Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

D-BSSE
Department of Biosystems
Science and Engineering

Causal Inference of Gene Networks with a Mixture Nested Effects Model

Project report

Wuxing Dong

dongwu@student.ethz.ch

Computational Biology Group
Department of Biosystems Science and Engineering
ETH Zürich

Supervisors:
Dr. Martin Pirkl
Prof. Dr. Niko Beerenwinkel

December 17, 2020

1 Introduction

In diseases under intense study, such as cancer (Marusyk *et al.*, 2012; Fisher *et al.*, 2013) and neurodegeneration (Prinz *et al.*, 2011; Hashemiaghdama and Mroczek, 2020), there have been accumulating evidence showing that sub-populations of cells could respond differently to treatments. This has caused a paradigm shift among researchers, who now focus on heterogeneity among cell populations, in order to develop more personalized and effective treatments (Meacham and Morrison, 2013). In the meantime, gene signaling pathways are essential to understand cells in normal and diseased states. Interacting genes can be viewed topologically, where a node denotes a gene, and a directed edge connecting two genes denotes a casual signaling relationship between them. A disease network may arise due to malfunction of certain genes. Treatments can attempt to tackle the vital nodes in the disease networks (Barabási *et al.*, 2010).

Markowetz *et al.* (2005, 2007) proposed Nested Effects Models (NEM) to decipher gene signaling pathways from perturbation experiments. In such experiments, a signaling gene (S-gene) is knocked down and the effect on a range of other genes (E-genes) can be obtained from high-throughput gene expression data. Comparing to the control group, where no S-gene is perturbed, if the expression level of an E-gene j changes under the perturbation of S-gene i, then S-gene i has an effect on E-gene j. If the set of E-genes that S-gene i has an effect on is a subset of the E-genes that S-gene k has an effect on, then we can conclude that gene i is downstream of gene k in the gene signaling network, and draw a causal edge from gene k to gene i. NEM solves the adjacency matrix $\Phi \in M_{n\times n}(\{0,1\})$, which represents a gene network of n S-genes as a directed acyclic graph, and the adjacency matrix $\Theta \in M_{n\times m}(\{0,1\})$, which represents how m E-genes from the gene expression data is attached to the n S-genes. For instance, $\theta_{ij} = 1$ means that E-gene j is attached to the S-gene i, i.e., S-gene i has an effect on E-gene j. We assume that each E-gene attaches to at most one S-gene. As a result, there is at most one non-zero entry in each column of Θ. Given Φ and Θ, we use $F = (f_{ij}) = \Phi\Theta \in M_{n\times m}(\mathbb{R})$ to express how E-gene j is expected to respond to the knock-down of S-gene i.

However, due to the heterogeneity of the cell population under inspection, distinct causal gene networks may co-exist in the sub-populations (Gaudet and Miller-Jensen, 2016). Since NEM cannot consider multiple sub-populations, Pirkl and Beerenwinkel (2018) developed the mixture Nested Effects Model (M&NEM). M&NEM considers K signaling pathways, or components, in a cell population and solves $(\Phi_k, \Theta_k)_{k=1,...K}$ with mixture weights $\pi = (\pi_1, ..., \pi_K)$, where $\sum_{k=1}^{K} \pi_k = 1$. The assignment of cells to components follows a soft clustering manner. Specifically, for each cell i and each component k, we have a responsibility γ_{ik}, which is the probability of cell i belonging to component k with $\sum_{k=1}^{K} \gamma_{ik} = 1$. M&NEM infers $(f_{k,ij}) = (\Phi_k, \Theta_k)_{k=1,...K}$ by maximizing the log likelihood of the data with an Expectation maximization (EM) algorithm.

Previously, M&NEM has been applied to perturbation data with small sets of genes (Pirkl and Beerenwinkel, 2018). However, it is unknown how M&NEM performs on larger gene networks, and how different hyper-parameters of the model influence the result. In this project, we chose a dataset from a Perturb-Seq study (Dixit *et al.*, 2016) that involves 24 transcription factors (S-genes). They knocked down the genes in bone marrow dendritic cells with small guide RNAs (sgRNAs) introduced by CRISPR lentiviral vectors, and exposed 32777 cells to lipopolysaccharide (LPS) for three hours. The effects of S-gene perturbations on the expression levels of 17775 genes (E-genes) were measured by single cell RNA sequencing (scRNA-seq).

The aims of the project are to 1) establish a data processing pipeline to convert raw scRNA-seq data into log odds for downstream M&NEM analysis, and 2) to explore how different modelling strategies affect the performance.

2 Methods

2.1 Data pre-processing

Data from a perturbation experiment by Dixit *et al.* (2016) were downloaded from the BROAD single-cell portal (https://singlecell.broadinstitute.org/single_cell/study/SCP24/perturb-seq#study-download). Each cell has a unique cell barcode (CBC). Each sgRNA for the knock-down of a transcription factor has a guide barcode (GBC), although several different sgRNAs may have been used for knocking down the same transcription factor. The file that contains the association between CBC and GBC were used to retrieve which S-gene was perturbed in each cell. Since this project focuses on single perturbation effects, cells with zero or more than one sgRNA detected were excluded. To reduce noise, only E-genes with median expression value ≥ 1 were included.

