**Supporting Information Text**

**Supplementary Note 1: A systematic evaluation of gene co-expression measurements**

Assessing associations between genes is challenging for single cell data since it’s measurements are typically sparse and noisy (Lähnemann et al. 2020). While some studies perform data imputation before measuring gene association, it remains a controversial issue how these corrections may affect the gene association qualifications (Ly and Vingron 2021; Stone et al. 2021). To select an appropriate metric, we chose seven gene co-expression strength measures and evaluated their robustness and stability systematically. The seven gene metrics were chosen either due to their broad use in gene co-expression analysis or they were originally designed for single cell data. Here, we provide a brief overview of each of the seven metrics.

1, The Pearson’s Correlation Coefficient (PCC) measures the linear relationship between two variables assuming a bivariate normal distribution, and it is the most widely used metric in co-expression analysis (Song et al. 2012)

2, is a measure of proportionality and quantifies the association between two log-ratio transformed gene expression vectors and yields an output analogous to a correlation matrix (Lovell et al. 2015). We used the function ‘perb’ from the R package ‘propr’ to perform the calculation (Quinn et al. 2017).

3, The co-dependency index (CDI) was proposed as a measure of association for sparse single-cell count data (Nakajima et al. 2021), and quantifies the degree of deviation of the joint gene co-occurrence frequency from that expected given the gene marginal probabilities. For gene i and j, in a total of N cells, cells have non-zero gene expression for gene i, cells for gene j, and cells have non-zero gene expression for both gene i and gene j. If these two genes are independent, we expect to see equal to /n \* /n. The co-dependency of genes is measured based on the deviation of these two terms. Denoting the observed frequency of both gene i and gene j being active as , and assuming a binomial distribution as a null model, the co-dependency index (CDI) is calculated as:

)

4, Bayesian correlation was proposed to measure gene expression similarity in single cell data (Sanchez-Taltavull et al. 2020). It assumes a beta distribution for individual gene expression and then updates this distribution with observed counts. The Bayesian correlation is calculated with Pearson correlation coefficients of the expectations of the updated beta distributions. We used the R code provided by the original publication to compute the correlation matrix (https://github.com/dsancheztaltavull/Bayesian-Correlations/).

5, ‘sclink’ was proposed to measure gene correlations in single cell data (Li and Li 2021). It models each individual gene expression using a Gamma-Normal mixture distribution then calculates a non-detection probability for each gene in each cell with the estimated parameters after fitting this two-component model. The correlation of a gene pair is calculated only with those cells whose non-detection probability is below a certain threshold. We used the ‘sclink\_norm’ function from the R package ‘sclink’ to do the calculation (https://github.com/Vivianstats/scLink).

6, ‘bigSCale2’ was proposed to measure gene correlations in single cell data (Iacono et al. 2019). It first groups cells into homogenous cell clusters, then performs differential expression (DE) analysis between all pairs of clusters. With N clusters, we obtain N\*(N-1)/2 unique comparisons and each comparison generates one Z-score for each gene, indicating the likelihood of an expression change between the corresponding two clusters. Finally, bigScale2 uses transformed Z-scores instead of original expression values to calculate Pearson correlation coefficients. We used the ‘compute.network’ function from the R package ‘bigSCale2’ to compute the gene correlation matrix (https://github.com/iaconogi/bigSCale2).

7, It was suggested that one way to cope with the sparseness in single cells is to aggregate data from similar cells on a nearest neighbor graph as pseudocells to increase gene representation (Feregrino and Tschopp 2021). We used the ‘construct\_metacells’ function from the R package ‘scWGCNA’ to acquire the pseudocells and then calculate the Pearson correlation coefficient matrix (https://github.com/smorabit/scWGCNA).

We implemented a single interface to access all 7 measures of gene co-expression which can be found at github: xxxx.

In our evaluation framework, we selected one cell cluster (cell cluster 1 in females) from fly brain single-cell data which contain more than 2000 cells. We filtered out genes which express in less than 10% of the cells, which resulted in a 2876 genes by 2023 cells gene count matrix. This count data was normalized using the ‘scran’ R package (Lun et al. 2016). We sampled n (n=400, 1000, 2000) cells with no replacement and split them into two groups. This procedure led to six datasets, two with 200 cells, two with 500 cells, and two with 1000 cells. Pairwise gene co-expression matrix was calculated with the selected metrics for each dataset independently.

