

# A graph-based practice of evaluating collective identities of cell clusters

Yuji Okano<sup>1,2</sup>, Yoshitaka Kase<sup>2</sup>, and Hideyuki Okano<sup>2</sup>

<sup>1</sup> Department of Extended Intelligence for Medicine, The Ishii-Ishibashi Laboratory, Keio University School of Medicine

<sup>2</sup> Division of CNS Regeneration and Drug Discovery, International Center for Brain Science, Fujita Health University

March 13, 2024

## Abstract

Random sentences

## Introduction

It has been more than 10 years since the birth of single-cell RNA-sequencing (scRNA-seq)[1], and the technology now is recognized as a prominent game changer of the molecular biology of this decade. Likewise the pioneering technologies, DNA micro array and bulk RNA-seq, scRNA-seq can observe multidimensional gene expression profiles, while it also can provide such information in single-cell-level. Although this informative assay have contributed to reveal detailed biology of various cell types, the excessive resolution blurred the conceptual boundary between static cell types and transient cell status[2]. Consequently, clusters, chunks of samples that shares similar geometrical properties in the data space overwrote the classical notion of cell types. As the cell clusters are dependent on sampling stochasticity of the dataset, and their biological properties might sway from the original doctrine of cell types[3]. Hence, a theoretical backbone and a effective method to glue the theory and real data are essential to identify universal characters of specific samples from piles of extrinsic noises.

In our previous research, we proposed a gene regulatory network (GRN)-based representation of cell clusters while edges of GRNs explain statistical dependencies between two genes, and demonstrated that similarity of two clusters can be defined as a quasi-pseudo-metric function  $d^*$ [3]. To discuss mathematical properties of the space of cell clusters, we defined novel terms, cell class and eigen-cascades, and step-by-step introduced their algebraic structures. Eigen-cascades refer to a set of marker genes and pairs of genes that are statistically dependent (i.e., isomorphic to the direct sum of the direct sums of the vertex set and the edge set of a GRN). A cell class refers to a cell cluster characterized with the corresponding eigen-cascades. Note that the nuances of cell clusters and cell classes are slightly different even though we might use those terms interchangeably in this article (See Appendix for more descriptions). When two cell classes  $\forall [x], [y]$  are represented by the GRNs regarding a set of genes  $G$ , and the two GRNs (eigen-cascades) respectively denoted as  $C_{[x]}(G)$  and  $C_{[y]}(G)$ , a bivariate function  $d^*$  that maps a pair of cell classes to real numbers are defined as follows[3]:

$$d^*([x], [y]) := 1 - \frac{|C_{[x]}(G) \cap C_{[y]}(G)|}{|C_{[x]}(G)|}. \quad (1)$$

Eq. (1) is derived from the Hamming distance function (a metric function that measures the difference of two character strings) and modified to embrace the tendency of the Peter and Clark (PC) algorithm, one of the most simple bayesian network algorithms[3, 4, 5]. With those concepts, we also proposed frameworks to compare the similarities of given two cell clusters. Our scheme comprises two fundamental steps: formation of GRNs and evaluation of their similarity (Figure 1A). As the concept of cell classes are independent from the choice of data analysis methods, this framowork itself can be applied into various cases regardless of any feature engineering (such as the data preprocessing) and clustering methods.

The framework can be applied into the annotation of scRNA-seq data when a referential dataset is available (Figure 1B). As the annotation is the act of tagging clusters with descriptions in natural languages, the biological features of annotated clusters often treated as general and preserved properties of the cell types which the clusters are named after. Accordingly, it is better to have a large enough referential dataset which seem to reflect canonical states of specific cell types which are shared with the query dataset. Using GRN-based characterization, cell classes are annotated with the name of the most similar cell class, however, comparisons of cellular identities can be bidirectional due to asymmetry of  $d^*$ . We named the similarity of cell classes from the perspectives of the query data as estimation, and the one from the point of view of the referential data as labeling. Those GRN-based annotations can be visualized with planet plots, where the subjective cell class (here we denote it  $[x]$ ) is located in the center and the radii of the circles reflect  $d^*([x], \cdot)$  values for all cell classes placed on the circumferences.

