Ovarian Cancer Histotypes: Report of Statistical Findings

Derek Chiu

April 12, 2025

Table of 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, and Kappa, 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.

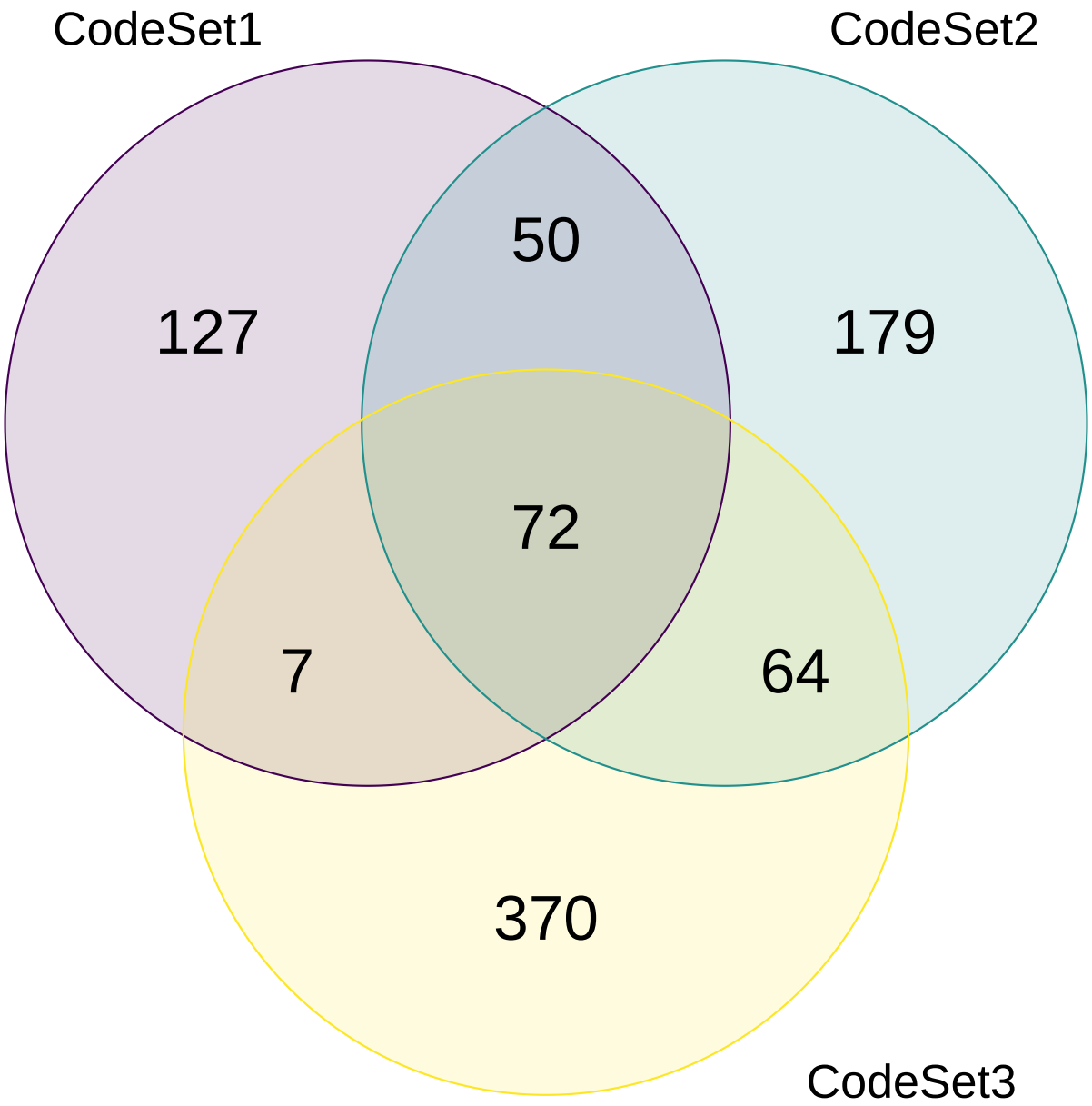


Figure 2.1: Venn diagram of common and unique gene targets covered by each CodeSet

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 carcinoma (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.

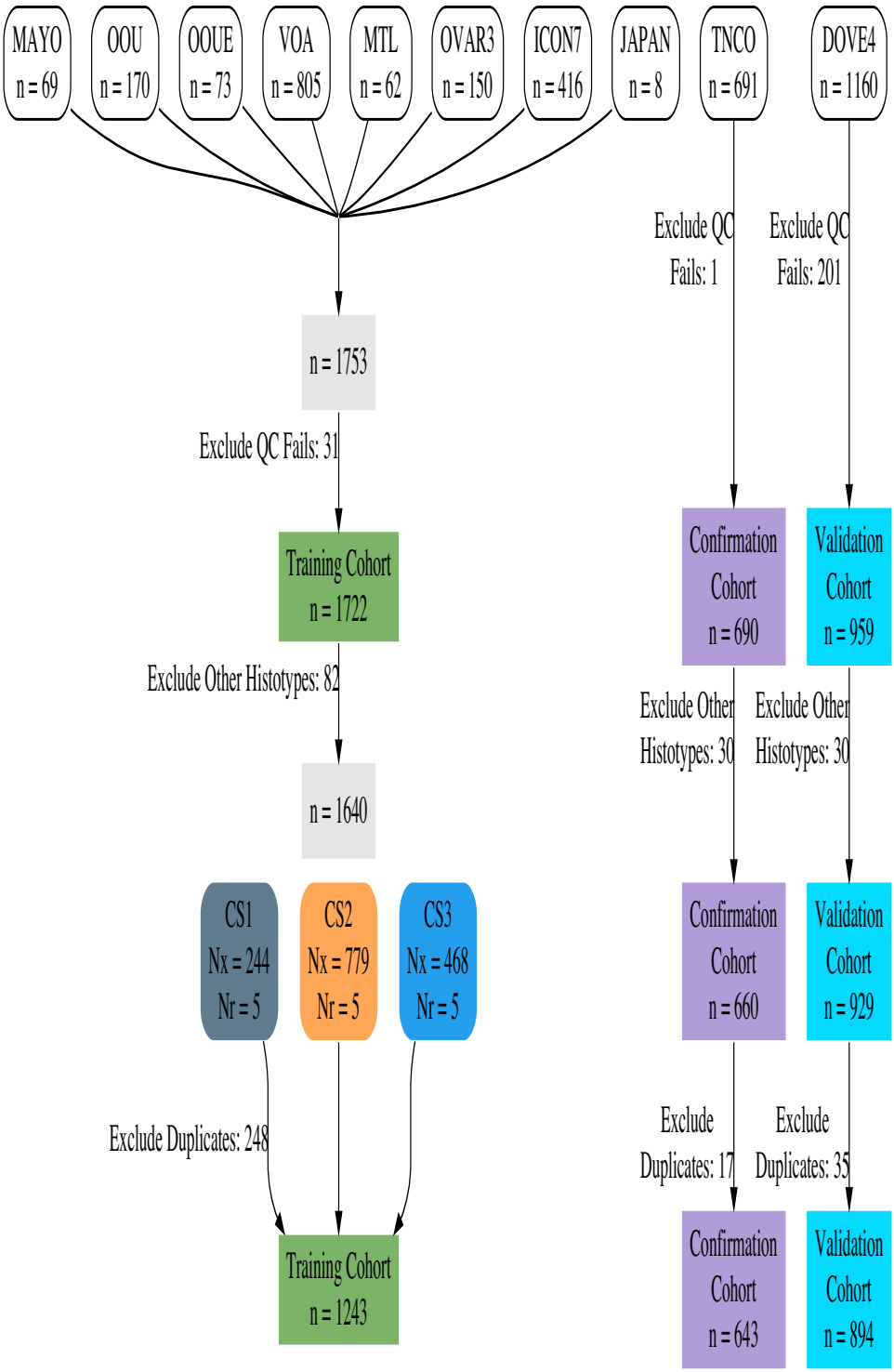


Figure 2.2: Cohorts Selection

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:

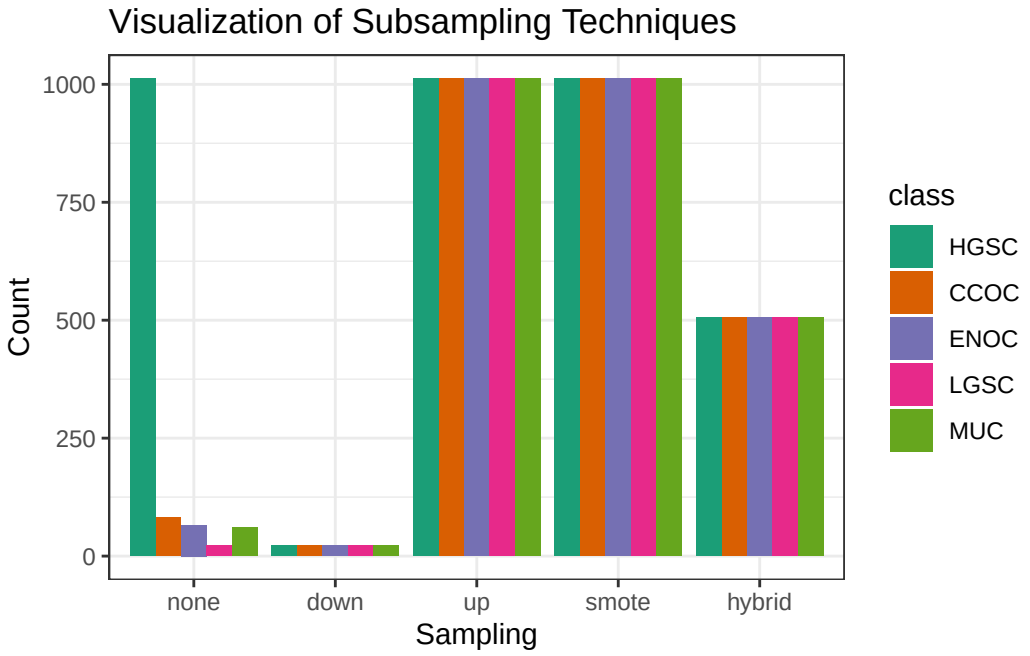


Figure 2.3: Visualization of Subsampling Techniques

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

$$\begin{aligned} X_k &= \text{Training data with k classes} \\ C_k &= \text{Class with highest } F_1 \text{ score from training } X_k \\ W_k &= \text{Workflow associated with } C_k \end{aligned} \tag{2.1}$$

Figure 2.4 shows how the two-step algorithm works:

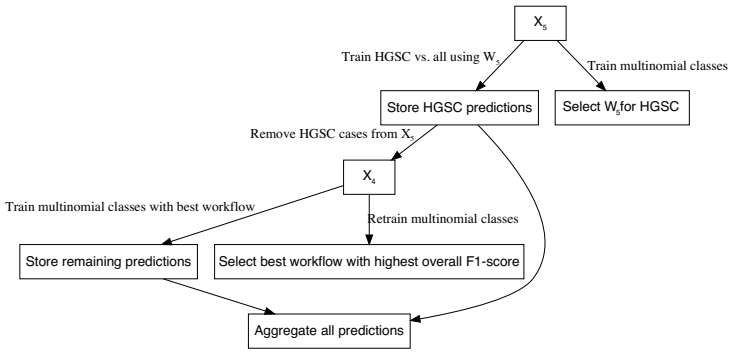


Figure 2.4: Two-Step Algorithm

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

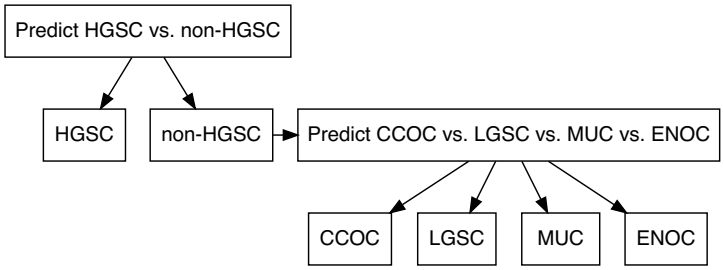


Figure 2.5: Aggregating Predictions for Two-Step Algorithm

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

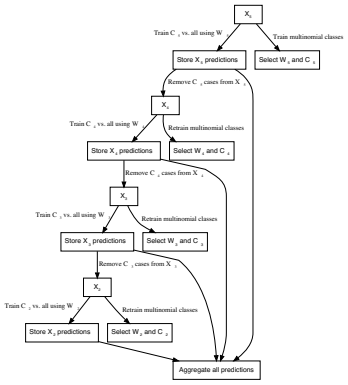


Figure 2.6: Sequential Algorithm

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 2.7 illustrates how the final predictions are assigned:

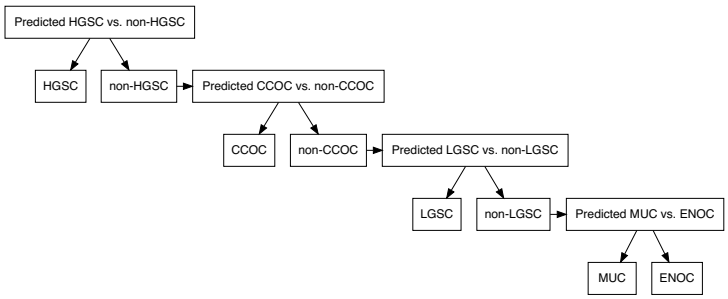


Figure 2.7: Aggregating Predictions for Sequential Algorithm

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.

Here is the breakdown of genes used and whether they belong to the PrOTYPE and/or SPOT sets:

Table 2.1: Gene Distribution

[H]

Genes	PrOTYPE	SPOT
TCF7L1	v	v
COL11A1	v	
CD74	v	
CD2	v	
TIMP3	v	
LUM	v	
CYTIP	v	
COL3A1	v	
THBS2	v	
HMGA2	v	
FN1	v	
POSTN	v	
COL1A2	v	
COL5A2	v	
PDZK1IP1	v	
FBN1	v	
HIF1A		v
CXCL10		v
DUSP4		v
SOX17		v
MITF		v
CDKN3		v
BRCA2		v
CEACAM5		v
ANXA4		v
SERPINE1		v
CRABP2		v
DNAJC9		v
C10orf116		
GAD1		
TPX2		
KGFLP2		
EGFL6		
KLK7		
PBX1		
LIN28B		
TFF3		
MUC5B		
FUT3		
STC1		
BCL2		
PAX8		
GCNT3		
GPR64		
ADCYAP1R1		
IGKC		
BRCA1		
IGJ		
TFF1		

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:

$$\text{accuracy} = \frac{TP}{TP + FP + FN + TN} \tag{2.2}$$

2.6.1.2 Sensitivity

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

$$\text{sensitivity} = \frac{TP}{TP + FN} \tag{2.3}$$

2.6.1.3 Specificity

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

$$\text{specificity} = \frac{TN}{TN + FP} \tag{2.4}$$

2.6.1.4 F1-Score

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

$$F_{meas} = \frac{(1 + \beta^2) \times precision \times recall}{(\beta^2 \times precision) + recall} \tag{2.5}$$

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:

$$F_{1_{macro}} = \frac{1}{k} \sum_{i=1}^{k} F_{1_i} \tag{2.6}$$

where each F_{1_i} 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_{1_i} is undefined because the per-class precision of class i is undefined. Those F_{1_i} terms are removed from the $F_{1_{macro}}$ equation and the resulting value may be inflated. Interpreting the F1-score in such a case would be misleading.

