Published in final edited form as:
Int J Comput Biol Drug Des. 2014 ; 7(0): 130–145. doi:10.1504/IJCBDD.2014.061652.

BOTUX: Bayesian-like operational taxonomic unit examiner

Vishal N. Koparde,
Center for Study of Biological Complexity, Virginia Commonwealth University, Richmond, VA 23284, USA

Ricky S. Adkins,
Center for Study of Biological Complexity, Virginia Commonwealth University, Richmond, VA 23284, USA

Jennifer M. Fettweis,
Center for Study of Biological Complexity and Department of Microbiology and Immunology, Virginia Commonwealth University, Richmond, VA 23284, USA

Myrna G. Serrano,
Center for Study of Biological Complexity and Department of Microbiology and Immunology, Virginia Commonwealth University, Richmond, VA 23284, USA

Gregory A. Buck,
Center for Study of Biological Complexity and Department of Microbiology and Immunology, Virginia Commonwealth University, Richmond, VA 23284, USA

Mark A. Reimers, and
Center for Study of Biological Complexity and Virginia Institute for Psychiatric and Behavioral Genetics, Virginia Commonwealth University, Richmond, VA 23298, USA

Nihar U. Sheth
Center for Study of Biological Complexity, Virginia Commonwealth University, Richmond, VA 23284, USA

Vishal N. Koparde: vnkoparde@vcu.edu; Ricky S. Adkins: shaun.adkins@gmail.com; Jennifer M. Fettweis: fettweisjm@vcu.edu; Myrna G. Serrano: mgserrano@vcu.edu; Gregory A. Buck: gabuck@vcu.edu; Mark A. Reimers: mreimers@vcu.edu; Nihar U. Sheth: nsheth@vcu.edu

Abstract

Bayesian-like operational taxonomic unit examiner (BOTUX) is a new tool for the classification of 16S rRNA gene sequences into operational taxonomic units (OTUs) that addresses the problem of overestimation caused by errors introduced during PCR amplification and DNA sequencing steps. BOTUX utilises a grammar-based assignment strategy, where Bayesian models are built from each word of a given length (e.g., 8-mers). *de novo* analysis is possible with BOTUX as it does not require a training set, and updates probabilistic models as new sequences are recruited to an OTU. In benchmarking tests performed with real and simulated datasets of 16S rDNA

Correspondence to: Mark A. Reimers, `mreimers@vcu.edu`; Nihar U. Sheth, `nsheth@vcu.edu`.

This paper is a revised and expanded version of a paper entitled ‘BOTUX: Bayesian-like Operational Taxonomic Unit eXaminer’ presented at the *International Conference on Intelligent Biology and Medicine (ICIBM 2013)*, Nashville, TN, USA, 11–13 August, 2013.

sequences, BOTUX accurately identifies OTUs with comparable or better clustering efficiency and lower execution times than other OTU algorithms tested. BOTUX is the only OTU classifier, which allows incremental analysis of large datasets, and is also adept in clustering both 454 and Illumina datasets in a reasonable timeframe.

Keywords

OTU; operational taxonomic unit; Bayesian; 16S; rRNA; sequence clustering; 454; Illumina

1 Background

Large-scale metagenomic assays of 16S ribosomal RNA genes, in which hyper-variable regions of the 16S rRNA genes are used to identify and quantify individual microbes in complex communities, have become common in recent years. Many metagenomic sequence classification methods that proceed by aligning sequences to a reference database (Mitra et al., 2011; Wang et al., 2007) often fail to identify novel organisms because the reference 16S rDNA databases are incomplete. As an alternative, *de novo* sequence clustering methods, such as UCLUST (Edgar, 2010) and other methods implemented by Mothur (Schloss et al., 2009) and QIIME (Caporaso et al., 2010), have been developed to classify 16S rDNA sequences into operational taxonomic units (OTUs). However, commonly used OTU-based algorithms that use pairwise and heuristic alignment algorithms often overestimate diversity due to errors introduced in the polymerase chain reaction (PCR) amplification and DNA sequencing steps (Huse et al., 2010). A number of solutions have recently been suggested to reduce OTU overestimation; these include the development of new chimera checking programs (Edgar et al., 2011; Wright et al., 2012), denoising tools (Quince et al., 2009, 2011; Reeder and Knight, 2010), and protocols for prefiltering sequences (Schloss et al., 2011). Other groups have devised new OTU assignment algorithms such as AbundantOTU (Ye, 2011) and GramCluster (Russell et al., 2010). These two algorithms use different approaches for OTU assignment. AbundantOTU infers consensus sequences and clusters sequencing reads that align to the consensus sequences, whereas GramCluster relies on a grammar-based distance metric to cluster sequences into OTUs.

Here, we present the Bayesian-like operational taxonomic unit examiner (BOTUX), a new OTU assignment method that performs clustering at the same or better precision than several other OTU algorithms (i.e., Mothur (Schloss et al., 2009), UCLUST (Edgar, 2010), AbundantOTU (Ye, 2011), and GramCluster (Russell et al., 2010)). Our algorithm is a naïve Bayesian-like classifier, where each of the attributes of a given class is considered to each contribute independently to the probability of class membership (Domingos and Pazzani, 1997). Bayesian methodology has been successfully utilised in other classification programs, such as the RDP Classifier (Wang et al., 2007), and some of the methodology used for BOTUX is based on the RDP algorithm. BOTUX conceptually differs from the RDP Classifier as it permits *de novo* clustering and probability models are updated as new sequences are recruited to an OTU. Moreover, BOTUX uses a different scoring approach for OTU assignment.

2 Implementation

2.1 BOTUX algorithm

After reading in the sequences, BOTUX sorts them starting from the longest to the shortest sequence. All sequences are the trimmed to a maximum of the *n*th percentile read length (default: *n* = 75). Duplicate sequences are collapsed into the same sequence with appropriate frequency. This results in significant savings in execution time, if the duplication levels are very high in the input sequences. The sequence string is then broken down into eight-base long subsequences or words. A frequency count of each possible 8-mer word is maintained relative to each sequence. It should be noted that the default word size of 8, which can be edited by the user at runtime, is shown to be the most accurate with the least memory requirements (Wang et al., 2007). The first sequence then becomes the first OTU, provided no OTU model is loaded in. An OTU can be considered as a word-bank, with the 8-mer words coming from the sequences it contains and their respective frequencies. The sequence identifier of each sequence assigned to an OTU is also stored to print detailed read-by-read OTU assignments after successful completion of BOTUX. The algorithm is represented as a flowchart in Figure 1. Each subsequent sequence is compared against all the existing OTUs, and the sequence is then either:

- assigned to an existing OTU if set conditions are met, or
- used as the seed for a new OTU.