The performance of those frameworks build around GRNs have its bottleneck in the step to create GRNs, and the process can be broken down into the configuration of the vertex sets and the choice of the network algorithm (Figure 1C). As well as the methods of the feature engineering and the clustering, each step of the GRN formation also has a variety of options. In the last paper, we introduced a combined method of manual curation referring review articles and a machine learning (ML)-based feature selection using a gradient boosting decision tree (GBDT) model with the L1 and the L2 regularizations[3]. For the manual supervision, gene ontology (GO) terms can be another information source. Nevertheless, a priori identification of the sample components are essential to create meaningful GRNs by injecting the domain-specific information. The differentially expressed gene (DEG)-based method can be a more heuristic and a less interactive option because the differential expression analysis (DEA) semi-automatically scoops DEGs. Regarding the network algorithms, we mentioned that there are several possible options as well. In our previous paper, we implemented our codes using the numerical (i.e., correlation-based) PC algorithms provided in Pgmpy[6], a python package for probabilistic graphical models. Another variation of the PC algorithm based on the chi-square test suitable for categorical data is also a realistic option if when the expression values can be binalized in some ways. We also mentioned that the max-min hill-climbing (MMHC) algorithm, which combines constraint-based and scoring-based methods[7], is one of promising alternatives of the PC algorithm.

So far, we have highlighted the versatility of our framework by providing examples that showcase its applicability across various data analysis methods. Our intention was to allow researchers to reflect their expertise in specific sample domains or preferences to the process of describing the samples with words of biology. This customization ensures that the metrics for cellular identities are crafted to align with the specific research scopes, providing both necessary and sufficient resolutions. However, this design choice has the drawback of making our algorithm less user-friendly, as it requires a significant amount of effort in annotation, even when it might not be the primary focus of their projects. To validate the legitimacy of our theory in as many cases as possible, it is essential to refine the practices related to GRN-based annotation, streamlining the overall workflow.

In this article, enumerating the three major topics where the former protocol has rooms for refinement, we provided more practical solutions for each while leveraging the backbone theory of GRN-based comparisons of cluster-wise cellular identities (i.e., cell classes).

## Results

### Challenges of the framework of GRN-based methods

Here in this section, we will point out the following three challenges regarding practical uses of the GRN-based annotation to solidify our goals for this article.

#### 1. Difficulty of effective gene selection

In our previous study, we proposed a method of choosing marker genes by combining supervised curation referring a review paper and a ML-based feature selection leveraging the feature importance in L1-regularized GBDT model. Although we have mentioned that there are various possible options for the marker gene selection, each strategy has its unique drawback.

Supervision by the experimenter struggles with completeness and arbitrariness even though we cite some reliable sources (e.g., review papers, or GO terms). For example, in the last paper, we selected SLC1A2, VIM, and AQP4 as glial markers referring a review article[8]. Those glial markers are subscribed to 92 GO terms in total, and there are other genes than the three tagged with those terms (Figure 2A). This example highlights the incompleteness of the three genes to represent all aspects of glia. Furthermore, the many-to-many correspondence of genes and GO terms would make it a significant challenge to draw a clear and reasonable line between the genes adopted as marker genes and the others.

