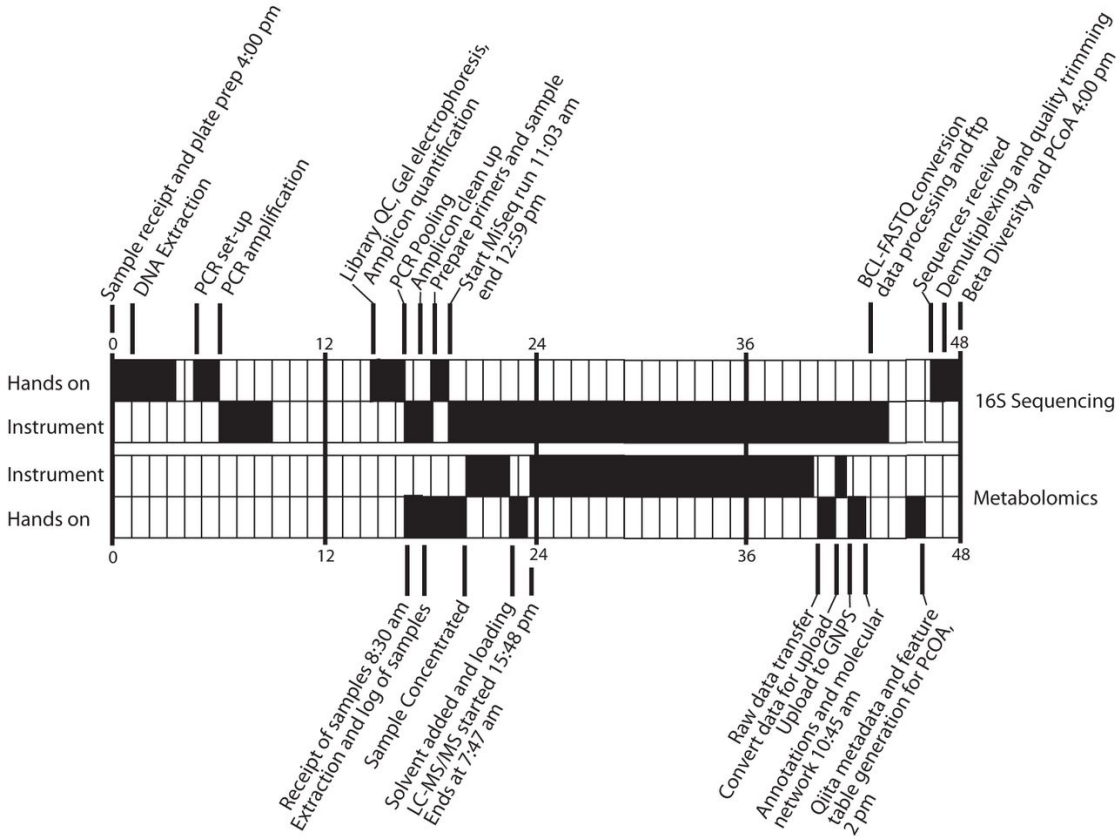


Figure 5.1: **Timeline of the multi-omics analysis of samples from four households and their fermented food products.**

data and process samples have been extensively benchmarked and standardized [4], allowing rapid assimilation with existing datasets and facilitating meaningful comparisons with other cohorts. Third, community analysis infrastructures, including Qiita, the microbial analysis infrastructure that houses microbiome analysis tools, and GNPS [5], a crowdsourced analysis infrastructure and public metabolomics knowledge repository, allowed rapid data processing and interpretation. And fourth, the servers that host Qiita and GNPS are linked, enabling normalization, processing, and cross-platform analysis of multi-omics data in an integrated fashion. Both these analysis platforms enable rapid comparisons to existing data in the public domain and are publicly available, facilitating data upload and analysis from any sequencer or tandem mass spectrometer, so long as the file formats are compatible. Linking the two platforms limits the need to move gigabytes or terabytes of data, making local analysis on ones own computer and integration with existing knowledge possible, rather than needing to download public data and new data to a personal computer first (e.g., the AGP data repository contains over 216 million reads). Tools available through this pipeline and utilized in this study include operational taxonomic unit (OTU) clustering of reads and generation of tables for multivariate statistical analysis of microbiome data, including alpha diversity, principle component analysis (PCoA) visualization through EMPeror, cluster significance testing with analysis of similarity (ANOSIM), and others. This pipeline

[4] `http://www.earthmicrobiome.org/emp-standard-protocols/`
[5] `http://gnps.ucsd.edu`

also allows immediate integration of data with the data in the AGP repository to visualize the relationships of samples with a large reference data set, which can provide context to the microbiome data generated. Metabolomics tools include library searching of the GNPS libraries (the largest currently available in the mass spectrometry field) [207], molecular network visualization to allow metabolite tracking, and metabolome abundance matrix generation to allow similar multivariate statistical analysis, including PCoA and EMPeror-based visualization of sample relationships.

Microbiome relationships Bacterial marker gene sequencing revealed rich microbial communities in most fermented food samples as judged by Faiths phylogenetic diversity (PD) metric [48], a biodiversity measure incorporating phylogenetic differences between the taxa present in a sample. The three most diverse samples were pickles, beet kvass, and port wine (PD values of 23.0, 16.6, and 16.2, respectively), while dairy kefir and symbiotic colony of bacteria and yeast (SCOBY) samples were the least diverse (average PD values of 2.21 and 1.91, respectively). The average PD of all fermented foods in the data set was 9.89, compared to 21.6, 11.9, and 18.5 for human skin, oral, and fecal samples, respectively. Surface microbiomes were also rich, with an average PD of 11.5. The unweighted UniFrac matrix [118] visualized via principle component analysis (PCoA) using EMPeror clustered the samples closely by type (ANOSIM R statistic = 0.477, P = 0.001), and the human sample types matched their corresponding AGP sample types (Figure 5.2a). While mouth, stool, and right and left hand samples each

formed relatively tight clusters, as expected [29], fermented food and indoor surface samples formed a looser cluster together, largely distinct from human sample clusters, although a few food and surface samples clustered near hand and fecal samples (Figure 5.2a). Combining these samples with a subset of the AGP cohort revealed that there was an increase in gut bacterial diversity that correlated with an increase in fermented food consumption ($R2 = 0.034$, $P = 0.02373$) (Figure 5.2b). Nonparametric Kruskal-Wallis tests corrected for multiple comparisons (false discovery rate [FDR]) identified 219 OTUs differing significantly in relative abundance across sample types. No OTU was significantly higher in fermented food samples than in any other sample type, though several were higher (FDR corrected $P < 0.05$) in stool (including OTUs classified as Blautia, Varibaculum, Bacteroides, Peptoniphilus, and Corynebacterium), hand (Corynebacterium, Staphylococcus, Neisseria, Haemophilus, and Rothia), and mouth (Prevotella, Neisseria, Lautropia, and Leptotrichia) samples. SourceTracker [88] analysis revealed that the microbial communities of items on or in which fermented foods were prepared (i.e., from surfaces, such as cutting boards, to containers, such as fermenters) were largely sourced from the foods and specific to the location in which the foods were prepared. Except for one household, where small percentages (9 to 30%) of hand microbial communities were sourced from food, no obvious patterns linked microbial source communities to human skin, mouth, or fecal microbiomes (Figure 5.2c).

PICRUSt metagenome predictions revealed a slightly dissimilar clustering

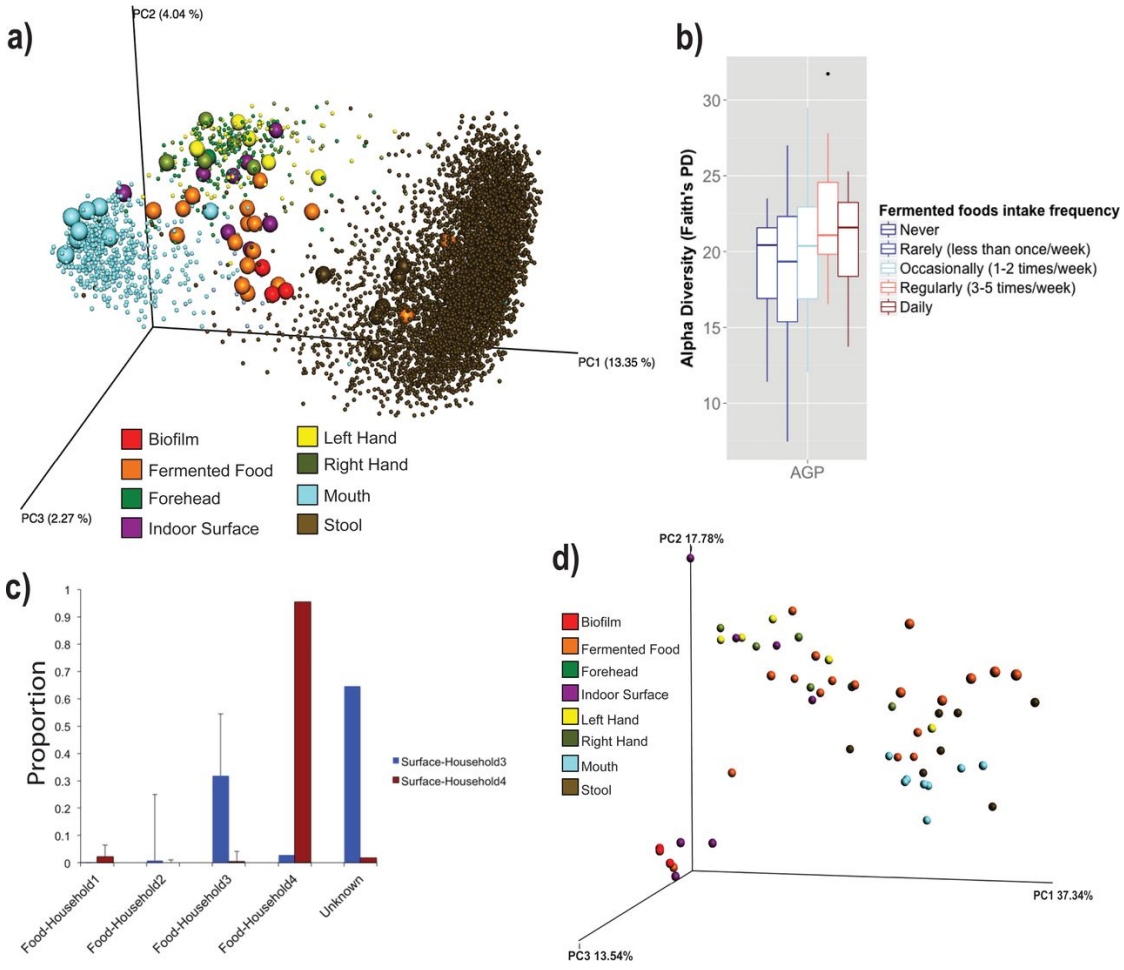


Figure 5.2: **Marker Gene results.** (a) PCoA of the abundance of unique OTUs per sample from the 16S marker gene sequencing data from the AGP data repository (small spheres) and the San Diego Fermentation Festival volunteer samples collected for this study (large spheres). (b) Alpha diversity as measured using 16S rRNA marker gene sequencing counts of OTUs in a subset of the American Gut Project data for which consumption of fermented foods is reported. (c) SourceTracker analysis of surface samples from households 3 and 4. SourceTracker measures the proportions of OTUs sourced from the fermented foods on the household surfaces where they were prepared. (d) PCoA clustering of microbiome data after metagenomic prediction with the PICRUSt algorithm.

pattern to that observed with 16S marker gene sequencing data based on sample type when the Bray-Curtis distance metric was applied to the BIOM table containing KEGG pathways. While fermented food and surface samples still formed a loose cluster, with body types more tightly clustered, oral samples clustered close to fecal samples based on KEGG pathways but not 16S marker gene data (Figure 5.2d). Nonparametric Kruskal-Wallis tests corrected for multiple comparisons (FDR) identified 119 KEGG pathways differing significantly across sample types. KEGG pathways that were significantly higher (FDR-corrected P value of <0.05) in fermented foods than on surfaces included aminosugar and nucleotide sugar metabolism, starch and sucrose metabolism, galactose metabolism, RNA transport, glycolysis/gluconeogenesis, and methane metabolism; KEGG pathways that were significantly higher on surface samples than in food samples included bacterial secretion systems, phenylalanine metabolism, fluorobenzoate degradation, aminobenzoate degradation, glycan biosynthesis and metabolism, tryptophan metabolism, and caprolactam degradation. Several KEGG pathways were also differentially abundant between fermented foods and stool or mouth samples. For example, aminobenzoate degradation, retinol metabolism, naphthalene degradation, ethylbenzene degradation, tyrosine metabolism, and butanoate metabolism pathways were all significantly higher (FDR-corrected P value of <0.05) in fermented food samples than in stool samples, while glycosaminoglycan degradation, other glycan degradation, methane metabolism, transcription machinery, sporulation, sphingolipid metabolism, and sporulation pathways were significantly higher

in stool samples than in fermented food samples. In mouth samples, n-glycan biosynthesis, translation factors and proteins, amino acid-related proteins, and lipopolysaccharide biosynthesis and biosynthesis proteins were significantly (FDR-corrected P value of ¡0.05) higher than in fermented food samples. Conversely, chloroalkane degradation, ethylbenzene degradation, aminobenzoate degradation, tyrosine metabolism, bisphenol degradation, naphthalene degradation, benzoate degradation, xylene degradation, butanoate metabolism, and several other pathways were significantly higher in fermented food samples than in mouth samples.

Metabolome relationships. PCoA of Bray-Curtis distances for the presence/absence of metabolites by sample showed that skin and mouth samples were distinct from other sample types and that fermented food samples clustered with biofilm samples from their containers (Figure 5.3). Stool samples, however, were mixed with other sample types, unlike the tight clustering seen using the 16S rRNA sequencing data (Figure 5.3). These clustering relationships showed that the chemistry of fermented foods and their associated human and environmental samples was more variable than the microbial profiles among sample types, likely due to the dynamic nature of metabolite production from microbial communities and the direct input of the foods themselves in stool chemistry.