2.6.1.5 Balanced Accuracy

Balanced accuracy is the arithmetic mean of sensitivity and specificity.

$$\text{Balanced Accuracy} = \frac{\text{Sensitivity} + \text{Specificity}}{2} \tag{2.7}$$

2.6.1.6 Kappa

Kappa is the defined as:

$$\text{kappa} = \frac{p_0 - p_e}{1 - p_e} \tag{2.8}$$

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

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 Histotype Distribution

Table 3.1: Histotype Distribution in Training Set by Processing Stage

[H]

Variable	Levels	CS1	CS2	CS3	Total
Selected Cohorts					
Histotype	HGSC	126 (43%)	655 (73%)	1779 (72%)	2560 (70%)
	CCOC	48 (16%)	61 (7%)	181 (7%)	290 (8%)
	ENOC	60 (20%)	34 (4%)	268 (11%)	362 (10%)
	MUC	20 (7%)	62 (7%)	77 (3%)	159 (4%)
	LGSC	21 (7%)	21 (2%)	42 (2%)	84 (2%)
	Other	19 (6%)	70 (8%)	130 (5%)	219 (6%)
Total	N (%)	294 (8%)	903 (25%)	2477 (67%)	3674 (100%)
QC					
Histotype	HGSC	120 (42%)	641 (73%)	1636 (72%)	2397 (70%)
	CCOC	48 (17%)	61 (7%)	173 (8%)	282 (8%)
	ENOC	60 (21%)	32 (4%)	229 (10%)	321 (9%)
	MUC	19 (7%)	60 (7%)	69 (3%)	148 (4%)
	LGSC	20 (7%)	21 (2%)	40 (2%)	81 (2%)
	Other	19 (7%)	67 (8%)	126 (6%)	212 (6%)
Total	N (%)	286 (8%)	882 (26%)	2273 (66%)	3441 (100%)
Main Histotypes					
Histotype	HGSC	120 (45%)	641 (79%)	1636 (76%)	2397 (74%)
	CCOC	48 (18%)	61 (7%)	173 (8%)	282 (9%)
	ENOC	60 (22%)	32 (4%)	229 (11%)	321 (10%)
	MUC	19 (7%)	60 (7%)	69 (3%)	148 (5%)
	LGSC	20 (7%)	21 (3%)	40 (2%)	81 (3%)
Total	N (%)	267 (8%)	815 (25%)	2147 (66%)	3229 (100%)
Removed Duplicates					
Histotype	HGSC	117 (47%)	623 (79%)	1540 (77%)	2280 (75%)
	CCOC	45 (18%)	55 (7%)	159 (8%)	259 (9%)
	ENOC	56 (22%)	28 (4%)	216 (11%)	300 (10%)
	MUC	16 (6%)	58 (7%)	59 (3%)	133 (4%)
	LGSC	15 (6%)	20 (3%)	36 (2%)	71 (2%)
Total	N (%)	249 (8%)	784 (26%)	2010 (66%)	3043 (100%)
Normalized and Recombined					
Histotype	HGSC	116 (48%)	622 (80%)	451 (96%)	1189 (80%)
	CCOC	44 (18%)	54 (7%)	4 (1%)	102 (7%)
	ENOC	55 (23%)	27 (3%)	4 (1%)	86 (6%)
	MUC	15 (6%)	57 (7%)	5 (1%)	77 (5%)
	LGSC	14 (6%)	19 (2%)	4 (1%)	37 (2%)
Total	N (%)	244 (16%)	779 (52%)	468 (31%)	1491 (100%)
Removed Replicates					
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 3.2: Histotype Distribution in Training, Confirmation, and Validation Sets

[H]

Variable	Levels	Training	Confirmation	Validation
Histotype	HGSC	1012 (81%)	422 (66%)	666 (74%)
	CCOC	81 (7%)	75 (12%)	79 (9%)
	ENOC	66 (5%)	106 (16%)	105 (12%)
	MUC	61 (5%)	27 (4%)	26 (3%)
	LGSC	23 (2%)	13 (2%)	18 (2%)
Total	N (%)	1243 (45%)	643 (23%)	894 (32%)

3.2 Cohort Distribution

Table 3.3: Pre-QC Cohort Distribution by CodeSet

CodeSet	**CS1**, N = 294	**CS2**, N = 903	**CS3**, N = 2,477
Cohort			
OOU	108 (37%)	43 (4.8%)	19 (0.8%)
OOUE	32 (11%)	30 (3.3%)	11 (0.4%)
VOA	145 (49%)	122 (14%)	538 (22%)
OVAR3	0 (0%)	150 (17%)	0 (0%)
ICON7	0 (0%)	416 (46%)	0 (0%)
MAYO	6 (2.0%)	63 (7.0%)	0 (0%)
DOVE4	0 (0%)	0 (0%)	1,160 (47%)
TNCO	0 (0%)	0 (0%)	691 (28%)
MTL	3 (1.0%)	59 (6.5%)	0 (0%)
JAPAN	0 (0%)	8 (0.9%)	0 (0%)
POOL-CTRL	0 (0%)	12 (1.3%)	0 (0%)
POOL-1	0 (0%)	0 (0%)	31 (1.3%)
POOL-2	0 (0%)	0 (0%)	14 (0.6%)
POOL-3	0 (0%)	0 (0%)	13 (0.5%)

3.3 Quality Control

3.3.1 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 Section 3.3.2 relationship below.

Table 3.4: Quality Control Summary

Quality Control Flag	**CS1**, N = 294	**CS2**, N = 903	**CS3**, N = 2,477
___Linearity___			
Failed	0 (0%)	4 (0.4%)	0 (0%)
Passed	294 (100%)	899 (100%)	2,477 (100%)
___Imaging___			
Failed	3 (1.0%)	0 (0%)	4 (0.2%)
Passed	291 (99%)	903 (100%)	2,473 (100%)
___Smallest Positive Control___			
Failed	0 (0%)	2 (0.2%)	0 (0%)
Passed	294 (100%)	901 (100%)	2,477 (100%)
___Normality___			
Failed	5 (1.7%)	19 (2.1%)	200 (8.1%)
Passed	289 (98%)	884 (98%)	2,277 (92%)
___Overall QC___			
Failed	8 (2.7%)	21 (2.3%)	204 (8.2%)
Passed	286 (97%)	882 (98%)	2,273 (92%)

3.3.2 %GD vs. SNR

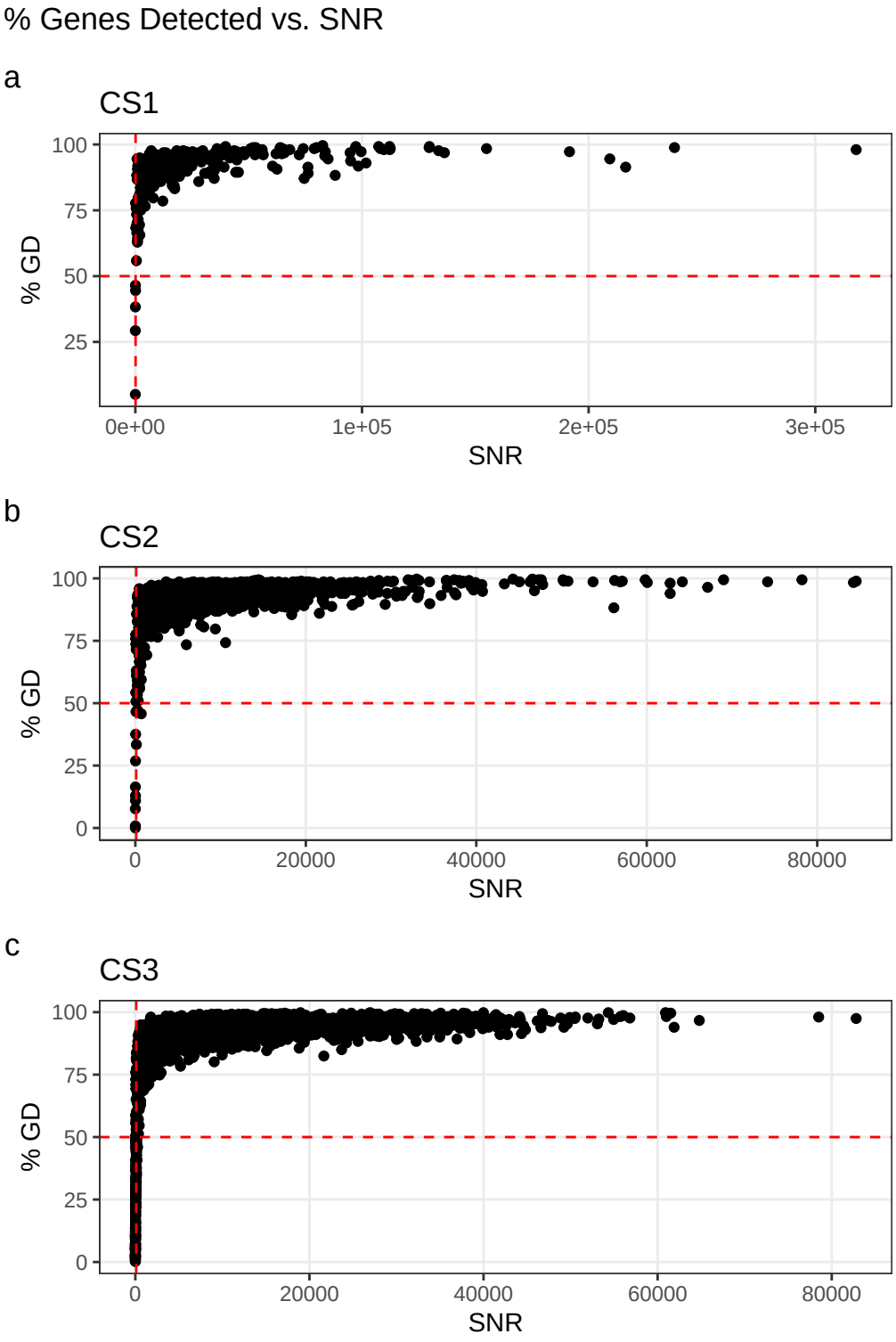


Figure 3.1: % Genes Detected vs. Signal to Noise Ratio

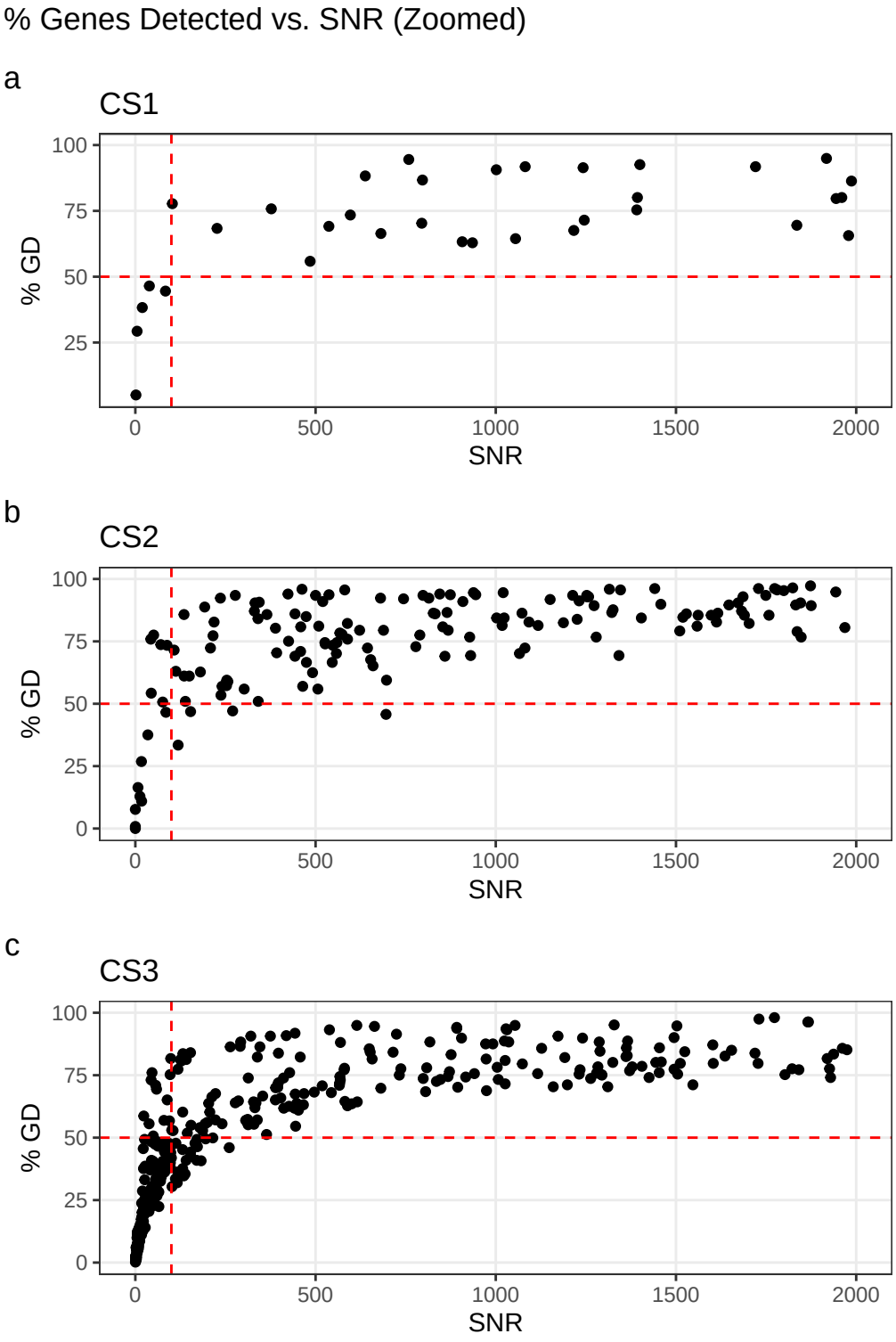


Figure 3.2: % Genes Detected vs. Signal to Noise Ratio (Zoomed)

