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

Yuji Okano[1,2], Yoshitaka Kase[2,3], and Hideyuki Okano[2,3]

[1] Department of Extended Intelligence for Medicine, The Ishii-Ishibashi Laboratory, Keio University School of Medicine; 35 Shinanomachi, Shinjuku-Ku, Tokyo 160-8582, Japan
[2] Division of CNS Regeneration and Drug Discovery, International Center for Brain Science, Fujita Health University; 1-98 Dengakugakubo, Kutsukake-Cho, Toyoake, Aichi 470-1192, Japan
[3] Keio University Regenerative Medicine Research Center; 3-25-10 Tonomachi, Kawasaki-Ku, Kawasaki, Kanagawa 210-0821, Japan

June 12, 2024

Abstract

The rise of single-cell RNA-sequencing (scRNA-seq) and computational algorithms have pushed forward today's biomedical science by visualizing multifaceted and diverse nature of cells in single-cell levels. In contrast, due to those technical advancements, cell clusters have played a pivotal role as instantiations of certain universal entities such as cell types and cell states, even though these clusters can be dataset-specific and method-dependent. In order to give them a structure comparable across different datasets or different compositions, in our previous paper, we introduced a graph-based representation of collections of cells that reflects statistical dependencies of their characteristic genes.

Despite we paid attention to theoretical insights in the previous work, refinement on practical implemtntation was left insufficient. Hence, in this article, we proposed a new practice to define and evaluate cellular identities with graph based on our theory. First we provided a concise summary for our theory and workflow we previously introduced. Then, we developped solutions that can point-by-point access issues after raising them to explain why they needed to be fixed. Leveraing alternative formats of cellular features such as gene ontology (GO) terms and dropouts, we upgraded our framework by improving utility. We also provided supplemental techniques to emphasize our standpoint or reinforce the versatility of our method.

Introduction

It has been more than 10 years since the birth of 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 Appendices 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]:

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

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, 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 probablistic 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 binarized 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 promissing 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 prectical 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 possibile options for the marker gene selection, each strategy has its unique drawback.

Supervison 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 subcribed to various 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, PBMC3k, 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 one-versus-rest (OvR) receiver operating characteristic (ROC) curve as well as the macro average of them, the average precision (AP) of the OvR precision-recall (PR) curve accompanied with the micro average of them, 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 because 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 PBMC3k 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 switfness 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.

So far we have point-by-point raised issues associated with respective marker-gene selection methods. Considering the fact that our method's primary application is the annotation of scRNA-seq data, which is unlikely to be the ultimate goals of the scRNA-seq data analyses, 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:

$$P(A \cap B) = P(A)P(B), \tag{2}$$

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:

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

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. As Eq. (3) holds if and only if both $Var[X]$ and $Var[Y]$ are non-zero values, under circumstances where all samples in a cluster exhibit zero counts for certain genes required in the vertex set, the correlation-based approach is inappropriate. This situation is by no means an imaginary counterexample unrealistically hypothesized just for criticism. A phenomenon called dropout is a characteristic of scRNA-seq data where gene expressions were not detected due to the inefficiency and the stochasticity of scRNA-seq[12], and it results in the high sparsity of scRNA-seq data matices.

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

The GRNs were designed to represent cellular functions by having edges between two statistically dependent genes. The correlation-based GRN generation is also rooted from the same idea and draws edges between two vertexes where they exhibit correlations. Even though those strategies can visualize the co-occurence or the mutual exclusivity of the gene expressions, actual expression values are dismissed. This failure leads to misassignments of cellular identities in practical cases as well as the example of PBMC3k; the GRNs of the cluster 0 and 8 showed exactly identical structures even though the expression patterns of the marker genes that forms the vertex set were significantly different (Figure 2F-G). This example highlights not only the difficulty of the gene selection but also the irresponsibility of GRNs to the gene expression values.

Semi-automated marker-gene suggestion

Although we intended to require experimenters to curate marker genes to use in GRNs, manual supervision struggles with arbitrariness and imperfection of the marker genes. Other semi-automated approaches, namely ML-based and

DEG-based methods, often require overly recurrsive trials to find optimal sets of marker genes. To improve the fluidity of the workflow, it is essential to introduce a novel method to collect marker genes efficiently.

To achieve that goal, we implemented an algorithm to automatically suggest similar genes to supplement given marker genes. We leveraged overlapped GO terms of the given marker genes and mapped them back to gene symbols. For instance, the three glial marker genes shares two GO terms in their intersection (Figure 3A), and the similarities of their GO terms can be set-theoretically defined with Jaccard index values (Figure 3B). Here we interpreted that: 1) the intersection of the Venn diagram contained the pivotal GO terms that reflected the biological semantics collectively defined by the given marker genes; and 2) the minimal Jaccard index value was the indicator of the similarity about the group of genes (therefore it could work as a threshold of acceptance when other genes are added). To find new genes without altering biological meaning of the list, we querried gene symbols tagged with the pivotal GO terms (Figure 3C), filtered out genes that exhibited lower Jaccard index values between any gene in the original list (Figure 3D), and the remainders formed the new gene list (Figure 3E).