Of the 7,425 unique MS/MS spectra detected, 100 were matched to reference libraries using GNPS molecular networking [215, 223]. This 1.3% match rate is similar to the 1.8% match rates for all metabolomics data in GNPS [32]. Most spectral matches were plant natural products associated with the fermented foods,

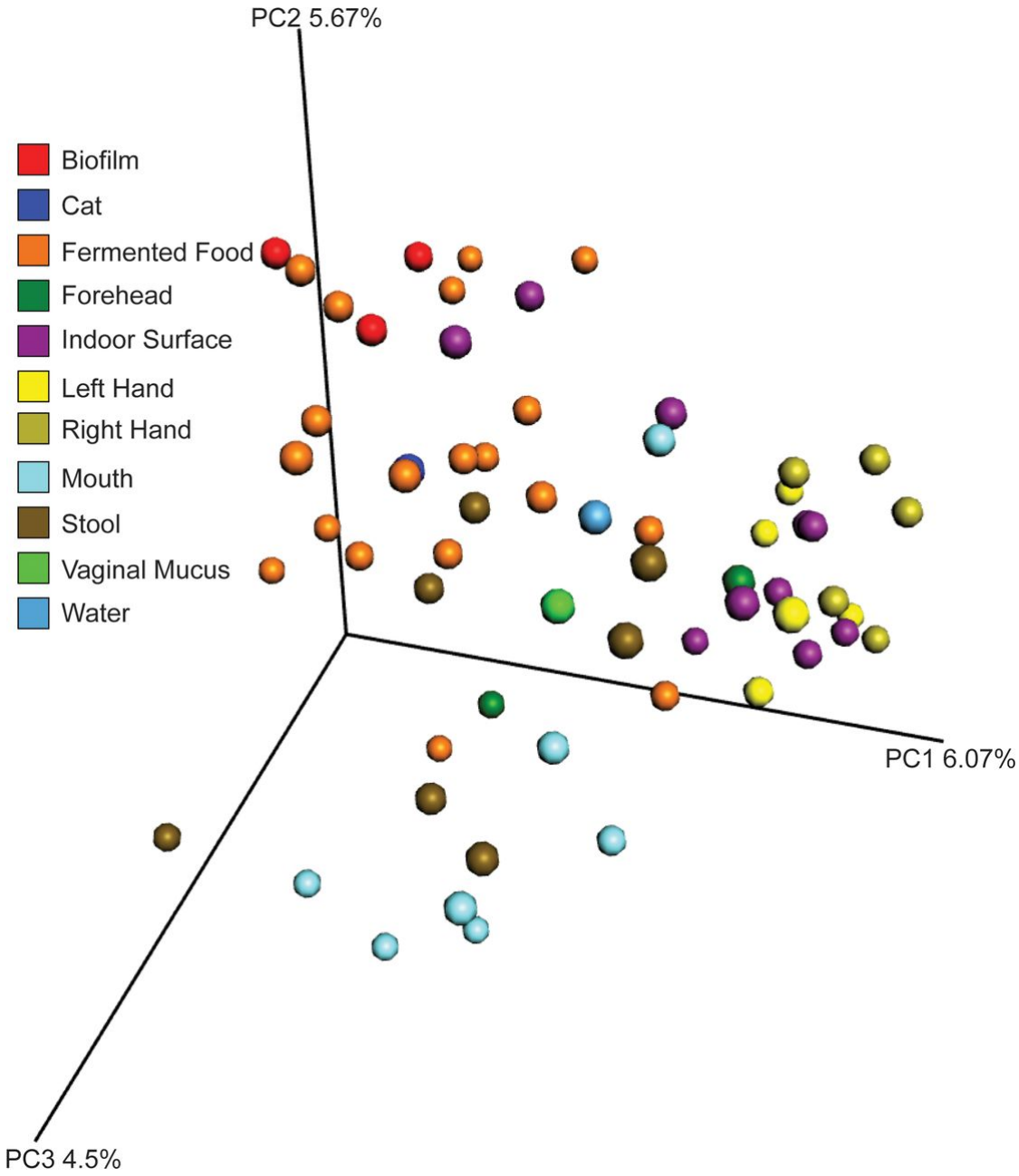


Figure 5.3: **PCoA of the metabolomics data from a presence/absence matrix of unique MS/MS spectra in all samples using the Bray-Curtis distance metric.**

including flavonoids, lipids, and plant sterols. Other, non-plant-related molecules were observed, including cholesterol and its derivatives on skin and avobenzone, an active ingredient in sunscreen. Gingerol, the spicy flavorant in the ginger root Zingiber officinale), was found in samples of fermented foods and the indoor surfaces of two households. Similarly, the spicy pepper plant (Piper nigrum) alkaloid piperine was found in fermented food, stool, indoor surface, and skin samples. The metabolite polanrazine B, isolated from Leptosphaeria maculans, a fungal pathogen of canola and rapeseed plants (Brassica spp.) [192], was prevalent in two of the four households sampled, including in food and stool samples. Spectral matching also identified the flavonoid procyanidin B2 (m/z 579.149), an antioxidant associated with many plants, such as apples, beans, grapes, and tea, and molecular networking detected an altered form with an additional pentose sugar (neutral loss of m/z 132.04 [156] [Figure 5.4a]). Procyanidin B2 was present in the biofilm, fermented food, indoor surface, human skin, and stool samples. This metabolite was present in all sample types from a single subject, including the foods the person ate, surfaces in the household, the persons body, and stool (Figure 5.4b). Although fermented foods from all four households contained procyanidin B2, only two of them had this molecule in their stool, indicating differential metabolism in different individuals. The modified form of procyanidin (m/z 711.189) was found in the same sample types except stool, suggesting that consumption of this metabolite from a fermented food resulted in removal of the sugar or the absorption of the molecule as it passed the digestive tract. Pheo-

phytin A, chlorophyll a without its metal ion, was only detected in samples of fermented foods of vegetable origin (except beer), their containers, and stool, indicating that this molecule remained intact through digestion (Figure 5.4a and b). Related metabolites, including bacteriopheophytin and pyropheophytin, were detected only in kimchi (Figure 5.4a). In sum, analysis of metabolites from human samples revealed molecules from fermented foods modified by human or microbial enzymes, molecules produced by organisms pathogenic for components of the fermented food, molecules from fermented food that passed completely through the volunteers digestive tracts without alteration, and differential metabolism of fermented food metabolites in different people.

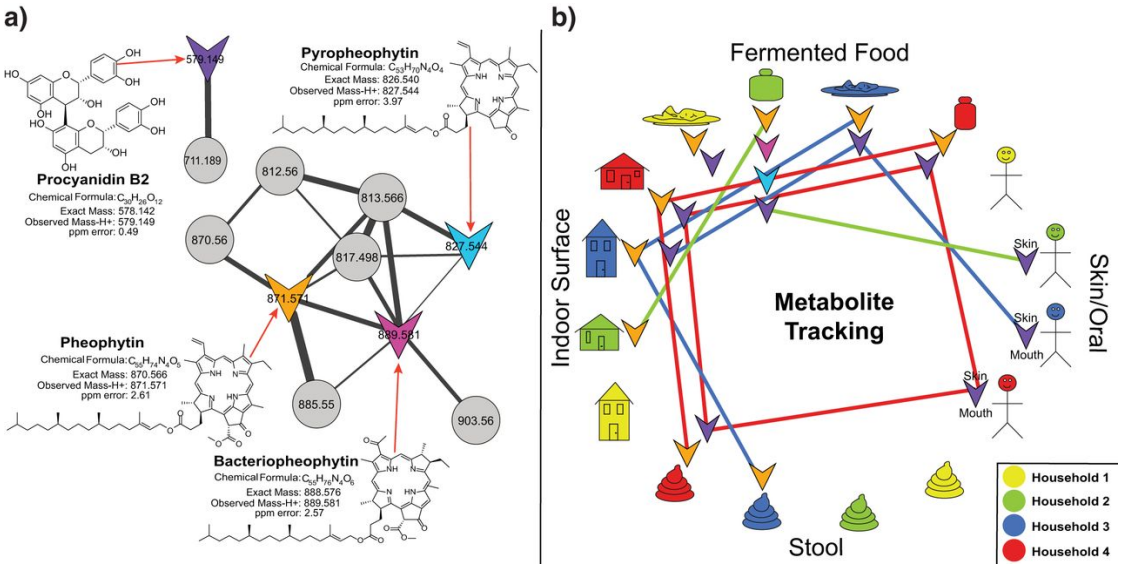


Figure 5.4: **Metabolomics results.** (a) Molecular network clusters of pheophytin and procyanidin and their related metabolites. (b) Metabolite tracking for the presence of those metabolites in the human and environmental samples from the four separate households sampled. Metabolites from network clusters, colored as in panel a, are shown next to the household samples they were detected in, and colored lines are used to visualize tracking of metabolites through the specific households as shown in the key.

Microbiome and metabolome integration. Using Procrustes analysis [206] to get an integrated look at metabolome and microbiome relationships, we mapped the principal coordinate analysis matrices of the 16S rRNA data to the metabolomics data. The overall patterns matched, except that two samples (kombucha and pickles) clustered with fecal microbiome samples in the microbiome space but with other fermented foods in the metabolomics space (Figure 5.5). These results underscore that microbial communities and their activities are environment specific and that the metabolite output of the sample type is consistent with the microbial community that produced it.

Conclusions. Rather than multi-omics analysis being an arduous and highly technical procedure, this study demonstrates that it can be performed on a rapid time scale with a small team of people (six authors of the manuscript contributed to data analysis). A major advantage to this pipeline is the ability to compare data to large data repositories, such as the AGP and GNPS, for sample relationships and metabolite identification. This more easily facilitates the identification of microbiome dysbiosis or metabolome changes that indicate disease. Context is required in any clinical or industrial application of multi-omics data, to better determine how the current structure of a microbial community compares to previous states or sample types, enabling diagnosis of an active dysbiosis. The present study focused on fermented foods and their effects on the people who prepared and consumed them. These foods are of enormous medical importance given that yogurt, a fermented food, is the single food most correlated epidemiologically

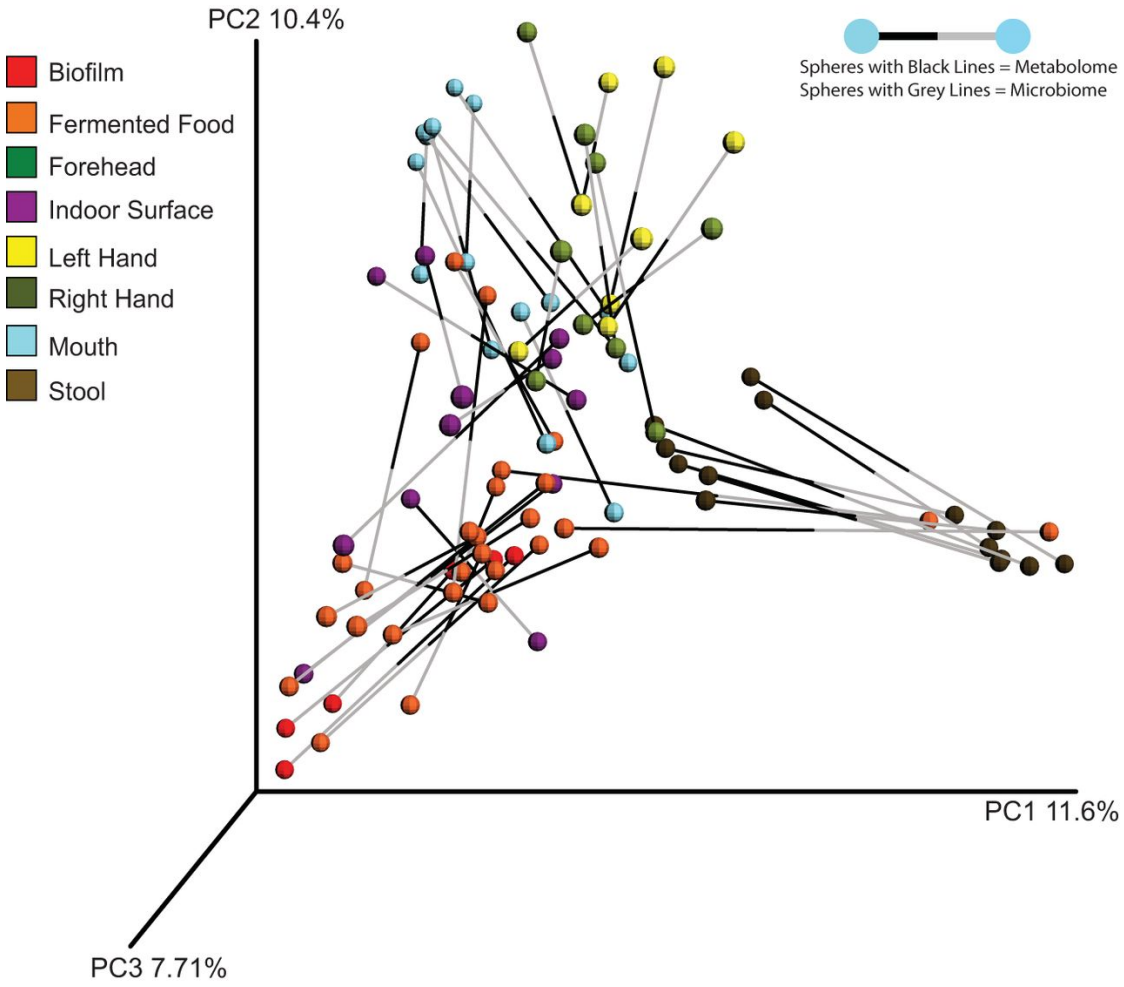


Figure 5.5: **Procrustes analysis of microbiome and metabolome data.** Spheres represent individual samples, and they are shown to be either metabolome or microbiome samples by being connected to a grey line or black line, respectively. Connections between the spheres represent microbiomes and metabolomes from the same sample and the distance between them.

with weight loss in the U.S. population [139], and they are of economic importance due to the billions of dollars per year that fermented foods contribute to the economy. Although this sample cohort did not require rapid data analysis, such as that required in a medical emergency or the potential loss of a large industrial fermentation, this study shows that consent could be obtained, samples collected, and data generated on microbiome-related samples collected from people located up to 100 miles away from the laboratory in a time frame matching that of classic microbiological culturing of common pathogens (approximately 2 days). The ability to do rapid-response multi-omics analysis and systems biology will have far reaching implications, from monitoring industrial fermentation processes, to guiding oil and gas drilling and fracking decisions, to providing rapid molecular analysis for patient care in infectious diseases and guiding the use of microbiome-based therapies, such as fecal microbiota transplant (FMT) [173] and probiotics. The combination of standardized protocols for subject recruitment and consent, sample collection, metadata capture, DNA sequencing, mass spectrometry, molecular networking, and data analysis and visualization now puts this technology in the hands of a broad spectrum of users. Broader and more rapid use of multi-omics methods will begin a sea change towards their implementation in clinical medicine.

5.1.3 Materials and methods

Participant recruitment and sample collection. For the first application of the pipeline, we chose a situation that, while time sensitive, was not

necessary for clinical decisions. All participants are members of a local fermenters club and ferment at home or operate a fermented food business; they learned about the study through the fermenters club. Participants willing to sample their own bodies, their fermented foods, and the surfaces that their foods are prepared on or in (i.e., kitchen counters, cutting boards, and fermenters) consented to be a part of the American Gut Project (AGP), the largest crowd-sourced, crowd-funded citizen science project in existence today. A total of seven people (two families and two individuals, designated households 1 to 4) received barcoded, dual-headed sterile cotton sampling swabs (BD Swube; Becton, Dickinson and Company, Franklin Lakes, NJ) and were instructed to sample their skin (right and left hands), mouths, stool, their fermented foods, and the surfaces touched by those foods. Some participants chose to sample alternative body sites (i.e., vagina and forehead), and one participant sampled the mouth of a pet cat. The food samples collected included beer, port wine, pickled cucumbers, pickled jalapenos, cottage cheese, curtido, kefir, kimchi, sauerkraut, miso, beet kvass, and fermented soda. The surface samples collected included cutting boards, countertops, refrigerator surfaces, skillets, kegerator parts, and fermentor parts. Samples were collected by subjects on 25, 26, and 27 January 2016, with the first sample in the data set collected at 8:05 a.m. on 25 January and the last sample in the data set collected at 12:05 p.m. on 27 January, for a total of 61 samples. Samples from six participants were delivered by hand to the laboratory, while one participant mailed their samples to the laboratory via overnight priority mail (FedEx). All samples were received in the

laboratory by 1:07 p.m. on 27 January 2016 (Figure 5.1). Upon arrival, one swab head from each dual-headed swab was immediately placed into a MoBio PowerSoil DNA extraction kit bead plate (MoBio, Inc., Carlsbad, CA) for bacterial DNA extraction. The second swab head was stored overnight at 20°C before preparation for metabolomics analysis using mass spectrometry.

Bacterial DNA extraction and generation of 16S rRNA V4 amplicons. Bacterial genomic DNA extraction, 16S rRNA gene variable region 4 (V4) amplicon generation, and amplicon preparation for sequencing were performed according to protocols benchmarked for the Earth Microbiome Project (EMP) that can be found on the EMP website [6]. Briefly, bacterial genomic DNA was extracted from samples using the PowerMag DNA isolation kit optimized for KingFisher (Mo Bio Laboratories, Carlsbad, CA), and then the V4 region of the 16S rRNA gene was amplified in triplicate from each sample and combined as follows. The PCR mixtures contained 13 μl Mo Bio PCR water, 10 μl 5 Prime HotMasterMix, 0.5 μl each of the barcoded forward and reverse primers (515f and 806rB; 10 μM final concentration), and 1.0 μl genomic DNA. The reaction mixtures were held at 94°C for 3 min (denaturation), with amplification proceeding for 35 cycles at 94°C for 45 s, 50°C for 60 s, and 72°C for 90 s, followed by a final extension for 10 min at 72°C. After amplification, the DNA concentration was quantified using PicoGreen double-stranded DNA (dsDNA) reagent in 10 mM Tris buffer (pH 8.0). A composite sample for sequencing was created by combining equimolar ratios of

[6] http://www.earthmicrobiome.org/emp-standard-protocols/

amplicons from the individual samples, followed by ethanol precipitation to remove any remaining contaminants and PCR artifacts.

