Reference-based OTU clustering for machine learning classification

Running title: Reference-based OTU clustering for ML classification

Courtney R. Armour[1], Kelly L. Sovacool[2], William L. Close[1,*], Begüm D. Topçuoğlu[1,#], Jenna Wiens[3], Patrick D. Schloss [1,†]

[1] Department of Microbiology and Immunology, University of Michigan, Ann Arbor, Michigan, USA

[2] Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, Michigan, USA

[3] Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, Michigan, USA

[*] Current Affiliation:

[#] Current Affiliation: Bristol Myers Squibb, Summit, New Jersey, USA

† To whom correspondence should be addressed: pschloss@umich.edu

observation format (max 1200 words, 2 figures, 25 ref)

Abstract

Machine learning classification of disease based on the gut microbiome often relies on clustering 16S rRNA gene sequences into operational taxonomic units (OTUs) to quantify microbial composition. The standard approach to clustering sequences into OTUs leverages the similarity of the sequences to each other rather than to a reference database. The abundance of each OTU is used to train a classification model. However, OTU assignments depend on the sequences in the data set and therefore can change if new data are added. This lack of stability complicates classification because in order to use the model to classify additional samples, the OTUs have to be reclustered to include the new sequences and the model retrained with the new OTU clusters. A new reference-based clustering algorithm, called OptiFit, addresses this issue by fitting new sequences into existing OTUs. While OptiFit is proven to produce high quality OTU clusters, it is unclear whether this method for fitting new sequence data into existing OTUs will impact the performance of classification models. We used OptiFit to cluster additional data into existing OTU clusters and quantified model performance in classifying a data set containing samples from patients with and without colonic screen relevant neoplasias (SRN). We compared the performance of this model to the standard procedure of clustering all the data together. We found that both approaches performed equally well in classifying SRNs. Moving forward, when OTUs are used in classification problems, OptiFit can be used to avoid the need to retrain models using reclustered sequences when classifying new samples.

Importance

There is great potential for using microbiome data to non-invasively diagnose people. One of the challenges with using classification models generated using the relative abundance of operational taxonomic unit (OTU) data is that 16S rRNA gene sequences are assigned to OTUs based on their similarity to other sequences in the dataset. If data are generated from new patients seeking a diagnosis, then it would be necessary to reassign sequences to OTUs and retrain the classification model. Yet there is a desire to have a single, validated model that can be deployed. To overcome this obstacle, we applied the OptiFit clustering algorithm which fits new sequence data to existing OTUs allowing the reuse of a consistent model. A random forest machine learning model deployed using OptiFit performed as well as the traditional reassignment and retraining approach. This result indicates there is potential for developing machine learning classification models based on OTU relative abundance data.

Gut community composition is useful as a resource for machine learning classification of diseases, including colorectal cancer (1, 2). Amplicon sequencing of the 16S rRNA gene is a reliable tool for assessing the taxonomic composition of microbial communities, which is the input to these models. Analysis of 16S rRNA gene sequence data generally relies on clustering of sequences based on similarity into operational taxonomic units (OTUs). The process of OTU clustering can either be reference-based or *de novo*. The quality of OTUs generated with reference-based clustering is generally poor compared to those generated with *de novo* clustering (3). While *de novo* clustering produces high-quality OTU clusters where sequences are accurately grouped based on similarity thresholds, the resulting OTU clusters depend on the data in the data set and the addition of new data could change the overall OTU clusters. The unstable nature of OTU clustering complicates deployment of machine learning models since integration of additional data requires reclustering all the data and retraining of the model. The ability to integrate new data into a validated model without reclustering and retraining could allow for deployment of a single model that new data can be continually added to. Recently Sovacool *et al* introduced OptiFit: a method for fitting new sequence data into existing OTUs (4). While OptiFit is proven to effectively fit new sequence data to existing OTU clusters, it is unknown if the use of OptiFit will have an impact on classification. Here we tested the ability of OptiFit to cluster new sequence data into existing OTU clusters for the purpose of classification of disease based on gut microbiome composition.