3.4 Pairwise Gene Expression

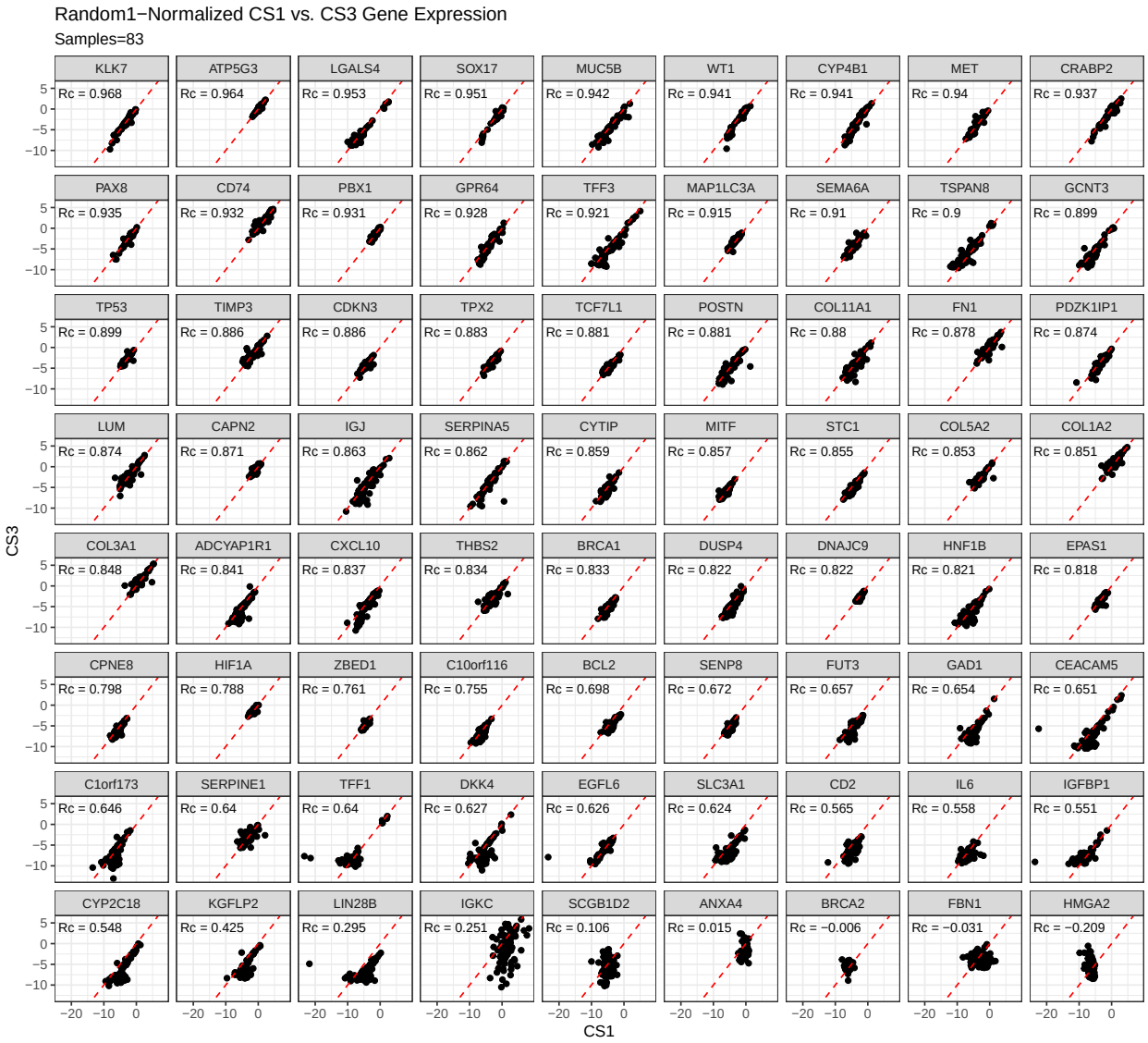


Figure 3.3: Random1-Normalized CS1 vs. CS3 Gene Expression

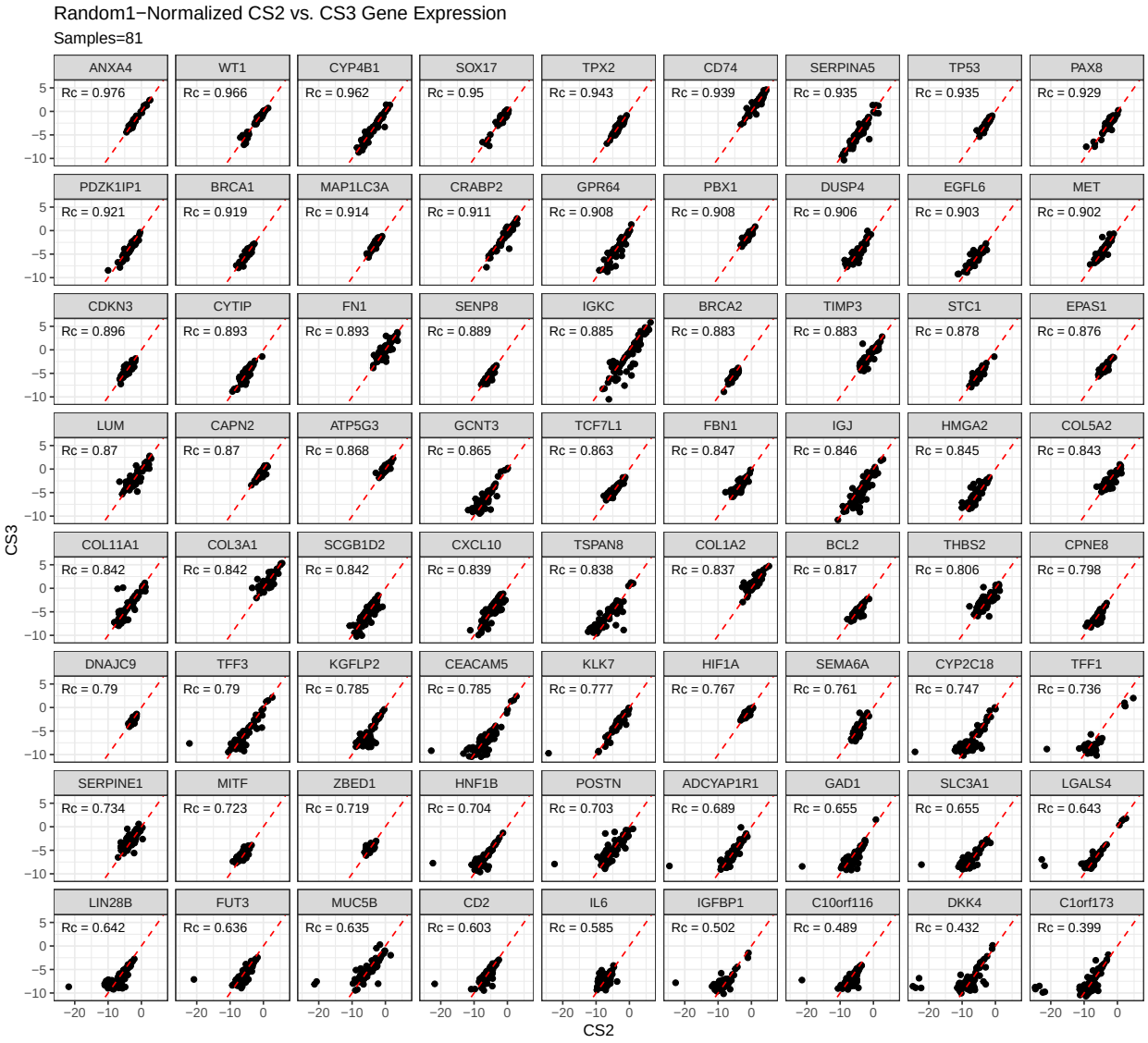


Figure 3.4: Random1-Normalized CS2 vs. CS3 Gene Expression

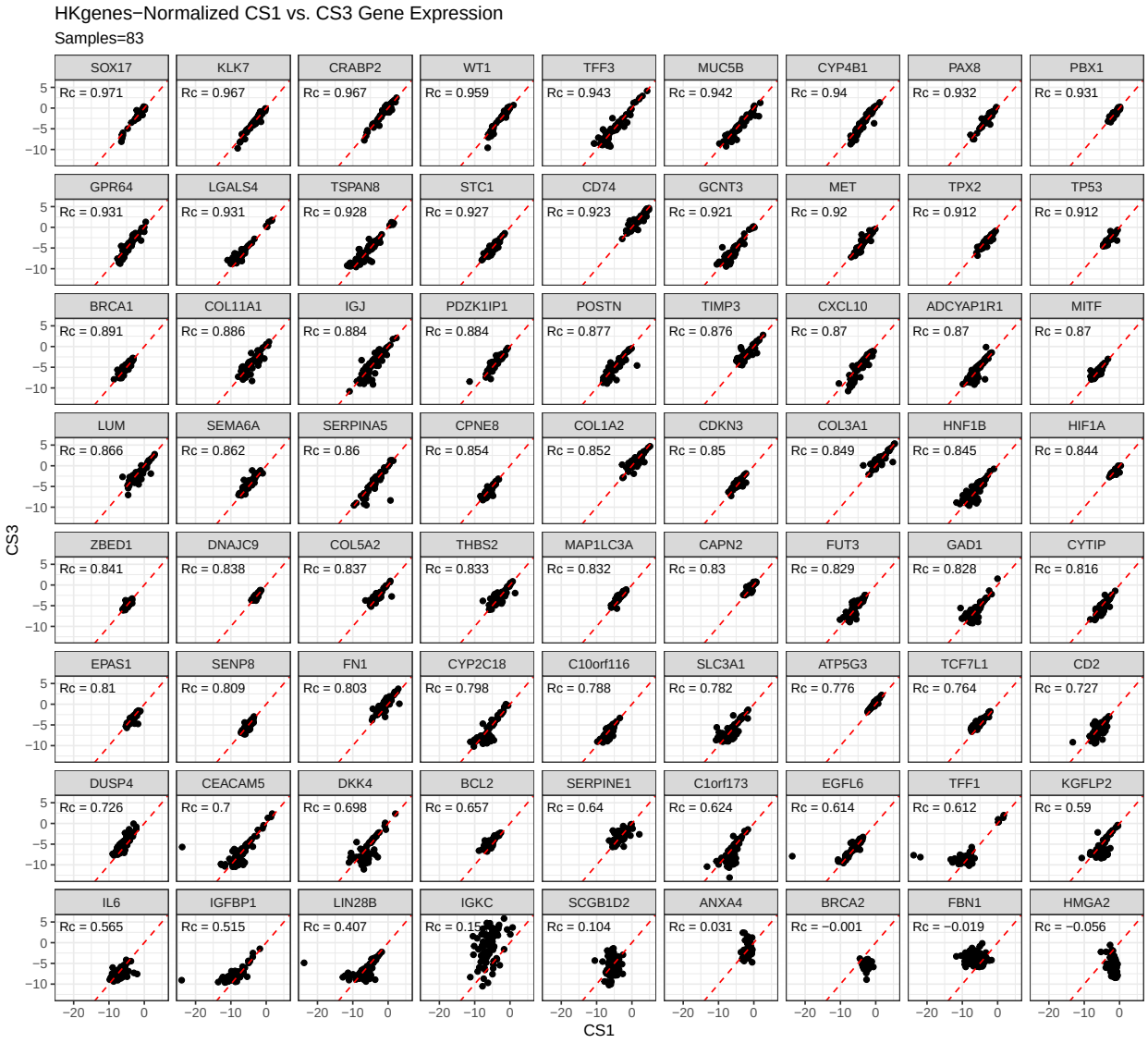


Figure 3.5: HKgenes-Normalized CS1 vs. CS3 Gene Expression

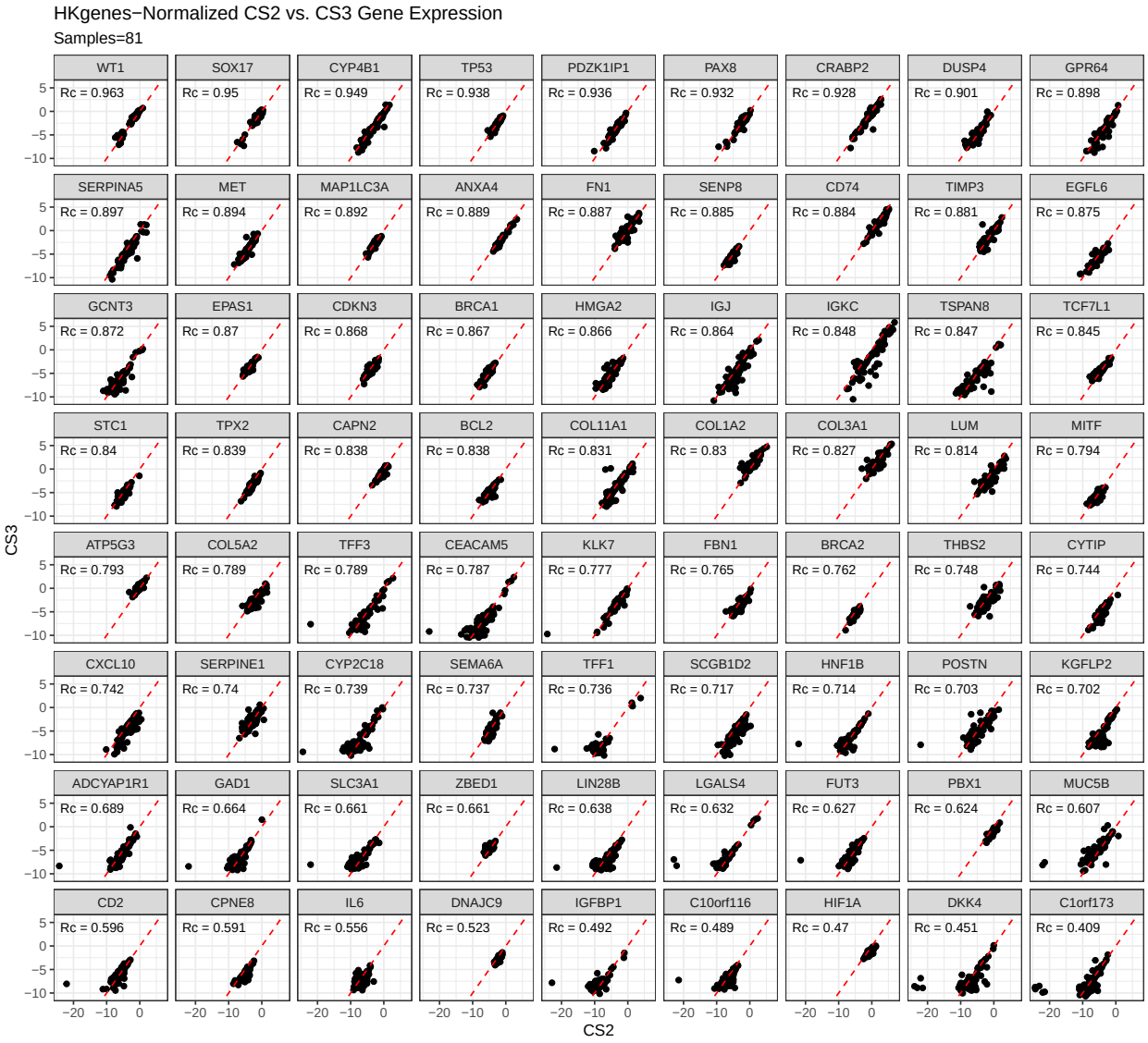


Figure 3.6: HKgenes-Normalized CS2 vs. CS3 Gene Expression

4 Results

We summarize cross-validated training performance of class metrics in the training set. The accuracy, F1-score, and kappa, 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

Table 4.1: Training Set Mean Accuracy

[H]

			Histotypes				
Subsampling	Algorithms	Overall	HGSC	CCOC	ENOC	LGSC	MUC
none	rf	0.93	0.947	0.982	0.965	0.981	0.985
	svm	0.941	0.961	0.982	0.97	0.985	0.985
	xgb	0.823	0.824	0.943	0.947	0.982	0.951
	mr	0.814	0.814	0.935	0.947	0.982	0.951
down	rf	0.829	0.861	0.976	0.945	0.909	0.968
	svm	0.797	0.831	0.965	0.924	0.901	0.973
	xgb	0.206	0.32	0.583	0.947	0.79	0.773
	mr	0.817	0.851	0.969	0.928	0.92	0.966
up	rf	0.928	0.949	0.979	0.969	0.977	0.982
	svm	0.932	0.959	0.973	0.963	0.986	0.982
	xgb	0.937	0.962	0.982	0.967	0.982	0.982
	mr	0.889	0.92	0.974	0.95	0.961	0.972
smote	rf	0.94	0.963	0.981	0.97	0.981	0.985
	svm	0.932	0.96	0.974	0.962	0.986	0.982
	xgb	0.934	0.957	0.981	0.967	0.982	0.981
	mr	0.885	0.914	0.974	0.956	0.947	0.979
hybrid	rf	0.932	0.957	0.979	0.969	0.981	0.978
	svm	0.92	0.948	0.973	0.957	0.981	0.982
	xgb	0.928	0.953	0.978	0.966	0.978	0.98
	mr	0.885	0.913	0.978	0.954	0.949	0.976