16S rRNA marker gene sequencing. Pooled amplicons were sequenced at the Institute for Genomic Medicine at the University of California, San Diego, using the Illumina MiSeq platform. The library concentration was measured using the HiSens Qubit dsDNA HS assay kit (Thermo Fisher Scientific). A total of 6 pM of 16S library combined with 0.9 pM (15%) PhiX sequencing control version 3 was sequenced with 150-bp paired-end (PE) reads on an Illumina MiSeq sequencing system using a MiSeq reagent kit version 2 (300 cycle). Fastq files for reads 1 and 2 and the index read were generated using the BCL-to-FASTQ file converter bcl2fastq version 2.17.1.14 (Illumina, Inc.).

16S rRNA marker gene data analysis. Sequencing data were prepared and analyzed using the online tool Qiita [7] and the QIIME pipeline [20] version 1.9. Illumina read 1 was quality filtered and demultiplexed according to the QIIME default parameters, as follows: no ambiguous bases allowed, only one bar code mismatch allowed, and a minimum required Phred quality score of 3. Quality filtering resulted in 6,830,655 high-quality reads, with the average number of sequences per sample being 84,329. Quality-filtered sequences were clustered using the closed-reference OTU picking workflow against the August 2013 release of the Greengenes database [37], with a sequence identity of 97% and sortmeRNA [92] as the underlying clustering algorithm. After OTU picking, 5 samples (fore-

[7] `https://qiita.microbio.me`

head, water, vaginal, fermented grape soda, and fermenter inner wall samples) were removed from the data set because they had sequence counts lower than the rarefaction cutoff (2,053 sequences per sample); thus, a total of 54 microbiome samples were included in downstream analyses.

The AGP team has identified a group of bacterial bloom sequences that increase during sample transit back to the laboratory, and in order to avoid a study bias, those sequences were filtered out of the data (code available at [8]). To facilitate direct comparisons and reduce study bias between data obtained from the fermentation cohort and the AGP cohort, fermentation cohort stool sample data were also filtered for blooms.

Five of the seven fecal samples from the fermentation cohort passed quality and sequencing depth filtering. The bacterial diversity levels observed in these five samples were compared to those in a subset of 122 randomly selected fecal samples from other AGP participants of a similar age group for whom data on the frequency of fermented food intake were available. Alpha diversity (measured as Faiths phylogenetic diversity [48]) was calculated for each sample from a rarefied OTU table of 2,053 sequences per sample. Barplots were generated in R [9] to visualize the distribution of diversity values across the various groups, and a linear regression model was fitted to the AGP portion of the data.

We used SourceTracker [88], a tool that uses a Bayesian model jointly with

[8] https://github.com/biocore/American-Gut/blob/master/ipynb/primary-processing/02-filter_sequences_for_blooms.md

[9] https://www.r-project.org/

Gibbs sampling to quantify the amount of taxa that a set of source environments contributes to a sink environment, to determine the proportions of human and surface microbes that were sourced from fermented food microbiomes. Fermented food samples were designated sources, while human and surface samples were designated sinks.

Statistical analyses were applied to determine the significance of groups by sample type on the PCoA plot (ANOSIM, 999 Monte Carlo permutations) and to identify OTUs with significantly different relative abundances (Kruskal-Wallis, 999 Monte Carlo permutations) across sample groups. Nonparametric tests were used to appropriately deal with microbiome data, which were not normally distributed. The significance cutoff for P values (ANOSIM) and FDR-corrected P values (Kruskal-Wallis) was set at 0.05.

PICRUSt metagenome predictions were performed using the Galaxy implementation of PICRUSt 1.0.0 [100]. The resulting BIOM table was then categorized by KEGG pathways (i.e., KEGG Orthology groups [KOs] were placed into functional categories). All eukaryote-specific pathways were removed from the table, and the table was rarefied to 572,338. The Bray-Curtis distance metric was then applied and visualized using EMPeror [206]. A Kruskall-Wallis test with 999 Monte Carlo permutations was applied to determine significant differences in KEGG pathway abundances between groups of samples.

Metabolomics data analysis. The metabolomics data for this project

are available under MassIVE data set ID MSV000079485 at GNPS [10]. To generate metabolomes, the swabs were added to a solution of 70% methanol in water and allowed to extract for 2 h at room temperature. The methanol extract was then dried down in a centrifugal evaporator and redissolved in 100% methanol. Samples were transferred into 2-ml vials with inserts and diluted 1:2. MS analysis was performed on a QExactive (Thermo Scientific) mass spectrometer with a heated electrospray ionization (HESI-II) probe source, controlled by Xcalibur 3.0 software. MS spectra were acquired in positive ion mode over a mass range of 100 to 1,500 m/z. An external calibration with Pierce LTQ Velos electrospray ionization (ESI) positive ion calibration solution (Thermo Scientific) was performed prior to data acquisition, with an error rate of less than 1 ppm. The following probe settings were used for flow aspiration and ionization: spray voltage of 3,500 V, sheath gas (N2) pressure of 53 lb/in^2, auxiliary gas (N2) pressure of 14 lb/in^2, ion source temperature of 270°C, S-lens radio frequency (RF) level of 50 Hz, and auxiliary gas heater temperature at 440°C. Data acquisition parameters were as follows. Minutes 0 to 0.5 were sent to waste. Minutes 0.5 to 12 were recorded with data-dependent MS/MS acquisition mode. Full scan at MS1 level was performed with resolution of 35,000 in profile mode. The 10 most intense ions with 1 m/z isolation window per MS1 scan were selected and subjected to normalized collision-induced dissociation with 30 eV. MS2 scans were performed at 17,500 resolution with maximum injection time of 60 ms in profile mode. The MS/MS active exclusion parame-

[10] `http://gnps.ucsd.edu`

ter was set to 5.0 s. The injected samples were chromatographically separated using a Vanquish ultrahigh-performance liquid chromatography (UHPLC) instrument (Thermo Scientific) controlled by Thermo SII for Xcalibur software (Thermo Scientific), with a 30- by 2.1-mm, 2.6 μM, C18, 100-A Kinetex chromatography column (Phenomenex) with 40°C column temperature, 0.5 ml/min flow rate, mobile phase A consisting of 99.9% water (LC-MS grade; J.T. Baker)0.1% formic acid (Fisher Scientific, Optima LC/MS), and mobile phase B consisting of 99.9% acetonitrile (LC-MS grade; J.T. Baker)0.1% formic acid (Fisher Scientific, Optima LC/MS), using the following gradient: 0 to 1 min, 5% B; 1 to 8 min, 100% B; 8 to 10.9 min, 100% B; 10.9 to 11 min, 5% A; and 11 to 12 min, 5% B. Raw data files were converted to the .mzXML format using ProteoWizard [11] and uploaded to the GNPS-MassIVE mass spectrometry database. The list of annotations from the search can be found at [12].

Molecular networking was performed to identify spectra shared between different sample types and to identify known molecules in the data set. All annotations are at level 2 according to the proposed minimum standards in metabolomics [194]. The molecular networking parameters were as follows: a minimum matched-peak threshold of 4, a cosine similarity score cutoff of 0.65, a minimum cluster size of 2, and a parent and ion tolerance of 0.5 Da. GNPS library search parameters were the same except that a cosine threshold of 0.7 was used. A feature table of

[11] `http://proteowizard.sourceforge.net/`

[12] `http://gnps.ucsd.edu/ProteoSAFe/result.jsp?task=efc4f1031f73471cbdfddcde0cc\181a6&view=view_all_annotations_DB`

metabolite presence and absence in each sample was generated from GNPS spectral alignments and downloaded. Similarity of metabolomes was determined using the Bray-Curtis distance metric, projected with principal coordinate analysis and visualized with EMPeror through the in-house tool ClusterApp. Molecular networks were visualized and mined using the Cytoscape software [180].

16S-metabolomics multivariate comparisons. Using the OTU table and the metabolite table, we generated a distance matrix for each, using unweighted UniFrac for 16S and Bray-Curtis for the metabolomics. We performed principal coordinate analysis on the two matrices separately and used Procrustes analysis as implemented in QIIME 1.9.1 to rotate, translate, and scale the matrices. The resulting transformed matrices were plotted using EMPeror [206].

Microarray data accession numbers. Mapping files and preprocessed data for human samples are available at https://qiita.ucsd.edu under Qiita study identification number (ID) 10317 (AGP), and sequences are publicly available in EMBL-EBI (accession number ERP012803) under accession numbers ERS1048817, ERS1048818, ERS1048819, ERS1048820, ERS1048821, ERS1048822, ERS1048823, ERS1048824, ERS1048825, ERS1048826, ERS1048827, ERS1048828, ERS1048829, ERS1048832, ERS1048833, ERS1048834, ERS1048835, ERS1048836, ERS1048837, ERS1048838, ERS1048839, ERS1048840, ERS1048841, ERS1048842, ERS1048843, ERS1048844, and ERS1048845. Mapping files and preprocessed data for food, environment, and cat samples are available at https://qiita.ucsd.edu under Qiita study ID 10395, and sequences are publicly available in EMBL-EBI (accession

number ERP015077). The 16S amplicon analyses outlined in this paper were conducted using the Knight laboratorys supercomputer Barnacle, using 26 CPU hours.

5.1.4 Acknowledgments

We acknowledge the Sloan foundation for funding the work on metabolomics and the microbiome of the human habitat and the development of strategies to integrate GNPS and Qiita, NIJ for support for the use of metabolomics as a way to determine lifestyle signature analysis, Lee Stein for supporting the development of the rapid response microbiome program. The National Science Foundation award 1341698 and the Extreme Science and Engineering Discovery Environment (XSEDE, grant no. ACI-1053575) contributed computational resources. Nonfinancial or indirect financial support was provided by the San Diego Fermentation Festival sponsors, American Gut Project for providing data prior to publication, and Kristen Jepsen at the Institute for Genomic Medicine Genomics Center for support for the sequencing.

Chapter 6

Conclusions

As I described in Chapter 1, improvements in current technology are pushing microbiome science to become a "Big Data" field. Big Data is being generated in multiple ways. First, advances in sequencing technologies can generate four orders of magnitude more data than 10 years ago. Second, the need for different 'omics technologies to study different aspects of the system requires aggregation of even more data per sample than ever before. Finally, the complex interactions of the microbes with their niche require an accurate description of their niche, represented in the form of sample metadata. This rapid increase of data volume and heterogeneity presents a wide range of challenges to investigators, many of whom, due to the various conditions and system for which the microbiome is important, are not microbiologists by training or do not have extensive background in microbial ecology, or in the tools that can be used to analyze the data. Some of these tools can prove intimidating to those with no background in computer science.

This thesis provides solutions to some of these challenges, specifically by improving usability of analysis tools and access to resources required to run those tools, and by providing solutions to standardize data handling, storage, and analysis.

6.1 Improving usability of analysis tools

Of the tools available to perform microbiome analyses, Quantitative Insights into Microbial Ecology (QIIME) is a choice popular in the community. QIIME is a collection of command line scripts that can be challenging to use for investigators who are not familiar with the Command Line Interface (CLI). Section 2.1 described the first gold standard approach for microbiome data analysis, starting from sample preparation and going through to publication-quality figures. This section is a step-by-step guide, so even researchers who lack CLI familiarity can successfully perform their analyses.

A CLI is prone to user error. A simple typographical error can make the script fail or generate undesired results. To assess this issue (among many others), Qiita (Section 4.1) was presented as a web-based solution that allows command execution with a Graphical User Interface (GUI), minimizing the amount of input provided by the user and removing typographical errors.

Tool developers typically focus on solving a complex problem and making their tool available to the community as soon as possible. Developing a GUI doubles the development time [141], a cost that developers do not want to incur in

a fast changing field like microbiome research. Qiita builds a bridge between the developers and biologists, freeing up the developers from building a GUI, but still providing an intuitive GUI for non command line-savvy researchers. Tool developers just provide information about the inputs and outputs of their commands and Qiita automatically generates a web-based GUI.

This ability to make new command line tools rapidly available to non command line savvy researchers will push microbiome research forward faster than ever before. The usual steep learning curve for a new tool gets completely removed, because all tools in Qiita are based on the same GUI, and the researcher can focus on the science of their results rather than on the specifics of a new CLI. This will enable researchers to perform all their multi-omics analyses in a single platform, with a common user interface, facilitating advances in multi-omics analyses likely to be critical for making the microbiome an integral part of precision medicine. For example, mass spectrometry analyses have historically been able to be done only by those with specific training. Leveraging the Qiita plugin system will bring this type of specialized data analysis to the researcher, who can then combine mass spectrometry data with other data types, such as microbiome sequence data (both marker gene and whole genome shotgun), host genome sequence data, and proteomics data, among others. The potential for providing a more complete picture than ever before possible places Qiita at the forefront of techniques that facilitate the future of microbiome multi-omics research.

6.2 Improving resource utilization of analysis tools

Section 2.2 identified one of the common bottlenecks when analyzing target gene sequencing data: sequence clustering. The microbiome field has been gaining interest from computational biologists, constantly presenting in the literature new tools that increase the quality of the results and reduce the time to solution. Section 2.2.1 presented a benchmark of available sequence clustering tools, and provided a comparative framework that can be used to compare new tools as they become available. This framework can objectively assess the quality and speed of new tools, enabling users to critically assess the best tool they want to use. This empowers users with the ability to choose a tool not based on the promises made in a specific publication, but rather on the actual results in well-described datasets with a different range of characteristics.

As the microbiome field keeps generating more and more data, analyzing the data using modern laptops or personal computers is becoming an impossible task. However, microbiologists do not necessarily have access to supercomputers, and they often rely on cloud services such as Amazon Elastic Compute Cloud (EC2) to perform their analysis. Being an Infrastructure as a Service (IaaS) cloud solution, microbiologists are presented with the challenge of choosing adequate resources to run their analysis tools. In Chapter 3, I showed that memory is the most critical cloud resource because it is the most expensive cloud resource, and a

shortage of memory translates into termination of the user's analysis and a loss of all work performed until that point. In Section 3.1, I describe a potential solution to this problem: CUDSwap, a Loadable Kernel Module (LKM) that monitors the system memory and dynamically adds swap space if the system memory falls below a given threshold. Section 3.2 presented an implementation of the previous design, removing the necessity to use an arbitrary user-defined threshold, and evaluated the performance of CUDSwap using a memory bounded step of the microbial analysis pipeline: sequence clustering. The sequence clustering step follows a semi-sequential memory access pattern that makes it suitable to use swap space under memory oversubscription situations, allowing the process to finish at the expense of a small increase of running time. CUDSwap is a system that improves user experience and increases the value of the resources by providing useful results from all computations. However, CUDSwap should not be used all the time, but rather as a safeguard when a mis-prediction occurs.

Historically, many tools for analyzing microbiome data have been developed by biologists who acquired programming knowledge later in their careers. As the field moves towards big data, more and more computer scientists have been involved in the development of these tools, applying techniques and optimizations that allow tools to operate on modern data scales. However, few developers acknowledge the big data nature of the field, and as a result, many tools are limited to small datasets, and do not scale well to large scale initiatives like the Earth Microbiome Project (EMP) (Section 2.3.2) or the American Gut Project (AGP)

(Section 2.3.3). To achieve the levels of scalability needed by these initiatives, applying just pure computer science optimization techniques or just domain specific optimizations is not enough. A multidisciplinary team of developers is needed, in which computer scientists can contribute algorithmic, user interface, and software engineering optimizations and domain experts can provide a better understanding of the nature of the data to find new ways of approaching the computational problems. This approach, as shown in sections 2.2.1 and 2.2.3, can generate analysis tools that scale to dataset sizes never before thought possible, enabling researches to push microbiome research forward to a whole new level.

