# QIIME 2: Reproducible, interactive, scalable, and extensible microbiome data science

Evan Bolyen[1,*], Jai Ram Rideout[1,*], Matthew R Dillon[1,*], Nicholas A Bokulich[1,*], Christian C Abnet[2], Gabriel A Al-Ghalith[3], Harriet Alexander[4,5], Eric J Alm[6,7], Manimozhiyan Arumugam[8], Francesco Asnicar[9], Yang Bai[10,11,12], Jordan E Bisanz[13], Kyle Bittinger[14,15], Asker Brejnrod[16], Colin J Brislawn[17], C Titus Brown[5], Benjamin J Callahan[18,19], Andrés Mauricio Caraballo-Rodríguez[20], John Chase[1], Emily K Cope[1,21], Ricardo Da Silva[20], Pieter C Dorrestein[20], Gavin M Douglas[22], Daniel M Durall[23], Claire Duvallet[6], Christian F Edwardson[24], Madeleine Ernst[20], Mehrbod Estaki[25], Jennifer Fouquier[26,27], Julia M Gauglitz[20], Deanna L Gibson[28,29], Antonio Gonzalez[30], Kestrel Gorlick[1], Jiarong Guo[31], Benjamin Hillmann[32], Susan Holmes[33], Hannes Holste[30,34], Curtis Huttenhower[35,36], Gavin A Huttley[37], Stefan Janssen[38], Alan K Jarmusch[20], Lingjing Jiang[39], Benjamin D Kaehler[37], Kyo Bin Kang[40,20], Christopher R Keefe[1], Paul Keim[1], Scott T Kelley[41], Dan Knights[42,32], Irina Koester[43,20], Tomasz Kosciolek[44], Jorden Kreps[1], Morgan GI Langille[45], Joslynn Lee[46], Ruth Ley[47,48], Yong-Xin Liu[10,11], Erikka Loftfield[2], Catherine Lozupone[49], Massoud Maher[50], Clarisse Marotz[30], Bryan D Martin[51], Daniel McDonald[30], Lauren J McIver[35,36], Alexey V Melnik[20], Jessica L Metcalf[52], Sydney C Morgan[53], Jamie T Morton[30,50], Ahmad Turan Naimey[1], Jose A Navas-Molina[50,30,54], Louis Felix Nothias[20], Stephanie B Orchanian[55], Talima Pearson[1], Samuel L Peoples[56,57], Daniel Petras[20], Mary Lai Preuss[58], Elmar Pruesse[49], Lasse Buur Rasmussen[16], Adam Rivers[59], Michael S Robeson, II[60], Patrick Rosenthal[58], Nicola Segata[9], Michael Shaffer[49,61], Arron Shiffer[1], Rashmi Sinha[2], Se Jin Song[30], John R Spear[62], Austin D Swafford[55], Luke R Thompson[63,64], Pedro J Torres[65], Pauline Trinh[66], Anupriya Tripathi[20,30,67], Peter J Turnbaugh[68], Sabah Ul-Hasan[69], Justin JJ van der Hooft[70], Fernando Vargas[67], Yoshiki Vázquez-Baeza[30], Emily Vogtmann[2], Max von Hippel[71], William Walters[47], Yunhu Wan[2], Mingxun Wang[20], Jonathan Warren[72], Kyle C Weber[59,73], Charles HD Williamson[1], Amy D Willis[74], Zhenjiang Zech Xu[30], Jesse R Zaneveld[75], Yilong Zhang[76], Qiyun Zhu[30], Rob Knight[30,77,55], and J Gregory Caporaso[1,21,+]

* These authors contributed equally to this work.

[+] Please address correspondence to gregcaporaso@gmail.com.

Author affiliations are provided following the Main Text References section.

To get help with QIIME 2, visit https://forum.qiime2.org.

## Abstract

We present QIIME 2, an open-source microbiome data science platform accessible to users spanning the microbiome research ecosystem, from scientists and engineers to clinicians and policy makers. QIIME 2 provides new features that will drive the next generation of microbiome research. These include interactive spatial and temporal analysis and visualization tools, support for metabolomics and shotgun metagenomics analysis, and automated data provenance tracking to ensure reproducible, transparent microbiome data science.

# Main text

Rapid advances in DNA sequencing and bioinformatics technologies in the past two decades have significantly improved our understanding of the microbial world. These include our growing understanding of the vast diversity of microorganisms; how our microbiota and microbiomes impact disease[1] and medical treatment[2]; how microorganisms impact the health of our planet[3]; and our nascent exploration of the medical[4], forensic[5], environmental[6], and agricultural[7] applications of microbiome biotechnology. Much of this work has been driven by marker gene surveys (e.g., bacterial/archaeal 16S rRNA genes, fungal ITS, eukaryal 18S rRNA genes), which profile microbiota with varying degrees of taxonomic specificity and phylogenetic information. The field is now transitioning to integrate other data types, such as metabolite[8] or metatranscriptome[9] profiles.

The QIIME 1 microbiome bioinformatics platform has supported many microbiome studies and gained a broad user and developer community. Interactions with QIIME 1 users in our online support forum, our workshops, and direct collaborations showed the potential to better serve an increasingly diverse array of microbiome researchers in academia, government, and industry. Here we present QIIME 2, a completely reengineered and rewritten system that will facilitate reproducible and modular analysis of microbiome data to enable the next generation of microbiome science.

QIIME 2 is developed based on a plugin architecture (Figure S1) that allows third-parties to contribute functionality (see https://library.qiime2.org). QIIME 2 plugins exist for latest-generation tools for sequence quality control from different sequencing platforms (DADA2[10] and Deblur[11]), taxonomy assignment[12], and phylogenetic insertion[13], that quantitatively improve results over QIIME 1 and other tools (detailed in the corresponding tool-specific publications). Plugins also support qualitatively new functionality including microbiome paired-sample and time-series analysis[14], critical for studying the impact of treatment on the microbiome, and for machine learning[15], including the ability to save trained models and apply them to new data and to interrogate models to identify important microbiome features. Several recently released plugins, including q2-cscs[16], q2-metabolomics[17], q2-shogun[18], q2-metaphlan2[19], and q2-picrust2[20], provide initial support for analysis of metabolomics and shotgun metagenomics data. This marks the potential of QIIME 2 to serve not only as a marker gene analysis tool, but also a multi-dimensional and powerful data science platform that can be rapidly adapted to analyze diverse microbiome features.

QIIME 2 provides many new interactive visualization tools facilitating exploratory analyses and result reporting. Static versions of interactive visualizations resulting from four worked examples are provided in Figure 1. QIIME 2 View (https://view.qiime2.org) is a unique new service (see Online Methods) that allows users to securely share and interact with results without installing QIIME 2. The QIIME 2 visualizations presented in Figure 1 are provided in Supplementary File 1 for readers to interact with using QIIME 2 View. Corresponding worked QIIME 2 example code is provided in Supplementary File 2.

Reproducibility, transparency, and clarity of microbiome data science are guiding principles in the QIIME 2 design. Toward this end, it includes a decentralized data provenance tracking system: details of all analysis steps with references to intermediate data are automatically stored in the results. Users can thus retrospectively determine exactly how any result was generated (Figure 2). QIIME 2 also detects corrupted results, indicating that provenance is no longer reliable and the results no longer contain information enabling reproducibility. Provenance of the visualizations presented in Figure 1 can be interactively reviewed by loading the contents of Supplementary File 1 with QIIME 2 View, providing far more detailed information than can typically be provided in Methods text. QIIME 2 results are also semantically typed (Figure 2) and actions indicate acceptable input types, clarifying the data that actions should be applied to and making complex workflows less error-prone.

Finally, QIIME 2 provides a software development kit (see https://dev.qiime2.org) that can be used to integrate it as a component of other systems (e.g., such as Qiita[21] or Illumina BaseSpace) and to develop interfaces targeted toward users with different levels of computational sophistication (Figure S2). QIIME 2 provides the QIIME 2 Studio graphical user interface and QIIME 2 View, interfaces designed for end-user biologists, clinicians, and policy makers; the QIIME 2 application programming interface, designed for data scientists who want to automate workflows or work interactively in Jupyter Notebooks; and q2cli and q2cwl, providing a command line interface and Common Workflow Language[22] wrappers for QIIME 2, designed for high-performance computing experts.

There are many other powerful open source software tools for microbiome data science, including mothur[23], phyloseq[24] and related tools available through Bioconductor[25], and the biobakery suite[19,20,26]. mothur is a microbiome bioinformatics platform that is often compared to QIIME 1 and QIIME 2. A major difference between the two lies in the interactive visualizations: QIIME 2 provides many interactive visualization tools (several examples are provided in Figure 1), while mothur focuses on generating data that can be easily loaded and visualized with other tools. phyloseq focuses on microbiome statistical analysis and generating publication-ready visualizations but, unlike QIIME 2, begins with a feature or OTU table, leaving "upstream" processing steps such as sequence demultiplexing and quality control to other processing pipelines, many of which (like phyloseq) are available through Bioconductor. The biobakery suite provides analytic functionality that complements that of QIIME 2, and we are actively working with biobakery developers to support interoperability by making their tools accessible as QIIME 2 plugins (for example, the q2-metaphlan2 plugin allows users to run MetaPhlAn2 through QIIME 2). QIIME 2 provides the only Python-based microbiome data science platform that supports retrospective data provenance tracking to ensure reproducibility, multi-omics analysis support, interfaces geared toward different user types to enhance usability, and an extensibility-focused design through the plugin architecture and software development kit. We share feedback from users of QIIME 2 on these and other features in Supplementary File 3.

The tools described in the preceding paragraph are all interoperable through plugins, exchange of files in standard formats, or using multi-language environments such as Jupyter Notebooks[27]. For example, the BIOM format[28] is supported by all of them. A diverse ecosystem of interoperable software is beneficial for the field, as it allows experienced users to get multiple perspectives on their data and novice bioinformaticians to work in programming environments that they are most comfortable with (e.g., phyloseq allows users to work in R, while QIIME 2 allows users to work in Python). We plan to continue working with the developers of these tools and organizations such as the Genomics Standards Consortium on plugins and standards to ensure interoperability.

Advances in microbiome research promise to improve many aspects of our health and our world, and QIIME 2 will help drive those advances by enabling accessible, community-driven microbiome data science.

# Figures and figure captions

**Figure 1:** QIIME 2 provides many interactive visualization tools. The products of four worked examples are presented here, and interactive versions of these screen captures are available in Supplementary File 1 and at https://github.com/qiime2/paper1. Detailed descriptions and methods, including the commands used to generate each of these visualizations, are provided in Online Methods. (A) Unweighted UniFrac PCoA plot containing 37,680 samples, illustrating the scalability of QIIME 2. Colors indicate sample type as described by the Earth Microbiome Project ontology (EMPO). (B) A feature volatility plot illustrating change in Bifidobacterium abundance over time in breast-fed and formula-fed infants. Temporally interesting features can be interactively discovered with this visualization. (C) Interactive taxonomic composition bar plot illustrating phylum-level composition of microbial mat samples collected along a temperature gradient in Yellowstone National Park Hot Spring outflow channels (Steep Cone Geyser). The many interactive controls available in this plot vastly reduce the burden of exploratory analysis over QIIME 1. (D) Molecular cartography of the human skin surface. Colored spots represent the abundance of the small molecule cosmetic, sodium laureth sulfate, on the human skin. Sample data can be interactively visualized on 3D models, supporting the discovery of spatial patterns.

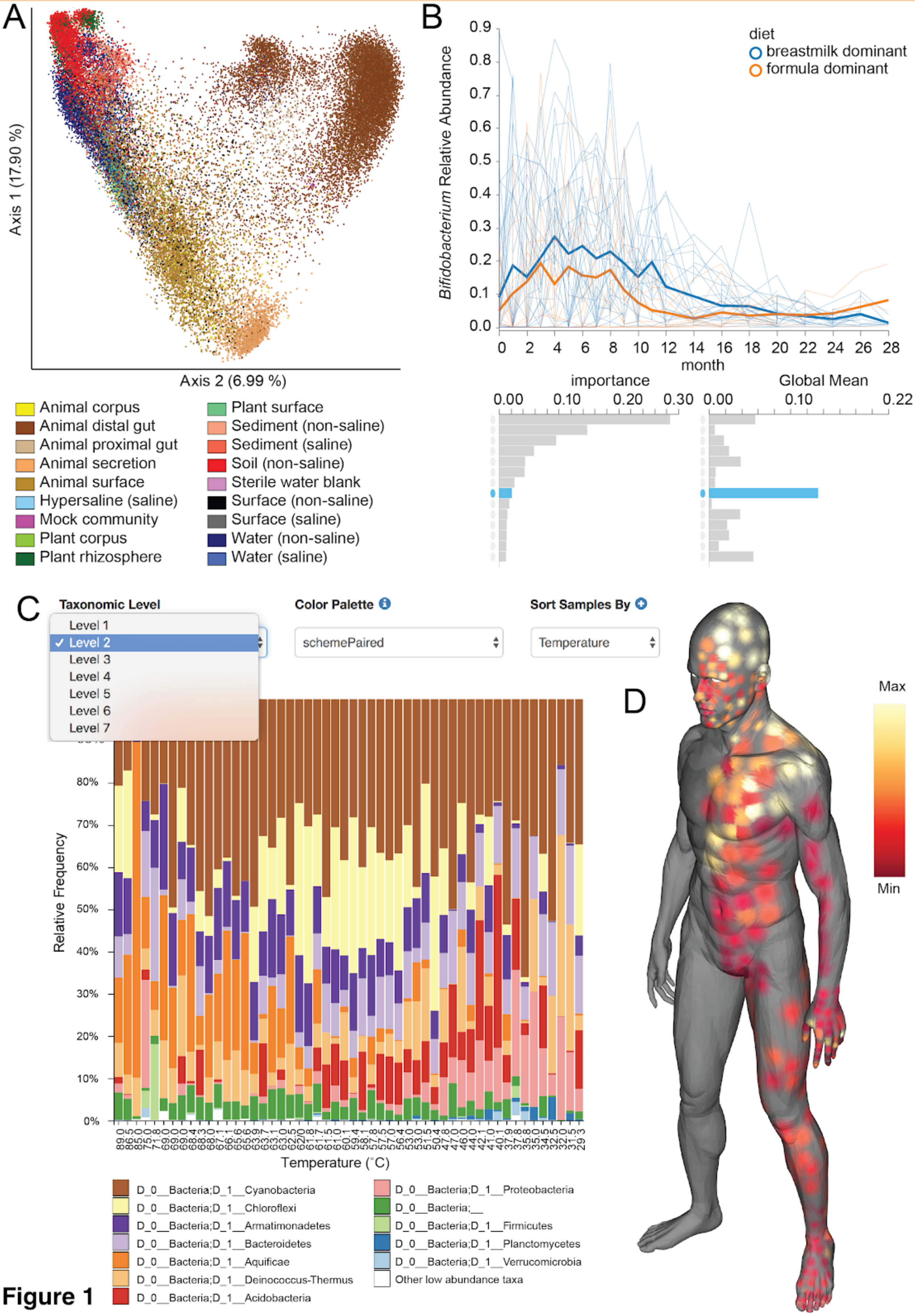

A
Axis 1 (17.90 %)
Axis 2 (6.99 %)
Animal corpus
Animal distal gut
Animal proximal gut
Animal secretion
Animal surface
Hypersaline (saline)
Mock community
Plant corpus
Plant rhizosphere
Plant surface
Sediment (non-saline)
Sediment (saline)
Soil (non-saline)
Sterile water blank
Surface (non-saline)
Surface (saline)
Water (non-saline)
Water (saline)
B
diet
breastmilk dominant
formula dominant
*Bifidobacterium* Relative Abundance
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28
month
importance
0.00 0.10 0.20 0.30
Global Mean
0.00 0.10 0.22
C
Taxonomic Level
Level 1
Level 2
Level 3
Level 4
Level 5
Level 6
Level 7
Color Palette
schemePaired
Sort Samples By
Temperature
Relative Frequency
0% 10% 20% 30% 40% 50% 60% 70% 80%
Temperature (°C)
D_0__Bacteria;D_1__Cyanobacteria
D_0__Bacteria;D_1__Chloroflexi
D_0__Bacteria;D_1__Armatimonadetes
D_0__Bacteria;D_1__Bacteroidetes
D_0__Bacteria;D_1__Aquificae
D_0__Bacteria;D_1__Deinococcus-Thermus
D_0__Bacteria;D_1__Acidobacteria
D_0__Bacteria;D_1__Proteobacteria
D_0__Bacteria;__
D_0__Bacteria;D_1__Firmicutes
D_0__Bacteria;D_1__Planctomycetes
D_0__Bacteria;D_1__Verrucomicrobia
Other low abundance taxa
D
Max
Min

**Figure 1**

**Figure 2:** QIIME 2 iteratively records data provenance, ensuring bioinformatics reproducibility. This simplified diagram illustrates the automatically tracked information about the creation of the taxonomy barplot presented in Figure 1c. QIIME 2 results (circles) contain network diagrams illustrating the data provenance stored in the result. Actions (quadrilaterals) are applied to QIIME 2 results and generate new results. Arrows indicate flow of QIIME 2 results through actions. TaxonomicClassifier and FeatureData[Sequence] inputs contain independent provenance (red and blue, respectively) and are provided to a classify action (yellow), which taxonomically annotates sequences. The result of the classify action, a FeatureData[Taxonomy] result, integrates the provenance of both inputs with the classify action. This result is then provided to the barplot action with a FeatureTable[Frequency] input, which shares some provenance with the FeatureData[Sequence] input as they were generated from the same upstream analysis. The resulting Visualization (Figure 1c), has the complete data provenance and correctly identifies shared processing of inputs. An interactive and complete version of this provenance graph (as well as those for other Figures 1 panels) can be accessed through Supplementary File 1.