The data was then processed by Linnorm (Yip *et al.*, 2017), which is tailored for normalizing and transforming scRNA-seq data for precise downstream statistical analysis. It normalizes data by conversion into relative scale, removes technical noises such as low count and highly variable genes that are typical to scRNA-seq, and transforms the data with a logarithm.

Lastly, as shown in Pirkl and Beerenwinkel (2018), we computed the log odds from the normalized transformed data D. Let data matrix be $D = (d_{ij}) \in M_{m\times l}(\mathbb{R})$, where there are m E-genes and l cells. Each cell has either one or no (negative control) S-gene perturbed. The log odd matrix $R = (r_{ij}) \in M_{m\times l}(\mathbb{R})$ represents the log ratio between the likelihood of observing an effect and observing no effect. For each E gene i, functions F_{i0} (for null model with no S-gene perturbed) and F_{ik} (for the effect of S-gene k on i) were empirically estimated as either a density estimation or a cumulative distribution estimation using Gaussian kernel smoothing. Note the set of all cells with knock-down of S-gene k by S_k and the set of all control cells as S_0. For kernel density estimation (kde) we have:

$$\begin{aligned} r_{ij} &= \log \frac{P(d_{ij}|F_{ik})}{P(d_{ij}|F_{i0})} \\ &= \log \frac{|S_k|^{-1} \sum_{j'\in S_k} K_h(d_{ij} - d_{ij'})}{|S_0|^{-1} \sum_{j'\in S_0} K_h(d_{ij} - d_{ij'})} \end{aligned} \tag{1}$$

where K_h is the normal (Gaussian) kernel with bandwidth h that is the standard deviation of the normal kernel (Chacón and Duong, 2018). For kernel cumulative distribution estimation (kcde) we have:

$$\begin{aligned} r_{ij} &= \log \frac{P(d_{ij}|F_{ik})}{P(d_{ij}|F_{i0})} \\ &= \log \frac{min(|S_k|^{-1} \sum_{j'\in S_k} \kappa_h(d_{ij} - d_{ij'}), 1 - |S_k|^{-1} \sum_{j'\in S_k} \kappa_h(d_{ij} - d_{ij'}))}{min(|S_0|^{-1} \sum_{j'\in S_0} \kappa_h(d_{ij} - d_{ij'}), 1 - |S_0|^{-1} \sum_{j'\in S_0} \kappa_h(d_{ij} - d_{ij'}))} \end{aligned} \tag{2}$$

where κ is the integrated Gaussian kernel (Duong, 2016).

2.2 M&NEM model

To infer the gene network(s) of the transcription factors from the log odd data R, we used the M&NEM model implemented in the R package "mnem".

We used "learnk" function to infer K, the number of components, i.e., the number of distinguishable gene networks. This function finds K by clustering the cells and has three hyper-parameters that are listed as follows. 1) Clustering algorithms: K-means or hierarchical clustering; 2) model selection criterion: silhouette value, Akaike information criterion (AIC) or Bayesian information criterion(BIC);

3) distance metrics: correlation ($dist(a,b) = \frac{1-cor(a,b)}{2}$ for cells a, b), Euclidean distance or Manhattan distance.

We used "mnem" function to train the M&NEM model. There are two strategies to calculate the log likelihood ratio, and they can be chosen by setting the hyper-parameter "complete" in the function. One way is to compute the observed log likelihood ratio as

$$\begin{aligned}\mathcal{L}_{observed} &= \sum_{i=1}^{l} \log \sum_{i=1}^{K} \pi_k \exp \sum_{j=1}^{m} \log \frac{P(d_{ij}|f_{k,ij})}{P(d_{ij}|N)} \\ &= \sum_{i=1}^{l} \log \sum_{k=1}^{K} \pi_k \exp \sum_{j=1}^{m} f_{ij,k} r_{ij}\end{aligned} \tag{3}$$

where l is the number of cells, K is the number of components, m is the number of E-genes and N is the null model. Note that it uses *exp* on summation, which can be a large number due to the data having many E-genes or large effects Pirkl and Beerenwinkel (2019). This may result in infinity in a practical setting. Alternatively, our second strategy is to compute the expectation of the complete log likelihood ratio as

$$\begin{aligned}\mathcal{L}_{expected} &= \sum_{i=1}^{l} \sum_{k=1}^{K} \gamma_{ik} (\log \pi_k + \sum_{j=1}^{m} \log \frac{P(d_{ij}|f_{k,ij})}{P(d_{ij}|N)}) \\ &= \sum_{i=1}^{l} \sum_{k=1}^{K} \gamma_{ik} (\log \pi_k + \sum_{j=1}^{m} f_{ij,k} r_{ij})\end{aligned} \tag{4}$$

We used five different strategies for initialization, which were defined as the hyper-parameter "type" in the "mnem" function. They are described as follows.

1) "random": initialize responsibility γ for each cell randomly; 2) "networks": initialize S-gene network Φ randomly; 3) "cluster": initialize γ for each cell by clustering each S-gene into K components and different clusters within each of the S-genes were randomly combined for each individual start; 4) "cluster2": cluster the whole data into K components, and within each cluster, Φ is inferred deterministically and initialized for each start; 5) "cluster3": cluster the data as in "cluster2", and within each cluster, γ is inferred deterministically and initialized for each start.

