Ovarian Cancer Histotypes: Report of Statistical Findings

Derek Chiu

2024-07-10

Contents

List of Figures

List of Tables

Preface

This report of statistical findings describes the classification of ovarian cancer histotypes using data from NanoString CodeSets.

Marina Pavanello conducted the initial exploratory data analysis, Cathy Tang implemented class imbalance techniques, Derek Chiu conducted the normalization and statistical analysis, and Lauren Tindale and Aline Talhouk are the project leads.

1. Introduction

Ovarian cancer has five major histotypes: high-grade serous carcinoma (HGSC), low-grade serous carcinoma (LGSC), endometrioid carcinoma (ENOC), mucinous carcinoma (MUC), and clear cell carcinoma (CCOC). A common problem with classifying these histotypes is that there is a class imbalance issue. HGSC dominates the distribution, commonly accounting for 70% of cases in many patient cohorts, while the other four histotypes are spread over the rest of the cases. Subsampling methods like up-sampling, down-sampling, and SMOTE can be used to mitigate this problem.

The supervised learning is performed under a consensus framework: we consider various classification algorithms and use evaluation metrics like accuracy, F1-score, Kappa, and G-mean to inform the decision of which methods to carry forward for prediction in confirmation and validation sets.

2. Methods

2.1 Pre-Processing

2.1.1 Case Selection

Raw data comes from three NanoString CodeSets (CS): CS1, CS2, and CS3. We divide the data into training, confirmation, and validation sets by using samples from these sets of cohorts:

- Training
  - CS1: MAYO, OOU, OOUE, VOA, MTL
  - CS2: MAYO, OOU, OOUE, OVAR3, VOA, ICON7, JAPAN, MTL, POOL-CTRL
  - CS3: OOU, OOUE, VOA, POOL-1, POOL-2, POOL-3
- Confirmation:
  - CS3: TNCO
- Validation:
  - CS3: DOVE4

2.1.2 Quality Control

Samples that failed any of the following NanoString quality control conditions were removed:

- `linFlag`: linearity of positive controls with positive control concentrations is less than 0.95, or linearity measures are unknown
- `imagingFlag`: percent of field of view is less than 75%
- `spcFlag`: smallest positive control is less than the lower limit of detection (negative control average expression less two times the negative control standard deviation), or negative control average expression equals zero
- `normFlag`: signal to noise ratio less than 100, or percent of genes detected is less than 50. Note: these thresholds were determined by examining the %GD vs. SNR relationship below.

2.1.3 Normalization

The full training set (n=1243) is comprised of data from CodeSets (CS) 1, 2, and 3. All CodeSets were first normalized to housekeeping genes, then different approaches were taken for subsequent normalizations of each CodeSet.

CS1 was normalized to CS3 using five "Random1" reference samples. These reference samples are randomly selected from CS1 among all samples in the three CodeSets that share common otta IDs, such that we obtain

one sample from each of the five histotypes. Then, we use the reference-based method to normalize CS1 to CS3 across their common genes, for the remaining expression samples Talhouk et al. (2016).

Similarly, CS2 was normalized to CS3 using the same "Random1" reference samples, now taken from CS2. Normalization was performed across common genes between CS2 and CS3.

For CS3, we first split the dataset into three sites: Vancouver, USC, and AOC. We use the CS3-Vancouver subset as a "reference standard", and normalized CS3-USC and CS3-AOC to CS3-Vancouver using a "Random1" reference set randomly selected among samples common between Vancouver, USC, and AOC. Finally, the CS3-Vancouver expression samples are included in the training set without further normalization.

2.1.4 Final Processing

We map ovarian histotypes to all remaining samples and keep the major histotypes for building the predictive model: high-grade serous carcinoma (HGSC), clear cell ovarian caricoma (CCOC), endometrioid ovarian carcinoma (ENOC), low-grade serous carcinoma (LGSC), mucinous carcinoma (MUC).

Duplicate cases (two samples with the same ottaID) were removed before generating the final training set to use for fitting the classification models. All CS3 cases were preferred over CS1 and CS2, and CS3-Vancouver cases were preferred over CS3-AOC and CS3-USC when selecting duplicates.

The final training set used only genes that were common across all three CodeSets.

2.2 Classifiers

We use 4 classification algorithms in the supervised learning framework for the Training Set. The pipeline was run using SLURM batch jobs submitted to a partition on a CentOS 7 server. All resampling techniques, pre-processing, model specification, hyperparameter tuning, and evaluation metrics were implemented using the `tidymodels` suite of packages. The classifiers we used are:

- Random Forest (`rf`)
- Support Vector Machine (`svm`)
- XGBoost (`xgb`)
- Regularized Multinomial Regression (`mr`)

2.2.1 Resampling of Training Set

We used a nested cross-validation design to assess each classifier while also performing hyperparameter tuning. An outer 5-fold CV stratified by histotype was used together with an inner 5-fold CV with 2 repeats stratified by histotype. This design was chosen such that the test sets of the inner resamples would still have a reasonable number of samples belonging to the smallest minority class.

The outer resampling method cannot be the bootstrap, because the inner training and inner test sets will likely contain the same samples as a result of sampling with replacement in the outer training set. This phenomenon might result in inflated performance as some observations are used both to train and evaluate the hyperparameter tuning in the inner loop.

2.2.2 Hyperparameter Tuning

The following specifications for each classifier were used for tuning hyperparameters:

- `rf` and `xgb`: The number of trees were fixed at 500. Other hyperparameters were tuned across 10 randomly selected points in a latin hypercube design.

- `svm`: Both the cost and sigma hyperparameters were tuned across 10 randomly selected points in a latin hypercube design. We tuned the cost parameter in the range [1, 8]. The range for tuning the sigma parameter was obtained from the 10% and 90% quantiles of the estimation using the `kernlab::sigest()` function.
- `mr`: We generated 10 randomly selected points in a latin hypercube design for the penalty (lambda) parameter. Then, we generated 10 evenly spaced points in [0, 1] for the mixture (alpha) parameter in the regularized multinomial regression model. These two sets of 10 points were crossed to generate a tuning grid of 100 points.

The hyperparameter combination that resulted in the highest average F1-score across the inner training sets was selected for each classifier to use as the model for assessing prediction performance in the outer training loop.

2.2.3 Subsampling

Here are the specifications of the subsampling methods used to handle class imbalance:

- None: No subsampling is performed
- Down-sampling: All levels except the minority class are sampled down to the same frequency as the minority class
- Up-sampling: All levels except the majority class are sampled up to the same frequency as the majority class
- SMOTE: All levels except the majority class have synthetic data generated until they have the same frequency as the majority class
- Hybrid: All levels except the majority class have synthetic data generated up to 50% of the frequency of the majority class, then the majority class is sampled down to the same frequency as the rest.

The figure below helps visualize how the distribution of classes changes when we apply subsampling techniques to handle class imbalance:

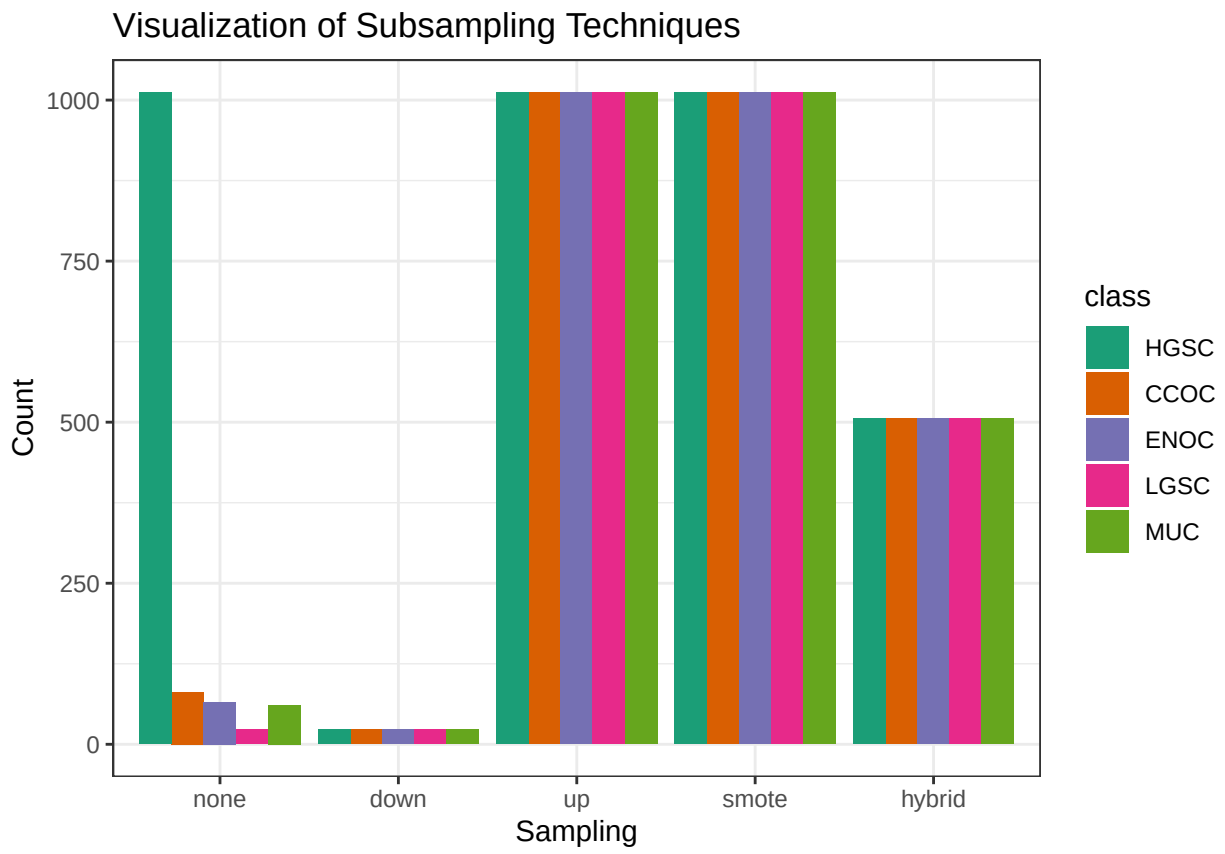

Figure 2.1: Visualization of Subsampling Techniques

2.2.4 Workflows

The 4 **algorithms** and 5 **subsampling** methods are crossed to create 20 different classification **workflows**. For example, the `hybrid_xgb` workflow is a classifier that first pre-processes a training set by applying a hybrid subsampling method, and then proceeds to use the XGBoost algorithm to classify ovarian histotypes.

2.3 Two-Step Algorithm

The HGSC histotype comprises of approximately 80% of cases among ovarian carcinoma patients, while the remaining 20% of cases are relatively, evenly distributed among ENOC, CCOC, LGSC, and MUC histotypes. We can implement a two-step algorithm as such:

- Step 1: use binary classification for HGSC vs. non-HGSC
- Step 2: use multinomial classification for the remaining non-HGSC classes

Let

$$X_k = \text{Training data with k classes}$$
$$C_k = \text{Class with highest } F_1 \text{ score from training } X_k$$
$$W_k = \text{Workflow associated with } C_k$$

Figure 2.2 shows how the two-step algorithm works:

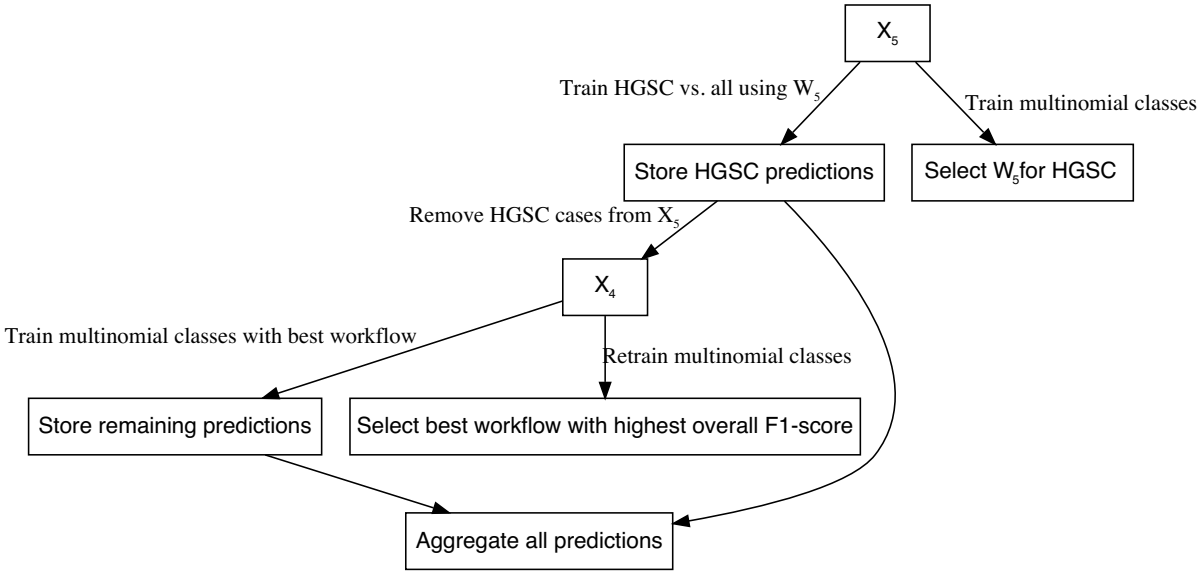

Figure 2.2: Two-Step Algorithm

Although the class imbalance problem is mostly eliminated in Step 2 after removing the HGSC cases, we still use the same subsampling method in Step 2 as was used in Step 1 to keep the algorithm consistent.

2.3.1 Aggregating Predictions

The aggregation for two-step predictions is quite straightforward:

1. Predict HGSC vs. non-HGSC
2. Among all non-HGSC cases, predict CCOC vs. LGSC vs. MUC vs. ENOC

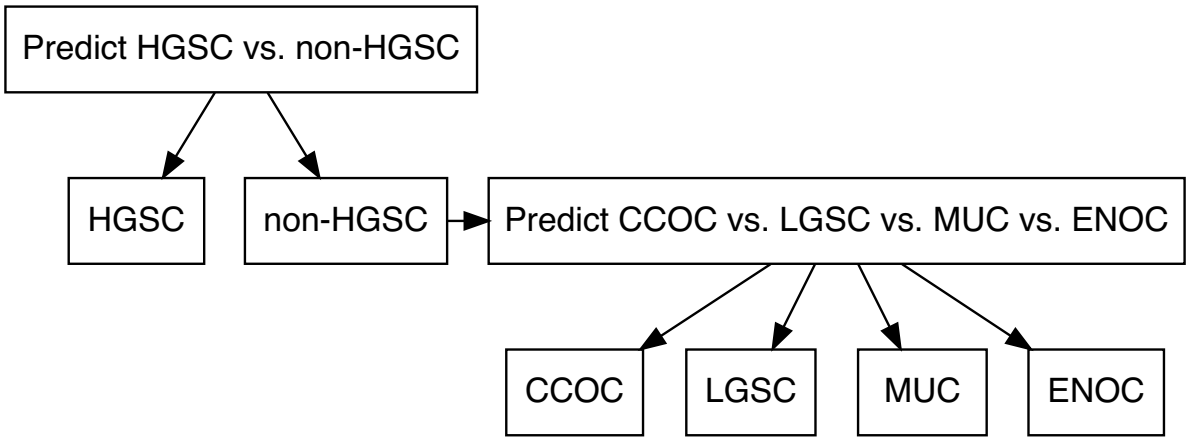

Figure 2.3: Aggregating Predictions for Two-Step Algorithm

2.4 Sequential Algorithm

Instead of training on k classes simultaneously using multinomial classifiers, we can use a sequential algorithm that performs k-1 one-vs-all binary classifications iteratively to obtain a final prediction of all cases. At each step in the sequence, we classify one class vs. all other classes, where the classes that make up the "other" class are those not equal to the current "one" class and excluding all "one" classes from previous steps. For example, if the "one" class in step 1 was HGSC, the "other" classes would include CCOC, ENOC, LGSC, and MUC. If the "one" class in step 2 was CCOC, the "other" classes include ENOC, LGSC, and MUC.

The order of classes and workflows to use at each step in the sequential algorithm must be determined using a retraining procedure. After removing the data associated with a particular class, we retrain using the remaining data using multinomial classifiers as described before. The class and workflow to use for the next step in the sequence is selected based on the best per-class evaluation metric value (e.g. F1-score).

Figure 2.4 illustrates how the sequential algorithm works for K=5, using ovarian histotypes as an example for the classes.

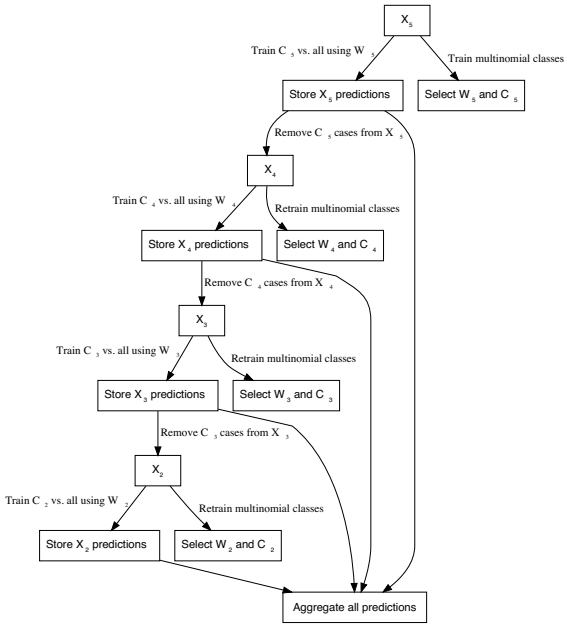

Figure 2.4: Sequential Algorithm

The subsampling method used in the first step of the sequential algorithm is used in all subsequent steps in order to maintain data pre-processing consistency. As a result, we are only comparing classification algorithms within one subsampling method across the entire sequential algorithm.