Words from the query sequence are compared to collective word banks for each existing OTU. An overall similarity score, that is analogous to a Bayesian posterior probability, is calculated as described below.

For each word from the query that occurs in the word bank of the target OTU, the proportion of occurrences of that word in that OTU word bank is added to the similarity score. Query words that do not match the target OTU's word bank do not contribute to the score.

$$S(Q \in O) = \frac{L_{seed}}{L_{query}} \times \sum S(W_Q \in O) \quad (1)$$

Equation (1) is used to calculate the score for a query sequence against a particular OTU, where S is the similarity score, Q is the current query, O is the current OTU, W_Q is a word from the query, L_{seed} is the length of the OTU's seed sequence, and L_{query} is the length of the query sequence. This formula was adapted from Bayes' Formula in probability theory and includes an adjustment factor to account for the length of the query sequence.

$$S(W_Q \in O) = \left[\frac{n_{(W_Q \in Q)} \times n_{(W_Q \in O)}}{\sum_{W_i \in O} n_{W_i}} \right] \quad (2)$$

Equation (2) shows the formula for calculating the score for a word from a given query that is present in the word bank of a particular OTU, where $n_{(W_Q \in Q)}$ represents the frequency of the word in the query, $n_{(W_Q \in Q)}$ represents the frequency of the word in the OTU's word bank, and n_{W_i} represents the frequency of a word W_i in the OTU's word bank. The query sequence is assigned to the OTU with the highest similarity score if the score exceeds the threshold score (default 0.75). Otherwise, the query is used to create a new OTU. The user has the option to decrease or increase the threshold score if more conservative or aggressive clustering is desired, respectively. Upon assignment of a query sequence to the OTU, the word bank frequencies of the OTU are updated and sequence ID information stored to the OTU. If the newly assigned query sequence has words not present in the OTU's word bank, then these words are appended with appropriate frequencies. After all query sequences are exhausted, read-by-read OTU assignments, overall OTU profile, and the existing OTU model are saved in different output files.

BOTUX saves the OTU model upon completion, which can be preloaded before running the next set of sequencing reads. This is referred to as 'incremental' mode of BOTUX. We believe that BOTUX is the only OTU classifier, which permits clustering in an incremental mode. Incremental analysis is very useful for sequence classification in studies that are periodically updated with new sequences, as most on-going studies are. The incremental mode can also be applied to clustering datasets using existing models, in a manner similar to how the RDP Classifier works with previously trained models. BOTUX, however, allows the creation of new OTUs in the incremental mode, thus new OTUs are added as sequences from new taxa are encountered.

In case of paired-end FASTQ files from an Illumina sequencing run, the concatenated read1 and read2 sequence is used for duplicate identification. The query sequences and OTUs each have two distinct word-banks corresponding to read1 and read2. For each query paired-end sequence to OTU comparison, two scores, corresponding to read1 and read2, are calculated and both need to exceed the threshold score for the best scoring OTU to claim assignment of the read.

2.2 BOTUX code

BOTUX is written using Python (version 2.7.2). Apart from the standard python modules, BOTUX utilises the HTSeq (https://pypi.python.org/pypi/HTSeq) module for reading in query sequences from different file formats. The program accepts a standard FASTA file from a 454 run or a standard FASTQ file as single-end Illumina data or a pair of FASTQ files as in paired-end Illumina data. Along with the input file, BOTUX also accepts a project name, which is used for naming output files. The output of successful BOTUX run contains three files

- read-by-read OTU assignments text file
- OTU profile text file
- OTU model file, which is a binary file generated using Python's pickle module.

A detailed manual is available together with the Python scripts at github.com/nisheth/BOTUX.

2.3 OTU algorithms for testing and validation

BOTUX, Mothur (Schloss et al., 2009) (version 1.8), UCLUST (Edgar, 2010) (version 6.0.301), GramCluster (Russell et al., 2010) (version 1.3), and AbundantOTU (Ye, 2011) (version 2.0) were used in testing and validation of different sample datasets. An identity distance threshold of 0.97 was used while running UCLUST and Mothur. Default parameters were selected for BOTUX, GramCluster, and AbundantOTU. In all cases, the query sequences were not subjected to any additional pre-processing steps before clustering into OTUs.

2.4 Real and simulated datasets

Four mock 16S rDNA sequence datasets of known composition were used in validation studies: the Priest (Quince et al., 2009) dataset of 38,351 454 GSFLX sequences of V5/V6 region amplicons from a mix of 23 divergent clones from an environmental 16S rDNA clone library obtained from a eutrophic lake (Short Read Archive: SRX002564); the Sue (Huse et al., 2007) dataset of 99,189 454 GS20 sequences of the V6 region from 43 divergent clones obtained from diffuse flow hypothermal vents; and the HMP Mock 1 (A-2) and HMP Mock 2 (B-18) datasets, which both contain V1-V3 region 16S rDNA reads derived from the same HMP (http://www.hmpdacc.org/) mock community sample containing 22 bacterial species. The Priest dataset had an average read length of 266 bp, while the Sue dataset was 97 bp. Vaginal Microbiome Consortium members at VCU performed genomic DNA extraction, PCR amplification and 454 FLX DNA sequencing independently for the two HMP Mock sample replicates according to the protocols of the Vaginal Human Microbiome Project (Buck et al., 2010). Selection of organisms and the preparation of the HMP mock community sample are to be described as part of a HMP consortium manuscript. The HMP Mock sample datasets contained 12,592 and 47,073 reads with average read lengths of 317 bp and 315 bp, respectively.