6.3 Standardization of metadata and analysis

Accurately describing the environment that a sample comes from is key to characterize the microbiome of the sample and its interactions with its environment. Using a standard to represent this information, such as the Minimum information about a marker gene sequence (MIMARKs) standard [225], allows researches to perform comparative analyses across multiple datasets, improving their ability to find new relationships between the microbiome and its niche. In Chapter 4, I described how meta-analyses move microbiome research forward by increasing the power of the findings. However, a researcher performing a meta-analysis should ensure that the samples have been handled, processed and analyzed in the same way, to reduce the impact of technical differences. In Section 4.1, I

presented Qiita, a system designed to facilitate meta-analyses by normalizing sample metadata and minimizing processing differences. Qiita simplifies the complex process of creating a meta-analysis to a few mouse clicks, cutting down the time and effort spent by researchers from months to few minutes. This ability to easily contextualize samples with massive initiatives like the Human Microbiome Project (HMP), EMP and AGP opens the door to a whole new world of possibilities, empowering researchers with new ways of looking at their data and finding new links between the microbiome and their niche. For example, researchers can find new links between the microbiome and diseases, develop new microbiome-based treatments, or engineer new biofuels. With the microbiome being important in so many fields, the possibilities are endless.

Work still needs to be done to help researchers to provide their sample metadata. A new system that guides the researchers through the MIMARKs standard and allows them to efficiently use the existing ontologies to encode their information would greatly increase the efficiency of microbiome research. This way, the tedious problem of formatting the sample metadata can be reduced from weeks of effort to hours, and ideally would be performed at the same time that the samples are collected. With such a system working jointly with Qiita, researchers could focus on the science and biological questions, rather than spending months of their time on basic data formatting.

6.4 Bringing microbiome research to the clinic

The standardization of sample handling, processing, analysis, and metadata curation, as well as improvements in the efficiency of data processing not only provide a common platform for microbiome research, but also increases efficiency and allows researchers to generate results at speeds never achieved before. In Section 5.1, I show how combining the improvements in the tools with a group of experts can generate multi-omics results in as little as 48 hours. These speeds provide the opportunity for microbiome analysis to be used in areas in which time is critical. One such area is human health, in which a multi-omics, microbiome-based analysis can provide new relevant information to the clinicians, enabling them to generate new hypothesis that guide the ordering of clinical tests to diagnose difficult cases. For example, sequencing can be used to understand *Mycobacterium tuberculosis* outbreaks [68], enabling researchers to identify specific mutations and even the origin of the outbreak. Current diagnostic techniques to find the best treatment are culture-based and can take up to 8 weeks to provide a definitive answer. Empowering clinicians with sequencing information that generates results in less than 8 weeks while identifying possible drug resistance genes present in the *Mycobacterium tuberculosis* strain infecting the patient will be an invaluable resource that can save lives. This is just one example, but with the microbiome being linked to other conditions like diabetes, Parkinson's, Autism, depression, and many others, the potential of improving the current health care would reach

a completely new level. Furthermore, with antibiotic resistance becoming an increasing problem, being able to treat patients with targeted antibiotics rather than broad-spectrum antibiotics will combat this pressing problem.

This thesis provides an efficient and extensible framework for multi-omics analyses. As new microbiome studies are being designed targeting specific links between the microbiome and disease, this framework becomes an invaluable resource for human health. With an increasing pool of samples available in Qiita, techniques such as neural networks or other machine learning approaches could be applied to find new biomarkers that facilitate patient diagnosis. New, non-invasive approaches could be used to diagnose diseases. The microbiome is becoming a key component of precision medicine, and providing a framework that enables faster advances in microbiome research will help to accelerate the use of the microbiome in the clinic.

Bibliography

[1] Rachel I Adams, Ashley C Bateman, Holly M Bik, and James F Meadow. Microbiota of the indoor environment: a meta-analysis. *Microbiome*, 3:49, oct 2015.

[2] Davide Albanese, Paolo Fontana, Carlotta De Filippo, Duccio Cavalieri, and Claudio Donati. MICCA: a complete and accurate software for taxonomic profiling of metagenomic data. *Scientific Reports*, 5:9743, may 2015.

[3] S F Altschul, W Gish, W Miller, E W Myers, and D J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.

[4] Katherine R Amato, Rodolfo Martinez-Mota, Nicoletta Righini, Melissa Raguet-Schofield, Fabiana Paola Corcione, Elisabetta Marini, Greg Humphrey, Grant Gogul, James Gaffney, Elijah Lovelace, LaShanda Williams, Albert Luong, Maria Gloria Dominguez-Bello, Rebecca M Stumpf, Bryan White, Karen E Nelson, Rob Knight, and Steven R Leigh. Phylogenetic and ecological factors impact the gut microbiota of two Neotropical primate species. *Oecologia*, 180(3):717–733, 2016.

[5] Amnon Amir, Daniel McDonald, Jose A Navas-Molina, Justine Debelius, James T Morton, Embriette Hyde, Adam Robbins-Pianka, and Rob Knight. Correcting for Microbial Blooms in Fecal Samples during Room-Temperature Shipping. *mSystems*, 2(2), apr 2017.

[6] Amnon Amir, Daniel McDonald, Jose A Navas-Molina, Evguenia Kopylova, James T Morton, Zhenjiang Zech Xu, Eric P Kightley, Luke R Thompson, Embriette R Hyde, Antonio Gonzalez, and Rob Knight. Deblur Rapidly Resolves Single-Nucleotide Community Sequence Patterns. *mSystems*, 2(2), apr 2017.

[7] Thomas E. Anderson, David E. Culler, and David A. Patterson. A Case for NOW (Networks of Workstations). *IEEE Micro*, 15(1):54–64, 1995.

[8] M Armbrust, A Fox, R Griffith, AD Joseph, and RH. Above the clouds: A Berkeley view of cloud computing. *University of California, Berkeley, Tech. Rep. UCB*, pages 07–013, 2009.

[9] Manimozhiyan Arumugam, Jeroen Raes, Eric Pelletier, Denis Le Paslier, Takuji Yamada, Daniel R. Mende, Gabriel R. Fernandes, Julien Tap, Thomas Bruls, Jean Michel Batto, Marcelo Bertalan, Natalia Borruel, Francesc Casellas, Leyden Fernandez, Laurent Gautier, Torben Hansen, Masahira Hattori, Tetsuya Hayashi, Michiel Kleerebezem, Ken Kurokawa, Marion Leclerc, Florence Levenez, Chaysavanh Manichanh, H. Bjãrn Nielsen, Trine Nielsen, Nicolas Pons, Julie Poulain, Junjie Qin, Thomas Sicheritz-Ponten, Sebastian Tims, David Torrents, Edgardo Ugarte, Erwin G. Zoetendal, Jun Wang, Francisco Guarner, Oluf Pedersen, Willem M. De Vos, Søren Brunak, Joel Doré, Jean Weissenbach, S. Dusko Ehrlich, and Peer Bork. Enterotypes of the human gut microbiome. *Nature*, 473(7346):174–180, 2011.

[10] RM Atlas and R Bartha. *Microbial ecology: fundamentals and applications*. Harlow: Benjamin/Cummings, Menlo Park, 1998.

[11] Wirt Atmar and Bruce D Patterson. The measure of order and disorder in the distribution of species in fragmented habitat. *Oecologia*, 96(3):373–382, 1993.

[12] Júlia Balog, László Sasi-Szabó, James Kinross, Matthew R. Lewis, Laura J. Muirhead, Kirill Veselkov, Reza Mirnezami, Balázs Dezso, László Damjanovich, Ara Darzi, Jeremy K. Nicholson, and Zoltán Takáts. Intraoperative tissue identification using rapid evaporative ionization mass spectrometry. *Science Translational Medicine*, 5(194), 2013.

[13] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the art of virtualization. In *Proceedings of the nineteenth ACM symposium on Operating systems principles - SOSP '03*, page 164, 2003.

[14] N A Bokulich, S Subramanian, J J Faith, D Gevers, J I Gordon, R Knight, D A Mills, and J G Caporaso. Quality-filtering vastly improves diversity estimates from Illumina amplicon sequencing. *Nat Methods*, 10(1):57–59, 2013.

[15] Amina Bouslimani, Carla Porto, Christopher M Rath, Mingxun Wang, Yurong Guo, Antonio Gonzalez, Donna Berg-Lyon, Gail Ackermann, Gitte Julie Moeller Christensen, Teruaki Nakatsuji, Lingjuan Zhang, Andrew W Borkowski, Michael J Meehan, Kathleen Dorrestein, Richard L Gallo, Nuno Bandeira, Rob Knight, Theodore Alexandrov, and Pieter C

Dorrestein. Molecular cartography of the human skin surface in 3D. *Proceedings of the National Academy of Sciences*, 112(17):E2120–E2129, apr 2015.

[16] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[17] Patrizia Brigidi, Beatrice Vitali, Erwin Swennen, Gabriele Bazzocchi, and Diego Matteuzzi. Effects of probiotic administration upon the composition and enzymatic activity of human fecal microbiota in patients with irritable bowel syndrome or functional diarrhea. *Research in Microbiology*, 152(8):735–741, oct 2001.

[18] Benjamin J Callahan, Paul J McMurdie, Michael J Rosen, Andrew W Han, Amy Jo A Johnson, and Susan P Holmes. DADA2: High-resolution sample inference from Illumina amplicon data. *Nat Meth*, advance on, may 2016.

[19] J. Gregory Caporaso, Kyle Bittinger, Frederic D. Bushman, Todd Z. Desantis, Gary L. Andersen, and Rob Knight. PyNAST: A flexible tool for aligning sequences to a template alignment. *Bioinformatics*, 26(2):266–267, 2010.

[20] J Gregory Caporaso, Justin Kuczynski, Jesse Stombaugh, Kyle Bittinger, Frederic D Bushman, Elizabeth K Costello, Noah Fierer, Antonio Gonzalez Peña, Julia K Goodrich, Jeffrey I Gordon, Gavin A Huttley, Scott T Kelley, Dan Knights, Jeremy E Koenig, Ruth E Ley, Catherine A Lozupone, Daniel McDonald, Brian D Muegge, Meg Pirrung, Jens Reeder, Joel R Sevinsky, Peter J Turnbaugh, William A Walters, Jeremy Widmann, Tanya Yatsunenko, Jesse Zaneveld, and Rob Knight. QIIME allows analysis of high-throughput community sequencing data. 7(5):335–336, 2010.

[21] J Gregory Caporaso, Christian L Lauber, Elizabeth K Costello, Donna Berg-Lyons, Antonio Gonzalez, Jesse Stombaugh, Dan Knights, Pawel Gajer, Jacques Ravel, Noah Fierer, Jeffrey I Gordon, and Rob Knight. Moving pictures of the human microbiome. *Genome Biology*, 12(5):R50, 2011.

[22] J Gregory Caporaso, Christian L Lauber, William a Walters, Donna Berg-Lyons, James Huntley, Noah Fierer, Sarah M Owens, Jason Betley, Louise Fraser, Markus Bauer, Niall Gormley, Jack a Gilbert, Geoff Smith, and Rob Knight. Ultra-high-throughput microbial community analysis on the Illumina HiSeq and MiSeq platforms. *The ISME Journal*, 6(8):1621–1624, 2012.

[23] Frederic A. Carvalho, Omry Koren, Julia K. Goodrich, Malin E.V. Johansson, Ilke Nalbantoglu, Jesse D. Aitken, Yueju Su, Benoit Chassaing, William A. Walters, Antonio González, Jose C. Clemente, Tyler C. Cullender, Nicolas Barnich, Arlette Darfeuille-Michaud, Matam Vijay-Kumar, Rob Knight, Ruth E. Ley, and Andrew T. Gewirtz. Transient inability to manage

proteobacteria promotes chronic gut inflammation in TLR5-deficient mice. *Cell Host and Microbe*, 12(2):139–152, 2012.

[24] José C. Carvalho, Pedro Cardoso, Paulo A. V. Borges, Dénes Schmera, and János Podani. Measuring fractions of beta diversity and their relationships to nestedness: a theoretical and empirical comparison of novel approaches. *Oikos*, 122(6):825–834, jun 2013.

[25] Sarah L Castro-Wallace, Charles Y Chiu, Kristen K John, Sarah E Stahl, Kathleen H Rubins, Alexa B R McIntyre, Jason P Dworkin, Mark L Lupisella, David J Smith, Douglas J Botkin, Timothy A Stephenson, Sissel Juul, Daniel J Turner, Fernando Izquierdo, Scot Federman, Doug Stryke, Sneha Somasekar, Noah Alexander, Guixia Yu, Christopher Mason, and Aaron S Burton. Nanopore DNA Sequencing and Genome Assembly on the International Space Station. *bioRxiv*, jan 2016.

[26] John Chakerian and Susan Holmes. Computational tools for evaluating phylogenetic and hierarchical clustering trees. *Journal of Computational and Graphical Statistics*, 21(3):581–599, 2012.

[27] Vincent B. Chen, Ian W. Davis, and David C. Richardson. KiNG (Kinemage, Next Generation): A versatile interactive molecular and scientific visualization program. *Protein Science*, 18(11):2403–2409, 2009.

[28] J R Cole, Q Wang, E Cardenas, J Fish, B Chai, R J Farris, A S Kulam-Syed-Mohideen, D M McGarrell, T Marsh, G M Garrity, and J M Tiedje. The Ribosomal Database Project: improved alignments and new tools for rRNA analysis. *Nucleic Acids Research*, 37(Database issue):D141–D145, jan 2009.

[29] The Human Microbiome Project Consortium. Structure, function and diversity of the healthy human microbiome. *Nature*, 486(7402):207–14, 2012.

[30] E. K. Costello, C. L. Lauber, M. Hamady, N. Fierer, J. I. Gordon, and R. Knight. Bacterial Community Variation in Human Body Habitats Across Space and Time. 326(5960):1694–1697, 2009.

[31] Laura M Cox and Martin J Blaser. Antibiotics in early life and obesity. *Nat Rev Endocrinol*, 11(3):182–190, mar 2015.

[32] Ricardo R. da Silva, Pieter C. Dorrestein, and Robert A. Quinn. Illuminating the dark matter in metabolomics: Fig. 1. *Proceedings of the National Academy of Sciences*, 112(41):12549–12550, 2015.

[33] Michael D Dahlin, Randolph Y Wang, Thomas E Anderson, and David a Patterson. Cooperative Caching: Using Remote Client Memory to Improve File System Performance. In *OSDI'94*, volume 23, pages 267–280, 1994.

[34] Lawrence A. David, Corinne F. Maurice, Rachel N. Carmody, David B. Gootenberg, Julie E. Button, Benjamin E. Wolfe, Alisha V. Ling, A. Sloan Devlin, Yug Varma, Michael A. Fischbach, Sudha B. Biddinger, Rachel J. Dutton, and Peter J. Turnbaugh. Diet rapidly and reproducibly alters the human gut microbiome. *Nature*, 505(7484):559–563, 2014.

[35] Carlotta De Filippo, Duccio Cavalieri, Monica Di Paola, Matteo Ramazzotti, Jean Baptiste Poullet, Sebastien Massart, Silvia Collini, Giuseppe Pieraccini, and Paolo Lionetti. Impact of diet in shaping gut microbiota revealed by a comparative study in children from Europe and rural Africa. *Proceedings of the National Academy of Sciences of the United States of America*, 107(33):14691–6, 2010.

[36] Justine Debelius, Se Jin Song, Yoshiki Vazquez-Baeza, Zhenjiang Zech Xu, Antonio Gonzalez, and Rob Knight. Tiny microbes, enormous impacts: what matters in gut microbiome studies? *Genome Biology*, 17:217, oct 2016.

[37] T. Z. DeSantis, P. Hugenholtz, N. Larsen, M. Rojas, E. L. Brodie, K. Keller, T. Huber, D. Dalevi, P. Hu, and G. L. Andersen. Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB. *Applied and Environmental Microbiology*, 72(7):5069–5072, 2006.

[38] Les Dethlefsen and David A Relman. Incomplete recovery and individualized responses of the human distal gut microbiota to repeated antibiotic perturbation. *Proceedings of the National Academy of Sciences of the United States of America*, (Suppl 1):4554–61, 2011.

[39] M. G. Dominguez-Bello, E. K. Costello, M. Contreras, M. Magris, G. Hidalgo, N. Fierer, and R. Knight. Delivery mode shapes the acquisition and structure of the initial microbiota across multiple body habitats in newborns. *Proceedings of the National Academy of Sciences*, 107(26):11971–11975, 2010.

[40] M. Drancourt, C. Bollet, A. Carlioz, R. Martelin, J. P. Gayral, and D. Raoult. 16S ribosomal DNA sequence analysis of a large collection of environmental and clinical unidentifiable bacterial isolates. *Journal of Clinical Microbiology*, 38(10):3623–3630, 2000.