2.4.1 Aggregating Predictions

We have to aggregate the one-vs-all predictions from each of the sequential algorithm workflows in order to obtain a final class prediction on a holdout test set. Each sequential workflow has to be assessed on every sample to ensure that cases classified into the "all" class from a previous step of the sequence are eventually assigned a predicted class. For example, say that based on certain class-specific metrics we determined that the order of classes in the sequential algorithm was to predict HGSC vs. non-HGSC, CCOC vs. non-CCOC, LGSC vs. non-LGSC, and then MUC vs. ENOC. Figure 2.5 illustrates how the final predictions are assigned:

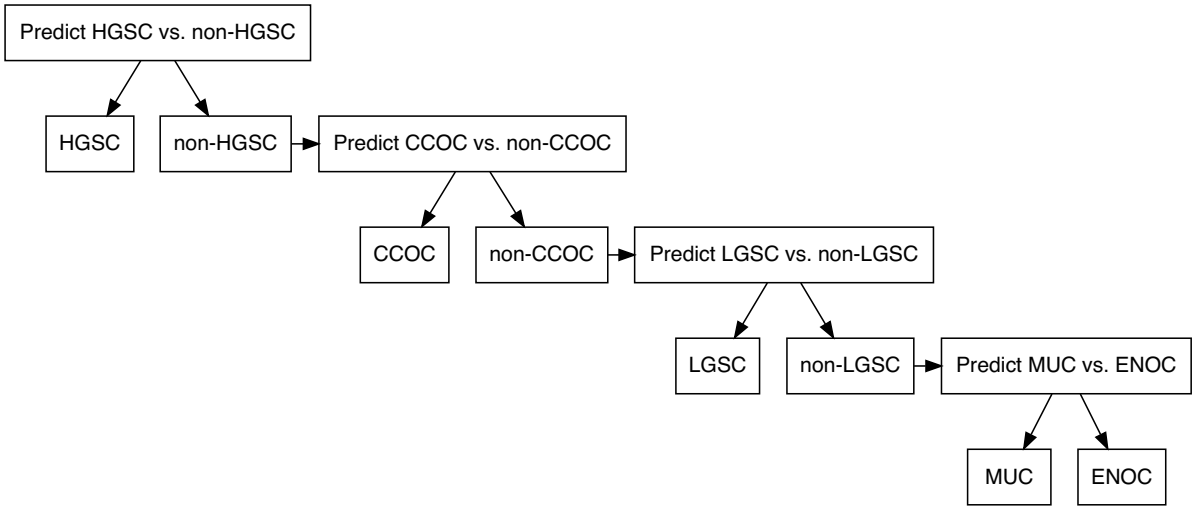

Figure 2.5: Aggregating Predictions for Sequential Algorithm

2.5 Gene Optimization

We want to discover an optimal set of genes for the classifiers while including specific genes from other studies such as PrOTYPE and SPOT. A total of 72 genes are used in the classifier training set.

There are 16 genes in the classifier set that overlap with the PrOTYPE classifier: COL11A1, CD74, CD2, TIMP3, LUM, CYTIP, COL3A1, THBS2, TCF7L1, HMGA2, FN1, POSTN, COL1A2, COL5A2, PDZK1IP1, FBN1.

There are also 13 genes in the classifier set that overlap with the SPOT signature: HIF1A, CXCL10, DUSP4, SOX17, MITF, CDKN3, BRCA2, CEACAM5, ANXA4, SERPINE1, TCF7L1, CRABP2, DNAJC9.

We obtain a total of 28 genes from the union of PrOTYPE and SPOT genes that we want to include in the final classifier, regardless of model performance. We then incrementally add genes one at a time from the remaining 44 candidate genes based on an overall variable importance rank to the set of 28 base genes and recalculate performance metrics. The number of genes at which the performance peaks or starts to plateau may indicate an optimal gene set model for us to compare with the full set model.

2.5.1 Variable Importance

Variable importance is calculated using either a model-based approach if it is available, or a permutation-based VI score otherwise. The variable importance scores are averaged across the outer training folds, and then ranked from highest to lowest.

For the sequential and two-step classifiers, we calculate an overall VI rank by taking the cumulative union of genes at each variable importance rank across all sequences, until all genes have been included.

The variable importance measures are:

- Random Forest: impurity measure (Gini index)
- XGBoost: gain (fractional contribution of each feature to the model based on the total gain of the corresponding features's splits)
- SVM: permutation based p-values
- Multinomial regression: absolute value of estimated coefficients at cross-validated lambda value

2.6 Performance Evaluation

2.6.1 Class Metrics

We use the accuracy, sensitivity, specificity, F1-score, kappa, balanced accuracy, and geometric mean, as class metrics to measure both training and test performance between different workflows. Multiclass extensions of these metrics can be calculated except for F1-score, where we use macro-averaging to obtain an overall metric. Class-specific metrics are calculated by recoding classes into one-vs-all categories for each class.

2.6.1.1 Accuracy

The accuracy is defined as the proportion of correct predictions out of all cases:

$$\text{accuracy} = \frac{TP}{TP + FP + FN + TN}$$

2.6.1.2 Sensitivity

Sensitivity is the proportional of correctly predicted positive cases, out of all cases that were truly positive

$$\text{sensitivity} = \frac{TP}{TP + FN}$$

2.6.1.3 Specificity

Specificity is the proportional of correctly predicted negative cases, out of all cases that were truly negative.

$$\text{specificity} = \frac{TN}{TN + FP}$$

2.6.1.4 F1-Score

The F-measure can be thought of as a harmonic mean between precision and recall:

$$F_{meas} = \frac{(1 + \beta^2) \times precision \times recall}{(\beta^2 \times precision) + recall}$$

The β value can be adjusted to place more weight upon precision or recall. The most common value is β is 1, which is also commonly known as the F1-score. A multiclass extension doesn't exist for the F1-score, so we use macro-averaging to calculate this metric when there are more than two classes. For example, with k classes, the macro-averaged F1-score is equal to:

$$F_{1macro} = \frac{1}{k} \sum_{i=1}^{k} F_{1i}$$

where each F_{1i} is the F1-score computed frrom recoding classes into $k = i$ vs. $k \neq i$.

In situations where there is not at least one predicted case for each of the classes (e.g. for a poor classifier), F_{1i} is undefined because the per-class precision of class i is undefined. Those F_{1i} terms are removed from the F_{1macro} equation and the resulting value may be inflated. Interpreting the F1-score in such a case would be misleading.

2.6.1.5 Kappa

Kappa is the defined as:

$$\text{kappa} = \frac{p_0 - p_e}{1 - p_e}$$

where p_0 is the observed agreement among raters and p_e is the hypothetical probability of agreement due to random chance.

2.6.1.6 Balanced Accuracy

Balanced accuracy is the arithmetic mean of sensitivity and specificity.

$$\text{Balanced Accuracy} = \frac{\text{Sensitivity} + \text{Specificity}}{2}$$

2.6.1.7 Geometric Mean

The geometric mean (G-mean) is the k^{th} root of the product of class-specific sensitivities for k classes:

$$\text{G-mean} = \sqrt[k]{\prod_{i=1}^{k} \text{Sensitivity}_k}$$

The G-mean generalizes easily for the multiclass scenario.

2.6.2 AUC

The area under the receiver operating curve (AUC) is calculated by adding up the area under the curve formed by plotting sensitivity vs. 1 - specificity. The Hand-till method is used as a multiclass extension for the AUC.

We did not use AUC to measure class-specific training set performance because combining predicted probabilities in a one-vs-all fashion might be potentially misleading. The sum of probabilities that add up to the "other" class is not equivalent to the predicted probability of the "other" class when using a multiclass classifier.

Instead, we only reported ROC curves and their associated AUCs for the test set performance of the sequential and two-step algorithms.

3. Distributions

3.1 CodeSet Counts

3.2 Histotypes in Classifier Data

3.3 Cohorts in Classifier Data

3.4 Quality Control

3.4.1 Failed Samples

We use an aggregated `QCFlag` that considers a sample to have failed QC if any of the following conditions are true:

- `linFlag`: linearity of positive controls with positive control concentrations is less than 0.95, or linearity measures are unknown
- `imagingFlag`: percent of field of view is less than 75%
- `spcFlag`: smallest positive control is less than the lower limit of detection (negative control average expression less two times the negative control standard deviation), or negative control average expression equals zero
- `normFlag`: signal to noise ratio less than 100, or percent of genes detected is less than 50. Note: these thresholds were determined by examining the %GD vs. SNR relationship below.

Table 3.1: Training Set Counts by CodeSet and Processing Stage

Processing Stage	CS1	CS2	CS3	Total
Selected Cohorts	294	903	2477	3674
QC	286	882	2273	3441
Normalized to Reference	263	827	2095	3185
CS3: remove test sets, add AOC/USC	263	827	514	1604
Major Histotypes	244	779	506	1529
Removed Duplicates	77	698	468	1243

Table 3.2: Histotype Distribution by CodeSet and Processing Stage

Variable	Levels	CS1	CS2	CS3	Total
Pre-QC					
Histotype	HGSC	120 (45%)	641 (79%)	515 (92%)	1276 (78%)
	CCOC	48 (18%)	61 (7%)	11 (2%)	120 (7%)
	ENOC	60 (22%)	32 (4%)	11 (2%)	103 (6%)
	MUC	19 (7%)	60 (7%)	12 (2%)	91 (6%)
	LGSC	20 (7%)	21 (3%)	9 (2%)	50 (3%)
Total	N (%)	267 (16%)	815 (50%)	558 (34%)	1640 (100%)
Major Histotypes					
Histotype	HGSC	116 (48%)	622 (80%)	475 (94%)	1213 (79%)
	CCOC	44 (18%)	54 (7%)	8 (2%)	106 (7%)
	ENOC	55 (23%)	27 (3%)	8 (2%)	90 (6%)
	MUC	15 (6%)	57 (7%)	9 (2%)	81 (5%)
	LGSC	14 (6%)	19 (2%)	6 (1%)	39 (3%)
Total	N (%)	244 (16%)	779 (51%)	506 (33%)	1529 (100%)
Deduplicated					
Histotype	HGSC	9 (12%)	552 (79%)	451 (96%)	1012 (81%)
	CCOC	25 (32%)	52 (7%)	4 (1%)	81 (7%)
	ENOC	37 (48%)	25 (4%)	4 (1%)	66 (5%)
	MUC	3 (4%)	53 (8%)	5 (1%)	61 (5%)
	LGSC	3 (4%)	16 (2%)	4 (1%)	23 (2%)
Total	N (%)	77 (6%)	698 (56%)	468 (38%)	1243 (100%)

Table 3.3: Histotype Distribution in Confirmation and Validation Sets

Variable	Levels	Confirmation	Validation
Histotype	HGSC	422 (66%)	667 (75%)
	CCOC	75 (12%)	79 (9%)
	ENOC	106 (16%)	105 (12%)
	MUC	27 (4%)	26 (3%)
	LGSC	13 (2%)	18 (2%)
Total	N (%)	643 (42%)	895 (58%)

Table 3.4: Cohort Distribution in Training, Confirmation, and Validation Sets

CodeSet	Cohort	Training	Confirmation	Validation
CS1	MAYO	2	0	0
CS1	MTL	1	0	0
CS1	OOU	53	0	0
CS1	OOUE	1	0	0
CS1	VOA	20	0	0
CS2	ICON7	365	0	0
CS2	JAPAN	8	0	0
CS2	MAYO	40	0	0
CS2	MTL	59	0	0
CS2	OOU	27	0	0
CS2	OOUE	18	0	0
CS2	OVAR3	135	0	0
CS2	VOA	46	0	0
CS3	OOU	18	0	0
CS3	OOUE	11	0	0
CS3	VOA	439	0	0
CS3	TNCO	0	643	0
CS3	DOVE4	0	0	895

Table 3.5: Number of failed samples by CodeSet and fail condition

CodeSet	CodeSet Total	linFlag	imagingFlag	spcFlag	normFlag	QCFlag	n
CS1	8	Passed	Failed	Passed	Passed	Failed	3
		Passed	Passed	Passed	Failed	Failed	5
CS2	21	Failed	Passed	Failed	Failed	Failed	2
		Failed	Passed	Passed	Passed	Failed	2
		Passed	Passed	Passed	Failed	Failed	17
CS3	204	Passed	Failed	Passed	Passed	Failed	4
		Passed	Passed	Passed	Failed	Failed	200

3.4.2 %GD vs. SNR

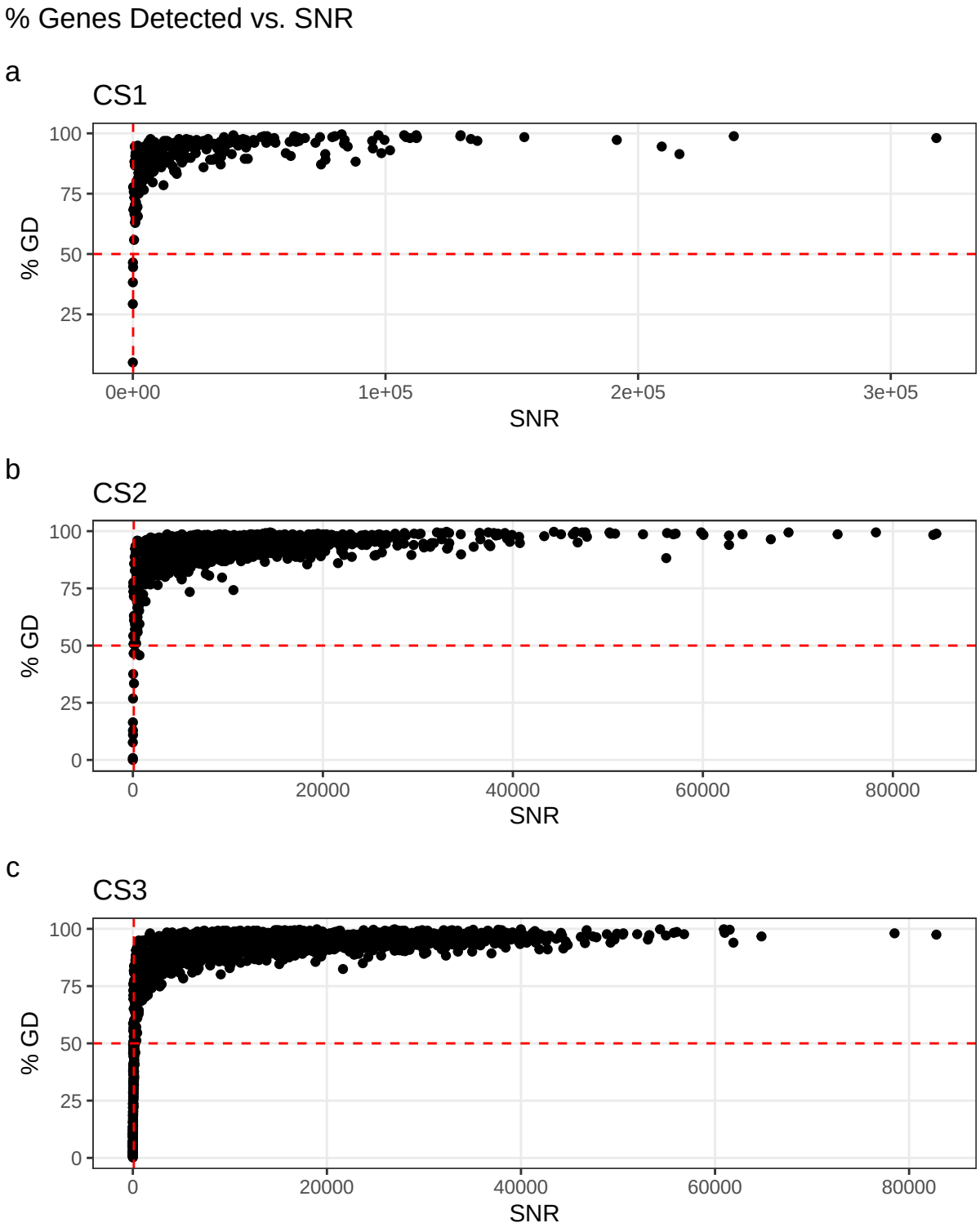

Figure 3.1: % Genes Detected vs. Signal to Noise Ratio

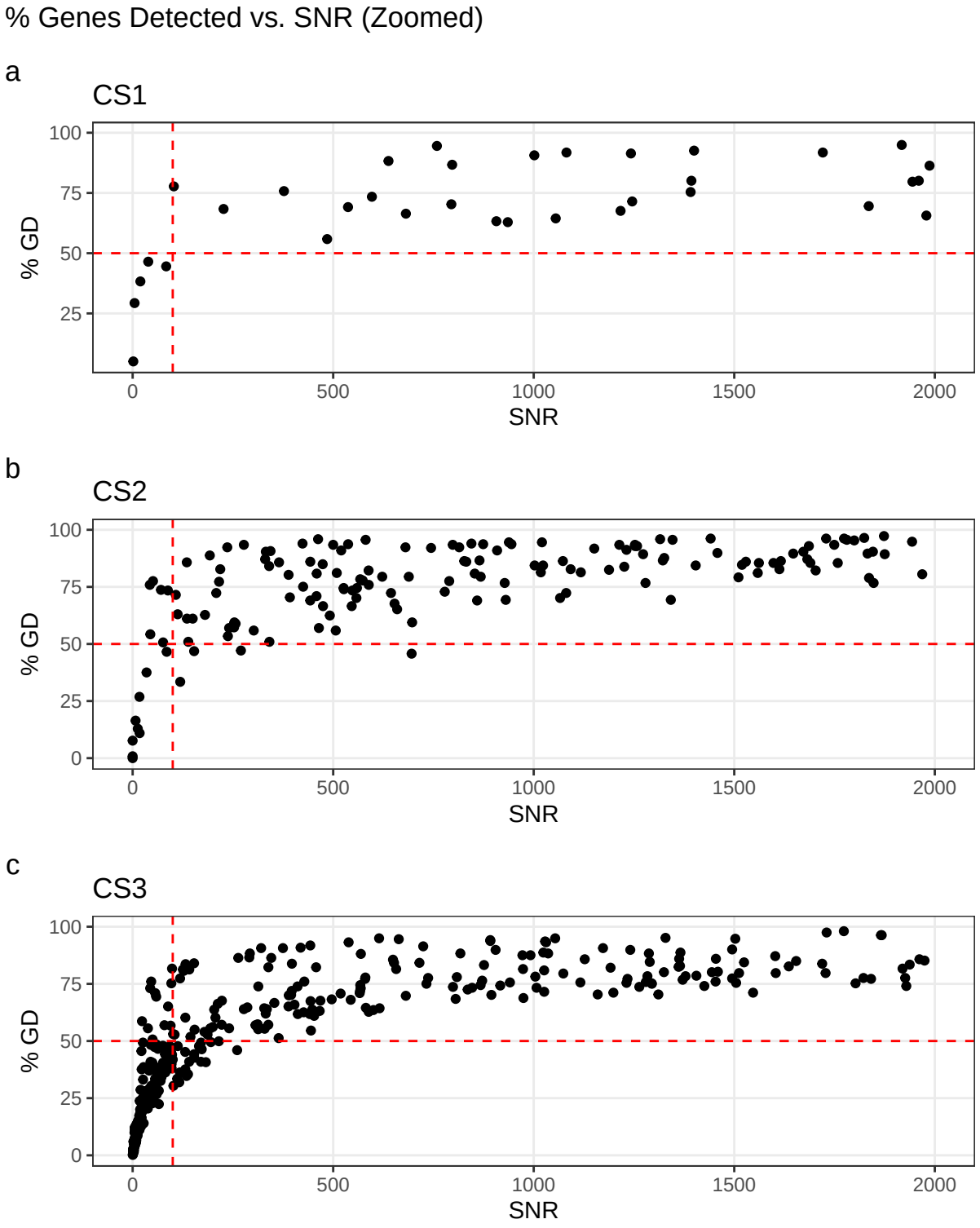

Figure 3.2: % Genes Detected vs. Signal to Noise Ratio (Zoomed)

