Journal Title Here, 2022, pp. 1–6

doi: DOI HERE
Advance Access Publication Date: Day Month Year
Paper

PAPER

Starlng: STability Analysis of coReguLated ”Nests” of Genes

Andi Munteanu [1,2*]

[1]Faculty of Computer Science, Alexandru Ioan Cuza University, Iasi, Romania and [2]Cambridge Stem Cell Institute, University of Cambridge, CB2 0AW, UK
[*]Corresponding author. am3019@cam.ac.uk

Abstract

Single-cell assays (both single and multi-omics) represent the state-of-the-art for detailed characterisation of (cell) phenotypes, a task frequently initialised with the identification of groups of genes linked to well-defined, stable subpopulations of cells. Gene expression covariation patterns are simplified precursors of Gene Regulatory Networks (GRNs), and play a crucial role in inferring causality and dynamics within GRNs. Expanding on the ClustAssess framework, which shifted the cluster-centric assessment of populations of cells to an element-centric perspective, we propose to assess the covariation induced by gene graph clustering. The Starlng (abbrev) framework provides an optimised, interactive environment for the identification of the filtered modules. The pipeline identifies the stable configuration of clustering parameters and robust gene modules (nests of genes), followed by the removal of the outlier clusters based on the pseudotime variance. We illustrate Starlng on the identification of reliable, reproducible communities of genes, as well as reveal unbiased hierarchical relationships between higher-level cell clusters and more granular subgroups of genes.

Key words: single-cell omics, clustering, gene modules, gene regulatory network, reproducibility, data accessibility

Introduction

The fast-paced development of single-cell technologies across modalities led to significant breakthroughs, both in terms of sequencing outputs and dynamic ranges, and in the number of samples per experiment, thus expanding the scope of biological studies (1). These advancements present ongoing challenges in identifying heterogeneous cell subpopulations, a crucial step for characterising the transitions between different cell types (2) or for understanding cell fate decisions and dynamic cellular processes (3).

A focus of single-cell methods is linking gene expression patterns to observed phenotypes. The transition from manual curation to the use of automated, computational approaches promoted significant discoveries that expanded our understanding of gene functional mechanisms [reference]. These approaches rely on various techniques that range from extrapolating a manually annotated database (4) of markers to statistical models and machine learning algorithms that focus on correlation and co-expression patterns of genes (5; 6).

The assessment of gene-centric pairwise covariation patterns offered the opportunity of assessing regulatory interactions, leading to an incremental construction of comprehensive gene regulatory networks (GRNs) at the genome-wide level (7; 8). Recent studies reinforce this, showing how GRNs successfully capture behavioural patterns globally and locally (9).

Current research efforts aim to enhance the granularity of GRN inference methods, which were traditionally applied to bulk transcriptomics data (5). There is growing interest in improving the specificity of local networks (10; 11), identifying key regulatory hubs (12; 13), and uncovering dynamic interaction patterns across multiple developmental stages (14; 15). The computational approaches addressing this task span a wide spectrum of algorithms from statistical models to machine learning-inspired methods, all with an emphasis on flexibility in defining local network structures and enabling the exploration of hierarchical relationships within regulatory systems.

A significant challenge in GRN inference is represented by the robustness and reproducibility of the results (16). Benchmark studies on real-world single-cell datasets revealed substantial variability in network inference outcomes across methods and datasets, underscoring the need for frameworks that assess and ensure the stability of GRN results (17; 18; 19; 20). Addressing the bias in choosing the target group of genes (7) would be a step forward in ensuring the robustness of the results.

To provide a bridge between stability analyses and inference of covariation patterns, and ultimately GRNs, we introduce Starlng (STability Analysis of coRreguLated Nests of Genes), a framework designed to evaluate and improve the robustness of graph-based gene clustering, leading to the identification of gene modules that characterise transitions between cell states across pseudotime trajectories. Starlng relies on stability assessment principles described and utilised in the ClustAssess framework (21) to find optimal configurations of parameters and gene modules, highly reproducible across multiple runs. Starlng determines the quality of the resulting modules and automatically discards the outliers. The filtered clusters are then used to define and characterise the subpopulation that drives the transition.