We compared two approaches, one using all of the data to generate OTU clusters and the other generating OTU clusters with a portion of the data and then fitting the remaining sequence data to the existing OTUs using OptiFit. In the first approach, all of the 16S rRNA sequence data was *de novo* clustered into OTUs with the OptiClust algorithm in mothur (5). The resulting abundance data was then split into training and testing sets, where the training set was used to tune hyperparameters and ultimately train the model. The testing set was then classified with the model and the performance of the model was quantified (Figure 1A). However, with this methodology we would have to regenerate the OTU clusters and retrain the model if we wanted to classify additional samples. The OptiFit algorithm (4) addresses this problem by enabling new sequences to be clustered into existing OTUs. The OptiFit workflow is similar to the OptiClust workflow where the data was clustered into OTUs and used to tune hyperparameters and ultimately train the model. Then, we used OptiFit to fit sequence data of samples not part of the original data set into the existing OTUs and used the same model to classify the samples (Figure 1B). To test how the model performance compares between these two methodologies, we used a publicly available data set of 16S rRNA gene sequences from stool samples of healthy subjects as well as subjects with SRN consisting of advanced adenoma and carcinoma (1). The data set was randomly split into an 80% train set and 20% test set. For

the standard OptiClust workflow, the training and test sets were *de novo* clustered together into OTUs then the resulting abundance table was split into the training and testing set. For the OptiFit workflow, the train set was clustered *de novo* into OTUs and the remaining test set was fit to the OTU clusters using the OptiFit algorithm. For both workflows, the abundance table of the train set was used to tune hyperparameters and train a random forest model to classify SRN. The test set was classified as either control or SRN using the trained models.To account for variation depending on the split of the data, the data set was randomly split 100 times and the process repeated for each of the 100 data splits. By comparing the model performance of classifying the samples in the test data set between the OptiFit and OptiClust algorithms, we quantified the impact of using OptiFit on model classification performance.

We first examined the quality of the resulting OTU clusters from the two algorithms using the Matthews correlation coefficient (MCC). The MCC score was quantified by examining all pairs of sequences and assessing whether they belonged together in an OTU based on their similarity (5). MCC scores range between negative one and one. A score of negative one means none of the sequences in an OTU are within the similarity threshold and any sequences within the similarity threshold are not in an OTU together. An MCC score of zero essentially means the sequences are randomly clustered. An MCC score of 1 means all sequences in an OTU are within the similarity threshold and all sequence pairs within the similarity threshold are in the same OTU. To ensure that OptiFit is appropriately integrating new sequence data into the existing OTUs, we expected the MCC scores produced by the OptiClust and OptiFit workflows to be similar. Since the data was only clustered once in the OptiClust workflow there was only one MCC score while the OptiFit workflow produced an MCC score for the OTU clusters from each data split. Overall the MCC scores were similar between OptiClust (MCC = 0.884) and OptiFit (average MCC = 0.879). This indicated that OptiFit performed as well as OptiClust when integrating new sequences into the existing OTUs.

After verifying that the quality of the OTUs was consistent between OptiClust and OptiFit, we examined the model performance for classifying samples in the held out test data set. To quantify model performance we used the OTU relative abundances from the training data from the OptiClust and OptiFit workflows to train a model to predict SRNs. Using the predicted and actual diagnosis classification, we calculated the area under the receiver operating characteristic curve (AUROC) for each data split to quantify model performance. During cross-validation (CV) training, the model performance was equivalent between the two algorithms (p-value = 0.13, OptiClust mean CV AUROC = 0.694, OptiFit mean CV AUROC = 0.697, Figure 2A). The trained model was then deployed to classify the samples of the test data as control or SRN. The performance on the test data was equivalent between the two algorithms (p-value = 0.63, OptiClust mean test AUROC = 0.709, OptiFit mean test AUROC = 0.712, Figure 2B,C) indicating that new data could be fit

to existing OTU clusters without impacting model performance.

We tested the ability of OptiFit to integrate new data into existing OTUs for the purpose of machine learning classification using OTU relative abundance. A potential problem with using OptiFit is that any sequences in the new data that do not map to the existing OTU clusters will be discarded resulting in a possible loss of information. However, we demonstrated that OptiFit can be used to fit new sequence data into existing OTU clusters and perform equally well in predicting SRN compared to clustering all of the sequence data together. The ability to integrate data from new samples into existing OTUs enables the deployment of a single machine learning model. These results are based on a single data set and disease. Further analysis is needed to determine the number of samples that are necessary to build a robust model capable of classifying diverse samples. A robust machine learning model could be implemented as part of a non-invasive and low-cost aid in diagnosing SRN and other diseases.