For further extensive validation of BOTUX against all other clustering software, we utilised the 10 MID-barcoded 16S rDNA bacterial samples with normal read length distribution, power law rank-abundance and richness, and varied β-diversity simulated using Grinder (Angly et al., 2012). It has been shown that Grinder is capable of generating realistic amplicon libraries and modelling the effect of 454 homopolymer errors on 16S microbial community profiling. All of the datasets used herein consist of 5000 simulated amplicon reads with simulated pyrosequencing errors and are freely available with the original Grinder publication (Angly et al., 2012).

To validate the utility of BOTUX in clustering Illumina reads, we generated three datasets each for 2 × 100, 2 × 150, 2 × 200 and 2 × 250 read length configurations, with an estimated 25, 40, and 50 OTUs, respectively. The curated dataset of the V1-V3 regions of 16S genes from 973 bacteria, which is distributed as part of STIRRUPS (Fettweis et al., 2012), was used as a reference for generating these simulated FASTQ files. Each dataset contained 100,000 reads and random mutation rate of 1% was used to simulate sequencing error (Claesson et al., 2010; Schloss et al., 2011).

2.5 Clustering similarity metrics

Rand indices, Jaccard coefficients (Halkidi et al., 2001), and sequence differentiation were calculated according to equations (3), (4) and (5) respectively. Rand index and Jaccard coefficient have a value ranging from 0 to 1 and are a quantitative measure of the accuracy the OTU clustering algorithms. The sequence differentiation fraction also ranges from 0 to 1 and verifies if the algorithm correctly separates distinct OTUs. We also calculated the cluster dilution, which is the ratio of the expected OTUs to the predicted OTUs, and the percentage of query reads that could not be clustered. The reads belonging to OTUs containing a total of four or less reads are termed as ‘unclustered’. This is a measure of the stringency of the clustering method.

$$Rand\ Index=\frac{ss+dd}{ss+sd+ds+dd} \quad (3)$$

$$Jaccard\ Coefficient=\frac{ss}{ss+sd+ds} \quad (4)$$

$$Sequence\ Differentiation=\frac{dd}{ds+dd} \quad (5)$$

Here, *ss* represents the number of sequence pairs derived from the same species that are placed in the same OTU; *sd* is the number of sequence pairs, which are derived from the same species, but classified in different OTUs; *ds* is the number of sequence pairs, which are not derived from the same species, but belong to the same OTU; while *dd* accounts for the number of sequence pairs, which are truly derived from different species and are classified into different OTUs. It should be noted that all possible sequence pairs are considered during these calculations. In addition to these metrics, for cases where we have prior knowledge of each read’s expected OTU assignment and the sample profile, i.e., species-level percent compositions, we also calculated the percent of incorrectly assigned reads, and the Pearson and Spearman correlation coefficients in order to examine the OTU clustering accuracy.

3 Results and discussion

3.1 Mock datasets

Four sequence datasets from Priest, Sue, and HMP DACC (A-2 and B-18) with prior knowledge of the number of distinct species present were classified using AbundantOTU (Ye, 2011), BOTUX, GramCluster (Russell et al., 2010), Mothur (Schloss et al., 2009) and UCLUST (Edgar, 2010). We measured the number of OTUs predicted by each program as the number of OTUs, which contained five or more query sequences assigned to them. All other reads were considered ‘unclustered’. Table 1 shows the clustering performances of all the clustering methods based on the number of OTUs predicted, percentage of unclustered reads, and the wall-time required for execution. It should be noted that all programs were run in a single processor mode even though parallelisation is possible in some cases. Mothur and UCLUST largely overestimated the number of OTUs in all cases, while GramCluster largely overestimated for one sample. AbundantOTU and BOTUX seemed to best predict the number of OTUs for all the samples, but AbundantOTU always has a significantly larger number of unclustered reads than BOTUX. It appears that UCLUST, which performs the fastest, has 3–16% of unclustered reads, while for all other programs this percentage is much smaller. The Sue dataset, which contains a high percentage of duplicate sequences (~95%), runs much faster with BOTUX than with AbundantOTU or GramCluster. This is because, like Mothur, BOTUX decreases the number of sequences to be clustered into OTUs by considering sequence duplication prior to OTU assignments.

3.2 Simulated realistic datasets with high diversity

Ten realistic metagenomic datasets freely available with the Grinder publication (Angly et al., 2012) are clustered into OTUs using AbundantOTU, BOTUX, GramCluster, Mothur, and UCLUST. Each of these samples contained simulated amplicon 5000 reads with simulated pyrosequencing errors. These reads are split into three files with 2000, 2000 and 1000 reads, respectively, and the OTU clustering was repeated using BOTUX in an incremental mode (BOTUX_incr), i.e., the OTU model from the first 2000 reads is preloaded for the next 2000 reads, which is then used for the last 1000 reads. This demonstrates BOTUX's ability of performing incremental analyses, which is its major differentiating feature when compared against other methods. Figure 2 indicates the results of OTU clustering quantified using various cluster similarity metrics. A Rand index and Jaccard coefficient value significantly less than 1 for GramCluster and UCLUST, indicates more aggressive clustering performance with default parameters. For all of the algorithms, the percent of unclustered reads are between 0.4–0.6%.

In all of these datasets, we have a prior knowledge about which species was used as a reference to generate each read in the dataset, and thus we can calculate the percent of reads that have been assigned to incorrect species or 'misassigned' by the clustering algorithm. With the exception of GramCluster, it can be seen from Figure 3, that all the clustering algorithms have low (<~5%) number of 'misassigned' reads. Figure 3 also shows the Pearson correlation coefficient and the Spearman rank correlation coefficient depicting the accuracy of the OTU clustering by each program. The Pearson coefficient shows the correlation between the expected and predicted percentage compositions of each species in the sample and a value closer to 1 suggests a strong positive correlation. The Spearman coefficient indicates the correlation between the expected and predicted relative abundance ranking of the species in the sample, and a value closer to 1 suggests more accurate predictions. The Pearson coefficient is found to be close to 1 for all programs except UCLUST. This is because almost all programs accurately predict the percentage compositions of high abundance species in each sample, and those of low abundance species are too insignificant to affect its overall value. However, inaccurate ranking of low abundance species in the samples will lower the value of the Spearman coefficient, as seen for AbundantOTU, GramCluster, and UCLUST. It should be noted that the performances of BOTUX and BOTUX_incr are almost identical indicating successful implementation of the incremental mode. Running BOTUX in incremental mode for real incremental samples can result in significant savings in computational resources.