3.5 Pairwise Gene Expression

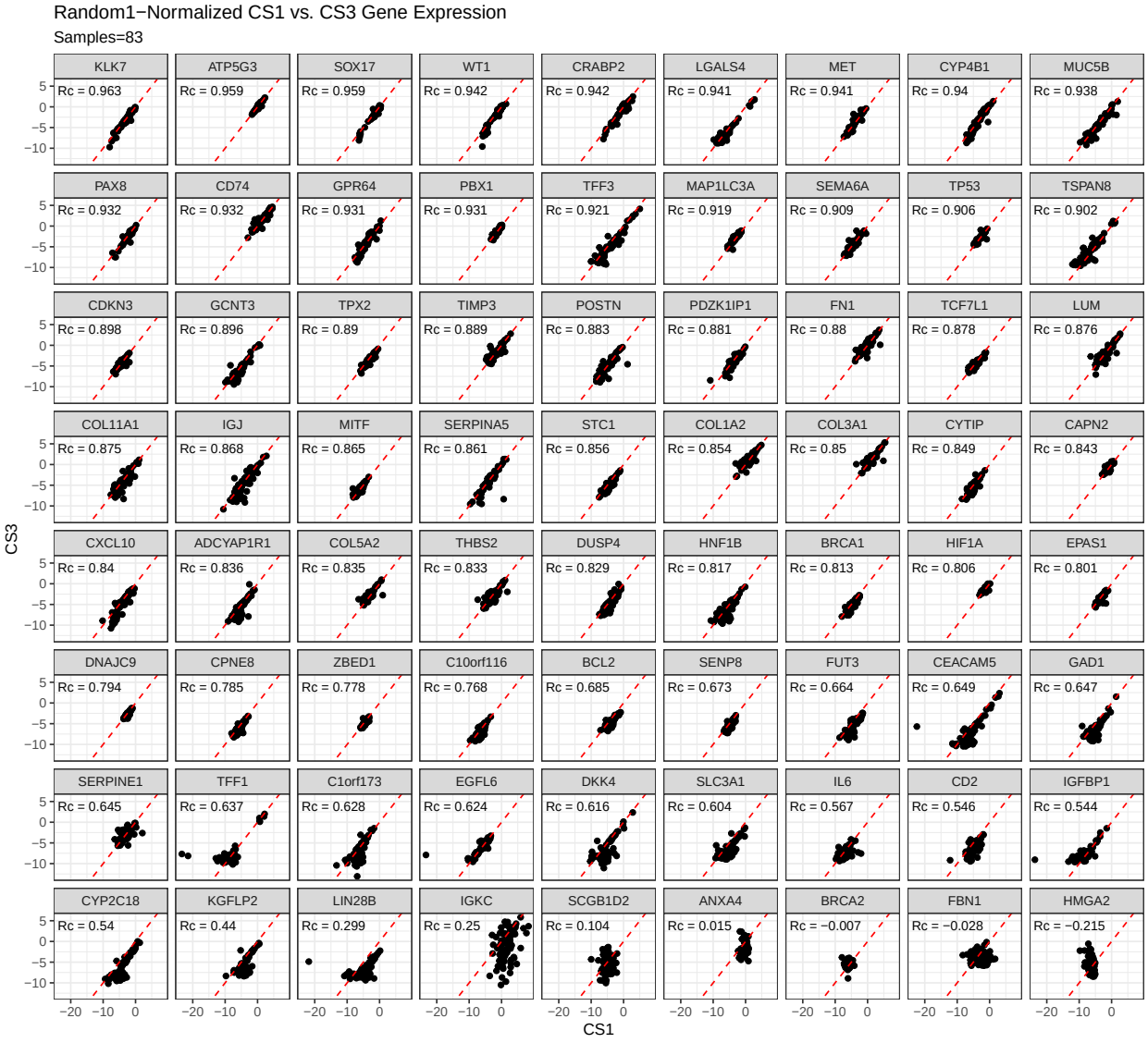

Figure 3.3: Random1-Normalized CS1 vs. CS3 Gene Expression

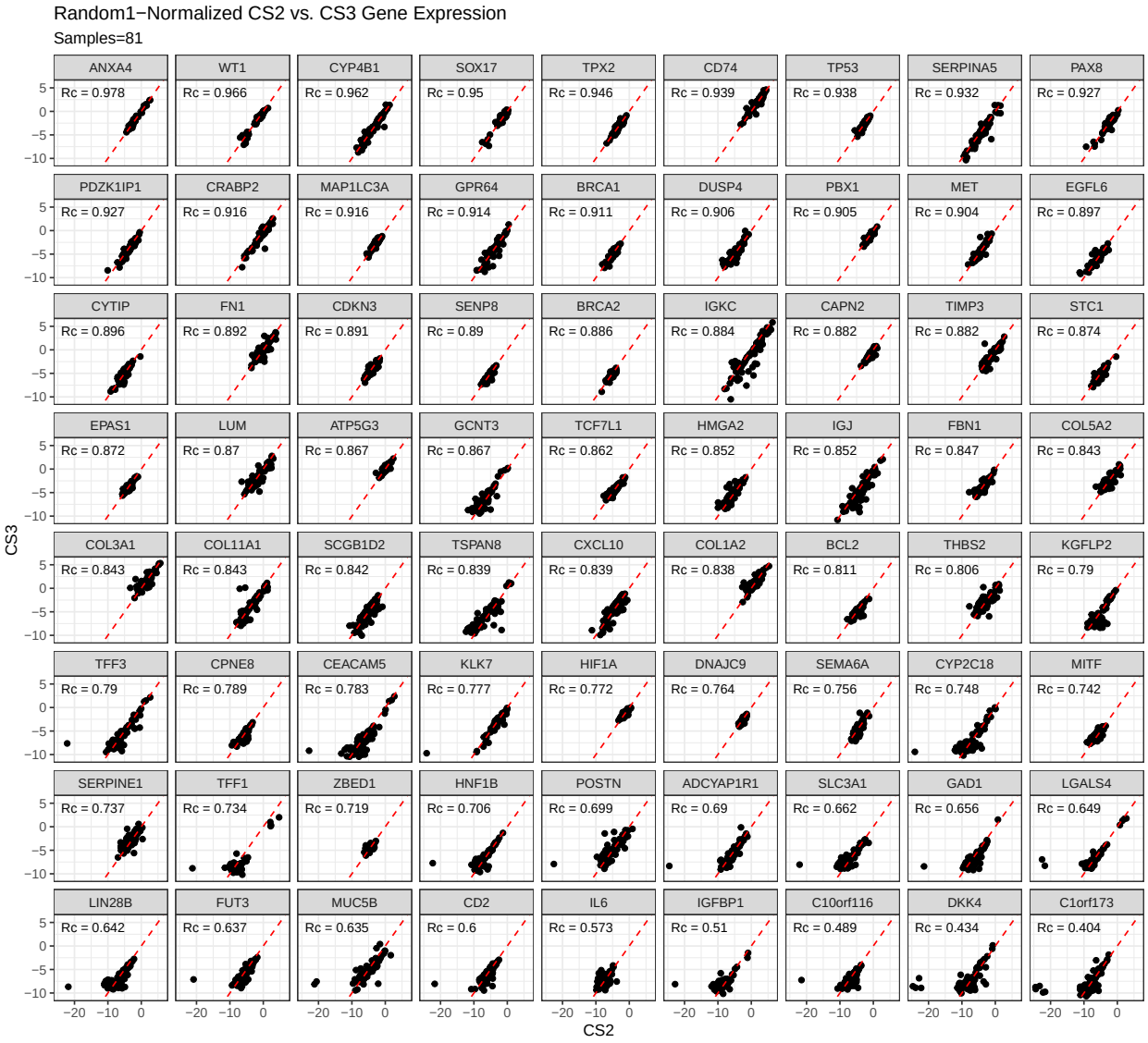

Figure 3.4: Random1-Normalized CS2 vs. CS3 Gene Expression

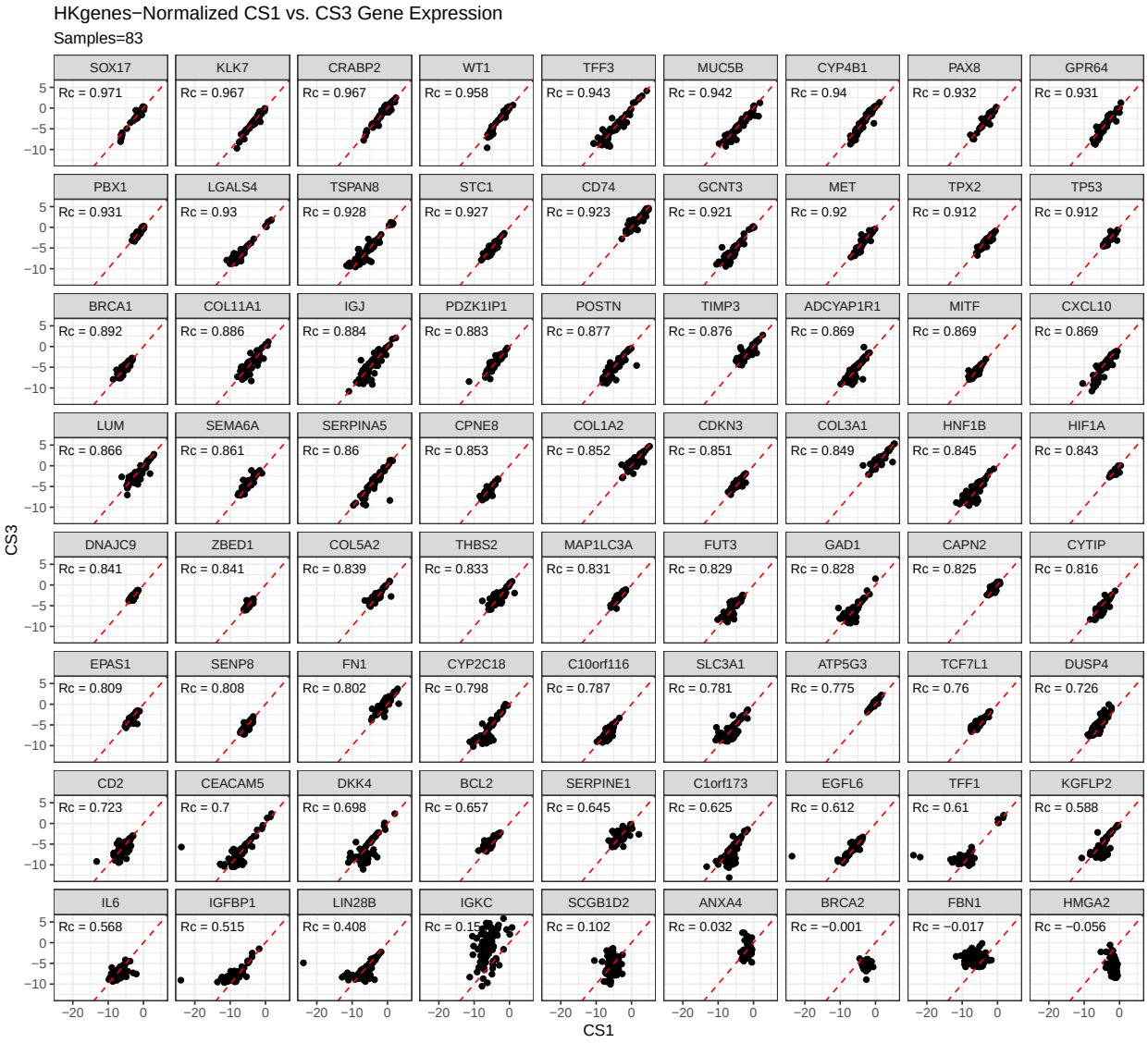

Figure 3.5: HKgenes-Normalized CS1 vs. CS3 Gene Expression

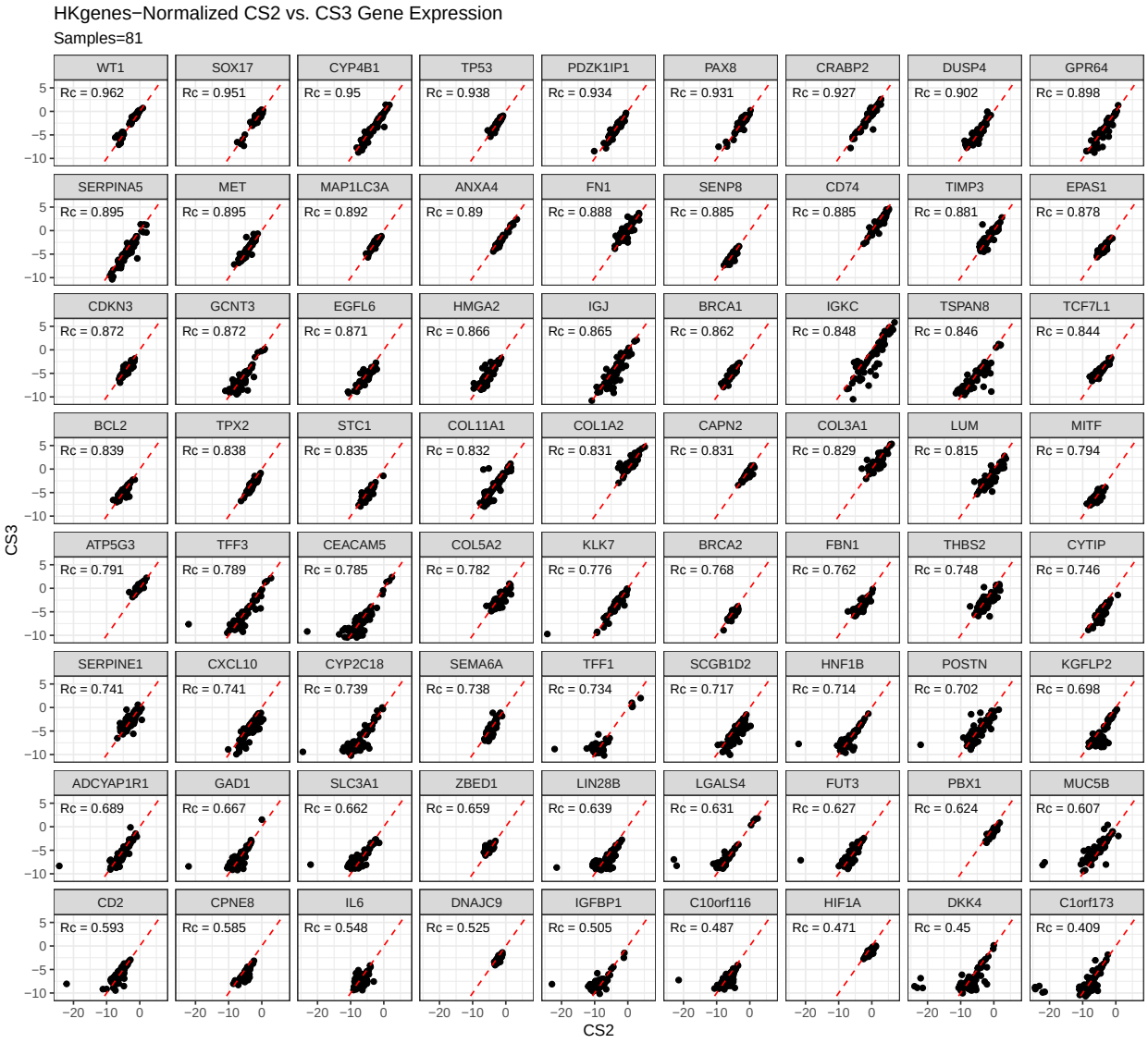

Figure 3.6: HKgenes-Normalized CS2 vs. CS3 Gene Expression

4. Results

We summarize cross-validated training performance of class metrics in the training set. The accuracy, F1-score, kappa, and G-mean are the metrics of interest. Workflows are ordered by their mean estimates across the outer folds of the nested CV for each metric.

4.1 Training Set

4.1.1 Accuracy

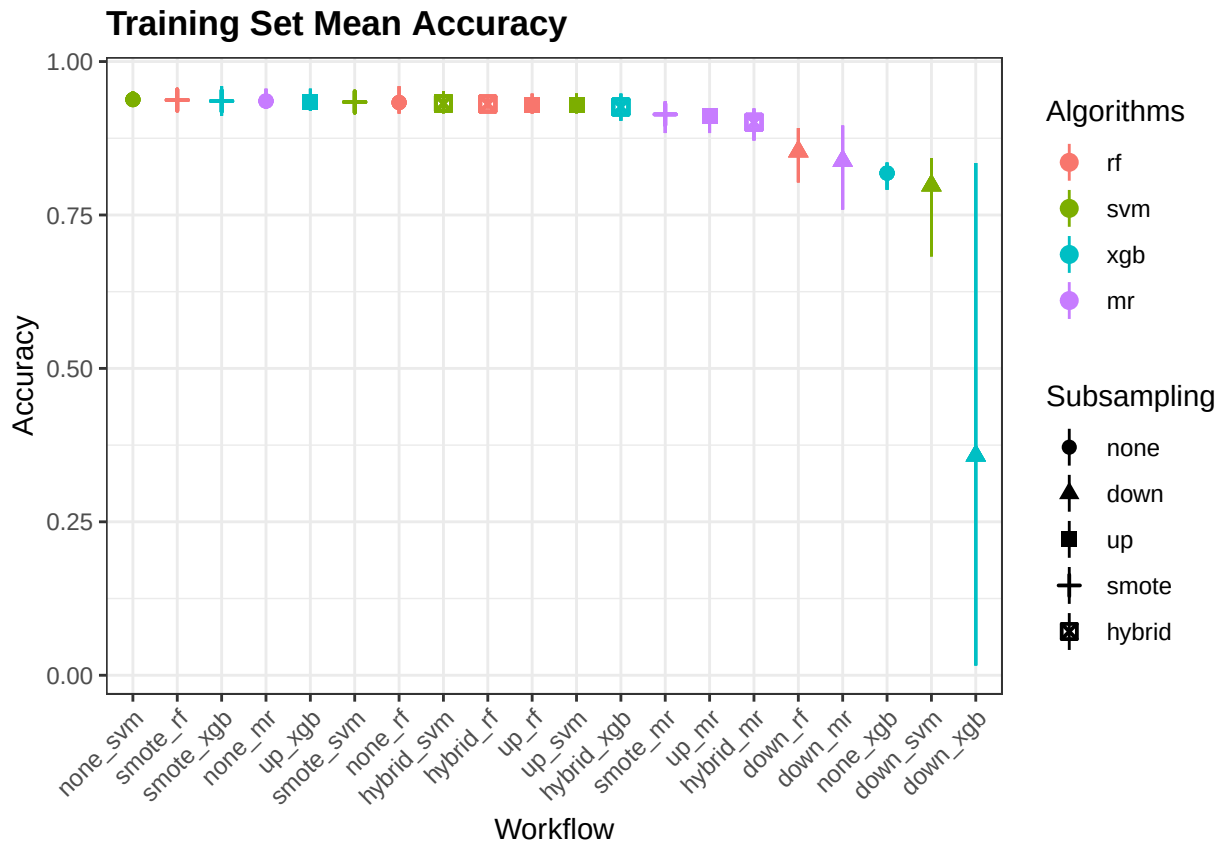

Figure 4.1: Training Set Mean Accuracy

Table 4.1: Training Set Mean Accuracy

Subsampling	Algorithms			
	rf	svm	xgb	mr
none	0.933	0.938	0.818	0.936
down	0.854	0.798	0.358	0.838
up	0.93	0.929	0.934	0.911
smote	0.937	0.934	0.936	0.914
hybrid	0.931	0.932	0.926	0.901

