Random Forest Classifier for e1 Cluster Detection

Jane Swingler
jeswingler@usfca.edu

September 30, 2025

Contents

1 Audit of Original Database

1.1 Data Source

The dataset originates from a single-cell transcriptomics study (J. Craig Venter Institute; Aevermann et al., 2018). Each row corresponds to one cell. The label column (`Label`) encodes the class: `1` indicates that the cell belongs to the e1 excitatory neuron cluster, while `0` indicates that the cell does not belong to this cluster. The 608 feature columns record gene expression levels for different genes in each cell.

1.2 Dataset Characteristics

- **Total samples:** 871
- **Total features:** 608
- **Class counts**: 572 (class 0), 299 (class 1)
- **Minority ratio:** 0.3433
- **Missing values:** 0 across all columns
- **Samples-to-features ratio:** 1.43

1.3 Audit Checklist

- **Description of features & formats documented:** All 608 predictors are continuous gene-expression measurements (`numeric`). The class label is stored as a factor with levels `0/1`.
- **Verified with respect to ground truth:** The labels come from prior biological research, which defined the e1 cell type based on selective expression of marker genes (TESPA1, LINC00507, SLC17A7) and lack of KCNIP1 expression.
- **Demography coverage:** These are single-cell measurements. There is no information provided regarding the population from which the data was sourced.
- **Class balance:** Class counts are 572 (0) and 299 (1). The minority-class ratio is 0.3433 ($\approx 34.3\%$), which is well above the 10% threshold and thus no re-balancing is required.
- **Feature types:** All predictors are `numeric`; no categorical predictors are used. The label is a `factor`.
- **Missing values:** No missing values were detected and so no imputation is needed.
- **Samples vs. features sufficiency:** The samples-to-features ratio is $871/608 = 1.43$. This does not meet the generic "10× samples per feature" heuristic. We therefore use Random Forest with large `ntree` and OOB, and report feature importance.

- **Privacy considerations:** No personally identifiable information is present. All predictors are gene-expression values at the single-cell level.

Conclusion of audit: Data are well-typed (numeric features, factor label), complete (no missing values), and reasonably balanced across classes. The only notable consideration is the high-dimensional setting ($p = 608$ with 871 samples).

2 Creation of Training DB and Verification DB

2.1 Procedure for Splitting

From the original dataset, one positive sample (label = 1) and one negative sample (label = 0) were randomly selected and removed. These two samples formed the Verification Database (Verification DB). The remaining data constituted the Training Database (Training DB). A fixed random seed ensured reproducibility of this split.

Table 1: Summary of Training and Verification Databases

Database	Samples	Features	Class 0	Class 1	Minority Ratio
Training DB	869	608	571	298	0.3429
Verification DB	2	608	1	1	0.5000

2.2 Training Database Summary

The Training DB contains 869 total samples with 608 features. The class distribution is 571 negatives (label = 0) and 298 positives (label = 1). The minority class ratio is 0.3429 ($\approx 34.3\%$), which indicates that the dataset is reasonably balanced.

2.3 Verification Database Summary

The Verification DB consists of exactly two records, one positive and one negative. These samples were withheld from training and later used to evaluate the runtime Random Forest classifier. The minority ratio is 0.5000 because both classes are equally represented in this very small set.

3 SW Tools

3.1 Programming Environment

All experiments were conducted in the `R` programming language, version 4.4.0 ("Puppy Cup" release, April 24, 2024). The Kaggle hosted Jupyter Notebook environment was used, running on Ubuntu 22.04.4 LTS (x86_64-pc-linux-gnu). Matrix computations used the OpenBLAS implementation of BLAS and LAPACK (v3.10.0).

The main R packages employed were:

- `randomForest_4.7-1.1` – training Random Forest classifiers, computing OOB error, and extracting feature importance.
- `caret_6.0-94` – generating confusion matrices and accuracy measures (precision, recall, F1).
- `pROC_1.18.5` – ROC curve and AUC calculations.
- `readr_2.1.5` – CSV import.

4 Experimental Methods and Setup

The Random Forest classifier was trained using the Training Database (original dataset minus two verification samples). A grid search was conducted across selected ranges of the main hyperparameters: number of trees (`ntree`), number of features considered at each split (`mtry`), and class cutoff values (`cutoff`). Larger values of `ntree` were used in accordance with class guidelines.