To assess the robustness of different metrics, we compared the results of pairwise datasets. We first sorted edges by their co-expression strength from large to small within each dataset, then selected the top K edges from each dataset. For two datasets, we define a ‘agreement score’ as the number of intersected edges of the two top K edge sets divided by K. We calculated this score for K values at 1000, 2000, 5000, 10000, 20000, 50000, 100000.

In **Figure S6**, we observed that as the number of cells increases, the agreement between two datasets increases, and the agreement score was highest with the PCC metric. Given this result, we chose the PCC to measure cell type-specific gene co-expression in the main text.

**Supplementary Note 2: Consensus matrix computation by multi-view non-negative matrix factorization**

Matrix factorization (MF) is a popular approach in single cell data analysis and it has been used before in extracting gene co-expression patterns (Stein-O’Brien et al. 2018). Multi-view non-negative matrix factorization (mvNMF) builds on the flexible framework of non-negative matrix factorization (NMF) and has showed great potential in gaining insight into large-scale single cell datasets (Rappoport and Shamir 2018; Nguyen and Wang 2020). Here, we describe the relationship between the general NMF technique and the mvNMF algorithm we used in our pipeline.

**Standard non-negative matrix factorization (NMF)**

For a matrix X with *p* features as rows and *n* observations as columns, NMF two non-negative matrices W and H whose product approximates the original matrix X (Lee and Seung 1999).

NMF solves this by optimizing the objective function defined as:

where ||.|| denotes the Frobenius matrix norm, represent the constraint that all the matrix elements are non-negative.

The solution *U* is referred to as a basis matrix and *V* the coefficient matrix. The number of columns in *U* is smaller than both *p* and *n*, which is typically pre-defined by the user. Every column of *U* is a basis, whose *p* entries are the weights of the *p* genes. In NMF, all weights are non-negative, leading to a sparse *U* with most weights as zeros.

**Multi-view NMF to find common pattern**

If a dataset provides a view of a system, then multiple datasets integrated together offer compatible and complementary information to pinpoint the underlying shared structures of this system. Multi-view NMF was proposed based on this idea and itis essentially a joint matrix factorization with certain constraints, designed to find the common patterns among input multiple matrices (Liu et al. 2013). Specifically, it added a soft regulation term which put each matrix *V* towards a common consensus. This common pattern reconstruction task is formulated as a Frobenius norm optimization problem, defined as:

Where parameter tunes the relative weight among different views. The Matrix *V\** is referred to as the consensus matrix and it can be interpreted as the underlying common structure among input datasets.

**Multi-view NMF on symmetric matrix**

If the multiple input matrix are symmetric, such as gene co-expression matrix, the multi-view NMF problem is translates into optimizing the following objective function:

Where is a symmetric matrix. In our pipeline, we included two feature matrix as two views, which corresponds to = 2. We refer to the matrix as the consensus matrix and each of its column as a consensus factor to be used for gene selection forming modules.

The objective function as above is not a convex function with respect to and together, and has multiple local minimums. Thus, previous studies have proposed a multiplicative update algorithm to find a local minimum of this problem. We used the algorithm developed in (Wang et al. 2018) and implemented it into a custom R code to find the solutions.

**Supplementary Note 3: Parameter choices in the pipeline**

The parameters used in our pipeline involves the weights of the two feature matrices, the number of columns (denoted as *K)* in the consensus matrix , and the cutoff for selecting genes (denoted as T) from the columns of into each module.

For the weights of the two feature matrices, they reflect both the relative importance of each view as well as the regularization constraint. A larger weight value enforces reaching consensus across views and a smaller one cannot tolerate matrix factorization error. We follow the recommendation proposed in (Liu et al. 2013), assigning equal weight to both views at value 0.01.

For the cutoff used in selecting genes into each module, T, a larger cutoff leads to a stronger significant signal in the consensus factors and smaller sizes of modules. As the gene importance in each consensus factor is z score transformed, a theta value larger than 2 can be interpreted as a p-value smaller than 0.05 is assuming a null normal distribution. We chose a theta cutoff at 2.5 as its associated p-value cutoff is around 0.01 which would give small sizes of modules and ensure a high accuracy.

For the number of columns in the consensus matrix , *K*, we used a random matrix theory (RMT) to guide parameter value selection which is explained in detail below.

**Determining the number of gene modules in single-cell RNA-seq data**

Determining the number of gene expression modules of a scRNA-seq dataset is an important but challenging task. To address this issue, we implemented a method to estimate the number of dimensions of a given gene correlation matrix based on random matrix theory (RMT) (Tracy and Widom 1994; Tao and Vu 2010).