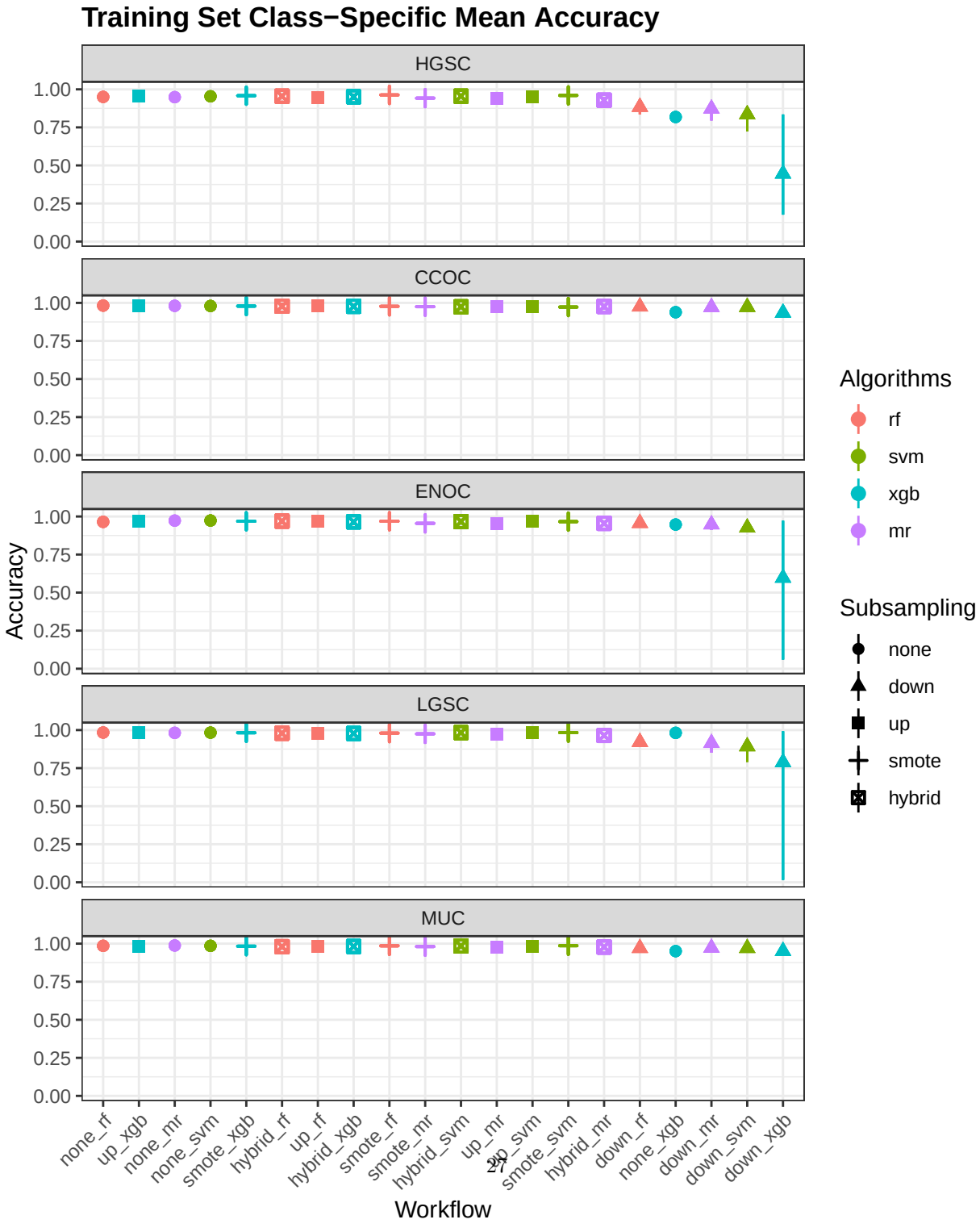

Table 4.2: Training Set Class-Specific Mean Accuracy

		Algorithms			
Subsampling	Histotype	rf	svm	xgb	mr
none	HGSC	0.95	0.954	0.818	0.949
	CCOC	0.982	0.98	0.939	0.981
	ENOC	0.965	0.973	0.947	0.973
	LGSC	0.984	0.983	0.982	0.982
	MUC	0.986	0.986	0.951	0.988
down	HGSC	0.884	0.835	0.446	0.872
	CCOC	0.975	0.973	0.935	0.972
	ENOC	0.957	0.926	0.597	0.947
	LGSC	0.921	0.892	0.788	0.914
	MUC	0.97	0.97	0.951	0.972
up	HGSC	0.948	0.952	0.958	0.94
	CCOC	0.98	0.973	0.981	0.977
	ENOC	0.971	0.967	0.967	0.954
	LGSC	0.977	0.982	0.982	0.973
	MUC	0.984	0.984	0.981	0.977
smote	HGSC	0.963	0.96	0.958	0.942
	CCOC	0.977	0.973	0.979	0.975
	ENOC	0.969	0.966	0.969	0.955
	LGSC	0.98	0.983	0.982	0.975
	MUC	0.986	0.986	0.983	0.981
hybrid	HGSC	0.955	0.955	0.951	0.928
	CCOC	0.978	0.974	0.977	0.977
	ENOC	0.969	0.966	0.965	0.955
	LGSC	0.979	0.983	0.977	0.965
	MUC	0.98	0.985	0.982	0.977

Table 4.3: Training Set Mean Sensitivity

	Algorithms			
Subsampling	rf	svm	xgb	mr
none	0.696	0.695	0.215	0.655
down	0.826	0.806	0.2	0.819
up	0.676	0.745	0.771	0.82
smote	0.714	0.746	0.788	0.818
hybrid	0.727	0.773	0.764	0.816

4.1.2 Sensitivity

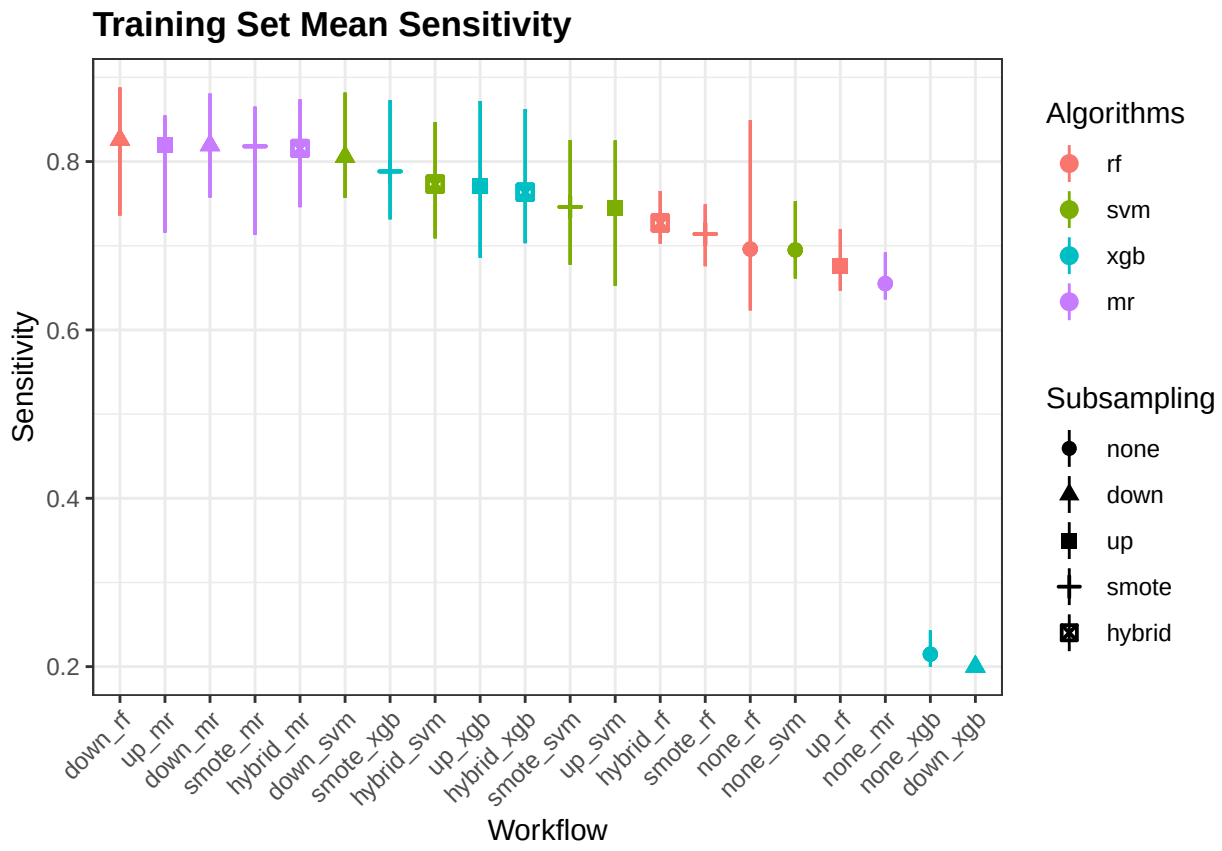

Figure 4.3: Training Set Mean Sensitivity

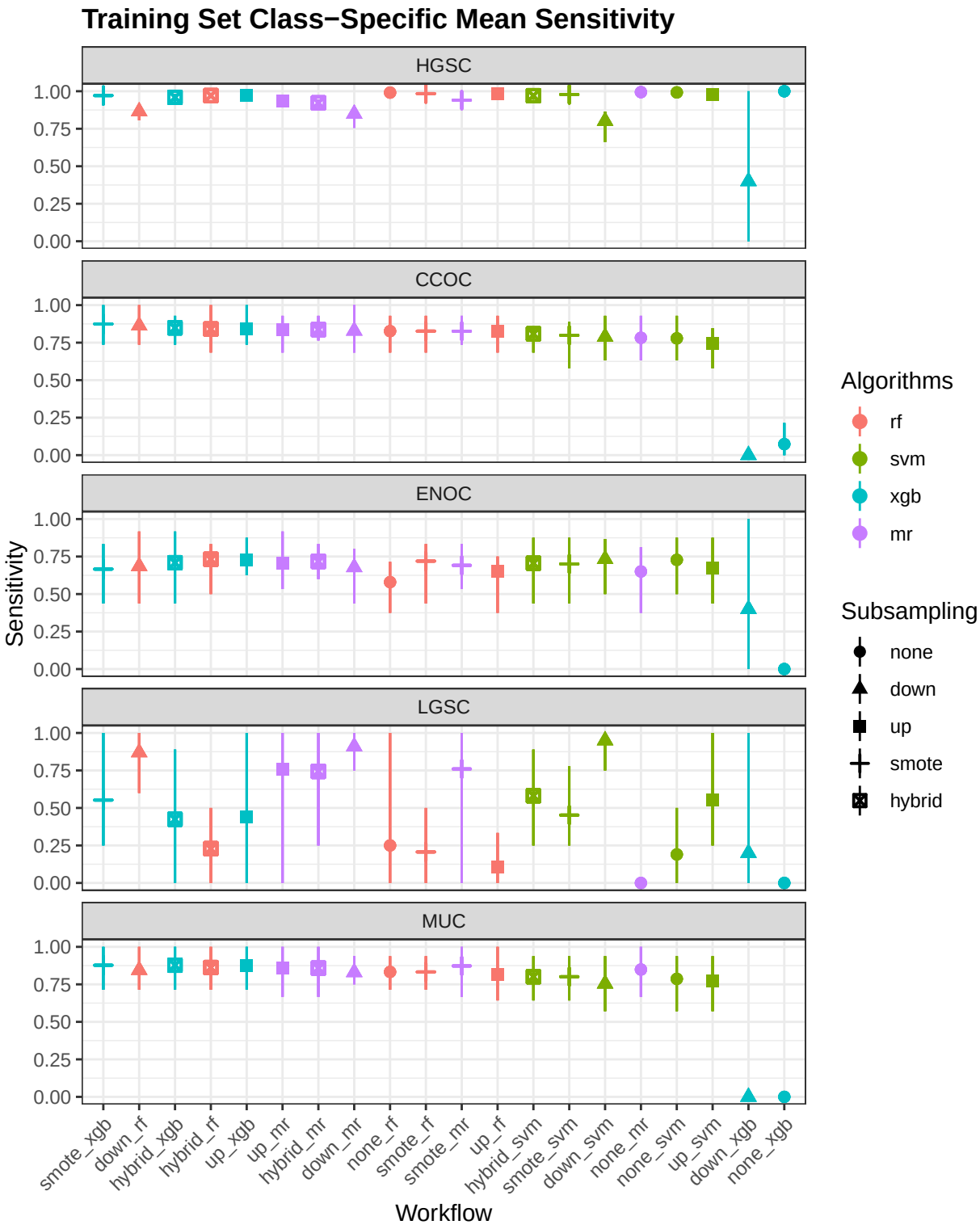

Figure 4.4: Training Set Class-Specific Mean Sensitivity

Table 4.4: Cross-Validated Training Set Class-Specific Mean Sensitivity

		Algorithms			
Subsampling	Histotype	rf	svm	xgb	mr
none	HGSC	0.991	0.992	1	0.994
	CCOC	0.827	0.779	0.074	0.782
	ENOC	0.58	0.728	0	0.65
	LGSC	0.25	0.19	0	0
	MUC	0.833	0.786	0	0.849
down	HGSC	0.866	0.802	0.4	0.849
	CCOC	0.863	0.789	0	0.829
	ENOC	0.685	0.733	0.4	0.678
	LGSC	0.87	0.95	0.2	0.91
	MUC	0.845	0.754	0	0.831
up	HGSC	0.982	0.978	0.97	0.935
	CCOC	0.827	0.747	0.84	0.839
	ENOC	0.652	0.675	0.726	0.708
	LGSC	0.107	0.552	0.44	0.76
	MUC	0.814	0.771	0.877	0.858
smote	HGSC	0.984	0.978	0.971	0.94
	CCOC	0.827	0.799	0.874	0.826
	ENOC	0.72	0.701	0.666	0.691
	LGSC	0.207	0.452	0.552	0.76
	MUC	0.833	0.801	0.877	0.873
hybrid	HGSC	0.971	0.97	0.959	0.923
	CCOC	0.841	0.809	0.848	0.837
	ENOC	0.732	0.705	0.709	0.717
	LGSC	0.229	0.581	0.424	0.743
	MUC	0.862	0.801	0.877	0.858

Table 4.5: Training Set Mean Specificity

Subsampling	Algorithms			
	rf	svm	xgb	mr
none	0.949	0.952	0.804	0.946
down	0.964	0.954	0.8	0.961
up	0.952	0.958	0.971	0.974
smote	0.966	0.966	0.97	0.973
hybrid	0.966	0.967	0.97	0.972

