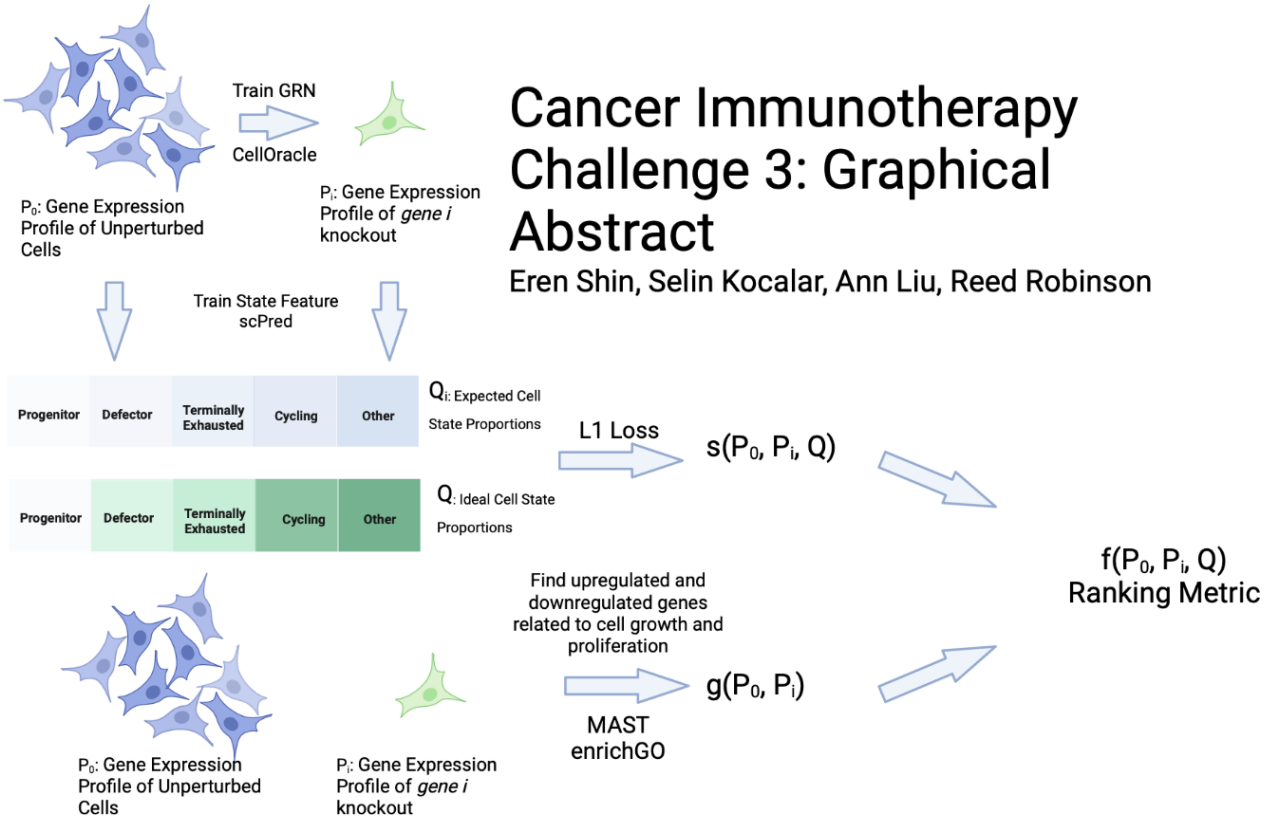

Part 1: Developing a Method for Comparing Predicted and Ideal Cell State Proportions

In order to develop a metric that compares the predicted (Q_i) and ideal cell state proportions (Q), we propose a computationally efficient method that relies on L1 loss to favor predicted cell state proportions that are closest to the ideal cell state proportions.

By determining the cell state proportions of the T-cells for each gene knockout-perturbation, we will evaluate which perturbation correlates with a greater immune response against cancer cells. Specifically, the cells in the “Progenitor” state have the ability to proliferate and differentiate into T-cells, replenishing the T-cell population as existing T-cells become terminally exhausted from fighting cancer cells. Thus, perturbations that maximize the proportion of the “Progenitor” cell state are expected to be more effective in fighting cancer cells. With this information determined *in-silico*, labs can easily implement our metric to choose which genes to test as knock-out perturbations.

In order to implement our similarity metric, we will use P_0, the gene expression matrix of unperturbed cells; P_i, the gene expression matrix of perturbed cells with the knockout of gene i; and Q_0, the 5-component vector of cell state proportions for the unperturbed cells. First, we will use CellOracle (Kamimoto et al., 2020) to train our `base_GRN` based on the “state” feature of the unperturbed cells, gene expression profiles, and use CellOracle’s `simulate_shift`

function to generate P_i. We will then use scPred (Alquicira-Hernandez et al., 2019) to train on P_0, in order to predict the “state” feature of each cell in P_i, which will give us the desired 5-component cell state vector, Q_i after simulating knockout of gene i.

Now that we have obtained Q and Q_i, the two matrices for the ideal and predicted cell state proportions, we will calculate the L1 loss between them. L1 loss is not only computationally efficient, but it also allows us to conserve computational power by not using a loss metric that utilizes normalization. Since our vector components have already been scaled to have a sum of 1, alternative loss methods, such as calculating Pearson distance, which incorporate additional normalization steps, are unnecessarily computationally intensive and do not provide valuable insight necessary to justify that computation.

