Genome Biology

RESEARCH

Open Access

Integrating temporal single-cell gene expression modalities for trajectory inference and disease prediction

Jolene S. Ranek[1,2], Natalie Stanley[2,3*] and Jeremy E. Purvis[1,2*]

*Correspondence: natalies@cs.unc.edu; purvisj@email.unc.edu

[1] Department of Genetics, University of North Carolina at Chapel Hill, Chapel Hill, USA
[2] Computational Medicine Program, University of North Carolina at Chapel Hill, Chapel Hill, USA
[3] Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, USA

Abstract

Background: Current methods for analyzing single-cell datasets have relied primarily on static gene expression measurements to characterize the molecular state of individual cells. However, capturing temporal changes in cell state is crucial for the interpretation of dynamic phenotypes such as the cell cycle, development, or disease progression. RNA velocity infers the direction and speed of transcriptional changes in individual cells, yet it is unclear how these temporal gene expression modalities may be leveraged for predictive modeling of cellular dynamics.

Results: Here, we present the first task-oriented benchmarking study that investigates integration of temporal sequencing modalities for dynamic cell state prediction. We benchmark ten integration approaches on ten datasets spanning different biological contexts, sequencing technologies, and species. We find that integrated data more accurately infers biological trajectories and achieves increased performance on classifying cells according to perturbation and disease states. Furthermore, we show that simple concatenation of spliced and unspliced molecules performs consistently well on classification tasks and can be used over more memory intensive and computationally expensive methods.

Conclusions: This work illustrates how integrated temporal gene expression modalities may be leveraged for predicting cellular trajectories and sample-associated perturbation and disease phenotypes. Additionally, this study provides users with practical recommendations for task-specific integration of single-cell gene expression modalities.

Keywords: Multi-omics, Data integration, RNA velocity, Trajectory inference, Single-cell prediction

Background

Single-cell RNA sequencing (scRNA-seq) technologies have enabled the functional characterization of cellular states associated with dynamic biological processes such as development [1–3] and disease progression [4–7]. While transcriptomic information holds

great promise for gaining insight into the biological mechanisms that govern phenotypic changes, inference has been traditionally limited to incompletely sampled static mature mRNA measurements. This poses two fundamental challenges for robust prediction of the dynamic progression of cell state. First, many gene regulatory mechanisms can give rise to the same distribution of mature mRNA measurements [8]. Second, snapshot data often fails to fully capture the large biological variability required for population-level inference by missing important transition states or rare cell populations [9–11].

More recently, computational tools such as RNA velocity have been used to extract directed dynamic information from single cells [12–16]. By leveraging unspliced pre-mRNA and spliced mature mRNA molecules in a kinetic model, RNA velocity can predict the future transcriptional state of a cell. Indeed, RNA velocity has been successfully incorporated into algorithms for inferring fate probabilities [17], gene regulatory networks [18], differentiation trajectories [19–21], and embeddings [22]. However, it is still unclear whether integrating spliced gene expression with either unspliced molecules or RNA velocity predictions is useful for predictive modeling at the data-level. Such an integrated approach may help uncover salient features predictive of cell type or response to therapy, enhance our understanding of the relationship between cell states, or provide insight into the molecular pathways that drive a cell's transition to a more pathological phenotype.

Single-cell multi-omics data integration methods have had great success in fusing different molecular data types, or modalities for disease subtyping, predicting biomarkers, or uncovering cross-modality correlations [23, 24]. Here, integration methods aim to merge individual layers of single-cell data (e.g., transcriptomic, proteomic, epigenomic) into a unified consensus representation, such as an integrated graph [25] or a joint-embedding [24, 26]. To do so, computational approaches have leveraged techniques, including kernel learning [27, 28], matrix factorization [29–33], or deep learning [34]. Moreover, downstream analysis of integrated multi-omics data has provided fundamental insights into the molecular mechanisms underlying complex biological processes, including disease heterogeneity and pathological development [35].

Motivated by identifying a new more biologically meaningful set of features underlying cellular dynamics, we investigate integration of gene expression modalities at three distinct temporal stages of gene regulation: unspliced, spliced, and RNA velocity. We benchmark ten integration approaches on ten biological datasets with applications ranging from cellular differentiation to disease progression. We show that unspliced and spliced integration improves predictive performance when inferring biological trajectories, perturbation conditions, and disease states. This work illustrates how integrated temporal gene expression modalities may be leveraged for predictive modeling of cellular dynamics.

Results

We compared ten integration approaches for recovering discrete and continuous variation in cell and disease states. In the sections that follow, we will describe the integration results in more detail. We will begin by giving an introduction of the datasets used in this study. Next, we will provide details about the benchmarking design, including the integration methods considered and the evaluation criteria for each

prediction task. We will then demonstrate how an integrative analysis can be used to obtain increased biological insight over spliced expression alone. Ultimately, we will end with practical recommendations for task-specific integration.

Description of datasets

We tested integration method performance on inferring biological trajectories or classifying cells according to perturbation condition or disease status across ten publicly available single-cell RNA sequencing datasets (see the "Methods" section, Additional file 1: Table S1). Datasets were grouped into three general categories according to the prediction task. Here, we briefly introduce the datasets used in this study.

Datasets for trajectory inference (TI)

We evaluated inference of biological trajectories using four single-cell RNA sequencing datasets representing the cell cycle and stem cell differentiation, denoted as mES cell cycle, hematopoiesis (Nestorowa), NKT cell differentiation, and hematopoiesis (Olsson), respectively. To assess inference of the cell cycle, we considered a mouse embryonic stem cell cycle dataset [36], where embryonic stem cells were collected along three stages of the cell cycle (G1, S, G2/M). Cell cycle phase was manually annotated a priori based on flow sorting cells according to the Hoeschst 33342 stained distribution. The authors of the original study used this dataset to assess the proportion of cell-cell heterogeneity that arises from cell cycle variation. To assess inference of complex differentiation trajectories, we considered three datasets spanning different biological systems and trajectory types. We first considered a mouse hematopoietic stem and progenitor (HPS) cell differentiation dataset from Nestorowa et al. [37]. Here, the transcriptomes of HPS cells were profiled and nine cell surface protein measurements (Additional file 1: Table S3) were used to annotate six subpopulations, including long-term hematopoeitic stem cells (LT-HSC), lymphoid multipotent progenitors (LMPP), multipotent progenitors (MPP), megakaryocyte-erythrocyte progenitors (MEP), common myeloid progenitors (CMP), and granulocyte-monocyte progenitors (GMP). Moreover, in the original study, reconstruction of the differentiation trajectory revealed dynamic gene expression patterns consistent with early lymphoid, erythroid, and granulocyte-macrophage differentiation. To assess inference of Natural Killer T (NKT) cell differentiation [38], we analyzed the transcriptomes of four NKT cell subpopulations (NKT0, NKT1, NKT2, and NKT17) manually annotated a priori based upon cell surface protein measurements (see Additional file 1: Table S3). In the original study, the authors elucidated both transcriptomic and epigenomic signatures of the differentiation and function of thymic NKT cell subsets. Lastly, we considered another mouse hematopoiesis differentiation dataset from Olsson et al. [39], where three subpopulations (lineage negative (LSK) cells, common myeloid progenitors (CMP), and granulocyte monocyte progenitor (GMP) cells) were annotated a priori according to the expression of cell surface proteins (Additional file 1: Table S3). For our analysis, cells were excluded if they did not have ground truth annotations.