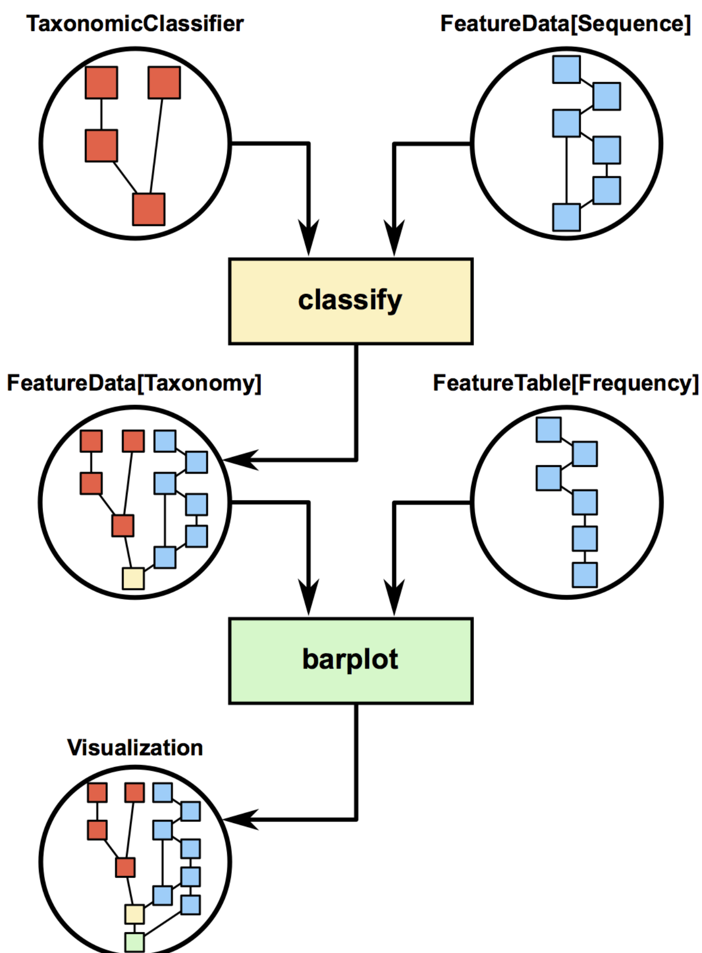



Figure 2

## Code availability

QIIME 2 is open source and free for all use, including commercial. It is licensed under the BSD 3-clause license. Source code is available at https://github.com/qiime2.

## Data availability

Data for the analyses presented in Figure 1 are available as follows: (a) Earth Microbiome Project data was obtained from ftp://ftp.microbio.me/emp/release1, and the American Gut Project (AGP) data was obtained from Qiita (http://qiita.microbio.me) study 10317. (b) Sequence data are available in Qiita under study id 10249 and EBI under accession number ERP016173. (c) Sequence data are available in Qiita under study id 925 and EBI under accession number ERP022167. (d) Data are available in the q2-ili GitHub repository (https://github.com/biocore/q2-ili). Interactive versions of the Figure 1 visualizations can be accessed at https://github.com/qiime2/paper1.

## Acknowledgements

QIIME 2 development was primarily funded by NSF Awards 1565100 to JGC and 1565057 to RK. Partial support was also provided from the following grants: NIH U54CA143925 (JGC, TP) and U54MD012388 (JGC, TP); grants from the Alfred P. Sloan Foundation (JGC, RK); ERC-STG project MetaPG (NS); Strategic Priority Research Program of the Chinese Academy of Sciences QYZDB-SSW-SMC021 (YB); from the Australian National Health and Medical Research Council APP1085372 (GAH, JGC, Von Bing Yap and RK); and from Natural Sciences and Engineering Research Council (NSERC) to DLG; and under the State of Arizona Technology and Research Initiative Fund (TRIF), administered by the Arizona Board of Regents, through Northern Arizona University. All NCI co-authors were supported by the Intramural Research Program of the National Cancer Institute.Thanks to the Yellowstone Center for Resources for research permit #5664 to JRS for Yellowstone access and sample collection. We thank Paul J. McMurdie for helpful discussion on the relationships between QIIME 2 and phyloseq. We would like to thank the users of QIIME 1 and 2, whose invaluable feedback has shaped QIIME 2. In particular, we would like to thank Ahmed Abdelfattah (Stockholm University, Sweden), Rozlyn C.T. Boutin (University of British Columbia, Canada), David J. Bradshaw II (Florida Atlantic University Harbor Branch Oceanographic Institute, USA), Lorinda Bullington (MPG Ranch, USA), Justine W. Debelius (Karolinska Institutet, Sweden), Claire Duvallet (Massachusetts Institute of Technology, USA), Erika Korzune Ganda (Cornell University, USA), Alexander Mahnert (Medical University of Graz, Austria), Melanie C Melendrez (St. Cloud State University, USA), Devon O'rourke (University of New Hampshire, USA), Adam R. Rivers (USDA-ARS, USA), Biswarup Sen (Tianjin University, China), Solveig Tangedal (Haukeland University Hospital and University of Bergen, Norway), Pedro J. Torres (San Diego State University, USA), and Jonathan Warren (National Laboratory Service, UK) for writing end-user reviews included in Supplementary File 3.

## Author contributions

EB, JRR, MRD, NAB, YB, JEB, CJB, AMC, EC, RD, CFE, MEs, JMG, DLG, AKJ, KBK, STK, IK, TK, JL, YL, AVM, JLM, LFN, SBO, DP, AS, SJS, ADS, LRT, PJTo, PJTu, SU, FV, JW, RK, and JGC developed documentation, educational materials, and/or user/developer support content. EB, JRR, MRD, NAB, RK, and

JGC wrote the manuscript; all authors assisted with revision of the manuscript. EB, JRR, MRD, NAB, and JGC designed and developed the QIIME 2 framework. DMD, RL, EL, SCM, RS, JRS, WW, CHDW, and RK contributed data used in the manuscript and/or testing of the QIIME 2. CCA, CTB, EC, PCD, SH, PK, EL, TP, RS, EV, YW, and RK contributed to the design of analytic methods. EB, JRR, MRD, NAB, GAA, HA, EJA, MA, FA, KB, AB, BJC, JC, GMD, CD, MEr, JF, AG, KG, JG, BH, HH, CH, GH, SJ, LJ, BK, CRK, DK, JK, MGIL, CL, MM, CM, BM, DM, LJM, JM, ATN, JAN, SLP, MLP, EP, LBR, AR, MSR, PR, NS, MS, PT, AT, JJJV, YV, MV, MW, KCW, ADW, ZZX, JRZ, YZ, QZ, and JGC contributed software to QIIME 2 plugins, interfaces, framework, and/or build and test systems.

# Main text references


1. Smith, M.I. et al. Science **339**, 548–554 (2013).
2. Gopalakrishnan, V. et al. Science **359**, 97–103 (2018).
3. Gehring, C.A., Sthultz, C.M., Flores-Rentería, L., Whipple, A.V. & Whitham, T.G. Proc. Natl. Acad. Sci. U. S. A. **114**, 11169–11174 (2017).
4. Lee, K., Pletcher, S.D., Lynch, S.V., Goldberg, A.N. & Cope, E.K. Front. Cell. Infect. Microbiol. **8**, 168 (2018).
5. Metcalf, J.L. et al. Science **351**, 158–162 (2016).
6. Rubin, R.L. et al. Ecol. Appl. **28**, 1594–1605 (2018).
7. Pineda, A., Kaplan, I. & Bezemer, T.M. Trends Plant Sci. **22**, 770–778 (2017).
8. Kapono, C.A. et al. Sci. Rep. **8**, 3669 (2018).
9. Barr, T. et al. Gut Microbes 1–44 (2018).
10. Callahan, B.J. et al. Nat. Methods (2016).doi:10.1038/nmeth.3869
11. Amir, A. et al. mSystems **2**, (2017).
12. Bokulich, N.A. et al. Microbiome **6**, 90 (2018).
13. Janssen, S. et al. mSystems **3**, e00021–18 (2018).
14. Bokulich, N.A. et al. mSystems **3**, e00219–18 (2018).
15. Bokulich, N. et al. JOSS **3**, 934 (2018).
16. Sedio, B.E., Rojas Echeverri, J.C., Boya P, C.A. & Wright, S.J. Ecology **98**, 616–623 (2017).
17. Wang, M. et al. Nat. Biotechnol. **34**, 828–837 (2016).
18. Hillmann, B. et al. bioRxiv 320986 (2018).doi:10.1101/320986
19. Truong, D.T. et al. Nat. Methods **12**, 902–903 (2015).
20. Langille, M.G.I. et al. Nat. Biotechnol. **31**, 814–821 (2013).
21. Gonzalez, A. et al. Nat. Methods **15**, 796–798 (2018).
22. Amstutz, P. et al. (2016).doi:10.6084/m9.figshare.3115156.v2
23. Schloss, P.D. et al. Appl. Environ. Microbiol. **75**, 7537–7541 (2009).
24. McMurdie, P.J. & Holmes, S. PLoS One **8**, e61217 (2013).
25. Huber, W. et al. Nat. Methods **12**, 115–121 (2015).
26. Franzosa, E.A. et al. Nat. Methods **15**, 962–968 (2018).
27. Kluyver, T. et al. Positioning and Power in Academic Publishing: Players, Agents and Agendas 87–90 (2016).
28. McDonald, D. et al. Gigascience **1**, 7 (2012).


# Author affiliations


[1]Pathogen and Microbiome Institute, Northern Arizona University, Flagstaff, AZ, USA. [2]Metabolic Epidemiology Branch, National Cancer Institute, Rockville, MD, USA. [3]Department of Computer Science and Engineering, University of Minnesota, Minneapolis, Minnesota, USA. [4]Biology Department, Woods Hole Oceanographic Institution, Woods Hole, MA, USA. [5]Department of Population Health and Reproduction, University of California, Davis, CA, USA. [6]Department of Biological Engineering, Massachusetts Institute of Technology,

Cambridge, MA, USA. [7]Center for Microbiome Informatics and Therapeutics, Massachusetts Institute of Technology, Cambridge, MA, USA. [8]University of Copenhagen, Faculty of Health and Medical Sciences, Novo Nordisk Foundation Center for Basic Metabolic Research, Copenhagen, Denmark. [9]Centre for Integrative Biology, University of Trento, Trento, Italy. [10]State Key Laboratory of Plant Genomics, Institute of Genetics and Developmental Biology, Chinese Academy of Sciences, Beijing, China. [11]Centre of Excellence for Plant and Microbial Sciences (CEPAMS), Institute of Genetics and Developmental Biology, Chinese Academy of Sciences & John Innes Centre, Beijing, China. [12]University of Chinese Academy of Sciences, Beijing, China. [13]Department of Microbiology and Immunology, University of California, San Francisco, CA, USA. [14]Division of Gastroenterology and Nutrition, Children's Hospital of Philadelphia, Philadelphia, PA, USA. [15]Hepatology, Children's Hospital of Philadelphia, Philadelphia, PA, USA. [16]Novo Nordisk Foundation Center for Basic Metabolic Research, Faculty of Health and Medical Sciences, University of Copenhagen, Denmark. [17]Earth and Biological Sciences Directorate, Pacific Northwest National Laboratory, Richland, WA, USA. [18]Department of Population Health & Pathobiology, North Carolina State University, Raleigh, NC, USA. [19]Bioinformatics Research Center, North Carolina State University, Raleigh, NC, USA. [20]Collaborative mass spectrometry innovation center, Skaggs School of Pharmacy and Pharmaceutical Sciences, University of California San Diego, San Diego, CA, USA. [21]Department of Biological Sciences, Northern Arizona University, Flagstaff, AZ, USA. [22]Department of Microbiology and Immunology, Dalhousie University, Halifax, Nova Scotia, Canada. [23]Irving K. Barber School of Arts and Sciences, University of British Columbia, Kelowna, British Columbia, Canada. [24]A. Watson Armour III Center for Animal Health and Welfare, Aquarium Microbiome Project, John G. Shedd Aquarium, Chicago, IL, USA. [25]Department of Biology, University of British Columbia Okanagan, Okanagan, BC, Canada. [26]Computational Bioscience Graduate Program, University of Colorado Denver Anschutz Medical Campus, Aurora, Colorado, USA. [27]Department of Medicine, Division of Biomedical Informatics and Personalized Medicine, University of Colorado Denver Anschutz Medical Campus, Aurora, Colorado, USA. [28]Irving K. Barber School of Arts and Sciences, Department of Biology, The University of British Columbia, Kelowna, BC, Canada. [29]Department of Medicine, The University of British Columbia, Kelowna, BC, Canada. [30]Department of Pediatrics, University of California San Diego, La Jolla, CA, USA. [31]Center for Microbial Ecology, Michigan State University, East Lansing, MI, USA. [32]Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA. [33]Stanford University, Statistics Department, Palo Alto, CA, USA. [34]Department of Computer Science and Engineering, University of California San Diego, La Jolla, CA, USA. [35]Department of Biostatistics, Harvard T.H. Chan School of Public Health, Boston, MA, USA. [36]Broad Institute of MIT and Harvard, Cambridge, MA, USA. [37]Research School of Biology, The Australian National University, Canberra, ACT, Australia. [38]Department of Pediatric Oncology, Hematology and Clinical Immunology, Heinrich-Heine University Dusseldorf, Dusseldorf, Germany. [39]Department of Family Medicine and Public Health, University of California San Diego, La Jolla, CA, USA. [40]College of Pharmacy, Sookmyung Women's University, Seoul, Republic of Korea. [41]San Diego State University, Department of Biology, San Diego, CA, USA. [42]Biotechnology Institute, University of Minnesota, Saint Paul, MN, USA. [43]Scripps Institution of Oceanography, University of California San Diego, La Jolla, CA, USA. [44]Department of Pediatrics, University of California San Diego, La Jolla, CA,. [45]Department of Pharmacology, Dalhousie University, Halifax, Nova Scotia, Canada. [46]Science Education, Howard Hughes Medical Institute, Ashburn, VA, USA. [47]Department of Microbiome Science, Max Planck Institute for Developmental Biology, Tübingen, Germany. [48]Department of Molecular Biology and Genetics, Cornell University, Ithaca, NY, USA. [49]Department of Medicine, Division of Biomedical Informatics and Personalized Medicine, University of Colorado Denver Anschutz Medical Campus, Aurora, CO, USA. [50]Department of Computer Science & Engineering, University of California San Diego, La Jolla, CA, USA. [51]Department of Statistics, University of Washington, Seattle, WA, USA. [52]Department of Animal Science, Colorado State University, Fort Collins, CO, USA. [53]Irving K. Barber School of Arts and Sciences, Unit 2 (Biology), University of British Columbia, Kelowna, BC, Canada. [54]Mountain View, Google LLC, Mountain View, CA, USA. [55]Center for Microbiome Innovation, University of California San Diego, La Jolla, CA, USA. [56]School of Information Studies, Syracuse University, Syracuse, NY, USA. [57]School of STEM, University of Washington Bothell, Bothell, WA, USA. [58]Department of Biological Sciences, Webster University, St Louis, MO, USA. [59]Agricultural Research Service, Genomics and Bioinformatics Research Unit, United States Department of Agriculture, Gainesville, FL, USA. [60]College of Medicine, Department of Biomedical Informatics, University of Arkansas for Medical Sciences, Little Rock, AR, USA. [61]Computational Bioscience Program, University of Colorado Denver Anschutz Medical Campus, Aurora, CO, USA. [62]Department of Civil and Environmental Engineering, Colorado School of Mines, Golden, CO, USA. [63]Department of Biological Sciences and Northern Gulf Institute, University of Southern Mississippi, Hattiesburg, Mississippi, USA. [64]Ocean Chemistry and Ecosystems Division, Atlantic Oceanographic and Meteorological Laboratory, National Oceanic and Atmospheric Administration, La Jolla, CA, USA. [65]Department of Biology, San Diego State University, San Diego, CA, USA. [66]Department of Environmental and Occupational Health Sciences, University of Washington, Seattle, WA, USA. [67]Division of Biological Sciences, University of California San Diego, San Diego, CA, USA. [68]Department of Microbiology and Immunology, University of California San Francisco, San Francisco, CA, USA. [69]Quantitative and Systems Biology Graduate Program, University of California Merced, Merced, CA, USA. [70]Bioinformatics Group, Wageningen University, Wageningen, The Netherlands. [71]Department of Mathematics, University of Arizona, Tucson, AZ, USA. [72]National Laboratory Service, Environment Agency, Starcross, UK. [73]College of Agriculture and Life Sciences, University of Florida, Gainesville, FL, USA. [74]Department of Biostatistics, University of Washington, Seattle, WA, USA. [75]University of Washington Bothell, School of STEM, Division of Biological Sciences, Bothell, WA, USA. [76]Merck & Co. Inc., Kenilworth, NJ, USA. [77]Department of Computer Science and Engineering, University of California San Diego, La Jolla, California, USA.