Materials and Methods

Data Set. Raw 16S rRNA gene sequence data isolated from human stool samples was downloaded from NCBI Sequence Read Archive (accession no. SRP062005) (1, 6). This data set contains stool samples from a total of 490 subjects. For this analysis, samples from subjects identified in the metadata as normal, high risk normal, or adenoma were categorized as “normal” while samples from subjects identified as advanced adenoma or carcinoma were categorized as “screen relevant neoplasia” (SRN). The resulting data set consisted of 261 normal samples and 229 SRN samples.

Data Processing. The full dataset was preprocessed with mothur (v1.47) (7) to join forward and reverse reads, merge duplicate reads, align to the SILVA reference database (v132) (8), precluster, remove chimeras with UCHIME (6), assign taxonomy, and remove non-bacterial reads following the Schloss Lab MiSeq standard operating procedure described on the mothur website (https://mothur.org/wiki/miseq_sop/). 100 splits of the 490 samples were generated where 80% of the samples (392 samples) were randomly assigned to the training set and the remaining 20% (98 samples) were assigned to the test set. Using 100 splits of the data accounts for the variation that may be observed depending on the samples that are in the training or test sets. Each sample was in the training set an average of 80 times (SD=4.1) and the test set an average of 20 times (SD=4.1).

The data was processed through two workflows. First, the standard workflow using the OptiClust algorithm (5). In this pathway, all of the data was clustered together with OptiClust to generate OTUs and the resulting abundance tables were split into the training and testing sets. In the second workflow, the preprocessed

data was split into the training and testing sets. The training set was clustered into OTUs, then the test set was fit to the OTUs of the training set using the OptiFit algorithm (4). The OptiFit algorithm was run with method open so that any sequences that did not map to the existing OTU clusters would form new OTUs. For both pathways, the shared files were sub-sampled to 10,000 reads per sample.

Machine Learning. Machine learning using Random Forest was conducted with the R package mikrompl (v 1.2.0) (9) to predict the diagnosis (SRN or normal) for the samples in the test set for each data split. The training set was preprocessed to normalize OTU counts (scale/center), collapse correlated OTUs, and remove OTUs with zero-variance. The preprocessing from the training set was then applied to the test set. Any OTUs in the test set that were not in the training set were removed. P values comparing model performance were calculated as previously described (10). The averaged ROC curves were plotted by taking the average and standard deviation of the sensitivity at each specificity value.

Code Availability. The analysis workflow was implemented in Snakemake (11) . Scripts for analysis were written in R (12) and GNU bash (13). The software used includes mothur v1.47.0 (7), RStudio (14), the Tidyverse metapackage (15), R Markdown (16), the SRA toolkit (17), and conda (18). The complete workflow and supporting files required to reproduce this study are available at: https://github.com/SchlossLab/Armour_OptiFitGLNE_XXXX_2021

Acknowledgements

This work was supported through a grant from the NIH (R01CA215574).

References

1. **Baxter NT**, **Ruffin MT**, **Rogers MAM**, **Schloss PD**. 2016. Microbiota-based model improves the sensitivity of fecal immunochemical test for detecting colonic lesions. Genome Medicine **8**:37. doi:10.1186/s13073-016-0290-3.

2. **Zackular JP**, **Rogers MAM**, **Ruffin MT,I**, **Schloss PD**. 2014. The human gut microbiome as a screening tool for colorectal cancer. Cancer Prevention Research **7**:1112–1121. doi:10.1158/1940-6207.CAPR-14-0129.

3. **Westcott SL**, **Schloss PD**. 2015. De novo clustering methods outperform reference-based methods for assigning 16S rRNA gene sequences to operational taxonomic units. PeerJ **3**:e1487. doi:10.7717/peerj.1487.

4. **Sovacool KL**, **Westcott SL**, **Mumphrey MB**, **Dotson GA**, **Schloss PD**. 2022. OptiFit: An improved method for fitting amplicon sequences to existing OTUs. mSphere **7**:e00916–21. doi:10.1128/msphere.00916-21.

5. **Westcott SL**, **Schloss PD**. 2017. OptiClust, an improved method for assigning amplicon-based sequence data to operational taxonomic units. mSphere **2**:e00073–17. doi:10.1128/mSphereDirect.00073-17.