Datasets for perturbation classification

To assess integration performance on classifying cells according to perturbation condition, we considered three diverse datasets with clinical relevance representing drug stimulation and treatment of cells, denoted as LPS stimulation, INFγ stimulation, and AML chemotherapy, respectively. In the LPS stimulation dataset [40], RAW 264.7 macrophage-like cells were treated with time course of lipopolysaccharide (0-min, 50-min, 150-min, 300-min LPS) to induce NF-κB expression. NF-κB is a transcription factor that serves as a master regulator of inflammatory responses from macrophages and other innate immune cells [41]. The authors of this study integrated live cell imaging with single-cell RNA sequencing to demonstrate that NF-κB signaling shapes gene expression and has a functional role on cellular phenotypes. Therefore, in our experiments, we sought to classify cells according to stimulation condition (e.g., 150-min LPS). In the INFγ stimulation dataset [42], pancreatic islet cells from three donors were stimulated with or without Interferon-γ (INFγ) for 24 h. INFγ is a proinflammatory cytokine that has been implicated in pancreatic beta cell damage during the pathogenesis of type I diabetes [43]. Here, the authors applied their method MELD to characterize INFγ treatment response across pancreatic islet cell populations and identified a non-responsive subpopulation of beta cells characterized by high expression of insulin. Consequently, we sought to classify INFγ stimulated from unstimulated cells. Lastly, the AML chemotherapy dataset [5] consisted of peripheral blood mononuclear cells (PBMCs) collected from a patient with acute myeloid leukemia (AML) at baseline or after 2 or 4 days of treatment with chemotherapy agents Venetoclax and Azacitidine. It is hypothesized that the persistence of leukemia stem cells (LSCs) following treatment drives disease severity, relapse, and results in worse clinical outcomes [7, 44]. Here, the authors demonstrate how chemotherapy treatment induces the depletion of LSCs through metabolic reprogramming, where oxidative phosphorylation, a critical pathway for LSC maintenance and survival, is suppressed. Thus, we sought to classify PBMCs according to treatment condition (day 0, day 2, day 4).

Datasets for disease status classification

To assess integration performance on classifying cells according to disease status, we considered three case/control datasets of two disease systems, acute myeloid leukemia (AML) and multiple sclerosis (MS). In the first dataset [7], leukemia stem cells (LSCs) were collected from AML patients at treatment-naive diagnosis ($N = 5$) and following relapse after chemotherapy treatment ($N = 5$). Here, the authors compared diagnosis from relapse samples to characterize gene expression heterogeneity during AML disease progression and show that differences were largely due to metabolic reprogramming, apoptotic signaling, and chemokine signaling. Therefore, in our experiments, we sought to classify diagnosis from relapse cells. For the second and third study, we considered a multiple sclerosis dataset [6], where PBMCs and cerebral spinal fluid (CSF) were collected from MS patients ($N = 5$) and controls ($N = 5$). MS is a chronic inflammatory disorder of the central nervous system that results in neurological dysfunction [45]. When examining the transcriptional profiles of MS patient cells as compared to controls, CSF exhibited differences in cell type composition, including an enrichment

of myeloid dendritic cells and the expansion of CD4+ cytotoxic T cells and late stage B cells. In contrast, PBMCs exhibited increased transcriptional diversity with an increased proportion of differentially expressed genes. Consequently, we sought to classify control from MS cells across patients using either CSF or PBMC biological samples.

Selection of integration methods

The power of multi-omics data integration methods lies in their ability to combine individual layers of data (e.g., spliced expression, RNA velocity) to identify a new set of cellular features that more holistically represents cell type or functional state [23, 46]. Once computed, these features can be used in machine learning models to jointly analyze cell type-specific differences or to obtain clinically meaningful predictions that can inform therapeutics [47, 48]. In this study, our goal is to compare integration approaches for merging gene expression data matrices across the same set of profiled cells in order to evaluate their performance on downstream analysis tasks, including trajectory inference or sample-associated classification of cells. Given the large variety of different integration approaches, we performed a systematic evaluation of ten integration methods by selecting and grouping approaches according to two categories: early integration approaches and intermediate integration approaches. First, we consider early integration approaches as baseline computational strategies for merging individual modalities into one input matrix. Here, we selected three representative baseline strategies (cell-wise concatentation, cell-wise sum, CellRank [17]), in addition to an unintegrated control. In contrast, we consider intermediate integration approaches as computational strategies that transform individual modalities into an intermediate representation prior to merging, such as a cell similarity graph or a subspace. Within this category, we selected six representative methods, including Seurat v4 [49], Multi-Omics Factor Analysis v2 (MOFA+) [30], similarity network fusion (SNF) [25], Grassmann joint embedding [26], integrated diffusion [24], and Patient Response Estimation Corrected by Interpolation of Subspace Embeddings (PRECISE) [50]. Here, we briefly define the ten integration approaches evaluated in this study. For more details on the overall problem formulation and integration method implementation, see the integration section in the "Methods" section.

1. *Unintegrated:* A representation consisting of one data modality. In this case, our unintegrated data consists of mature spliced expression counts, as this is what is traditionally used for downstream single-cell analysis, as outlined by current best practices [51].
2. *Concatenation:* Modalities are merged through cell-wise concatenation of data matrices. This baseline approach, along with element-wise sum, is a common fusion strategy for merging multi-modal data in deep learning [46, 52, 53].
3. *Sum:* Modalities are merged through cell-wise summing data matrices.
4. *CellRank:* CellRank [17] merges data modalities by computing a weighted sum of gene expression similarity and RNA velocity transition matrices. We refer to this approach as an early integration strategy as it simply reweights the edges of the original gene expression cell similarity graph according to RNA velocity transition probabilities. Notably, this method is specific to integrating RNA velocity data.