# Online Methods

## Overview of QIIME 2

We provide a high-level overview of the QIIME 2 system. `Monospace font` is used to indicate literal terms, such as objects defined by QIIME 2. The most up-to-date information on these topics is available in the QIIME 2 developer documentation at https://dev.qiime2.org.

There are three core components of the QIIME 2 system architecture: the **framework**, the **interfaces**, and the **plugins** (Figure S1). **Interfaces** are responsible for turning user intent into action. **Plugins** define all domain-specific functionality. The most important restriction of the architecture, which is evident in Figure S1, is that interfaces and plugins do not communicate directly with one another -- that communication is always mediated by the **framework**. In other words, the domain-specific analytic functionality (defined in plugins) is entirely decoupled from how users interface with the system (defined in interfaces). This important constraint allows multiple kinds of interfaces to be dynamically generated, and as a result QIIME 2 can adapt its user interface to the audience and the task at hand (Figure S2).

Third-party developers can create and distribute both plugins and interfaces for QIIME 2 independently of the core QIIME 2 development group, which forms the basis for our goal of decentralized QIIME 2 development (see https://library.qiime2.org and https://dev.qiime2.org). By removing our team as a bottleneck in developers delivering their new methods to users through QIIME 2, microbiome research can advance more quickly by ensuring that QIIME 2 users can have access to the latest microbiome analytic methods as quickly as bioinformatics researchers and developers can distribute them. This model makes QIIME 2 (and tools that build on it, such as Qiita) a platform for microbiome data science, not only a tool for a specific type of analysis. Since plugins conform to requirements specified by the framework, framework features such as data provenance tracking and multiple interface support are available for all plugins without the plugin developers having to be aware of these features.

In the terminology of QIIME 2, an **`Action`** creates a **`Result`**, and a `Result` can be either an **`Artifact`** or a **`Visualization`**. An `Artifact` is data generated by one or more QIIME 2 `Actions` which can be used as the input to other QIIME 2 `Actions`. A `Visualization` on the other hand is a terminal output of QIIME 2, which could be an interactive visualization (as in the Figure 1 examples) or any other result that is intended to be consumed by humans (not by a QIIME 2 `Action`). QIIME 2 assigns version 4 universally unique identifiers (UUIDs) to each execution of an `Action`, and to all `Results`.
QIIME 2 stores information about the series of Actions that led to a Result, along with information about the environment (including versions of all QIIME 2 packages and other Python dependencies) where each Action was executed, and the data itself. We refer to this process as data provenance tracking, or simply provenance tracking. We did not want to create new bioinformatics file formats to support the storage of data provenance, so QIIME 2 Results are instead stored as zip files containing a data directory that contains only the data in a relevant format (e.g., fasta or fastq for sequence data, newick for phylogenetic trees, etc), plus QIIME-2-specific metadata in other directories (such as provenance). These files use the extension .qza (for QIIME zipped artifact) or .qzv (for QIIME zipped visualization), but they are standard zip files that could be unzipped using common tools such as unzip, WinZip, or 7-Zip. Additional motivations for the storage of QIIME 2 Results in these structured zip files include the ability to submit as supplementary material to journals (the extension can simply be changed to .zip if required by the journal); "future-proofing" of QIIME 2 Results (even if QIIME 2 weren't used anymore, Results could still be accessed by unzipping .qza or .qzv files - see Extracting data from QIIME 2 archives below); zip files contain an index, allowing them to be inspected for certain information without uncompressing them; and data are always compressed, facilitating data sharing. Because provenance is stored alongside data in .qza and .qzv files, provenance tracking is decentralized (no QIIME 2

server or database needs to be keeping track of this information) ensuring that information on how data was generated will not be lost as long as the data is intact. However, assignment of UUIDs to all QIIME 2 `Results` (as described above) lends itself to managing these data in a database if that is desired.

Another important component of QIIME 2 is its **semantic type system**. All `Artifacts` used in QIIME 2 are annotated with a semantic description of their type which conveys the meaning of the data. Semantic types differ from data types (how data is represented in memory) or file formats (how data is stored on disk), and allow QIIME 2 to constrain the composition of multiple actions to only those combinations which are semantically meaningful without needing to consider the specific file formats or data types. This also makes it possible to determine what Actions could be applied (and in what order) to generate a given Artifact from some set of input Artifacts. For example, phylogenetic trees in QIIME 2 can be either rooted or unrooted, and these two concepts are represented by the semantic types `Phylogeny[Rooted]` and `Phylogeny[Unrooted]`, respectively. QIIME 2 could support loading these into multiple different data types, including a scikit-bio `TreeNode` object or an ete3 `Tree` object. Both of these types are typically stored on disk in a newick-formatted file, but this format doesn't contain easily accessible information on whether the phylogeny is rooted or unrooted. Some QIIME 2 `Actions` can only generate a `Phylogeny[Unrooted]` (such as `fasttree`), and some other `Actions` only work on `Phylogeny[Rooted]`(such as `beta-phylogenetic`, which computes UniFrac distances). The semantic type system allows QIIME 2 to determine that the output of `fasttree` should not be directly provided as input to `beta-phylogenetic`, and to provide the user with that information prior to execution. This can help a researcher who is new to microbiome data science avoid using data incorrectly. This will also enable QIIME 2 to automatically assist users in identifying relevant workflows to generate desired data or further explore data they already have.

Due to recent advances in package management systems and bioinformatics package repositories (e.g., Anaconda, Bioconda[1], and Bioconductor[2]), QIIME 2 is straightforward to install.

# QIIME 2 View

QIIME 2 View (https://view.qiime2.org) is a unique and novel contribution to the microbiome data science ecosystem that facilitates collaborative research. A user who has generated QIIME 2 visualizations can share those visualizations with a collaborator who can explore the results interactively without having QIIME 2 installed. QIIME 2 View achieves this simplified sharing of complex interactive visualizations through a novel combination of modern web browser APIs within a single-page application. It allows a user's browser to open and read `.qza` and `.qzv` files without the need to transfer the files over the network by utilizing a Service Worker to redirect HTTP requests directly into the archive which is retained on the user's computer. This approach of data unpackaging and local command execution makes QIIME 2 View well suited to cases where the results are unpublished or contain private information (that information will not be stored on any remote server). It is also possible to create "smart" URLs which automatically fetch content from a CORS-enabled web-server (for example, see the links in the README.md file at https://github.com/qiime2/paper1). This makes it very simple to share a single link with a collaborator that will be resolved into a fully interactive visualization on a user's computer automatically. The structured nature of the archive format (Figure S3) also allows QIIME 2 View to generate a dynamic provenance visualization, summarizing the entire provenance of the archive in question.

# Extracting data from QIIME 2 archives

QIIME 2 `.qza` and `.qzv` files are zip file containers with a defined internal directory structure. It's very easy to get data out in the canonical formats (Figure S3). If QIIME 2 and the q2cli command line interface are installed, this can be achieved using the `qiime tools export` command. If QIIME 2 is not installed, this can be achieved using standard decompression utilities such as `unzip`, WinZip, or 7-zip. We illustrate how this can be achieved using `unzip` on macOS. This can similarly be achieved on Windows or Linux. We illustrate this here to further future-proof QIIME 2 `Results` - even if the QIIME 2 documentation were no longer accessible, users could follow these steps to access QIIME 2 `Results`.

First, obtain a `.qza` file. Here we use the `FeatureData[Sequence]` artifact generated during the QIIME 2 Moving Pictures tutorial.

```
$ wget https://docs.qiime2.org/2018.8/data/tutorials/moving-pictures/rep-seqs.qza
```

Next, unzip that file with the macOS (or Linux) `unzip` program. This will create a new directory. The name of that directory will be the UUID of the artifact being unzipped, in this case `8dc793b8-7284-462a-8578-6370ffccebdc`.

```
$ unzip rep-seqs.qza
Archive:  rep-seqs.qza
  inflating: 8dc793b8-7284-462a-8578-6370ffccebdc/metadata.yaml
  inflating: 8dc793b8-7284-462a-8578-6370ffccebdc/VERSION
  inflating: 8dc793b8-7284-462a-8578-6370ffccebdc/provenance/metadata.yaml
  inflating: 8dc793b8-7284-462a-8578-6370ffccebdc/provenance/citations.bib
  inflating: 8dc793b8-7284-462a-8578-6370ffccebdc/provenance/VERSION
  inflating:
8dc793b8-7284-462a-8578-6370ffccebdc/provenance/artifacts/bdaa3214-f883-4c8b-8db3-f6ea4910d724/metadata.yaml
  inflating:
8dc793b8-7284-462a-8578-6370ffccebdc/provenance/artifacts/bdaa3214-f883-4c8b-8db3-f6ea4910d724/citations.bib
  inflating:
8dc793b8-7284-462a-8578-6370ffccebdc/provenance/artifacts/bdaa3214-f883-4c8b-8db3-f6ea4910d724/VERSION
  inflating:
8dc793b8-7284-462a-8578-6370ffccebdc/provenance/artifacts/bdaa3214-f883-4c8b-8db3-f6ea4910d724/action/action.yam
l
  inflating:
8dc793b8-7284-462a-8578-6370ffccebdc/provenance/artifacts/7097fc98-ad5f-4b9d-a33e-39cd36857a0d/metadata.yaml
  inflating:
8dc793b8-7284-462a-8578-6370ffccebdc/provenance/artifacts/7097fc98-ad5f-4b9d-a33e-39cd36857a0d/citations.bib
  inflating:
8dc793b8-7284-462a-8578-6370ffccebdc/provenance/artifacts/7097fc98-ad5f-4b9d-a33e-39cd36857a0d/VERSION
  inflating:
8dc793b8-7284-462a-8578-6370ffccebdc/provenance/artifacts/7097fc98-ad5f-4b9d-a33e-39cd36857a0d/action/action.yam
l
  inflating:
8dc793b8-7284-462a-8578-6370ffccebdc/provenance/artifacts/7097fc98-ad5f-4b9d-a33e-39cd36857a0d/action/barcodes.t
sv
  inflating: 8dc793b8-7284-462a-8578-6370ffccebdc/provenance/action/action.yaml
  inflating: 8dc793b8-7284-462a-8578-6370ffccebdc/data/dna-sequences.fasta
```

The last entry that is unzipped in this example is `data/dna-sequences.fasta`. All other directories and files are QIIME 2 specific metadata (such as information about the semantic type of the artifact and the data provenance). If you're only interested in the sequence data, you can safely ignore all of that information. The `data/dna-sequences.fasta` file is a typical fasta file containing sequence identifiers and sequences. The first four lines of this file can be viewed as follows:

```
$ head -4 8dc793b8-7284-462a-8578-6370ffccebdc/data/dna-sequences.fasta
>f352c1f1efecf483511c2270aabd0ae6
TACGTAGGGTGCGAGCGTTAATCGGAATTACTGGGCGTAAAGCGTGCGCAGGCGGTTTTGTAAGACAGAGGTGAAATCCCCGGGCTCAACCTGGGAACTG
CCTTTGTGACTGCAAGGCTG
>82e72255267397b777a1afd44ea22755
```

```
TACGGAGGATCCAAGCGTTATCCGGAATCATTGGGTTTAAAGGGTCCGTAGGCGGTTTAGTAAGTCAGTGGTGAAAGCCCATCGCTCAACGGTGGAACGG
CCATTGATACTGCTAGACTT
```

# QIIME 2 user and developer community

QIIME 2 officially succeeded QIIME 1 (http://www.qiime.org)[3] in January of 2018, and has developed an engaged user base and community. As of this writing there are over 1980 active users (users who have performed an action, such as creating or liking a post) on the QIIME 2 Forum; over 3000 monthly downloads of QIIME 2 from Anaconda; over 8000 unique visitors to the QIIME 2 Forum according to Google Analytics; and our multi-day workshops are frequently filled to capacity (https://workshops.qiime2.org). QIIME 2 is also being adopted by third-party bioinformatics developers who are choosing to make their software accessible through plugins, and who are motivated to develop for QIIME 2 by access to its integrated provenance tracking, multiple interfaces, standardization of data types provided by the semantic type system, large user community, and supportive developer community.

A core goal of QIIME 2 is to cultivate a diverse and inclusive community of scientists, software engineers, statisticians, educators, students, and other microbiome stakeholders who are openly sharing methods, data, and knowledge to advance microbiome research.

# Supplementary figures, files, and captions

**Figure S1. Schematic diagram of the QIIME 2 system.** Interfaces define how users interact with the system; plugins define all domain-specific functionality; and the framework mediates communication between plugins and interfaces, and performs core functionality such as provenance tracking. Arrows indicate dependencies. Interfaces interact only with the `qiime2.sdk` submodule, while plugins interact only with the `qiime2.plugin` submodule. This design has led to a system that is readily extended by third-party plugin and interface developers.

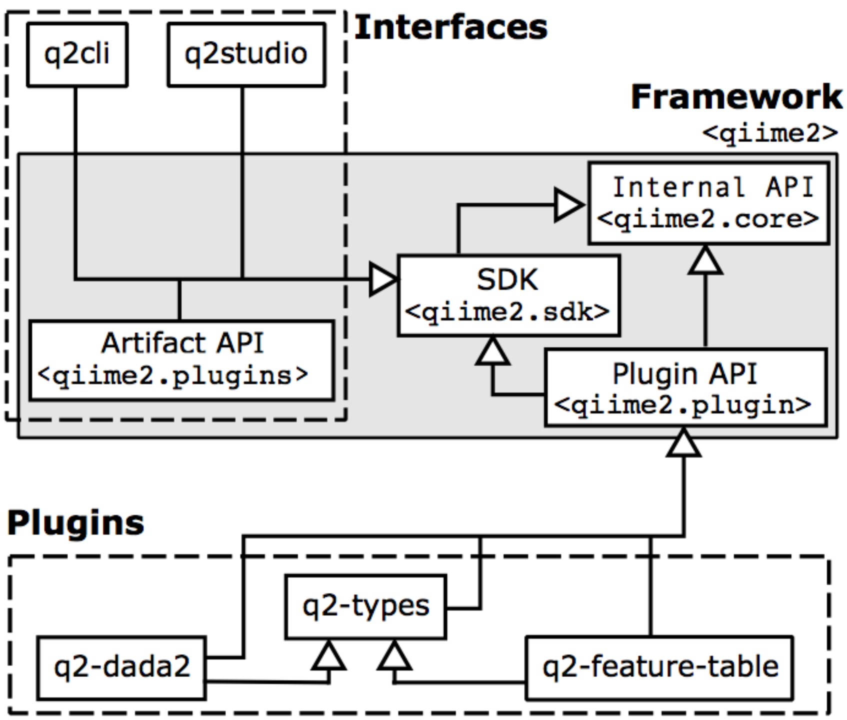


**Figure S2. QIIME 2 is interface agnostic.** The full suite of QIIME 2 functionality is useful to and usable by researchers ranging widely in their computational sophistication, a major advantage over technologies such as QIIME 1 that provide a single interface. (a) Users wanting to view QIIME 2 results or data provenance can use QIIME 2 View without installing QIIME 2, which is convenient for lead investigators, clinicians, or policy makers who may want to explore interactive visualizations generated by others. (b) Researchers who prefer graphical interfaces can use QIIME 2 Studio, our prototype graphical interface. This is convenient for users without command line or programming skills. (c) Power users (e.g., who are comfortable with the Linux command line and/or regularly work on institutional computer clusters), can use QIIME 2 through the command line interface, q2cli. (d) "Data scientists" (e.g., users who are programmers, who work in Jupyter Notebooks, or who are interested in automating QIIME 2 workflows), can use QIIME 2 through the Python 3 "artifact API".