Results

Monocle3 (22) was one of the first frameworks which used the PhenoGraph pipeline (23) to determine the modules of coexpressed genes. Acknowledging the coexistence of signal and noise (24), this approach was accompanied by a pre-filtering of the genes based on an autocorrelation metric. Monocle3 opted to apply Moran's I test on the inferred trajectory graph.

Starlng Framework

To enhance and optimise the Monocle approach, we developed Starlng, a framework written in R, whose aim is to identify stable, relevant gene modules that describe cell populations and the transition between cell states. The workflow is presented in Fig. 1 A.

Firstly, Starlng calculates the trajectory graph using an extended version of the `learn_graph` function from Monocle3 that provides better control on the parameters, i.e. the the ratio between the graph nodes and the size of the data. This is immediately followed by the inference of the pseudotime ordering. For this step, we have employed a heuristic that searches through existing categorical metadata and selects the cell group whose centroid yields the highest pseudotime range.

Afterwards, the autocorrelation score is calculated and then used to filter out genes that do not show population-specific patterns. The remaining features are then clustered using the Leiden community detection algorithm (25). For clustering, we follow the same principles as described in the ClustAssess framework (21): the Element-Centric Consistency (ECC) score is used to infer the most stable configuration of parameters and number of gene modules. We note that, in comparison to ClustAssess, the first step of dimensionality reduction is not assessed. This is justified by the choice of using all cells to generate the feature loading from PCA that will act as input.

Finally, the results are incorporated into a single interactive Shiny web application that expands the analysis of the modules and their functions.

Applying Starlng on experimental data

We will exemplify the use of the Starlng framework on the immune subset of the Griben et al. MASLD\MASH dataset (2). The Starlng run identifies the 40-Shared Nearest Neighbour graph as the optimal configuration in terms of stability. The available number of modules ranges between 4 and 38. We show the stability of each value, defined by the average ECC score and the frequency. The highest number of stable modules, 37, was selected. The partitions identified as outliers or redundant were excluded from the analysis, leaving 29 usable modules.

Projecting the filtered modules on a 2D UMAP space, as shown in Fig. 1 B, shows a clear grouping in three main partitions. The violin plot from Fig. 1 C contains the pseudotime distribution of the cells that are defined by each gene module. It confirms the previous remark and shows three main partitions: from module 29 to 31, from 26 to 1 and from 11 to 21.

To understand the identity of the module-defined populations, we display their unified distribution on the reduced space of the cells. Fig. 1 D shows that all three main cell types are well represented by more than one module. This plot also helps us map the relationship between the three main partitions and the cell annotations. Analysing panels C and D together drives the conclusion that the modules display a good complementarity in describing a particular transitioning state across the pseudotime ordering.

Fig. 1 E assists us in understanding the coverage of the gene communities. Every ClustAssess cluster is well-represented by at least one module. Moreover, we show that the entire pseudotime landscape is described by one or multiple gene clusters.

Finally, we have employed an enrichment analysis on each module. Fig. 1 F shows the first 20 biological process terms according to significance. We note the presence of function terms that can be linked to particular cell types, such as cell killing in Lymphocytes (26), cholesterol efflux in Macrophages (27), and antigen processing via mast cells in the transition between Neutrophils and Macrophages (28).

Comparison with Scenic

In the previous sections, we presented the workflow of Starlng and its ability to capture the biological identity of the cell populations. In this section, we will compare the results with Scenic (7), a popular tool that identifies gene modules by binding Transcription Factors to candidate target genes. In this comparison, we will introduce two runs of Starlng: one with the default filtering (Moran's I > 0.1) and one with a more permissive threshold (Moran's I > 0.02).

The first comparison regards the percentage of usability of the resulting gene modules. The concept of usability has already been introduced in the previous sections and describes the gene clusters that are not outliers or redundant. The top left panel of Fig. 1 G shows the distribution of the three categories of modules. We note a high percentage of both outlier and redundant Scenic modules; in comparison, both Starlng runs achieve a lower proportion of modules belonging to these two categories. Moreover, the default filtering noticeably improves the usability percentage from 7% and 50% (achieved by Scenic and the Starlng run with low filtering) to 75%.