5 *Seurat v4:* Seurat v4 [49] merges data through a weighted nearest neighbor graph approach. First, individual *k*-nearest neighbor graphs are constructed for each modality. Next, cell-specific modality weights are learned by computing within and cross modality predictions according a cell's local neighborhood. Lastly, an integrated *k*-nearest neighborhood graph is constructed according to a similarity metric defined by a weighted average of modality affinities.

6 *Multi-Omics Factor Analysis v2 (MOFA+):* MOFA+ [30] merges data modalities through a statistical matrix factorization approach formulated in a probabilistic Bayesian setting that leverages a stochastic variational inference framework for enforcing sparsity. In particular, MOFA+ decomposes each modality into a product of a shared factor matrix (shared latent space that captures the global variation in the data) as well as a weight matrix for each modality (captures individual feature contribution).

7 *Similarity network fusion (SNF):* SNF [25] merges data by first computing a cell affinity graph for each data type. Next, individual modality networks are merged through nonlinear diffusion iterations to obtain a fused network.

8 *Grassmann joint embedding:* Grassmann joint embedding [26] integrates data modalities by first computing a cell affinity graph for each data modality and then merges networks through subspace analysis on a Grassmann manifold.

9 *Integrated diffusion:* Integrated diffusion [24] merges data modalities by first computing a diffusion operator for each denoised data type. Next, individual operators are merged by computing a joint diffusion operator.

10 *Patient Response Estimation Corrected by Interpolation of Subspace Embeddings (PRECISE):* PRECISE merges data by first performing principal components analysis (PCA) on each individual modality. Next, principal components are geometrically aligned and consensus features are determined through interpolation. For this analysis, we implement two versions by projecting spliced expression onto (1) the principal vectors (denoted as PRECISE) or (2) the consensus features (denoted as PRECISE consensus).

Benchmarking overview

Given that gene expression modalities are collected along a temporal axis of gene regulation, we evaluated the performance of integrating unspliced, spliced, or RNA velocity modalities on predicting discrete and continuous variation in cell and disease states across a range of biological scenarios (Additional file 1: Table S1). Following transcriptomic profiling, spliced and unspliced counts were preprocessed and jointly batch effect-corrected prior to RNA velocity estimation (see the "Methods" section, Additional file 1: Table S1, Figs. S1-S10). For each set of modalities (spliced and unspliced counts, moments of spliced and RNA velocity), our goal is to identify a consensus representation that we can use as input to a predictive model (Fig. 1A). We benchmarked ten integration approaches for combining these gene expression modalities by evaluating how well integrated features infer biological trajectories, classify a cell's response to a drug perturbation, or classify the disease status of a cell. Moreover, to quantify the predictive performance of an integration strategy, we computed several metrics for each prediction

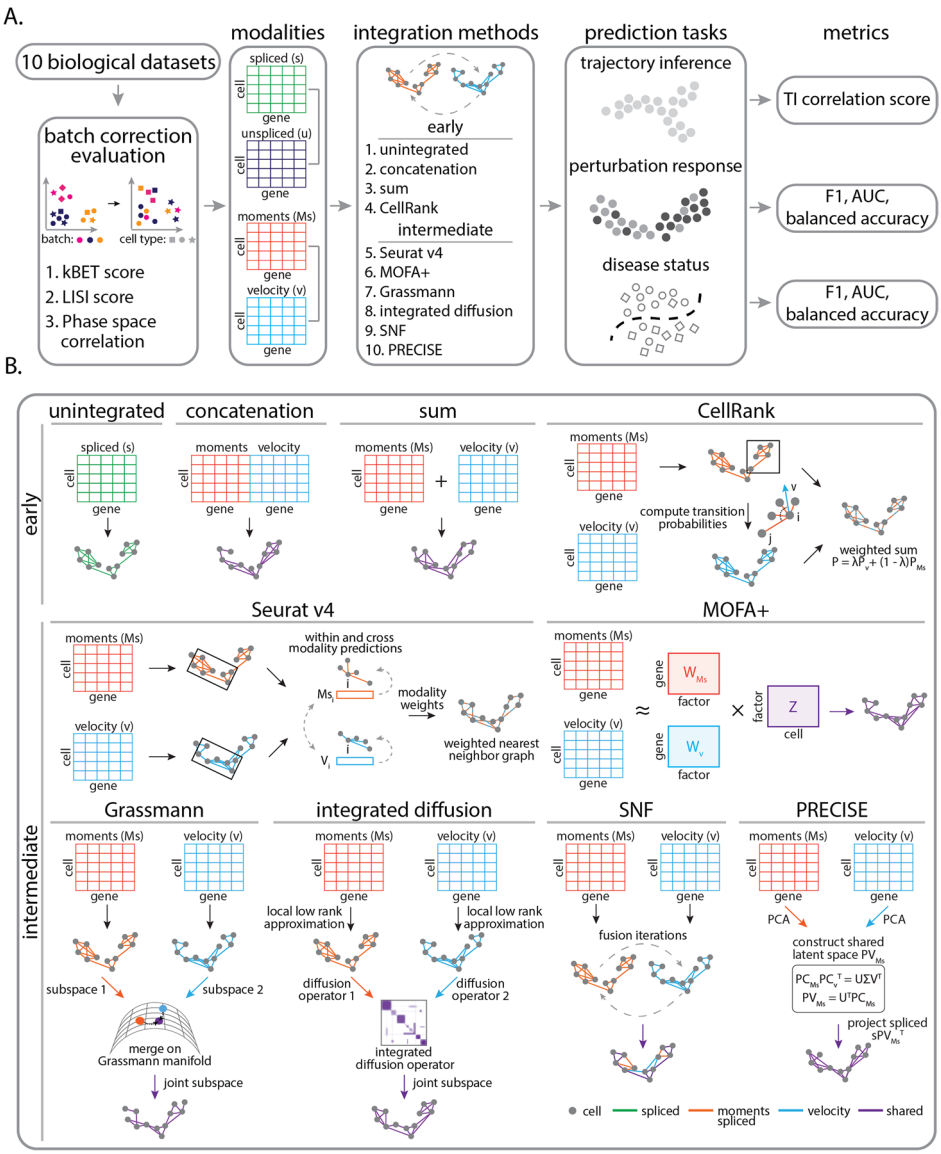

Fig. 1 Schematic overview of benchmarking design. **A** Workflow of integration method evaluation. Ten integration approaches and four temporal mRNA modalities are evaluated on three prediction tasks. Data are first preprocessed and jointly batch effect corrected. Next cross-modality integration (spliced and unspliced counts or moments of spliced and RNA velocity) is performed using ten different integration approaches. Features specified through the integration strategy are used to infer trajectories, predict response to drug treatment, and classify patient cells. **B** Overview of data integration strategies (unintegrated, concatenation, sum, CellRank [17], Seurat v4 [49], MOFA+ [30], Grassmann joint embedding [26], integrated diffusion [24], SNF [25], and PRECISE [50])