4.1 Hyperparameter Grid

Parameter	**Values Tested**
`ntree`	1000, 2000, 4000, 8000
`mtry`	$\lfloor 0.5\sqrt{p} \rfloor$, $\lfloor \sqrt{p} \rfloor$, $\lfloor 2\sqrt{p} \rfloor$ (with $p = 608$ features)
`cutoff`	(0.7, 0.3), (0.5, 0.5), (0.3, 0.7)

Table 2: Grid of Random Forest hyperparameters used in training.

The grid search tested values of `ntree` = 1000, 2000, 4000, and 8000, providing both moderate and large forests to observe error convergence and ensure stability of OOB estimates. The global minimum OOB error was observed at `ntree` = 1000. However, because the dataset has a large number of features relative to samples, a larger forest helps exercise more feature combinations and stabilize results. Therefore, the final model was retrained with `ntree` = 8000 to ensure robust and reliable OOB estimates.

4.2 Accuracy Estimation

Model accuracy was evaluated using the Out-of-Bag (OOB) error estimate during training. OOB was chosen because it is built into the Random Forest algorithm and provides an unbiased estimate of test error without requiring explicit cross-validation folds.

4.3 Accuracy Metrics

The following standard classification metrics were computed from the best model's OOB confusion matrix, using raw counts rather than normalized values. All measures are reported to five decimal places.

Metric	Definition
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Recall (Sensitivity)	$\frac{TP}{TP+FN}$
Precision	$\frac{TP}{TP+FP}$
F1 Score	$2 \cdot \frac{Precision \cdot Recall}{Precision+Recall}$

Table 3: Accuracy measures used to evaluate Random Forest models.

4.4 APIs and Functions

The Random Forest training and evaluation were implemented using the following R functions and APIs:

- `randomForest()` (from the `randomForest` package) – core function used to train Random Forest classifiers with specified `ntree`, `mtry`, and `cutoff` hyperparameters.
- `predict()` – applied to trained Random Forest models for generating class predictions and class probabilities on both the Training DB and Verification DB.
- `confusionMatrix()` (from the `caret` package) – used to compute confusion matrices and extract Accuracy, Precision, Recall, and F1 metrics.
- `importance()` (from the `randomForest` package) – used to extract feature ranking by Mean Decrease Accuracy (MDA).
- `read_csv()` (from the `readr` package) – used for efficient import of the original dataset from CSV format.

5 Results of RF Training and Accuracy Estimates

5.1 Results Across Parameter Settings

The grid search was conducted across combinations of `ntree`, `mtry`, and `cutoff`. Values of `ntree` = 1000, 2000, 4000, and 8000 were tested. The cutoff values $(0.7, 0.3)$, $(0.5, 0.5)$, and $(0.3, 0.7)$ were compared. Among the tested settings, the balanced threshold $(0.5, 0.5)$ achieved the lowest OOB error. For the number of variables tried at each split, the setting $\lfloor \sqrt{p} \rfloor = 12$ yielded the best performance.

5.2 Best Model Summary

The global minimum OOB error occurred with the following parameter combination:

- **ntree:** 1000
- **mtry:** 12
- **cutoff:** (0.5, 0.5)

- **OOB error rate:** 0.00345 (0.35%)

The corresponding OOB confusion matrix and derived metrics are shown below.

	Predicted 0	Predicted 1
Actual 0	568	3
Actual 1	0	298

Table 4: OOB confusion matrix for best RF model (ntree = 1000, mtry = 12, cutoff = (0.5, 0.5)). Rows represent true class labels; columns represent predicted labels.

Metric	**Value**
Accuracy	0.99655
Precision	0.99003
Recall	1.00000
F1 Score	0.99499

Table 5: Performance metrics computed from OOB confusion matrix of best model. All values reported to five decimal places.

5.3 Analysis of Results

The model achieved very high performance. Recall was equal to 1.00000, meaning no positive (e1 cluster) samples were missed. Three false positives were recorded, which reduced precision to 0.99003. Overall accuracy was 0.99655 and the F1 score was 0.99499. The OOB error rate of 0.00345 indicates a very low overall misclassification rate.