Because predicted cell state proportions that are closer to the ideal cell state proportions will have a smaller L1 loss, we define $s(P_0, P_i, Q)$ = 2 - (L1 loss) so that $s(P_0, P_i, Q)$ ranges from 0 to 2 and is positively correlated with the similarity of our prediction to the ideal cell state proportions.

Ultimately, our algorithm provides a simple method to compare *in-silico* knockouts or perturbed cells to an ideal vector of cell states, Q. Our method is only computationally efficient, but it also allows for tremendous flexibility in Q (i.e. the values of Q will not change the implementation of the L1 loss comparison). Thus, $s(P_0, P_i, Q)$ is a summary statistic that uses a predicted cell state proportion vector, Q_i, similar to in Challenge 2, but importantly, $s(P_0, P_i, Q)$ is derived from training CellOracle on just the unperturbed cells, P_0, whereas our approach in Challenge 2 trained CellOracle on all cells (all 64 perturbed gene conditions, plus the unperturbed cells). In both cases, only the unperturbed cells were used for `simulate_shift` to obtain P_i, but training CellOracle on just the unperturbed cells to get $s(P_0, Pi, Q)$ takes into account differences in sample sizes between P_0 and P_i, as both gene expression matrices have the same shape.

Part 2: Metric for Selecting Knockouts of Interest

In order to take into account the differences in growth rates and number of cells caused by different gene perturbations, we propose a scoring function $f(P_0, P_i, Q)$ that scales the $s(P_0, P_i, Q)$ statistic described in Part 1 by a statistic $g(P_0, P_i)$ that measures the difference in growth rates between P_0 and P_i such that greater values of $g(P_0, P_i)$ represent higher predicted growth rates of cells in P_i.

$g(P_0, P_i)$ ranges from 0 to 1 and will be calculated as follows:

1. Using the MAST package in R (Finak et al., 2015), find up-regulated and down-regulated genes in P_i compared to P_0. This step can be done either by using the full P_0 and P_i gene expression matrices, or by subsetting each gene expression matrix by the cell states and comparing each cell state in P_0 to the respective cell state in P_i.

2. Filter the up-regulated and down-regulated genes from MAST using a p-value threshold of 0.05 (other threshold values can also be tested for significance), and use the `enrichGO` function in the clusterProfiler package in R to find enriched GO terms based on the sets of up-regulated and down-regulated genes (Ashburner et al., 2000; Wu et al., 2021). Filter the GO terms based on adjusted p-value < 0.05, and choose GO terms that relate to growth/proliferation, based on cell growth/proliferation-associated GO classes (GO ID: 0008283, 0016049, 0051301, 0048869) (Gene Ontology Consortium).

3. Once you have these GO terms, look at the "geneID" column of the output of `enrichGO` in order to find the list of genes that are involved in that GO term. Take the union of all genes involved in GO terms related to growth/proliferation, and compute the gene ratio, which we define as the size of the union of all genes divided by the size of the gene set (the total number of up-regulated or down-regulated genes). The $g(P_0, P_i)$ statistic is equal to the gene ratio in most cases.

4. In the case where MAST does not detect any significantly up-regulated or down-regulated genes in P_i relative to P_0 or `enrichGO` does not detect any significantly enriched GO terms relating to growth/proliferation, we assume that P_i does not differ significantly from P_0 in cell growth rate. In this case, the $g(P_0, P_i)$ statistic can be set to a default value of 0.1 since calculating $g(P_0, P_i)$ as described in step 3 will result in a $g(P_0, P_i)$ value of 0. Different default values can be tested to find the optimal value.

Once $g(P_0, P_i)$ is calculated, the scoring function $f(P_0, P_i, Q)$ is defined as $g(P_0, P_i) \times s(P_0, P_i, Q)$, which ranges from 0 to 2 and is still positively correlated with the similarity of our predicted cell state vector to the ideal cell state vector Q, as in Part 1, but also takes into account transcriptional differences in cell growth that can be induced by knocking out gene i.

References

Alquicira-Hernandez, J., Sathe, A., Ji, H.P., Nguyen, Q., Powell, J.E. (2019). scPred: accurate supervised method for cell-type classification from single-cell RNA-seq data. *Genome Biol* 20, 264. https://doi.org/10.1186/s13059-019-1862-5

Ashburner et al. Gene ontology: tool for the unification of biology. *Nat Genet.* May 2000;25(1):25-9.

Finak, G., McDavid, A., Yajima, M., Deng, J., Gersuk, V., Shalek, A.K., Slichter, C.K., Miller, H.W., McElrath, M.J., Prlic, M., Linsley, P.S., Gottardo, R. (2015). MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biol* 16, 278. https://doi.org/10.1186/s13059-015-0844-5.

Gene Ontology Consortium. (2015). Gene ontology consortium: going forward. *Nucleic acids research*, 43(D1), D1049-D1056.

Kamimoto, K., Hoffmann, C.M., and Morris, S.A. (2020). CellOracle: dissecting cell identity via network inference and in silico gene perturbation. *bioRxiv.* https://doi.org/10.1101/2020.02.17.947416.

The Gene Ontology resource: enriching a GOld mine. *Nucleic Acids Res.* Jan 2021;49(D1):D325-D334.

Wu, T., Hu, E., Xu, S., Chen, M., Guo, P., Dai, Z., Feng, T., Zhou, L., Tang, W., Zhan, L., Fu, X., Liu, S., Bo, X., Yu, G. (2021). “clusterProfiler 4.0: A universal enrichment tool for interpreting omics data.” *The Innovation*, 2(3), 100141. doi: 10.1016/j.xinn.2021.100141.