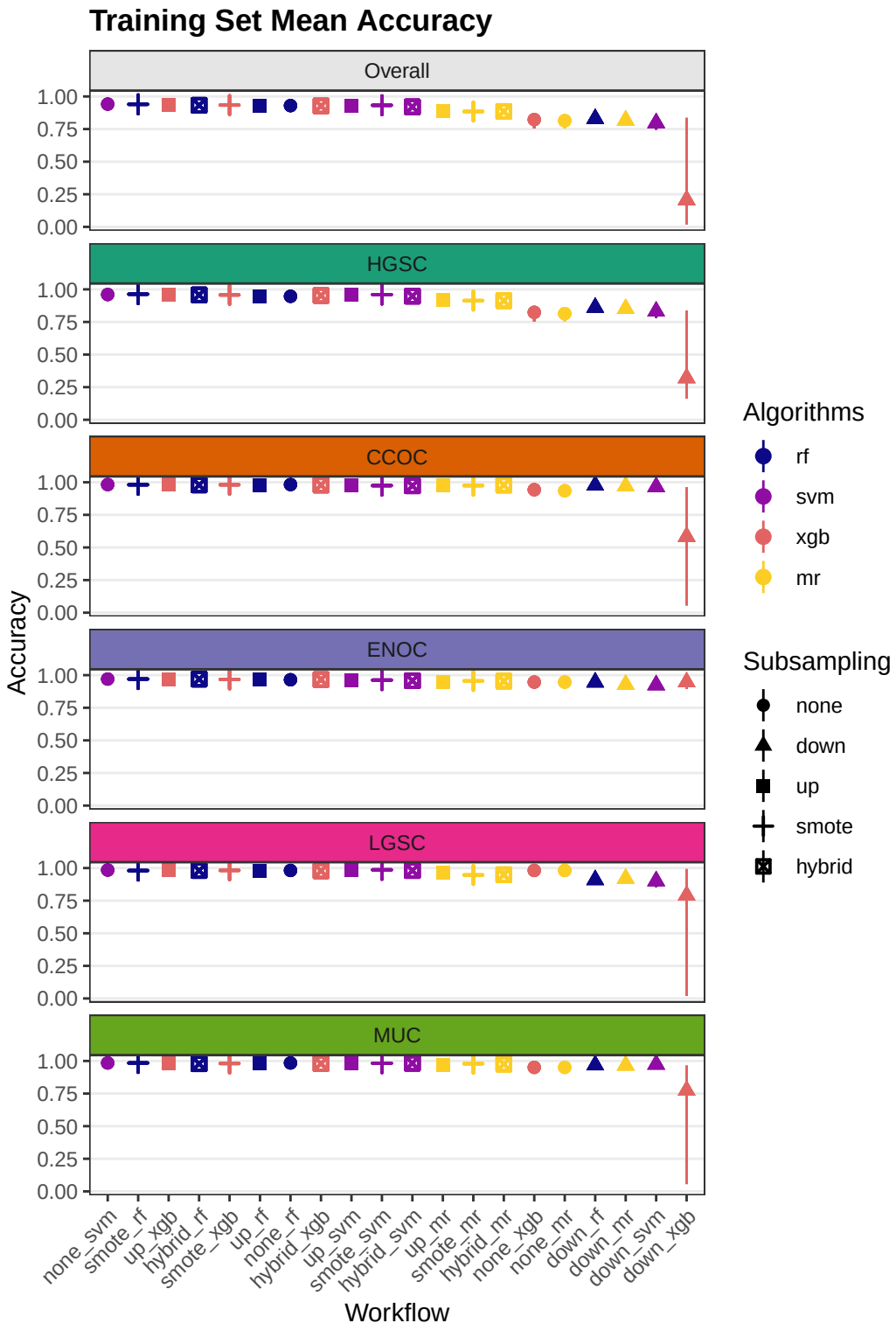


Figure 4.1: Training Set Mean Accuracy

4.1.2 Sensitivity

Table 4.2: Training Set Mean Sensitivity

[H]

			Histotypes				
Subsampling	Algorithms	Overall	HGSC	CCOC	ENOC	LGSC	MUC
none	rf	0.666	0.991	0.797	0.585	0.133	0.822
	svm	0.713	0.993	0.802	0.695	0.264	0.81
	xgb	0.241	1	0.203	0	0	0
	mr	0.2	1	0	0	0	0
down	rf	0.805	0.838	0.816	0.684	0.865	0.824
	svm	0.814	0.8	0.743	0.777	0.971	0.781
	xgb	0.2	0.2	0.4	0	0.2	0.2
	mr	0.79	0.826	0.772	0.683	0.865	0.804
up	rf	0.695	0.981	0.805	0.636	0.28	0.77
	svm	0.725	0.989	0.748	0.617	0.528	0.744
	xgb	0.75	0.98	0.805	0.715	0.397	0.854
	mr	0.834	0.91	0.789	0.766	0.871	0.833
smote	rf	0.755	0.984	0.805	0.762	0.38	0.841
	svm	0.738	0.988	0.748	0.661	0.534	0.76
	xgb	0.796	0.965	0.84	0.738	0.596	0.839
	mr	0.806	0.905	0.792	0.79	0.703	0.839
hybrid	rf	0.775	0.97	0.801	0.777	0.47	0.855
	svm	0.811	0.957	0.783	0.764	0.763	0.788
	xgb	0.792	0.96	0.825	0.727	0.594	0.854
	mr	0.821	0.903	0.812	0.782	0.77	0.839

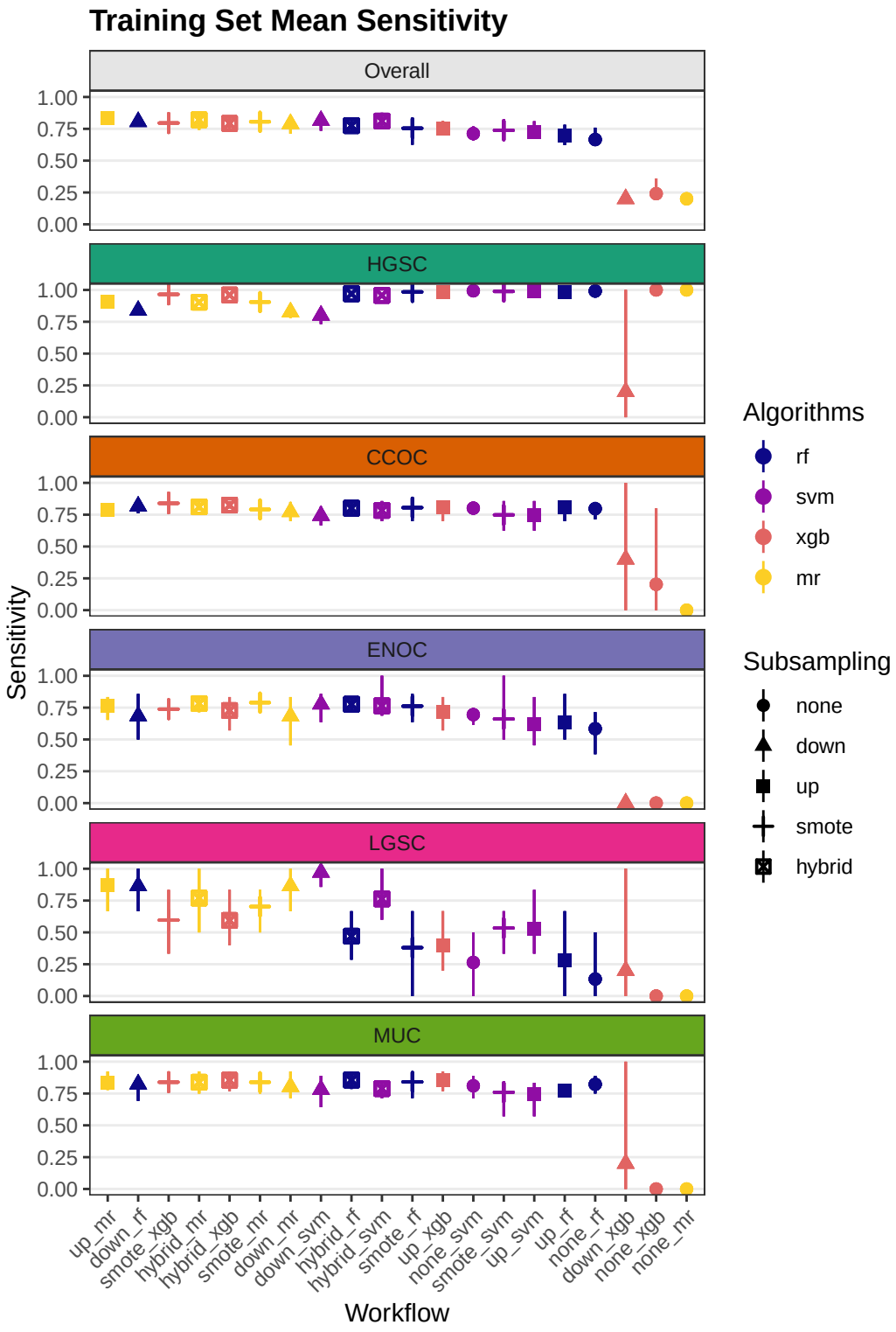


Figure 4.2: Training Set Mean Sensitivity

4.1.3 Specificity

Table 4.3: Training Set Mean Specificity

[H]

			Histotypes				
Subsampling	Algorithms	Overall	HGSC	CCOC	ENOC	LGSC	MUC
none	rf	0.947	0.761	0.995	0.988	0.998	0.993
	svm	0.959	0.821	0.995	0.986	0.998	0.994
	xgb	0.811	0.055	0.999	1	1	1
	mr	0.8	0	1	1	1	1
down	rf	0.958	0.961	0.987	0.958	0.91	0.975
	svm	0.952	0.963	0.98	0.933	0.9	0.984
	xgb	0.8	0.8	0.6	1	0.8	0.8
	mr	0.955	0.959	0.983	0.94	0.92	0.975
up	rf	0.953	0.803	0.991	0.987	0.991	0.992
	svm	0.959	0.833	0.99	0.984	0.993	0.995
	xgb	0.968	0.881	0.995	0.981	0.993	0.988
	mr	0.971	0.964	0.987	0.962	0.963	0.979
smote	rf	0.966	0.866	0.993	0.983	0.993	0.992
	svm	0.96	0.841	0.991	0.982	0.994	0.994
	xgb	0.974	0.923	0.991	0.979	0.989	0.988
	mr	0.968	0.95	0.987	0.965	0.951	0.986
hybrid	rf	0.97	0.903	0.991	0.981	0.991	0.985
	svm	0.969	0.91	0.986	0.97	0.985	0.992
	xgb	0.971	0.916	0.989	0.979	0.985	0.987
	mr	0.968	0.954	0.99	0.964	0.952	0.983

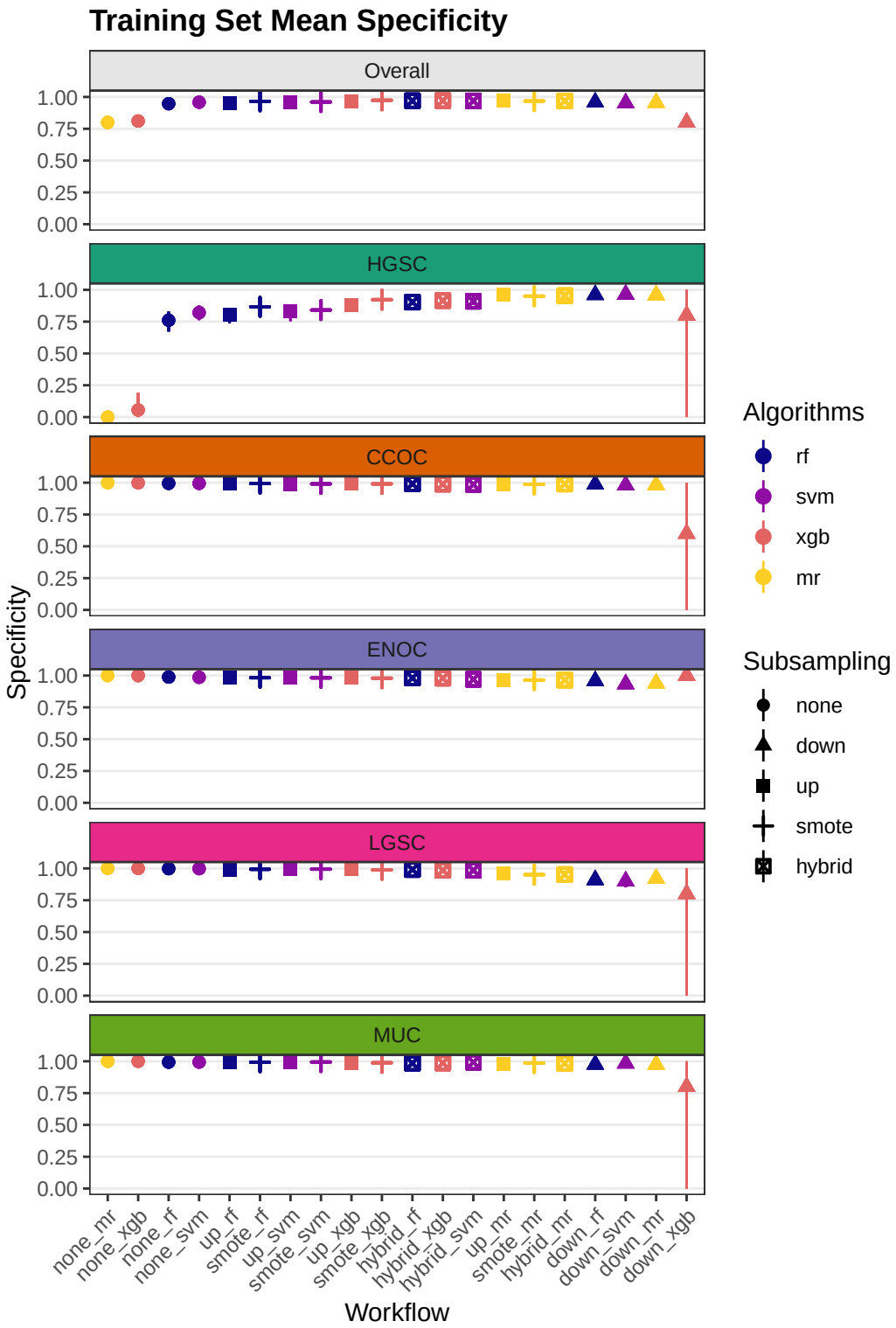


Figure 4.3: Training Set Mean Specificity

4.1.4 F1-Score

Table 4.4: Training Set Mean F1-Score

[H]

			Histotypes				
Subsampling	Algorithms	Overall	HGSC	CCOC	ENOC	LGSC	MUC
none	rf	0.751	0.968	0.849	0.622	0.262	0.838
	svm	0.775	0.976	0.852	0.695	0.433	0.837
	xgb	0.841	0.902	0.611	NaN	NaN	NaN
	mr	0.897	0.897	NaN	NaN	NaN	NaN
down	rf	0.648	0.907	0.813	0.545	0.255	0.719
	svm	0.623	0.884	0.728	0.486	0.275	0.744
	xgb	0.257	0.91	0.114	NaN	0.04	0.106
	mr	0.622	0.9	0.76	0.474	0.273	0.704
up	rf	0.707	0.969	0.829	0.657	0.278	0.803
	svm	0.741	0.975	0.776	0.612	0.54	0.802
	xgb	0.747	0.977	0.855	0.678	0.408	0.817
	mr	0.706	0.949	0.804	0.587	0.448	0.742
smote	rf	0.748	0.978	0.837	0.716	0.37	0.84
	svm	0.751	0.975	0.783	0.626	0.566	0.805
	xgb	0.763	0.973	0.845	0.663	0.528	0.806
	mr	0.7	0.945	0.797	0.631	0.334	0.793
hybrid	rf	0.748	0.974	0.828	0.698	0.448	0.791
	svm	0.751	0.967	0.785	0.622	0.569	0.81
	xgb	0.753	0.97	0.829	0.678	0.484	0.803
	mr	0.708	0.944	0.829	0.628	0.367	0.769