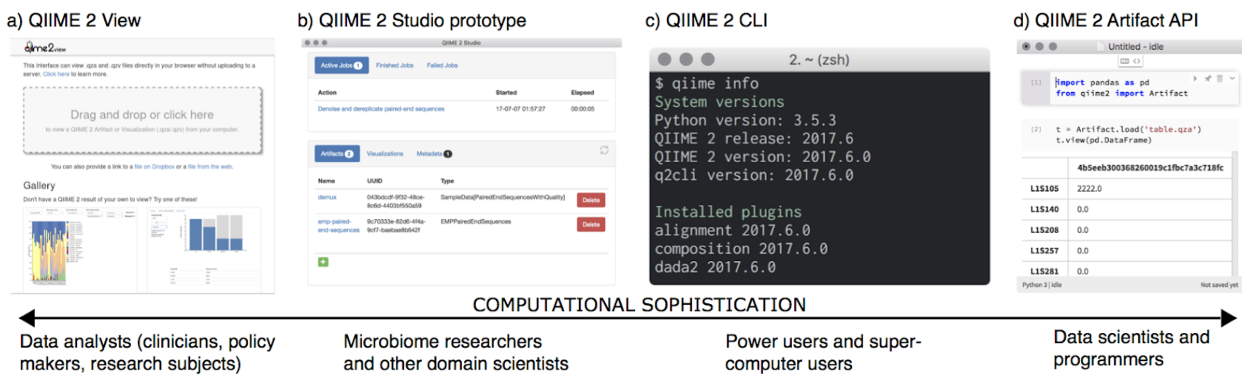
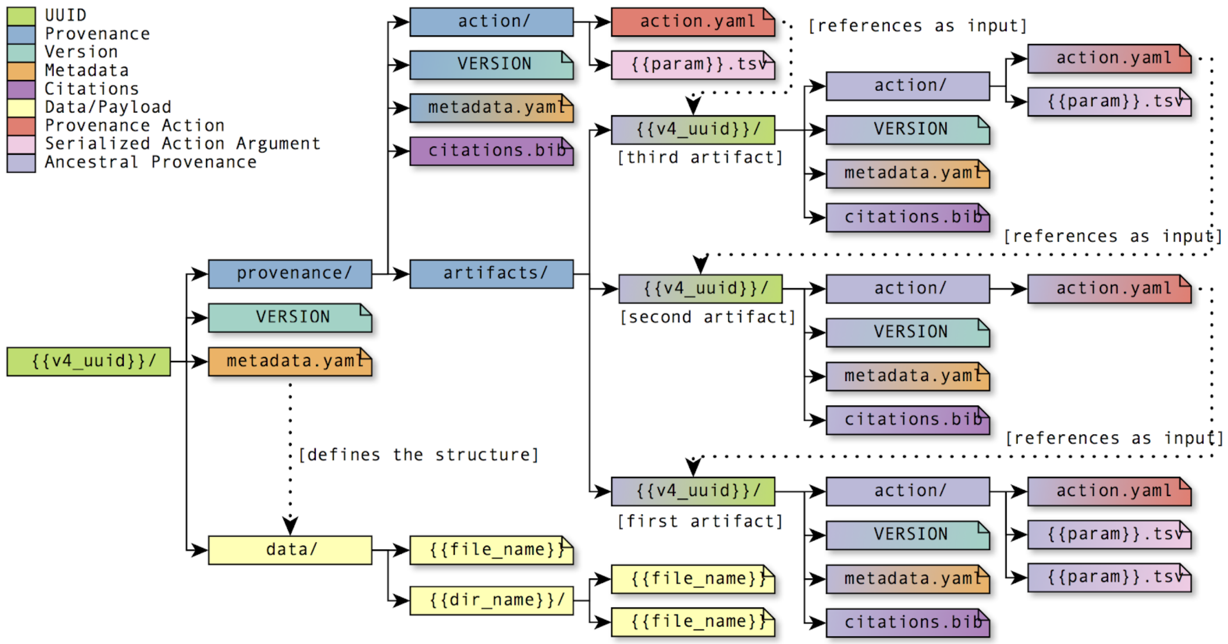


**Figure S3. Anatomy of a QIIME 2 `Archive` (i.e., .qza or .qzv file).** QIIME 2 stores data in a directory structure called an `Archive`. These archives are zipped to make moving data convenient. The directory structure has a single root directory named with a UUID which serves as the identity of the archive.



**Supplementary File 1** contains the QIIME 2 `.qzv` files corresponding to **Figure 1a-d**.
**Supplementary File 2** contains four worked examples of using QIIME 2.
**Supplementary File 3** contains notes from end users on QIIME 2.

# Online methods references


1. Grüning, B. et al. Nat. Methods **15**, 475–476 (2018).
2. Huber, W. et al. Nat. Methods **12**, 115–121 (2015).
3. Caporaso, J.G. et al. Nat. Methods **7**, 335–336 (2010).
4. Thompson, L.R. et al. Nature **551**, 457–463 (2017).
5. McDonald, D. et al. mSystems **3**, e00031–18 (2018).
6. McDonald, D. et al. Gigascience **1**, 7 (2012).
7. Amir, A. et al. mSystems **2**, (2017).
8. Lozupone, C. & Knight, R. Appl. Environ. Microbiol. **71**, 8228–8235 (2005).
9. Vázquez-Baeza, Y., Pirrung, M., Gonzalez, A. & Knight, R. Gigascience **2**, 16 (2013).
10. Bokulich, N.A. et al. Sci. Transl. Med. **8**, 343ra82 (2016).
11. Callahan, B.J. et al. Nat. Methods (2016).doi:10.1038/nmeth.3869
12. McDonald, D. et al. ISME J. **6**, 610–618 (2012).
13. Bokulich, N.A. et al. Microbiome **6**, 90 (2018).
14. Bokulich, N. et al. bioRxiv 223974 (2017).doi:10.1101/223974
15. Protsyuk, I. et al. Nat. Protoc. **13**, 134–154 (2018).
16. Bouslimani, A. et al. Proc. Natl. Acad. Sci. U. S. A. **112**, E2120–9 (2015).

# Supplementary File 2: QIIME 2 worked examples

Figure 1 presents the output of four worked examples of QIIME 2, and Supplementary File 1 contains the QIIME 2 `.qzv` files corresponding to Figure 1a-d. These are also accessible at https://github.com/qiime2/paper1 and can be viewed using QIIME 2 View (https://view.qiime2.org) where readers can interact with the results, and explore the methods used to generate them (see the Provenance tab after loading a `.qzv` file with QIIME 2 View).

Here we describe the methods used to generate each of these visualizations at a high level, and present the QIIME 2 steps to generate these visualizations both as a series of command line interface (CLI) commands and as application programmer interface (API) calls. The steps presented are derived from the data provenance of each of the visualizations, and can be compared directly to the Provenance tab on QIIME 2 View. These worked examples represent real-world working conditions where different analysis steps were conducted at different times by different individuals running multiple releases of QIIME 2. This is common, for example as in Figure 2, when a researcher uses a pre-trained taxonomic classifier such as those available on the QIIME 2 website and forum. Because detailed information on system and plugin versions is tracked in provenance, consumers of QIIME 2 results and data provenance have comprehensive information about the software and dependency environment where results were generated ensuring reproducibility. In some examples data processed with other pipelines are imported into QIIME 2 (Figures 1a and 1d), while in others analysis begins with DNA sequencing data (Figures 1b and 1c). This illustrates that QIIME 2 can be used as an “end-to-end” microbiome analysis pipeline, or as a component of workflows that use other bioinformatics tools.

## Figure 1a (a-pcoa.qzv in Supplementary File 1)

Emperor PCoA plot presenting a meta-analysis of the first release of the Earth Microbiome Project (EMP)[4] and the first release of the American Gut Project (AGP)[5]. The EMP data was obtained from ftp://ftp.microbio.me/emp/release1, and the AGP data was obtained from Qiita study 10317 for the set of samples used in its publication (samples described in the AGP supplemental data accession table). Both projects were downloaded and imported into QIIME 2 as BIOM tables[6]. Those BIOM tables were combined, filtered for blooms[7], rarefied at an even depth (1000 sequences per sample), and compared using the unweighted UniFrac[8] metric. Lastly the samples were projected into a small dimensional space using principal coordinates analysis and visualized using Emperor[9]. The samples were colored according to the Earth Microbiome Project Ontology[4].

### Command line interface steps

```
# versions: {'types': '2018.8.0'}
qiime tools import \
  --type 'FeatureData[Sequence]' \
  --input-path combined-features.fasta \
  --input-format DNAFASTAFormat \
  --output-path combined-features.fasta.qza

# versions: {'types': '2018.8.0', 'feature-table': '2018.8.0'}
qiime feature-table filter-seqs \
  --i-data combined-features.fasta.qza \
```

```
  --m-metadata-file metadata1.txt \
  --p-exclude-ids \
  --output-dir feature-table-filter_seqs_1

# versions: {'types': '2018.8.0', 'fragment-insertion': '2018.6.17'}
qiime fragment-insertion sepp \
  --i-representative-sequences feature-table-filter_seqs_1/filtered_data.qza \
  --p-threads 20 \
  --p-alignment-subset-size 1000 \
  --p-placement-subset-size 5000 \
  --p-no-debug \
  --output-dir fragment-insertion-sepp_1

# versions: {'types': '2018.8.0'}
qiime tools import \
  --type 'FeatureTable[Frequency]' \
  --input-path emp.biom \
  --input-format BIOMV210Format \
  --output-path emp.biom.qza

# versions: {'types': '2018.8.0'}
qiime tools import \
  --type 'FeatureTable[Frequency]' \
  --input-path agp.upper.biom \
  --input-format BIOMV210Format \
  --output-path agp.upper.biom.qza

# versions: {'types': '2018.8.0', 'feature-table': '2018.8.0'}
qiime feature-table merge \
  --i-tables agp.upper.biom.qza \
  --i-tables emp.biom.qza \
  --p-overlap-method 'error_on_overlapping_sample' \
  --output-dir feature-table-merge_1

# versions: {'types': '2018.8.0', 'feature-table': '2018.8.0'}
qiime feature-table filter-features \
  --i-table feature-table-merge_1/merged_table.qza \
  --m-metadata-file metadata2.txt \
  --p-min-frequency 0 \
  --p-min-samples 0 \
  --p-exclude-ids \
  --output-dir feature-table-filter_features_1

# versions: {'types': '2018.8.0', 'feature-table': '2018.8.0'}
qiime feature-table rarefy \
  --i-table feature-table-filter_features_1/filtered_table.qza \
  --p-sampling-depth 1000 \
  --output-dir feature-table-rarefy_1
```

```
# versions: {'types': '2018.8.0', 'diversity': '2018.8.0'}
qiime diversity beta-phylogenetic-alt \
  --i-table feature-table-rarefy_1/rarefied_table.qza \
  --i-phylogeny fragment-insertion-sepp_1/tree.qza \
  --p-metric 'unweighted_unifrac' \
  --p-n-jobs 16 \
  --p-no-variance-adjusted \
  --p-bypass-tips \
  --output-dir diversity-beta_phylogenetic_alt_1

# versions: {'types': '2018.8.0', 'diversity': '2018.8.0'}
qiime diversity filter-distance-matrix \
  --i-distance-matrix diversity-beta_phylogenetic_alt_1/distance_matrix.qza \
  --m-metadata-file metadata3.txt \
  --p-where 'project!="Not Available"' \
  --p-no-exclude-ids \
  --output-dir diversity-filter_distance_matrix_1

# versions: {'types': '2018.8.0', 'diversity': '2018.8.0'}
qiime diversity pcoa \
  --i-distance-matrix
diversity-filter_distance_matrix_1/filtered_distance_matrix.qza \
  --p-number-of-dimensions 5 \
  --output-dir diversity-pcoa_1

# versions: {'types': '2018.8.0', 'emperor': '2018.8.0'}
qiime emperor plot \
  --i-pcoa diversity-pcoa_1/pcoa.qza \
  --m-metadata-file metadata4.txt \
  --output-dir emperor-plot_1
```

## Application programmer interface steps

```
import qiime2
from qiime2.plugins import fragment_insertion
from qiime2.plugins import feature_table
from qiime2.plugins import diversity
from qiime2.plugins import emperor

metadata1_txt = qiime2.Metadata.load('metadata1.txt')
metadata2_txt = qiime2.Metadata.load('metadata2.txt')
metadata3_txt = qiime2.Metadata.load('metadata3.txt')
metadata4_txt = qiime2.Metadata.load('metadata4.txt')

# versions: {'types': '2018.8.0'}
combined-features_fasta = qiime2.Artifact.import_data(
  'FeatureData[Sequence]', 'combined-features.fasta', view_type='DNAFASTAFormat'
```

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27295v2 | CC BY 4.0 Open Access | rec: 3 Dec 2018, publ: 3 Dec 2018

```
)

# versions: {'types': '2018.8.0', 'feature-table': '2018.8.0'}
feature_table_filter_seqs_1 = feature_table.actions.filter_seqs(
  data=combined-features_fasta,
  metadata=metadata1_txt,
  where=None,
  exclude_ids=True,
)

# versions: {'types': '2018.8.0', 'fragment-insertion': '2018.6.17'}
fragment_insertion_sepp_1 = fragment_insertion.actions.sepp(
  representative_sequences=feature_table_filter_seqs_1.filtered_data,
  threads=20,
  alignment_subset_size=1000,
  placement_subset_size=5000,
  debug=False,
)

# versions: {'types': '2018.8.0'}
emp_biom = qiime2.Artifact.import_data(
  'FeatureTable[Frequency]', 'emp.biom', view_type='BIOMV210Format'
)

# versions: {'types': '2018.8.0'}
agp_upper_biom = qiime2.Artifact.import_data(
  'FeatureTable[Frequency]', 'agp.upper.biom', view_type='BIOMV210Format'
)

# versions: {'types': '2018.8.0', 'feature-table': '2018.8.0'}
feature_table_merge_1 = feature_table.actions.merge(
  tables=agp_upper_biom,
  tables=emp_biom,
  overlap_method='error_on_overlapping_sample',
)

# versions: {'types': '2018.8.0', 'feature-table': '2018.8.0'}
feature_table_filter_features_1 = feature_table.actions.filter_features(
  table=feature_table_merge_1.merged_table,
  metadata=metadata2_txt,
  min_frequency=0,
  max_frequency=None,
  min_samples=0,
  max_samples=None,
  where=None,
  exclude_ids=True,
)
```

```
# versions: {'types': '2018.8.0', 'feature-table': '2018.8.0'}
feature_table_rarefy_1 = feature_table.actions.rarefy(
  table=feature_table_filter_features_1.filtered_table,
  sampling_depth=1000,
)

# versions: {'types': '2018.8.0', 'diversity': '2018.8.0'}
diversity_beta_phylogenetic_alt_1 = diversity.actions.beta_phylogenetic_alt(
  table=feature_table_rarefy_1.rarefied_table,
  phylogeny=fragment_insertion_sepp_1.tree,
  metric='unweighted_unifrac',
  n_jobs=16,
  variance_adjusted=False,
  alpha=None,
  bypass_tips=True,
)

# versions: {'types': '2018.8.0', 'diversity': '2018.8.0'}
diversity_filter_distance_matrix_1 = diversity.actions.filter_distance_matrix(
  distance_matrix=diversity_beta_phylogenetic_alt_1.distance_matrix,
  metadata=metadata3_txt,
  where='project!="Not Available"',
  exclude_ids=False,
)

# versions: {'types': '2018.8.0', 'diversity': '2018.8.0'}
diversity_pcoa_1 = diversity.actions.pcoa(
  distance_matrix=diversity_filter_distance_matrix_1.filtered_distance_matrix,
  number_of_dimensions=5,
)

# versions: {'types': '2018.8.0', 'emperor': '2018.8.0'}
emperor_plot_1 = emperor.actions.plot(
  pcoa=diversity_pcoa_1.pcoa,
  metadata=metadata4_txt,
  custom_axes=None,
)
```

## Figure 1b (b-feature-volatility.qzv in Supplementary File 1)

Data were generated on five sequencing runs of V4 16S rRNA gene amplicons from the ECAM study[10]. Forward reads were imported separately in `EMPSingleEndDirFmt` format, demultiplexed with q2-demux's `emp_single` method, and denoised using q2-dada2's `denoise_single` method (`trunc_len`=150, other parameters used default values)[11]. Denoised feature tables and sequences were merged using q2-feature-table's `merge` and `merge-seqs` methods, respectively. q2-feature-table's `filter-samples` method was used to remove samples with fewer than 2000 sequences, and to perform metadata-based filtering to retain only children's samples. A naive Bayes taxonomy classifier was trained on the Greengenes[12] reference sequences (clustered at 99% similarity) using q2-feature-classifier's `fit-classifier-naive-bayes` method[13]. This classifier was used to taxonomically classify the ECAM ASVs using q2-feature-classifier's `classify-sklearn` method[13]. ASVs were collapsed based on genus-level taxonomy using q2-taxa's collapse method. Temporally predictive features were identified using q2-longitudinal's `feature-volatility` pipeline[14] using default parameters. Data contained in this artifact have been described in a previous publication[14].