3 Results

3.1 Data pre-processing

The raw data was normalized and transformed with R package Linnorm. 47.2% cells were filtered out since they had no, or more than one sgRNA found (Fig 1.A). Cells containing one sgRNA showed uneven distribution over the particular S-genes perturbed (Fig 1.C). Uniform Manifold Approximation and Projection (UMAP) was used as a dimension reduction tool for data visualization, which showed poor separation and little clustering with respect to S-gene (Fig 1.B).

The normalized and transformed count data D was then converted into log odds matrix R by using either kernel cumulative density estimation (kcde) or kernel density estimation (kde). For each cell, a positive correlation between the log odds generated by these two methods was found (Fig 1.D, left), however, kcde values (sd = 0.391) was more spread out than kde values (sd = 0.274)(Fig 1.D, right). Therefore, the kcde result was used as the log odds data matrix R for the following M&NEM modeling, in order to better reflect the diversity of cells.

3.2 Hyper-parameter tuning of M&NEM

We used the R package "mnem" to train the M&NEM models. We first aimed to find the optimal number of components, K. Using the "mnemk" function, we directly set $K = \{1, 2, 3, 4, 5\}$ and trained

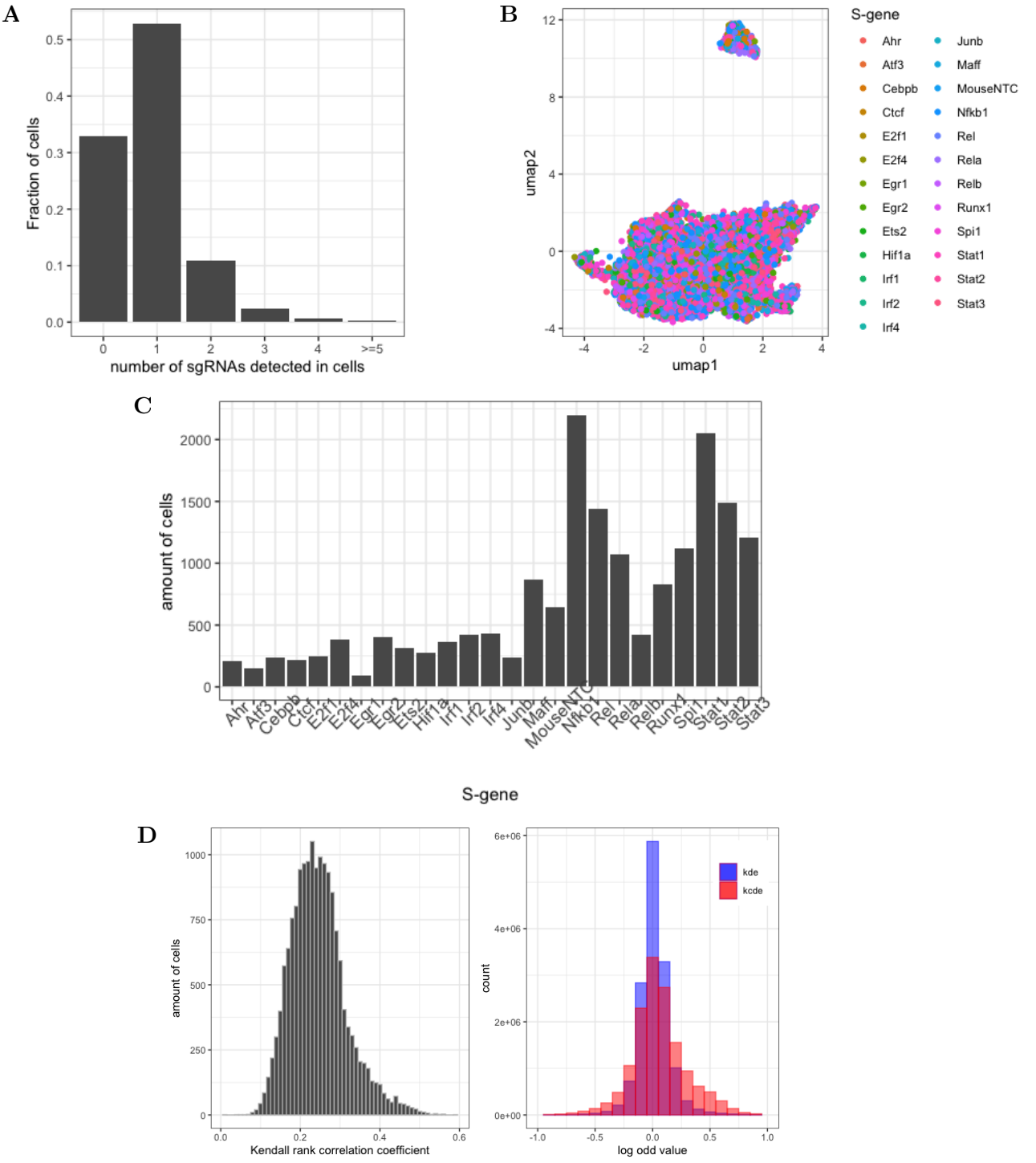

Figure 1: Pre-processing Perturb-Seq data. **A)** Distribution of number of sgRNAs detected in cells. Each cell has a unique CBC. The number of sgRNAs in a cell is the number of GBCs that are linked to the cell's CBC. **B)** Dimension reduction of the normalized transformed count data with UMAP. UMAP was configured to use Manhattan distance. Cells were colored in regarding to their perturbed S-genes. **C)** Distribution of S-gene perturbations among cells with single sgRNA. The particular S-gene knocked down in each cell was identified with a unique sgRNA. MouseNTC is the non-targeting negative control. **D)** Left, histogram on Kendall rank correlation between log odds values generated from kernel cumulative density estimation (kcde) or kernel density estimation (kde) (mean=0.25 across all cells). Each Kendall rank correlation value describes for each cell, the correlation between log odds calculated by using kde and kcde. Right, histogram of all log odds between kcde (red) and kde (blue) results. Each value is a log odd calculated by either kde (blue) or kcde (red), regardless of which cell the value belongs to.

five M&NEM models with each value of K. The raw log likelihood ratio increased with K (Fig 2.A, blue line). However, an optimal "elbow point" occurred at $K = 2$, where the minimum of the penalized log likelihood ratio (penalizing complex and redundant models as in Pirkl and Beerenwinkel (2018)) can be seen (Fig 2.A, red line). All higher Ks showed worse result than the single network (Fig 2.A,