Likewise we implemented the combined method of the manual and ML-based marker gene selection on the GRN-based annotation using a referential dataset (the labeling, which evaluates the d^* values from the referential clusters to the query clusters), such methods that require manual assignment of marker genes are suitable for characterizing clusters of known cellualr identities (i.e., pre-annotated clusters in referential datasets). Accordingly, our new proposal can be applied to similar cases.

Dropout-based binarization

As we discussed above, the dropout can be considered as an example for the pitfalls of the corrlation-based algorithm. On the otherhand, the dropouts are recently studied well to be turned out that the zero inflation are closely related to data attributes such as cell types[12, 13]. Inheriting from these ideas, we considered that the dropouts can be a good indicator of enriched gene expressions. In details, the binarization algorithm of our proposal determines non-zero expression values as positive and zeros as negative. For example, the top two DEGs of the cluster 8 in PBMC3k, PF4 and GNG11 (both are megakaryocyte markers[14]), were raraly expressed in the cluster 5 (Figure 2F), accordingly, the 2×2 contingency table based on the identification algorithm showed that the majority of the cluster 5 were classified double-negative (Figure 4A).

Although we adopt the standpoint of dropout as a practical feature, some experts have opposed the idea of exploiting dropouts and have developped dropout imputation algorithms[15]. To clarify our point, here we examine how dropouts explain the data features.

First, we validated if the dropout rate (DOR), the proportion of zeros in the count data of a gene, associated with the mean values ($log_2(RPM+1)$) in the PBMC3k data. Although the DOR values and the mean values exhibited a non-linear correspondence, we could successfully establish a linear formulation with a simple logistic transformation on the mean expression values (See Appendices for details). The logistic-transformed mean values fitted well to the linear calibration curve scoring 0.993 in the coefficient of determination (R^2), and we named the inverse-transformed curve aligned with the data distribution in the scatter plot of the mean values and the DOR (Figure 4B). Hence, we could demonstrate that the DOR values are closely related to the mean expression values, which are the most frequently-used summary statistics. As the DOR values are comparable across different datasets, while the mean expression values are unsuitable for trans-dataset comparison, it was suggested that the potential of the calibration curve of DOR and the mean expression to work effectively in cell class comparison using multiple data source by interchangeably translating the comparable feature and the uncomparable but meaningful one. To benchmark the performance of the model, we coined the name logistic model (LM), and compared with a Poisson regression model and a negative binomial (NB) regression model (Figure 4C), which are well-known models of dropout events[16]. The mean squared error (MSE) scores of those models indicated the LM best fitted to PBMC3k dataset compared to the other competitors (Figure 4D), and its mean absolute error (MAE) (i.e., expected prediction error) in DOR value turned out to be less than 0.005 (Figure 4E). To measure the errors produced when turning DOR values back to mean expressions, we made the inverse prediction models of those three models (See Appendices for details), and tested their performance. As described in Appendices, the all inverse prediction models have a fundamental issue in predicting mean expression values for zero DOR, we excluded those data from performance evaluation and we visualized maximum absolute error (MaxAE) values in addition to MAE values so that we could quantify the prediction performance for data of low DOR. LM exhibited lowest MAE scoring less than 0.1 errors in mean expression values on average (Figure 4F), and it scored the best in MaxAE as well (Figure 4G).

Followingly, we tested if there is correspondence between DOR and some data attributes unique to individual datasets using a group of datasets obtained by Mereu and the colleagues[17] (hereby we called the group of datasets Mereu2020). Mereu2020 includes 15 superfamilies where the same sample components were measured across different protocols (e.g., different platforms or different sequencing depth) in order to benchmark scRNA-seq protocols[17], including Chromium V2 (deep), Chromium V2 (shallow), Chromium V2 (sn), Chromium V3, C1HT-medium, C1HT-small, CEL-seq2, Drop-seq, ICELL8, MARS-Seq, Quartz-Seq2, gmcSCRB-seq, ddSEQ, inDrop, and Smart-Seq2 (for detailed descriptions, please refer to the original article[17] and URLs to the corresponding webpages on Gene Expression Omnibus we respectively provided in Methods). Datasets included in Mereu2020 exhibited wide range of variations in sample sizes and total reads (Figure S6A). When we visualized coverages of gene expressions (in other words, proportions of non-zero values which is equivalent to $1 - DOR$), numbers of unique molecular identifier (UMI), and total reads per sample, datasets with high coverages were enriched in UMI and read counts (Figure S6B-D). Therefore, it was suggested that DOR reflected those metadata attributes, which is also discussed in previous studies[12, 13]. Futhermore, we tested if we could reproduce LMs explaining the intertwinement between DOR and mean expressions well in Mereu2020 datasets (Figure S7A-O). As well as we have shown with PBMC3k dataset, logistic-transformed mean expression values of all datasets fitted well to the linear calibration curves with high coefficients of determination (Figure S8A). We also benchmarked their performance comparing with Poisson and NB regression models (Figure S7A-O, S8B-E). As there provided detailed descriptions in Appendices, LM showed its ability to work as a calibration curve of DOR and mean expression values in a wide range of datasets. Consequently, it was suggested that DOR reflects metadata features and per-gene characteristics.

