Supplementary Table 1. Tuning parameters of the machine learning models.

Random Forest (RF)	**Parameters**	n_estimators=10 criterion=gini max_depth=None min_samples_split=2	min_samples_leaf=1 min_weight_fraction_leaf=0.0 max_features=sqrt(N) max_leaf_nodes=None	min_impurity_decrease=0.0 min_impurity_split=None bootstrap=True class_weight=None
	Notes	Gini impurity is used due to being less computationally intense than information gain. These parameters lead to fully grown and unpruned trees, which is memory intensive, given the high dimensionality of the data. The issue of overfitting is not a concern due to the random selection of variables.		
Gradient Boosting Machine (GBM)	**Parameters**	loss=deviance learning_rate=0.1 n_estimators=100 subsample=1.0 criterion=friedman_mse min_samples_split=2	min_samples_leaf=1 min_weight_fraction_leaf=0.0 max_depth=3 min_impurity_decrease=0.0 min_impurity_split=None	init=None max_features=n_features max_leaf_nodes=None presort=auto n_iter_no_change=None
	Notes	This model observes all features when searching for splits to attain high variance and low bias trees. Although this algorithm can be computationally intense, early stopping is disabled. There is a risk of overfitting the data but the learning_rate and n_estimators were limited to combat this drawback.		
Neural Network (NN)	**Parameters**	hidden_layer_sizes=(100,) activation=relu solver=adam alpha=0.0001 batch_size=auto learning_rate=constant	learning_rate_init=0.001 power_t=0.5 max_iter=200 shuffle=True tol=0.0001 momentum=0.9	early_stopping=False validation_fraction=0.1 beta_1=0.9 beta_2=0.999 epsilon=1e-08
	Notes	Four activation functions (identity, logistic, tanh, and relu) were tested and relu outperformed the other functions significantly when the genes were reduced. On the full dataset, there was no difference. Adam was used because of its computational efficiency and low memory requirements.		
K-Nearest Neighbour (KNN)	**Parameters**	n_neighbors=5 weights=distance algorithm=kd_tree	leaf_size=30 p=1	metric=minkowski metric_params=None
	Notes	When classifying the testing set with all features (16,718), the highest accuracy was achieved by n_neighbors=1. Only when the genes were reduced did we observe an improvement in accuracy at higher n_neighbors. There was a significant increase in accuracy from p=2 to p=1. Manhattan distance is preferable to Euclidean when processing high dimensional data. Less emphasis is also placed on outliers.		
Support Vector Machine (SVM)	**Parameters**	C=10 cache_size=200 class_weight=None	decision_function_shape=ovr gamma=auto kernel=rbf	max_iter=-1 shrinking=True tol=0.001
	Notes	C is set to 10 due to the imbalance of the classes for both datasets. Gamma is set to auto which means it is 1 / number of features.		