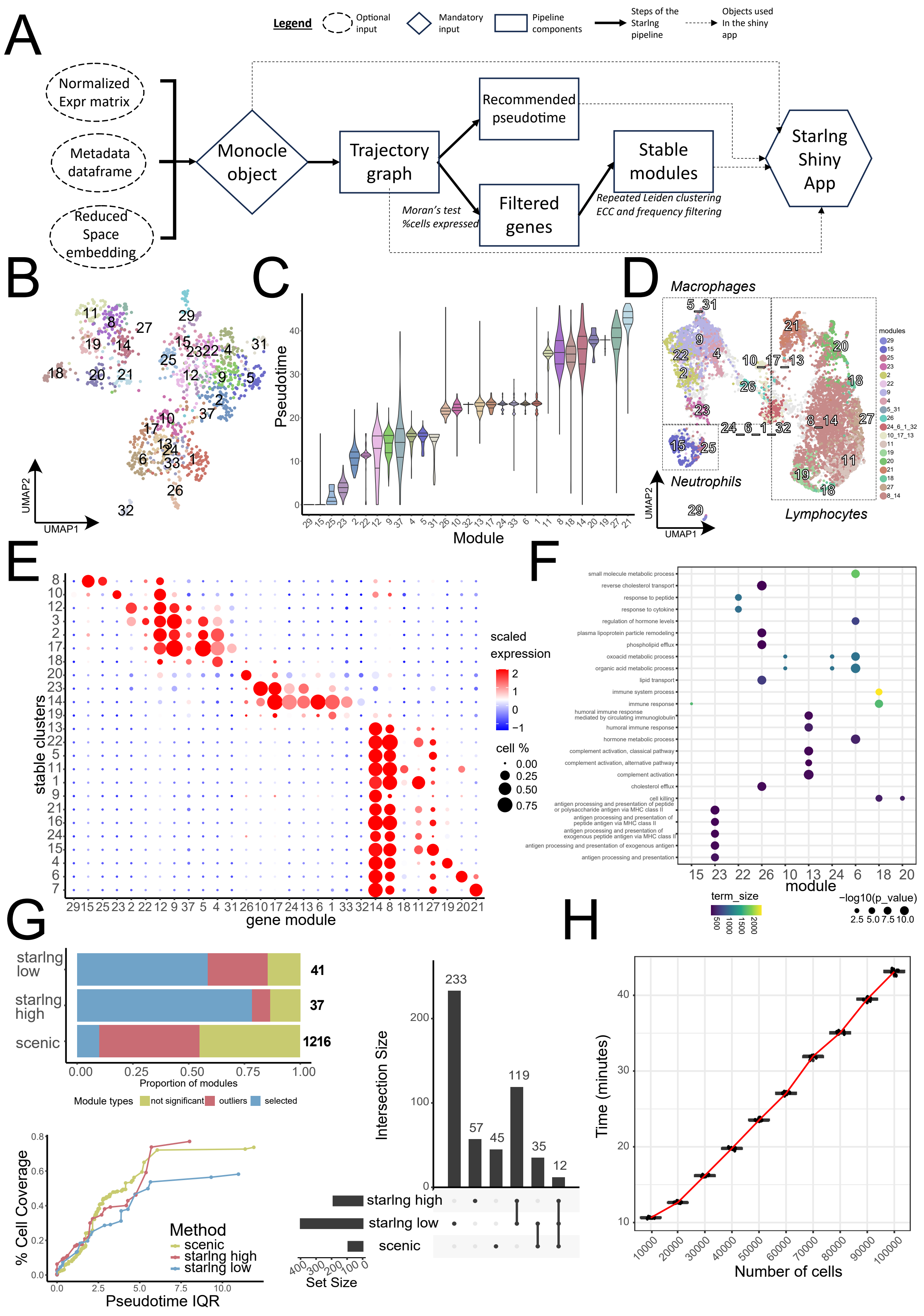
A
Legend
Optional input
Mandatory input
Pipeline components
Steps of the Starlng pipeline
Objects used In the shiny app
Normalized Expr matrix
Metadata dataframe
Reduced Space embedding
Monocle object
Trajectory graph
Recommended pseudotime
Filtered genes
Moran's test
%cells expressed
Stable modules
Repeated Leiden clustering
ECC and frequency filtering
Starlng Shiny App
B
UMAP1
UMAP2
C
Pseudotime
Module
D
Macrophages
Neutrophils
Lymphocytes
modules
UMAP1
UMAP2
E
stable clusters
gene module
scaled expression
cell %
F
small molecule metabolic process
reverse cholesterol transport
response to peptide
response to cytokine
regulation of hormone levels
plasma lipoprotein particle remodeling
phospholipid efflux
oxoacid metabolic process
organic acid metabolic process
lipid transport
immune system process
immune response
humoral immune response mediated by circulating immunoglobulin
humoral immune response
hormone metabolic process
complement activation, classical pathway
complement activation, alternative pathway
complement activation
cholesterol efflux
cell killing
antigen processing and presentation of peptide or polysaccharide antigen via MHC class II
antigen processing and presentation of peptide antigen via MHC class II
antigen processing and presentation of exogenous peptide antigen via MHC class II
antigen processing and presentation of exogenous antigen
antigen processing and presentation
module
term_size
-log10(p_value)
G
starlng low
starlng high
scenic
41
37
1216
Proportion of modules
Module types
not significant
outliers
selected
% Cell Coverage
Pseudotime IQR
Method
scenic
starlng high
starlng low
Intersection Size
233
57
45
119
35
12
Set Size
H
Time (minutes)
Number of cells