Given those examples we have so far demonstrated, we concluded that DOR can be a useful statistic that reflects collective features of scRNA-seq data including mean expression values and other metadata including information about sequencing depth.

Weighted evaluation function

As we stated above, the GRN formation dismisses actual mean expression values of a cell cluster by encoding only the co-occurence (or co-absence) of gene expressions. To fix this issue, we introduced a new metric which can play a role as an evaluation function of GRNs in lieu of d^*, so that we can assign weights to the abundance of gene expressions on the similarity of graph structures. To quantify the amount of gene expressions in a manner comparable across different datasets, we applied the coverage (the presense of the non-zero gene expressions which is equivalent to $1 - DOR$) expecting DOR to indirectly reflect the mean expressions of the marker genes forming the edges of the GRNs. With a map $q : \Gamma \times X \to \mathbb{N}$ which returns a raw gene counts of gene ${}^\forall g \in \Gamma$ for sample ${}^\forall x \in X$ where Γ is the whole set of genes and X is the whole set of samples, we formulated the coverage function $Coverage_{[x]} : \Gamma \to \mathbb{Q}$ of cell class $[x]$ (which indicates the coverage value of the given gene g in the designated cell class $[x]$) as follows:

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

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 $\psi : \mathbb{N} \to \mathbb{R}$ such that $\psi^{-1}[\{0\}] = \{0\}$ can be used instead of $q(g,x)$. For instance, RPM values and $\log_2(RPM + 1)$ are accepted (see Appendices for more detailed explanations).

Given that Eq.(1) can also be denoted as Eq.(5), we introduced our new evaluation function, the weighted Hamming quasi-pseudo-metric (WHQPM) $Whqpm$, by multiplying the cardinality of the gene set $|G|$ respectively with the coverage values resulting in Eq.(6):

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

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

Note that WHQPM cannot be defined if $Coverage_{[x]}(g) = 0$ for all ${}^\forall g \in G$, and this property of WHQPM prohibits a cell class get its similarity to other cell classes characterized with totally irrelevant genes exhibiting zero expressions (See also Appendices for detailed explanations).

As WHQPM depends on coverage values, not only biological variations but technical factors including choices of sequencing pipelines as well affect the result. If one considers that differences in DOR are also realistic features of the data, WHQPM is available for comparing cell classes across different datasets. Otherwise, optimal transport (OT)-based domain adaptation can mitigate the gap if the experimenter prefer to standardize the various effects that have impact on DOR, as we described in Figure S9A-H and Appendices.

To demonstrate the benefit from the use of WHQPM, first we computed GRNs of the clusters 0 through 8 on their top 5 DEGs using the dropout-based binarization technique and the PC algorithm for categorical data (Figure 5A), and then calculated $Whqpm$ values to visualize the similarities of the clusters (Figure 5B). Although the PC algorithm for categorical data inferred the exact same GRNs for different clusters in some cases (e.g., the GRNs of the top 5 DEGs of the cluster 8, namely PPBP, SPARC, SDPR, PF4, and GNG11), $Whqpm$ distinguished the differnces between the cluster 8 and the other clusters as it returned non-zero values except for the cluster 8 itself. As d^* returns zero if the subjective cell class has no edges in its GRN, we could resolve this issue with WHQPM.

Discussion

In general, scRNA-seq data processing is driven by statistical, geometrical, and information-theoretical approach even though the results from those algorithms are perceived by testing if they can recite the storyline of biology. In other words, detailed aspects of algorithms are less significant if the results make sense in some way. Therefore, heurisiticity is valued rather than theoretical rigor in some context. As scRNA-seq data are acceleratedly accumulated even though they are sensitive to fluctuation of the surrounding conditions, we belive that a framework that can handle scRNA-seq data in a tentative but comparable format would help us land on universal truth yet to be unveiled by balancing context-dependency and generalization.

We designed the GRN-based definition of cellular identities and the metric d^* to quantify the similarity of them in order to meet the need, however, impracticality that we have so far pointed out had remained. Therefore, we proposed a series of solutions hoping for improved functionality. We also launched a python package GRNet (pronounced garnet) to provide a platform for our proposal concepts, which needs further validations in various cases.

Methods

GRNet Impletemtations

GO term-assisted gene selection referring Jaccard Index

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

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 $X_1, X_2, \cdots, X_k$ are introduced:

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

When a collection of genes $g_1, \cdots, g_k$ collectively explain certain type of cells, and they are tagged with respective sets GO terms $G_1, \cdots, G_k$, we considered $min_{i,j \in \{1 \cdots k\}} J(G_i, G_j)$ as a threshold of biological correspondence to the type of cells. For example, let G_{k+1}, G_{k+2} are the sets of GO terms tagged with g_{k+1} and g_{k+2} ($g_{k+1}, g_{k+2} \notin \{g_1, \cdots, g_k\}$), new gene g_{k+1} would be important for the type of cells if $min_{i \in \{1 \cdots k\}} J(G_i, G_{k+1})$ is less than $min_{i,j \in \{1 \cdots k\}} J(G_i, G_j)$, and g_{k+2} would be irresponsible if $min_{i \in \{1 \cdots k\}} J(G_i, G_{k+2})$ is greater than $min_{i,j \in \{1 \cdots k\}} J(G_i, G_j)$. Under those rules, we implemented to search for important markers from genes tagged with GO terms in $\bigcap_{i \in \{1, \cdots, k\}} G_i$.