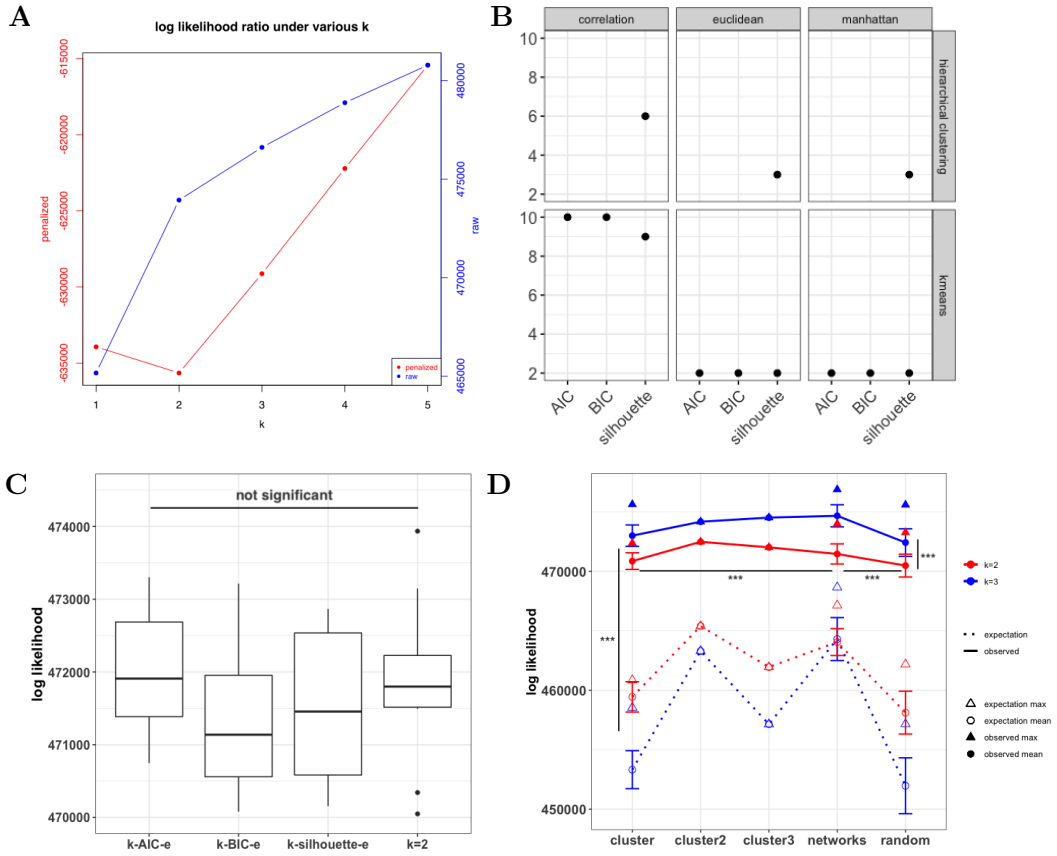

Figure 2: Hyper-parameter tuning of M&NEM model. **A)** Penalized (red) and raw (blue) log likelihoods under $k = 1, 2, 3, 4, 5$. Results are produced by "mnemk" function. Initialization method was "networks" (default). Observed log likelihood was calculated (default). **B)** Effect of clustering methods on number of components K, using the "learnk" function. Result consists of all combinations of two clustering algorithms (upper and lower boxes), three model selection criteria (bottom x-axis) and three distance metrics (left, middle and right boxes). Y-axis represents the inferred K values. Missing values were due to unavailability of the program under such parameter settings. **C)** Box plot of log likelihoods from M&NEM models with the same K generated from different methods (clustering with "learnk" or direct setting). The methods were represented on x-axis, where "k" stands for K-means algorithm and "e" stands for Euclidean distance. "k=2" stands for directly setting $k = 2$. Each group consists of ten replications. No significant difference was found in any of the group means (One-way ANOVA, $p = 0.487$). **D)** Mean and max log likelihoods of M&NEM models trained using the "mnem" function. $K = 2$ (red) was generated using K-means algorithm with silhouette score and Euclidean distance; $K = 3$ (blue) was generated using hierarchical clustering with silhouette score and Euclidean distance. Initialization methods were as labeled on x-axis. Solid lines represent observed log likelihoods; dashed lines represent the expectation of the complete log likelihoods; hollow triangles and circles represent expectation of the complete log likelihoods' maximum and mean respectively; filled triangles and circles represent observed log likelihoods' maximum and mean respectively. Error bars are standard deviations. When using "cluster2" or "cluster3" for initialization, one replication was performed since they produce the same result, otherwise 100 replications were performed. Mean of all observed log likelihood ratios (solid lines) and all expectations of the complete log likelihood ratios (dashed lines) were compared. Mean of observed log likelihood ratios under $K = 2$ (solid, red line) and $K = 3$ (solid, blue line) were compared. Under $K = 2$ and "observed" (solid, red line), mean of "networks" was compared to mean of "cluster" and "random" separately. All statistical tests were Wilcoxon test,*** $p < 0.001$