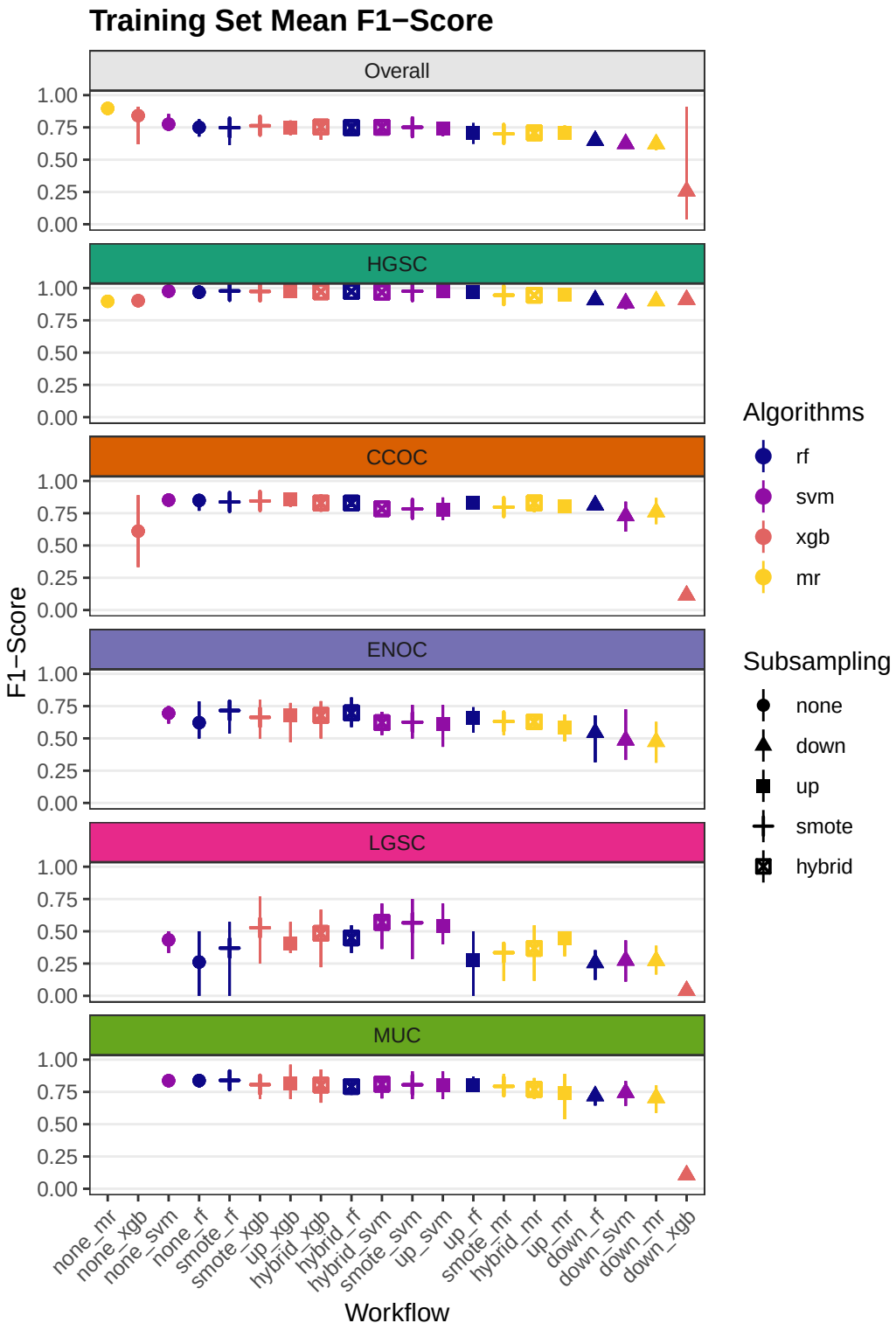


Figure 4.4: Training Set Mean F1-Score

4.1.5 Balanced Accuracy

Table 4.5: Training Set Mean Balanced Accuracy

[H]

			Histotypes				
Subsampling	Algorithms	Overall	HGSC	CCOC	ENOC	LGSC	MUC
none	rf	0.806	0.876	0.896	0.786	0.566	0.908
	svm	0.836	0.907	0.898	0.841	0.631	0.902
	xgb	0.526	0.528	0.601	0.5	0.5	0.5
	mr	0.5	0.5	0.5	0.5	0.5	0.5
down	rf	0.882	0.899	0.902	0.821	0.887	0.9
	svm	0.883	0.882	0.862	0.855	0.936	0.882
	xgb	0.5	0.5	0.5	0.5	0.5	0.5
	mr	0.873	0.892	0.878	0.812	0.893	0.889
up	rf	0.824	0.892	0.898	0.812	0.636	0.881
	svm	0.842	0.911	0.869	0.801	0.761	0.87
	xgb	0.859	0.931	0.9	0.848	0.695	0.921
	mr	0.903	0.937	0.888	0.864	0.917	0.906
smote	rf	0.86	0.925	0.899	0.872	0.687	0.917
	svm	0.849	0.915	0.869	0.822	0.764	0.877
	xgb	0.885	0.944	0.915	0.858	0.792	0.914
	mr	0.887	0.927	0.889	0.877	0.827	0.913
hybrid	rf	0.872	0.937	0.896	0.879	0.731	0.92
	svm	0.89	0.933	0.885	0.867	0.874	0.89
	xgb	0.882	0.938	0.907	0.853	0.79	0.92
	mr	0.895	0.928	0.901	0.873	0.861	0.911

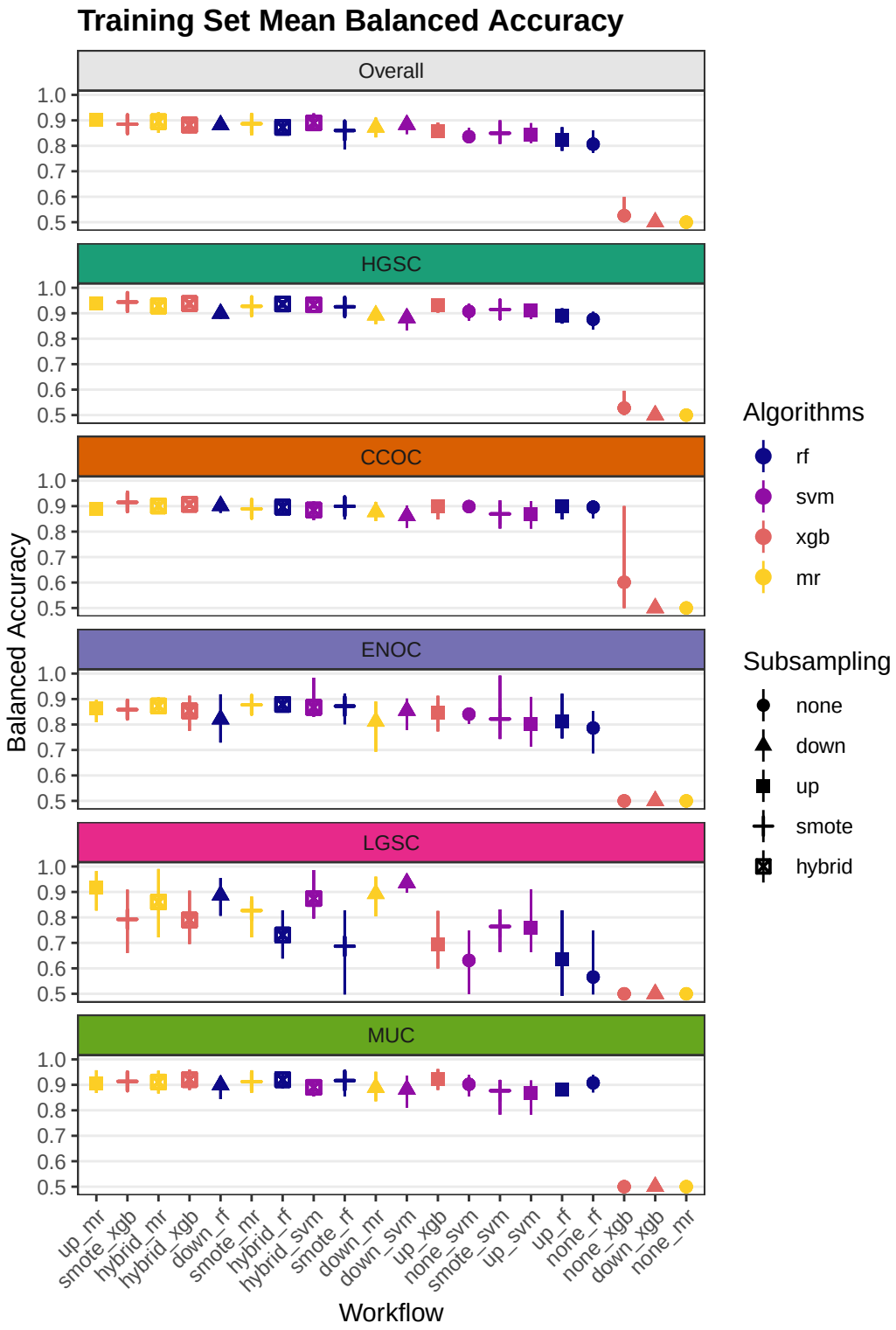


Figure 4.5: Training Set Mean Balanced Accuracy

4.1.6 Kappa

Table 4.6: Training Set Mean Kappa

[H]

			Histotypes				
Subsampling	Algorithms	Overall	HGSC	CCOC	ENOC	LGSC	MUC
none	rf	0.767	0.811	0.84	0.604	0.154	0.83
	svm	0.808	0.862	0.843	0.679	0.342	0.829
	xgb	0.084	0.083	0.24	0	0	0
	mr	0	0	0	0	0	0
down	rf	0.597	0.633	0.8	0.517	0.232	0.702
	svm	0.547	0.577	0.709	0.452	0.252	0.73
	xgb	0	0	0	0	0	0
	mr	0.574	0.613	0.744	0.439	0.251	0.687
up	rf	0.764	0.819	0.818	0.641	0.269	0.793
	svm	0.779	0.859	0.762	0.593	0.533	0.793
	xgb	0.804	0.871	0.845	0.66	0.399	0.807
	mr	0.706	0.766	0.79	0.562	0.434	0.728
smote	rf	0.806	0.87	0.827	0.7	0.363	0.832
	svm	0.783	0.862	0.77	0.606	0.56	0.796
	xgb	0.803	0.862	0.835	0.646	0.519	0.796
	mr	0.695	0.747	0.783	0.608	0.316	0.782
hybrid	rf	0.793	0.858	0.817	0.682	0.439	0.78
	svm	0.765	0.831	0.77	0.601	0.56	0.801
	xgb	0.786	0.846	0.817	0.661	0.475	0.793
	mr	0.696	0.745	0.818	0.604	0.349	0.757

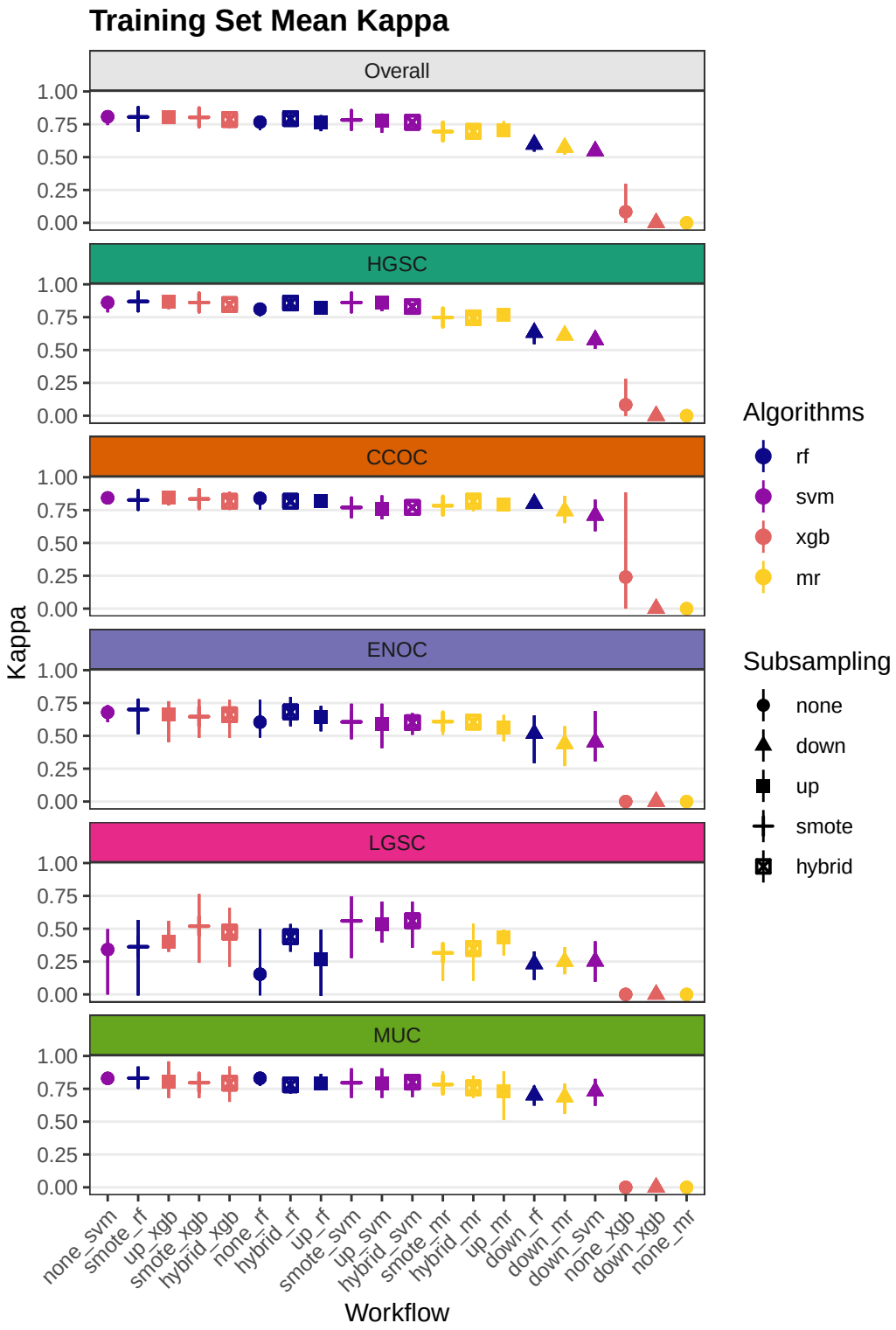


Figure 4.6: Training Set Mean Kappa

4.1.7 G-mean

DEPRECATED

Table 4.7: Training Set Mean G-mean

[H]

			Histotypes				
Subsampling	Algorithms	Overall	HGSC	CCOC	ENOC	LGSC	MUC
none	rf	0.261	0.868	0.89	0.755	0.223	0.903
	svm	0.54	0.903	0.893	0.827	0.453	0.897
	xgb	0	0.146	0.271	0	0	0
	mr	0	0	0	0	0	0
down	rf	0.798	0.897	0.897	0.806	0.884	0.895
	svm	0.809	0.878	0.853	0.85	0.934	0.875
	xgb	0	0	0	0	0	0
	mr	0.782	0.89	0.871	0.796	0.889	0.884
up	rf	0.421	0.887	0.893	0.789	0.403	0.874
	svm	0.699	0.907	0.859	0.775	0.715	0.859
	xgb	0.706	0.929	0.894	0.836	0.614	0.918
	mr	0.829	0.936	0.882	0.858	0.913	0.903
smote	rf	0.607	0.923	0.894	0.864	0.545	0.913
	svm	0.714	0.911	0.859	0.799	0.724	0.867
	xgb	0.778	0.944	0.912	0.849	0.759	0.91
	mr	0.801	0.927	0.884	0.872	0.814	0.909
hybrid	rf	0.749	0.936	0.891	0.872	0.676	0.917
	svm	0.804	0.933	0.878	0.859	0.863	0.884
	xgb	0.778	0.938	0.903	0.842	0.759	0.918
	mr	0.816	0.928	0.896	0.868	0.851	0.907