For detailed method of implementation, we calculated the JIM of the related GO terms of given seed markers. We used mygene.py[18] to query the GO database, and Numpy[19] to calculate JIM.

GRNs and the evaluation function

Following our previous report[3], we implemented correlation-baed PC algorithm for GRN formation and the evaluation function d^* for similarity of GRN structures using Numpy, Pandas[20], and pgmpy. We also implemented dropout-based binarization, chi-squared test-based PC algorithm and WHQPM accordingly.

scRNA-seq data analysis

Dataset List

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

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

- Mereu2020: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE133549
  - Chromium V2 (deep): https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE133535
  - Chromium V2 (shallow): https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE133536
  - Chromium V2 (sn): https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE133546
  - Chromium V3: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE141469
  - C1HT-medium: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE133537
  - C1HT-small: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE133538
  - CEL-seq2: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE133539
  - Drop-seq: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE133540
  - ICELL8: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE133541
  - MARS-Seq: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE133542
  - Quartz-Seq2: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE133543
  - gmcSCRB-seq: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE133544
  - ddSEQ: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE133547
  - inDrop: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE133548
  - Smart-Seq2: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE133545

Preprocessing, dimensionality reduction, and visualization

We performed data preprocessing, dimensionality reduction, data visualization of the scRNA-seq datasets using Python packages (including Scanpy[21], Polars[22], Pandas, Numpy, Matplotlib[23], and Seaborn[24]).

Clustering and DEA

We performed leiden clustering, DEA using Scanpy.

Multiclass classification GBDT model

We randomly split the PBMC3k data into training/validation/test data (3:1:1), and with the traiuning and validation data, a GBDT model minimizing the multiclass-logarithtic 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, 25].

GO analysis

We performed the GO analysis using gprofiler2[26], and visualized the results with Matplotlib and Seaborn.

Statistical models of DOR and the benchmarking

For LM, we optimized b of the calibration curve by minimizing the MSE between DOR and $\frac{2}{1+e^{-b \cdot Mean}} + 2$ with AdaGrad. We implemented LM and plotting functions with AnnData, Matplotlib, Numpy, Pandas, and PyTorch[27]. We implemented Poisson regression models with Statsmodels[28]. For NB regression models, we built them on implementation of Statsmodels and optimized hyperparameters using Optuna[29].

OT-based coverage standardization

We made OT-based domain adaptation models using the EMDTransport class of POT[30] with the squared euclidean cost. We visualized the results with Matplotlib, Numpy, Pandas, and Seaborn.

Other visualizations

Alluvial plot and Venn diagram about 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 alluvial plot was created with Matplotlib, Numpy, and Pandas, and the Venn diagram was visualized with Matplotlib-Venn[31].

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 provived (https://grnet.readthedocs.io).

Author contributions

Conceptualization YO

Methodology YO

Impletemtation YO

Investigation YO

Visualization YO

Funding acquisition YO, YK, HO

Project administration YO, YK, HO

Supervision HO

Senior author YK

Original draft YO

Editing YK, HO

Acknowledgements

This work was supported by the Keio University Medical Science Fund (to YO).

Abbreviations

AP average precision

AUC area under the curve

DEA differential expression analysis

DEG differentially expressed gene

DOR dropout rate

GBDT gradient boosting decision tree

GO gene ontology

GRN gene regulatory network

HVG highly variable gene

JIM Jaccard index matrix

LM logistic model

MAE mean absolute error

MaxAE maximum absolute error

ML machine learning

MMHC max-min hill-climbing

MSE mean squared error

NB negative binomial

OT optimal transport

OvR one-versus-rest

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

UMI unique molecular identifier

WHQPM weighted Hamming quasi-pseudo-metric

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] P. Qiu, "Embracing the dropouts in single-cell rna-seq analysis," *Nature communications*, vol. 11, no. 1, p. 1169, 2020.

[13] L. Zappia, B. Phipson, and A. Oshlack, "Splatter: simulation of single-cell rna sequencing data," *Genome biology*, vol. 18, no. 1, p. 174, 2017.

[14] F. Puhm, A. Laroche, and E. Boilard, "Diversity of megakaryocytes," *Arteriosclerosis, Thrombosis, and Vascular Biology*, vol. 43, no. 11, pp. 2088–2098, 2023.

[15] T. H. Kim, X. Zhou, and M. Chen, "Demystifying "drop-outs" in single-cell umi data," *Genome biology*, vol. 21, no. 1, p. 196, 2020.

[16] K. Choi, Y. Chen, D. A. Skelly, and G. A. Churchill, "Bayesian model selection reveals biological origins of zero inflation in single-cell transcriptomics," *Genome biology*, vol. 21, pp. 1–16, 2020.