task. To assess the quality of trajectory inference prediction following integration, we computed a trajectory inference correlation score to a ground truth reference that takes into account cellular positioning and trajectory-specific dynamically expressed genes. To assess classification performance, we computed the accuracy of predicted labels from an integration strategy using three complementary metrics, such as F1 score, balanced accuracy, and area under the receiver operator curve. For integration methods that required user-specified input parameters (Additional file 1: Table S2), we performed

hyperparameter tuning to select the best performers. We then ranked the overall predictive performance of integration strategies for each task by averaging scores across all datasets (see the "Methods" section). This measures how well incorporating dynamic mRNA information aids in recovering intermediate transitions or classifying the state of a cell.

In selecting an appropriate data integration strategy, it is crucial that the approach is able to satisfy computational challenges that are specific to each modality. First, a method must be robust to varying amounts of sparsity between data types. Single-cell RNA sequencing modalities produce matrices which contain a large proportion of zeros, where only a small fraction of total transcripts are detected due to capture inefficiency, amplification noise, and stochasticity [54]. This sparsity is far greater in unspliced molecules due to polyadenylation enrichment in library preparation [12]. Moreover, given that unspliced, spliced, and RNA velocity predictions are influenced by biological and technical noise, a method must be able to resolve noisy signals for robust prediction. To address these challenges, we compared two classes of integration approaches for combining temporal sequencing modalities, including early integration approaches (concatenation, sum, CellRank) and intermediate integration approaches (Seurat v4, MOFA+, Grassmann joint embedding, integrated diffusion, SNF, PRECISE) (see the "Selection of integration methods" and "Methods" sections, Fig. 1B).

Integration performance on inference of biological trajectories

When undergoing dynamic processes such as differentiation, cells exhibit a continuum of cell states with fate transitions marked by external stimuli, cell-cell interactions, and stochastic gene expression [55]. One limitation of trajectory inference (TI) reconstruction from snapshot single-cell data is the fact that many gene regulatory mechanisms and cellular dynamics could give rise to the same distribution of cell states [8]. We reasoned that incorporation of unspliced counts or RNA velocity data may provide increased granularity of the state space to more accurately recapitulate the underlying trajectory. To test this hypothesis, we evaluated integration method performance on inferring two types of biological trajectories, cell cycle and differentiation, by measuring their ability to (1) recover known cell population transitions and (2) infer lineage-specific dynamically expressed genes.

In order to construct reference trajectories for evaluation, we chose well-studied biological systems and selected datasets that had gold standard cell type annotations according to the expression of particular characteristic phenotypic markers. Therefore, we selected datasets consisting of mouse embryonic stem cell cycle, mouse hematopoietic stem and progenitor cell differentiation (Nestorowa), NKT cell differentiation, and mouse hematopoiesis (Olsson) differentiation trajectories (see the "Description of datasets" and "Methods" sections). We then quantified how well integrated features recapitulated cell cycle or differentiation trajectories by adapting an approach previously used to assess the accuracy of trajectory inference methods [56] (see the "Methods" section). Briefly, we constructed predicted trajectories for each integration approach by applying partition-based graph abstraction (PAGA) [57], a state-of-the-art trajectory inference method, on the joint graph following integration. First, PAGA was used on the integrated *k*-nearest neighbor graph to determine directed weighted edges between known

cell types according to FACS annotations. Here, the edge weights quantify the strength in connectivity between cell populations, which represents the overall confidence of a cell population transition. Next, we applied diffusion pseudotime [58] to determine an individual cell's progression through those high-confidence paths. Since integrated features are used as input, the inferred trajectory now contains transcriptional information from a transitional process at or following a measured time point. To assess the accuracy of predicted trajectories, we defined a trajectory inference correlation score that compares predicted trajectories to a ground truth reference trajectory we curated from the literature (see the "Methods" section). By taking into account a cell's position along the trajectory, as well as the features that are dynamically expressed, this correlation metric reflects how well integration infers known cellular dynamics. Moreover, to ensure a robust comparison across integration approaches, we generated predicted trajectories and correlation scores with respect to the same ten random root cells selected from the annotated root cluster (mouse embryonic cell cycle: G1, mouse hematopoiesis (Nestorowa): long-term hematopoietic stem cells (LT-HSC), NKT cell differentiation: NKT0, mouse hematopoiesis (Olsson): lineage negative cells (LSK)).

When comparing predicted trajectories across integration approaches, we found spliced and unspliced as well as moments of spliced and RNA velocity integrated features led to a higher trajectory inference correlation score when compared to unintegrated data (Fig. 2A). Across all four datasets, integration with spliced and unspliced counts obtained median TI correlation scores (best performing: 0.849, 0.792, 0.712, 0.953), as compared to unintegrated data (0.750, 0.579, 0.472, 0.739), for mES cell cycle, hematopoiesis (Nestorowa), NKT cell differentiation, and hematopoiesis (Olsson) datasets, respectively. In contrast, for RNA velocity integration, the median trajectory inference correlation scores were (0.856, 0.787, 0.672, 0.866) for mES cell cycle, hematopoiesis (Nestorowa), NKT cell differentiation, and hematopoiesis (Olsson) datasets respectively. We next investigated how incorporating temporal gene expression modalities alter the inferred PAGA trajectories and diffusion map embeddings for the top integration performers with respect to unintegrated data (Fig. 2B). When examining the PAGA graphs, we found that all predicted trajectories captured the major cell state transitions supported by the literature. For example, mouse embryonic cell cycle predicted trajectories included the cyclical transition through the proliferative phases of the cell cycle [36]. Moreover, for mouse hematopoiesis (Nestorowa), predicted trajectories inferred known developmental lineages, with cells transitioning from the multipotent progenitor (MPP) population to early lymphoid (LMPP), erythroid (MEP), and granulocyte-macrophage (GMP) cell populations [37, 59]. In addition to capturing known transitions, predicted

(See figure on next page.)