3.3 Simulated illumina datasets

Twelve 100,000 read datasets, three datasets each for four different read lengths, were simulated to illustrate the performance of BOTUX with paired-end Illumina data. Figure 4 shows the various metrics used to evaluate the performance of the algorithm. All of the metrics seem to be within acceptable limits for all the samples considered and accuracy seems to be improving with read length. Sequence differentiation equal to 1, no unclustered reads, and cluster dilution of 1 suggests that BOTUX performs best at a read length of 2 × 250. The execution takes between 265 to 840 seconds as shown in Figure 5. As expected, the longest sequence dataset with the most number of expected OTUs takes the most time to execute.

4 Conclusions

Overall, we found the clustering efficiency of BOTUX is comparable to or better than the other clustering tools tested, including tools that have been widely adopted, as well as more recently-developed algorithms (i.e., AbundantOTU and GramCluster) where the OTU assignment step was directly targeted to address the issue of overestimation. Furthermore, BOTUX has the option of being run in incremental mode resulting in considerable savings in time and computational resources when applied to large datasets of the same samples coming from different sequencing runs. To our knowledge, BOTUX is the only OTU clustering tool with this feature. It is clear from our tests that running BOTUX in incremental mode does not adversely affect the results when compared to a *de novo* approach.

Although not the fastest OTU algorithm, BOTUX in most cases utilises less memory than programs that do exhaustive comparisons, and is significantly faster than those using more time-consuming pairwise alignment. Elimination of duplicate sequences prior to clustering also results in significant reductions in runtime and memory usage for high duplication datasets. While both GramCluster and BOTUX proceed by breaking the sequence into words, GramCluster relies more on the size of the dictionaries, whereas BOTUX matches individual words, discriminating words that are more conserved from others less so within a species.

BOTUX has been extended to single-end and paired-end Illumina datasets with considerable accuracy and efficiency. It performs acceptably at read lengths ranging from 100 bp to 250 bp, with accuracy improving with increasing read length. Finally, although BOTUX has been tested here using 16S rDNA genes as targets, it can easily be extended to other gene sets

Acknowledgments

Partial funding for Ricky S. Adkins, Jennifer M. Fettweis, Myrna G. Serrano, Gregory A. Buck and Mark A. Reimers came from NIH grant 4UH3A1083263-02.

References

Angly FE, Willner D, Rohwer F, Hugenholtz P, Tyson GW. Grinder: a versatile amplicon and shotgun sequence simulator. Nucleic Acids Res. 2012; 40:e94. [PubMed: 22434876]

Fettweis JM, Alves JP, Borzelleca JF, Brooks JP, Friedline CJ, Gao Y, Gao X, Girerd P, Harwich MD, Hendricks SL, Jefferson KK, Lee V, Mo H, Neale MC, Puma FA, Reimers MA, Rivera MC, Roberts SB, Serrano MG, Sheth NU, Silbert JL, Voegtly L, Prom-Wormley EC, Xie B, York TP, Cornelissen C, Strauss JL, Eaves LJ, Buck GA. The vaginal microbiome: disease, genetics and the environment. Nat Preced. 2010

Caporaso JG, Kuczynski J, Stombaugh J, Bittinger K, Bushman FD, Costello EK, Fierer N, Pena AG, Goodrich JK, Gordon JI, Huttley GA, Kelley ST, Knights D, Koenig JE, Ley RE, Lozupone CA, McDonald D, Muegge BD, Pirrung M, Reeder J, Sevinsky JR, Turnbaugh PJ, Walters WA, Widmann J, Tatsunenko T, Zaneveld J, Knight R. QIIME allows analysis of high-throughput community sequencing data. Nat Methods. 2010; 7:335–336. [PubMed: 20383131]

Claesson MJ, Wang Q, O'Sullivan O, Greene-Diniz R, Cole JR, Ross RP, O'Toole PW. Comparison of two next-generation sequencing technologies for resolving highly complex microbiota composition using tandem variable 16S rRNA gene regions. Nucleic Acids Res. 2010; 38:e200. [PubMed: 20880993]

Domingos P, Pazzani M. On the optimality of the simple Bayesian classifier under zero-one loss. Mach Learn. 1997; 29:103–130.

Edgar RC. Search and clustering orders of magnitude faster than BLAST. Bioinformatics. 2010; 26:2460–2461. [PubMed: 20709691]

Edgar RC, Haas BJ, Clemente JC, Quince C, Knight R. UCHIME improves sensitivity and speed of chimera detection. Bioinformatics. 2011; 27:2194–2200. [PubMed: 21700674]

Fettweis JM, Serrano MG, Sheth NU, Mayer CM, Glascock AL, Brooks JP, Jefferson KK, Buck GA. Species-level classification of the vaginal microbiome. BMC Genomics. 2012; 13:S17. [PubMed: 23282177]

Halkidi M, Batistakis Y, Vazirgiannis M. On clustering validation techniques. J Intell Inf Syst. 2001; 17:107–145.

Huse SM, Huber JA, Morrison HG, Sogin ML, Welch DM. Accuracy and quality of massively parallel DNA pyrosequencing. Genome Biol. 2007; 8:R143. [PubMed: 17659080]

Huse SM, Welch DM, Morrison HG, Sogin ML. Ironing out the wrinkles in the rare biosphere through improved OTU clustering. Environ Microbiol. 2010; 12:1889–1898. [PubMed: 20236171]

Mitra S, Stärk M, Huson DH. Analysis of 16S rRNA environmental sequences using MEGAN. BMC Genomics. 2011; 12:S17. [PubMed: 22369513]

Quince C, Lanzén A, Curtis TP, Davenport RJ, Hall N, Head IM, Read LF, Sloan WT. Accurate determination of microbial diversity from 454 pyrosequencing data. Nat Methods. 2009; 6:639–641. [PubMed: 19668203]