4.1.3 Specificity

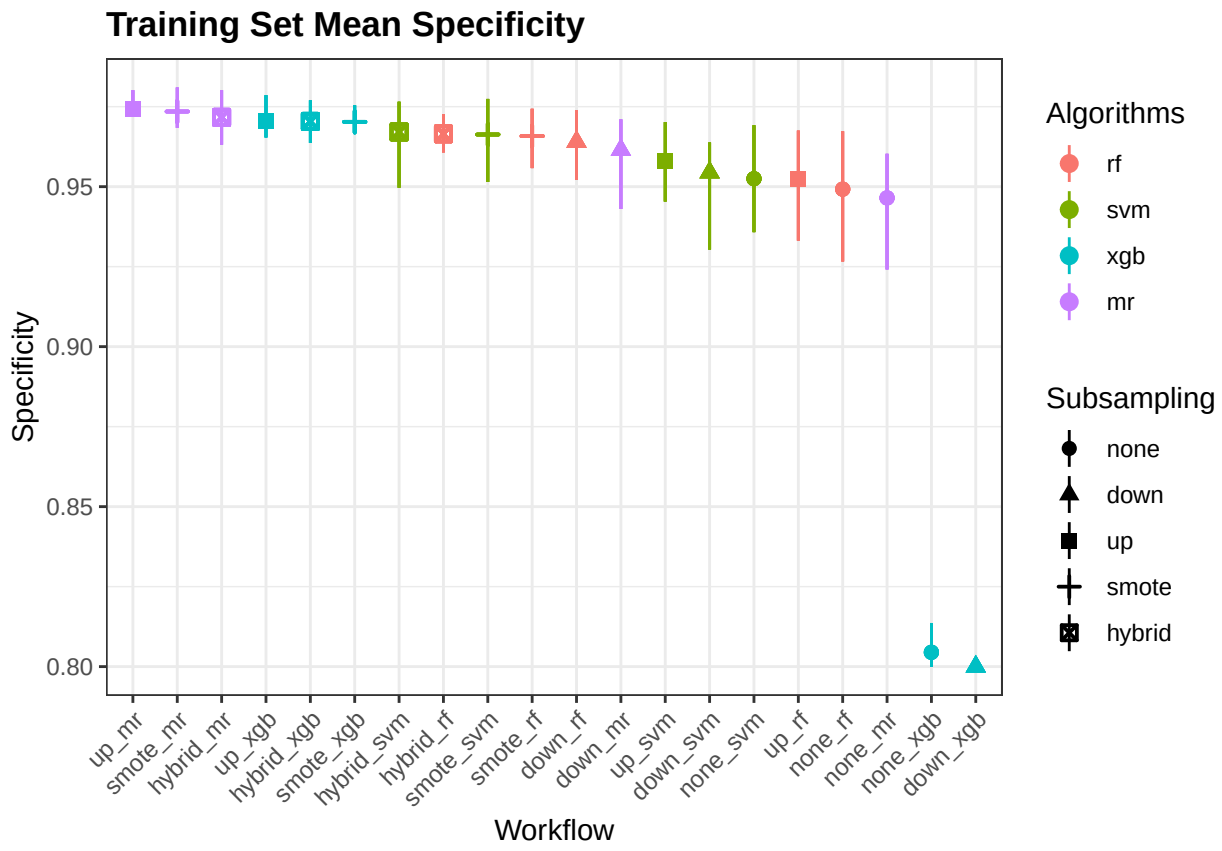

Figure 4.5: Training Set Mean Specificity

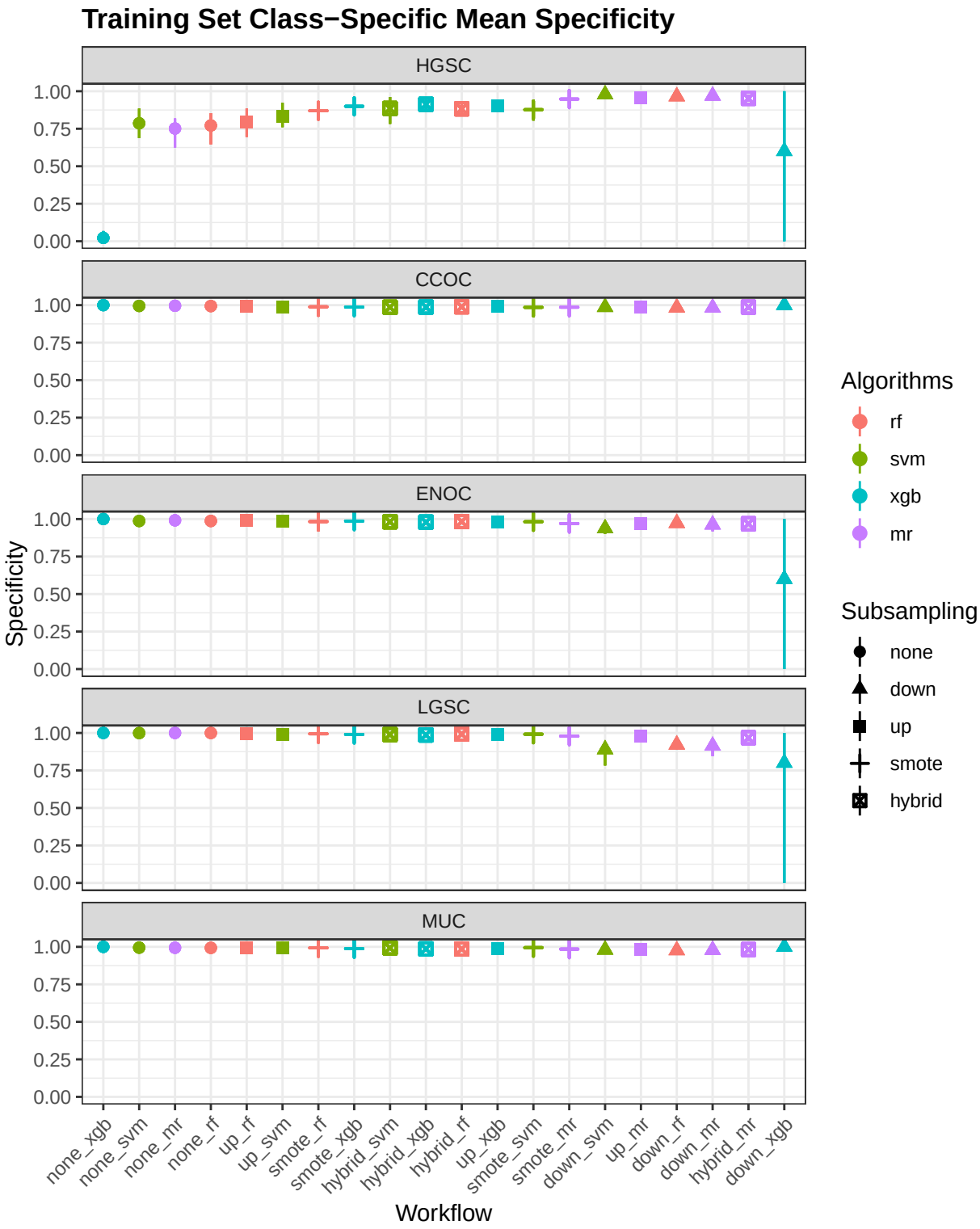

Figure 4.6: Training Set Class-Specific Mean Specificity

Table 4.6: Cross-Validated Training Set Class-Specific Mean Specificity

		Algorithms			
Subsampling	Histotype	rf	svm	xgb	mr
none	HGSC	0.772	0.786	0.022	0.751
	CCOC	0.994	0.995	1	0.996
	ENOC	0.987	0.987	1	0.991
	LGSC	1	0.999	1	1
	MUC	0.993	0.995	1	0.994
down	HGSC	0.965	0.978	0.6	0.97
	CCOC	0.984	0.986	1	0.983
	ENOC	0.973	0.937	0.6	0.963
	LGSC	0.922	0.891	0.8	0.914
	MUC	0.976	0.981	1	0.979
up	HGSC	0.795	0.832	0.904	0.956
	CCOC	0.991	0.99	0.991	0.987
	ENOC	0.99	0.984	0.98	0.969
	LGSC	0.993	0.99	0.992	0.977
	MUC	0.992	0.994	0.986	0.982
smote	HGSC	0.87	0.877	0.899	0.948
	CCOC	0.989	0.985	0.987	0.986
	ENOC	0.983	0.982	0.986	0.97
	LGSC	0.994	0.992	0.99	0.979
	MUC	0.993	0.995	0.988	0.985
hybrid	HGSC	0.882	0.885	0.913	0.953
	CCOC	0.989	0.986	0.987	0.987
	ENOC	0.983	0.981	0.98	0.968
	LGSC	0.993	0.989	0.986	0.968
	MUC	0.986	0.993	0.986	0.982

Table 4.7: Training Set Mean F1-Score

Subsampling	Algorithms			
	rf	svm	xgb	mr
none	0.819	0.799	0.78	0.851
down	0.674	0.622	0.415	0.659
up	0.7	0.749	0.752	0.723
smote	0.724	0.747	0.771	0.731
hybrid	0.717	0.76	0.731	0.716

4.1.4 F1-Score

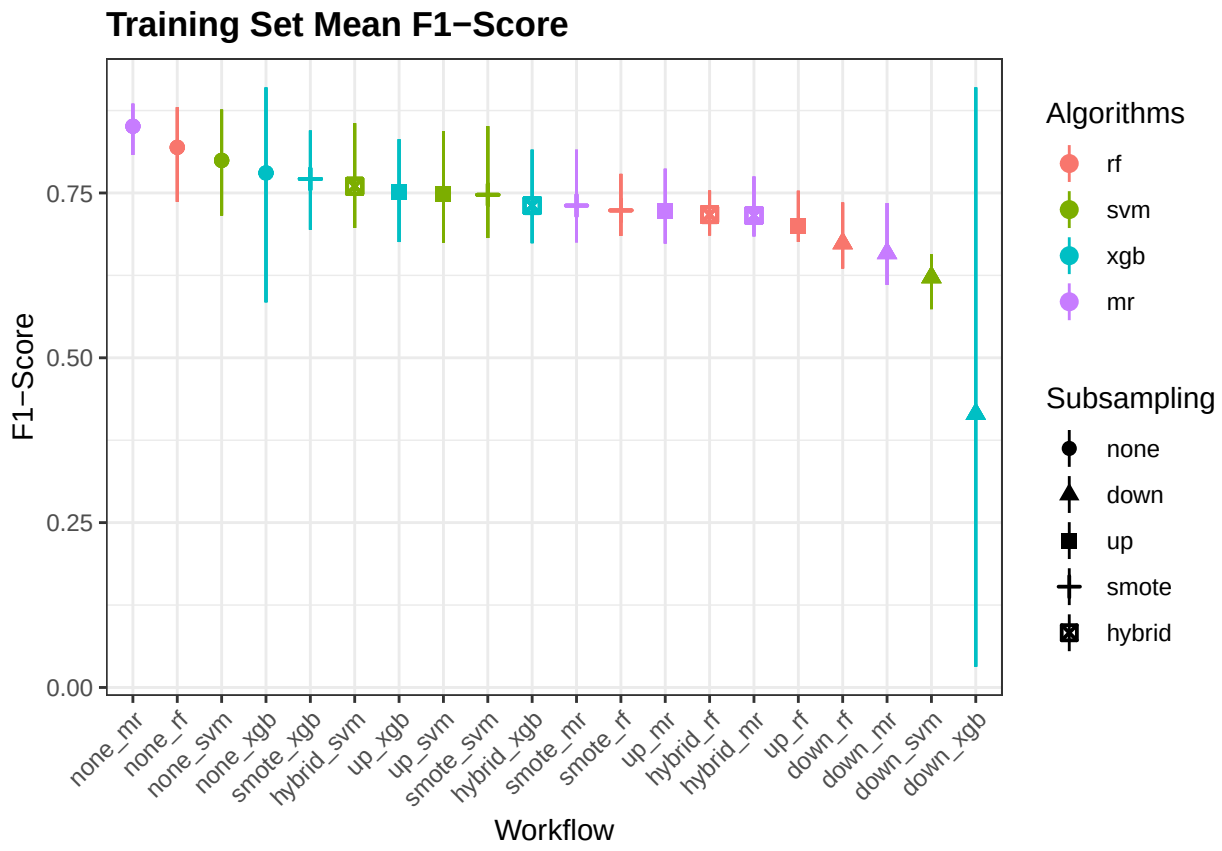

Figure 4.7: Training Set Mean F1-Score

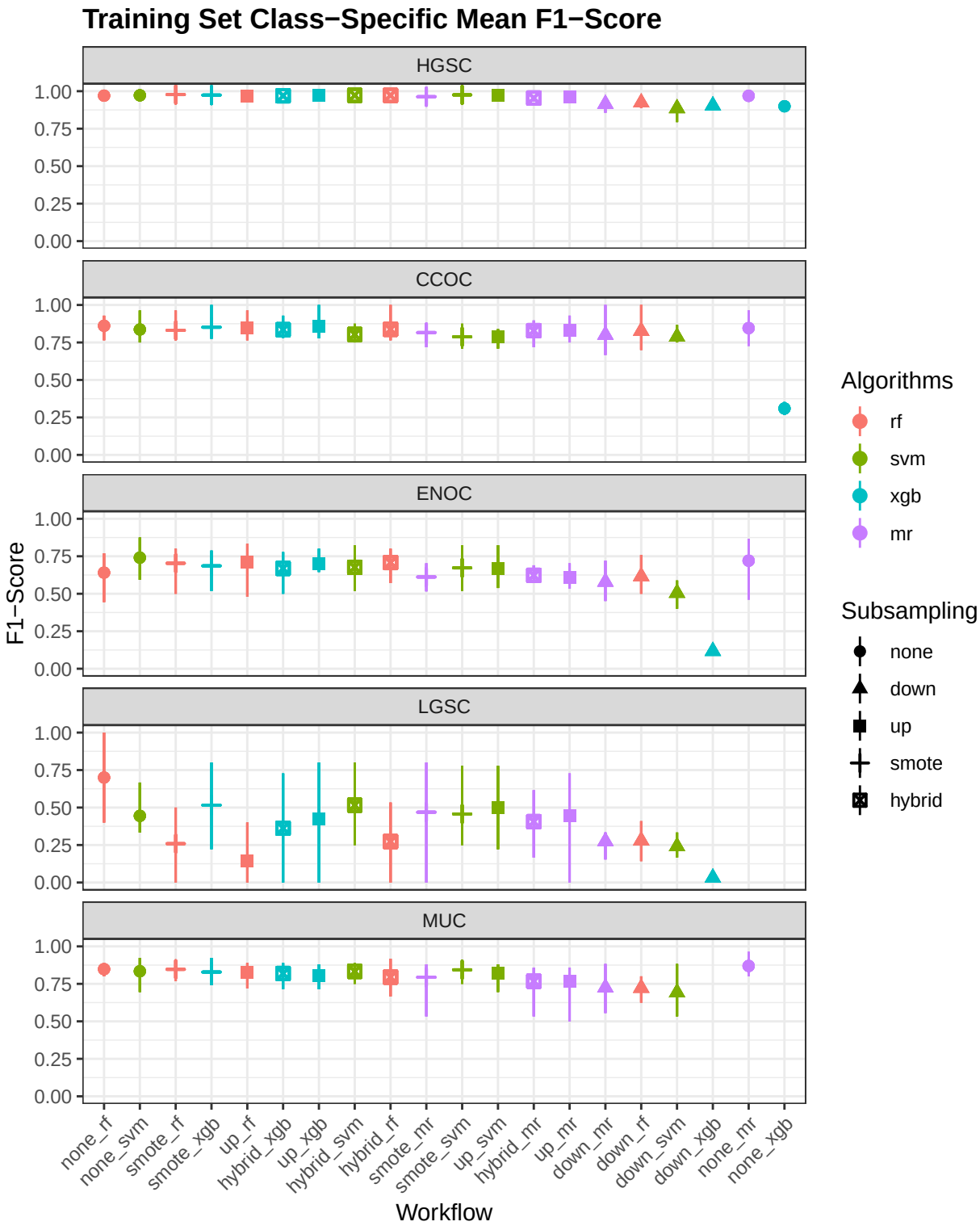

Figure 4.8: Training Set Class-Specific Mean F1-Score

Table 4.8: Cross-Validated Training Set Class-Specific Mean F1-Score

		Algorithms			
Subsampling	Histotype	rf	svm	xgb	mr
none	HGSC	0.97	0.972	0.9	0.969
	CCOC	0.861	0.836	0.31	0.846
	ENOC	0.641	0.741	NaN	0.72
	LGSC	0.7	0.444	NaN	NaN
	MUC	0.848	0.834	NaN	0.869
down	HGSC	0.924	0.885	0.904	0.914
	CCOC	0.827	0.788	NaN	0.8
	ENOC	0.616	0.505	0.117	0.579
	LGSC	0.279	0.24	0.032	0.275
	MUC	0.722	0.692	NaN	0.725
up	HGSC	0.968	0.971	0.974	0.962
	CCOC	0.847	0.786	0.856	0.829
	ENOC	0.71	0.669	0.699	0.608
	LGSC	0.147	0.498	0.425	0.448
	MUC	0.828	0.82	0.807	0.766
smote	HGSC	0.977	0.975	0.974	0.963
	CCOC	0.831	0.788	0.852	0.816
	ENOC	0.703	0.673	0.686	0.612
	LGSC	0.259	0.457	0.516	0.469
	MUC	0.847	0.844	0.829	0.794
hybrid	HGSC	0.972	0.972	0.969	0.954
	CCOC	0.838	0.803	0.835	0.829
	ENOC	0.708	0.676	0.669	0.623
	LGSC	0.273	0.516	0.362	0.406
	MUC	0.795	0.833	0.819	0.767

Table 4.9: Training Set Mean Balanced Accuracy

Subsampling	Algorithms			
	rf	svm	xgb	mr
none	0.823	0.824	0.51	0.801
down	0.895	0.88	0.5	0.89
up	0.814	0.851	0.871	0.897
smote	0.84	0.856	0.879	0.896
hybrid	0.847	0.87	0.867	0.894

4.1.5 Balanced Accuracy

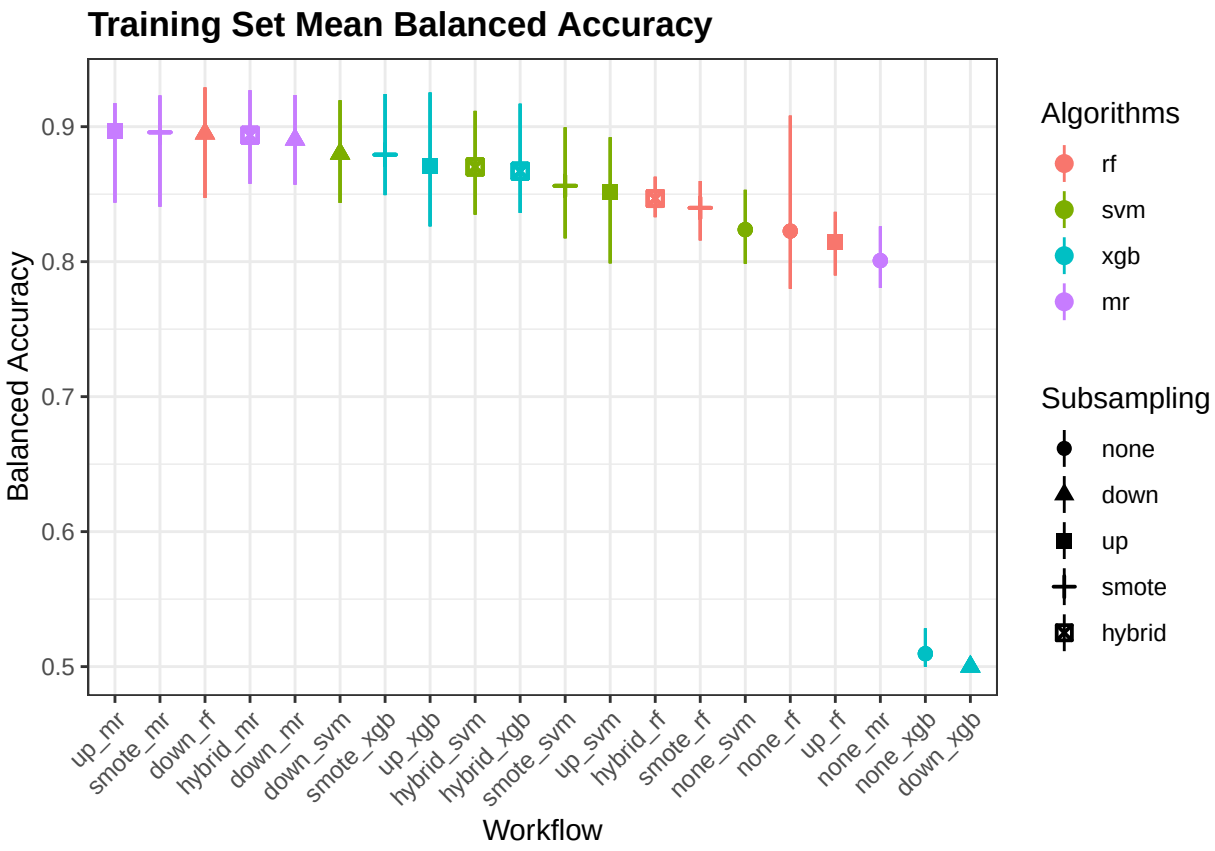

Figure 4.9: Training Set Mean Balanced Accuracy

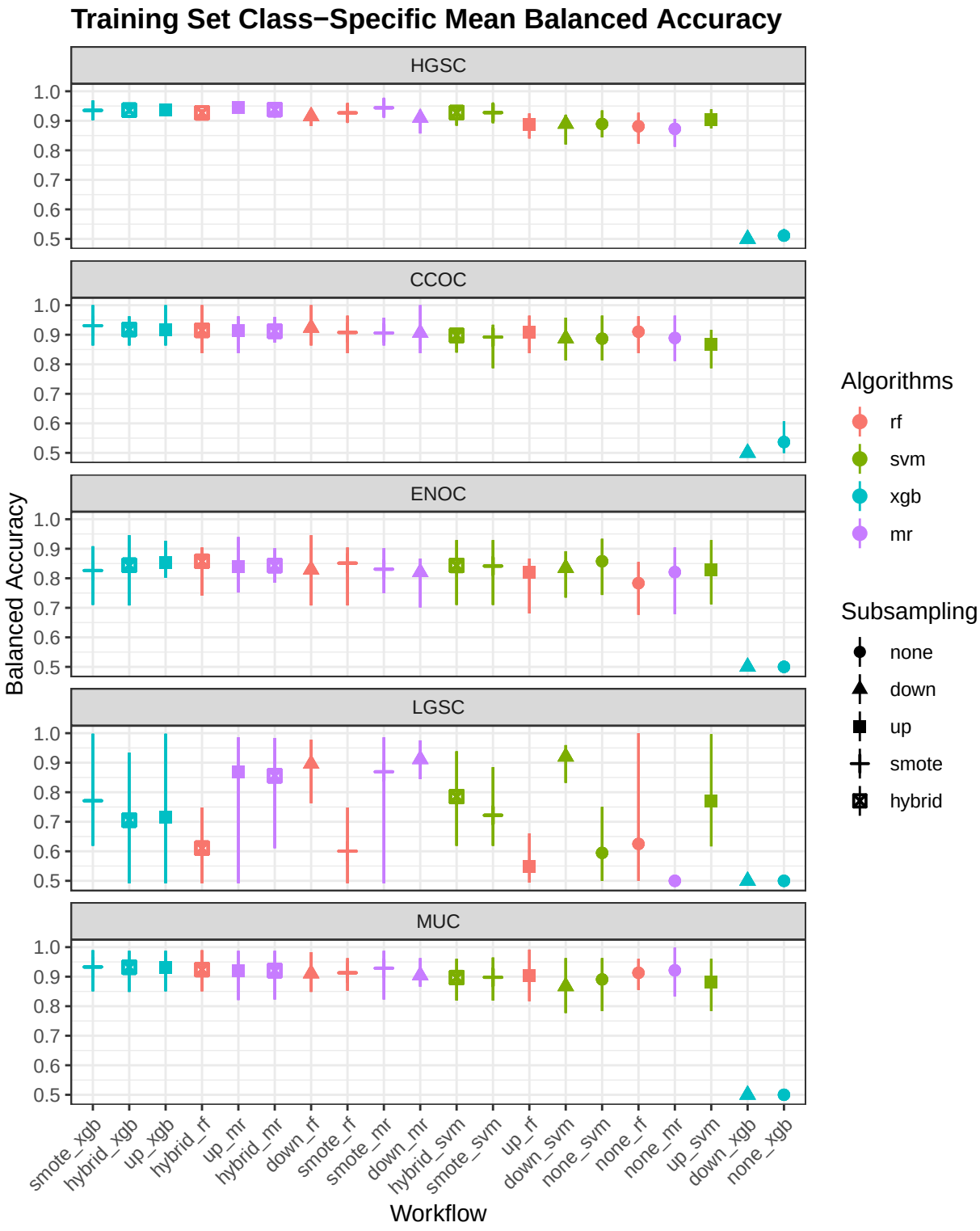

Figure 4.10: Training Set Class-Specific Mean Balanced Accuracy

Table 4.10: Training Set Class-Specific Mean Balanced Accuracy

		Algorithms			
Subsampling	Histotype	rf	svm	xgb	mr
none	HGSC	0.881	0.889	0.511	0.873
	CCOC	0.91	0.887	0.537	0.889
	ENOC	0.783	0.858	0.5	0.821
	LGSC	0.625	0.595	0.5	0.5
	MUC	0.913	0.891	0.5	0.921
down	HGSC	0.915	0.89	0.5	0.909
	CCOC	0.923	0.888	0.5	0.906
	ENOC	0.829	0.835	0.5	0.82
	LGSC	0.896	0.92	0.5	0.912
	MUC	0.911	0.867	0.5	0.905
up	HGSC	0.889	0.905	0.937	0.946
	CCOC	0.909	0.869	0.916	0.913
	ENOC	0.821	0.83	0.853	0.838
	LGSC	0.55	0.771	0.716	0.869
	MUC	0.903	0.883	0.932	0.92
smote	HGSC	0.927	0.928	0.935	0.944
	CCOC	0.908	0.892	0.931	0.906
	ENOC	0.851	0.841	0.826	0.831
	LGSC	0.6	0.722	0.771	0.869
	MUC	0.913	0.898	0.933	0.929
hybrid	HGSC	0.927	0.928	0.936	0.938
	CCOC	0.915	0.898	0.918	0.912
	ENOC	0.858	0.843	0.844	0.843
	LGSC	0.611	0.785	0.705	0.856
	MUC	0.924	0.897	0.932	0.92