Fig. 2 Integration improves inference of cell cycle and differentiation trajectories. Trajectory inference was performed with partition-based graph abstraction to assess the quality of inferred embryonic cell cycle, hematopoiesis differentiation (Nestorowa), NKT cell differentiation, and hematopoiesis differentiation (Olsson) trajectories from (**A** top panel) spliced and unspliced or (**A** bottom panel) moments of spliced and RNA velocity integrated features generated from ten integration methods. The boxplots represent trajectory inference correlation scores (TI_{corr}) for ten random root cells. Asterisk indicates the method with the highest median TI_{corr} score. **B** PAGA predicted trajectories and diffusion map embeddings representing the inferred biological trajectory for unintegrated data, as well as high ranking performers for unspliced and RNA velocity integration

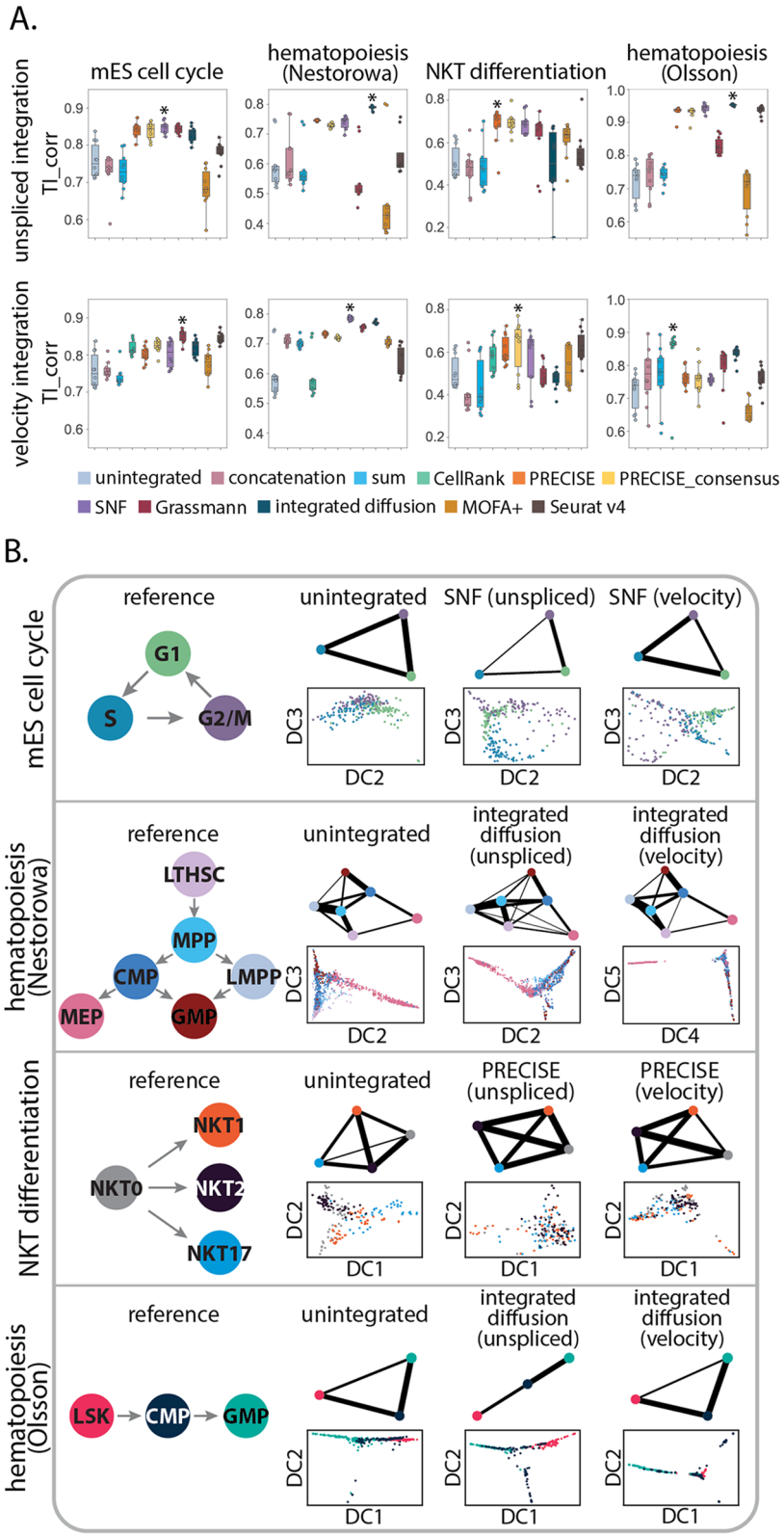

Fig. 2 (See legend on previous page.)

trajectories with integrated data resulted in an improved recovery of cellular dynamics. For example, integration of spliced and unspliced counts with SNF better resolves the smooth cyclical progression through the embryonic cell cycle, with cells following a clear trajectory from G1 to S to G2/M (Fig. 2B). Moreover, by comparing the change in PAGA connectivity across the same integration strategy for different input modalities (Fig. 2B), we observe how temporal gene expression modalities influences the confidence of an inferred cell state transition. When integrating unspliced and spliced features for cell cycle inference, we observe an increase in PAGA connectivity from G2/M to G1 to S phases, whereas RNA velocity integration illustrates the next time point and provides stronger transition weights from G1 to S to G2/M. This added layer of granularity demonstrates prioritized cell type transitions with respect to the underlying gene expression dynamics, which may provide additional insight into the gene regulatory programs that drive specific paths of temporal variation.

As a secondary approach, we performed trajectory inference on the integrated embedding from an integration strategy using Slingshot [60] (see the "Methods" section). Similar to the trajectory inference results with PAGA, we found unspliced and spliced integration led to a higher trajectory inference correlation score for mES cell cycle, NKT cell differentiation, and hematopoiesis differentiation (Olsson) datasets (Additional file 1: Fig. S11A). In contrast, integration with RNA velocity generally led to a similar or modest increase in median trajectory inference correlation scores when compared to unintegrated data (Additional file 1: Fig. S11B). Lastly, by aggregating trajectory inference correlation scores across datasets, we find Grassmann joint embedding and and similarity network fusion amongst the best ranking methods for predicting trajectories with both sets of modalities (Additional file 1: Fig. S12). Taken together, these results indicate that integrating gene expression data improves the ability to predict temporal changes in gene expression along progressive changes in cell state.