[41] Paul B Eckburg and David A Relman. The role of microbes in Crohn's disease. *Clinical infectious diseases : an official publication of the Infectious Diseases Society of America*, 44(2):256–62, jan 2007.

[42] Robert C. Edgar. MUSCLE: Multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32(5):1792–1797, 2004.

[43] Robert C. Edgar. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, 26(19):2460–2461, 2010.

[44] Robert C Edgar. UPARSE: highly accurate OTU sequences from microbial amplicon reads. *Nature Methods*, 10(10):996–8, 2013.

[45] Robert C Edgar. UNOISE2: improved error-correction for Illumina 16S and ITS amplicon sequencing. *bioRxiv*, jan 2016.

[46] Robert C. Edgar, Brian J. Haas, Jose C. Clemente, Christopher Quince, and Rob Knight. UCHIME improves sensitivity and speed of chimera detection. *Bioinformatics*, 27(16):2194–2200, 2011.

[47] Jason Evans, Luke Sheneman, and James Foster. Relaxed neighbor joining: A fast distance-based phylogenetic tree construction method. *Journal of Molecular Evolution*, 62(6):785–792, 2006.

[48] Daniel P. Faith. Conservation evaluation and phylogenetic diversity. 61(1):1–10, 1992.

[49] N. Fierer, M. Hamady, C. L. Lauber, and R. Knight. The influence of sex, handedness, and washing on the diversity of hand surface bacteria. *Proceedings of the National Academy of Sciences*, 105(46):17994–17999, 2008.

[50] Noah Fierer, Christian L Lauber, Nick Zhou, Daniel McDonald, Elizabeth K Costello, and Rob Knight. Forensic identification using skin bacterial communities. *Proceedings of the National Academy of Sciences*, 107(14):6477–6481, apr 2010.

[51] Daniel N. Frank. BARCRAWL and BARTAB: Software tools for the design and implementation of barcoded primers for highly multiplexed DNA sequencing. *BMC Bioinformatics*, 10:362, 2009.

[52] Eric A. Franzosa, Tiffany Hsu, Alexandra Sirota-Madi, Afrah Shafquat, Galeb Abu-Ali, Xochitl C. Morgan, and Curtis Huttenhower. Sequencing and beyond: Integrating molecular 'omics' for microbial community profiling, 2015.

[53] Friedrich Leisch. Sweave: Dynamic generation of statistical reports using literate data analysis. *Compstat 2002 - Proceedings in Computational Statistics*, (69):575–580, 2002.

[54] Joëlle V. Fritz, Mahesh S. Desai, Pranjul Shah, Jochen G. Schneider, and Paul Wilmes. From meta-omics to causality: Experimental models for human microbiome research, 2013.

[55] Kevin Gaston and Tim Blackburn. *Pattern and Process in Macroecology*. Wiley-Blackwell, 2000.

[56] Dirk Gevers, Rob Knight, Joseph F Petrosino, Katherine Huang, Amy L McGuire, Bruce W Birren, Karen E Nelson, Owen White, Barbara A Methé, and Curtis Huttenhower. The Human Microbiome Project: A Community Resource for the Healthy Human Microbiome. *PLoS Biology*, 10(8):e1001377, aug 2012.

[57] Dirk Gevers, Subra Kugathasan, Lee A Denson, Yoshiki Vázquez-Baeza, Will Van Treuren, Boyu Ren, Emma Schwager, Dan Knights, Se Jin Song, Moran Yassour, Xochitl C Morgan, Aleksandar D Kostic, Chengwei Luo, Antonio González, Daniel McDonald, Yael Haberman, Thomas Walters, Susan Baker, Joel Rosh, Michael Stephens, Melvin Heyman, James Markowitz, Robert Baldassano, Anne Griffiths, Francisco Sylvester, David Mack, Sandra Kim, Wallace Crandall, Jeffrey Hyams, Curtis Huttenhower, Rob Knight, and Ramnik J Xavier. The treatment-naïve microbiome in new-onset Crohn's disease. *Cell host & microbe*, 15(3):382–392, mar 2014.

[58] Jack A Gilbert, Janet K Jansson, and Rob Knight. The Earth Microbiome project: successes and aspirations. *BMC Biology*, 12(1):69, 2014.

[59] Jack A Gilbert, Folker Meyer, Dion Antonopoulos, Pavan Balaji, C Titus Brown, Christopher T Brown, Narayan Desai, Jonathan A Eisen, Dirk Evers, Dawn Field, Wu Feng, Daniel Huson, Janet Jansson, Rob Knight, James Knight, Eugene Kolker, Kostas Konstantindis, Joel Kostka, Nikos Kyrpides, Rachel Mackelprang, Alice McHardy, Christopher Quince, Jeroen Raes, Alexander Sczyrba, Ashley Shade, and Rick Stevens. Meeting report: the terabase metagenomics workshop and the vision of an Earth microbiome project. *Standards in genomic sciences*, 3(3):243–248, 2010.

[60] Ellie J. C. Goldstein, Kerin L. Tyrrell, Diane M. Citron, Cathleen R. Cox, Ian M. Recchio, Ben Okimoto, Judith Bryja, and Bryan G. Fry. Anaerobic and aerobic bacteriology of the saliva and gingiva from 16 captive Komodo dragons (Varanus komodoensis): new implications for the "bacteria as venom" model. *Journal of Zoo and Wildlife Medicine*, 44(2):262–72, jun 2013.

[61] Antonio Gonzalez and Rob Knight. Advancing analytical algorithms and pipelines for billions of microbial sequences. 23(1):64–71, 2012.

[62] Antonio Gonzalez, Jesse Stombaugh, Christian L. Lauber, Noah Fierer, and Rob Knight. SitePainter: A tool for exploring biogeographical patterns. *Bioinformatics*, 28(3):436–438, 2012.

[63] JuliaK. Goodrich, JillianL. Waters, AngelaC. Poole, JessicaL. Sutter, Omry Koren, Ran Blekhman, Michelle Beaumont, William VanTreuren, Rob Knight, JordanaT. Bell, TimothyD. Spector, AndrewG. Clark, and RuthE.

Ley. Human Genetics Shape the Gut Microbiome. *Cell*, 159(4):789–799, nov 2017.

[64] Sara Goodwin, John D McPherson, and W Richard McCombie. Coming of age: ten years of next-generation sequencing technologies. *Nat Rev Genet*, 17(6):333–351, jun 2016.

[65] Kingshuk Govil, Dan Teodosiu, Yongqiang Huang, and Mendel Rosenblum. Cellular disco: resource management using virtual clusters on shared-memory multiprocessors. *ACM Transactions on Computer Systems*, 18(3):229–262, 2000.

[66] J C Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53(3-4):325–338, 1966.

[67] Alexander L. Greninger, Samia N. Naccache, Scot Federman, Guixia Yu, Placide Mbala, Vanessa Bres, Doug Stryke, Jerome Bouquet, Sneha Somasekar, Jeffrey M. Linnen, Roger Dodd, Prime Mulembakani, Bradley S. Schneider, Jean Jacques Muyembe-Tamfum, Susan L. Stramer, and Charles Y. Chiu. Rapid metagenomic identification of viral pathogens in clinical samples by real-time nanopore sequencing analysis. *Genome Medicine*, 7(1), 2015.

[68] Jennifer L Guthrie and Jennifer L Gardy. A brief primer on genomic epidemiology: lessons learned from Mycobacterium tuberculosis. *Annals of the New York Academy of Sciences*, 1388(1):59–77, jan 2017.

[69] Brian J. Haas, Dirk Gevers, Ashlee M. Earl, Mike Feldgarden, Doyle V. Ward, Georgia Giannoukos, Dawn Ciulla, Diana Tabbaa, Sarah K. Highlander, Erica Sodergren, Barbara Methé, Todd Z. DeSantis, Joseph F. Petrosino, Rob Knight, and Bruce W. Birren. Chimeric 16S rRNA sequence formation and detection in Sanger and 454-pyrosequenced PCR amplicons. *Genome Research*, 21(3):494–504, 2011.

[70] Henry J. Haiser, David B. Gootenberg, Kelly Chatman, Gopal Sirasani, Emily P. Balskus, and Peter J. Turnbaugh. Predicting and manipulating cardiac drug inactivation by the human gut bacterium Eggerthella lenta. *Science*, 341(6143):295–298, 2013.

[71] Jonas Halfvarson, Colin J Brislawn, Regina Lamendella, Yoshiki Vázquez-Baeza, William A Walters, Lisa M Bramer, Mauro D'Amato, Ferdinando Bonfiglio, Daniel McDonald, Antonio Gonzalez, Erin E McClure, Mitchell F Dunklebarger, Rob Knight, and Janet K Jansson. Dynamics of the human gut microbiome in Inflammatory Bowel Disease. *Nature microbiology*, 2:17004, feb 2017.

[72] Micah Hamady and Rob Knight. Microbial community profiling for human microbiome projects: Tools, techniques, and challenges. 19(7):1141–1152, 2009.

[73] Micah Hamady, Jeffrey J. Walker, J. Kirk Harris, Nicholas J. Gold, and Rob Knight. Error-correcting barcoded primers for pyrosequencing hundreds of samples in multiplex. *Nature Methods*, 5(3):235–237, 2008.

[74] R. D. Heijtz, S. Wang, F. Anuar, Y. Qian, B. Bjorkholm, A. Samuelsson, M. L. Hibberd, H. Forssberg, and S. Pettersson. Normal gut microbiota modulates brain development and behavior. *Proceedings of the National Academy of Sciences*, 108(7):3047–3052, 2011.

[75] Matthias Hess, Alexander Sczyrba, Rob Egan, Tae-Wan Kim, Harshal Chokhawala, Gary Schroth, Shujun Luo, Douglas S Clark, Feng Chen, Tao Zhang, Roderick I Mackie, Len A Pennacchio, Susannah G Tringe, Axel Visel, Tanja Woyke, Zhong Wang, and Edward M Rubin. Metagenomic Discovery of Biomass-Degrading Genes and Genomes from Cow Rumen. *Science*, 331(6016):463–467, jan 2011.

[76] Krissi M. Hewitt, Frank L. Mannino, Antonio Gonzalez, John H. Chase, J. Gregory Caporaso, Rob Knight, and Scott T. Kelley. Bacterial Diversity in Two Neonatal Intensive Care Units (NICUs). *PLoS ONE*, 8(1), 2013.

[77] Falk Hildebrand, Raul Tadeo, Anita Voigt, Peer Bork, and Jeroen Raes. LotuS: an efficient and user-friendly OTU processing pipeline. *Microbiome*, 2(1):30, sep 2014.

[78] Cheng Chih Hsu, Mariam S. Elnaggar, Yao Peng, Jinshu Fang, Laura M. Sanchez, Samantha J. Mascuch, Kirsten A. Møller, Emad K. Alazzeh, Jiri Pikula, Robert A. Quinn, Yi Zeng, Benjamin E. Wolfe, Rachel J. Dutton, Lena Gerwick, Lixin Zhang, Xueting Liu, Maria Mansson, and Pieter C. Dorrestein. Real-time metabolomics on living microorganisms using ambient electrospray ionization flow-probe. *Analytical Chemistry*, 85(15):7014–7018, 2013.

[79] Weichun Huang, Leping Li, Jason R. Myers, and Gabor T. Marth. ART: A next-generation sequencing read simulator. *Bioinformatics*, 28(4):593–594, 2012.

[80] Embriette R Hyde, Jessica L Metcalf, Sibyl R Bucheli, Aaron M Lynne, and Rob Knight. Microbial communities associated with decomposing corpses. In *Forensic Microbiology*, pages 245–273. John Wiley & Sons, Ltd, 2017.

[81] Embriette R Hyde, Jose A Navas-Molina, Se Jin Song, Jordan G Kueneman, Gail Ackermann, Cesar Cardona, Gregory Humphrey, Don Boyer, Tom

Weaver, Joseph R Mendelson, Valerie J McKenzie, Jack A Gilbert, and Rob Knight. The Oral and Skin Microbiomes of Captive Komodo Dragons Are Significantly Shared with Their Habitat. *mSystems*, 1(4), aug 2016.

[82] Sarah S Johnson, Elena Zaikova, David S Goerlitz, Yu Bai, and Scott W Tighe. Real-Time DNA Sequencing in the Antarctic Dry Valleys Using the Oxford Nanopore Sequencer. *Journal of Biomolecular Techniques : JBT*, 28(1):2–7, apr 2017.

[83] Lou Jost. Partitioning diversity into independent alpha and beta components. *Ecology*, 88(10):2427–2439, 2007.

[84] Kazutaka Katoh, Kazuharu Misawa, Kei-ichi Kuma, and Takashi Miyata. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic acids research*, 30(14):3059–3066, 2002.

[85] W. James Kent. BLAT - The BLAST-like alignment tool. *Genome Research*, 12(4):656–664, 2002.

[86] Rob Knight, Janet Jansson, Dawn Field, Noah Fierer, Narayan Desai, Jed A. Fuhrman, Phil Hugenholtz, Daniel Van Der Lelie, Folker Meyer, Rick Stevens, Mark J. Bailey, Jeffrey I. Gordon, George A. Kowalchuk, and Jack A. Gilbert. Unlocking the potential of metagenomics through replicated experimental design, 2012.

[87] Dan Knights, Elizabeth K. Costello, and Rob Knight. Supervised classification of human microbiota, 2011.

[88] Dan Knights, Justin Kuczynski, Emily S Charlson, Jesse Zaneveld, Michael C Mozer, Ronald G Collman, Frederic D Bushman, Rob Knight, and Scott T Kelley. Bayesian community-wide culture-independent microbial source tracking. *Nat Meth*, 8(9):761–763, sep 2011.

[89] Dan Knights, Justin Kuczynski, Omry Koren, Ruth E. Ley, Dawn Field, Rob Knight, Todd Z. Desantis, and Scott T. Kelley. Supervised classification of microbiota mitigates mislabeling errors, 2011.

[90] Jeremy E Koenig, Aymé Spor, Nicholas Scalfone, Ashwana D Fricker, Jesse Stombaugh, Rob Knight, Largus T Angenent, and Ruth E Ley. Succession of microbial consortia in the developing infant gut microbiome. *Proceedings of the National Academy of Sciences*, 108(Supplement 1):4578–4585, mar 2011.

[91] Robert A. Koeth, Zeneng Wang, Bruce S. Levison, Jennifer A. Buffa, Elin Org, Brendan T. Sheehy, Earl B. Britt, Xiaoming Fu, Yuping Wu, Lin Li, Jonathan D. Smith, Joseph A. Didonato, Jun Chen, Hongzhe Li, Gary D. Wu, James D. Lewis, Manya Warrier, J. Mark Brown, Ronald M. Krauss,

W. H.Wilson Tang, Frederic D. Bushman, Aldons J. Lusis, and Stanley L. Hazen. Intestinal microbiota metabolism of l-carnitine, a nutrient in red meat, promotes atherosclerosis. *Nature Medicine*, 19(5):576–585, 2013.

[92] Evguenia Kopylova, Laurent Noé, and Hélène Touzet. SortMeRNA: Fast and accurate filtering of ribosomal RNAs in metatranscriptomic data. *Bioinformatics*, 28(24):3211–3217, 2012.

[93] Omry Koren, Julia K. Goodrich, Tyler C. Cullender, Aymé Spor, Kirsi Laitinen, Helene Kling Bäckhed, Antonio Gonzalez, Jeffrey J. Werner, Largus T. Angenent, Rob Knight, Fredrik Bäckhed, Erika Isolauri, Seppo Salminen, and Ruth E. Ley. Host remodeling of the gut microbiome and metabolic changes during pregnancy. *Cell*, 150(3):470–480, 2012.

[94] R Krishnakumar. Kernel korner: kprobes-a kernel debugger. *Linux J.*, 2005(133):11, 2005.

[95] Justin Kuczynski, Elizabeth K. Costello, Diana R. Nemergut, Jesse Zaneveld, Christian L. Lauber, Dan Knights, Omry Koren, Noah Fierer, Scott T. Kelley, Ruth E. Ley, Jeffrey I. Gordon, and Rob Knight. Direct sequencing of the human microbiome readily reveals community differences. 11(5), 2010.