ML-based approach is another method we implemented in our last report, and has its unique problems. As we shown in Figure S1, the standard workflow of the scRNA-seq data analysis consists several steps: the quality control (QC) of the data; normalization such as reads per million (RPM) transformation and logarithmic transformation; highly variable gene (HVG) extraction; demensionality reduction such as principal component analysis (PCA), truncated singular value decomposition (TSVD), Uniform Manifold Approximation and Projection (UMAP)[9], etc.; clustering; DEA; annotation; and other downstream analysis[10]. Given that creating a good ML model requires plenty of time other than actual run times for fine tuning the model configurations, those trials might take excessive efforts just for gene selection for GRNs even when the annotation is unlikely to be the ultimate goal of the data analysis. Additionally, even with a ML model that performs well, extracting informative features can suffer from arbitrariness about the selection. To demonstrate those difficulties, we analyzed an open source scRNA-seq data of peripheral blood mononuclear cells distributed from 10X Genomics (for short, we called the dataset with an alias, PMBC3k, in this research). Starting from QC, we proceeded to leiden clustering to have 9 clusters (0~8) as shown in Figure 2B. Then, we created a GBDT model for multiclass classification (which predicts clusters from the gene expression values). The model seemed to perform well in regards of the area under the curve (AUC) of the receiver operating characteristic (ROC) curve, the average precision (AP) of the precision-recall (PR) curve, and the accuracy score (Figure S2A-D). In the previous article, we made a three-class classification model and utilized the feature importance as a criterion (Figure 2C). However, the same approach did not work for the nine-class classification model this time because there is no clear boundary between key features and negligible ones even within the top 10 features of importance. Note that GRNs require pairwise calculation of genes, accordingly, modelers should avoid using an excessive number of genes for computational efficiency. Other than the feature importance scores, the shapley additive explanations (SHAP) scores can be an alternative metric to visualize the correspondence between the features and the classifications[11]. Even though SHAP scores provided more intuitive and precise explanations (Figure 2D and S3A-I), it is still a challenge to introduce an objective thresholds for gene selection beacuse the distributions of SHAP scores drastically vary across different classes. Consequently, the ML-based approach is not the most effective way to select marker genes to represent the cell classes because it also requires the modeler’s subjectivity as well as reference based supervision despite it is more time-consuming. ML-based approaches might work well if the character of the samples are completely unknown, or the concensus among the experts is yet to be settled. However, even under such conditions, alternative methods such as DEG-based approach should be accounted.

The DEG-based approach is another alternative that can be smoothly added onto the regualr scRNA-seq data analysis pipeline. In spite of its heuristicity and promptness, this method also has a shortcoming. Using the PMBC3k data processed in the exact same way as what we performed in the last section, we will exhibit an example hereby. The advantage of the GRN-based approach is the swiftness of the overall procedure by directly applying the top DEGs into the vertex sets. Accordingly, we applied the top 5 DEGs of each cluster (Figure S4A), created GRNs based on those genes (Figure S4B), and calculated the  $d^*$  values (Figure 2E). Looking at the bottom row of the heatmap, the  $d^*$  values were all zero from the point of view of cluster 8 even though it showed significantly different expression patterns of the top 5 marker genes (Figure 2F). Additionally, the top 30 upregulated GO terms of each cluster indicate that cluster 8 could exclusively be annotated as megakaryocytes, but the remainders exhibited different cellular characters (Figure S5A-I). Those results indicated that the GRNs did not work properly for identification of the cluster 8 because the zero  $d^*$  values for the clusters 0~7 implied that those clusters and the cluster 8 were indistinguishable in terms of the GRNs with the given vertex sets. Increasing the number of DEGs to 10, a edge added to the GRN for the cluster 8, which alteration made the  $d^*$  values non-zero (Figure 2H-I). As the GO terms suggested that there were no cluster of megakaryocytes other than the cluster 8, the new  $d^*$  values indicating that the clusters 0~7 were equally different from the cluster 8 seemed correct. Likewise we explored the optimal number of the DEGs to use for the vertex sets, the marker-gene selection is an intricate step that requires repetitive adjustments and validations.

Considering the fact that our method’s primary application is the annotation of scRNA-seq data, we need an alternative method to find marker genes to reduce computational costs and streamline the overall time required to initiate main analyses.