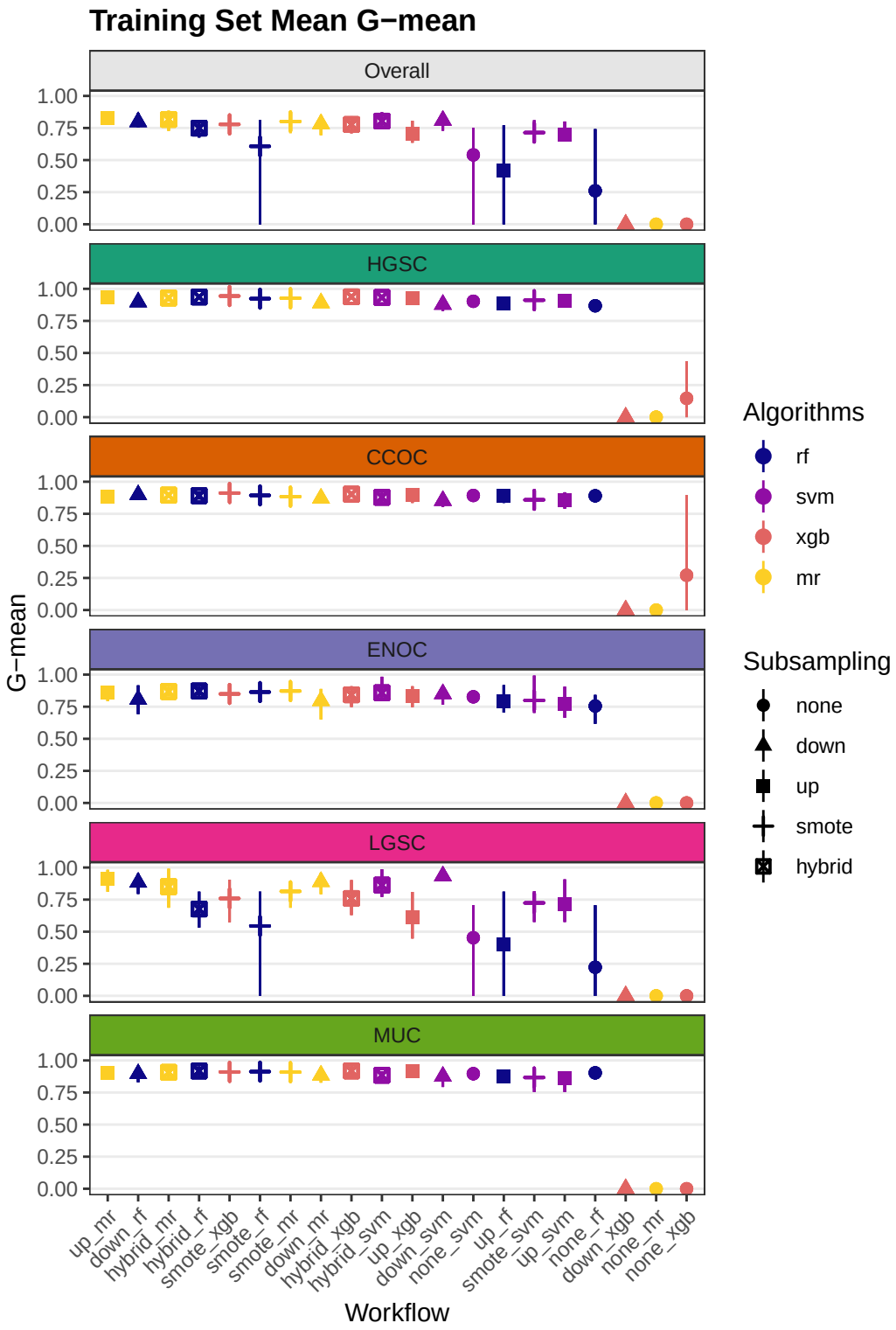


Figure 4.7: Training Set Mean G-mean

4.2 Rank Aggregation

Multi-step methods:

- **sequential**: sequential algorithm sequence of subsampling methods and algorithms used are:
  - HGSC vs. non-HGSC using SMOTE subsampling and random forest
  - CCOC vs. non-CCOC using hybrid subsampling and XGBoost
  - ENOC vs. non-ENOC using upsampling and support vector machine
  - LGSC vs. MUC using hybrid subsampling and regularized multinomial regression
- **two_step**: two-step algorithm sequence of subsampling methods and algorithms used are:
  - HGSC vs. non-HGSC using SMOTE subsampling and random forest
  - CCOC vs. ENOC vs. MUC vs. LGSC using hybrid subsampling and support vector machine

We conduct rank aggregation using a two-stage nested appraoch:

1. First we rank aggregate the per-class metrics for F1-score, balanced accuracy and kappa.
2. Then we take the aggregated lists from the three metrics and perform a final rank aggregation.
3. The top workflows from the final rank aggregation are used for gene optimization in the confirmation set

4.2.1 Across Classes

4.2.1.1 F1-Score

Table 4.8: F1-Score Rank Aggregation Summary

Workflow	Rank	HGSC	CCOC	ENOC	LGSC	MUC
sequential	1	0.969	0.855	0.891	0.919	0.967
two_step	2	0.969	0.865	0.738	0.782	0.864
smote_rf	3	0.978	0.837	0.716	0.37	0.84
none_svm	4	0.976	0.852	0.695	0.433	0.837
up_xgb	5	0.977	0.855	0.678	0.408	0.817
smote_xgb	6	0.973	0.845	0.663	0.528	0.806
hybrid_xgb	7	0.97	0.829	0.678	0.484	0.803
smote_svm	8	0.975	0.783	0.626	0.566	0.805
hybrid_rf	9	0.974	0.828	0.698	0.448	0.791
up_rf	10	0.969	0.829	0.657	0.278	0.803
up_svm	11	0.975	0.776	0.612	0.54	0.802
hybrid_mr	12	0.944	0.829	0.628	0.367	0.769
hybrid_svm	13	0.967	0.785	0.622	0.569	0.81
none_rf	14	0.968	0.849	0.622	0.262	0.838
smote_mr	15	0.945	0.797	0.631	0.334	0.793
up_mr	16	0.949	0.804	0.587	0.448	0.742
down_rf	17	0.907	0.813	0.545	0.255	0.719
down_mr	18	0.9	0.76	0.474	0.273	0.704
down_svm	19	0.884	0.728	0.486	0.275	0.744

4.2.1.2 Balanced Accuracy

Table 4.9: Balanced Accuracy Rank Aggregation Summary

Workflow	Rank	HGSC	CCOC	ENOC	LGSC	MUC
All	All	All	All	All	All	All
sequential	1	0.919	0.889	0.905	0.955	0.955
hybrid_xgb	2	0.938	0.907	0.853	0.79	0.92
smote_xgb	3	0.944	0.915	0.858	0.792	0.914
hybrid_rf	4	0.937	0.896	0.879	0.731	0.92
smote_rf	5	0.925	0.899	0.872	0.687	0.917
up_xgb	6	0.931	0.9	0.848	0.695	0.921
up_mr	7	0.937	0.888	0.864	0.917	0.906
hybrid_mr	8	0.928	0.901	0.873	0.861	0.911
smote_mr	9	0.927	0.889	0.877	0.827	0.913
two_step	10	0.919	0.893	0.819	0.924	0.908
hybrid_svm	11	0.933	0.885	0.867	0.874	0.89
none_svm	12	0.907	0.898	0.841	0.631	0.902
smote_svm	13	0.915	0.869	0.822	0.764	0.877
down_rf	14	0.899	0.902	0.821	0.887	0.9
down_mr	15	0.892	0.878	0.812	0.893	0.889
down_svm	16	0.882	0.862	0.855	0.936	0.882
up_rf	17	0.892	0.898	0.812	0.636	0.881
up_svm	18	0.911	0.869	0.801	0.761	0.87
none_rf	19	0.876	0.896	0.786	0.566	0.908
down_xgb	20	0.5	0.5	0.5	0.5	0.5
none_mr	21	0.5	0.5	0.5	0.5	0.5
none_xgb	22	0.528	0.601	0.5	0.5	0.5

4.2.1.3 Kappa

Table 4.10: Kappa Rank Aggregation Summary

Workflow	Rank	HGSC	CCOC	ENOC	LGSC	MUC
sequential	1	0.833	0.774	0.819	0.886	0.886
smote_rf	2	0.87	0.827	0.7	0.363	0.832
up_xgb	3	0.871	0.845	0.66	0.399	0.807
none_svm	4	0.862	0.843	0.679	0.342	0.829
two_step	5	0.833	0.796	0.632	0.758	0.818
smote_xgb	6	0.862	0.835	0.646	0.519	0.796
hybrid_rf	7	0.858	0.817	0.682	0.439	0.78
hybrid_xgb	8	0.846	0.817	0.661	0.475	0.793
smote_svm	9	0.862	0.77	0.606	0.56	0.796
up_svm	10	0.859	0.762	0.593	0.533	0.793
hybrid_svm	11	0.831	0.77	0.601	0.56	0.801
up_rf	12	0.819	0.818	0.641	0.269	0.793
none_rf	13	0.811	0.84	0.604	0.154	0.83
hybrid_mr	14	0.745	0.818	0.604	0.349	0.757
smote_mr	15	0.747	0.783	0.608	0.316	0.782
up_mr	16	0.766	0.79	0.562	0.434	0.728
down_rf	17	0.633	0.8	0.517	0.232	0.702
down_mr	18	0.613	0.744	0.439	0.251	0.687
down_svm	19	0.577	0.709	0.452	0.252	0.73
down_xgb	20	0	0	0	0	0
none_mr	21	0	0	0	0	0
none_xgb	22	0.083	0.24	0	0	0

4.2.2 Across Metrics

Table 4.11: Rank Aggregation Comparison of Metrics Used

[H]

Rank	F1	Balanced Accuracy	Kappa
1	sequential	sequential	sequential
2	two_step	hybrid_xgb	smote_rf
3	smote_rf	smote_xgb	up_xgb
4	none_svm	hybrid_rf	none_svm
5	up_xgb	smote_rf	two_step
6	smote_xgb	up_xgb	smote_xgb
7	hybrid_xgb	up_mr	hybrid_rf
8	smote_svm	hybrid_mr	hybrid_xgb
9	hybrid_rf	smote_mr	smote_svm
10	up_rf	two_step	up_svm
11	up_svm	hybrid_svm	hybrid_svm
12	hybrid_mr	none_svm	up_rf
13	hybrid_svm	smote_svm	none_rf
14	none_rf	down_rf	hybrid_mr
15	smote_mr	down_mr	smote_mr
16	up_mr	down_svm	up_mr
17	down_rf	up_rf	down_rf
18	down_mr	up_svm	down_mr
19	down_svm	none_rf	down_svm
20	NA	down_xgb	down_xgb
21	NA	none_mr	none_mr
22	NA	none_xgb	none_xgb

Table 4.12: Top 5 Workflows from Final Rank Aggregation

[H]

Rank	Workflow
1	sequential
2	smote_rf
3	two_step
4	none_svm
5	up_xgb

4.2.3 Top Workflows

We look at the per-class evaluation metrics of the top 5 workflows.

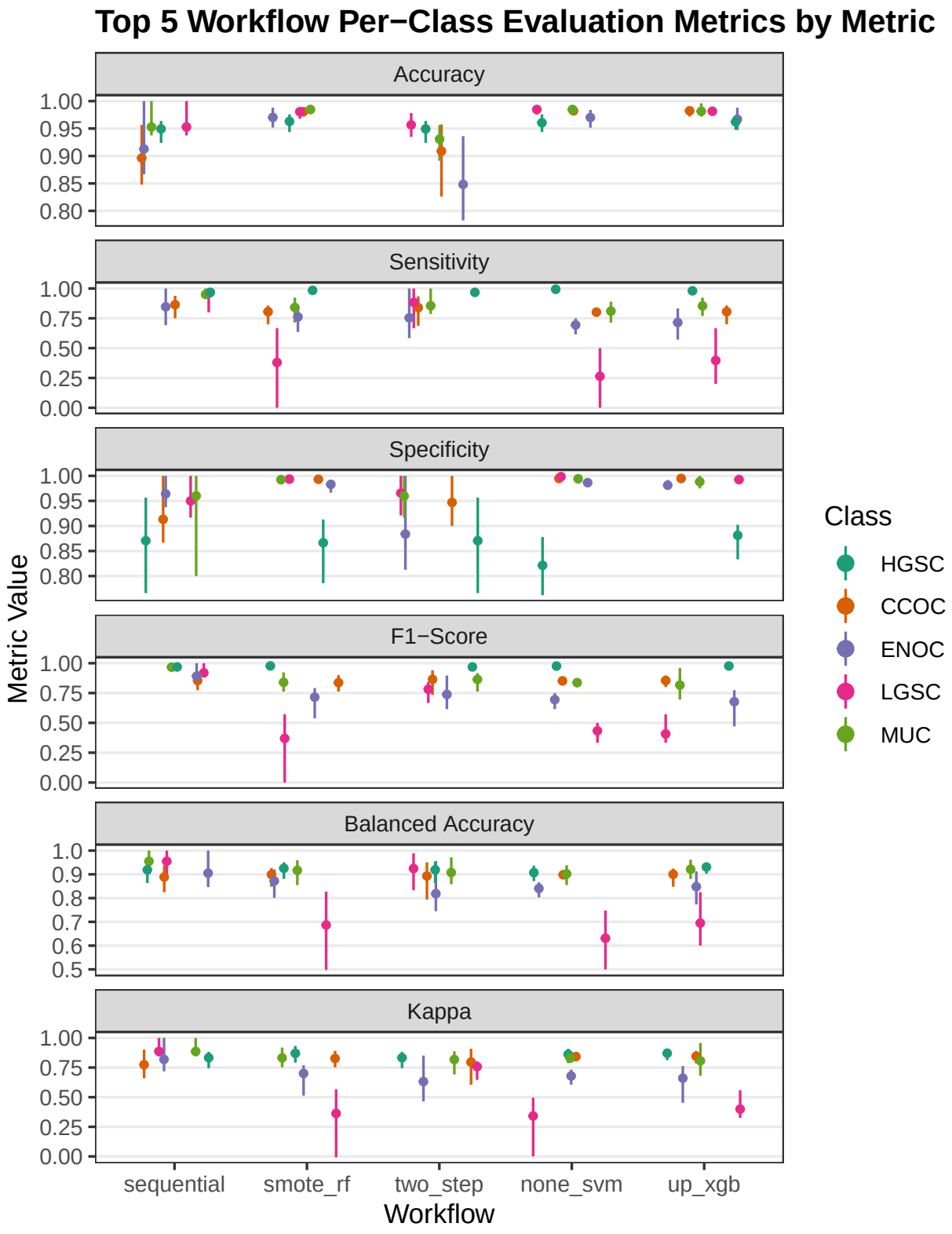
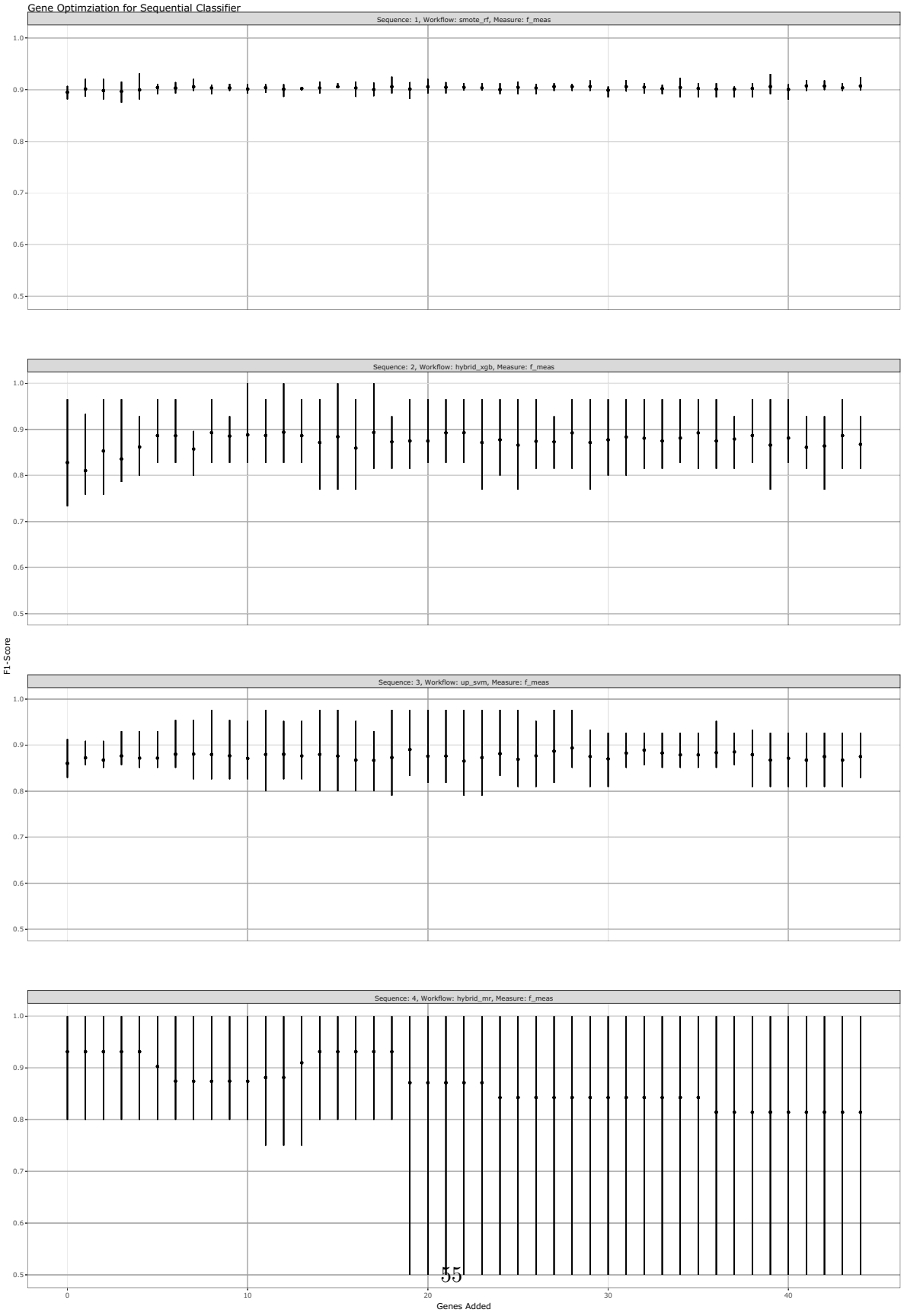