[96] Justin Kuczynski, Christian L Lauber, William A Walters, Laura Wegener Parfrey, José C Clemente, Dirk Gevers, and Rob Knight. Experimental and analytical tools for studying the human microbiome. *Nature reviews. Genetics*, 13(1):47–58, dec 2011.

[97] Justin Kuczynski, Zongzhi Liu, Catherine Lozupone, Daniel McDonald, Noah Fierer, and Rob Knight. Microbial community resemblance methods differ in their ability to detect biologically relevant patterns. *Nature Methods*, 7(10):813–819, 2010.

[98] Jordan G. Kueneman, Laura Wegener Parfrey, Douglas C. Woodhams, Holly M. Archer, Rob Knight, and Valerie J. McKenzie. The amphibian skin-associated microbiome across species, space and life history stages. *Molecular Ecology*, 23(6):1238–1250, mar 2014.

[99] H Andrés Lagar-Cavilla, Joseph A Whitney, Adin Scannell, Philip Patchin, Stephen M Rumble, Eyal De Lara, Michael Brudno, and M Satyanarayanan. SnowFlock: Rapid Virtual Machine Cloning for Cloud Computing. *Proceedings of the 4th ACM European conference on Computer systems*, pages 1–12, 2009.

[100] Mgi Langille, Jesse Zaneveld, J Gregory Caporaso, Daniel McDonald, Dan Knights, Ja Reyes, Jc Clemente, De Burkepile, Rl Vega Thurber, Rob Knight, Rg Beiko, and Curtis Huttenhower. Predictive functional profiling

of microbial communities using 16S rRNA marker gene sequences. *Nature biotechnology*, 31(9):814–21, 2013.

[101] Morgan G. I. Langille, Jacques Ravel, and W. Florian Fricke. "available upon request": not good enough for microbiome data! *Microbiome*, 6(1):8, Jan 2018.

[102] M. A. Larkin, G. Blackshields, N. P. Brown, R. Chenna, P. A. Mcgettigan, H. McWilliam, F. Valentin, I. M. Wallace, A. Wilm, R. Lopez, J. D. Thompson, T. J. Gibson, and D. G. Higgins. Clustal W and Clustal X version 2.0. *Bioinformatics*, 23(21):2947–2948, 2007.

[103] Christian L Lauber, Micah Hamady, Rob Knight, and Noah Fierer. Pyrosequencing-based assessment of soil pH as a predictor of soil bacterial community structure at the continental scale. *Applied and environmental microbiology*, 75(15):5111–20, aug 2009.

[104] Christian L Lauber, Nicholas Zhou, Jeffrey I Gordon, Rob Knight, and Noah Fierer. Effect of storage conditions on the assessment of bacterial community structure in soil and human-associated samples. *FEMS Microbiology Letters*, 307(1):80–86, jun 2010.

[105] Simon Lax, Cathryn R Nagler, and Jack A Gilbert. Our interface with the built environment: immunity and the indoor microbiota. *Trends in Immunology*, 36(3):121–123, nov 2015.

[106] Simon Lax, Daniel P Smith, Jarrad Hampton-Marcell, Sarah M Owens, Kim M Handley, Nicole M Scott, Sean M Gibbons, Peter Larsen, Benjamin D Shogan, Sophie Weiss, Jessica L Metcalf, Luke K Ursell, Yoshiki Vázquez-Baeza, Will Van Treuren, Nur A Hasan, Molly K Gibson, Rita Colwell, Gautam Dantas, Rob Knight, and Jack A Gilbert. Longitudinal analysis of microbial interaction between humans and the indoor environment. *Science*, 345(6200):1048 LP – 1052, aug 2014.

[107] Joshua Lederberg and A T McCray. 'Ome Sweet 'Omics - a genealogical treasury of words. *Scientist*, 15(8), 2001.

[108] Ruth E Ley, Micah Hamady, Catherine Lozupone, Peter Turnbaugh, Rob Roy Ramey, J Stephen Bircher, Michael L Schlegel, Tammy A Tucker, Mark D Schrenzel, Rob Knight, and Jeffrey I Gordon. Evolution of mammals and their gut microbes. *Science (New York, N.Y.)*, 320(5883):1647–1651, jun 2008.

[109] Ruth E Ley, Catherine A Lozupone, Micah Hamady, Rob Knight, and Jeffrey I Gordon. Worlds within worlds: evolution of the vertebrate gut microbiota. *Nature reviews. Microbiology*, 6(10):776–788, oct 2008.

[110] M. Li, B. Wang, M. Zhang, M. Rantalainen, S. Wang, H. Zhou, Y. Zhang, J. Shen, X. Pang, M. Zhang, H. Wei, Y. Chen, H. Lu, J. Zuo, M. Su, Y. Qiu, W. Jia, C. Xiao, L. M. Smith, S. Yang, E. Holmes, H. Tang, G. Zhao, J. K. Nicholson, L. Li, and L. Zhao. Symbiotic gut microbes modulate human metabolic phenotypes. *Proceedings of the National Academy of Sciences*, 105(6):2117–2122, 2008.

[111] W Li, L Jaroszewski, and A Godzik. Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics (Oxford, England)*, 17(3):282–283, 2001.

[112] Weizhong Li and Adam Godzik. Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, 2006.

[113] a Liaw and M Wiener. Classification and Regression by randomForest. *R news*, 2(December):18–22, 2002.

[114] Losee L Ling, Tanja Schneider, Aaron J Peoples, Amy L Spoering, Ina Engels, Brian P Conlon, Anna Mueller, Till F Schaberle, Dallas E Hughes, Slava Epstein, Michael Jones, Linos Lazarides, Victoria A Steadman, Douglas R Cohen, Cintia R Felix, K Ashley Fetterman, William P Millett, Anthony G Nitti, Ashley M Zullo, Chao Chen, and Kim Lewis. A new antibiotic kills pathogens without detectable resistance. *Nature*, 517(7535):455–459, jan 2015.

[115] Zongzhi Liu, Todd Z. Desantis, Gary L. Andersen, and Rob Knight. Accurate taxonomy assignments from 16S rRNA sequences produced by highly parallel pyrosequencers. *Nucleic Acids Research*, 36(18), 2008.

[116] Mark V. Lomolino. Investigating causality of nestedness of insular communities: selective immigrations or extinctions? *Journal of Biogeography*, 23(5):699–703, sep 1996.

[117] T Loua. *Atlas statistique de la population de Paris*. J. Dejey & cie, Paris, 1873.

[118] Catherine Lozupone and Rob Knight. UniFrac: A new phylogenetic method for comparing microbial communities. *Applied and Environmental Microbiology*, 71(12):8228–8235, 2005.

[119] Catherine Lozupone, Manuel E Lladser, Dan Knights, Jesse Stombaugh, and Rob Knight. UniFrac: an effective distance metric for microbial community comparison. *The ISME journal*, 5(2):169–172, 2011.

[120] Catherine A Lozupone and Rob Knight. Global patterns in bacterial diversity. *Proceedings of the National Academy of Sciences*, 104(27):11436–11440, jul 2007.

[121] Catherine A. Lozupone and Rob Knight. Species divergence and the measurement of microbial diversity, 2008.

[122] Catherine A Lozupone, Jesse Stombaugh, Antonio Gonzalez, Gail Ackermann, Doug Wendel, Yoshiki Vázquez-Baeza, Janet K Jansson, Jeffrey I Gordon, and Rob Knight. Meta-analyses of studies of the human microbiota. *Genome Research*, 23(10):1704–1714, oct 2013.

[123] Wolfgang Ludwig, Oliver Strunk, Ralf Westram, Lothar Richter, Harald Meier, A. Yadhukumar, Arno Buchner, Tina Lai, Susanne Steppi, Gangolf Jacob, Wolfram Förster, Igor Brettske, Stefan Gerber, Anton W. Ginhart, Oliver Gross, Silke Grumann, Stefan Hermann, Ralf Jost, Andreas König, Thomas Liss, Ralph Lüßbmann, Michael May, Björn Nonhoff, Boris Reichel, Robert Strehlow, Alexandros Stamatakis, Norbert Stuckmann, Alexander Vilbig, Michael Lenke, Thomas Ludwig, Arndt Bode, and Karl Heinz Schleifer. ARB: A software environment for sequence data. *Nucleic Acids Research*, 32(4):1363–1371, 2004.

[124] Frédéric Mahé, Torbjørn Rognes, Christopher Quince, Colomban de Vargas, and Micah Dunthorn. Swarm : robust and fast clustering method for amplicon-based studies PrePrints PrePrints. *PeerJ*, (May):1–12, 2014.

[125] Frédéric Mahé, Torbjørn Rognes, Christopher Quince, Colomban de Vargas, and Micah Dunthorn. Swarm v2: highly-scalable and high-resolution amplicon clustering. *PeerJ*, 3:e1420, 2015.

[126] Siddhartha Mandal, Will Van Treuren, Richard A White, Merete Eggesbø, Rob Knight, and Shyamal D Peddada. Analysis of composition of microbiomes: a novel method for studying microbial composition. *Microbial Ecology in Health and Disease*, 26:10.3402/mehd.v26.27663, may 2015.

[127] Nathan Mantel. The Detection of Disease Clustering and a Generalized Regression Approach. *Cancer Research*, 27(2 Part 1):209 LP – 220, feb 1967.

[128] Corinne Ferrier Maurice, Henry Joseph Haiser, and Peter James Turnbaugh. Xenobiotics shape the physiology and gene expression of the active human gut microbiome. *Cell*, 152(1-2):39–50, jan 2013.

[129] A. Mavinakayahalli, P. Panchamukhi, J. Keniston, A. Keshavamurthy, and M. Hiramatsu. Probing the guts of kprobes. In *Proceeding of Linux Symposium*, 2006.

[130] Daniel McDonald, Jose C Clemente, Justin Kuczynski, Jai Rideout, Jesse Stombaugh, Doug Wendel, Andreas Wilke, Susan Huse, John Hufnagle, Folker Meyer, Rob Knight, and J Caporaso. The Biological Observation Matrix (BIOM) format or: how I learned to stop worrying and love the ome-ome. *GigaScience*, 1(1):7, 2012.

[131] Daniel McDonald, Morgan N Price, Julia Goodrich, Eric P Nawrocki, Todd Z DeSantis, Alexander Probst, Gary L Andersen, Rob Knight, and Philip Hugenholtz. An improved Greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea, 2012.

[132] PE McGovern. *Ancient wine: the search for the origin of viticulture*. Princeton University Press, Princeton, 2003.

[133] P. G. McLean, G. E. Bergonzelli, S. M. Collins, and P. Bercik. Targeting the microbiota-gut-brain axis to modulate behavior: Which bacterial strain will translate best to humans? *Proceedings of the National Academy of Sciences*, 109(4):E174–E174, 2012.

[134] Paul J. McMurdie and Susan Holmes. Phyloseq: An R Package for Reproducible Interactive Analysis and Graphics of Microbiome Census Data. *PLoS ONE*, 8(4), 2013.

[135] Alexander Mellmann, Dag Harmsen, Craig A. Cummings, Emily B. Zentz, Shana R. Leopold, Alain Rico, Karola Prior, Rafael Szczepanowski, Yongmei Ji, Wenlan Zhang, Stephen F. McLaughlin, John K. Henkhaus, Benjamin Leopold, Martina Bielaszewska, Rita Prager, Pius M. Brzoska, Richard L. Moore, Simone Guenther, Jonathan M. Rothberg, and Helge Karch. Prospective genomic characterization of the german enterohemorrhagic Escherichia coli O104:H4 outbreak by rapid next generation sequencing technology. *PLoS ONE*, 6(7), 2011.

[136] J. N. Molina and S. Mishra. Addressing Memory Exhaustion Failures in Virtual Machines in a Cloud Environment. In *The Third International Workshop on Dependability of Clouds, Data Centers and Virtual Machine Technology (DCDV 2013)*, 2013.

[137] Joel M. Montgomery, Don Gillespie, Putra Sastrawan, Terry M. Fredeking, and George L. Stewart. Aerobic salivary bacteria in wild and captive Komodo dragons. *Journal of wildlife diseases*, 38(3):545–551, jul 2002.

[138] James T Morton, Jon Sanders, Robert A Quinn, Daniel McDonald, Antonio Gonzalez, Yoshiki Vázquez-Baeza, Jose A Navas-Molina, Se Jin Song, Jessica L Metcalf, Embriette R Hyde, Manuel Lladser, Pieter C Dorrestein, and Rob Knight. Balance Trees Reveal Microbial Niche Differentiation. *mSystems*, 2(1), 2017.

[139] Dariush Mozaffarian, Tao Hao, Eric B Rimm, Walter C Willett, and Frank B Hu. Changes in diet and lifestyle and long-term weight gain in women and men. *The New England journal of medicine*, 364(25):2392–404, 2011.

[140] Brian D Muegge, Justin Kuczynski, Dan Knights, Jose C Clemente, Antonio González, Luigi Fontana, Bernard Henrissat, Rob Knight, and Jeffrey I Gordon. Diet drives convergence in gut microbiome functions across mammalian phylogeny and within humans. *Science (New York, N.Y.)*, 332(6032):970–4, 2011.

[141] Brad A Myers and Mary Beth Rosson. Survey on User Interface Programming. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '92, pages 195–202, New York, NY, USA, 1992. ACM.

[142] Samia N. Naccache, Scot Federman, Narayanan Veeraraghavan, Matei Zaharia, Deanna Lee, Erik Samayoa, Jerome Bouquet, Alexander L. Greninger, Ka Cheung Luk, Barryett Enge, Debra A. Wadford, Sharon L. Messenger, Gillian L. Genrich, Kristen Pellegrino, Gilda Grard, Eric Leroy, Bradley S. Schneider, Joseph N. Fair, Miguel A. Martínez, Pavel Isa, John A. Crump, Joseph L. DeRisi, Taylor Sittler, John Hackett, Steve Miller, and Charles Y. Chiu. A cloud-compatible bioinformatics pipeline for ultrarapid pathogen identification from next-generation sequencing of clinical samples. *Genome Research*, 24(7):1180–1192, 2014.

[143] José A. Navas-Molina, Juan M. Peralta-Sánchez, Antonio González, Paul J. McMurdie, Yoshiki Vázquez-Baeza, Zhenjiang Xu, Luke K. Ursell, Christian Lauber, Hongwei Zhou, Se Jin Song, James Huntley, Gail L. Ackermann, Donna Berg-Lyons, Susan Holmes, J. Gregory Caporaso, and Rob Knight. Advancing our understanding of the human microbiome using QIIME. *Methods in Enzymology*, 531:371–444, 2013.

[144] Eric P. Nawrocki, Diana L. Kolbe, and Sean R. Eddy. Infernal 1.0: Inference of RNA alignments. *Bioinformatics*, 25(10):1335–1337, 2009.

[145] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

[146] Michael Nelson, Beng-hong Lim, and Greg Hutchins. Fast Transparent Migration for Virtual Machines. *ATC*, pages(6009):391–394, 2005.

[147] J.D. Neufeld, K. Engel, J. Cheng, G. Moreno-Hagelsieb, D.R. Rose, and T.C. Charles. Open resource metagenomics: a model for sharing metagenomic libraries. 5(2):203–210, 2011.

[148] Tia Newhall, Sean Finney, Kuzman Ganchev, and Michael Spiegel. Nswap : A Network Swapping Module for Linux Clusters. *Lecture Notes in Computer Science*, 2790:1160–1169, 2003.

[149] David J Novo, Nancy G Perlmutter, Richard H Hunt, and Howard M Shapiro. Multiparameter Flow Cytometric Analysis of Antibiotic Effects on Membrane Potential, Membrane Permeability, and Bacterial Counts of Staphylococcus aureus and Micrococcus luteus. *Antimicrobial Agents and Chemotherapy*, 44(4):827–834, apr 2000.

[150] Gary J Olsen, Niels Larsen, and Carl R Woese. The ribosomal RNA Database project. *Nucleic Acids Research*, 19(Suppl):2017–2021, apr 1991.

[151] Gary J Olsen, Ross Overbeek, Niels Larsen, Terry L Marsh, Michael J McCaughey, Michael A Maciukenas, Wen-Min Kuan, Thomas J Macke, Yuqing Xing, and Carl R Woese. The Ribosomal Database Project. *Nucleic Acids Research*, 20(Suppl):2199–2200, may 1992.