## 2. Statistical issue: independence v.s. uncorrelation

The statistical independence of two events  $A$  and  $B$  is defined as a situation where the following equation holds:

$$P(A \cap B) = P(A)P(B), \quad (2)$$

where  $P(\cdot)$  is the probability of an event. On the other hand, the correlation coefficient  $Corr(X, Y)$  of stochastic variables  $X$  and  $Y$  is defined as follows:

$$Corr(X, Y) := \frac{Cov(X, Y)}{\sqrt{Var[X]Var[Y]}} = \frac{E[(X - E[X])(Y - E[Y])]}{\sqrt{Var[X]Var[Y]}}, \quad \text{if } Var[X]Var[Y] > 0, \quad (3)$$

where  $E[\cdot]$  is the expected value,  $Var[\cdot]$  is the variance, and  $Cov(\cdot, \cdot)$  is the covariance. Independent variables exhibit a correlation coefficient of zero, the converse is false (e.g., when  $X \sim U(-1, 1)$ , where  $U(-1, 1)$  refers to the uniform distribution over the interval from -1 to 1,  $Corr(X, X^2) = 0$  although  $X$  and  $X^2$  are dependent). Therefore, strictly speaking, it is not appropriate to substitute the chi-square test or the exact test with the t test of correlation. Furthermore, the correlation-based method does not work well when the gene expression matrices regarding the selected genes are highly sparse.

However, we introduced a correlation-based algorithm to get GRNs compromising rigor in order to adjust to continuous gene expression values. To address this issue, we need to implement an effective method to binarize the gene expression values so that the new algorithm would rely on the statistical tests of independence. This update would make our algorithm align better to the original concept of our theory.

### 3. Irresponsibility to gene expression values

To demonstrate this difficulty, we analyzed an open source scRNA-seq data of peripheral blood mononuclear cells from 10X Genomics (for short, we named the dataset PMBC3k in this research).

### Automated marker-gene suggestion

Although we intended to require experimenters to curate marker genes to use in GRNs, overly recursive trials to find

### Dropout-based binarization

With a map  $q : \Gamma \times X \rightarrow \mathbb{N}$  which returns a raw gene counts of gene  $\forall g \in \Gamma$  for sample  $\forall x \in X$  where  $\Gamma$  is the whole set of genes and  $X$  is the whole set of samples, here we introduce the coverage function  $Coverage_{[x]} : \Gamma \rightarrow \mathbb{Q}$  of cell class  $[x]$  as follows:

$$Coverage_{[x]}(g) := \frac{|\{x \mid x \in [x] \text{ s.t. } q(g, x) \neq 0\}|}{|[x]|} \quad (4)$$

Note that  $Coverage_{[x]}$  relies on  $q$  only for identifying zeros in raw counts, therefore, any kind of values converted from raw counts by a transformation  $\psi : \mathbb{N} \rightarrow \mathbb{R}$  such that  $\psi^{-1}[\{0\}] = \{0\}$  can be used in lieu of  $q(g, x)$ . For instance, RPM values and  $\log_2(RPM + 1)$  are accepted (see Appendix for more detailed explanations).

### Weighted evaluation function

$$d^*([x], [y]) := 1 - \frac{|C_{[x]}(G) \cap C_{[y]}(G)|}{|C_{[x]}(G)|} = 1 - \frac{|E_{[x]}(G) \cap E_{[y]}(G)| + |G|}{|E_{[x]}(G)| + |G|} \quad (5)$$

$$Whqpm([x], [y]) := 1 - \frac{|E_{[x]}(G) \cap E_{[y]}(G)| + \sum_{g \in G} Coverage_{[x]}(g)}{|E_{[x]}(G)| + \sum_{g \in G} Coverage_{[x]}(g)}. \quad (6)$$

Weighted Hamming quasi-pseudo-metric

## Discussion

I have no idea.

# Methods

## GRNet Impletementations

### GO term-assisted gene selection referring Jaccard Index

$$J(A, B) := \frac{A \cap B}{A \cup B} \quad (7)$$

Jaccard Index of two sets  $A, B$  is defined as Eq. (7). We expanded this definition to pairwise comparisons of multiple elements by forming a matrix where each element is the corresponding Jaccard Index, and we named the matrix Jaccard index matrix (JIM). For example, the element in  $i$ -th row and  $j$ -th column (where  $i, j, k \in \mathbb{N}$  and  $i \leq k, j \leq k$ ),  $JIM_{i,j}$ , can be defined as follows when a JIM of sets  $X_1, X_2, \dots, X_k$  are considered:

$$JIM_{i,j} := J(X_i, X_j) \quad (8)$$

Especially for seed markers, sets of subscribed GO terms (let  $G_1, \dots, G_k$ ) and their JIM are calculated in order to set  $\min_{i,j}(J(G_i, G_j))$  as a threshold of biological correspondence.