Quince C, Lanzen A, Davenport RJ, Turnbaugh PJ. Removing noise from pyrosequenced amplicons. BMC Bioinformatics. 2011; 12:38. [PubMed: 21276213]

Reeder J, Knight R. Rapidly denoising pyrosequencing amplicon reads by exploiting rank-abundance distributions. Nat Methods. 2010; 7:668–669. [PubMed: 20805793]

Russell DJ, Way SF, Benson AK, Sayood K. A grammar-based distance metric enables fast and accurate clustering of large sets of 16S sequences. BMC Bioinformatics. 2010; 11:601. [PubMed: 21167044]

Schloss PD, Gevers D, Westcott SL. Reducing the effects of PCR amplification and sequencing artifacts on 16S rRNA-based studies. PLoS ONE. 2011; 6:e27310. [PubMed: 22194782]

Schloss PD, Westcott SL, Ryabin T, Hall JR, Hartmann M, Hollister EB, Lesniewski RA, Oakley BB, Parks DH, Robinson CJ, Sahl JW, Stres B, Thallinger GG, Van Horn DJ, Weber CF. Introducing Mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. Appl Environ Microbiol. 2009; 75:7537–7541. [PubMed: 19801464]

Wang Q, Garrity GM, Tiedje JM, Cole JR. Naïve Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. Appl Environ Microbiol. 2007; 73:5261–5267. [PubMed: 17586664]

Wright ES, Yilmaz LS, Noguera DR. DECIPHER, a search-based approach to chimera identification for 16S rRNA sequences. Appl Environ Microbiol. 2012; 78:717–725. [PubMed: 22101057]

Ye Y. Identification and quantification of abundant species from pyrosequences of 16S rRNA by consensus alignment. Proc IEEE Int Conf Bioinforma Biomed. 2011; 2010:153–157.

Biographies

Vishal N. Koparde received his PhD from Vanderbilt University in 2006 and is currently working in conjunction with the Nucleic Acid Research Facilities, Bioinformatics

Computational Core Laboratories and the Center for Study of Biological Complexity at the Virginia Commonwealth University, Richmond, VA. His research includes developing novel approaches and algorithms to analyse and interpret next generation DNA sequencing data generated on various platforms.

Ricky S. Adkins received his MBin at Virginia Commonwealth University in 2011. Currently he works as a bioinformatics software engineer in the Institute for Genome Sciences at the University of Maryland School of Medicine. Over the two years he has worked at IGS, he has been responsible for development and maintenance of the prokaryotic annotation pipeline, provides user support for the Ergatis pipeline management framework, and develops for the Cloud Virtual Resource (CloVR).

Jennifer M. Fettweis received her PhD in Microbiology and Immunology from Virginia Commonwealth University and her BA in Mathematics and Economics from the University of Virginia. She is currently the Project Director for the Vaginal Microbiome Consortium at VCU, a Fellow of the Center for the Study of Biological Complexity and a Postdoctoral Research Fellow in the Microbiology and Immunology Department at VCU. Her current research interests include vaginal microbiome studies, women's health, next-generation DNA sequencing, bioinformatics, genomics, host-microbiome interactions and the characterisation of novel species.

Myrna G. Serrano received his PhD in Biology of the Host/Pathogen Relationship from the University of Sao Paulo in Sao Paulo, Brazil in 2000 and performed postdoctoral research on molecular parasitology at the Virginia Commonwealth University. She is currently an Assistant Professor of Center for the Study of Biological Complexity at Virginia Commonwealth University. She manages the Genomics and Microarray Cores of the Nucleic Acids Research Facilities. Her current research involves next-generation sequencing analysis, comparative genomics of kinetoplastid protozoa and the human microbiome.

Gregory A. Buck received his PhD in Microbiology and Immunology from the University of Washington in Seattle and performed postdoctoral research on molecular parasitology at the Institut Pasteur in Paris. He is currently a Professor of Microbiology and Immunology and Director of the Center for the Study of Biological Complexity at Virginia Commonwealth University. He directs the Nucleic Acids Research Facilities, which includes next generation sequencing sequencing core, and oversees the Bioinformatics Computational Core Laboratories and the Center for High Performance Computing at VCU. His current research involves studies of the genomics of kinetoplastid protozoa and the human microbiome.

Mark A. Reimers obtained his PhD in Mathematics from the University of British Columbia in Canada, and he is now at the Virginia Institute for Psychiatric and Behavioral Genetics in Richmond. His research work focuses on analysing and interpreting the very large datasets now coming out of neuroscience and genomics.

Nihar U. Sheth received his MS in Bioinformatics from Indiana University in Bloomington in 2005. He is currently the Technical Director of Bioinformatics Computational Core Laboratories (BCCL) and an instructor of Bioinformatics at Virginia Commonwealth

University since 2006. His research interest includes Bioinformatics and Computational Biology approaches to study human microbiome, next generation sequence analysis, comparative genomes and whole exome studies to understand genetic variation among cancer diseases.