6. **Edgar RC**, **Haas BJ**, **Clemente JC**, **Quince C**, **Knight R**. 2011. UCHIME improves sensitivity and speed of chimera detection. Bioinformatics **27**:2194–2200. doi:10.1093/bioinformatics/btr381.

7. **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**. 2009. Introducing mothur: Open-source, platform-independent, community-supported software for describing and comparing microbial communities. Applied and Environmental Microbiology **75**:7537–7541. doi:10.1128/AEM.01541-09.

8. **Quast C**, **Pruesse E**, **Yilmaz P**, **Gerken J**, **Schweer T**, **Yarza P**, **Peplies J**, **Glöckner FO**. 2013. The SILVA ribosomal RNA gene database project: Improved data processing and web-based tools. Nucleic Acids Research **41**:D590–D596. doi:10.1093/nar/gks1219.

9. **Topçuoğlu BD**, **Lapp Z**, **Sovacool KL**, **Snitkin E**, **Wiens J**, **Schloss PD**. 2021. mikropml: User-Friendly R Package for Supervised Machine Learning Pipelines. Journal of Open Source Software **6**:3073. doi:10.21105/joss.03073.

10. **Topçuoğlu BD**, **Lesniak NA**, **Ruffin MT**, **Wiens J**, **Schloss PD**. 2020. A framework for effective application of machine learning to microbiome-based classification problems. mBio **11**:e00434–20. doi:10.1128/mBio.00434-20.

11. **Koster J**, **Rahmann S**. 2012. Snakemake–a scalable bioinformatics workflow engine. Bioinformatics **28**:2520–2522. doi:10.1093/bioinformatics/bts480.

12. **R Core Team**. 2020. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

13. **GNU Project**. Bash reference manual.

14. **RStudio Team**. 2019. RStudio: Integrated development environment for r. RStudio, Inc., Boston, MA.

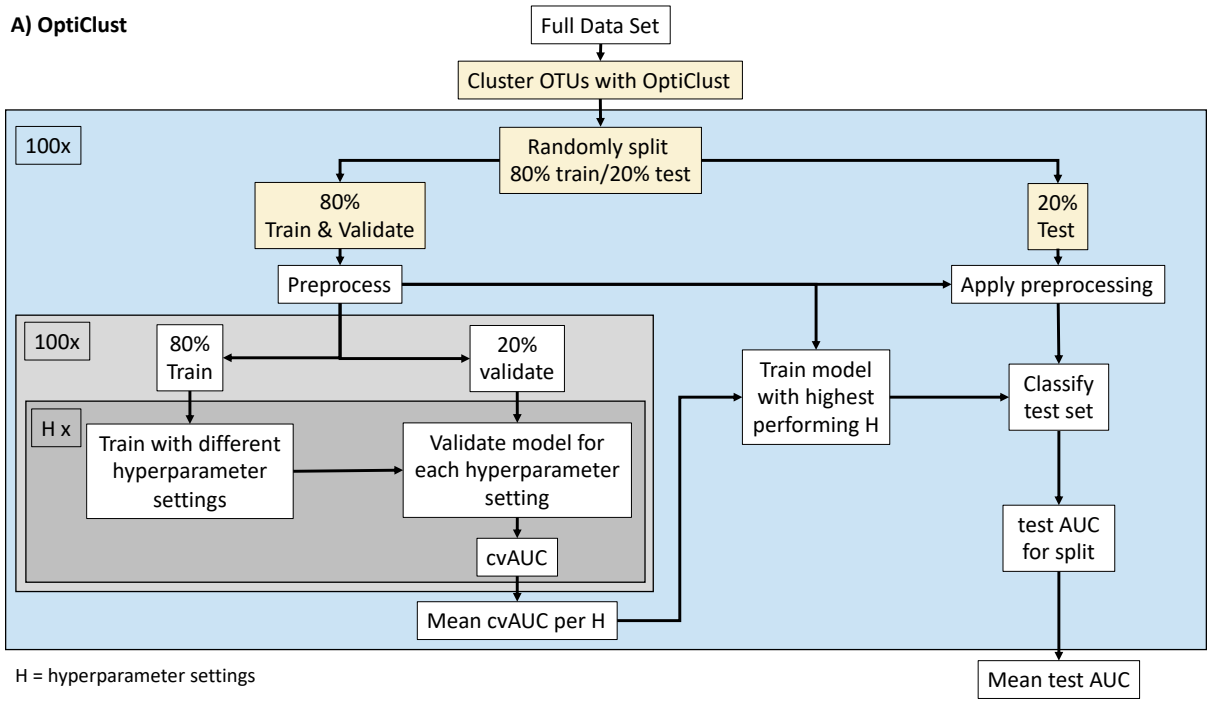
15. **Wickham H**, **Averick M**, **Bryan J**, **Chang W**, **McGowan LD**, **François R**, **Grolemund G**, **Hayes A**, **Henry L**, **Hester J**, **Kuhn M**, **Pedersen TL**, **Miller E**, **Bache SM**, **Müller K**, **Ooms J**, **Robinson D**, **Seidel DP**, **Spinu V**, **Takahashi K**, **Vaughan D**, **Wilke C**, **Woo K**, **Yutani H**. 2019. Welcome to the Tidyverse. Journal of Open Source Software **4**:1686. doi:10.21105/joss.01686.