⋮

For detailed method of implementation, we calculated the JIM of the related GO terms of given seed markers. We used mygene.py[12] to query the GO database, and Numpy[13] to calculate JIM.

### GRNs and the evaluation function

Following our previous report[3], we computed GRNs by calculating correlations of continuous gene expression values (e.g.,  $\log_2(RPM + 1)$ ) using pgmpy. In this study, we introduced

## scRNA-seq data analysis

### Dataset List

The scRNA-seq data we used in this research were publicly available as online resources as follows:

M1C10X: <https://portal.brain-map.org/atlas-and-data/rnaseq/human-m1-10x>

hFB: <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE165388>

PBMC3k: <https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/pbmc3k>

aHSPC: <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE137864>

BCA: <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE149938>

### Preprocessing, dimensionality reduction, and visualization

We performed data preprocessing, dimensionality reduction, data visualization of the scRNA-seq datasets using Python packages (including Scanpy[14], Polars, Pandas[15], Numpy, Matplotlib[16], Seaborn[17]) and Julia packages.

### Clustering and DE analysis

We performed leiden clustering, DE analysis using Scanpy.

### Multiclass classification GBDT model

We randomly split the PBMC3k data into training/validation/test data (3:1:1), and with the training and validation data, a GBDT model minimizing the multiclass-logarithmic loss function was created using LightGBM's framework. We implemented the model with a wrapper in Optuna to automatically tune the hyperparameters. The models performance were tested with the ROC curves and the PR curves using Scikit-learn and Matplotlib. We also visualized the feature importance values implemented in LightGBM. The SHAP scores were calculated and visualized with a Python package, Shap[11, 18].

### GO analysis

We performed the GO analysis using gprofiler2[19], and visualized the results with Matplotlib and Seaborn.

## Other visualizations

### Alluvial plot of gene symbols and GO terms

The glial markers were selected referring review articles, and the tagged GO terms were queried using mygene.py. Then, all gene symbols subscribed with each GO terms were queried again. The visualization was implemented with Matplotlib, Numpy, Pandas.

## Resource availability

### Data availability

Not applicable

### Code availability

GRNet and the analysis codes are available on GitHub (<https://github.com/yo-aka-gene/grnet>). Online documentation for GRNet is also provided (<https://grnet.readthedocs.io>).

## Author contributions

YO designed the project, implemented the algorithms, executed the analyses, and wrote the manuscript. YK contributed as the senior author, and edited the manuscript. HO edited the manuscript, and supervised the project.

## Acknowledgements

We thank hogehoge for thorough support.