[17] E. Mereu, A. Lafzi, C. Moutinho, C. Ziegenhain, D. J. McCarthy, A. Álvarez-Varela, E. Batlle, n. Sagar, D. Gruen, J. K. Lau, *et al.*, "Benchmarking single-cell rna-sequencing protocols for cell atlas projects," *Nature biotechnology*, vol. 38, no. 6, pp. 747–755, 2020.

[18] C. Wu, A. Mark, and A. I. Su, "Mygene. info: gene annotation query as a service," *bioRxiv*, p. 009332, 2014.

[19] 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.

[20] 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.

[21] 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.

[22] R. Vink, "Polars: Blazingly fast dataframes in rust, python, node.js, r, and sql." https://github.com/pola-rs/polars, 2024. Version 0.20.10.

[23] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[24] M. L. Waskom, "seaborn: statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021.

[25] 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.

[26] 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.

[27] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” 2017.

[28] S. Seabold and J. Perktold, “Statsmodels: econometric and statistical modeling with python.,” *SciPy*, vol. 7, p. 1, 2010.

[29] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A next-generation hyperparameter optimization framework,” in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631, 2019.

[30] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, *et al.*, “Pot: Python optimal transport,” *Journal of Machine Learning Research*, vol. 22, no. 78, pp. 1–8, 2021.

[31] K. Tretyakov, “matplotlib-venn: Venn diagram plotting routines for python/matplotlib.” https://github.com/konstantint/matplotlib-venn, 2024. Version 0.11.10.

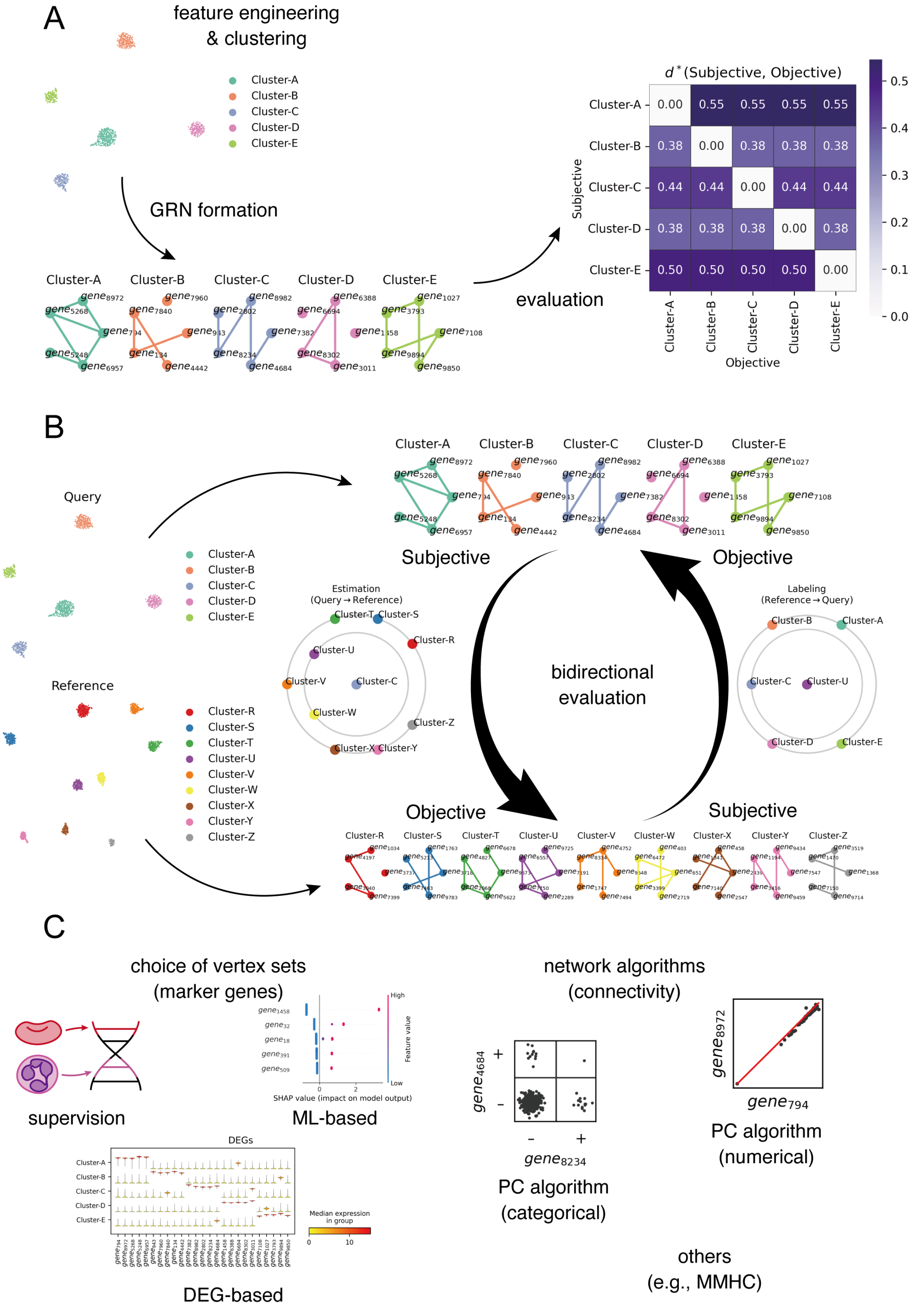

Figure 1: The framework of the GRN-based characterization and annotation of cell classes

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 assymetrical 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.