red line). This suggests that $K = 2$ is the value that balances both performance (high log likelihood) and model simplicity (low K). However, training a separate M&NEM model for each K value is time-consuming and ineffective. We then used a systematic method to infer K, with the function "learnk". This method learns the K that best clusters the cells. The clustering algorithms were K-means and hierarchical clustering; the model selection criteria were silhouette score, AIC score or BIC score; the

distance metrics were correlation, Euclidean and Manhattan. We performed a grid search to examine all combinations of these three hyper-parameters. Regardless of model selection criteria, K-means clustering consistently concluded $K = 2$ and hierarchical clustering concluded $K = 3$, when using Manhattan or Euclidean distance (Fig 2.B). When cell distances were represented by correlation, the number of K was larger than three (Fig 2.B). However, all but two or three components had very small mixture weights, indicating an over fitting situation. Therefore, for this dataset, we can find the optimal K by using K-means clustering of cells in either Euclidean or Manhattan distance and all three model selection criteria.

To train an M&NEM model, one can directly set a range of Ks and choose the model with the optimal K using the "mnemk" function as in Fig 2.A. Otherwise, one can use the "mnem" function, which firstly calls "learnk" function to infer K, which is then used to train the M&NEM model. When K and other hyper-parameters are the same, the two methods should not affect the log likelihood result. To examine this, we compared four kinds of M&NEM models under the optimal $K = 2$ with ten replications in each group. For the first three groups, we trained M&NEM models after using K-means clustering in Euclidean distance with 1) silhouette 2) AIC and 3) BIC score to infer K. We know all three methods produce $K = 2$ from Fig 2.B. For the fourth group, we picked the "mnemk" result under $K = 2$, which contains ten replications of M&NEM models whose K was directly set to be two. Data followed a normal distribution (Shapiro test, $p = 0.371$) and had variance homogeneity (Levene test, $P = 0.746$). Indeed, no significant difference was found among the different groups (Fig 2.C, One-way ANOVA, $p = 0.487$).

Next, we tested how initialization methods, strategies for log likelihood calculation (observed log likelihood ratio vs expectation of the complete log likelihood ratio) and K affect the performance of the M&NEM model when they are combined. 100 replications of M&NEM were trained for each combination of the three factors, using the "mnem" function. When initialization was "cluster2" or "cluster3", models were trained for only once, since these two methods produce deterministic results over replications. We clustered the data with silhouette score and Euclidean distance, and used K-means and hierarchical clustering to get $K = 2$ and $K = 3$, respectively, as shown in Fig 2.B. Converging on observed log likelihood ratios (Fig 2.D, solid lines and filled circles) reached significantly higher mean values than converging on expectations of the complete log likelihood ratios (Fig 2.D, dashed lines and hollow circles) (Wilcoxon test, $p < 0.001$). Therefore, we chose observed log likelihood ratio as the favorable hyper-parameter. When using observed log likelihood ratios, models with $K = 3$ (Fig 2.D, solid blue line and filled circles) achieved higher mean values than those with $K = 2$ (Fig 2.D, solid red line and filled circles) (Wilcoxon test, $p < 0.001$). However, we chose $K = 2$ for a better trade-off between performance and model simplicity in order to avoid over fitting (Fig 2.A). It is interesting to note that a higher K does not necessary increase the log likelihoods, as shown in Fig 2.C, dashed lines. Lastly, we chose the initialization method under $K = 2$ and "observed" (Fig 2.D, solid red line). Given these two conditions, "networks" showed a higher mean log likelihood value over the other initialization methods with 100 replications, namely "random" (Wilcoxon test, $p < 0.001$) and "cluster" (Wilcoxon test, $p < 0.001$)(Fig 2.D, solid red line and filled circles). Note that for "cluster2" and "cluster3", the means (Fig 2.D, circles) and maximums (Fig 2.D, triangles) were identical since they only had one replication each. Although "cluster2" had the highest mean log likelihood (Fig 2.D, filled circle in red), "networks" had the best maximum log likelihood (Fig 2.D, filled triangle in red). To conclude, the best model was achieved by calculating observed log likelihood ratios and initializing with "networks" under $K = 2$.