## Abbreviations

**AP** average precision

**AUC** area under the curve

**DEA** differential expression analysis

**DEG** differentially expressed gene

**GBDT** gradient boosting decision tree

**GO** gene ontology

**GRN** gene regulatory network

**HVG** highly variable gene

**JIM** Jaccard index matrix

**ML** machine learning

**MMHC** max-min hill-climbing

**PC** Peter and Clark

**PCA** principal component analysis

**PR** precision-recall

**QC** quality control

**ROC** receiver operating characteristic

**RPM** reads per million

**scRNA-seq** single-cell RNA-sequencing

**SHAP** shapley additive explanations

**TSVD** truncated singular value decomposition

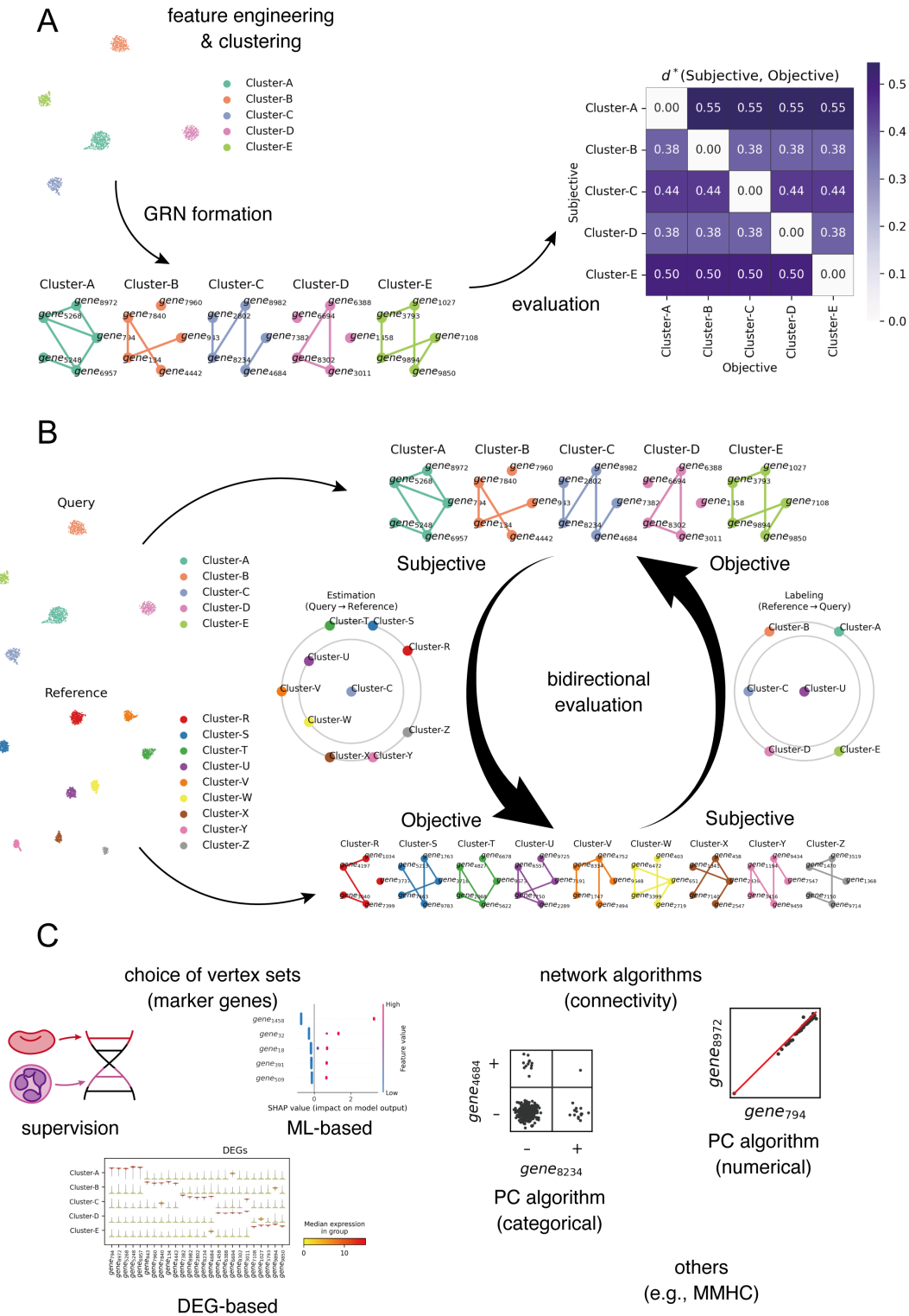
**UMAP** Uniform Manifold Approximation and Projection