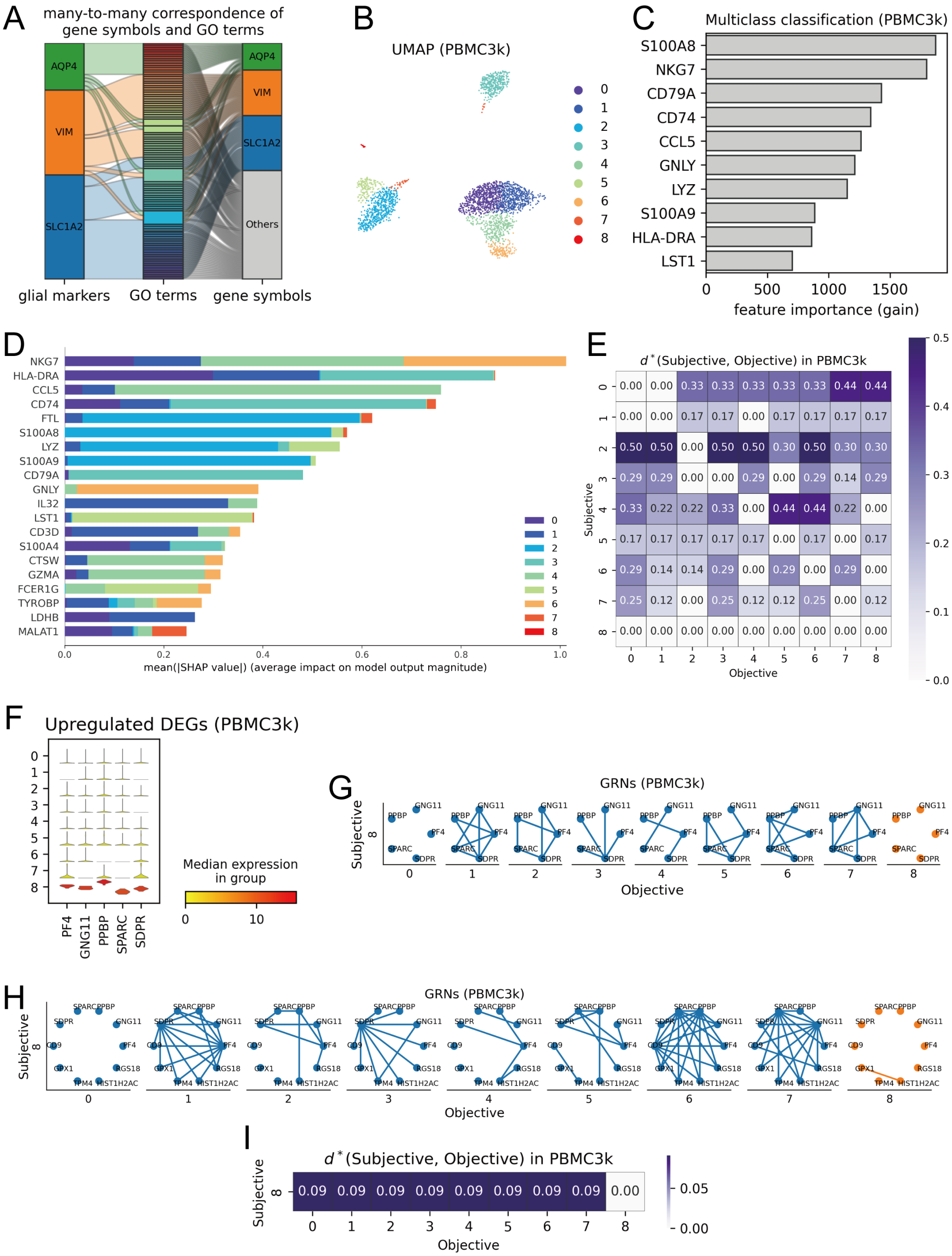

Figure 2: Examples of the major issues on the GRN-based frameworks

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 imporance of the multiclass classification LightGBM model. **D**: The top 20 genes of the highest mean SHAP values of the multiclass classification LightGBM model. **E**: The d^* values based on the GRNs of the top 5 DEGs. The rows correspond to the subjective cell classes, and the columns correspond to the objective ones. **F**: The top 5 DEGs of the cluster 8. **G**: The GRNs of the clusters 0∼8 based on the top 5 DEGs of the cluster 8. **H**: The GRNs of the clusters 0∼8 based on the top 10 DEGs of the cluster 8. **I**: The re-calculated d^* values among the GRNs based on the top 10 DEGs of the cluster 8.