16. **Xie Y**, **Allaire JJ**, **Grolemund G**. 2018. R Markdown: The Definitive Guide. Taylor & Francis, CRC Press.

17. SRA-Tools - NCBI. http://ncbi.github.io/sra-tools/.

18. 2016. Anaconda Software Distribution. Anaconda Documentation. Anaconda Inc.

Figures



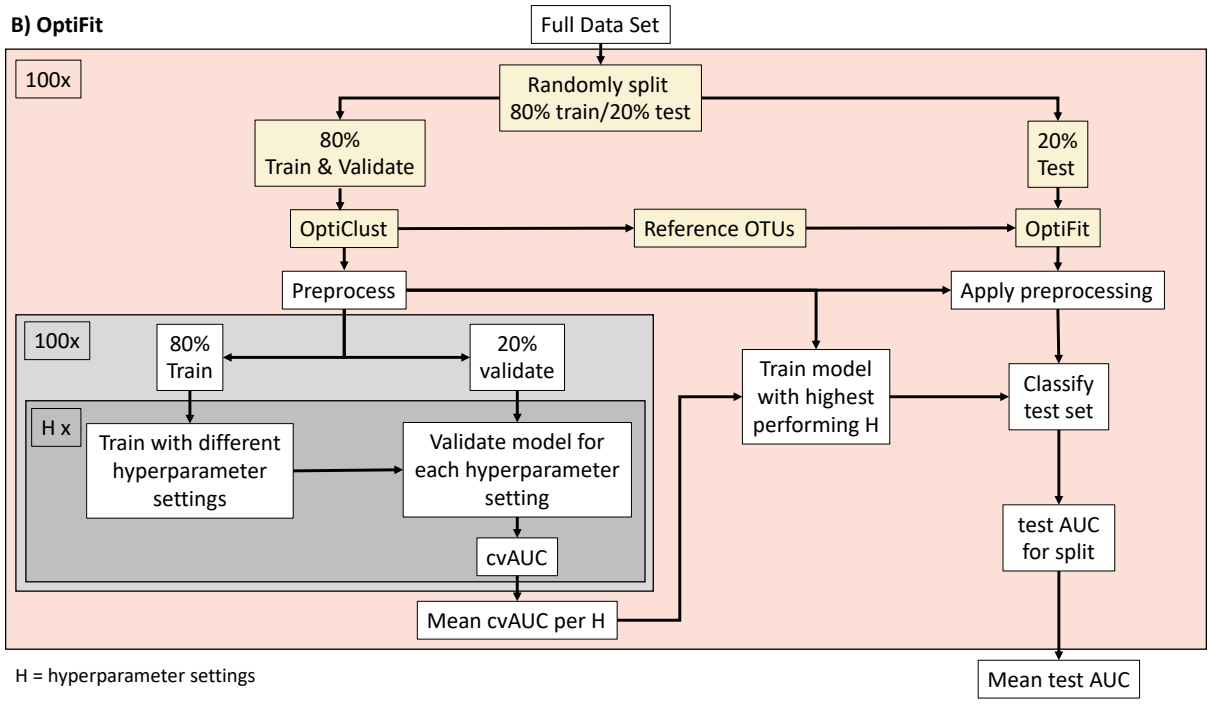
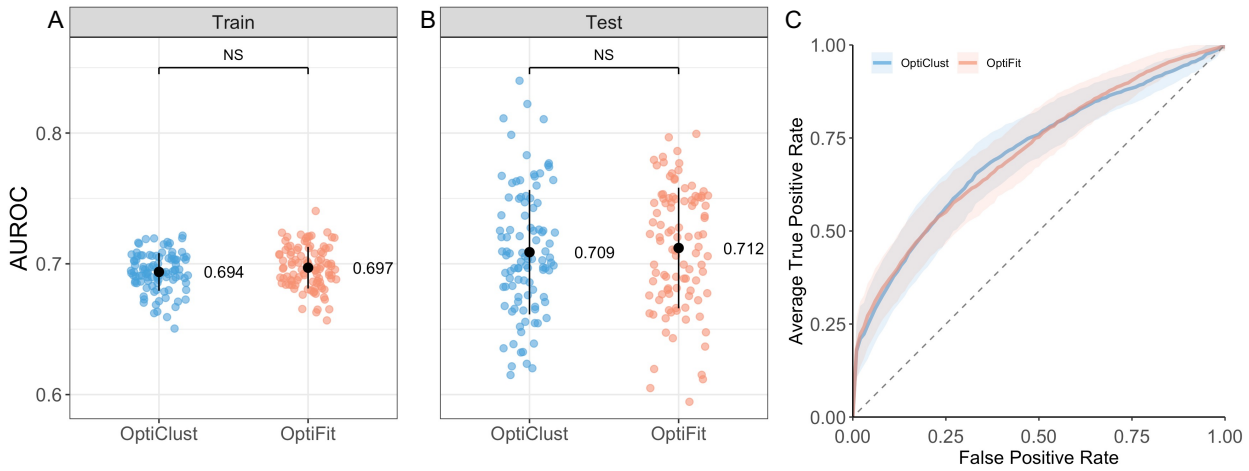