## References

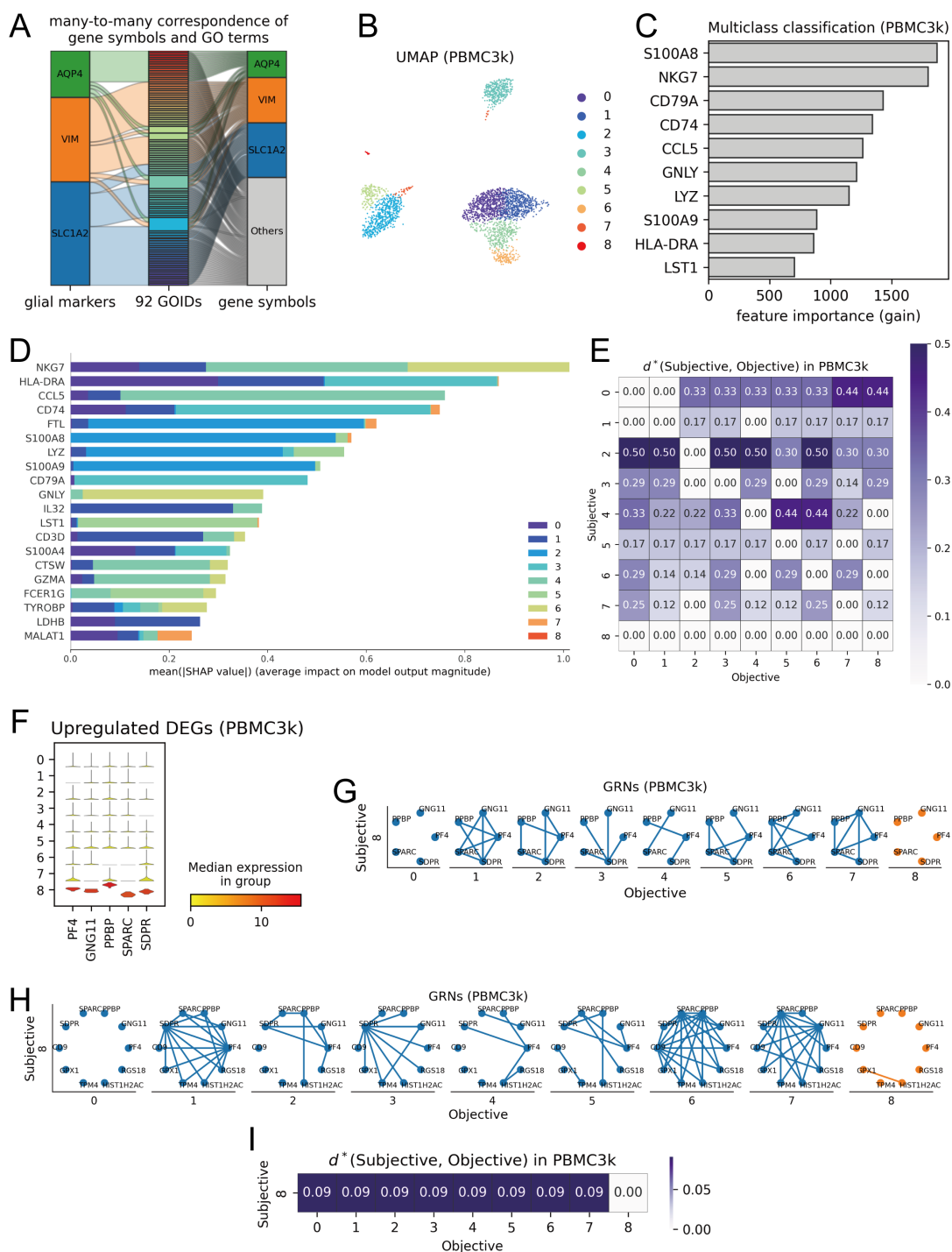
- [1] F. Tang, C. Barbacioru, Y. Wang, E. Nordman, C. Lee, N. Xu, X. Wang, J. Bodeau, B. B. Tuch, A. Siddiqui, *et al.*, “mrna-seq whole-transcriptome analysis of a single cell,” *Nature methods*, vol. 6, no. 5, pp. 377–382, 2009.
- [2] A. Regev, S. A. Teichmann, E. S. Lander, I. Amit, C. Benoist, E. Birney, B. Bodenmiller, P. Campbell, P. Carninci, M. Clatworthy, *et al.*, “The human cell atlas,” *elife*, vol. 6, p. e27041, 2017.
- [3] Y. Okano, Y. Kase, and H. Okano, “A set-theoretic definition of cell types with an algebraic structure on gene regulatory networks and application in annotation of rna-seq data,” *Stem cell reports*, vol. 18, no. 1, pp. 113–130, 2023.
- [4] A. Bookstein, V. A. Kulyukin, and T. Raita, “Generalized hamming distance,” *Information Retrieval*, vol. 5, pp. 353–375, 2002.
- [5] P. Spirtes, C. N. Glymour, and R. Scheines, *Causation, prediction, and search*. MIT press, 2000.
- [6] A. Ankan and A. Panda, “pgmpy: Probabilistic graphical models using python,” in *Proceedings of the 14th Python in Science Conference (SCIPY 2015)*, Citeseer, 2015.
- [7] I. Tsamardinos, L. E. Brown, and C. F. Aliferis, “The max-min hill-climbing bayesian network structure learning algorithm,” *Machine learning*, vol. 65, pp. 31–78, 2006.
- [8] J. Zhang, J. Jiao, *et al.*, “Molecular biomarkers for embryonic and adult neural stem cell and neurogenesis,” *BioMed research international*, vol. 2015, 2015.
- [9] L. McInnes, J. Healy, N. Saul, and L. Großberger, “Umap: Uniform manifold approximation and projection,” *Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.
- [10] M. D. Luecken and F. J. Theis, “Current best practices in single-cell rna-seq analysis: a tutorial,” *Molecular systems biology*, vol. 15, no. 6, p. e8746, 2019.
- [11] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 4765–4774, Curran Associates, Inc., 2017.
- [12] C. Wu, A. Mark, and A. I. Su, “Mygene. info: gene annotation query as a service,” *bioRxiv*, p. 009332, 2014.
- [13] C. R. Harris, K. J. Millman, S. J. Van Der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, *et al.*, “Array programming with numpy,” *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.
- [14] F. A. Wolf, P. Angerer, and F. J. Theis, “Scanpy: large-scale single-cell gene expression data analysis,” *Genome biology*, vol. 19, pp. 1–5, 2018.
- [15] W. McKinney *et al.*, “Data structures for statistical computing in python,” in *Proceedings of the 9th Python in Science Conference*, vol. 445, pp. 51–56, Austin, TX, 2010.
- [16] J. D. Hunter, “Matplotlib: A 2d graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