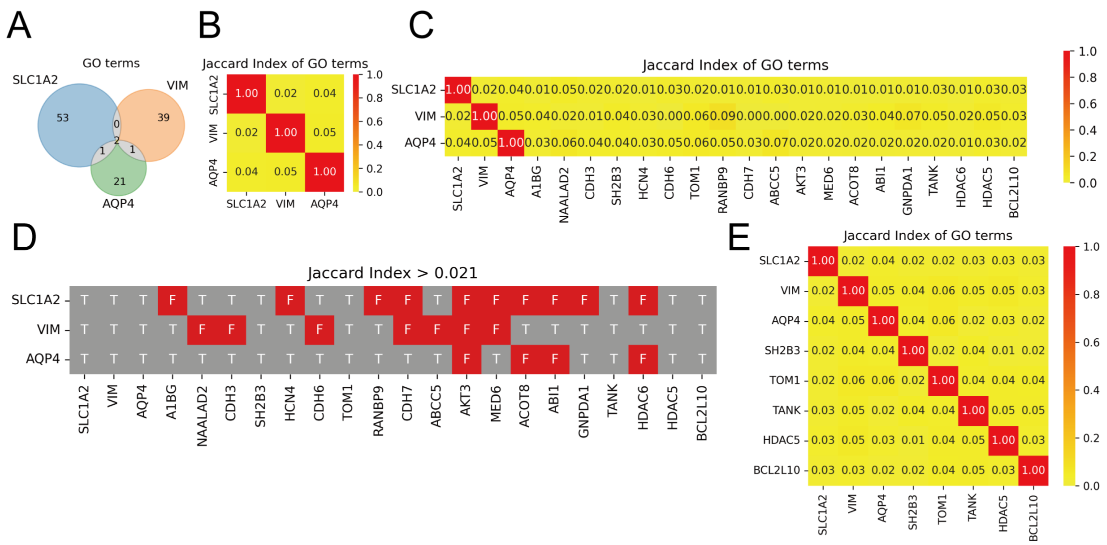

Figure 3: Jaccard index based automated marker gene suggestion

A: Venn diagram of the GO terms related to the three glial marker genes. Here we considered the intersection of the whole three set as a set of the pivotal GO terms defined by the three marker genes. **B**: Jaccard index values of the GO terms related to the three glial marker genes. The minimal value were adopted as the threshold for the automated gene selection. **C**: Jaccard index values of the GO terms related to the three glial marker genes and other gene symbols subscribed to the pivotal GO terms. **D**: Jaccard index values smaller than the threshold were shown in red and the others were shown in gray. **E**: Jaccard index values of the GO terms related to the gene symbols included in the output gene list.