[152] Emmanuel Paradis, Julien Claude, and Korbinian Strimmer. APE: Analyses of phylogenetics and evolution in R language. *Bioinformatics*, 20(2):289–290, 2004.

[153] F. Perez and B. E. Granger. Ipython: A system for interactive scientific computing. *Computing in Science & Engineering*, 9(3):21–29, 2007.

[154] Meg Pirrung, Ryan Kennedy, J. Gregory Caporaso, Jesse Stombaugh, Doug Wendel, and Rob Knight. TopiaryExplorer: Visualizing large phylogenetic trees with environmental metadata. *Bioinformatics*, 27(21):3067–3069, 2011.

[155] Dorota L. Porazinska, Robin M. Giblin-Davis, Lina Faller, William Farmerie, Natsumi Kanzaki, Krystalynne Morris, Thomas O. Powers, Abraham E. Tucker, Way Sung, and W. Kelley Thomas. Evaluating high-throughput sequencing as a method for metagenomic analysis of nematode diversity. *Molecular Ecology Resources*, 9(6):1439–1450, 2009.

[156] Jeevan K. Prasain, Kenneth Jones, Marion Kirk, Landon Wilson, Michelle Smith-Johnson, Connie Weaver, and Stephen Barnes. Profiling and quantification of isoflavonoids in kudzu dietary supplements by high-performance liquid chromatography and electrospray ionization tandem mass spectrometry. *Journal of Agricultural and Food Chemistry*, 51(15):4213–4218, 2003.

[157] Morgan N. Price, Paramvir S. Dehal, and Adam P. Arkin. Fasttree: Computing large minimum evolution trees with profiles instead of a distance matrix. *Molecular Biology and Evolution*, 26(7):1641–1650, 2009.

[158] Christian Quast, Elmar Pruesse, Pelin Yilmaz, Jan Gerken, Timmy Schweer, Pablo Yarza, Jörg Peplies, and Frank Oliver Glöckner. The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. *Nucleic Acids Research*, 41(Database issue):D590–D596, jan 2013.

[159] Joshua Quick, Nicholas J Loman, Sophie Duraffour, Jared T Simpson, Ettore Severi, Lauren Cowley, Joseph Akoi Bore, Raymond Koundouno, Gytis Dudas, Amy Mikhail, Nobila Ouédraogo, Babak Afrough, Amadou Bah, Jonathan H J Baum, Beate Becker-Ziaja, Jan-Peter Boettcher, Mar Cabeza-Cabrerizo, Alvaro Camino-Sanchez, Lisa L Carter, Juiliane Doerrbecker, Theresa Enkirch, Isabel Graciela García Dorival, Nicole Hetzelt, Julia Hinzmann, Tobias Holm, Liana Eleni Kafetzopoulou, Michel Koropogui, Abigail Kosgey, Eeva Kuisma, Christopher H Logue, Antonio Mazzarelli, Sarah Meisel, Marc Mertens, Janine Michel, Didier Ngabo, Katja Nitzsche, Elisa Pallash, Livia Victoria Patrono, Jasmine Portmann, Johanna Gabriella Repits, Natasha Yasmin Rickett, Andrea Sachse, Katrin Singethan, Inês Vitoriano, Rahel L Yemanaberhan, Elsa G Zekeng, Racine Trina, Alexander Bello, Amadou Alpha Sall, Ousmane Faye, Oumar Faye, N'Faly Magassouba, Cecelia V Williams, Victoria Amburgey, Linda Winona, Emily Davis, Jon Gerlach, Franck Washington, Vanessa Monteil, Marine Jourdain, Marion Bererd, Alimou Camara, Hermann Somlare, Abdoulaye Camara, Marianne Gerard, Guillaume Bado, Bernard Baillet, Déborah Delaune, Koumpingnin Yacouba Nebie, Abdoulaye Diarra, Yacouba Savane, Raymond Bernard Pallawo, Giovanna Jaramillo Gutierrez, Natacha Milhano, Isabelle Roger, Christopher J Williams, Facinet Yattara, Kuiama Lewandowski, Jamie Taylor, Philip Rachwal, Daniel Turner, Georgios Pollakis, Julian A Hiscox, David A Matthews, Matthew K O'Shea, Andrew McD Johnston, Duncan Wilson, Emma Hutley, Erasmus Smit, Antonino Di Caro, Roman Woelfel, Kilian Stoecker, Erna Fleischmann, Martin Gabriel, Simon A Weller, Lamine Koivogui, Boubacar Diallo, Sakoba Keita, Andrew Rambaut, Pierre Formenty, Stephan Gunther, and Miles W Carroll. Real-time, portable genome sequencing for Ebola surveillance. *Nature*, 530(7589):228–232, feb 2016.

[160] Christopher Quince, Anders Lanzen, Thomas P. Curtis, Russell J. Davenport, Neil Hall, Ian M. Head, L. Fiona Read, and William T. Sloan. Accurate determination of microbial diversity from 454 pyrosequencing data. *Nature Methods*, 6(9):639–641, 2009.

[161] Christopher Quince, Anders Lanzen, Russell J Davenport, and Peter J Turnbaugh. Removing Noise From Pyrosequenced Amplicons. *BMC Bioinformatics*, 12(1):38, 2011.

[162] Benjamin Ragan-Kelley, William Anton Walters, Daniel McDonald, Justin Riley, Brian E Granger, Antonio Gonzalez, Rob Knight, Fernando Perez,

and J Gregory Caporaso. Collaborative cloud-enabled tools allow rapid, reproducible biological insights. *ISME J*, 7(3):461–464, mar 2013.

[163] Benjamin Ragan-Kelley, William Anton Walters, Daniel McDonald, Justin Riley, Brian E Granger, Antonio Gonzalez, Rob Knight, Fernando Perez, and J Gregory Caporaso. Collaborative cloud-enabled tools allow rapid, reproducible biological insights. *ISME J*, 7(3):461–464, mar 2013.

[164] Satwik Rajaram and Yoshi Oono. NeatMap - non-clustering heat map alternatives in R. *BMC Bioinformatics*, 11, 2010.

[165] Kelly S Ramirez, Jonathan W Leff, Albert Barberán, Scott Thomas Bates, Jason Betley, Thomas W Crowther, Eugene F Kelly, Emily E Oldfield, E Ashley Shaw, Christopher Steenbock, Mark A Bradford, Diana H Wall, and Noah Fierer. Biogeographic patterns in below-ground diversity in New York City's Central Park are similar to those observed globally. *Proceedings of the Royal Society of London B: Biological Sciences*, 281(1795), oct 2014.

[166] Jacques Ravel, Pawel Gajer, Li Fu, Christine K. Mauck, Sara S.K. Koenig, Joyce Sakamoto, Alison A. Motsinger-Reif, Gustavo F. Doncel, and Steven L. Zeichner. Twice-daily application of HIV microbicides alters the vaginal microbiota. *mBio*, 3(6), 2012.

[167] T D Read, S L Salzberg, M Pop, M Shumway, L Umayam, L Jiang, E Holtzapple, J D Busch, K L Smith, J M Schupp, D Solomon, P Keim, and C M Fraser. Comparative genome sequencing for discovery of novel polymorphisms in Bacillus anthracis. *Science*, 296(5575):2028–2033, 2002.

[168] Jens Reeder and Rob Knight. Rapidly denoising pyrosequencing amplicon reads by exploiting rank-abundance distributions, 2010.

[169] Vanessa K Ridaura, Jeremiah J Faith, Federico E Rey, Jiye Cheng, Alexis E Duncan, Andrew L Kau, Nicholas W Griffin, Vincent Lombard, Bernard Henrissat, James R Bain, Michael J Muehlbauer, Olga Ilkayeva, Clay F Semenkovich, Katsuhiko Funai, David K Hayashi, Barbara J Lyle, Margaret C Martini, Luke K Ursell, Jose C Clemente, William Van Treuren, William A Walters, Rob Knight, Christopher B Newgard, Andrew C Heath, and Jeffrey I Gordon. Gut Microbiota from Twins Discordant for Obesity Modulate Metabolism in Mice. *Science*, 341(6150), sep 2013.

[170] Jai Ram Rideout, Yan He, Jose A Navas-Molina, William A Walters, Luke K Ursell, Sean M Gibbons, John Chase, Daniel McDonald, Antonio Gonzalez, Adam Robbins-Pianka, Jose C Clemente, Jack A Gilbert, Susan M Huse, Hong-Wei Zhou, Rob Knight, and J Gregory Caporaso. Subsampled open-reference clustering creates consistent, comprehensive OTU definitions and scales to billions of sequences. *PeerJ*, 2:e545, 2014.

[171] Luiz F.W. Roesch, Roberta R. Fulthorpe, Alberto Riva, George Casella, Alison K.M. Hadwin, Angela D. Kent, Samira H. Daroub, Flavio A.O. Camargo, William G. Farmerie, and Eric W. Triplett. Pyrosequencing enumerates and contrasts soil microbial diversity. *ISME Journal*, 1(4):283–290, 2007.

[172] Torbjørn Rognes, Tomáš Flouri, Ben Nichols, Christopher Quince, and Frédéric Mahé. VSEARCH: a versatile open source tool for metagenomics. *PeerJ*, 4:e2584, 2016.

[173] Noortje G. Rossen, John K. MacDonald, Elisabeth M. De Vries, Geert R. D'Haens, Willem M. De Vos, Erwin G. Zoetendal, and Cyriel Y. Ponsioen. Fecal microbiota transplantation as novel therapy in gastroenterology: A systematic review. *World Journal of Gastroenterology*, 21(17):5359–5371, 2015.

[174] D C Savage. Microbial Ecology of the Gastrointestinal Tract. *Annual Review of Microbiology*, 31(1):107–133, 1977.

[175] Patrick D. Schloss and Jo Handelsman. Introducing DOTUR, a computer program for defining operational taxonomic units and estimating species richness. *Applied and Environmental Microbiology*, 71(3):1501–1506, 2005.

[176] Patrick D. Schloss and Jo Handelsman. Introducing SONS, a tool for operational taxonomic unit-based comparisons of microbial community memberships and structures. *Applied and Environmental Microbiology*, 72(10):6773–6779, 2006.

[177] Patrick D Schloss, Sarah L Westcott, Thomas Ryabin, Justine R Hall, Martin Hartmann, Emily B Hollister, Ryan A Lesniewski, Brian B Oakley, Donovan H Parks, Courtney J Robinson, Jason W Sahl, Blaz Stres, Gerhard G Thallinger, David J Van Horn, and Carolyn F Weber. Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Applied and environmental microbiology*, 75(23):7537–41, dec 2009.

[178] Angela Sessitsch and Birgit Mitter. 21st century agriculture: integration of plant microbiomes for improved crop production and food security. *Microbial Biotechnology*, 8(1):32–33, jan 2015.

[179] Ashley Shade, Stuart E Jones, J Gregory Caporaso, Jo Handelsman, Rob Knight, Noah Fierer, and Jack a Gilbert. Conditionally rare taxa disproportionately contribute to temporal changes in microbial diversity. *mBio*, 5(4):e01371–14, 2014.

[180] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S. Baliga, Jonathan T. Wang, Daniel Ramage, Nada Amin, Beno Schwikowski, and Trey Ideker.

Cytoscape: A software Environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11):2498–2504, 2003.

[181] Rashmi Sinha, Christian C Abnet, Owen White, Rob Knight, and Curtis Huttenhower. The microbiome quality control project: baseline study design and future directions. *Genome Biology*, 16:276, dec 2015.

[182] Rashmi Sinha, Galeb Abu-Ali, Emily Vogtmann, Anthony A Fodor, Boyu Ren, Amnon Amir, Emma Schwager, Jonathan Crabtree, Siyuan Ma, Christian C Abnet, Rob Knight, Owen White, and Curtis Huttenhower. Assessment of variation in microbial community amplicon sequencing by the Microbiome Quality Control (MBQC) project consortium. *Nature Biotechnology*, 2017.

[183] Rashmi Sinha, Jun Chen, Amnon Amir, Emily Vogtmann, Jianxin Shi, Kristin S Inman, Roberto Flores, Joshua Sampson, Rob Knight, and Nicholas Chia. Collecting Fecal Samples for Microbiome Analyses in Epidemiology Studies. *Cancer Epidemiology Biomarkers & Prevention*, 25(2):407 LP – 416, feb 2016.

[184] Michelle I. Smith, Tanya Yatsunenko, Mark J. Manary, Indi Trehan, Rajhab Mkakosya, Jiye Cheng, Andrew L. Kau, Stephen S. Rich, Patrick Concannon, Josyf C. Mychaleckyj, Jie Liu, Eric Houpt, Jia V. Li, Elaine Holmes, Jeremy Nicholson, Dan Knights, Luke K. Ursell, Rob Knight, and Jeffrey I. Gordon. Gut microbiomes of Malawian twin pairs discordant for kwashiorkor. *Science*, 339(6119):548–554, 2013.

[185] Temple F. Smith and Michael S. Waterman. Comparison of biosequences. *Advances in Applied Mathematics*, 2(4):482–489, 1981.

[186] PHA Sneath and RR Sokal. *Numerical taxonomy: the principles and practice of numerical classification.* Freeman, San Francisco, 1973.

[187] David A W Soergel, Neelendu Dey, Rob Knight, and Steven E. Brenner. Selection of primers for optimal taxonomic classification of environmental 16S rRNA gene sequences. *ISME Journal*, 6(7):1440–1444, 2012.

[188] RR Sokal and PHA Sneath. *Principles of numerical taxonomy.* Freeman, San Francisco, 1963.

[189] Se Jin Song, Amnon Amir, Jessica L Metcalf, Katherine R Amato, Zhenjiang Zech Xu, Greg Humphrey, and Rob Knight. Preservation Methods Differ in Fecal Microbiome Stability, Affecting Suitability for Field Studies. *mSystems*, 1(3), jun 2016.

[190] Erica D. Sonnenburg, Samuel A. Smits, Mikhail Tikhonov, Steven K. Higginbottom, Ned S. Wingreen, and Justin L. Sonnenburg. Diet-induced extinctions in the gut microbiota compound over generations. *Nature*, 529(7585):212–215, 2016.

[191] Melanie D Spencer, Timothy J Hamp, Robert W Reid, Leslie M Fischer, Steven H Zeisel, and Anthony A Fodor. Association between composition of the human gastrointestinal microbiome and development of fatty liver with choline deficiency. *Gastroenterology*, 140(3):976–986, 2011.

[192] Susan J. Sprague, Michelle Watt, John A. Kirkegaard, and Barbara J. Howlett. Pathways of infection of Brassica napus roots by Leptosphaeria maculans. *New Phytologist*, 176(1):211–222, 2007.

[193] A. Stamatakis, T. Ludwig, and H. Meier. RAxML-III: A fast program for maximum likelihood-based inference of large phylogenetic trees. *Bioinformatics*, 21(4):456–463, 2005.

[194] Lloyd W. Sumner, Alexander Amberg, Dave Barrett, Michael H. Beale, Richard Beger, Clare A. Daykin, Teresa W.M. Fan, Oliver Fiehn, Roysston Goodacre, Julian L. Griffin, Thomas Hankemeier, Nigel Hardy, James Harnly, Richard Higashi, Joachim Kopka, Andrew N. Lane, John C. Lindon, Philip Marriott, Andrew W. Nicholls, Michael D. Reily, John J. Thaden, and Mark R. Viant. Proposed minimum reporting standards for chemical analysis: Chemical Analysis Working Group (CAWG) Metabolomics Standards Initiative (MSI). *Metabolomics*, 3(3):211–221, 2007.

[195] Neslihan Tas, Emmanuel Prestat, Jack W McFarland, Kimberley P Wickland, Rob Knight, Asmeret Asefaw Berhe, Torre Jorgenson, Mark P Waldrop, and Janet K Jansson. Impact of fire on active layer and permafrost microbial communities and metagenomes in an upland Alaskan boreal forest. *ISME J*, 8(9):1904–1919, sep 2014.

[196] The Human Microbiome Project Consortium. A framework for human microbiome research. *Nature*, 486(7402):215–21, 2012.