Fig. 1. Starlng workflow and results A. Workflow diagram showing the main steps of the Starlng framework. B. 2D representation of the genes, coloured by the modules identified and filtered by Starlng. The UMAP was built on the feature loading calculated using PCA at the cell level. C. Violin plot showing the distribution of pseudotime (y-axis) of the filtered Starlng gene modules (x-axis). Three main groups can be noticed. The plot also helps visualise the tightness of the distribution, as well as the transitioning points (such as modules 23, 2, 22). D. 2D representation of the cell populations defined by the filtered Starlng gene modules. The cell UMAP was used to show the coverage of the inferred population across all three cell types. E. Bubbleplot showing the association between the stable 24 cell clusters identified by ClustAssess and the filtered gene modules. The size of the point is proportional to the number of cells from the clusters that are identified by a module. The colour gradient indicates the strength of the scaled average expression of the genes belonging to the module. F. Bubbleplot showing the modules that were used to identify the top 20 enriched GO:BP terms. The size of the point is proportional to the pvalue significance. The colour gradient reflects the number of associated genes from the database given the term. G. Comparison results between Starlng and Scenic. Top-left panel represents a barplot showing the percentages of filtered (blue), redundant (yellow) and outlier (red) modules. The number of clusters is displayed on the right of each barplot. Bottom-left panel illustrates the cumulative coverage of cells from the gene modules while progressing on the pseudotime IQR. Right panel is an UpSet plot showing the intersection of the set of enriched terms. H. Boxplot presenting the execution runtime, measured in minutes (y-axis) of the Starlng frameworks on datasets with sizes ranging from 10,000 to 100,000 (x-axis). We notice a tight distribution and a virtually linear relationship between the two dimensions.

The quality of the usable modules was assessed in the second experiment and is illustrated in the bottom left panel of Fig. 1 G. For each module, we identify the unique population of cells; this information is used to calculate the coverage, i.e. the proportion of cells that are described by the gene clusters. The plot shows close values for the total coverage between Starlng and Scenic. However, we note that the former uses modules with lower pseudotime variance than the latter (7.5 maximum IQR vs 11).
Finally, we compare the methods based on the enrichment analysis results. The right panel of Fig. 1 G shows that Starlng identifies more terms from the GO:BP database than Scenic. We note that this is an overall pattern across all databases except TF, where Scenic scores better. Using transcription factors as drivers for clustering might explain part of this behaviour.