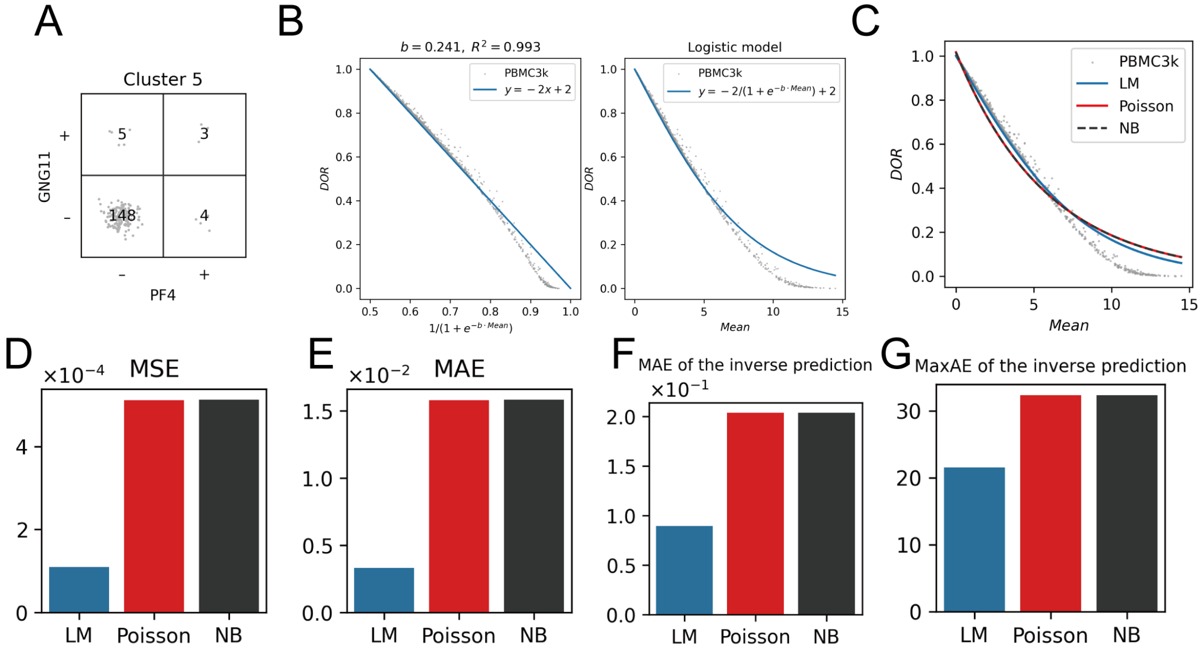

Figure 4: Dropout-based binarization and empirical investigations on DOR

A: A dropout-based 2×2 contingency table of PF4 and GNG11 for the cluster 5 in PBMC3k (+: non-zero expression values, $-$: zeros). **B**: The LM of DOR. **C**: The performance comparison with the Poisson regression model (Poisson) and the negative-binomial regression model (NB). **D**: Performance comparison of LM, Poisson, and NB with MSE values. **E**: Performance comparison of LM, Poisson, and NB with MAE values. **F**: Performance comparison of the inverse predictions of LM, Poisson, and NB with MAE values. **G**: Performance comparison of the inverse predictions of LM, Poisson, and NB with MaxAE values.

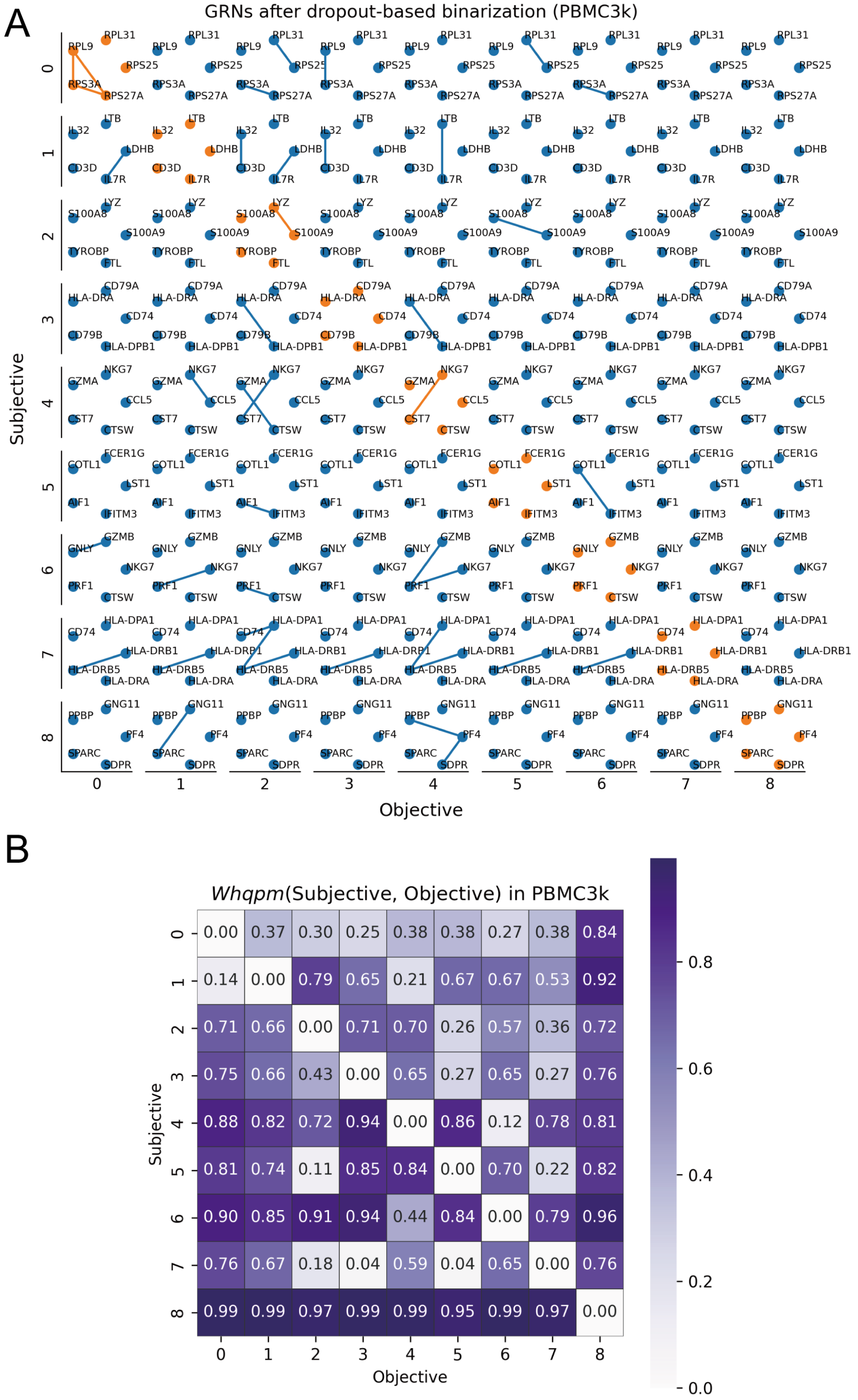

Figure 5: Combination of dropout-based bunarization and WHQPM

A: GRNs of the clusters in PBMC3k generated with dropout-based binarization and PC algorithm for categorical data. GRNs in a row share the same set of genes (DEGs of the subjective clusters) selected for the vertex sets. **B**: The $Whqpm$ values based on the GRNs of the top 5 DEGs generated after dropout-based binarization.