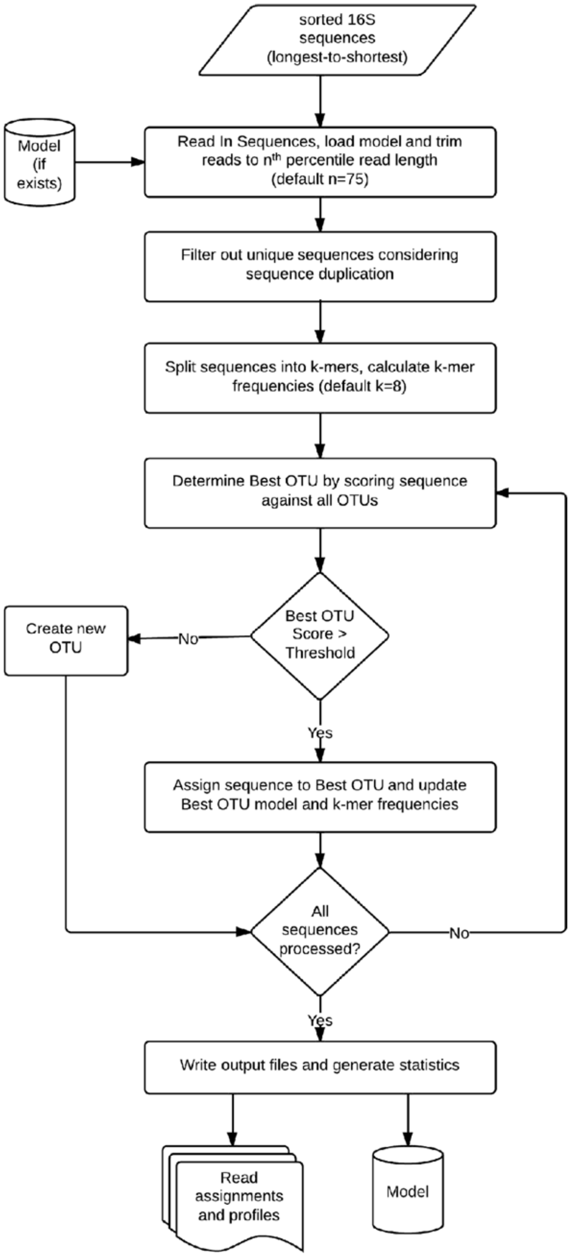
sorted 16S sequences (longest-to-shortest)
Model (if exists)
Read In Sequences, load model and trim reads to nth percentile read length (default n=75)
Filter out unique sequences considering sequence duplication
Split sequences into k-mers, calculate k-mer frequencies (default k=8)
Determine Best OTU by scoring sequence against all OTUs
Best OTU Score > Threshold
No
Create new OTU
Yes
Assign sequence to Best OTU and update Best OTU model and k-mer frequencies
All sequences processed?
No
Yes
Write output files and generate statistics
Read assignments and profiles
Model

Figure 1.

BOTUX algorithm flowchart. This shows the various steps involved in classifying sequences into operational taxonomical units (OTUs) using BOTUX

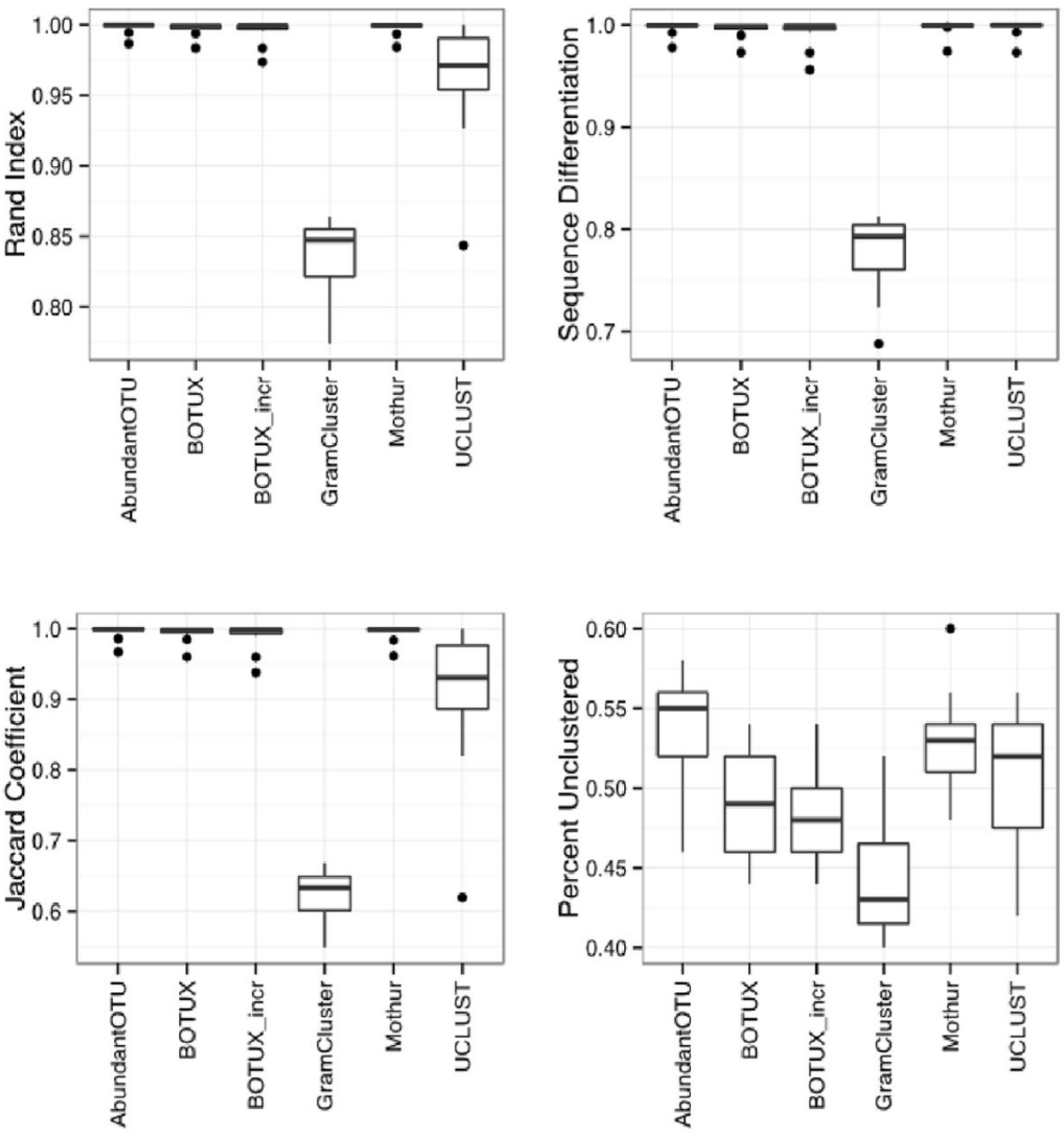

Figure 2.