Table 4.11: Training Set Mean Kappa

	Algorithms			
Subsampling	rf	svm	xgb	mr
none	0.776	0.793	0.038	0.782
down	0.642	0.558	0	0.618
up	0.772	0.775	0.802	0.754
smote	0.803	0.794	0.808	0.761
hybrid	0.788	0.792	0.784	0.734

4.1.6 Kappa

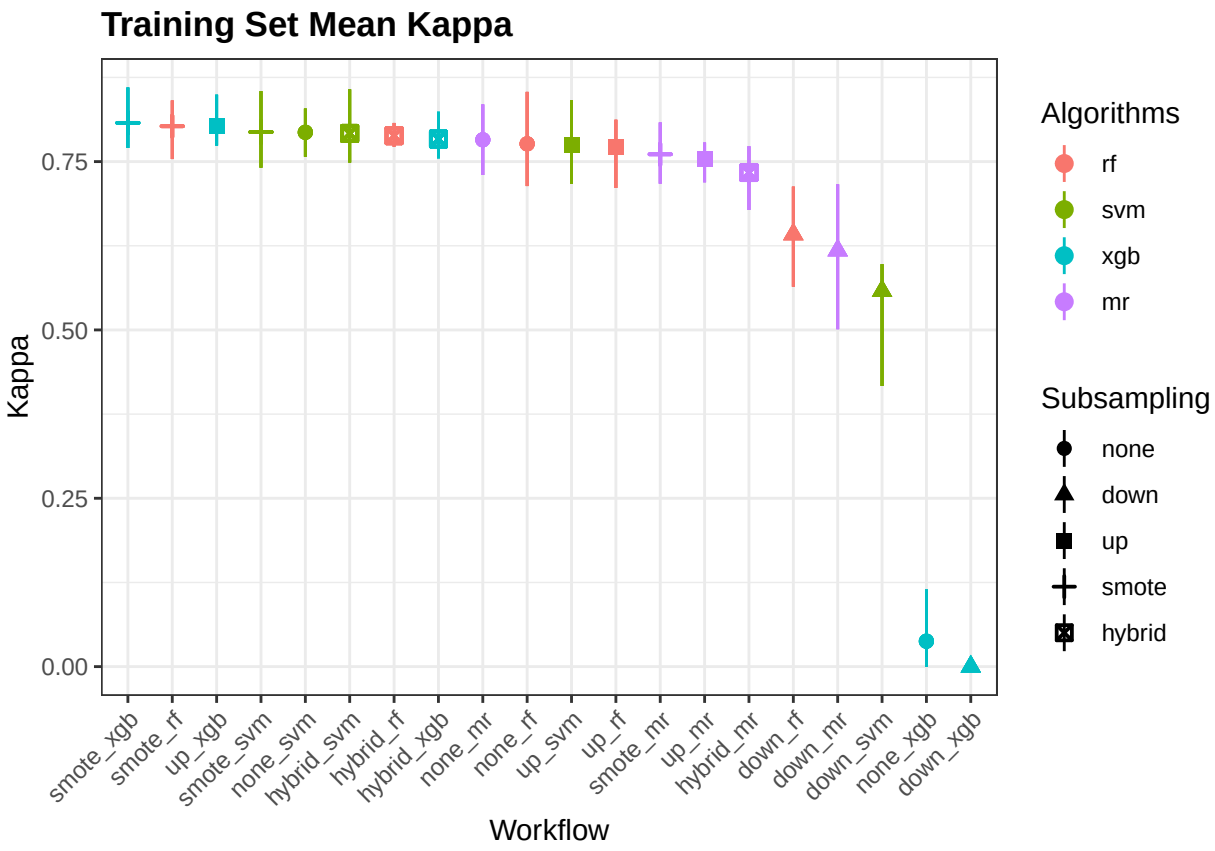

Figure 4.11: Training Set Mean Kappa

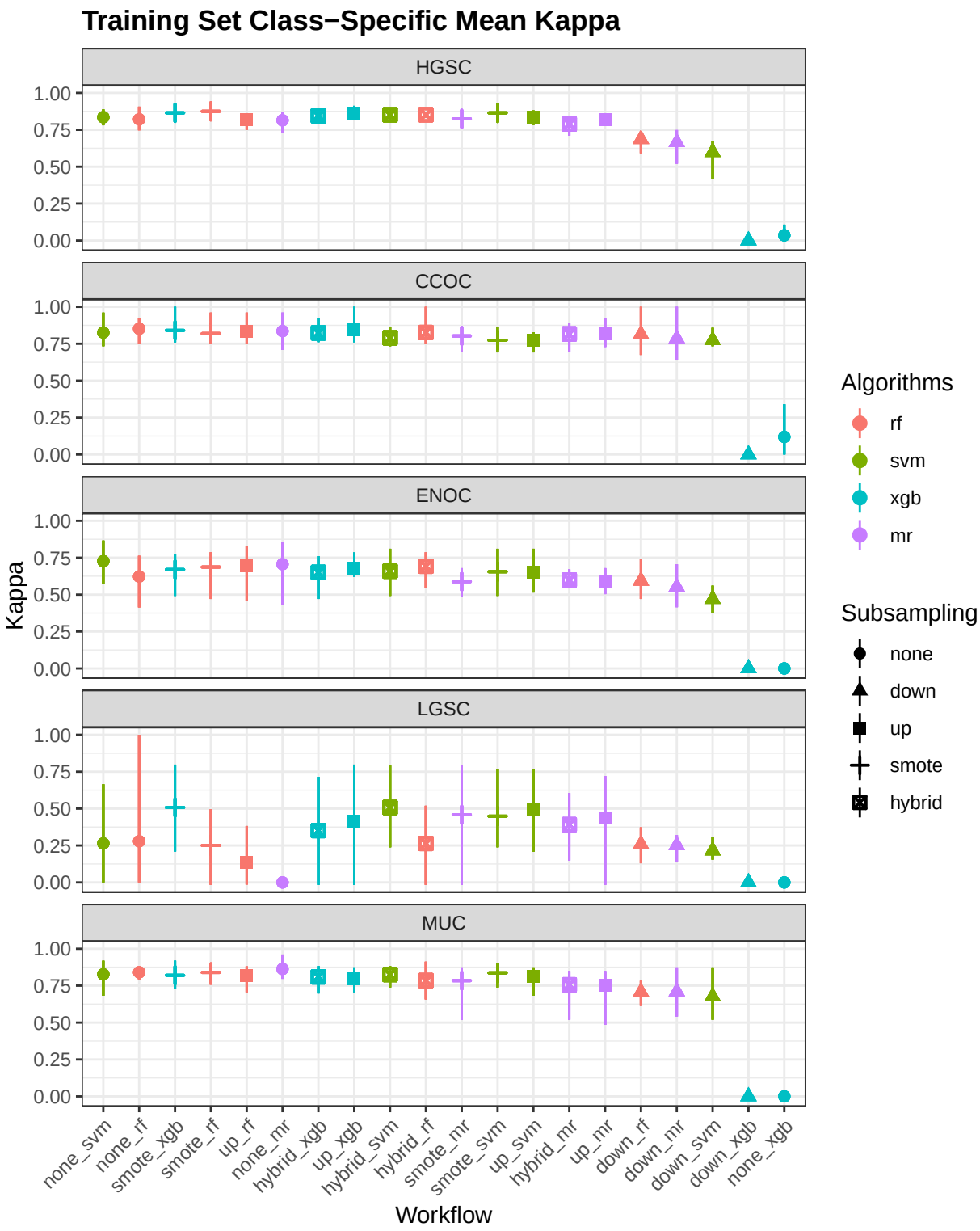

Figure 4.12: Training Set Class-Specific Mean Kappa

Table 4.12: Training Set Class-Specific Mean Kappa

		Algorithms			
Subsampling	Histotype	rf	svm	xgb	mr
none	HGSC	0.821	0.835	0.035	0.814
	CCOC	0.852	0.826	0.119	0.836
	ENOC	0.622	0.727	0	0.706
	LGSC	0.279	0.264	0	0
	MUC	0.84	0.827	0	0.863
down	HGSC	0.686	0.598	0	0.665
	CCOC	0.813	0.773	0	0.785
	ENOC	0.594	0.47	0	0.552
	LGSC	0.258	0.216	0	0.252
	MUC	0.707	0.676	0	0.711
up	HGSC	0.818	0.834	0.864	0.819
	CCOC	0.836	0.773	0.846	0.817
	ENOC	0.696	0.652	0.681	0.584
	LGSC	0.137	0.489	0.416	0.437
	MUC	0.82	0.812	0.797	0.755
smote	HGSC	0.875	0.865	0.865	0.824
	CCOC	0.819	0.773	0.84	0.803
	ENOC	0.687	0.655	0.67	0.588
	LGSC	0.25	0.449	0.508	0.458
	MUC	0.839	0.836	0.82	0.784
hybrid	HGSC	0.852	0.851	0.845	0.789
	CCOC	0.826	0.79	0.823	0.816
	ENOC	0.692	0.659	0.651	0.599
	LGSC	0.264	0.507	0.351	0.391
	MUC	0.784	0.825	0.809	0.756

Table 4.13: Training Set Mean G-mean

Subsampling	Algorithms			
	rf	svm	xgb	mr
none	0.293	0.389	0	0
down	0.815	0.795	0	0.81
up	0.246	0.71	0.602	0.666
smote	0.403	0.709	0.749	0.665
hybrid	0.415	0.747	0.598	0.794

4.1.7 G-mean

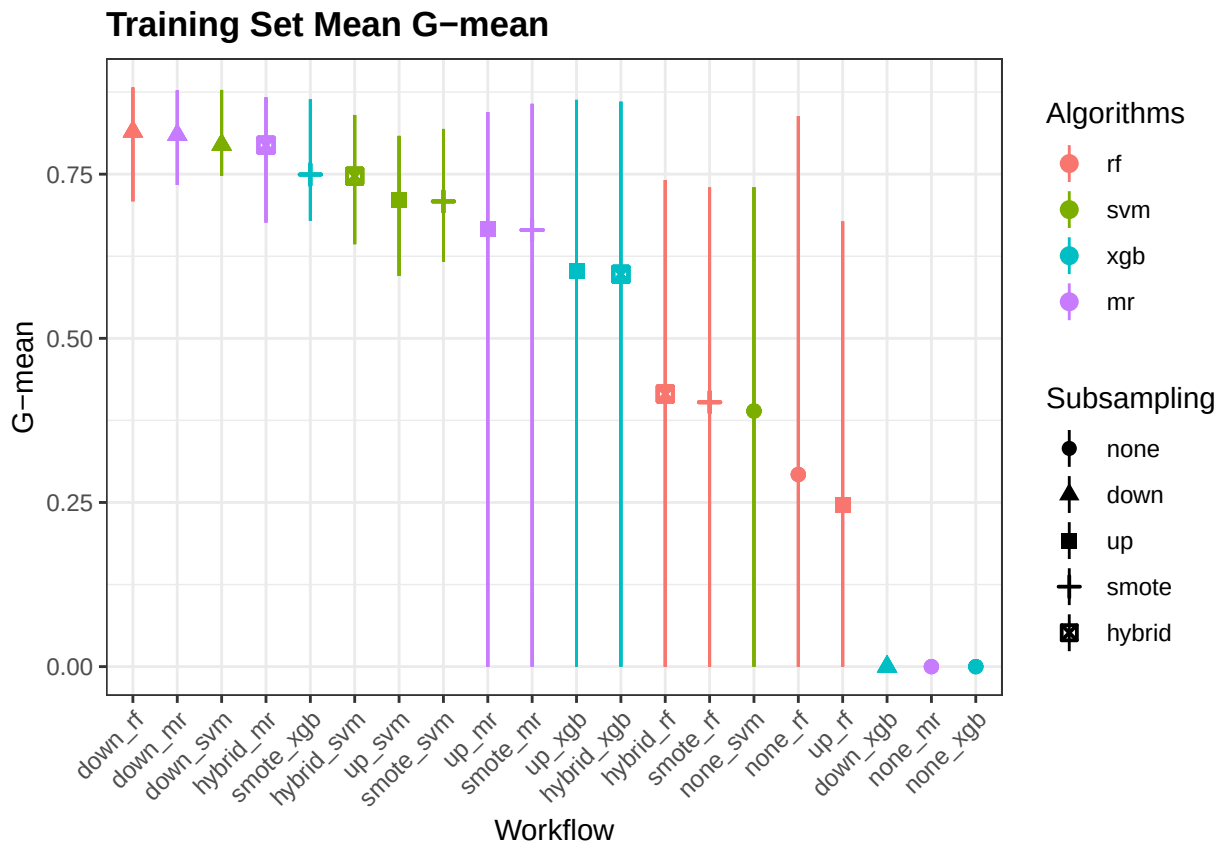

Figure 4.13: Training Set Mean G-mean

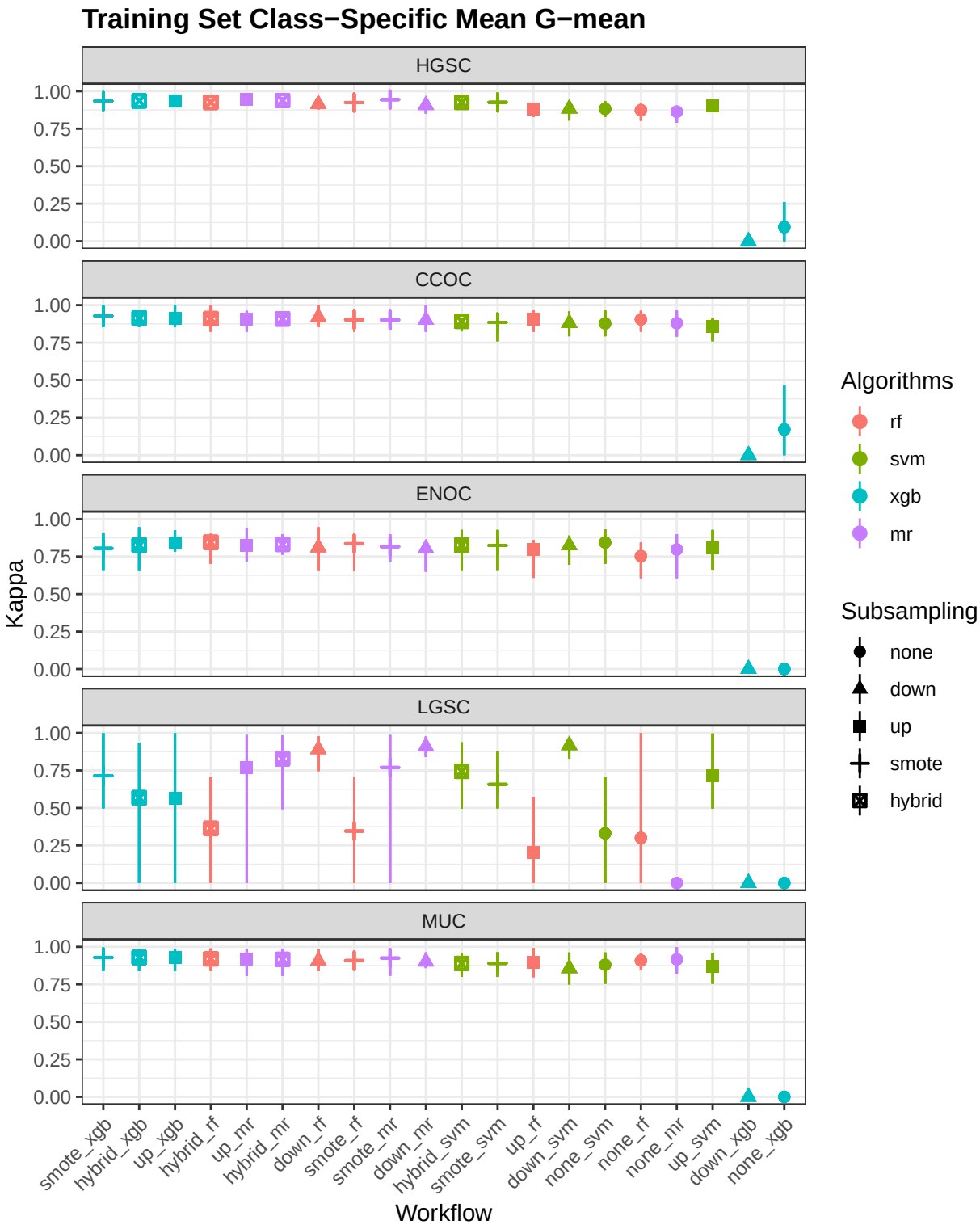

Figure 4.14: Training Set Class-Specific Mean G-mean

Table 4.14: Training Set Class-Specific Mean G-mean

		Algorithms			
Subsampling	Histotype	rf	svm	xgb	mr
none	HGSC	0.873	0.882	0.094	0.863
	CCOC	0.905	0.878	0.171	0.88
	ENOC	0.753	0.844	0	0.797
	LGSC	0.3	0.331	0	0
	MUC	0.909	0.88	0	0.916
down	HGSC	0.914	0.884	0	0.907
	CCOC	0.92	0.88	0	0.9
	ENOC	0.811	0.824	0	0.803
	LGSC	0.891	0.918	0	0.909
	MUC	0.907	0.857	0	0.901
up	HGSC	0.883	0.901	0.936	0.946
	CCOC	0.904	0.858	0.911	0.908
	ENOC	0.798	0.81	0.842	0.823
	LGSC	0.204	0.715	0.566	0.769
	MUC	0.897	0.872	0.928	0.916
smote	HGSC	0.925	0.926	0.934	0.944
	CCOC	0.903	0.885	0.928	0.902
	ENOC	0.836	0.824	0.805	0.816
	LGSC	0.346	0.657	0.715	0.77
	MUC	0.909	0.89	0.93	0.925
hybrid	HGSC	0.926	0.926	0.936	0.937
	CCOC	0.91	0.892	0.914	0.908
	ENOC	0.845	0.826	0.826	0.831
	LGSC	0.363	0.744	0.568	0.828
	MUC	0.92	0.889	0.929	0.916