Testing integration under perturbation conditions

A key application of scRNA sequencing is the ability to identify subpopulations of cells that are either responsive or resistant to drug therapy [61]. To examine if integration of unspliced or RNA velocity data can aid in these tasks, we tested integration performance on classifying perturbation condition labels from three diverse datasets with clinical relevance, including lipopolysaccharide (LPS) stimulated macrophage-like cells, Interferonγ (INFγ) stimulated pancreatic islet cells, and peripheral blood mononuclear cells (PBMCs) collected from a patient with acute myeloid leukemia (AML) after chemotherapy treatment (see the "Description of datasets" section). Using these perturbation datasets, we constructed a set of integrated features corresponding to a cell's transcriptional response following a perturbation. We then considered the problem of cell state classification, where our goal is to learn the annotated condition labels (e.g., INFγ stimulated or unstimulated) from the underlying feature set. We labeled or classified cells using label propagation [62] (see the "Methods" section) and compared predictions to the ground truth labels using three complementary accuracy metrics, including area under the receiver operator curve (AUC), F1 score, and balanced accuracy (acc_b). Across all three datasets, we found that integration of spliced and unspliced counts led to higher classification accuracy than unintegrated data (Fig. 3A), with median AUCs (best performing

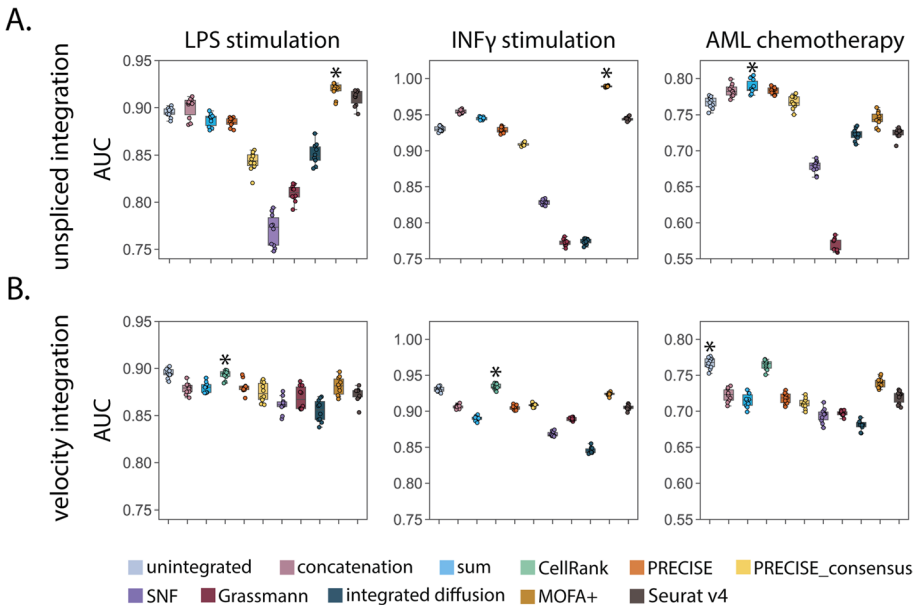

Fig. 3 Integrating spliced and unspliced counts improves drug treatment condition prediction. Label propagation was used to classify cells according to treatment response from (**A**) spliced and unspliced or (**B**) moments of spliced and RNA velocity integrated features generated from ten integration approaches. The boxplots represent classification accuracy according to area under the receiver operator curve (AUC) and the asterisk represents the method with the highest median AUC. Across all three datasets, spliced and unspliced integration achieves increased classification accuracy over unintegrated data

integrated: 0.921, 0.989, 0.785; unintegrated: 0.895, 0.930, 0.768) for LPS, INFγ, and AML chemotherapy datasets, respectively. In contrast, we found that RNA velocity integration generally led to worse classification accuracy than unintegrated data (Fig. 3B). One notable exception was integration performed with CellRank, which resulted in a similar performance to unintegrated data, with median AUCs (CellRank: 0.895, 0.934, 0.766, unintegrated: 0.895, 0.930, 0.768). Similar results were obtained for additional metrics, such as F1 score and balanced accuracy (Additional file 1: Fig. S13). As a secondary validation, we trained a support vector machine (SVM) classifier to learn perturbation labels from the shared lower dimensional space following integration. We performed nested 10-fold cross validation to obtain a distribution of predictions for each method and dataset (see the "Methods" section). We observed similar classification results with unspliced integration outperforming unintegrated data (Additional file 1: Fig. S14).

To rank methods according to how accurately they can predict a cell's perturbation, we computed aggregate scores by taking the mean of individual method scores across datasets (see the "Methods" section). Overall, we found that early integration strategies (concatenation, sum, CellRank) as well as MOFA+ tended to outperform intermediate embedding-based approaches (SNF, Grassmann joint embedding, integrated diffusion, Seurat v4) (Additional file 1: Fig. S15). The best performing method for unspliced integration was MOFA+ (Additional file 1: Fig. S15A), whereas the best performing method for RNA velocity integration was CellRank (Additional file 1: Fig. S15B). Overall, these results suggest that a straightforward integration of spliced and unspliced counts with concatenation or integration using matrix factorization approaches like MOFA+ may provide the best strategy to most accurately predict a cell's associated perturbation.