Various clustering similarity metrics comparing AbundantOTU, BOTUX, GramCluster, Mothur and UCLUST for Grinder simulated datasets. Sequence clustering similarity is quantified using Rand index, Jaccard coefficient and sequence differentiation fraction. BOTUX_incr represents execution of BOTUX in incremental mode

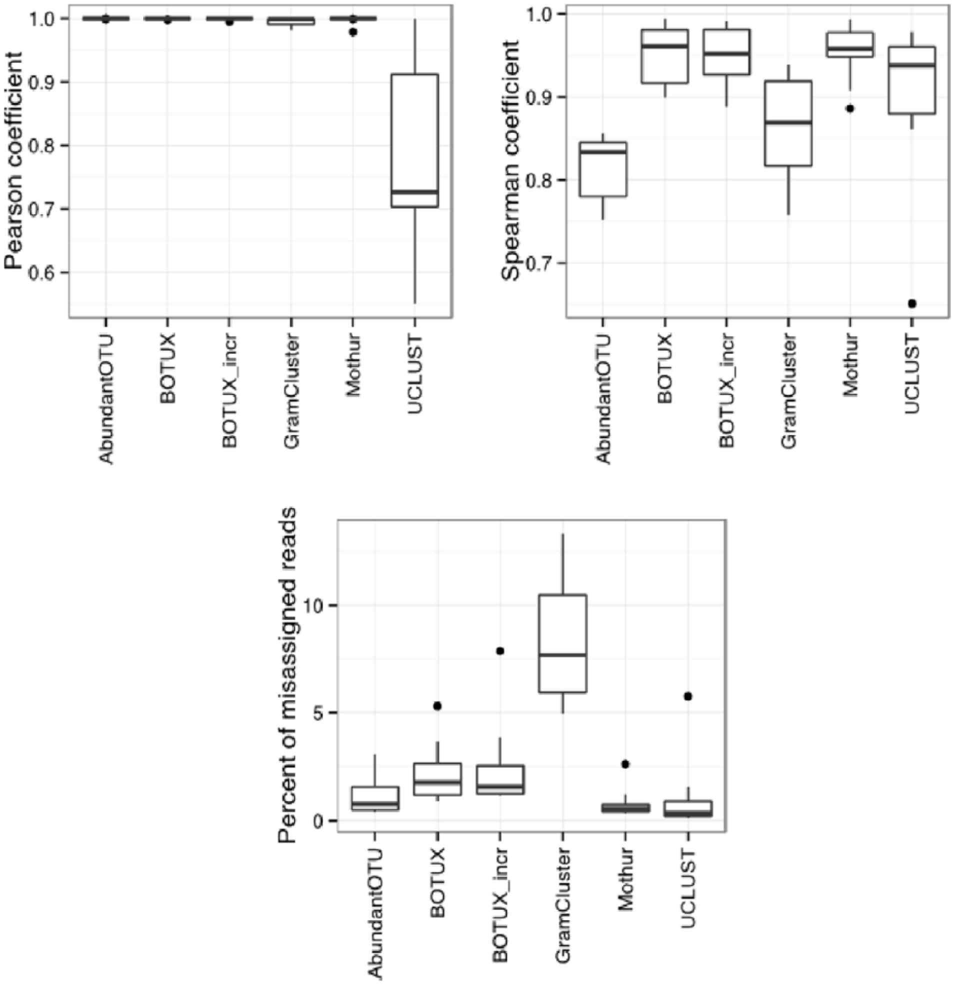

Figure 3.

OTU clustering accuracy for Grinder simulated datasets using AbundantOTU, GramCluster, Mothur, BOTUX and UCLUST. Pearson correlation coefficient, Spearman rank correlation coefficient and Percent of misassigned reads for various clustering algorithms on the Grinder simulated datasets. BOTUX_incr represents execution of BOTUX in incremental mode

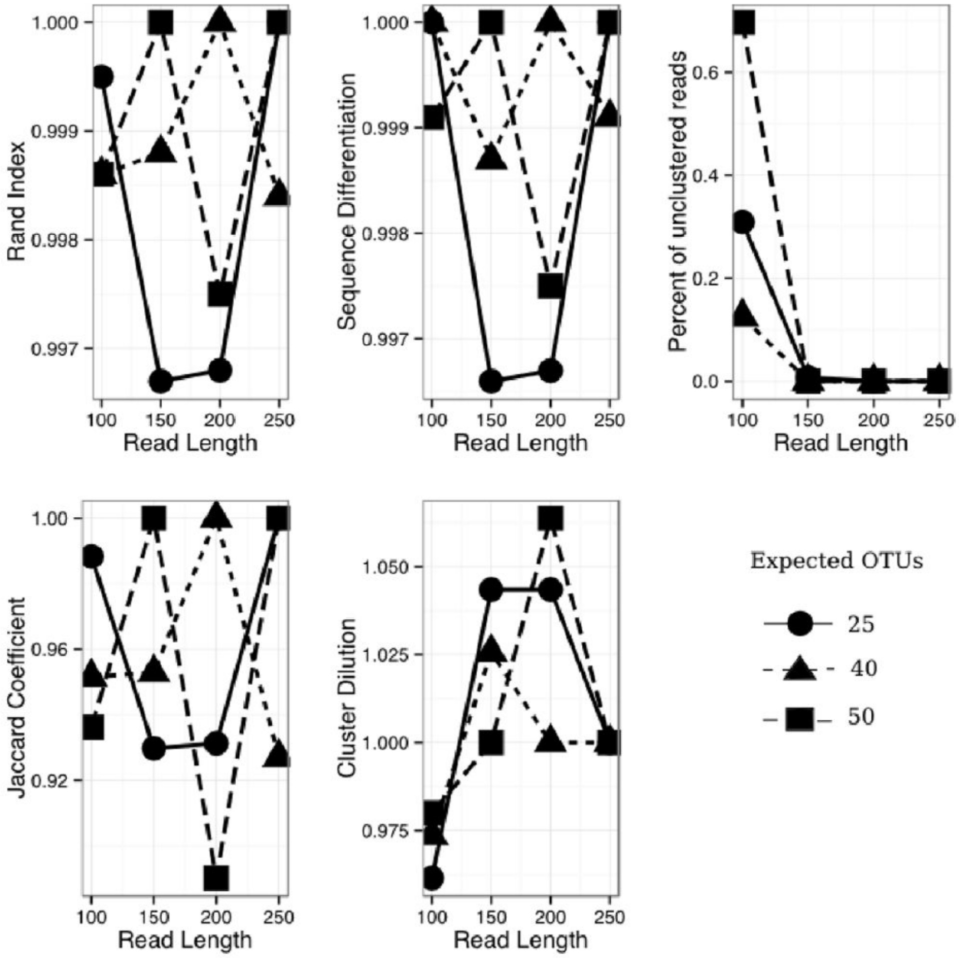

Figure 4.