- [17] M. L. Waskom, “seaborn: statistical data visualization,” *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021.
- [18] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, “From local explanations to global understanding with explainable ai for trees,” *Nature Machine Intelligence*, vol. 2, no. 1, pp. 2522–5839, 2020.
- [19] L. Kolberg, U. Raudvere, I. Kuzmin, J. Vilo, and H. Peterson, “gprofiler2– an r package for gene list functional enrichment analysis and namespace conversion toolset g:profiler,” *F1000Research*, vol. 9 (ELIXIR), no. 709, 2020. R package version 0.2.3.





**A:** The foundation of the GRN-based characterization of cell classes. After clustering in designed data space by arbitrary methods, cell classes (the clusters) can be represented by GRNs of corresponding genes of choice. The similarity of two GRNs of the same vertex (marker gene) are evaluated with the asymmetrical function  $d^*$ , where the return values reflects the similarity from the viewpoint of the subjective clusters. **B:** Schematic of the GRN-based scRNA-seq data annotation. Expecting the referential data to reflect canonical states of target sample domains, the evaluation of the similarity among cell classes can be performed bidirectionally. **C:** Methodological variations of the selection of vertex sets (marker genes) and the algorithms to compute the network structures of GRNs.



**Figure 2:** Examples of the major issues on the GRN-based frameworks

**A:** Alluvial plot showing the many-to-many correspondence of gene symbols and GO terms. **B:** UMAP of the PBMC3k dataset. The markers were colored according to the cluster. **C:** The top 10 genes of the highest feature importance of the multiclass classification LightGBM model. **D:** **E:** **F:** **G:** **H:** **I:**