3.3 Inference of gene networks and visualization of sub-population clustering

We analyzed the M&NEM model with the optimal hyper-parameters concluded in 3.2, and the maximal log likelihood among the replications. All models under such hyper-parameters reached convergence within 30 EM iterations (appendix A). The responsibilities assigned to cells were mostly very high or very low (Fig 3.A). 40.1% of cells had over 95% max confidence ($max(\gamma_{ik}, k \in \{1, 2\})$ for cell i), which describes the confidence to be assigned to the most likely component (Fig 3.B). No cell had low (less than 50%) max confidence (Fig 3.B). Together with Fig 3.A, this indicated that for most of cells, a

very high responsibility was assigned to one component while a very low responsibility was assigned to the other component. In 2D space, cells with similar component assignment profiles tended to cluster together, in both UMAP (Fig 3.D, left) and t-SNE (Fig 3.D, right). Almost equal amount of cells were assigned to each of the two components, which also had equal mixture weight (Fig 3.C). With respect to the network structures, one was more intertwined (Fig 3.C, red) and the other was more linearly shaped towards the downstream (Fig 3.C, green). The edges of the S-genes also showed considerable variations. The two networks shared the same source node but they differed in their end nodes.

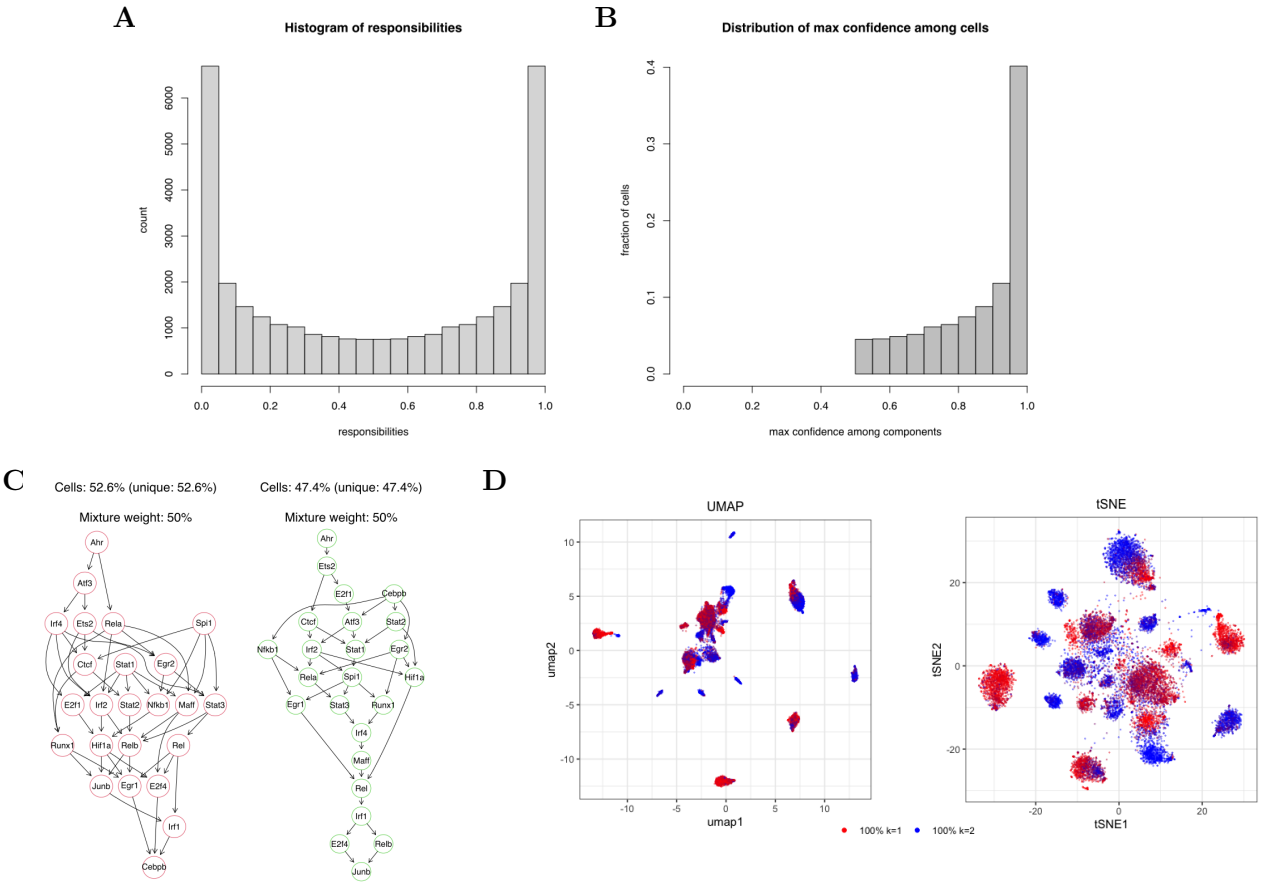

Figure 3: M&NEM model with optimal hyper-parameters and maximal log likelihood. **A)** Histogram of responsibilities γ. Histogram contains all γ_{ik} for components k and cell i. **B)** Distribution of maximal confidence for each cell. The maximal confidence for cell i is $max(\gamma_{i1}, \gamma_{i2})$ for the $K = 2$ model. **C)** The two distinct gene networks retrieved from the cell population. There are 52.6% and 47.4% cells assigned to each of the two sub-populations by hard clustering (i.e., when assigning each cell to either one of the components). The mixture weight π was 50% and 50% for the two components respectively. We define $\pi_k = \frac{\sum_{i=1}^{l} \gamma_{ik}}{\sum_{j=1}^{K} \sum_{i=1}^{l} \gamma_{ij}}$, where K is number of components and l is number of cells. **D)** Visualizations of log odds gene expression profile with UMAP (left) and t-SNE (right). Both dimension reduction methods were configured to use Manhattan distance metric. The "rgb" function was used to color each cell with two gradients according to their two responsibilities for each component. For example, for cell i, we have color gradients $red_i = \gamma_{i1}$, $blue_i = \gamma_{i2}$, with $red_i + blue_i = 1$.