4.2 Rank Aggregation

Show 50 entries Search:

F1-Score Summary by Workflow and Class

Workflow	HGSC	CCOC	LGSC	ENOC	MUC	Rank
All	All	All	All	All	All	All
two_step	0.969	0.87	0.907	0.793	0.859	1
sequential	0.969	0.87	0.893	0.907	0.959	2
none_rf	0.97	0.861	0.7	0.641	0.848	3
smote_xgb	0.974	0.852	0.516	0.686	0.829	4
smote_rf	0.977	0.831	0.259	0.703	0.847	5
none_svm	0.972	0.836	0.444	0.741	0.834	6
hybrid_svm	0.972	0.803	0.516	0.676	0.833	7
hybrid_rf	0.972	0.838	0.273	0.708	0.795	8
smote_svm	0.975	0.788	0.457	0.673	0.844	9
up_xgb	0.974	0.856	0.425	0.699	0.807	10
up_svm	0.971	0.786	0.498	0.669	0.82	11
hybrid_xgb	0.969	0.835	0.362	0.669	0.819	12
up_rf	0.968	0.847	0.147	0.71	0.828	13
smote_mr	0.963	0.816	0.469	0.612	0.794	14
hybrid_mr	0.954	0.829	0.406	0.623	0.767	15
up_mr	0.962	0.829	0.448	0.608	0.766	16
down_rf	0.924	0.827	0.279	0.616	0.722	17
down_mr	0.914	0.8	0.275	0.579	0.725	18
down_svm	0.885	0.788	0.24	0.505	0.692	19

Showing 1 to 19 of 19 entries Previous 1 Next

The 19 workflows are ordered in the table by their aggregated ranks using the Genetic Algorithm. We see that the best performing methods involve the sequential and two-step algorithms.

4.2.1 Top Workflows

We look at the per-class evaluation metrics of the top 4 workflows.

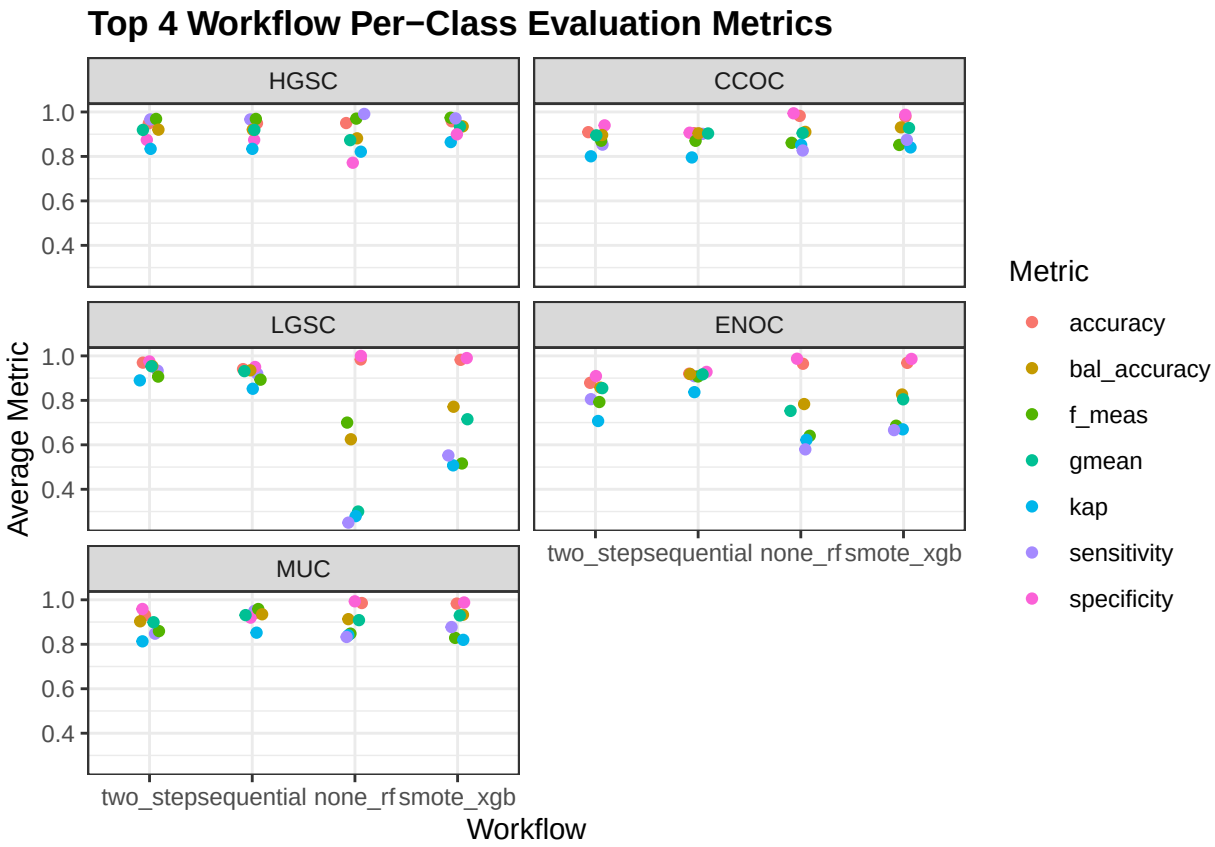

Figure 4.15: Top 4 Workflow Per-Class Evaluation Metrics

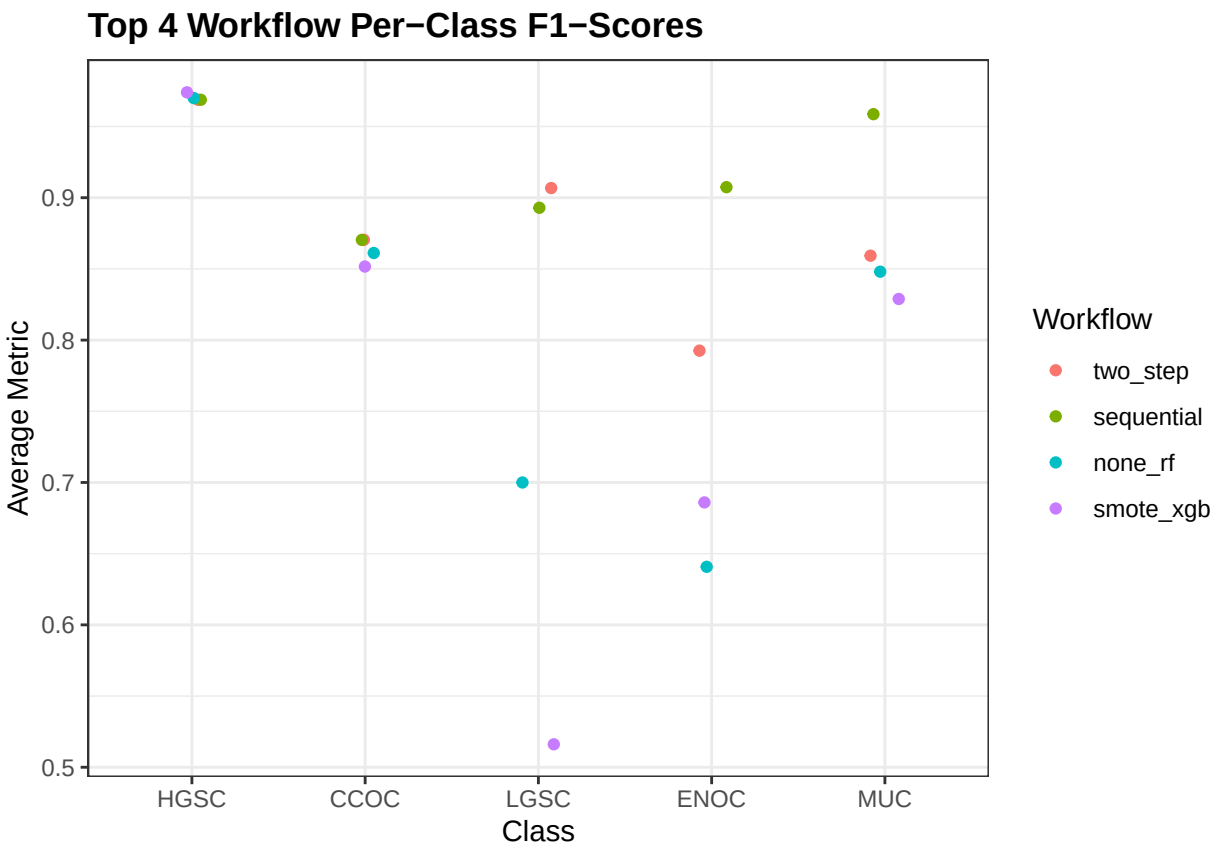

Figure 4.16: Top 4 Workflow Per-Class F1-Scores

Misclassified cases from a previous step of the sequence of classifiers are not included in subsequent steps of the training set CV folds. Thus, we cannot piece together the test set predictions from the sequential and two-step algorithms to obtain overall metrics.

4.3 Optimal Gene Sets

4.3.1 Sequential Algorithm

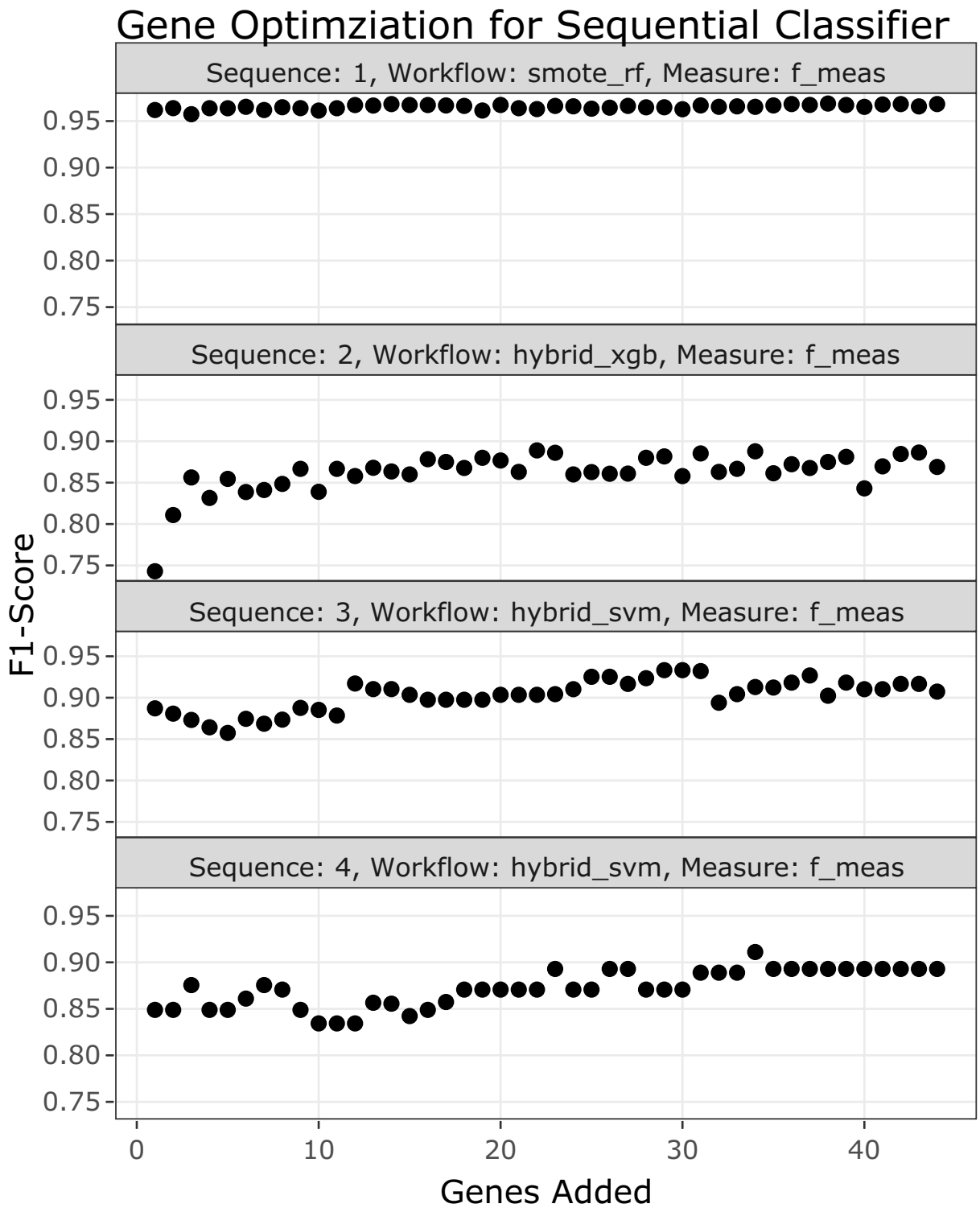

Figure 4.17: Gene Optimization for Sequential Classifier

In the sequential algorithm, sequences 1, 2, and 4 have relatively flat average F1-scores across the number of genes added. However, we can observe in sequence 3, the F1-score stabilizes at around 0.91 when we reach 12 genes added, hence the optimal number of genes used will be n=28+12=40 The added genes are: CYP2C18, TFF3, HNF1B, SLC3A1, MAP1LC3A, IL6, BRCA1, EGFL6, IGFBP1, MUC5B, BCL2 and WT1.

4.3.2 Two-Step Algorithm

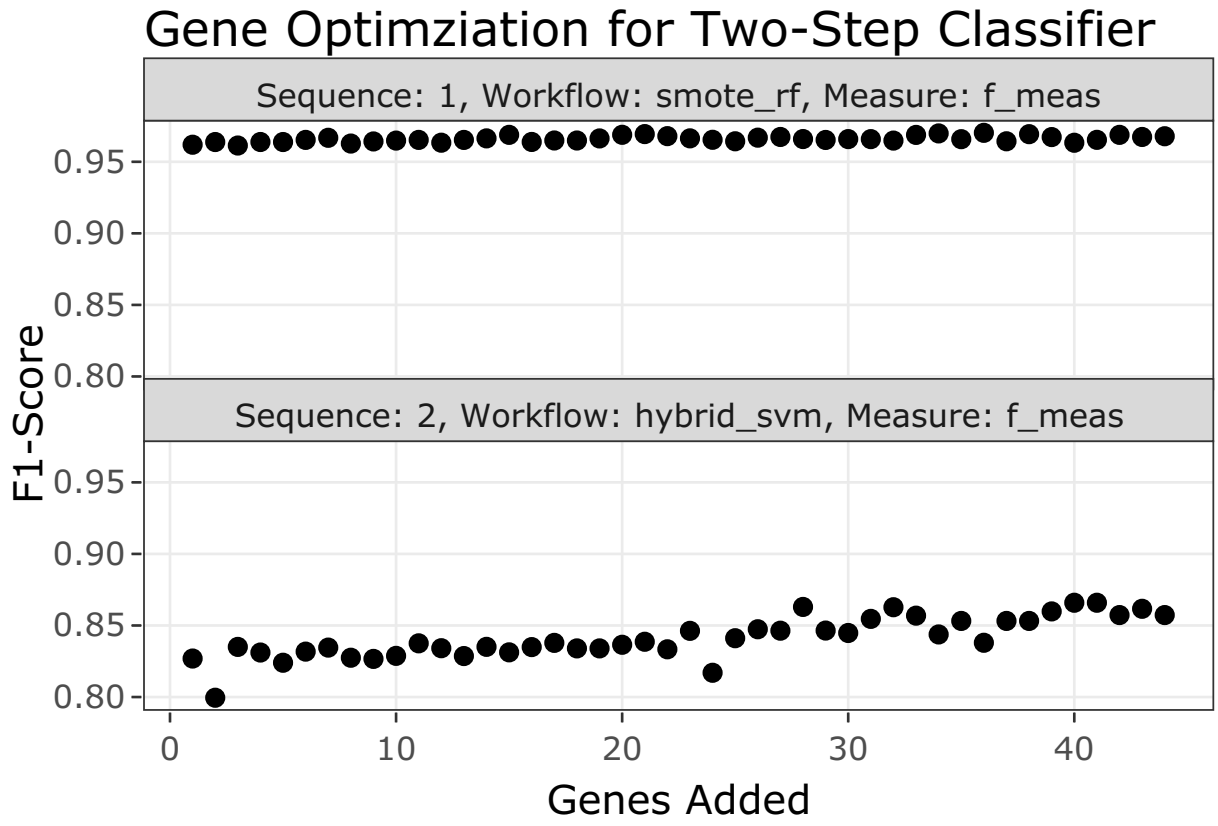

Figure 4.18: Gene Optimization for Two-Step Classifier

Since the second step of the classifier fits a multinomial model, we use the macro F1-score as the measure to analyze gene entry. In the two-step classifier, we see that in Step 2, the F1-score stabilizes at around 0.84 when we reach 11 added. The optimal number of genes used will be n=28+11=39. The added genes are: CYP2C18, MUC5B, HNF1B, SLC3A1, EGFL6, WT1, TFF3, MET, IGFBP1, TP53 and KLK7.

4.4 Test Set Performance

Now we'd like to see how our best methods perform in the confirmation and validation sets. The class-specific F1-scores will be used.