[197] Luke R Thompson, Jon G Sanders, Daniel McDonald, Amnon Amir, Joshua Ladau, Kenneth J Locey, Robert J Prill, Anupriya Tripathi, Sean M Gibbons, Gail Ackermann, Jose A Navas-Molina, Stefan Janssen, Evguenia Kopylova, Yoshiki Vázquez-Baeza, Antonio González, James T Morton, Siavash Mirarab, Zhenjiang Zech Xu, Lingjing Jiang, Mohamed F Haroon, Jad Kanbar, Qiyun Zhu, Se Jin Song, Tomasz Kosciolek, Nicholas A Bokulich, Joshua Lefler, Colin J Brislawn, Gregory Humphrey, Sarah M Owens, Jarrad Hampton-Marcell, Donna Berg-Lyons, Valerie McKenzie, Noah Fierer, Jed A

Fuhrman, Aaron Clauset, Rick L Stevens, Ashley Shade, Katherine S Pollard, Kelly D Goodwin, Janet K Jansson, Jack A Gilbert, Rob Knight, and The Earth Microbiome Project Consortium. A communal catalogue reveals Earth's multiscale microbial diversity. *Nature*, advance on, nov 2017.

[198] Julien Tremblay, Kanwar Singh, Alison Fern, Edward S Kirton, Shaomei He, Tanja Woyke, Janey Lee, Feng Chen, Jeffery L Dangl, and Susannah G Tringe. Primer and platform effects on 16S rRNA tag sequencing. *Frontiers in Microbiology*, 6:771, aug 2015.

[199] Peter J Turnbaugh, Micah Hamady, Tanya Yatsunenko, Brandi L Cantarel, Alexis Duncan, Ruth E Ley, Mitchell L Sogin, William J Jones, Bruce A Roe, Jason P Affourtit, Michael Egholm, Bernard Henrissat, Andrew C Heath, Rob Knight, and Jeffrey I Gordon. A core gut microbiome in obese and lean twins. *Nature*, 457(7228):480–484, 2009.

[200] Peter J Turnbaugh, Ruth E Ley, Micah Hamady, Claire Fraser-Liggett, Rob Knight, and Jeffrey I Gordon. The human microbiome project: exploring the microbial part of ourselves in a changing world. *Nature*, 449(7164):804–810, oct 2007.

[201] Peter J Turnbaugh, Ruth E Ley, Michael A Mahowald, Vincent Magrini, Elaine R Mardis, and Jeffrey I Gordon. An obesity-associated gut microbiome with increased capacity for energy harvest. *Nature*, 444(7122):1027–1131, dec 2006.

[202] Peter J Turnbaugh, Vanessa K Ridaura, Jeremiah J Faith, Federico E Rey, Rob Knight, and Jeffrey I Gordon. The Effect of Diet on the Human Gut Microbiome: A Metagenomic Analysis in Humanized Gnotobiotic Mice. *Science Translational Medicine*, 1(6):6ra14 LP – 6ra14, nov 2009.

[203] Johan E.T. van Hylckama Vlieg, Patrick Veiga, Chenhong Zhang, Muriel Derrien, and Liping Zhao. Impact of microbial transformation of food on health-from fermented foods to fermentation in the gastro-intestinal tract, 2011.

[204] Els van Nood, Anne Vrieze, Max Nieuwdorp, Susana Fuentes, Erwin G Zoetendal, Willem M de Vos, Caroline E Visser, Ed J Kuijper, Joep F W M Bartelsman, Jan G P Tijssen, Peter Speelman, Marcel G W Dijkgraaf, and Josbert J Keller. Duodenal Infusion of Donor Feces for Recurrent Clostridium difficile. *New England Journal of Medicine*, 368(5):407–415, jan 2013.

[205] Yoshiki Vázquez-Baeza, Embriette R Hyde, Jan S Suchodolski, and Rob Knight. Dog and human inflammatory bowel disease rely on overlapping yet distinct dysbiosis networks. 1:16177, oct 2016.

[206] Yoshiki Vázquez-Baeza, Meg Pirrung, Antonio Gonzalez, and Rob Knight. EMPeror: a tool for visualizing high-throughput microbial community data. *GigaScience*, 2(1):16, 2013.

[207] Maria Vinaixa, Emma L. Schymanski, Steffen Neumann, Miriam Navarro, Reza M. Salek, and Oscar Yanes. Mass spectral databases for LC/MS- and GC/MS-based metabolomics: State of the field and future prospects, 2016.

[208] A. J A Vinten, R. R E Artz, N. Thomas, J. M. Potts, L. Avery, S. J. Langan, H. Watson, Y. Cook, C. Taylor, C. Abel, E. Reid, and B. K. Singh. Comparison of microbial community assays for the assessment of stream biofilm ecology. *Journal of Microbiological Methods*, 85(3):190–198, 2011.

[209] Paola Vitaglione, Ilario Mennella, Rosalia Ferracane, Angela A Rivellese, Rosalba Giacco, Danilo Ercolini, Sean M Gibbons, Antonietta La Storia, Jack A Gilbert, Satya Jonnalagadda, Frank Thielecke, Maria A Gallo, Luca Scalfi, and Vincenzo Fogliano. Whole-grain wheat consumption reduces inflammation in a randomized controlled trial on overweight and obese subjects with unhealthy dietary and lifestyle behaviors: role of polyphenols bound to cereal dietary fiber. *The American journal of clinical nutrition*, 101(2):251–61, feb 2015.

[210] Antonina A Votintseva, Phelim Bradley, Louise Pankhurst, Carlos del Ojo Elias, Matthew Loose, Kayzad Nilgiriwala, Anirvan Chatterjee, E Grace Smith, Nicolas Sanderson, Timothy M Walker, Marcus R Morgan, David H Wyllie, A Sarah Walker, Tim E A Peto, Derrick W Crook, and Zamin Iqbal. Same-Day Diagnostic and Surveillance Data for Tuberculosis via Whole-Genome Sequencing of Direct Respiratory Samples. *Journal of Clinical Microbiology*, 55(5):1285–1298, may 2017.

[211] William A. Walters, J. Gregory Caporaso, Christian L. Lauber, Donna Berg-Lyons, Noah Fierer, and Rob Knight. PrimerProspector: De novo design and taxonomic analysis of barcoded polymerase chain reaction primers. *Bioinformatics*, 27(8):1159–1161, 2011.

[212] Mingxun Wang, Jeremy J. Carver, Vanessa V. Phelan, Laura M. Sanchez, Neha Garg, Yao Peng, Don Duy Nguyen, Jeramie Watrous, Clifford A. Kapono, Tal Luzzatto-Knaan, Carla Porto, Amina Bouslimani, Alexey V. Melnik, Michael J. Meehan, Wei Ting Liu, Max Crüsemann, Paul D. Boudreau, Eduardo Esquenazi, Mario Sandoval-Calderón, Roland D. Kersten, Laura A. Pace, Robert A. Quinn, Katherine R. Duncan, Cheng Chih Hsu, Dimitrios J. Floros, Ronnie G. Gavilan, Karin Kleigrewe, Trent Northen, Rachel J. Dutton, Delphine Parrot, Erin E. Carlson, Bertrand Aigle, Charlotte F. Michelsen, Lars Jelsbak, Christian Sohlenkamp, Pavel Pevzner, Anna Edlund, Jeffrey McLean, Jörn Piel, Brian T. Murphy, Lena

Gerwick, Chih Chuang Liaw, Yu Liang Yang, Hans Ulrich Humpf, Maria Maansson, Robert A. Keyzers, Amy C. Sims, Andrew R. Johnson, Ashley M. Sidebottom, Brian E. Sedio, Andreas Klitgaard, Charles B. Larson, Cristopher A.P. Boya, Daniel Torres-Mendoza, David J. Gonzalez, Denise B. Silva, Lucas M. Marques, Daniel P. Demarque, Egle Pociute, Ellis C. O'Neill, Enora Briand, Eric J.N. Helfrich, Eve A. Granatosky, Evgenia Glukhov, Florian Ryffel, Hailey Houson, Hosein Mohimani, Jenan J. Kharbush, Yi Zeng, Julia A. Vorholt, Kenji L. Kurita, Pep Charusanti, Kerry L. McPhail, Kristian Fog Nielsen, Lisa Vuong, Maryam Elfeki, Matthew F. Traxler, Niclas Engene, Nobuhiro Koyama, Oliver B. Vining, Ralph Baric, Ricardo R. Silva, Samantha J. Mascuch, Sophie Tomasi, Stefan Jenkins, Venkat Macherla, Thomas Hoffman, Vinayak Agarwal, Philip G. Williams, Jingqui Dai, Ram Neupane, Joshua Gurr, Andrés M.C. Rodríguez, Anne Lamsa, Chen Zhang, Kathleen Dorrestein, Brendan M. Duggan, Jehad Almaliti, Pierre Marie Allard, Prasad Phapale, Louis Felix Nothias, Theodore Alexandrov, Marc Litaudon, Jean Luc Wolfender, Jennifer E. Kyle, Thomas O. Metz, Tyler Peryea, Dac Trung Nguyen, Danielle VanLeer, Paul Shinn, Ajit Jadhav, Rolf Müller, Katrina M. Waters, Wenyuan Shi, Xueting Liu, Lixin Zhang, Rob Knight, Paul R. Jensen, Bernhard Palsson, Kit Pogliano, Roger G. Linington, Marcelino Gutiérrez, Norberto P. Lopes, William H. Gerwick, Bradley S. Moore, Pieter C. Dorrestein, and Nuno Bandeira. Sharing and community curation of mass spectrometry data with Global Natural Products Social Molecular Networking. 34(8):828–837, 2016.

[213] Qiong Wang, George M. Garrity, James M. Tiedje, and James R. Cole. Naïve Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. *Applied and Environmental Microbiology*, 73(16):5261–5267, 2007.

[214] Zeneng Wang, Elizabeth Klipfell, Brian J. Bennett, Robert Koeth, Bruce S. Levison, Brandon Dugar, Ariel E. Feldstein, Earl B. Britt, Xiaoming Fu, Yoon Mi Chung, Yuping Wu, Phil Schauer, Jonathan D. Smith, Hooman Allayee, W. H.Wilson Tang, Joseph A. Didonato, Aldons J. Lusis, and Stanley L. Hazen. Gut flora metabolism of phosphatidylcholine promotes cardiovascular disease. *Nature*, 472(7341):57–65, 2011.

[215] J. Watrous, P. Roach, T. Alexandrov, B. S. Heath, J. Y. Yang, R. D. Kersten, M. van der Voort, K. Pogliano, H. Gross, J. M. Raaijmakers, B. S. Moore, J. Laskin, N. Bandeira, and P. C. Dorrestein. Mass spectral molecular networking of living microbial colonies. *Proceedings of the National Academy of Sciences*, 109(26):E1743–E1752, 2012.

[216] Alexa Weingarden, Antonio González, Yoshiki Vázquez-Baeza, Sophie Weiss, Gregory Humphry, Donna Berg-Lyons, Dan Knights, Tatsuya Unno, Aleh

Bobr, Johnthomas Kang, Alexander Khoruts, Rob Knight, and Michael J Sadowsky. Dynamic changes in short- and long-term bacterial composition following fecal microbiota transplantation for recurrent Clostridium difficile infection. *Microbiome*, 3:10, mar 2015.

[217] K A Wetterstrand. DNA Sequencing Costs: Data from the NHGRI Large-Scale Genome Sequencing Program, 2013.

[218] Hadley Wickham. *ggplot2 Elegant Graphics for Data Analysis*, volume 35. 2009.

[219] Leland Wilkinson and Michael Friendly. History corner the history of the cluster heat map, 2009.

[220] Dan Williams, Hani Jamjoom, Yew-Huey Liu, and Hakim Weatherspoon. Overdriver: handling memory overload in an oversubscribed cloud. *ACM SIGPLAN Notices*, 46(7):205, 2011.

[221] Benjamin E. Wolfe and Rachel J. Dutton. Fermented foods as experimentally tractable microbial ecosystems. *Cell*, 161(1):49–55, 2015.

[222] Yihui Xie. *Dynamic Documents with R and knitr*. 2013.

[223] Jane Y. Yang, Laura M. Sanchez, Christopher M. Rath, Xueting Liu, Paul D. Boudreau, Nicole Bruns, Evgenia Glukhov, Anne Wodtke, Rafael De Felicio, Amanda Fenner, Weng Ruh Wong, Roger G. Linington, Lixin Zhang, Hosana M. Debonsi, William H. Gerwick, and Pieter C. Dorrestein. Molecular networking as a dereplication strategy. *Journal of Natural Products*, 76(9):1686–1699, 2013.

[224] Tanya Yatsunenko, Federico E. Rey, Mark J. Manary, Indi Trehan, Maria Gloria Dominguez-Bello, Monica Contreras, Magda Magris, Glida Hidalgo, Robert N. Baldassano, Andrey P. Anokhin, Andrew C. Heath, Barbara Warner, Jens Reeder, Justin Kuczynski, J. Gregory Caporaso, Catherine A. Lozupone, Christian Lauber, Jose Carlos Clemente, Dan Knights, Rob Knight, and Jeffrey I. Gordon. Human gut microbiome viewed across age and geography. 2012.

[225] Pelin Yilmaz, Renzo Kottmann, Dawn Field, Rob Knight, James R Cole, Linda Amaral-Zettler, Jack a Gilbert, Ilene Karsch-Mizrachi, Anjanette Johnston, Guy Cochrane, Robert Vaughan, Christopher Hunter, Joonhong Park, Norman Morrison, Philippe Rocca-Serra, Peter Sterk, Manimozhiyan Arumugam, Mark Bailey, Laura Baumgartner, Bruce W Birren, Martin J Blaser, Vivien Bonazzi, Tim Booth, Peer Bork, Frederic D Bushman, Pier Luigi Buttigieg, Patrick S G Chain, Emily Charlson, Elizabeth K Costello, Heather Huot-Creasy, Peter Dawyndt, Todd DeSantis,

Noah Fierer, Jed a Fuhrman, Rachel E Gallery, Dirk Gevers, Richard a Gibbs, Inigo San Gil, Antonio Gonzalez, Jeffrey I Gordon, Robert Guralnick, Wolfgang Hankeln, Sarah Highlander, Philip Hugenholtz, Janet Jansson, Andrew L Kau, Scott T Kelley, Jerry Kennedy, Dan Knights, Omry Koren, Justin Kuczynski, Nikos Kyrpides, Robert Larsen, Christian L Lauber, Teresa Legg, Ruth E Ley, Catherine a Lozupone, Wolfgang Ludwig, Donna Lyons, Eamonn Maguire, Barbara a Methé, Folker Meyer, Brian Muegge, Sara Nakielny, Karen E Nelson, Diana Nemergut, Josh D Neufeld, Lindsay K Newbold, Anna E Oliver, Norman R Pace, Giriprakash Palanisamy, Jörg Peplies, Joseph Petrosino, Lita Proctor, Elmar Pruesse, Christian Quast, Jeroen Raes, Sujeevan Ratnasingham, Jacques Ravel, David a Relman, Susanna Assunta-Sansone, Patrick D Schloss, Lynn Schriml, Rohini Sinha, Michelle I Smith, Erica Sodergren, Aymé Spo, Jesse Stombaugh, James M Tiedje, Doyle V Ward, George M Weinstock, Doug Wendel, Owen White, Andrew Whiteley, Andreas Wilke, Jennifer R Wortman, Tanya Yatsunenko, and Frank Oliver Glöckner. Minimum information about a marker gene sequence (MIMARKS) and minimum information about any (x) sequence (MIxS) specifications. *Nature biotechnology*, 29(5):415–20, 2011.

[226] David Zeevi, Tal Korem, Niv Zmora, David Israeli, Daphna Rothschild, Adina Weinberger, Orly Ben-Yacov, Dar Lador, Tali Avnit-Sagi, Maya Lotan-Pompan, Jotham Suez, Jemal Ali Mahdi, Elad Matot, Gal Malka, Noa Kosower, Michal Rein, Gili Zilberman-Schapira, Lenka Dohnalová, Meirav Pevsner-Fischer, Rony Bikovsky, Zamir Halpern, Eran Elinav, and Eran Segal. Personalized Nutrition by Prediction of Glycemic Responses. *Cell*, 163(5):1079–1094, oct 2017.