4 Discussion

Current project analyzed scRNA-seq perturbation data from the Perturb-Seq study by inferring gene networks in sub-populations with M&NEM model. A data pre-processing pipeline was built, to filter raw data, normalize and transform it, and convert it into log odds by using kernel cumulative density estimation. The association between cells (CBC) and the S-gene perturbed (GBC) was based on a strict filtering schema (Dixit *et al.*, 2016), which may explain why as high as one third of the data being filtered out due to having no sgRNA detected (Fig 1.A). However, this gives higher confidence in the presence of sgRNAs in cells. A range of hyper-parameters were tuned for the current dataset. The balance between model performance and simplicity was achieved at $K = 2$ (Fig 2.A, red line). To infer

the optimal K systematically, one can also use certain combinations of hyper-parameters to cluster the cells (Fig 2.B). In addition, the choice of methods for initialization and log likelihood calculation affected the performance of the model (Fig 2.D). Therefore, hyper-parameter tuning is necessary for this dataset.

The trained M&NEM model showed good convergence, since all cells could be assigned to a certain subpopulation with high ($\geq 50\%$) confidence (Fig 3.B). Dimension reduction results (Fig 3.D) supported that M&NEM is a powerful technique to cluster the heterogeneous scRNA-seq dataset into distinct components. When cells were clustered in a 2D local structure, they were more likely to share the same gene network.

Both transcription factor networks had highly intertwined areas (i.e., multiple nodes on each level in the pathway) (Fig 3.C). This gives the cell population resilience to the breakdown of nodes due to mutations or malfunctions. Dixit *et al.* (2016) proposed a single network with a linear model. A comparison between Fig 3.C and their model featured some interesting differences. For instance, $Hif1\alpha$ was affected by many nodes in Dixit *et al.* (2016). It was directly affected by only three nodes in both of our networks. However, some of the interactions suggested by Dixit *et al.* (2016) were present in only one of our equally weighted networks, such as $Stat2$ and $Nfkb1$ (Fig 3.C). This indicates that a single network may lose information on heterogeneity, therefore, considering the pathways in a heterogeneous manner helps to better understand their roles in cells. $Stat3$ was reported to stabilize $Hif1\alpha$ (Jung *et al.*, 2008). Its activation on $Hif1\alpha$ was also present in the network of Dixit *et al.* (2016). In our model, this connection is only present in one of our networks and is indirect (i.e., via additional nodes). Although non-identifiability issue reduces the accuracy of M&NEM (Pirkl and Beerenwinkel, 2018), our model suggests that extra genes may involve in known gene pathways.

In conclusion, M&NEM can predict a mixture of causal gene networks involving over 20 genes from high-throughput perturbation data. Future study may focus on including cells with multiple sgRNAs detected (i.e., multiple S-gene perturbed) to investigate accumulative perturbation effects.

5 Implementation

Code is available from `https://github.com/dominicdongwuxing/mnem`.

Bibliography

Barabási, A.-L. *et al.* (2010). Network medicine: a network-based approach to human disease. *Nature Reviews Genetics*, **12**, 56.

Chacón, J. and Duong, T. (2018). *Multivariate Kernel Smoothing and Its Applications*. CRC Press.

Dixit, A. *et al.* (2016). Perturb-seq: Dissecting molecular circuits with scalable single cell rna profiling of pooled genetic screens. *Cell*, **167**, 1853–1866.e17.

Duong, T. (2016). Non-parametric smoothed estimation of multivariate cumulative distribution and survival functions, and receiver operating characteristic curves. *Journal of the Korean Statistical Society*, **45**, 33.

Fisher, R. *et al.* (2013). Cancer heterogeneity: implications for targeted therapeutics. *British Journal of Cancer*, **108**, 479.

Gaudet, S. and Miller-Jensen, K. (2016). Redefining signaling pathways with an expanding single-cell toolbox. *Trends in Biotechnology*, **34**, 458.

Hashemiaghdama, A. and Mroczek, M. (2020). Microglia heterogeneity and neurodegeneration: The emerging paradigm of the role of immunity in alzheimer's disease. *Journal of Neuroimmunology*, **341**, 577185.

Jung, J.-E. *et al.* (2008). Stat3 inhibits the degradation of hif-1α by pvhl-mediated ubiquitination. *Experimental & Molecular Medicine*, **40**, 479.

Markowetz, F. *et al.* (2005). Non-transcriptional pathway features reconstructed from secondary effects of rna interference. *Bioinformatics*, **21**, 4026.