This method compares the observed eigenvalues of a gene correlation matrix with those expected under a null model to estimate the number of dimensions. Previous studies have shown that RMT can be used to estimate the number of latent cell states and the number of independent gene expression programs of single-cell data (Kiselev et al. 2017; Nitzan and Brenner 2021).

If *X* is a matrix containing N genes as rows and L cells as columns, assuming each column is independently drawn from a distribution with mean zero and variance , the ‘random’ correlation matrix is given by

The properties of eigenvalues and eigenvectors of random matrices are described by RMT. In the limit, P → , N→ , such that Q = N/L is fixed. The analytical distribution of eigenvalues of the random correlation matrix is given by:

where, and are the maximum and minimum eigenvalues of R, respectively, given by

Thus, the theoretical maximal and minimal eigenvalues for any random matrices are known.

We exploit this theory and estimate the number of gene modules as the number of eigenvalues that are significantly different from the analytical maximum eigenvalue.

In practice, we used the ‘svd’ function in R to compute the eigenvalues of a correlation matrix, as for a N×N real symmetric matrix with non-negative eigenvalues, the eigenvalues and singular values coincide. Then, we find the largest *K* singular values exceeding the theoretical maximal values as the estimated number of gene modules. With *m* gene correlation matrix generated from *m* cell clusters, each would generate one *k* value, we chose the largest one to be the parameter value used in running the pipeline.

**Supplementary Note 4: Evaluation of the pipeline performance through network simulations**

To evaluate the performance and accuracy of the pipeline, we simulated two sets of networks, the one containing common modules shared by all or subset of networks, the other containing overlapped common modules by all or subset of networks. In both simulation schemes, we simulated weighted networks by adding Gaussian noise.

In the first simulation scheme, we simulated 100 networks, each containing 1000 genes. We generate 5 modules of 30 gene members each. The 5 modules were randomly assigned to 10, 20, 50, 80 100 networks, respectively. The resultant simulated networks contain 1 or up to 5 modules. We connected genes irrespective of their module membership with a probability , and connect genes inside a module with a probability of inside each module. To introduce edge weights in the simulated networks, we added Gaussian noise with two sets of parameters, the first with mean 0.25 and variance 0.1, and the second with mean 0.3 and variance 0.2.

In the second simulation scheme, we simulated 90 networks each containing 1000 genes. In each network, a module consists of two parts, a shared part which is common to a set of networks, and a specific part which is present only in this individual network.

We set two overlapped modules, M1 and M2. M1 contains 60 genes as the shared part and M2 contains 40 genes as the shared part. For simulated network 1 to 30, only M1 exists, for simulated network 31 to 60, both M1 and M2 exist, for simulated networks 61 to 90, only M2 exists.

Similar as above, in the network background, genes are connected with each other with probability , and genes inside each module are connected with each other with a probability of . The simulated binary networks are transformed into weighted networks adding Gaussian noise.

In simulations, we set α = 0.1, 0.3, 0.5 and 0.7 and β = 0.05. Lower value of α means modules are relatively loosely connected and are harder to detect.

To evaluate our pipeline performance, we use true positive rate (TPR), false positive rate (FPR) and Matthew’s correlation coefficient (MCC) to quantify the performance of methods (Xiao et al. 2014).

True positive (TP) case is defined as two related genes in the simulation setup are assigned to the same module after running the algorithm. A false positive (FP) case assigns two unrelated genes assigned to the same module. A true negative (TN) case assigns two unrelated nodes to different modules and a false negative (FN) case two related nodes to different modules. Typically, a TPR=1 and the corresponding FPR=0 indicate a perfect prediction. MCC values range between -1 and 1, with 1 means a perfect prediction, 0 means no better than random prediction and − 1 shows total disagreement between prediction and observation.

When running our pipeline on the simulated datasets, we used the parameters λ = 0.01 for both views, *T* = 1.5, *K* = 5 for the first simulation scheme and 2 for the second. Each simulation parameter combination has 20 replicates, the average results are shown in the figures.

**Figure S7** showed that our pipeline can accurately identify the common modules in both simulation schemes and the accuracy is higher for larger values and lower levels of Gaussian noise.

**References**

Feregrino C, Tschopp P. 2021. Assessing evolutionary and developmental transcriptome dynamics in homologous cell types. *Dev Dyn*.