## Command line interface steps

```
# versions: {'framework': '2017.2.0'}
qiime tools import \
  --type 'FeatureData[Taxonomy]' \
  --input-path 99_otu_taxonomy.txt \
  --input-format TaxonomyFormat \
  --output-path 99_otu_taxonomy.txt.qza

# versions: {'framework': '2017.2.0'}
qiime tools import \
  --type 'FeatureData[Sequence]' \
  --input-path 99_otus.fasta \
  --input-format DNAFASTAFormat \
  --output-path 99_otus.fasta.qza

# versions: {'feature-classifier': '2017.2.0'}
qiime feature-classifier fit-classifier-naive-bayes \
  --i-reference-reads 99_otus.fasta.qza \
  --i-reference-taxonomy 99_otu_taxonomy.txt.qza \
  --p-classify--alpha 0.01 \
  --p-classify--chunk-size -1 \
  --p-classify--class-prior 'null' \
  --p-classify--fit-prior \
  --p-feat-ext--analyzer 'char_wb' \
  --p-no-feat-ext--binary \
  --p-feat-ext--decode-error 'strict' \
  --p-feat-ext--encoding 'utf-8' \
  --p-feat-ext--input 'content' \
  --p-feat-ext--lowercase \
  --p-feat-ext--n-features 8192 \
  --p-feat-ext--ngram-range '[8, 8]' \
  --p-feat-ext--non-negative \
  --p-feat-ext--norm 'l2' \
  --p-feat-ext--preprocessor 'null' \
  --p-feat-ext--stop-words 'null' \
  --p-feat-ext--strip-accents 'null' \
  --p-feat-ext--token-pattern '(?u)\b\w\w+\b' \
  --p-feat-ext--tokenizer 'null' \
  --output-dir feature-classifier-fit_classifier_naive_bayes_1
```

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27295v2 | CC BY 4.0 Open Access | rec: 3 Dec 2018, publ: 3 Dec 2018

```
# versions: {'framework': '2017.4.0'}
qiime tools import \
  --type 'RawSequences' \
  --input-path EMPSingleEndDirFmtimport_dir \
  --input-format EMPSingleEndDirFmt \
  --output-path EMPSingleEndDirFmtimport_dir.qza

# versions: {'demux': '2017.4.0'}
qiime demux emp-single \
  --i-seqs EMPSingleEndDirFmtimport_dir.qza \
  --m-barcodes-file metadata1.txt \
  --m-barcodes-column 'ColumnName' \
  --p-no-rev-comp-barcodes \
  --p-rev-comp-mapping-barcodes \
  --output-dir demux-emp_single_1

# versions: {'dada2': '2017.4.0'}
qiime dada2 denoise-single \
  --i-demultiplexed-seqs demux-emp_single_1/per_sample_sequences.qza \
  --p-trunc-len 150 \
  --p-trim-left 0 \
  --p-max-ee 2.0 \
  --p-trunc-q 2 \
  --p-chimera-method 'pooled' \
  --p-min-fold-parent-over-abundance 1.0 \
  --p-n-threads 1 \
  --p-n-reads-learn 1000000 \
  --p-hashed-feature-ids \
  --output-dir dada2-denoise_single_1

# versions: {'framework': '2017.4.0'}
qiime tools import \
  --type 'RawSequences' \
  --input-path EMPSingleEndDirFmtimport_dir \
  --input-format EMPSingleEndDirFmt \
  --output-path EMPSingleEndDirFmtimport_dir.qza

# versions: {'demux': '2017.4.0'}
qiime demux emp-single \
  --i-seqs EMPSingleEndDirFmtimport_dir.qza \
  --m-barcodes-file metadata2.txt \
  --m-barcodes-column 'ColumnName' \
  --p-no-rev-comp-barcodes \
  --p-rev-comp-mapping-barcodes \
  --output-dir demux-emp_single_2

# versions: {'dada2': '2017.4.0'}
```

```
qiime dada2 denoise-single \
  --i-demultiplexed-seqs demux-emp_single_2/per_sample_sequences.qza \
  --p-trunc-len 150 \
  --p-trim-left 0 \
  --p-max-ee 2.0 \
  --p-trunc-q 2 \
  --p-chimera-method 'pooled' \
  --p-min-fold-parent-over-abundance 1.0 \
  --p-n-threads 1 \
  --p-n-reads-learn 1000000 \
  --p-hashed-feature-ids \
  --output-dir dada2-denoise_single_2

# versions: {'framework': '2017.4.0'}
qiime tools import \
  --type 'RawSequences' \
  --input-path EMPSingleEndDirFmtimport_dir \
  --input-format EMPSingleEndDirFmt \
  --output-path EMPSingleEndDirFmtimport_dir.qza

# versions: {'demux': '2017.4.0'}
qiime demux emp-single \
  --i-seqs EMPSingleEndDirFmtimport_dir.qza \
  --m-barcodes-file metadata3.txt \
  --m-barcodes-column 'ColumnName' \
  --p-no-rev-comp-barcodes \
  --p-rev-comp-mapping-barcodes \
  --output-dir demux-emp_single_3

# versions: {'dada2': '2017.4.0'}
qiime dada2 denoise-single \
  --i-demultiplexed-seqs demux-emp_single_3/per_sample_sequences.qza \
  --p-trunc-len 150 \
  --p-trim-left 0 \
  --p-max-ee 2.0 \
  --p-trunc-q 2 \
  --p-chimera-method 'pooled' \
  --p-min-fold-parent-over-abundance 1.0 \
  --p-n-threads 1 \
  --p-n-reads-learn 1000000 \
  --p-hashed-feature-ids \
  --output-dir dada2-denoise_single_3

# versions: {'framework': '2017.4.0'}
qiime tools import \
  --type 'RawSequences' \
  --input-path EMPSingleEndDirFmtimport_dir \
  --input-format EMPSingleEndDirFmt \
```

```
  --output-path EMPSingleEndDirFmtimport_dir.qza

# versions: {'demux': '2017.4.0'}
qiime demux emp-single \
  --i-seqs EMPSingleEndDirFmtimport_dir.qza \
  --m-barcodes-file metadata4.txt \
  --m-barcodes-column 'ColumnName' \
  --p-no-rev-comp-barcodes \
  --p-rev-comp-mapping-barcodes \
  --output-dir demux-emp_single_4

# versions: {'dada2': '2017.4.0'}
qiime dada2 denoise-single \
  --i-demultiplexed-seqs demux-emp_single_4/per_sample_sequences.qza \
  --p-trunc-len 150 \
  --p-trim-left 0 \
  --p-max-ee 2.0 \
  --p-trunc-q 2 \
  --p-chimera-method 'pooled' \
  --p-min-fold-parent-over-abundance 1.0 \
  --p-n-threads 1 \
  --p-n-reads-learn 1000000 \
  --p-hashed-feature-ids \
  --output-dir dada2-denoise_single_4

# versions: {'framework': '2017.4.0'}
qiime tools import \
  --type 'RawSequences' \
  --input-path EMPSingleEndDirFmtimport_dir \
  --input-format EMPSingleEndDirFmt \
  --output-path EMPSingleEndDirFmtimport_dir.qza

# versions: {'demux': '2017.4.0'}
qiime demux emp-single \
  --i-seqs EMPSingleEndDirFmtimport_dir.qza \
  --m-barcodes-file metadata5.txt \
  --m-barcodes-column 'ColumnName' \
  --p-no-rev-comp-barcodes \
  --p-rev-comp-mapping-barcodes \
  --output-dir demux-emp_single_5

# versions: {'dada2': '2017.4.0'}
qiime dada2 denoise-single \
  --i-demultiplexed-seqs demux-emp_single_5/per_sample_sequences.qza \
  --p-trunc-len 150 \
  --p-trim-left 0 \
  --p-max-ee 2.0 \
  --p-trunc-q 2 \
```

```
  --p-chimera-method 'pooled' \
  --p-min-fold-parent-over-abundance 1.0 \
  --p-n-threads 1 \
  --p-n-reads-learn 1000000 \
  --p-hashed-feature-ids \
  --output-dir dada2-denoise_single_5

# versions: {'feature-table': '2017.4.0'}
qiime feature-table merge-seq-data \
  --i-data1 dada2-denoise_single_5/representative_sequences.qza \
  --i-data2 dada2-denoise_single_4/representative_sequences.qza \
  --output-dir feature-table-merge_seq_data_1

# versions: {'feature-table': '2017.4.0'}
qiime feature-table merge-seq-data \
  --i-data1 feature-table-merge_seq_data_1/merged_data.qza \
  --i-data2 dada2-denoise_single_3/representative_sequences.qza \
  --output-dir feature-table-merge_seq_data_2

# versions: {'feature-table': '2017.4.0'}
qiime feature-table merge-seq-data \
  --i-data1 feature-table-merge_seq_data_2/merged_data.qza \
  --i-data2 dada2-denoise_single_2/representative_sequences.qza \
  --output-dir feature-table-merge_seq_data_3

# versions: {'feature-table': '2017.4.0'}
qiime feature-table merge-seq-data \
  --i-data1 feature-table-merge_seq_data_3/merged_data.qza \
  --i-data2 dada2-denoise_single_1/representative_sequences.qza \
  --output-dir feature-table-merge_seq_data_4

# versions: {'feature-classifier': '2017.5.0'}
qiime feature-classifier classify-sklearn \
  --i-reads feature-table-merge_seq_data_4/merged_data.qza \
  --i-classifier feature-classifier-fit_classifier_naive_bayes_1/classifier.qza \
  --p-chunk-size 262144 \
  --p-n-jobs 4 \
  --p-pre-dispatch '2*n_jobs' \
  --p-confidence 0.7 \
  --output-dir feature-classifier-classify_sklearn_1

# versions: {'feature-table': '2017.4.0'}
qiime feature-table merge \
  --i-table1 dada2-denoise_single_5/table.qza \
  --i-table2 dada2-denoise_single_4/table.qza \
  --output-dir feature-table-merge_1

# versions: {'feature-table': '2017.4.0'}
```

```
qiime feature-table merge \
  --i-table1 feature-table-merge_1/merged_table.qza \
  --i-table2 dada2-denoise_single_3/table.qza \
  --output-dir feature-table-merge_2

# versions: {'feature-table': '2017.4.0'}
qiime feature-table merge \
  --i-table1 feature-table-merge_2/merged_table.qza \
  --i-table2 dada2-denoise_single_2/table.qza \
  --output-dir feature-table-merge_3

# versions: {'feature-table': '2017.4.0'}
qiime feature-table merge \
  --i-table1 feature-table-merge_3/merged_table.qza \
  --i-table2 dada2-denoise_single_1/table.qza \
  --output-dir feature-table-merge_4

# versions: {'feature-table': '2017.4.0'}
qiime feature-table filter-samples \
  --i-table feature-table-merge_4/merged_table.qza \
  --p-min-frequency 2000 \
  --p-min-features 0 \
  --output-dir feature-table-filter_samples_1

# versions: {'feature-table': '2017.4.0'}
qiime feature-table filter-samples \
  --i-table feature-table-filter_samples_1/filtered_table.qza \
  --m-sample-metadata-file metadata6.txt \
  --p-min-frequency 0 \
  --p-min-features 0 \
  --output-dir feature-table-filter_samples_2

# versions: {'taxa': '2017.9.0.dev0+2.g2ebb91d'}
qiime taxa collapse \
  --i-table feature-table-filter_samples_2/filtered_table.qza \
  --i-taxonomy feature-classifier-classify_sklearn_1/classifier.qza \
  --p-level 6 \
  --output-dir taxa-collapse_1

# versions: {'longitudinal': '2018.8.0'}
qiime longitudinal feature-volatility \
  --i-table taxa-collapse_1/collapsed_table.qza \
  --m-metadata-file metadata7.txt \
  --p-state-column 'month' \
  --p-individual-id-column 'studyid' \
  --p-cv 5 \
  --p-n-jobs 4 \
  --p-n-estimators 100 \
```

```
  --p-estimator 'RandomForestRegressor' \
  --p-no-parameter-tuning \
  --p-missing-samples 'error' \
  --output-dir longitudinal-feature_volatility_1
```