6 Feature Ranking

6.1 Method Used (MDA for R)

Feature importance was computed using the Mean Decrease in Accuracy (MDA) method, implemented by the `importance()` function in R's `randomForest` package. This approach permutes each feature's values in the Out-of-Bag (OOB) samples and measures the reduction in model accuracy. Features that cause a larger decrease in accuracy when permuted are ranked as more important. For this step, the model was retrained with `ntree` = 8000, `mtry` = 12, and `cutoff` = (0.5, 0.5), with `importance=TRUE` enabled to generate stable MDA scores.

6.2 Top 10 Features

The ten most important features, ranked by Mean Decrease in Accuracy, are shown below.

Feature	MDA Score
TESPA1	32.59954
SLC17A7	29.77607
LINC00507	28.05387
KCNIP1	27.17743
ANKRD33B	26.98186
SLIT3.1	26.86996
SLIT3.2	26.65708
SLIT3	26.47532
SFTA1P	26.30224
NRGN	26.12710

Table 6: Top 10 features ranked by Mean Decrease in Accuracy (MDA) from the final RF model with ntree = 8000.

6.3 Analysis and Discussion

The feature ranking results show clear correspondence with the biological ground truth for the e1 cluster:

  - **Ground truth markers:** e1 excitatory neurons express **TESPA1**, **LINC00507**, and **SLC17A7**, and lack expression of **KCNIP1**.
  - **Model agreement:** The model ranked **TESPA1**, **LINC00507**, and **SLC17A7** as the top three features. The lack of expression of **KCNIP1** was also highly predictive of e1 cells, explaining its strong feature importance.

- **Other top features:** ANKRD33B, the SLIT3 family genes, SFTA1P, and NRGN were not highlighted in the biological ground truth but were identified by the model as important. These may represent co-expressed or correlated markers within the dataset.
- **Feature dominance:** The highest ranked features (TESPA1, LINC00507, SLC17A7, KCNIP1) stand out from the rest, indicating that a relatively small subset of genes is sufficient to define the e1 cluster with high accuracy.

In summary, the Random Forest feature ranking not only supported high predictive accuracy but also aligned closely with known biological markers of the e1 excitatory neuron cluster, demonstrating both predictive performance and biological interpretability.

7 RF Run Time Test

7.1 Runtime RF Engine

After identifying the optimal hyperparameters and training the final Random Forest model with `ntree` = 8000, the classifier was saved using `saveRDS()` and subsequently reloaded as a runtime prediction engine. At runtime, predictions are generated by passing new samples

through all trees in the forest, and the final decision is obtained by majority vote. In addition, the model outputs class probabilities corresponding to the fraction of trees voting for each class (0 = negative, 1 = positive).

7.2 Predictions on Verification DB

Two samples, one positive (class 1) and one negative (class 0), were held out from training to serve as a verification database. Table 7 shows the predicted class and associated probabilities compared to the ground truth labels.

True Label	Predicted Class	Prob0	Prob1
1	1	0.02863	0.97138
0	0	0.94438	0.05563

Table 7: Runtime predictions on the two verification samples. Probabilities represent the fraction of trees voting for each class.

7.3 Discussion of Confidence

The runtime RF engine correctly classified both verification samples. The positive sample (class 1) was predicted correctly with a high probability of 0.97138, reflecting strong confidence in the positive prediction. The negative sample (class 0) was also predicted correctly with a probability of 0.94438, again indicating high confidence in the decision. These results demonstrate that the trained Random Forest model generalizes well to unseen data, maintaining both accuracy and probability calibration when applied in a runtime setting. The high probability margins suggest that the model is not only accurate but also confident in distinguishing between the two classes.

8 References

1. Aevermann, B. D., Novotny, M., Bakken, T., Miller, J. A., Diehl, A. D., Osumi-Sutherland, D., Lasken, R., Lein, E., & Scheuermann, R. H. (2018). Cell type discovery using single cell transcriptomics: implications for ontological representation. *Human Molecular Genetics, 27*(R1), R40–R47.

2. J. Craig Venter Institute. Retrieved September 28, 2025, from `https://www.jcvi.org/`