Figure 1: Workflows. A) OptiClust workflow: The full data set was clustered into OTUs using the OptiClust algorithm in mothur. The data was then split into two sets where 80% of the samples were assigned to the training set and 20% to the testing set. The training set was preprocessed with mikropml to normalize values (scale/center), collapse correlated features, and remove features with zero-variance. Using mikropml, the

training set was split into train and validate sets to compare results using different hyperparameter settings. The highest performing hyperparameter setting was then used to train the model with the full training set. The preprocessing scale from the training set was applied to the test data set, then the trained model was used to classify the samples in the test set. Based on the actual classification and predicted classification, the area under the receiver operating characteristic curve (AUROC) was calculated to summarize model performance. The entire process was repeated 100 times to account for variability depending on the split of the data resulting in a total of 100 AUROC values summarizing the performance of the standard OptiClust workflow. **B)** OptiFit workflow: The data set was first split into two sets where 80% of the samples were assigned to the training set and 20% to the testing set. The training set was then clustered into OTUs using the OptiClust algorithm in mothur. The resulting abundance data was preprocessed with mikropml to normalize values (scale/center), collapse correlated features, and remove features with zero-variance. Using mikropml, the training set was split into train and validate sets to compare results using different hyperparameter settings. The highest performing hyperparameter setting was then used to train the model with the full training set. The OptiFit algorithm in mothur was used to cluster the left out testing data set using the OTUs of the training set as a reference. The preprocessing scale from the training set was applied to the test data set, then the trained model was used to classify the samples in the test set. Based on the actual classification and predicted classification, the area under the receiver operating characteristic curve (AUROC) was calculated to summarize model performance. The entire process was repeated 100 times to account for variability depending on the split of the data resulting in a total of 100 AUROC values summarizing the performance of the new OptiFit workflow.



205

206 **Figure 2: Model Performance. A)** Area under the receiver operating characteristic (AUROC) curve during
207 cross-validation for the OptiClust and OptiFit workflows. Mean and standard deviation of the AUROC is
208 represented by the black dot and whiskers. Mean AUROC is printed to the right of the points. **B)** AUROC
209 on the test data for the OptiClust and OptiFit workflows. Mean and standard deviation of the AUROC is
210 represented by the black dot and whiskers. Mean AUROC is printed to the right of the points. **C)** Receiver
211 operating characteristic (ROC)Averaged ROC curves