The top 2 methods are:

- **sequential**: sequential algorithm with hybrid subsampling at every step. The sequence of algorithms used are:

Table 4.15: Overall Evaluation Metrics on Confirmation Set Models

Method	accuracy	sensitivity	specificity	f1	bal_accuracy	kappa	gmean
sequential_full	0.832	0.646	0.927	0.664	0.786	0.660	0.599
sequential_optimal	0.829	0.609	0.924	0.628	0.767	0.651	0.516
two_step_full	0.838	0.674	0.928	0.694	0.801	0.673	0.638
two_step_optimal	0.832	0.672	0.923	0.694	0.798	0.655	0.639

  - HGSC vs. non-HGSC using random forest
  - CCOC vs. non-CCOC using XGBoost
  - ENOC vs. non-ENOC using support vector machine
  - LGSC vs. MUC using support vector machine
- **two_step**: two-step algorithm with hybrid subsampling at both steps. The sequence of algorithms used are:
  - HGSC vs. non-HGSC using random forest
  - CCOC vs. ENOC vs. MUC vs. LGSC support vector machine

We can test 2 additional methods by using either the full set of genes or the optimal set of genes for both of these methods.

4.4.1 Confirmation Set

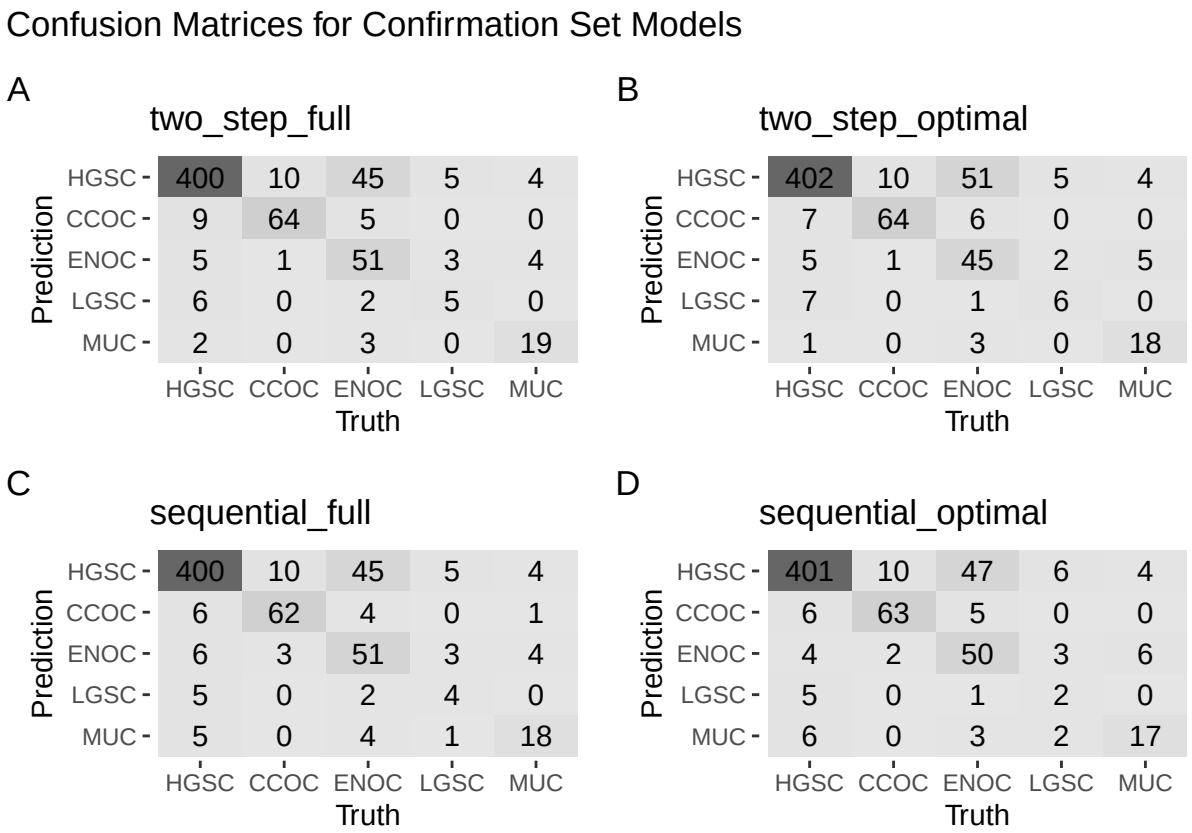

Figure 4.19: Confusion Matrices for Confirmation Set Models

Table 4.16: Per-Class Evaluation Metrics on Confirmation Set Model

Method	Metric	Histotypes				
		HGSC	CCOC	ENOC	LGSC	MUC
two_step_full	accuracy	0.866	0.961	0.894	0.975	0.980
	sensitivity	0.948	0.853	0.481	0.385	0.704
	specificity	0.710	0.975	0.976	0.987	0.992
	f1	0.903	0.837	0.600	0.385	0.745
	bal_accuracy	0.829	0.914	0.728	0.686	0.848
	kappa	0.689	0.815	0.543	0.372	0.735
	gmean	0.821	0.912	0.685	0.616	0.835
two_step_optimal	accuracy	0.860	0.963	0.885	0.977	0.980
	sensitivity	0.953	0.853	0.425	0.462	0.667
	specificity	0.683	0.977	0.976	0.987	0.994
	f1	0.899	0.842	0.549	0.444	0.735
	bal_accuracy	0.818	0.915	0.700	0.724	0.830
	kappa	0.672	0.821	0.489	0.433	0.724
	gmean	0.807	0.913	0.644	0.675	0.814
sequential_full	accuracy	0.866	0.963	0.890	0.975	0.970
	sensitivity	0.948	0.827	0.481	0.308	0.667
	specificity	0.710	0.981	0.970	0.989	0.984
	f1	0.903	0.838	0.590	0.333	0.655
	bal_accuracy	0.829	0.904	0.726	0.648	0.825
	kappa	0.689	0.817	0.530	0.321	0.639
	gmean	0.821	0.900	0.683	0.552	0.810
sequential_optimal	accuracy	0.863	0.964	0.890	0.974	0.967
	sensitivity	0.950	0.840	0.472	0.154	0.630
	specificity	0.697	0.981	0.972	0.990	0.982
	f1	0.901	0.846	0.585	0.190	0.618
	bal_accuracy	0.824	0.910	0.722	0.572	0.806
	kappa	0.681	0.825	0.525	0.178	0.601
	gmean	0.814	0.908	0.677	0.390	0.786

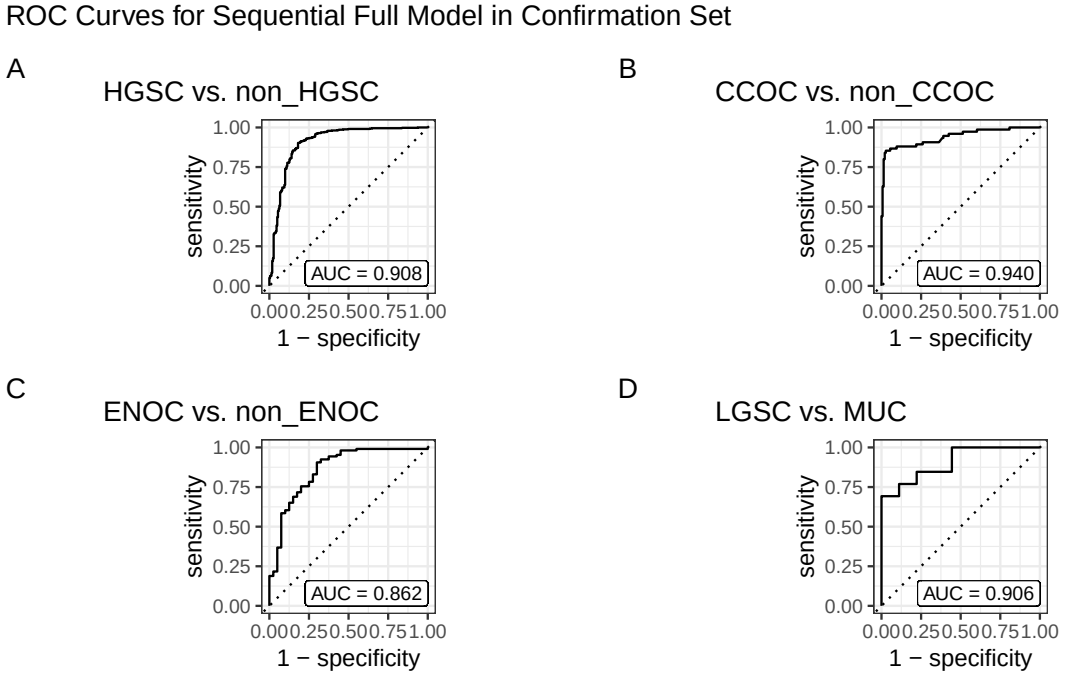

Figure 4.20: ROC Curves for Sequential Full Model in Confirmation Set

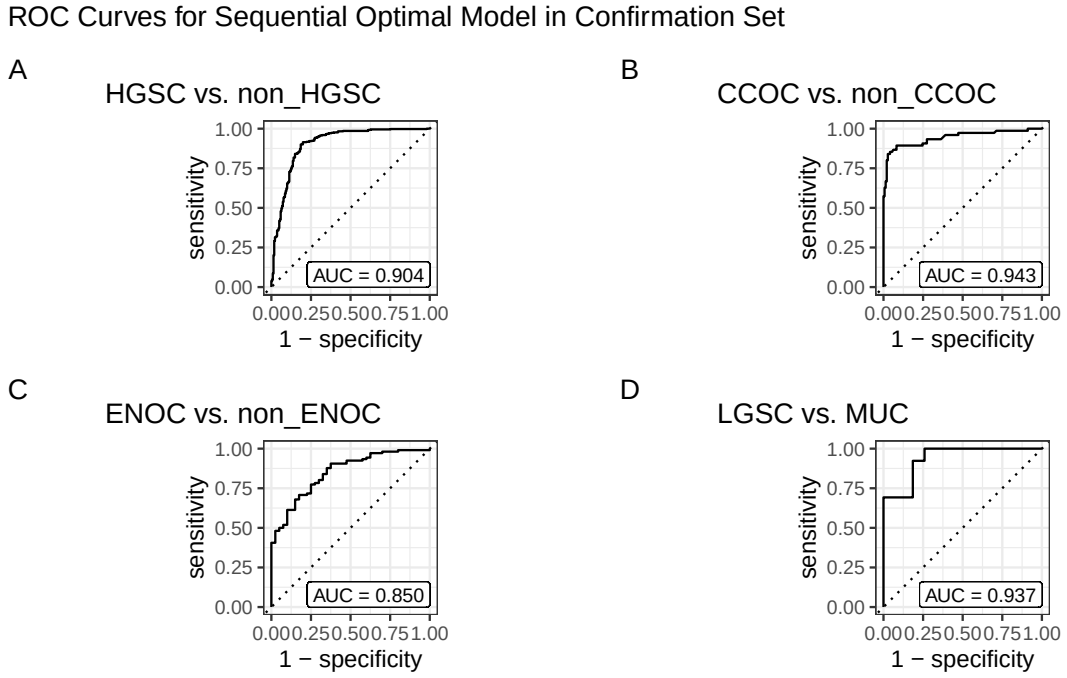

Figure 4.21: ROC Curves for Sequential Optimal Model in Confirmation Set

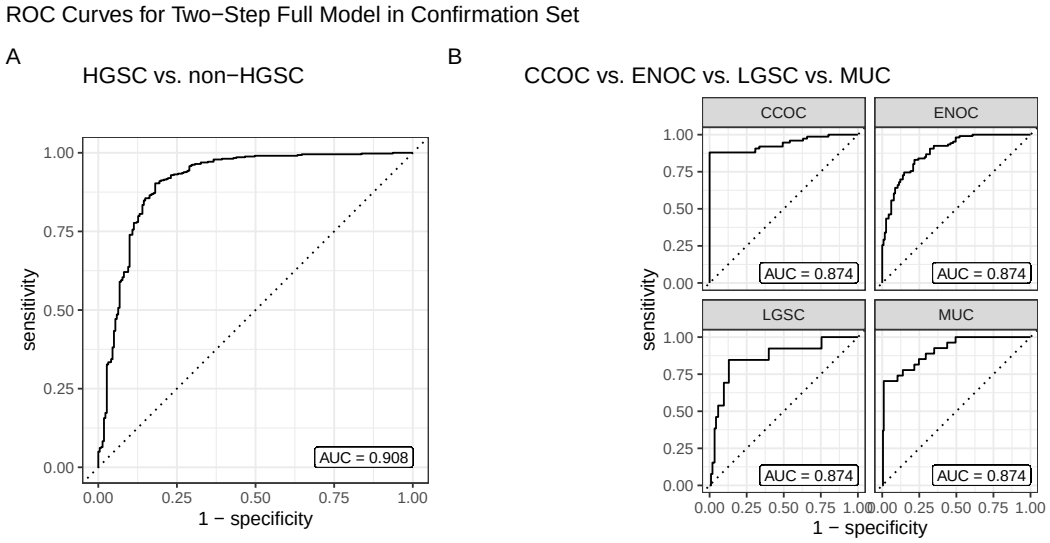

Figure 4.22: ROC Curves for Two-Step Full Model in Confirmation Set

Table 4.17: Overall Evaluation Metrics on Validation Set Model

Method	accuracy	sensitivity	specificity	f1	bal_accuracy	kappa	gmean
two_step_optimal	0.87	0.713	0.94	0.679	0.826	0.691	0.662

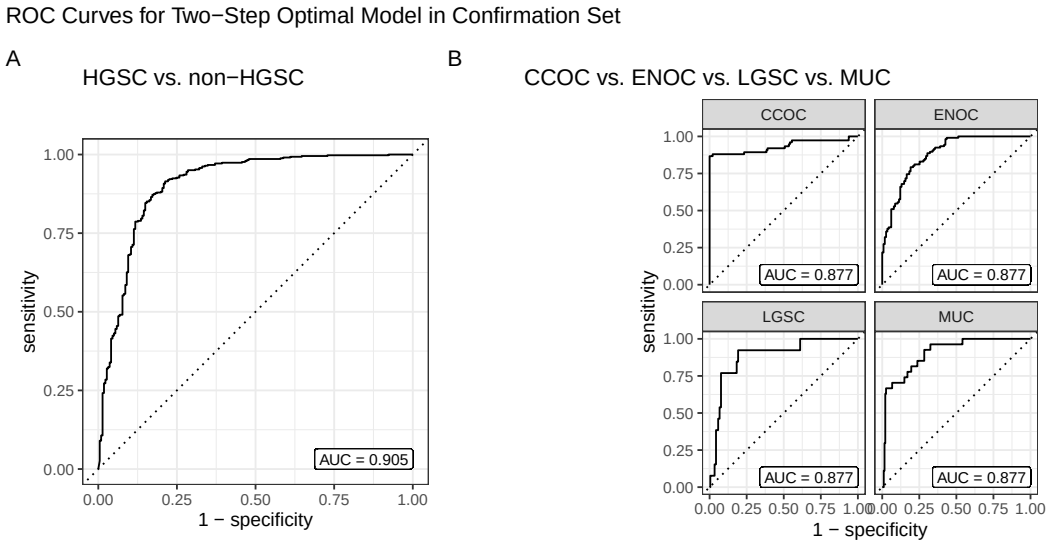

Figure 4.23: ROC Curves for Two-Step Optimal Model in Confirmation Set

Table 4.18: Per-Class Eevaluation Metrics on Validation Set Model

Method	Metric	Histotypes				
		HGSC	CCOC	ENOC	LGSC	MUC
two_step_optimal	accuracy	0.894	0.972	0.928	0.974	0.972
	sensitivity	0.934	0.937	0.514	0.333	0.846
	specificity	0.776	0.975	0.984	0.987	0.976
	f1	0.929	0.855	0.628	0.343	0.638
	bal_accuracy	0.855	0.956	0.749	0.660	0.911
	kappa	0.718	0.840	0.590	0.330	0.624
	gmean	0.852	0.956	0.711	0.574	0.909

4.4.2 Validation Set

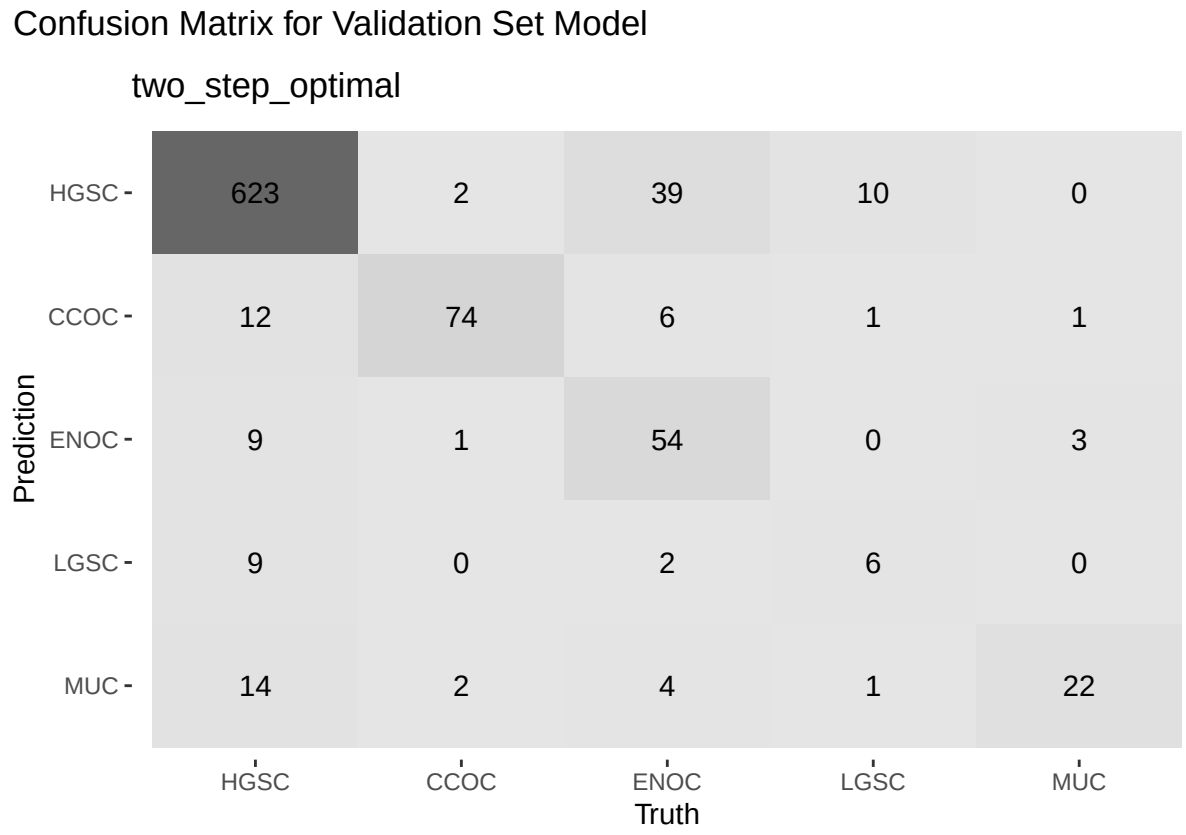

Figure 4.24: Confusion Matrix for Validation Set Model

References

Talhouk, A., Kommoss, S., Mackenzie, R., Cheung, M., Leung, S., Chiu, D. S., Kalloger, S. E., Huntsman, D. G., Chen, S., Intermaggio, M., Gronwald, J., Chan, F. C., Ramus, S. J., Steidl, C., Scott, D. W., and Anglesio, M. S. (2016). Single-patient molecular testing with nanostring ncounter data using a reference-based strategy for batch effect correction. *PLOS ONE*, 11(4):e0153844.