## Application programmer interface steps

```
import qiime2
from qiime2.plugins import feature_table
from qiime2.plugins import feature_classifier
from qiime2.plugins import demux
from qiime2.plugins import dada2
from qiime2.plugins import longitudinal
from qiime2.plugins import taxa

metadata1_txt_ColumnName =
qiime2.Metadata.load('metadata1.txt').get_column('ColumnName')
metadata2_txt_ColumnName =
qiime2.Metadata.load('metadata2.txt').get_column('ColumnName')
metadata3_txt_ColumnName =
qiime2.Metadata.load('metadata3.txt').get_column('ColumnName')
metadata4_txt_ColumnName =
qiime2.Metadata.load('metadata4.txt').get_column('ColumnName')
metadata5_txt_ColumnName =
qiime2.Metadata.load('metadata5.txt').get_column('ColumnName')
metadata6_txt = qiime2.Metadata.load('metadata6.txt')
metadata7_txt = qiime2.Metadata.load('metadata7.txt')

# versions: {'framework': '2017.2.0'}
99_otu_taxonomy_txt = qiime2.Artifact.import_data(
  'FeatureData[Taxonomy]', '99_otu_taxonomy.txt', view_type='TaxonomyFormat'
)

# versions: {'framework': '2017.2.0'}
99_otus_fasta = qiime2.Artifact.import_data(
  'FeatureData[Sequence]', '99_otus.fasta', view_type='DNAFASTAFormat'
)

# versions: {'feature-classifier': '2017.2.0'}
feature_classifier_fit_classifier_naive_bayes_1 =
feature_classifier.actions.fit_classifier_naive_bayes(
  reference_reads=99_otus_fasta,
  reference_taxonomy=99_otu_taxonomy_txt,
  classify__alpha=0.01,
  classify__chunk_size=-1,
  classify__class_prior='null',
  classify__fit_prior=True,
  feat_ext__analyzer='char_wb',
```

```
  feat_ext__binary=False,
  feat_ext__decode_error='strict',
  feat_ext__encoding='utf-8',
  feat_ext__input='content',
  feat_ext__lowercase=True,
  feat_ext__n_features=8192,
  feat_ext__ngram_range='[8, 8]',
  feat_ext__non_negative=True,
  feat_ext__norm='l2',
  feat_ext__preprocessor='null',
  feat_ext__stop_words='null',
  feat_ext__strip_accents='null',
  feat_ext__token_pattern='(?u)\b\w\w+\b',
  feat_ext__tokenizer='null',
)

# versions: {'framework': '2017.4.0'}
EMPSingleEndDirFmtimport_dir = qiime2.Artifact.import_data(
  'RawSequences', 'EMPSingleEndDirFmtimport_dir', view_type='EMPSingleEndDirFmt'
)

# versions: {'demux': '2017.4.0'}
demux_emp_single_1 = demux.actions.emp_single(
  seqs=EMPSingleEndDirFmtimport_dir,
  barcodes=metadata1_txt_ColumnName,
  rev_comp_barcodes=False,
  rev_comp_mapping_barcodes=True,
)

# versions: {'dada2': '2017.4.0'}
dada2_denoise_single_1 = dada2.actions.denoise_single(
  demultiplexed_seqs=demux_emp_single_1.per_sample_sequences,
  trunc_len=150,
  trim_left=0,
  max_ee=2.0,
  trunc_q=2,
  chimera_method='pooled',
  min_fold_parent_over_abundance=1.0,
  n_threads=1,
  n_reads_learn=1000000,
  hashed_feature_ids=True,
)

# versions: {'framework': '2017.4.0'}
EMPSingleEndDirFmtimport_dir = qiime2.Artifact.import_data(
  'RawSequences', 'EMPSingleEndDirFmtimport_dir', view_type='EMPSingleEndDirFmt'
)
```

```
# versions: {'demux': '2017.4.0'}
demux_emp_single_2 = demux.actions.emp_single(
  seqs=EMPSingleEndDirFmtimport_dir,
  barcodes=metadata2_txt_ColumnName,
  rev_comp_barcodes=False,
  rev_comp_mapping_barcodes=True,
)

# versions: {'dada2': '2017.4.0'}
dada2_denoise_single_2 = dada2.actions.denoise_single(
  demultiplexed_seqs=demux_emp_single_2.per_sample_sequences,
  trunc_len=150,
  trim_left=0,
  max_ee=2.0,
  trunc_q=2,
  chimera_method='pooled',
  min_fold_parent_over_abundance=1.0,
  n_threads=1,
  n_reads_learn=1000000,
  hashed_feature_ids=True,
)

# versions: {'framework': '2017.4.0'}
EMPSingleEndDirFmtimport_dir = qiime2.Artifact.import_data(
  'RawSequences', 'EMPSingleEndDirFmtimport_dir', view_type='EMPSingleEndDirFmt'
)

# versions: {'demux': '2017.4.0'}
demux_emp_single_3 = demux.actions.emp_single(
  seqs=EMPSingleEndDirFmtimport_dir,
  barcodes=metadata3_txt_ColumnName,
  rev_comp_barcodes=False,
  rev_comp_mapping_barcodes=True,
)

# versions: {'dada2': '2017.4.0'}
dada2_denoise_single_3 = dada2.actions.denoise_single(
  demultiplexed_seqs=demux_emp_single_3.per_sample_sequences,
  trunc_len=150,
  trim_left=0,
  max_ee=2.0,
  trunc_q=2,
  chimera_method='pooled',
  min_fold_parent_over_abundance=1.0,
  n_threads=1,
  n_reads_learn=1000000,
  hashed_feature_ids=True,
```

```
)

# versions: {'framework': '2017.4.0'}
EMPSingleEndDirFmtimport_dir = qiime2.Artifact.import_data(
  'RawSequences', 'EMPSingleEndDirFmtimport_dir', view_type='EMPSingleEndDirFmt'
)

# versions: {'demux': '2017.4.0'}
demux_emp_single_4 = demux.actions.emp_single(
  seqs=EMPSingleEndDirFmtimport_dir,
  barcodes=metadata4_txt_ColumnName,
  rev_comp_barcodes=False,
  rev_comp_mapping_barcodes=True,
)

# versions: {'dada2': '2017.4.0'}
dada2_denoise_single_4 = dada2.actions.denoise_single(
  demultiplexed_seqs=demux_emp_single_4.per_sample_sequences,
  trunc_len=150,
  trim_left=0,
  max_ee=2.0,
  trunc_q=2,
  chimera_method='pooled',
  min_fold_parent_over_abundance=1.0,
  n_threads=1,
  n_reads_learn=1000000,
  hashed_feature_ids=True,
)

# versions: {'framework': '2017.4.0'}
EMPSingleEndDirFmtimport_dir = qiime2.Artifact.import_data(
  'RawSequences', 'EMPSingleEndDirFmtimport_dir', view_type='EMPSingleEndDirFmt'
)

# versions: {'demux': '2017.4.0'}
demux_emp_single_5 = demux.actions.emp_single(
  seqs=EMPSingleEndDirFmtimport_dir,
  barcodes=metadata5_txt_ColumnName,
  rev_comp_barcodes=False,
  rev_comp_mapping_barcodes=True,
)

# versions: {'dada2': '2017.4.0'}
dada2_denoise_single_5 = dada2.actions.denoise_single(
  demultiplexed_seqs=demux_emp_single_5.per_sample_sequences,
  trunc_len=150,
  trim_left=0,
  max_ee=2.0,
```

```
    trunc_q=2,
    chimera_method='pooled',
    min_fold_parent_over_abundance=1.0,
    n_threads=1,
    n_reads_learn=1000000,
    hashed_feature_ids=True,
)

# versions: {'feature-table': '2017.4.0'}
feature_table_merge_seq_data_1 = feature_table.actions.merge_seq_data(
    data1=dada2_denoise_single_5.representative_sequences,
    data2=dada2_denoise_single_4.representative_sequences,
)

# versions: {'feature-table': '2017.4.0'}
feature_table_merge_seq_data_2 = feature_table.actions.merge_seq_data(
    data1=feature_table_merge_seq_data_1.merged_data,
    data2=dada2_denoise_single_3.representative_sequences,
)

# versions: {'feature-table': '2017.4.0'}
feature_table_merge_seq_data_3 = feature_table.actions.merge_seq_data(
    data1=feature_table_merge_seq_data_2.merged_data,
    data2=dada2_denoise_single_2.representative_sequences,
)

# versions: {'feature-table': '2017.4.0'}
feature_table_merge_seq_data_4 = feature_table.actions.merge_seq_data(
    data1=feature_table_merge_seq_data_3.merged_data,
    data2=dada2_denoise_single_1.representative_sequences,
)

# versions: {'feature-classifier': '2017.5.0'}
feature_classifier_classify_sklearn_1 =
feature_classifier.actions.classify_sklearn(
    reads=feature_table_merge_seq_data_4.merged_data,
    classifier=feature_classifier_fit_classifier_naive_bayes_1.classifier,
    chunk_size=262144,
    n_jobs=4,
    pre_dispatch='2*n_jobs',
    confidence=0.7,
    read_orientation=None,
)

# versions: {'feature-table': '2017.4.0'}
feature_table_merge_1 = feature_table.actions.merge(
    table1=dada2_denoise_single_5.table,
    table2=dada2_denoise_single_4.table,
```

```
)

# versions: {'feature-table': '2017.4.0'}
feature_table_merge_2 = feature_table.actions.merge(
  table1=feature_table_merge_1.merged_table,
  table2=dada2_denoise_single_3.table,
)

# versions: {'feature-table': '2017.4.0'}
feature_table_merge_3 = feature_table.actions.merge(
  table1=feature_table_merge_2.merged_table,
  table2=dada2_denoise_single_2.table,
)

# versions: {'feature-table': '2017.4.0'}
feature_table_merge_4 = feature_table.actions.merge(
  table1=feature_table_merge_3.merged_table,
  table2=dada2_denoise_single_1.table,
)

# versions: {'feature-table': '2017.4.0'}
feature_table_filter_samples_1 = feature_table.actions.filter_samples(
  table=feature_table_merge_4.merged_table,
  min_frequency=2000,
  max_frequency=None,
  min_features=0,
  max_features=None,
  sample_metadata=None,
  where=None,
)

# versions: {'feature-table': '2017.4.0'}
feature_table_filter_samples_2 = feature_table.actions.filter_samples(
  table=feature_table_filter_samples_1.filtered_table,
  sample_metadata=metadata6_txt,
  min_frequency=0,
  max_frequency=None,
  min_features=0,
  max_features=None,
  where=None,
)

# versions: {'taxa': '2017.9.0.dev0+2.g2ebb91d'}
taxa_collapse_1 = taxa.actions.collapse(
  table=feature_table_filter_samples_2.filtered_table,
  taxonomy=feature_classifier_classify_sklearn_1.classifier,
  level=6,
)
```

```
# versions: {'longitudinal': '2018.8.0'}
longitudinal_feature_volatility_1 = longitudinal.actions.feature_volatility(
  table=taxa_collapse_1.collapsed_table,
  metadata=metadata7_txt,
  state_column='month',
  individual_id_column='studyid',
  cv=5,
  random_state=None,
  n_jobs=4,
  n_estimators=100,
  estimator='RandomForestRegressor',
  parameter_tuning=False,
  missing_samples='error',
)
```

## Figure 1c (c-taxa-barplot.qzv in Supplementary File 1)

Data were imported into QIIME 2 as multiplexed 2x150 MiSeq reads and demultiplexed. DADA2[11] was applied to single-end reads (as approximately 30% of reads failed to join due to the relatively short sequence length) with no trimming of reads. Taxonomy was assigned to the resulting amplicon sequence variants (ASVs) against the SILVA version 132 99% OTUs (trimmed to the 515F/806R region of the 16S) using q2-feature-classifier's `classify-sklearn` method[13].

### Command line interface steps

```
# versions: {'types': '2018.6.0'}
qiime tools import \
  --type 'FeatureData[Taxonomy]' \
  --input-path 7_level_taxonomy.txt \
  --input-format HeaderlessTSVTaxonomyFormat \
  --output-path 7_level_taxonomy.txt.qza

# versions: {'types': '2018.6.0'}
qiime tools import \
  --type 'FeatureData[Sequence]' \
  --input-path silva132_99.fna \
  --input-format DNAFASTAFormat \
  --output-path silva132_99.fna.qza

# versions: {'feature-classifier': '2018.6.0', 'types': '2018.6.0'}
qiime feature-classifier extract-reads \
  --i-sequences silva132_99.fna.qza \
  --p-f-primer 'GTGCCAGCMGCCGCGGTAA' \
  --p-r-primer 'GGACTACHVGGGTWTCTAAT' \
  --p-trunc-len 0 \
  --p-trim-left 0 \
  --p-identity 0.8 \
```

```
  --output-dir feature-classifier-extract_reads_1

# versions: {'feature-classifier': '2018.6.0', 'types': '2018.6.0'}
qiime feature-classifier fit-classifier-naive-bayes \
  --i-reference-reads feature-classifier-extract_reads_1/reads.qza \
  --i-reference-taxonomy 7_level_taxonomy.txt.qza \
  --p-classify--alpha 0.001 \
  --p-classify--chunk-size 20000 \
  --p-classify--class-prior 'null' \
  --p-no-classify--fit-prior \
  --p-no-feat-ext--alternate-sign \
  --p-feat-ext--analyzer 'char_wb' \
  --p-no-feat-ext--binary \
  --p-feat-ext--decode-error 'strict' \
  --p-feat-ext--encoding 'utf-8' \
  --p-feat-ext--input 'content' \
  --p-feat-ext--lowercase \
  --p-feat-ext--n-features 8192 \
  --p-feat-ext--ngram-range '[7, 7]' \
  --p-no-feat-ext--non-negative \
  --p-feat-ext--norm 'l2' \
  --p-feat-ext--preprocessor 'null' \
  --p-feat-ext--stop-words 'null' \
  --p-feat-ext--strip-accents 'null' \
  --p-feat-ext--token-pattern '(?u)\b\w\w+\b' \
  --p-feat-ext--tokenizer 'null' \
  --output-dir feature-classifier-fit_classifier_naive_bayes_1

# versions: {'types': '2018.8.0'}
qiime tools import \
  --type 'SampleData[SequencesWithQuality]' \
  --input-path se-64-manifest.csv \
  --input-format SingleEndFastqManifestPhred64 \
  --output-path se-64-manifest.csv.qza

# versions: {'quality-filter': '2018.8.0', 'types': '2018.8.0'}
qiime quality-filter q-score \
  --i-demux se-64-manifest.csv.qza \
  --p-min-quality 4 \
  --p-quality-window 3 \
  --p-min-length-fraction 0.75 \
  --p-max-ambiguous 0 \
  --output-dir quality-filter-q_score_1

# versions: {'types': '2018.8.0', 'deblur': '2018.8.0'}
qiime deblur denoise-16S \
  --i-demultiplexed-seqs quality-filter-q_score_1/filtered_sequences.qza \
  --p-trim-length 85 \
```

```
  --p-no-sample-stats \
  --p-mean-error 0.005 \
  --p-indel-prob 0.01 \
  --p-indel-max 3 \
  --p-min-reads 10 \
  --p-min-size 2 \
  --p-jobs-to-start 8 \
  --p-hashed-feature-ids \
  --output-dir deblur-denoise_16S_1

# versions: {'types': '2018.8.0', 'feature-table': '2018.8.0'}
qiime feature-table filter-samples \
  --i-table deblur-denoise_16S_1/table.qza \
  --m-metadata-file metadata1.txt \
  --p-min-frequency 2000 \
  --p-min-features 0 \
  --p-where "Site='Steep'" \
  --p-no-exclude-ids \
  --output-dir feature-table-filter_samples_1

# versions: {'types': '2018.8.0', 'feature-table': '2018.8.0'}
qiime feature-table filter-features \
  --i-table feature-table-filter_samples_1/filtered_table.qza \
  --p-min-frequency 100 \
  --p-min-samples 0 \
  --p-no-exclude-ids \
  --output-dir feature-table-filter_features_1

# versions: {'types': '2018.8.0', 'feature-table': '2018.8.0'}
qiime feature-table filter-seqs \
  --i-data deblur-denoise_16S_1/representative_sequences.qza \
  --i-table feature-table-filter_features_1/filtered_table.qza \
  --p-no-exclude-ids \
  --output-dir feature-table-filter_seqs_1

# versions: {'feature-classifier': '2018.8.0', 'types': '2018.8.0'}
qiime feature-classifier classify-sklearn \
  --i-reads feature-table-filter_seqs_1/filtered_data.qza \
  --i-classifier feature-classifier-fit_classifier_naive_bayes_1/classifier.qza \
  --p-reads-per-batch 0 \
  --p-n-jobs 8 \
  --p-pre-dispatch '2*n_jobs' \
  --p-confidence 0.7 \
  --output-dir feature-classifier-classify_sklearn_1

# versions: {'types': '2018.11.0.dev0', 'feature-table':
'2018.11.0.dev0+3.g345ac6e'}
qiime feature-table group \
```

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27295v2 | CC BY 4.0 Open Access | rec: 3 Dec 2018, publ: 3 Dec 2018

```
  --i-table feature-table-filter_features_1/filtered_table.qza \
  --m-metadata-file metadata2.txt \
  --m-metadata-column 'ColumnName' \
  --p-axis 'sample' \
  --p-mode 'median-ceiling' \
  --output-dir feature-table-group_1

# versions: {'taxa': '2018.11.0.dev0', 'types': '2018.11.0.dev0'}
qiime taxa barplot \
  --i-table feature-table-group_1/grouped_table.qza \
  --i-taxonomy feature-classifier-classify_sklearn_1/classification.qza \
  --m-metadata-file metadata3.txt \
  --output-dir taxa-barplot_1
```