Iacono G, Massoni-Badosa R, Heyn H. 2019. Single-cell transcriptomics unveils gene regulatory network plasticity. *Genome Biol* **20**: 1–20.

Kiselev VY, Kirschner K, Schaub MT, Andrews T, Yiu A, Chandra T, Natarajan KN, Reik W, Barahona M, Green AR. 2017. SC3: consensus clustering of single-cell RNA-seq data. *Nat Methods* **14**: 483–486.

Lähnemann D, Köster J, Szczurek E, McCarthy DJ, Hicks SC, Robinson MD, Vallejos CA, Campbell KR, Beerenwinkel N, Mahfouz A. 2020. Eleven grand challenges in single-cell data science. *Genome Biol* **21**: 1–35.

Lee DD, Seung HS. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* **401**: 788–791.

Li WV, Li Y. 2021. sclink: Inferring sparse gene co-expression networks from single-cell expression data. *Genomics Proteomics Bioinformatics*.

Liu J, Wang C, Gao J, Han J. 2013. Multi-view clustering via joint nonnegative matrix factorization. In *Proceedings of the 2013 SIAM international conference on data mining*, pp. 252–260, SIAM.

Lovell D, Pawlowsky-Glahn V, Egozcue JJ, Marguerat S, Bähler J. 2015. Proportionality: a valid alternative to correlation for relative data. *PLoS Comput Biol* **11**: e1004075.

Lun ATL, Bach K, Marioni JC. 2016. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biol* **17**: 1–14.

Ly L-H, Vingron M. 2021. Effect of imputation on gene network reconstruction from single-cell RNA-seq data. *bioRxiv*.

Nakajima N, Hayashi T, Fujiki K, Shirahige K, Akiyama T, Akutsu T, Nakato R. 2021. Codependency and mutual exclusivity for gene community detection from sparse single-cell transcriptome data. *Nucleic Acids Res* **49**: e104.

Nguyen ND, Wang D. 2020. Multiview learning for understanding functional multiomics. *PLoS Comput Biol* **16**: e1007677.

Nitzan M, Brenner MP. 2021. Revealing lineage-related signals in single-cell gene expression using random matrix theory. *Proc Natl Acad Sci* **118**.

Quinn TP, Richardson MF, Lovell D, Crowley TM. 2017. propr: an R-package for identifying proportionally abundant features using compositional data analysis. *Sci Rep* **7**: 1–9.

Rappoport N, Shamir R. 2018. Multi-omic and multi-view clustering algorithms: review and cancer benchmark. *Nucleic Acids Res* **46**: 10546–10562.

Sanchez-Taltavull D, Perkins TJ, Dommann N, Melin N, Keogh A, Candinas D, Stroka D, Beldi G. 2020. Bayesian correlation is a robust gene similarity measure for single-cell RNA-seq data. *NAR genomics Bioinforma* **2**: lqaa002.

Song L, Langfelder P, Horvath S. 2012. Comparison of co-expression measures: mutual information, correlation, and model based indices. *BMC Bioinformatics* **13**: 1–21.

Stein-O’Brien GL, Arora R, Culhane AC, Favorov A V., Garmire LX, Greene CS, Goff LA, Li Y, Ngom A, Ochs MF, et al. 2018. Enter the Matrix: Factorization Uncovers Knowledge from Omics. *Trends Genet* **34**: 790–805. https://doi.org/10.1016/j.tig.2018.07.003.

Stone M, McCalla SG, Siahpirani AF, Periyasamy V, Shin J, Roy S. 2021. Identifying strengths and weaknesses of methods for computational network inference from single cell RNA-seq data. *bioRxiv*.

Tao T, Vu V. 2010. Random matrices: Universality of local eigenvalue statistics up to the edge. *Commun Math Phys* **298**: 549–572.

Tracy CA, Widom H. 1994. Level-spacing distributions and the Airy kernel. *Commun Math Phys* **159**: 151–174.

Wang P, Gao L, Hu Y, Li F. 2018. Feature related multi-view nonnegative matrix factorization for identifying conserved functional modules in multiple biological networks. *BMC Bioinformatics* **19**: 1–13.

Xiao X, Moreno-Moral A, Rotival M, Bottolo L, Petretto E. 2014. Multi-tissue analysis of co-expression networks by higher-order generalized singular value decomposition identifies functionally coherent transcriptional modules. *PLoS Genet* **10**: e1004006.