Discussion

Our findings highlight the need to define quality metrics that decide whether a gene module can be used in the analysis or not. Cluster filtering has improved the separation of module partitions. The results indicate that the filtered modules can be confidently used to describe any cell type in terms of coverage and functional interpretation. With the development of the Starlng shiny app, we have also addressed the challenge of mediating data interrogation towards any user, irrespective of their informatics experience. Using the ClustAssess approach of assessing the stability ensures similar outcomes in terms of the robustness and reproducibility of the results.
We evaluated the scalability of the Starlng framework by running the default configuration (except for the autocorrelation threshold, which was lowered from 0.1 to 0) on a synthetic dataset with cells from 10,000 to 100,000 and 3,000 genes. The experiment was run on 30 parallel threads.
Fig. 1 H showcases the linear relationship between runtime and the size of the dataset. This is expected, as the majority of the steps have a time complexity proportional to the number of genes. The identification of the feature loading represents the main exception of this behaviour, as the initial PCA is applied on the entire matrix. For the largest dataset, 45 minutes were required to execute the whole pipeline. Increasing the number of cores from 1 to 5 already achieves a two-fold improvement in the runtime. We have attempted to measure the amount of memory used during execution. We conclude that there is no direct relationship between the number of cells and the memory used. The presence of spikes advises using fewer cores on personal machines when running Starlng on a moderately large dataset. We note that the memory evaluation is not entirely precise and relies on querying the system-wide usage every 5 seconds.
Further steps in the development of this idea would be linked to assessing both the stability and the quality of the gene modules in relationship with the pseudotime ordering and to the improvement of the relationship between the localised gene expression profiles and the inferred trajectory. Extending the flexibility towards other methods of pseudotime trajectory inference (29) would also benefit the generalisability of the package.

Materials and Methods

Starlng was applied to the immune cells from a MASLD-MASH study (2). Quality checks and filters were aligned with the original manuscript. The normalisation and scaling of the count matrix was performed using the `SCTransform` function from Seurat v.5.1.0 (30).
The trajectory graph, pseudotime ordering and identification of auto-correlated genes were performed using the Monocle3 package v1.3.7(22).
The feature loadings of the filtered genes were determined using `prcomp` (base R package v4.4.0). The nearest neighbour graph was calculated using the RANN package v2.6.1 (11), converted to an igraph object v2.0.3 (31) and clustered using the R implementation of the Leiden (25) algorithm (leidenbase package v0.1.30 (32)). UMAP (33) was used for the 2D representation of genes via the uwot package v0.2.2.
The objects used in the Shiny application (34) were stored using the qs2 package v0.26.3 (35). The expression matrix was saved using the rhdf5 R package v2.48.0 (36). For visualisation, we used ggplot2 v3.5.1 (37) and ComplexHeatmap v2.20.0 (38) packages. The tables were rendered using the DT package v0.33 (39). The enrichment analysis was performed using the gprofiler2 package v0.2.3 (40).
To run Scenic, we have used the pyScenic package. The tool was installed using the latest version from the GitHub repository (commit 06bafba412792f6efa5a552a23bb221cc3bdea1b).

For benchmarking, we have applied the zinbwave R package v1.26.0 (41) to generate the synthetic data. This is done by using the given dataset to fit a zero-inflated negative binomial model. The simulated count matrix was then normalised using the `SCTransform` function from the Seurat package. The resulting normalised matrix was finally used as input for the StarIng pipeline.
All experiments were performed on a high-memory machine (Dell Precision 7780, Intel i9-13950HX processor with 32 cores, 128 GB RAM, Ubuntu 22.04.4 LTS Linux distro used under WSL2).

Code and availability

The StarIng package is publicly available as open-source on Github: `https://github.com/Core-Bioinformatics/StarIng` and is accessible via Zenodo at 10.5281/zenodo.17423753.

Competing interests

AM declares no competing interests.

