**20.440 ANALYSIS PIPELINE**

**Part 1.1: Dataframe organization**

We will take the **raw counts of RNA** (non-normalized) from the original RNA assay and the **raw counts of L-Pha** from the original ADT assay.

We will filter cells for **T Cells only**, and each cell should **be paired with its T cell subtype (from ProjecTIL analysis.**

**End result:** we will create a **pandas dataframe** in this structure

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Cell 1 barcode | Cell 2 barcode | (…) | Cell n  (= ~ 19,000)  barcode |
|  | T cell subtype | T cell subtype | T cell subtype | T cell subtype |
| Gene 1 | count # | count # | count # | count # |
| Gene 2 | count # | count # | count # | count # |
| (…) | count # | count # | count # | count # |
| Gene n | count # | count # | count # | count # |
| L-Pha (Biotin) | count # | count # | count # | count # |

* Data frame should contain: two column labels (one for barcode, one for T cell subtype)
* Row labels of all genes, with last row being L-Pha (“Biotin” in ADT matrix)

**Part 1.2: Dataframe filtering for glycogenes only**

We will make a copy of dataframe

We will filter the dataframe so that if the row names do not match to a list of glycogenes + L-Pha, they are removed from dataframe

**End result (new pandas df:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Cell 1 barcode | Cell 2 barcode | (…) | Cell n  (= ~ 19,000-300)  barcode |
|  | T cell subtype | T cell subtype | T cell subtype | T cell subtype |
| Glycogene 1 | count # | count # | count # | count # |
| Glycogene 2 | count # | count # | count # | count # |
| (…) | count # | count # | count # | count # |
| Glycogene 230ish | count # | count # | count # | count # |
| L-Pha | count # | count # | count # | count # |

**Part 2: Making new data frames organized for ML pipeline**

**Part 2.1: Global sort for model of all T cells**

We will take dataframe result of 1.2, create a copy, and only keep the cells where L-Pha counts are in the bottom 25% or top 25% values. We will add another row to the bottom of the dataframe. If the value is in the bottom 25%, we will add a 0 to that row, and if the L-Pha value is in the top 25%, we will add a 1 to the new row.

**Final result:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Cell 1 barcode | Cell 2 barcode | (…) | Cell n  (= ~ 8000)  barcode |
|  | T cell subtype | T cell subtype | T cell subtype | T cell subtype |
| Glycogene 1 | count # | count # | count # | count # |
| Glycogene 2 | count # | count # | count # | count # |
| (…) | count # | count # | count # | count # |
| Glycogene 230ish | count # | count # | count # | count # |
| L-Pha | count # | count # | count # | count # |
| 0 or 1 | 0 | 1 | 0 | 1 |

**Part 2.2: Local sort for model of T cell subtypes**

We will take dataframe result of 1.2, create a copy, and only keep the cells for **T cell Subtype X.**

We will copy this dataframe, and keep cells with L-Pha counts in the bottom 25% or top 25% values. We will add another row to the bottom of the dataframe. If the value is in the bottom 25%, we will add a 0 to that row, and if the L-Pha value is in the top 25%, we will add a 1 to the new row.

L-Pha counts are in the bottom 25% or top 25% values. We will add another row to the bottom of the dataframe. If the value is in the bottom 25%, we will add a 0 to that row, and if the L-Pha value is in the top 25%, we will add a 1 to the new row.

**Final result:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Cell 1 barcode | Cell 2 barcode | (…) | Cell n  (= few hundred)  barcode |
|  | **T cell Subtype X** | **T cell Subtype X** | **T cell Subtype X** | **T cell Subtype X** |
| Glycogene 1 | count # | count # | count # | count # |
| Glycogene 2 | count # | count # | count # | count # |
| (…) | count # | count # | count # | count # |
| Glycogene 230ish | count # | count # | count # | count # |
| L-Pha | count # | count # | count # | count # |
| 0 or 1 | 0 | 1 | 0 | 1 |

**Part 3: Create random forest classifier on GLOBAL SET and evaluate**

**Big picture:**

* We will use scikit learn package in python
* Our model will be trained to take counts of glycogenes as input, with a paired outcome as 1 or 0, to predict outcome of 1 or 0 given a glycogene input. **Thus, our model should predict the “1 or 0” row value from row values corresponding to glycogenes 1-230ish, with each column being one data/event.**

1. Using **dataframe from 2.1**, We will divide our columns into training and test sets, and evaluate our model (can test different arguments) with the same metrics used in the other ML paper (ROC, AUC, etc).
2. If the result is unsatisfactory, we will look for strong correlations of glycogenes to other genes in the original dataset, append them to our list, and try again.

**Part 4: Create random forest classifier on LOCAL SET and evaluate**

We will use the same model and arguments obtained for our global set and apply to **dataframes from 2.2.** The accuracy of model for different subsets will give us insights into T cell subset glycobiology.

**Part 5: Bringing in another dataset**

We will take another dataset of **T cells in the TME with and without perturbation (some immunotherapy drug).**

We will compare the results of our model on T cell subsets with and without perturbation, to understand how perturbation (some immunotherapy drug) may be influencing the glycan coat.