Figure 4.8: Top 5 Workflow Per-Class Evaluation Metrics by Metric

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



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 0.88 when we reach 7 genes added, hence the optimal number of genes used will be n=28+7=35 The added genes are: CYP2C18, TFF3, TP53, HNF1B, WT1, MAP1LC3A and SLC3A1.

Table 4.13: Gene Profile of Optimal Set in Sequential Algorithm

[H]

Set	Genes	PrOTYPE	SPOT	Optimal Set	Candidate Rank
Base	COL11A1	v		(*)	
	CD74	v		(*)	
	CD2	v		(*)	
	TIMP3	v		(*)	
	LUM	v		(*)	
	CYTIP	v		(*)	
	COL3A1	v		(*)	
	THBS2	v		(*)	
	TCF7L1	v	v	(*)	
	HMGA2	v		(*)	
	FN1	v		(*)	
	POSTN	v		(*)	
	COL1A2	v		(*)	
	COL5A2	v		(*)	
	PDZK1IP1	v		(*)	
	FBN1	v		(*)	
	HIF1A		v	(*)	
	CXCL10		v	(*)	
	DUSP4		v	(*)	
	SOX17		v	(*)	
	MITF		v	(*)	
	CDKN3		v	(*)	
	BRCA2		v	(*)	
	CEACAM5		v	(*)	
	ANXA4		v	(*)	
	SERPINE1		v	(*)	
	CRABP2		v	(*)	
	DNAJC9		v	(*)	
Candidate	CYP2C18			(*)	1
	TFF3			(*)	2
	TP53			(*)	3
	HNF1B			(*)	4
	WT1			(*)	5
	MAP1LC3A			(*)	6
	SLC3A1			(*)	7
	EPAS1			(*)	8
	EGFL6			(*)	9
	IL6			(*)	10
	TFF1			(*)	11
	BRCA1			(*)	12
	IGFBP1			(*)	13
	ATP5G3			(*)	14
	MUC5B			(*)	15
	SEMA6A			(*)	16
	FUT3			(*)	17
	MET			(*)	18
	GPR64			(*)	19
	ZBED1			(*)	20
	CPNE8			(*)	21
	SCGB1D2			(*)	22

4.3.2 SMOTE-Random Forest

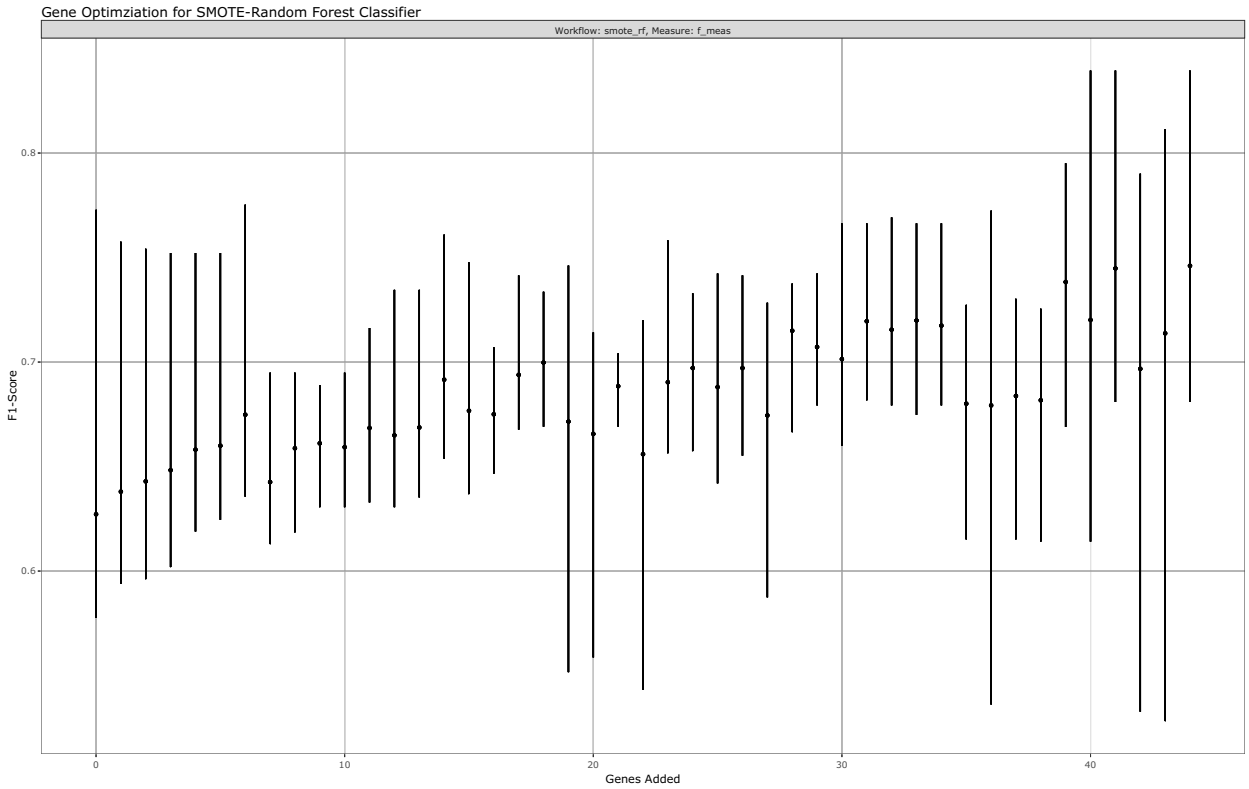


Figure 4.10: Gene Optimization for SMOTE-Random Forest Classifier

In the SMOTE-Random Forest classifier, the F1-score stabilizes at around 0.7 when we reach 18 genes added, hence the optimal number of genes used will be n=28+18=46 The added genes are: TFF1, HNF1B, TFF3, LGALS4, SLC3A1, WT1, KLK7, TPX2, CYP2C18, GAD1, IGFBP1, CAPN2, FUT3, DKK4, C1orf173, GCNT3, C10orf116 and MUC5B.

Table 4.14: Gene Profile of Optimal Set in SMOTE-Random Forest Workflow

[H]

Set	Genes	PrOTYPE	SPOT	Optimal Set	Candidate Rank
Base	COL11A1	v		(*)	
	CD74	v		(*)	
	CD2	v		(*)	
	TIMP3	v		(*)	
	LUM	v		(*)	
	CYTIP	v		(*)	
	COL3A1	v		(*)	
	THBS2	v		(*)	
	TCF7L1	v	v	(*)	
	HMGA2	v		(*)	
	FN1	v		(*)	
	POSTN	v		(*)	
	COL1A2	v		(*)	
	COL5A2	v		(*)	
	PDZK1IP1	v		(*)	
	FBN1	v		(*)	
	HIF1A		v	(*)	
	CXCL10		v	(*)	
	DUSP4		v	(*)	
	SOX17		v	(*)	
	MITF		v	(*)	
	CDKN3		v	(*)	
	BRCA2		v	(*)	
	CEACAM5		v	(*)	
	ANXA4		v	(*)	
	SERPINE1		v	(*)	
	CRABP2		v	(*)	
	DNAJC9		v	(*)	
	TFF1			(*)	1
	HNF1B			(*)	2
	TFF3			(*)	3
	LGALS4			(*)	4
	SLC3A1			(*)	5
	WT1			(*)	6
	KLK7			(*)	7
	TPX2			(*)	8
	CYP2C18			(*)	9
	GAD1			(*)	10
	IGFBP1			(*)	11
	CAPN2			(*)	12
	FUT3			(*)	13
	DKK4			(*)	14
	C1orf173			(*)	15
	GCNT3			(*)	16
	C10orf116			(*)	17
	MUC5B			(*)	18
	ATP5G3			(*)	19
	PAX8			(*)	20
	IL6			(*)	21
	GPR64			(*)	22

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 the sequential and SMOTE-Random Forest classifiers. We can test 2 additional methods by using either the full set of genes or the optimal set of genes for both of these classifiers.

4.4.1 Confirmation Set

Table 4.15: Evaluation Metrics on Confirmation Set Models

[H]

			Histotypes				
Method	Metric	Overall	HGSC	CCOC	ENOC	LGSC	MU
Sequential, Full Set	Accuracy	0.834	0.869	0.960	0.894	0.977	0.9
	Sensitivity	0.655	0.948	0.853	0.462	0.308	0.7
	Specificity	0.928	0.719	0.974	0.980	0.990	0.9
	F1-Score	0.664	0.905	0.831	0.590	0.348	0.6
	Balanced Accuracy	0.792	0.834	0.913	0.721	0.649	0.8
	Kappa	0.665	0.697	0.808	0.535	0.336	0.6
Sequential, Optimal Set	Accuracy	0.821	0.865	0.949	0.879	0.983	0.9
	Sensitivity	0.633	0.953	0.840	0.396	0.385	0.5
	Specificity	0.923	0.697	0.963	0.974	0.995	0.9
	F1-Score	0.659	0.902	0.792	0.519	0.476	0.6
	Balanced Accuracy	0.778	0.825	0.902	0.685	0.690	0.7
	Kappa	0.635	0.684	0.763	0.457	0.468	0.5
SMOTE-Random Forest, Full Set	Accuracy	0.843	0.871	0.969	0.896	0.980	0.9
	Sensitivity	0.659	0.962	0.867	0.453	0.308	0.7
	Specificity	0.928	0.697	0.982	0.983	0.994	0.9
	F1-Score	0.682	0.907	0.867	0.589	0.381	0.6
	Balanced Accuracy	0.793	0.829	0.925	0.718	0.651	0.8
	Kappa	0.677	0.697	0.849	0.535	0.371	0.6
SMOTE-Random Forest, Optimal Set	Accuracy	0.851	0.876	0.966	0.904	0.981	0.9
	Sensitivity	0.695	0.957	0.853	0.500	0.385	0.7
	Specificity	0.932	0.719	0.981	0.983	0.994	0.9
	F1-Score	0.715	0.910	0.853	0.631	0.455	0.7
	Balanced Accuracy	0.813	0.838	0.917	0.742	0.689	0.8
	Kappa	0.697	0.710	0.834	0.580	0.445	0.7