Acknowledgments

The method was developed in the Core Bioinformatics group, Camrbdige Stem Cell Institute, University of Cambridge, under the supervision of Dr Irina Mohorianu. I acknowledge the feedback and constructive discussions with all members of the Core Bioinformatics Group and Dr Floris Roos, who tested the early versions of the StarIng Shiny app.

References

1. A. Regev, S. A. Teichmann, E. S. Lander, I. Amit, C. Benoist, E. Birney, B. Bodenmiller, P. Campbell, P. Carninci, M. Clatworthy, H. Clevers, B. Deplancke, I. Dunham, J. Eberwine, R. Eils, W. Enard, A. Farmer, L. Fugger, B. Göttgens, N. Hacohen, M. Haniffa, M. Hemberg, S. Kim, P. Klenerman, A. Kriegstein, E. Lein, S. Linnarsson, E. Lundberg, J. Lundeberg, P. Majumder, J. C. Marioni, M. Merad, M. Mhlanga, M. Nawijn, M. Netea, G. Nolan, D. Pe'er, A. Phillipakis, C. P. Ponting, S. Quake, W. Reik, O. Rozenblatt-Rosen, J. Sanes, R. Satija, T. N. Schumacher, A. Shalek, E. Shapiro, P. Sharma, J. W. Shin, O. Stegle, M. Stratton, M. J. T. Stubbington, F. J. Theis, M. Uhlen, A. van Oudenaarden, A. Wagner, F. Watt, J. Weissman, B. Wold, R. Xavier, and N. Yosef, "The human cell atlas," *eLife*, vol. 6, Dec. 2017.
2. C. Gribben, V. Galanakis, A. Calderwood, E. C. Williams, R. Chazarra-Gil, M. Larraz, C. Frau, T. Puengel, A. Guillot, F. J. Rouhani, K. Mahbubani, E. Godfrey, S. E. Davies, E. Athanasiadis, K. Saeb-Parsy, F. Tacke, M. Allison, I. Mohorianu, and L. Vallier, "Acquisition of epithelial plasticity in human chronic liver disease," *Nature*, vol. 630, p. 166–173, May 2024.
3. C. Trapnell, "Defining cell types and states with single-cell genomics," *Genome Research*, vol. 25, p. 1491–1498, Oct. 2015.
4. M. Stock, C. Losert, M. Zambon, N. Popp, G. Lubatti, E. Hörmanseder, M. Heinig, and A. Scialdone, "Leveraging prior knowledge to infer gene regulatory networks from single-cell rna-sequencing data," *Molecular Systems Biology*, vol. 21, p. 214–230, Feb. 2025.
5. P. Langfelder and S. Horvath, "Wgcna: an r package for weighted correlation network analysis," *BMC Bioinformatics*, vol. 9, Dec. 2008.
6. S. Morabito, F. Reese, N. Rahimzadeh, E. Miyoshi, and V. Swarup, "hdwgcna identifies co-expression networks in high-dimensional transcriptomics data," *Cell Reports Methods*, vol. 3, p. 100498, June 2023.
7. S. Aibar, C. B. González-Blas, T. Moerman, V. A. Huynh-Thu, H. Imrichova, G. Hulselmans, F. Rambow, J.-C. Marine, P. Geurts, J. Aerts, J. van den Oord, Z. K. Atak, J. Wouters, and S. Aerts, "Scenic: single-cell regulatory network inference and clustering," *Nature Methods*, vol. 14, p. 1083–1086, Oct. 2017.
8. V. A. Huynh-Thu, A. Irrthum, L. Wehenkel, and P. Geurts, "Inferring regulatory networks from expression data using tree-based methods," *PLoS ONE*, vol. 5, p. e12776, Sept. 2010.
9. C. L. Hartl, G. Ramaswami, W. G. Pembroke, S. Muller, G. Pintacuda, A. Saha, P. Parsana, A. Battle, K. Lage, and D. H. Geschwind, "Coexpression network architecture reveals the brain-wide and multiregional basis of disease susceptibility," *Nature Neuroscience*, vol. 24, p. 1313–1323, July 2021.
10. E. H. Cui, D. Song, W. K. Wong, and J. J. Li, "Single-cell generalized trend model (scgtm): a flexible and interpretable model of gene expression trend along cell pseudotime," *Bioinformatics*, vol. 38, p. 3927–3934, June 2022.
11. Y. Jia, H. Wu, and J. Ding, "scgenerythm: Using neural networks and fourier transformation to cluster genes by time-frequency patterns in single-cell data," Nov. 2023.
12. Q. Gao, Z. Ji, L. Wang, K. Owzar, Q.-J. Li, C. Chan, and J. Xie, "Sifinet: a robust and accurate method to identify feature gene sets and annotate cells," *Nucleic Acids Research*, vol. 52, p. e46–e46, Apr. 2024.
13. Q. Xu, G. Li, D. Osorio, Y. Zhong, Y. Yang, Y.-T. Lin, X. Zhang, and J. J. Cai, "scintime: A computational method leveraging single-cell trajectory and gene regulatory networks to identify master regulators of cellular differentiation," *Genes*, vol. 13, p. 371, Feb. 2022.
14. D. Song and J. J. Li, "Pseudotimede: inference of differential gene expression along cell pseudotime with well-calibrated p-values from single-cell rna sequencing data," *Genome Biology*, vol. 22, Apr. 2021.
15. Q. Song, J. Wang, and Z. Bar-Joseph, "scstem: clustering pseudotime ordered single-cell data," *Genome Biology*, vol. 23, July 2022.