## Application programmer interface steps

```
import qiime2
from qiime2.plugins import feature_table
from qiime2.plugins import feature_classifier
from qiime2.plugins import taxa
from qiime2.plugins import quality_filter
from qiime2.plugins import deblur

metadata1_txt = qiime2.Metadata.load('metadata1.txt')
metadata2_txt_ColumnName =
qiime2.Metadata.load('metadata2.txt').get_column('ColumnName')
metadata3_txt = qiime2.Metadata.load('metadata3.txt')

# versions: {'types': '2018.6.0'}
7_level_taxonomy_txt = qiime2.Artifact.import_data(
  'FeatureData[Taxonomy]', '7_level_taxonomy.txt',
view_type='HeaderlessTSVTaxonomyFormat'
)

# versions: {'types': '2018.6.0'}
silva132_99_fna = qiime2.Artifact.import_data(
  'FeatureData[Sequence]', 'silva132_99.fna', view_type='DNAFASTAFormat'
)

# versions: {'feature-classifier': '2018.6.0', 'types': '2018.6.0'}
feature_classifier_extract_reads_1 = feature_classifier.actions.extract_reads(
  sequences=silva132_99_fna,
  f_primer='GTGCCAGCMGCCGCGGTAA',
  r_primer='GGACTACHVGGGTWTCTAAT',
  trunc_len=0,
  trim_left=0,
  identity=0.8,
)
```

```
# versions: {'feature-classifier': '2018.6.0', 'types': '2018.6.0'}
feature_classifier_fit_classifier_naive_bayes_1 =
feature_classifier.actions.fit_classifier_naive_bayes(
  reference_reads=feature_classifier_extract_reads_1.reads,
  reference_taxonomy=7_level_taxonomy_txt,
  classify__alpha=0.001,
  classify__chunk_size=20000,
  classify__class_prior='null',
  classify__fit_prior=False,
  feat_ext__alternate_sign=False,
  feat_ext__analyzer='char_wb',
  feat_ext__binary=False,
  feat_ext__decode_error='strict',
  feat_ext__encoding='utf-8',
  feat_ext__input='content',
  feat_ext__lowercase=True,
  feat_ext__n_features=8192,
  feat_ext__ngram_range='[7, 7]',
  feat_ext__non_negative=False,
  feat_ext__norm='l2',
  feat_ext__preprocessor='null',
  feat_ext__stop_words='null',
  feat_ext__strip_accents='null',
  feat_ext__token_pattern='(?u)\b\w\w+\b',
  feat_ext__tokenizer='null',
)

# versions: {'types': '2018.8.0'}
se-64-manifest_csv = qiime2.Artifact.import_data(
  'SampleData[SequencesWithQuality]', 'se-64-manifest.csv',
view_type='SingleEndFastqManifestPhred64'
)

# versions: {'quality-filter': '2018.8.0', 'types': '2018.8.0'}
quality_filter_q_score_1 = quality_filter.actions.q_score(
  demux=se-64-manifest_csv,
  min_quality=4,
  quality_window=3,
  min_length_fraction=0.75,
  max_ambiguous=0,
)

# versions: {'types': '2018.8.0', 'deblur': '2018.8.0'}
deblur_denoise_16S_1 = deblur.actions.denoise_16S(
  demultiplexed_seqs=quality_filter_q_score_1.filtered_sequences,
  trim_length=85,
  sample_stats=False,
```

```
  mean_error=0.005,
  indel_prob=0.01,
  indel_max=3,
  min_reads=10,
  min_size=2,
  jobs_to_start=8,
  hashed_feature_ids=True,
)

# versions: {'types': '2018.8.0', 'feature-table': '2018.8.0'}
feature_table_filter_samples_1 = feature_table.actions.filter_samples(
  table=deblur_denoise_16S_1.table,
  metadata=metadata1_txt,
  min_frequency=2000,
  max_frequency=None,
  min_features=0,
  max_features=None,
  where="Site='Steep'",
  exclude_ids=False,
)

# versions: {'types': '2018.8.0', 'feature-table': '2018.8.0'}
feature_table_filter_features_1 = feature_table.actions.filter_features(
  table=feature_table_filter_samples_1.filtered_table,
  min_frequency=100,
  max_frequency=None,
  min_samples=0,
  max_samples=None,
  metadata=None,
  where=None,
  exclude_ids=False,
)

# versions: {'types': '2018.8.0', 'feature-table': '2018.8.0'}
feature_table_filter_seqs_1 = feature_table.actions.filter_seqs(
  data=deblur_denoise_16S_1.representative_sequences,
  table=feature_table_filter_features_1.filtered_table,
  metadata=None,
  where=None,
  exclude_ids=False,
)

# versions: {'feature-classifier': '2018.8.0', 'types': '2018.8.0'}
feature_classifier_classify_sklearn_1 =
feature_classifier.actions.classify_sklearn(
  reads=feature_table_filter_seqs_1.filtered_data,
  classifier=feature_classifier_fit_classifier_naive_bayes_1.classifier,
  reads_per_batch=0,
```

```
  n_jobs=8,
  pre_dispatch='2*n_jobs',
  confidence=0.7,
  read_orientation=None,
)

# versions: {'types': '2018.11.0.dev0', 'feature-table':
'2018.11.0.dev0+3.g345ac6e'}
feature_table_group_1 = feature_table.actions.group(
  table=feature_table_filter_features_1.filtered_table,
  metadata=metadata2_txt_ColumnName,
  axis='sample',
  mode='median-ceiling',
)

# versions: {'taxa': '2018.11.0.dev0', 'types': '2018.11.0.dev0'}
taxa_barplot_1 = taxa.actions.barplot(
  table=feature_table_group_1.grouped_table,
  taxonomy=feature_classifier_classify_sklearn_1.classification,
  metadata=metadata3_txt,
)
```

## Figure 1d (d-ili-plot.qzv in Supplementary File 1)

The input files for this visualization are a stereolithography file (STL) and a sample metadata file with a mapping between samples and the spatial coordinates (`x`, `y` and `z`). Both files were obtained from `ili's GitHub page[15,16]. The comma-separated file was converted into a tab-separated format (to make it compatible with QIIME 2).

### Command line interface steps

```
# versions: {'ili': 'v0.1.1'}
qiime tools import \
  --type 'Model' \
  --input-path model.stl \
  --input-format STLFile \
  --output-path model.stl.qza

# versions: {'ili': 'v0.1.1'}
qiime ili plot \
  --i-model model.stl.qza \
  --m-metadata-file metadata1.txt \
  --output-dir ili-plot_1
```

### Application programmer interface steps

```
import qiime2
from qiime2.plugins import ili
```

```
metadata1_txt = qiime2.Metadata.load('metadata1.txt')

# versions: {'ili': 'v0.1.1'}
model_stl = qiime2.Artifact.import_data(
  'Model', 'model.stl', view_type='STLFile'
)

# versions: {'ili': 'v0.1.1'}
ili_plot_1 = ili.actions.plot(
  model=model_stl,
  metadata=metadata1_txt,
)
```

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27295v2 | CC BY 4.0 Open Access | rec: 3 Dec 2018, publ: 3 Dec 2018

# Supplementary File 3: Notes from end users on QIIME 2

We solicited feedback from a broad contingent of QIIME 2 users and developers (N = 15), asking them to review different aspects of QIIME 2 that they use in their research. The majority of these reviewers are independent users whom we selected on the basis of their activity on the QIIME 2 Forum. Four reviewers (identified below) are co-authors of this article who contributed to QIIME 2 (e.g., in the form of plugins compatible with QIIME 2, documentation, or other major contributions), but are currently unaffiliated with the Caporaso or Knight research groups (the two groups who derive direct funding from the primary QIIME 2 NSF grant). Their use of QIIME 2 is what initiated involvement in QIIME 2 development, and hence we feel that these investigators can provide unique insight on how QIIME 2 attracts a diverse community of users and developers. Reviewers are listed in alphabetical order. All reviewers gave permission to be quoted in this document, and approved the inclusion of their statements as they appear here (e.g., in some cases, typos were corrected or statements were abridged).

Some end-users have commented specifically on plugins that are included in the worked examples in the text (Fig 1), including q2-emperor (Fig 1a) and q2-longitudinal (Fig 1b).

**Ahmed Abdelfattah, PhD.**
**Postdoctoral Fellow**
**Department of Ecology, Environment and Plant Sciences, Stockholm University, Sweden**

> QIIME is by far my favorite pipeline for microbial data analysis. I started using it while I was doing my PhD. And I have not stopped since. While very comprehensive, it is user-friendly, even for those who have little experience in bioinformatics. QIIME is a perfect example of open source software, making it both versatile and able to cope with the ever-expanding field of microbial ecology. QIIME 2 takes microbial data analysis a big step further. While keeping the main essential features of QIIME 1, the newly added plugins are very useful. Perhaps my favorite plugin is q2-longitudinal, a plugin for longitudinal and paired-sample analyses. This plugin contains pipelines, methods, and visualizers, all of which provide complex analysis to evaluate paired samples over a period of time.

**Rozlyn C.T. Boutin, BScH**
**MD/PhD student**
**University of British Columbia, Vancouver, Canada**

> I have been working with a range of features available in QIIME 2, including many of those described in the “Moving Pictures Tutorial”. I have found each of these plugins straightforward to use and well described in the online tutorial. I found it especially useful that the tutorial and QIIME 2 websites describe what each plugin does, what parameters in each analysis can be changed and how, as well as provide guidance regarding how to modify the parameters of a command to more appropriately fit the study in question.
>
> More recently, I have been using the q2-longitudinal plugin to compute a gut microbiome “maturity index” for stool samples collected at multiple time points from the same subjects and ultimately compare how the maturity of the gut microbiome differs between subjects with different disease states... The ability to perform analyses comparing gut microbiota maturity between groups of individuals with differing disease susceptibilities demonstrates the unique ability of QIIME 2 to perform high-level bioinformatic and statistical analyses while also taking complex biological contexts and ecological factors into consideration. Indeed,

QIIME 2 does a great job of bridging the gaps in language and scientific expertise between the disciplines of biology, bioinformatics, and statistics. Moreover, the available plugins accommodate novel, complex analyses being implemented in the latest literature.

While R code for gut microbiome maturity index predictions is available, the QIIME 2 plugin provides an easy-to-use platform with customizable input and output options. User-friendly tuning parameters, such as the ability to add spaghetti vectors to volatility charts, modify the fraction of samples used for training vs. testing of the regressor, stratify training and test data among metadata categories, and set a seed to ensure repeatability among users working with the same data are all useful and relevant features of the plugin that are easily and intuitively manipulated... Although it is not possible to obtain a microbiota-by-age Z (MAZ) score for samples used to train the model, the output of the q2-longitudinal maturity index prediction provides both intuitive visualization output files as well as the option of downloading a TSV file containing a MAZ score for samples not used for training, which can subsequently be used in downstream analyses. Allowing for yet more advanced downstream analyses, the plugin also provides outputs with detailed information on how to interpret the output of the command; for instance, the feature-importance scores can be exported to determine which features are most important for discriminating microbial maturity indices, and how each feature changes over time in each experimental group can be visualized in the cluster map output. Each of these output files are intuitive to interpret and the resulting visualization files aesthetically pleasing.

All of the plugins I have used in QIIME 2 thus far have been easy to use and I have been impressed by the advanced and innovative analyses available. These tools are complemented with a well maintained user website and easy-to-follow tutorials that provide helpful guidance without burdening the reader with too many details. Details, however, are still available elsewhere on the website for those who are interested and the QIIME 2 forum is exceedingly helpful and responsive.

**David J. Bradshaw II**
**Ph.D. Student**
**Florida Atlantic University - Harbor Branch Oceanographic Institute**

QIIME 2 has been essential to my PhD work, allowing me an easy to use and very flexible way to analyze my 16S rRNA amplicon sequences. My colleagues and I also plan on using this system for 18S analysis. I have been a user of QIIME 2 for about a year now, and I have been able to teach myself how to use it with the assistance of their well written guides, along with some very timely and useful advice from creators and users. I do not have access to a microbiome expert at my school, so these resources have been very important in helping me complete my work. QIIME 2's system of plugins basically allows me to do anything that I need to do in order to make sure that I have the best quality sequences for analysis and also provides many ways to analyze the data for my dissertation and publications. There are so many different ways to analyze microbiome data, and QIIME 2 is robust enough to allow you to analyze it however you prefer, such as using the Deblur, vsearch, or DADA2. **[note from the QIIME 2 authors: these are QIIME 2 plugins available to perform various actions for denoising, dereplication, and/or OTU clustering.]**

**Lorinda Bullington**
**Molecular Ecologist**
**MPG Ranch, Missoula, Montana**

The developers of the QIIME 2 bioinformatics platform foster an interactive, collaborative environment wherein users can directly influence the functional ability of the platform. The versatility of the many plugins allows researchers to use a single platform to analyze amplicon data from various primer pairs and multiple target organisms including bacteria, fungal endophytes, and arbuscular mycorrhizal fungi. The QIIME 2 forum provides quick and thorough answers to questions and allows direct communication between users and developers which greatly increases QIIME 2's accessibility for early career scientists. Data provenance tracking ensures reproducible results and is very helpful when returning to old datasets or merging new ones. These aspects, along with the excellent data visualizations associated with each step help to eliminate the black box that is so often associated with bioinformatics pipelines.

**Justine W. Debelius, Ph.D.**
**Postdoctoral Scholar**
**Karolinska Institutet Dept. of Medical Epidemiology and Biostatistics, Stockholm, Sweden**
Dr. Debelius is not an author of this paper, but worked previously in the laboratory of Dr. Rob Knight and has taught QIIME 1 workshops.

As someone who considers herself somewhere on the advanced to expert level of user, I love the capacity [in QIIME 2] to have multiple interfaces. I appreciate that I can generate an object in QIIME 2 and then use a standard python API to pull it into a notebook for my own modifications or perform more complex operations. It means that I can easily perform more complex analyses (like specialised analyses with Gneiss) that aren't possible via the command line interface.

I'm also a huge fan of QIME 2 view: it makes it super easy to share results with my PI and collaborators. We're discussing including a QIIME 2 view object in the supplement of a paper I'm a co-author of, because it makes the interaction easier.

Finally, I like the more "choose your own adventure" aspect of the plugin architecture. Rather than trying to structure a parameters file in QIIME 2, it's much easier to choose the best or preferred algorithm for each step. I can even mix tools: paired end joining in vsearch, denoising in Deblur, and chimera slaying with another algorithm...

**Claire Duvallet**
**Ph.D. Candidate**
**Dept. of Biological Engineering, Massachusetts Institute of Technology, Cambridge, MA**
Dr. Duvallet is a co-author of this paper.

My favorite (non-obvious) thing about QIIME 2 is that it provides a really easy "in" for people to join the ranks of the "legit" microbiome research community. I think this works in two big ways:

1. Researchers who work at institutions without many resources now have access to the latest and greatest tools in our field. I love that QIIME 2 does a lot to even the playing field and open up this field to many more researchers. There are many axes to this: scientists in developing countries, clinicians without much data background, citizen scientists who got their microbiomes sequenced, etc — all now have access where before they didn't! QIIME 2 is unique in this way over other bioinformatics software because there are so many different ways to engage with it, and it was obviously created with this impact in mind.
2. Established researchers now get to add their "school of thought" to the mainstream conversation... Now, with QIIME 2, I get to write in my school of thought. I can write plugins for my favorite methods, add parameters that are missing in existing wrappers, give my perspective on how things "should" be done on the forum, and incorporate my school of thought into sort-of canon via writing tutorials or documentation... Now, if you don't like what the field is doing, you can actively change it!

A few other more specific experiences from my own interactions with QIIME 2:

1. The plugin philosophy makes it suuuuper easy to get any new method I make into as many hands as possible, and really incentivizes making code more useable. For example, I helped a postdoc in our lab develop a new method to normalize data to reduce batch effects, and was really excited when Greg [Caporaso] reached out and asked if we wanted to add our method into QIIME 2. The common framework that many people use and which provided a clear approach for how to convert our scripts into a bona-fide tool was the nudge I needed to convince me to put in the extra effort to turn our scripts into something more useable (the method is now available in QIIME 2 as q2-perc-norm). Everybody wins here: the community has a new method that is easy to install and use, and I get to share my method broadly and also say that I've contributed to open-source software development.
2. Although I don't personally use QIIME 2 that much (I'm past most of the heavy data-processing parts of my PhD), I tell all my peers to use it if only because of the convenience of the platform: having one standard framework that wraps existing functions across varying software packages and tools is so nice because installation issues aren't such huge obstacles and you don't have to constantly write intermediate scripts to reformat input and output files. It's just so nice to have one bash script that can go from raw data to feature table in a really streamlined way.
3. Before I "converted" to QIIME 2, I was the main person in charge of maintaining and developing our lab's in-house data processing pipeline. This took me between a few hours to a few days per month, ranging from helping lab members learn to use the pipeline to fixing bugs to implementing new functionalities. Now that QIIME 2 exists, I can point lab members to the forum for help, raise issues for the core developers to address when I find difficult bugs (or solve them myself if they are easy to address), and spend the rest of that time saved on either my own research or on adding new functionality to QIIME 2 that is unique from my lab. As a specific example, I'm pretty sure it took me more time to incorporate distribution-based OTU clustering into my lab's pipeline (which serves a maximum of 20 people) than it did to write it up as a plugin (which can theoretically serve thousands of people as q2-dbotu). It's clear that joining the open-source QIIME 2 community is a much better investment of everybody's time!

**Erika Korzune Ganda, DVM, Ph.D.**
**Postdoctoral Associate**
**Department of Food Science, Cornell University, Ithaca, NY**

I could not have finished my PhD without the software and the amazing help of the developers through the forum…

I love how I can save my code and be able to go back to it six months later and not only know exactly what I did, but also be able to analyze a new dataset with much more ease. And it is at this point that QIIME 2 offers an amazing advantage with provenance tracking and UUID identifiers for reproducible identification of sequence variants.

In addition to making analysis more straightforward and traceable, the ability to share visualization files over email with collaborator is an AMAZING feature. I am a veterinarian that knows a little about coding and microbiome research. Now I am able to collaborate with clinicians working on data on the microbiome of bovine rumen, companion animals' mouths with periodontal disease, milk, salmon and trout skin, and murine feces. And the best of it is that I can email a .qzv [QIIME 2 visualization] file with some explanation on variable names, and they can then look at the interactive interface without having to go through any coding (which can be pretty scary, if you are not used to it). The web-based interface makes our meetings and collaborations much more effective.