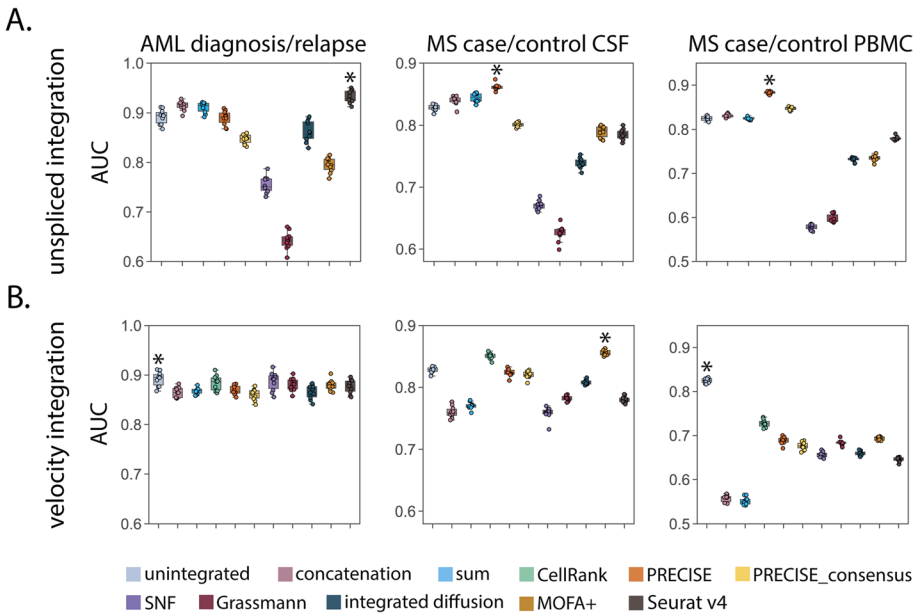

Fig. 4 Integrating spliced and unspliced counts improves disease state classification. Label propagation was used to classify cells according to patient disease status from (**A**) spliced and unspliced or (**B**) moments of spliced and RNA velocity integrated features generated from ten integration approaches. The boxplots represent classification accuracy according to area under the receiver operator curve (AUC) and the asterisk represents the method with the highest median AUC. Across all three datasets, spliced and unspliced integration achieves increased classification accuracy over unintegrated data

Furthermore, these results illustrate how an integrated analysis of gene expression modalities may provide the granularity necessary for better identifying cells that are strongly associated with a particular treatment condition, which may provide insights into the biological mechanisms conferring a phenotypic response.

Spliced and unspliced integration improves disease state classification

We next asked if an integrative analysis of unspliced or RNA velocity data can help distinguish discrete disease cell states. In particular, we aimed to evaluate integration performance on predicting whether or not cells were from control or disease patients using three datasets, including an acute myeloid leukemia (AML) diagnosis/relapse dataset, a multiple sclerosis (MS) case/control dataset of cerebral spinal fluid (CSF), and a MS case/control dataset of peripheral blood mononuclear cells (PBMCs) (see the "Description of datasets" section). To test whether leveraging temporal gene expression modalities can aid in this tasks, we used the same label propagation strategy; however, now formulated as a binary classification task based on the disease status labels for each cell. Similar to the perturbation results, we found that unspliced integration achieves higher classification accuracy for predicting disease status, with the median AUCs for the best performing methods (0.930, 0.861, 0.884) compared to unintegrated data (0.895, 0.828, 0.825) for AML, MS-CSF, and MS-PBMC datasets, respectively (Fig. 4A). Interestingly, we observe differences in the predictive performance of integrated modalities across biological samples (CSF, PBMCs) collected from the same cohort of patients. Overall trends for integration performance were consistent across additional metrics and

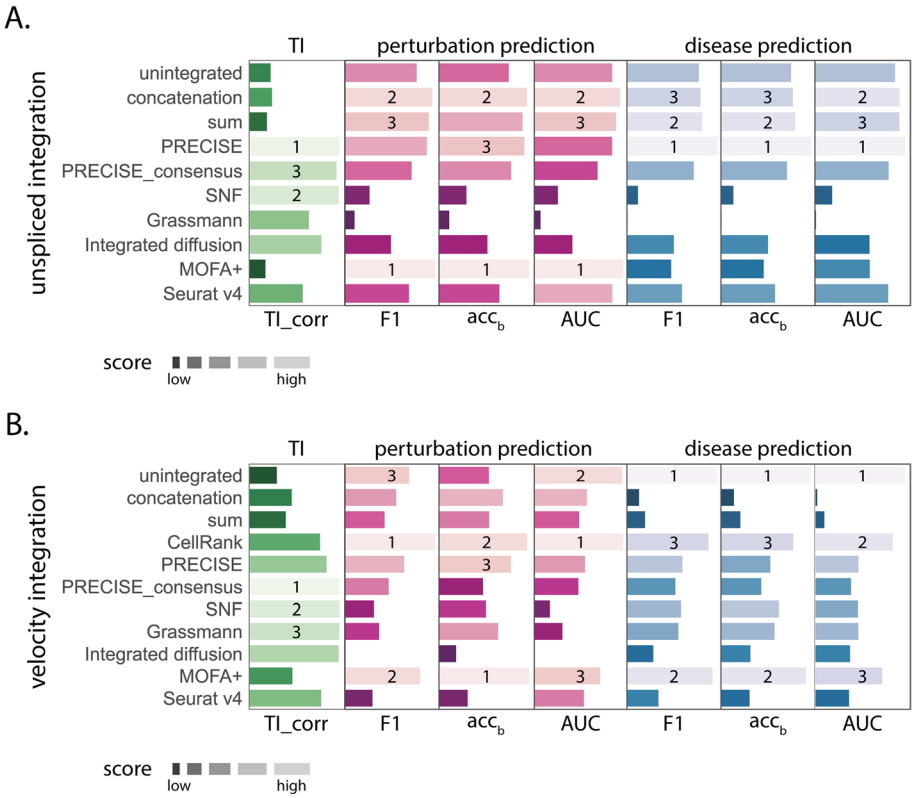

Fig. 5 Ranked integration method performance across prediction tasks. Integration methods were ranked by averaging their overall performance across datasets for each prediction task (trajectory inference with PAGA: green, perturbation classification: blue, and classification of disease status: pink). Ranked scores were computed for several metrics for evaluating a prediction task: TI_{corr}, F1 score, balanced accuracy (acc_b), and area under the receiver operator curve (AUC). Here, higher ranked method scores are indicated by a longer lighter bar. **A** Overall quality of spliced and unspliced integration performance according to several metrics for evaluating prediction tasks. **B** Overall quality of moments of spliced and RNA velocity integration performance according to several metrics for evaluating prediction tasks. Of note, CellRank was not performed on unspliced and spliced integration, as it relies on RNA velocity data. Across all three prediction tasks, unspliced integration outperforms unintegrated data, while RNA velocity integration often achieves increased trajectory inference correlation and perturbation classification scores

classifiers (Additional file 1: Figs. S14, Fig. S16). When ranking each particular method's performance on classifying the disease status of a cell across datasets, we found the best performing methods for unspliced integration to be PRECISE, sum and concatenation (Additional file 1: Fig. S17).