16. D. Lähnemann, J. Köster, E. Szczurek, D. J. McCarthy, S. C. Hicks, M. D. Robinson, C. A. Vallejos, K. R. Campbell, N. Beerenwinkel, A. Mahfouz, L. Pinello, P. Skums, A. Stamatakis, C. S.-O. Attolini, S. Aparicio, J. Baaijens, M. Balvert, B. d. Barbanson, A. Cappuccio, G. Corleone, B. E. Dutilh, M. Florescu, V. Guryev, R. Holmer, K. Jahn, T. J. Lobo, E. M. Keizer, I. Khatri, S. M. Kielbasa, J. O. Korbel, A. M. Kozlov, T.-H. Kuo, B. P. Lelieveldt, I. I. Mandoiu, J. C. Marioni, T. Marschall, F. Mölder, A. Niknejad, A. Raczkowska, M. Reinders, J. d. Ridder, A.-E. Saliba, A. Somarakis, O. Stegle, F. J. Theis, H. Yang, A. Zelikovsky, A. C. McHardy, B. J. Raphael, S. P. Shah, and A. Schönhuth, "Eleven grand challenges in single-cell data science," *Genome Biology*, vol. 21, Feb. 2020.
17. P. Langfelder, R. Luo, M. C. Oldham, and S. Horvath, "Is my network module preserved and reproducible?," *PLoS Computational Biology*, vol. 7, p. e1001057, Jan. 2011.
18. W. Saelens, R. Cannoodt, and Y. Saeys, "A comprehensive evaluation of module detection methods for gene expression data," *Nature Communications*, vol. 9, Mar. 2018.
19. M. Chevalley, Y. H. Roohani, A. Mehrjou, J. Leskovec, and P. Schwab, "A large-scale benchmark for network inference from single-cell perturbation data," *Communications Biology*, vol. 8, Mar. 2025.
20. Y. Kang, D. Thieffry, and L. Cantini, "Evaluating the reproducibility of single-cell gene regulatory network inference algorithms," *Frontiers in Genetics*, vol. 12, Mar. 2021.
21. A. Shahsavari, A. Munteanu, and I. Mohorianu, "Clustassess: tools for assessing the robustness of single-cell clustering," *bioRxiv*, 2022.
22. J. Cao, M. Spielmann, X. Qiu, X. Huang, D. M. Ibrahim, A. J. Hill, F. Zhang, S. Mundlos, L. Christiansen, F. J. Steemers, C. Trapnell, and J. Shendure, "The single-cell transcriptional landscape of mammalian organogenesis," *Nature*, vol. 566, p. 496–502, Feb. 2019.
23. J. H. Levine, E. F. Simonds, S. C. Bendall, K. L. Davis, E.-a. D. Amir, M. D. Tadmor, O. Litvin, H. G. Fienberg, A. Jager, E. R. Zunder, R. Finck, A. L. Gedman, I. Radtke, J. R. Downing, D. Pe'er, and G. P. Nolan, "Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells that Correlate with Prognosis," *Cell*, vol. 162, pp. 184–197, July 2015.
24. I. Moutsopoulos, L. Maischak, E. Lauzikaite, S. A. Vasquez Urbina, E. C. Williams, H.-G. Drost, and I. I. Mohorianu, "noisyR: Enhancing biological signal in sequencing datasets by characterizing random technical noise," *Nucleic Acids Research*, vol. 49, p. e83, Aug. 2021.
25. V. A. Traag, L. Waltman, and N. J. van Eck, "From louvain to leiden: guaranteeing well-connected communities," *Scientific Reports*, vol. 9, Mar. 2019.