Markowetz, F. *et al.* (2007). Nested effects models for high-dimensional phenotyping screens. *Bioinformatics*, **23**, i305.

Marusyk, A. *et al.* (2012). Intra-tumour heterogeneity: a looking glass for cancer? *Nature Reviews Cancer*, **12**, 323.

Meacham, C. and Morrison, S. (2013). Tumour heterogeneity and cancer cell plasticity. *Nature*, **501**, 328.

Pirkl, M. and Beerenwinkel, N. (2018). Single cell network analysis with a mixture of nested effects models. *Bioinformatics*, **34**, i964.

Pirkl, M. and Beerenwinkel, N. (2019). Mixture nested effects models simultaneous inference of causal networks and corresponding subpopulations from single cell perturbation data.

Prinz, M. *et al.* (2011). Heterogeneity of cns myeloid cells and their roles in neurodegeneration. *Nature Neuroscience*, **14**, 1227.

Yip, S. *et al.* (2017). Linnorm: improved statistical analysis for single cell rna-seq expression data. *Nucleic Acids Research*, **45**, e179.

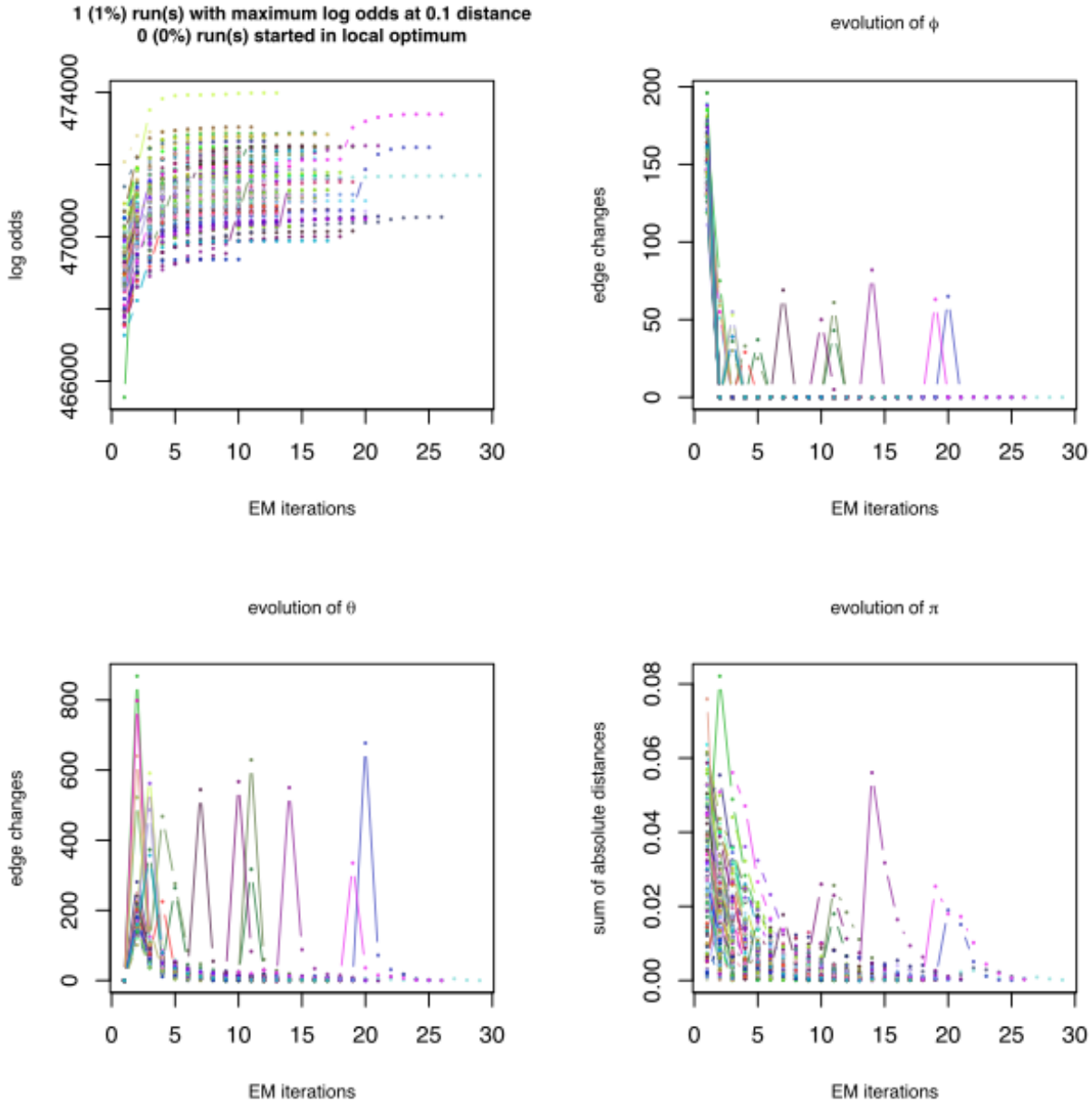

Figure A.1: Evolution of log odds, signaling network ϕ, E-gene attachment θ, and mixture weights π over the EM iterations of the M&NEM model trained with optimal hyper-parameters: $K = 2$, initialization with "networks", calculation of observed log likelihood ratio. Graph was produced by "plotConvergence" function from the "mnem" package.