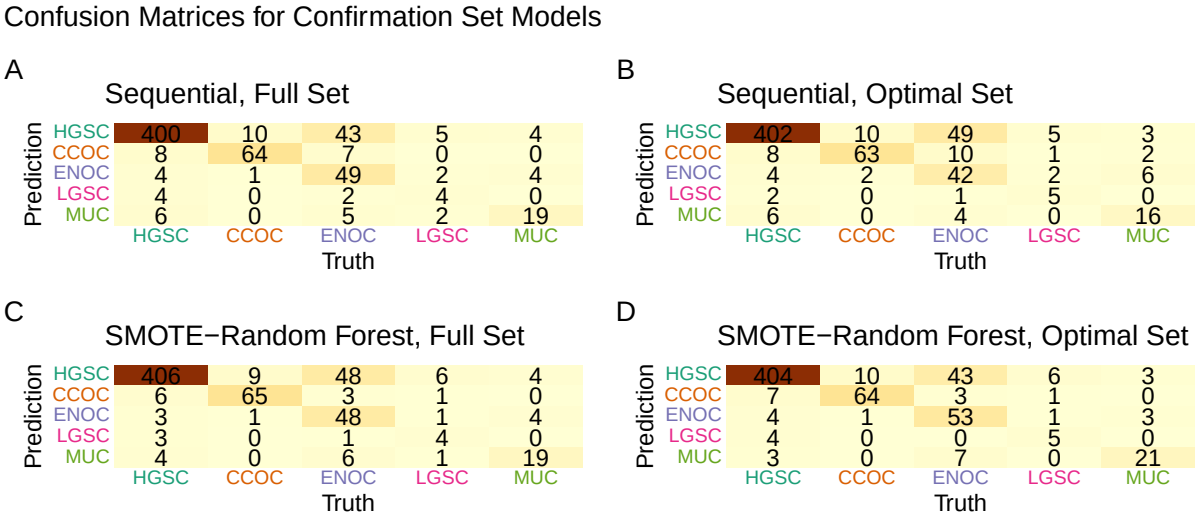


Figure 4.11: Confusion Matrices for Confirmation Set Models

4.4.1.1 Sequential, Full

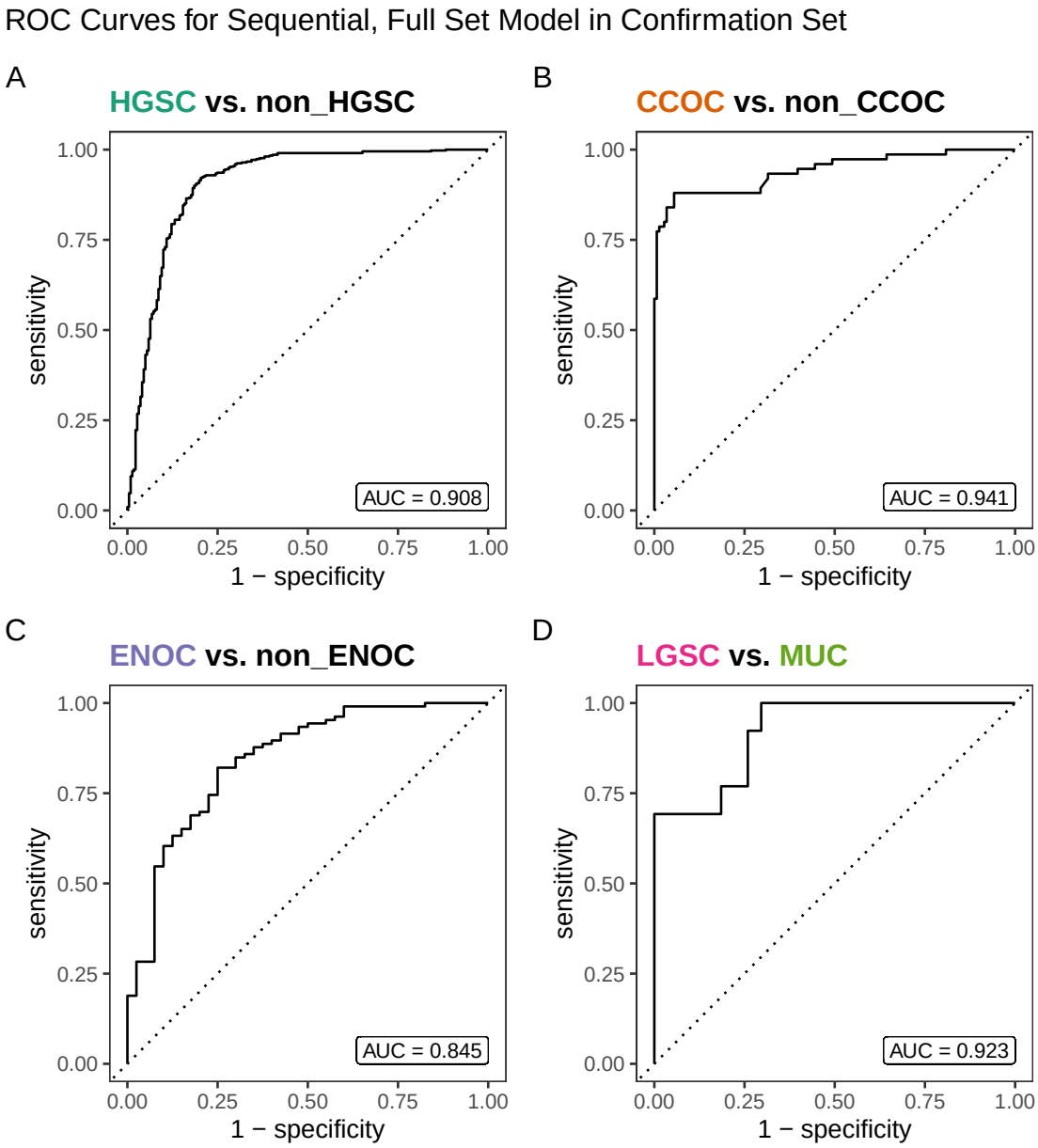


Figure 4.12: ROC Curves for Sequential Full Model in Confirmation Set

4.4.1.2 Sequential, Optimal

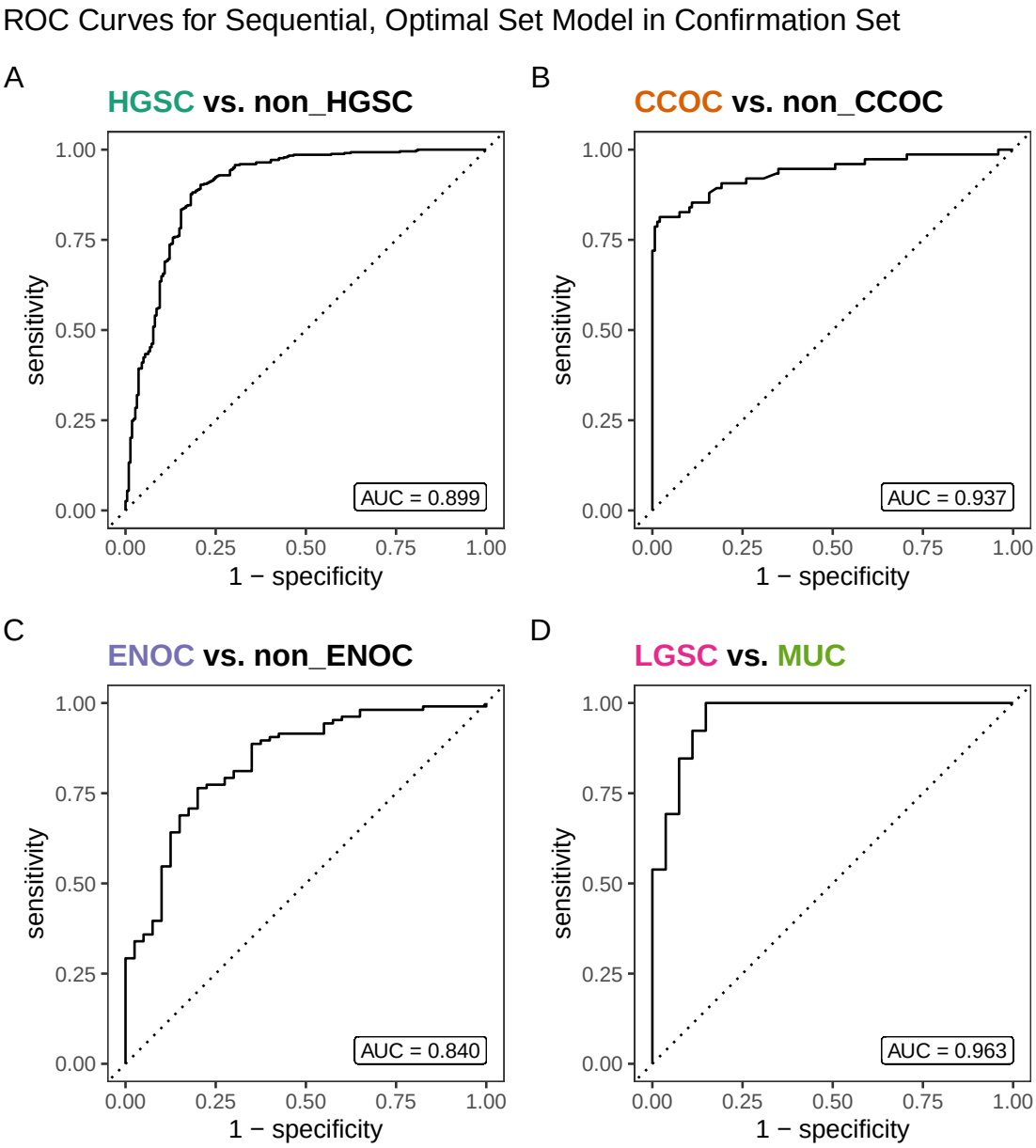


Figure 4.13: ROC Curves for Sequential, Optimal Model in Confirmation Set

4.4.1.3 SMOTE-Random Forest, Full

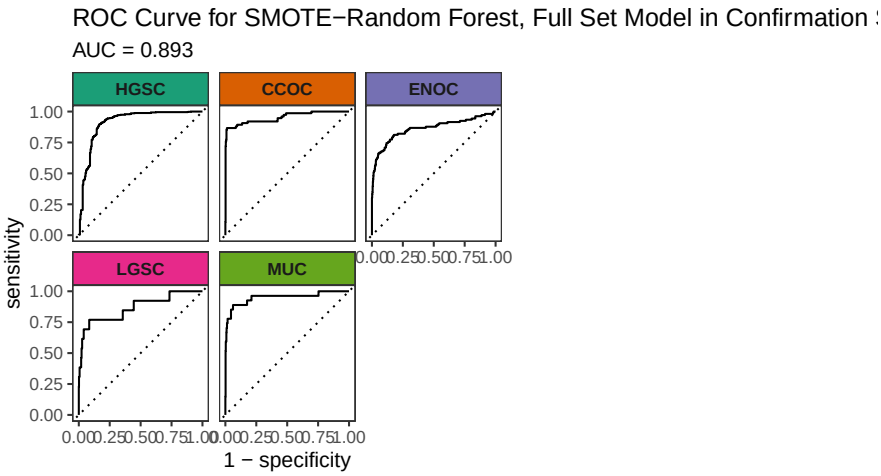


Figure 4.14: ROC Curves for SMOTE-Random Forest, Full Set Model in Confirmation Set

4.4.1.4 SMOTE-Random Forest, Optimal

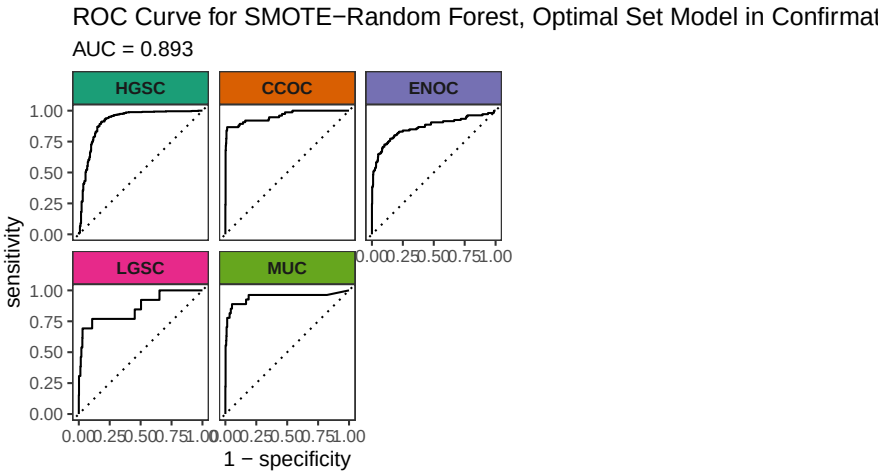


Figure 4.15: ROC Curves for SMOTE-Random Forest, Optimal Set Model in Confirmation Set

4.4.2 Validation Set

Table 4.16: Evaluation Metrics on Validation Set Model, SMOTE-Random Forest, Optimal Set

[H]

		Histotypes				
Metric	Overall	HGSC	CCOC	ENOC	LGSC	MUC
Accuracy	0.890	0.907	0.971	0.952	0.972	0.979
Sensitivity	0.774	0.926	0.937	0.714	0.444	0.846
Specificity	0.955	0.851	0.974	0.984	0.983	0.983
F1-Score	0.731	0.937	0.851	0.777	0.390	0.698
Balanced Accuracy	0.864	0.889	0.955	0.849	0.714	0.914
Kappa	0.748	0.761	0.835	0.750	0.376	0.688

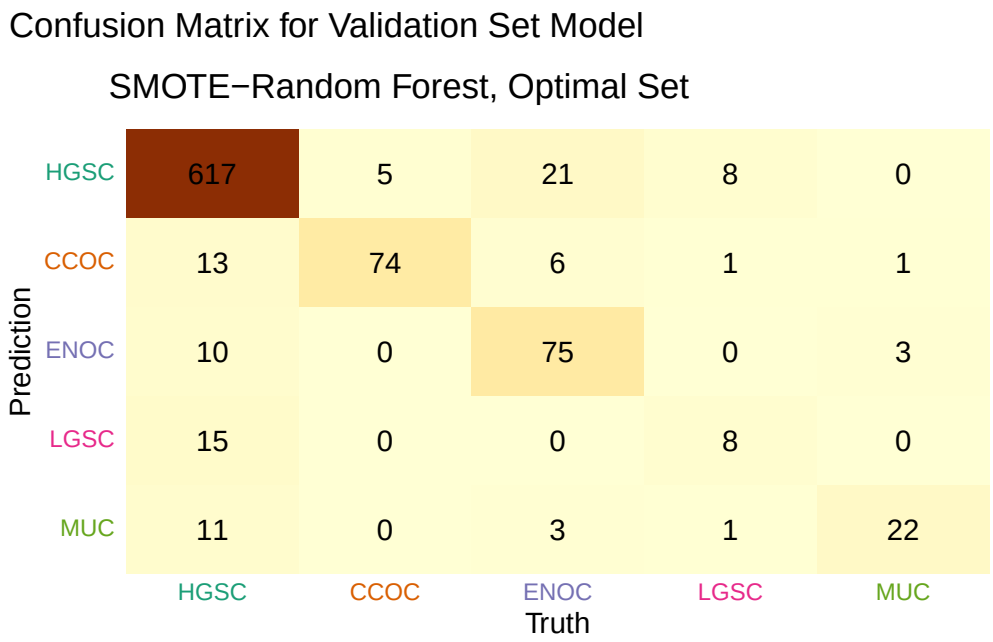


Figure 4.16: Confusion Matrix for Validation Set Model

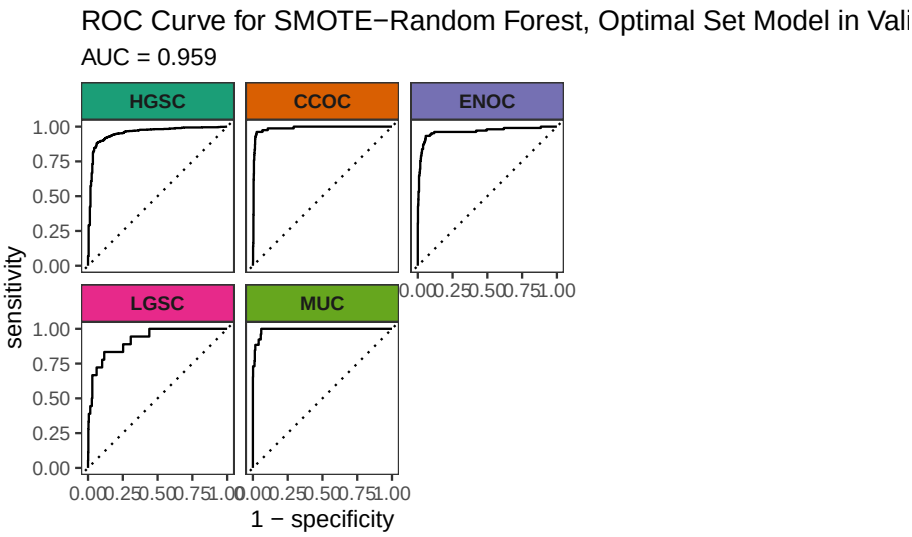


Figure 4.17: ROC Curves for SMOTE-Random Forest, Optimal Set Model in Validation Set

References

Talhouk, Aline, Stefan Kommoss, Robertson Mackenzie, Martin Cheung, Samuel Leung, Derek S. Chiu, Steve E. Kalloger, et al. 2016. "Single-Patient Molecular Testing with NanoString nCounter Data Using a Reference-Based Strategy for Batch Effect Correction." Edited by Benjamin Haibe-Kains. *PLOS ONE* 11 (4): e0153844. https://doi.org/10.1371/journal.pone.0153844.