Another great thing is that we are not hostage to the outputs of view.qiime2.org. I can, at any point of my analysis, extract my raw data and make plots using my software of preference.

However, it is not enough to have a good interface without good people working behind the scenes. I give kudos to the developer team that answered my endless questions and helped me through numerous error messages from 2016 to today...

In terms of data analysis, it is no secret that microbiome research is a rapidly evolving field. It is important to keep up to date with the most appropriate analysis methods, as there is still a lot of debate in terms of what the most appropriate data handling and analysis techniques are. This is another advantage of the plugin-based architecture of QIIME 2. It allows for people working on the forefront of data analysis to make their tools available to people like me, who wouldn't necessarily come across them if they weren't available in a user-friendly platform.

**Dr. Alexander Mahnert**
**Postdoctoral researcher**
**Department of Internal Medicine, Medical University of Graz, Austria**

I'm now working with QIIME 2 since the Virtual box image of version 2.0.6. Back then I was already an experienced user of QIIME 1 due to regular analysis of amplicon and shotgun data. I got very excited about QIIME 2 when I realized its main improvements over QIIME 1.9.1: 1st, denoising instead of clustering, 2nd, defined semantic types for different data formats, and 3rd, classifiers to assign taxonomy. We already used denoising algorithms like DADA2 before they were implemented in QIIME since we realized that they usually give us a more conservative view on the profile (e.g., OTUs, RSVs, ASVs) of human-associated Archaea[1]. This was a first argument to use QIIME 2 for our data analysis so that we could circumvent inconvenient switching from QIIME 1.9.1 to R and back again. Another argument was my curiosity about the clear core concept of QIIME 2 with artifacts, defined semantics, plugins, methods and visualizers. This clear concept not only guarantees the correct usage of a certain plugin or method, but also helps to keep track of a bunch of qza and qzv files during your analysis by the possibility to check the provenance of each file. In addition, it also saves a lot of hard disk space in the frame of an analysis. And finally, our study[2] on the microbial dynamics in an isolated and confined built environment (ICE) and the chance to apply plugins like q2-longitudinal on our >500 day observation of some built environment surfaces sealed my transition to QIIME 2 (although we had to publish the manuscript without the longitudinal analysis of QIIME 2 due to time limitations). I'm a big fan of this plugin and its methods like pairwise difference and distance, linear mixed effect models, first rate of change and especially the volatility analysis. All is so well documented and provides a lot of possibilities to visualize your data. And after several improvements to the error messages of certain methods, I think even beginners can make a good analysis of their longitudinal datasets. Now, after the use of QIIME 2 for almost 2 years now, I think it is a good moment to make a first conclusion about the pro's and con's:

First of all, QIIME 2 (as QIIME 1) has a great documentation with elaborate tutorials to give users a step by step introduction to certain plugins (for beginners – https://docs.qiime2.org/2018.11/tutorials/overview/ or experienced users - https://docs.qiime2.org/2018.11/tutorials/qiime2-for-experienced-microbiome-researchers/). With example data, many figures and tables, the developers do not only make sure that users apply their tools for the right purpose, but also help them to understand a tool and to learn how to interpret their own results. And if something is still unclear or buggy, the active QIIME 2 forum is a good address to solve many obstacles.

However, as I now realize that my lines about QIIME 2 are pretty positive I also want to indicate several places for improvements. Not all plugins are covered by such great tutorials as in q2-longitudinal, q2-gneiss, q2-quality-control, q2-feature-classifier and q2-sample-classifier. It would be great to see more elaborate tutorials also for other plugins [e.g., q2-composition and q2-picrust2]. I'm a big fan of interactive plots like q2-emperor and [the volatility action in q2-longitudinal]. I think this interactivity is a big part of the fun you experience while using QIIME 2 for your data analysis. Therefore, more options to adjust simple things like colors or export each generated plot also as a svg file would be a nice add in my eyes. Depending on your input data, it might be still necessary to use QIIME 1 before data import to QIIME 2. I think this is a nasty peculiarity. Why isn't there a plugin in QIIME 2 yet to extract barcodes from fastq files to allow direct compatibility with the available import options? ...

To come to an end – I think QIIME 2 is one of the best free packages to analyze amplicon data at the moment and already quite complete. For sure there are still things to update and improve and always will be, but from my point of view there is no sense to teach students still how they could work with QIIME 1.9.1. when there is QIIME 2 available. There are already a few publications out there where we used QIIME 2 for the data analysis[3] and there are plenty waiting to be wrapped up into a manuscript. I even like to re-analyze old data processed in QIIME 1 and use QIIME 2 to get a new perspective on it.

Finally, I want to thank the developers of QIIME for QIIME 2, that this software is free and not commercial, and encourage all of them to keep on improving methods and adding tools to the Quantitative Insights Into Microbial Ecology!

**Dr. Melanie C Melendrez**
**Computational Microbiologist**
**St. Cloud State University, St. Cloud, MN**

One of the best aspects of the QIIME 2 workflow is provenance and command consistency. While the learning curve for QIIME could be considered a bit of a climb, once you become familiar with syntax and if you aren't shy about utilizing and searching the forum, the curve is less of a climb and more like a steady walk uphill that is not unpleasant.

I think it's fantastic the developers of QIIME have recognized the need for consistent tools in the area of metabolomics, metatranscriptomics, and metabolic pathway inference. There's such an inundation in the literature recently of microbiome studies and investigators are under increasing pressure to ensure their study doesn't become a J.A.M - (just another microbiome) study/session. The challenge in expanding beyond 'who is there' is the lack of correspondence and formats required by all the different tools that would have to be navigated; and, in the end, each tool may have a different sequence quality control workflow, a different taxonomic characterization database requirement (or workflow) or a different format requirement - forcing the investigator or manipulate the data in a separate framework that may introduce errors or inconsistency. These are examples of differences that would then potentially greatly impact more complex downstream analyses involving metabolic pathway inferences, metabolomics or metatranscriptomics. By developing plugins for the QIIME 2 framework to address these downstream, more complex analyses, all data stays in the same framework, consistent formats that can be exported to known formats or shared directly through QIIME 2 view, is subject to the same quality control protocols (which are adjustable), analyzed using the same databases and versions of databases. It greatly enhances and streamlines the ability to do reproducible science. In short, it's neat.

I am also looking forward to q2-metaphlan2 and q2-SCNIC as I have used metaphlan in the past and I am moving toward network analysis in the future. **[note from the QIIME 2 authors: these are plugins for**

**shotgun metagenome taxonomic profiling and network analysis, respectively, which will facilitate multi-omics analysis in QIIME 2.]**

I am particularly fond of QIIME 2 view. It is very useful as I have worked with a variety of studies and it is great to have the option of sharing my results without requiring those on the receiving end to figure out how to get the program, install it, and run it in order to see the results I generated.

I am really good at breaking things… and the QIIME forum has become an indispensable tool for me. The QIIME platform is highly utilized in part because of its fantastic user support. If I have issues with linux I go to stackoverflow typically. For various bioinformatics tools they often have no user support beyond Github issues or a Google group.  They are useful but what is compelling about using QIIME as a scientist is the non-judgemental, seriously-we-want-to-help-you vibe of a forum dedicated to this platform... QIIME developers and the forum community as a whole are committed, timely with their responses to requests, and have been very friendly from the most basic syntax mistakes to the more complex conceptual or statistical error requests. They are also open to assisting in results interpretation to ensure you are using the plugins correctly.

Currently I am educator that builds all my informatics teaching modules from scratch - I would incorporate QIIME 2 in my teaching as a basic platform to learn about metagenomics and teach students how to analyze microbiome data as well as teach the value of reproducibility and consistency in study design and analysis. I also use it as an example of 'how' scientific inquiry incorporating computational biology should be done — collaboratively; we are all works in progress, moving forward in a field that changes rapidly.

**Devon O'rourke**
**Ph.D. Student**
**University of New Hampshire**

There are a lot of technical reasons to love QIIME 2, but most of my reasons have to do with collaboration. Provenance is a game changer when working in groups - it's exactly the data management version history platform you'd dream up (and greatly simplifies citation tracking to boot). There's obviously a diversity of tools to QC data, merge projects, explore suites of diversity metrics and visualizations, but the biggest asset in terms of collaboration is the ease with which an artifact file is shared - just provide access to a file and you can view it on a browser on any machine. My favorite part though is the broader community with which the QIIME ecosystem exists - when you're thinking about what tools you can depend on, reach out for help with, and possibly contribute back to, there's nothing like it. The documentation is superb, but the community forum (and its members) is ultimately what propels me through the challenges I routinely face.

**Adam R. Rivers, Ph.D.**
**Computational Biologist**
**USDA-ARS, Gainesville, FL**
Dr. Rivers is a co-author of this paper.

I recently developed a stand-alone software package, ITSxpress, to trim fungal sequences for amplicon sequencing analysis and decided to extend it to integrate into QIIME 2. Working with the QIIME 2 community has been a very positive experience. QIIME is a very popular bioinformatics package. In the 8 years since its initial release the paper for QIIME 1 has had over 12,700 citations[4]. Creating software for this preexisting user base has helped my software gain much broader adoption. Since introducing the QIIME 2 plugin several months ago I've had 473 downloads of the Bioconda package for ITSxpress. Having a larger user base has also helped me improve the quality of the software. QIIME 2 users have identified several bugs that I have been able to rapidly fix. QIIME 2 has a very active user forum and I think that has encouraged users to report issues to me that they may not have reported if they had to email me as the maintainer of a stand-alone package.

QIIME 2 has a very professional design that allows for data provenance tracking, semantic typing and a nice API and good file checking functionalities. The software introduces a number of concepts that you need to understand before writing a package. The documentation for developers is improving, but some more explanation about data types and transformers would be helpful. Fortunately, the QIIME 2 development team is extremely helpful and welcoming to new contributors. I intend to develop more tools for QIIME 2 in the future.

**Biswarup Sen**
**Associate Professor**
**Tianjin University, School of Environmental Science and Engineering, Tianjin, China**

With the rapid growth in the application of high-throughput sequence (HTS) data in contemporary research, the need of a multitasking bioinformatics platform becomes inevitable. The contribution of QIIME 2 as one of the most favorite bioinformatics platforms is unmeasurable in the advancement of HTS data analysis. As a QIIME 2 user and a member of the QIIME 2 forum, I have found QIIME 2 as the most user-friendly and versatile platform that can perform a myriad of analyses rapidly and with much less effort. The tutorials that are provided by the QIIME 2 developers are comprehensive and one with a minimal training on executing commands can begin to use this platform on the chosen operating system. I found the QIIME 2 forum very helpful in troubleshooting the issues that are often faced while running the QIIME 2 plugins. The workflow designed by the developers is well-structured and easy to grasp for initiating data analysis starting from the import of raw sequence data files to taxonomy assignment. QIIME 2 is applicable for analysis of both prokaryotic and eukaryotic communities hailing from any environment, known or underexplored. Besides, the opportunity for other developers to add new plugins allows the creation of useful plugins that are task-specific, for example the ITSxpress plugin can do a quick trimming of the flanking regions in the fungal ITS1/ITS2 sequences, which results in accurate taxonomic assignments down to species level. The ability of post-analysis with the QIIME 2 artifacts on other platforms, e.g. R, makes QIIME 2 more user-friendly and allows the user to perform all statistical analysis and generate high-quality and intuitive plots ready for publication. The developers of QIIME 2 are indeed performing a huge task of providing such a versatile platform which undergoes continuous evaluation, troubleshooting, and upgradation in order to provide the user the ease to conduct their analysis without much challenges and hurdles.

It would take pages for me to describe the various advantages and the variety of tools and plugins wrapped in QIIME 2 platform. In conclusion, I would strongly recommend QIIME 2 for the modern data scientists and young researchers working on HTS data analysis. Appended are a couple of my research group's latest articles where the QIIME platform was used[5,6]. A few others are in the pipeline.

**Solveig Tangedal, M.D.**
**Dept. of Thoracic Medicine, Haukeland University Hospital, Bergen, Norway**
**Ph.D. Candidate, Dept. of Clinical Science, Faculty of Medicine and Dentistry, University of Bergen, Norway**

I am a medical doctor, and a PhD candidate studying microbiota in airways disease. QIIME 1 was our preferred pipeline for the first article published on microbiota[7], and it was natural to use QIIME 2 for our next paper.

With the transition from QIIME 1 to QIIME 2, several useful tools have been made available. The integration of DADA2 as a wrapper including also chimera removal has provided a more accurate separation of sequences into units for analyses (amplicon sequence variants) compared to QIIME 1 operational taxonomic units. Different visualization tools like the quality plots improve the understanding of how sequencing data are processed before accepting sequences for final analyses. The pipeline also allows for better control of how processes change the datasets during different filtering steps. New statistical, analytical tools are implemented for longitudinal data and for evaluating dynamics in taxonomic compositions in samples. For my data this could be further improved with an even more comprehensive development for paired analyses with only two timepoints. This would also improve analyses for many other medical research studies with two sampling points.

The output data can be made available for other analytical tools like R... This ensures that even if certain tools are not made available directly in QIIME 2, the data derived from the pipeline still can be analyzed with appropriate statistical tests.

For us it is especially useful to have access to direct advice from the QIIME 2 development team at the QIIME 2 forum. This helps researchers in my group make informed decisions, and the forum contributes to a continuous development and improvement of the pipeline. Also available are tutorials that clarify how the different plugins work. Without a background in bioinformatics and statistics, the forum and tutorials are an absolute must to be able to run the plugins correctly. All in all, QIIME 2 is a free tool making complex and advanced data curation possible for me as a researcher with no pre-existing knowledge of bioinformatics or computer programming. From my experience, it has made it possible for researchers in my research group to obtain hands-on knowledge with this new and complex field of research, rather than leaving bioinformatics to externals.

**Pedro J. Torres**
**Ph.D. Candidate**
**Department of Biology, San Diego State University, San Diego, California**
Mr. Torres is a co-author of this paper.

The ability to add on plugins as needed and seeing the transition of QIIME from a platform for amplicon based (specifically 16S rRNA) to multiomics analysis is very exciting and needed in a time when using different omics technologies in a single study is starting to become the norm.

One of the greatest issues I have seen time and time again in analysis of next generation sequencing (NGS) data is, as with many other things, REPRODUCIBILITY! Often when analyzing large amounts of data, good notes for reproducibility later is not on the top of everyone's list. QIIME 2's automated data provenance allows for tracking of not only the parameters and versions used to generate your data, but also the machine used allows for transparent microbiome research without the need to even think about it. This is a big step forward in allowing for data transparency and reproducibility.

I have also used many different bioinformatics software tools and it can be quite frustrating learning about a new tool, wanting to use a new tool only to find out that there are issues that keep piling up and the authors do not respond with help. The QIIME community is amazing! Any issue that I have had has been answered in an appropriate time without the sarcasm or bullying you can encounter in other community forums. QIIME 2 has really done a great job at creating not only a tool but an online community resource well suited for the novice and experienced data scientist alike unlike any I have ever seen.

Some of the interesting new plugins that are helping our lab advance our research is q2-longitudinal to look at factors influencing alpha and beta diversity changes over time and q2-metabolomics to import our metabolomics data into QIIME 2 for analysis.

**Jonathan Warren**
**National Laboratory Service, Environment Agency, Starcross, UK**
Mr. Warren is a co-author of this paper.

I started using QIIME 2 as an introduction to bioinformatics with no previous experience. It made learning the basics of manipulating my DNA data much easier, and allowed me to try out the packaged tools much easier than if they were to be used standalone. I like the fact QIIME 2 is open source as I believe that it is better for the community and makes using DNA sequence data much easier for those without a computer science background, as it can be tried for free with no cost to the user.

I contributed to QIIME 2 as I wanted to just import a lot of fastq files at once and I knew they were already in a very specific format, and so I started writing some code in order to import using a whole folder at once. Once I did this and tested it, I shared my code for all to use.

# References

1. Koskinen, K. et al. mBio **8** (6), e00824-17 (2017). doi: 10.1128/mBio.00824-17
2. Schwendner, P. et al. Microbiome **5** (1), 129. doi:10.1186/s40168-017-0345-8
3. Mahnert, A. et al. Front. Microbiol. (2018). doi: 10.3389/fmicb.2018.02985
4. Caporaso, J.G. et al. Nature Methods (2010). Doi: 10.1038/nmeth.f.303
5. Bai, M. et al. Microb. Ecol. (2018). doi.:10.1007/s00248-018-1235-8.
6. Xie, N. et al. Env. Microbiol. 20(8), 3042-3056 (2018).
7. Tangedal et al., Respir. Res. 18, 164 (2017).