26. L. Chávez-Galán, M. C. Arenas-Del Angel, E. Zenteno, R. Chávez, and R. Lascurain, "Cell death mechanisms induced by cytotoxic lymphocytes," *Cellular amp; Molecular Immunology*, vol. 6, p. 15–25, Feb. 2009.
27. M. Westerterp, P. Fotakis, M. Ouimet, A. E. Bochem, H. Zhang, M. M. Molusky, W. Wang, S. Abramowicz, S. la Bastide-van Gemert, N. Wang, C. L. Welch, M. P. Reilly, E. S. Stroes, K. J. Moore, and A. R. Tall, "Cholesterol efflux pathways suppress inflammasome activation, netosis, and atherogenesis," *Circulation*, vol. 138, p. 898–912, Aug. 2018.
28. S. J. Galli, N. Borregaard, and T. A. Wynn, "Phenotypic and functional plasticity of cells of innate immunity: macrophages, mast cells and neutrophils," *Nature Immunology*, vol. 12, p. 1035–1044, Oct. 2011.
29. W. Saelens, R. Cannoodt, H. Todorov, and Y. Saeys, "A comparison of single-cell trajectory inference methods," *Nature Biotechnology*, vol. 37, p. 547–554, Apr. 2019.
30. Y. Hao, T. Stuart, M. H. Kowalski, S. Choudhary, P. Hoffman, A. Hartman, A. Srivastava, G. Molla, S. Madad, C. Fernandez-Granda, and R. Satija, "Dictionary learning for integrative, multimodal and scalable single-cell analysis," *Nature Biotechnology*, 2023.
31. G. Csárdi, T. Nepusz, V. Traag, S. Horvát, F. Zanini, D. Noom, and K. Müller, *igraph: Network Analysis and Visualization in R*, 2024. R package version 2.0.3.9049.
32. B. Ewing, "leidenbase: R and c/c++ wrappers to run the leiden find_partition() function," Feb. 2022.
33. L. McInnes, J. Healy, N. Saul, and L. Grossberger, "Umap: Uniform manifold approximation and projection," *The Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.
34. W. Chang, J. Cheng, J. Allaire, C. Sievert, B. Schloerke, Y. Xie, J. Allen, J. McPherson, A. Dipert, and B. Borges, *shiny: Web Application Framework for R*, 2024. R package version 1.9.1.9000, https://github.com/rstudio/shiny.
35. T. Ching, "qs2: Efficient serialization of r objects," Sept. 2024.
36. Bernd Fischer, Gregoire Pau, Mike Smith, "rhdf5," 2017.
37. H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016.
38. Z. Gu, R. Eils, and M. Schlesner, "Complex heatmaps reveal patterns and correlations in multidimensional genomic data," *Bioinformatics*, vol. 32, p. 2847–2849, May 2016.
39. Y. Xie, J. Cheng, and X. Tan, "Dt: A wrapper of the javascript library "datatables"," June 2015.
40. 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.
41. D. Risso, F. Perraudeau, S. Gribkova, S. Dudoit, and J.-P. Vert, "A general and flexible method for signal extraction from single-cell rna-seq data," *Nature Communications*, vol. 9, Jan. 2018.