Clustering similarity metrics for simulated paired-end Illumina data for various read lengths. Note that 1% of the bases are randomly mutated to simulate sequencing error

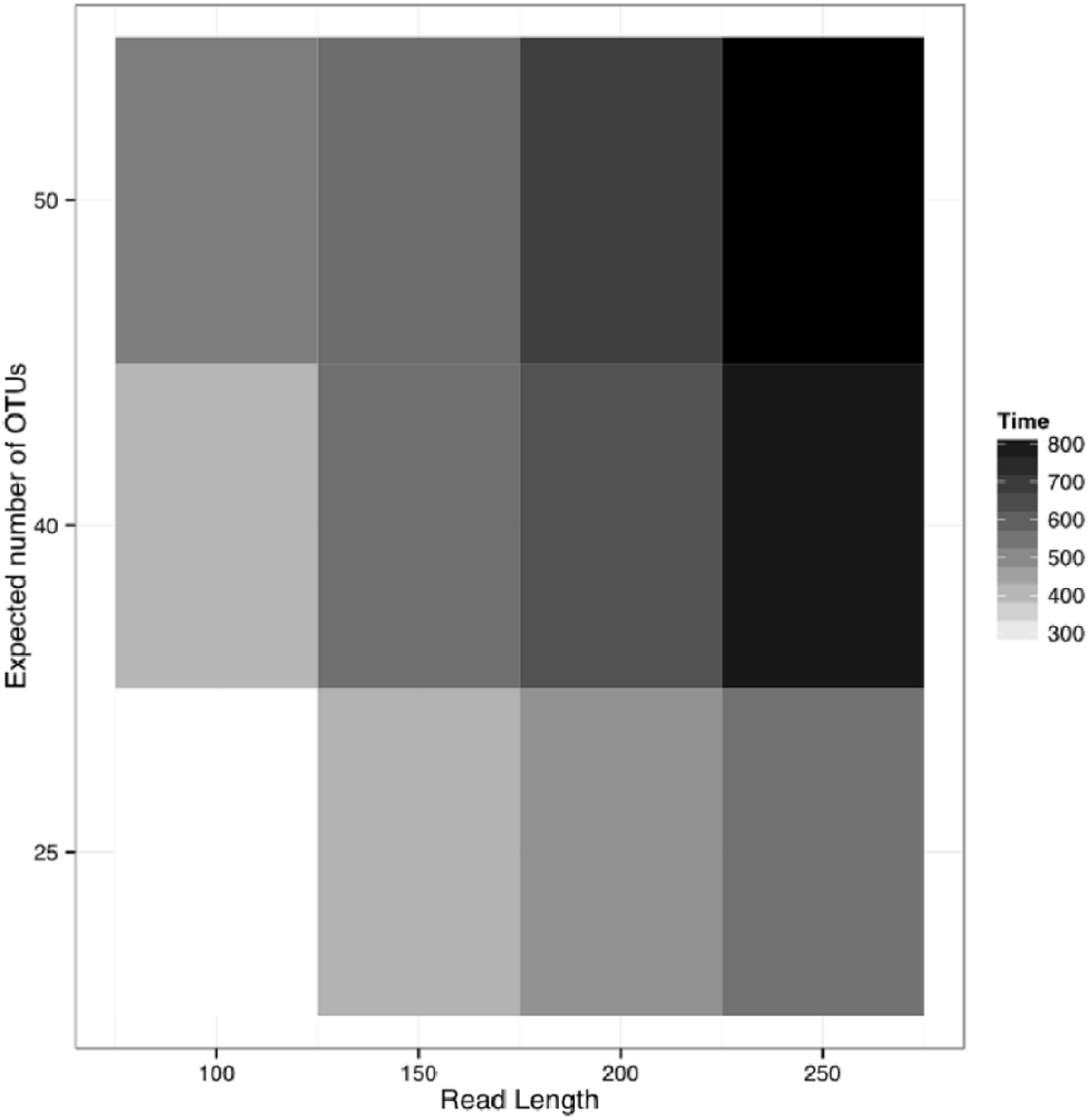

Figure 5.

BOTUX execution times in seconds for simulated Illumina data. The read length refers to the length of one of the reads in a paired-end read pair from a simulated Illumina dataset containing 100000 pairs. As expected, the longest read length with highest expected number of OTUs takes longest to cluster

Table 1

Comparison of clustering algorithms performance using mock datasets. Four mock datasets where clustered using AbundantOTU, BOTUX, GramCluster, Mothur and UCLUST. They are compared based on number of OTUs predicted, percent of unclustered reads and wall-time for execution. A-2 and B-18 are datasets from HMP DACC with average read lengths of 316 and 314, respectively. Priest and Sue datasets have average read lengths of 265 and 96, respectively

Sample	**Expected**	**AbundantOTU**	**BOTUX**	**GramCluster**	**Mothur**	**UCLUST**
A. Number of predicted OTUs with 5 or more sequences						
A-2	22	28	23	28	27	33
B-18	22	28	27	40	28	75
Priest	23	24	24	24	36	26
Sue	43	50	55	56	136	76

Sample	**AbundantOTU**	**BOTUX**	**GramCluster**	**Mothur**	**UCLUST**
B. Percent of unclustered reads					
A-2	3.00	0.14	0.20	0.78	11.46
B-18	1.22	0.05	0.08	0.47	16.04
Priest	0.16	0.02	0.02	0.26	10.63
Sue	0.43	0.10	0.06	1.04	3.07

Sample	*AbundantOTU*	*BOTUX*	*GramCluster*	*Mothur*[*]	*UCLUST*
C. Wall-time in seconds					
A-2	60.67	40.00	11.13	2012	11.22
B-18	102.34	178.13	52.94	34,142	50.37
Priest	84.82	57.14	26.59	1922	8.93
Sue	68.33	20.44	50.10	199.36	4.17

[*] Mothur is run in single-core mode.