Overall integration method performance across datasets and tasks

Figure 5 displays the overall ranked aggregate scores for each method colored according to task (green: trajectory inference, pink: perturbation classification, blue: disease state classification). Across all three tasks, we found unspliced integration (Fig. 5A) to be more predictive of cellular state than RNA velocity integration (Fig. 5B) or no integration (unintegrated Fig. 5A, B). While integration method performance varied across datasets, experimental modalities, and tasks, some clear trends emerged. When inferring biological trajectories using PAGA, unspliced integration with PRECISE and similarity network fusion (SNF) provided the highest

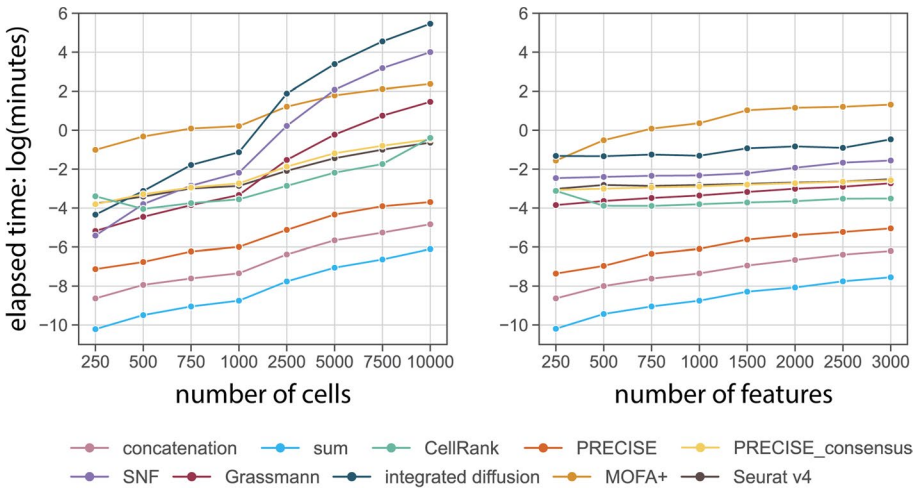

Fig. 6 Overall runtime scalability. Runtime comparison of ten integration approaches after varying the number of cells or features. Total elapsed time was averaged over five random trials and is plotted on natural log scale. All the integration strategies were run on a Linux server allocating 16 cores and 32GB of memory (Intel Xeon E5-2680 v3 processors). MOFA+ was evaluated with GPU mode enabled (1 GPU Nvidia GeForce GTX1080)

trajectory inference correlation score to the ground truth (Fig. 5A). In comparison, when evaluating perturbation or disease cell state classification, concatenation, sum, and PRECISE were among the best ranking methods across all three metrics and datasets (Fig. 5A). Collectively, these results indicate that integration method performance is task-specific, with intermediate embedding-based approaches outperforming unintegrated data on inferring biological trajectories and early baseline approaches achieving increased classification performance.

Lastly, as computational cost is an important practical consideration when selecting an integration strategy, we compared the runtime scalability of all integration methods by randomly downsampling the IFNγ dataset and varying the total number of cells or features prior performing integration (see the "Runtime scalability" section). By measuring the total elapsed time, we found the simple integration strategies including sum, concatenation, and PRECISE, to be the most efficient when varying the number of cells or features (Fig. 6). In contrast, the intermediate integration strategies such as SNF, MOFA+, and integrated diffusion required increasing amounts of compute time. These results further highlight the practical usefulness of baseline integration strategies for providing increased predictive performance, while easily scaling to large-scale datasets for the tasks outlined in this study.

Discussion

Several limitations should be considered when integrating gene expression modalities for cellular trajectory inference or disease state classification. In this study, we evaluated methods for constructing integrated graphs or joint embeddings with a priori knowledge of ground truth labels. For trajectory inference evaluation, we explored how integrated data influences the change in connectivity or inferred cell state transitions between known cell types identified via FACS. We found that integrated data

resulted in increased trajectory inference correlation with respect to a reference trajectory. However, given that the results are sensitive to choice in hyperparameters, it may be challenging to select optimal hyperparameters without a priori knowledge of cell types or expected cell type transitions. Here, a range of hyperparameters should be considered when using the intermediate integration methods outlined in this study. Of note, we observed that baseline integration approaches, such as sum and concatenation of spliced and unspliced counts perform consistently well on classifying sample-associated cell phenotypes. This is particularly useful as these approaches are less computationally expensive and do not require hyperparameter tuning. Of note, these baseline methods did not perform well when integrating moments of spliced data with RNA velocity predictions for classification.

Furthermore, the limitations of integration performance are an extension of the modalities used as input. RNA velocity is a noisy extrapolation of gene regulation that can be biased by insufficient sampling of unspliced molecules [63], relies on model assumptions that may be violated [64], and is sensitive to choice in preprocessing tools, such as the quantification of mRNA abundances [65]. Notably, the accuracy of RNA velocity estimation can be improved by incorporating both gene expression and chromatin accessibility data [16]. Moreover, there is currently no consensus on how to appropriately batch effect correct linked gene expression modalities [63]. We chose to jointly correct spliced and unspliced count matrices according to the three metrics and three methods outlined in this study; however, we note that this challenge may bias or limit the interpretation of our results. We anticipate improved performance as bioinformatics tools are developed to better analyze such data. Lastly, although RNA velocity often did not result in an increase in classification accuracy for the datasets selected in this study, this does not preclude it from being informative for the analysis of other datasets. RNA velocity captures gene expression dynamics over the timescale of hours and thus may provide crucial information for longitudinal datasets with finer temporal sampling.

In this study, we showed how integrated features can be used to build a predictive model for trajectory inference or classification tasks when ground truth annotations are provided. However, future work could focus on evaluating temporal gene expression integration for its ability to gain increased biological insight according to a wider range of tasks, such as unsupervised cell population identification [66], characterizing phenotypic-related cells [42], characterizing differentially abundant cell populations [67, 68], or gene regulatory network inference [69]. As temporal gene expression modalities provide a window into a cell's regulatory response following a drug treatment, integrating modalities may provide additional signal required for refining cell state or for identifying sources of variation and gene signatures that are specific to a cell's immediate transcriptional response. Moreover, this work could also be extended to the analysis of other extrapolated regulatory modalities, including RNA velocity in-situ [14], protein velocity [15], or chromatin velocity [70].

Conclusions

In this study, we investigated integration of unspliced, spliced, and RNA velocity gene expression modalities for resolving discrete and continuous variation in cell and disease states. We found that integrating modalities along a temporal axis of gene regulation provides additional information necessary for robustly predicting cellular trajectories during differentiation and cell cycle. Additionally, we show how spliced and unspliced integrated features can be used to better classify cells according to sample-associated phenotypes acquired after an experimental perturbation or within a disease state. Lastly, by benchmarking ten data integration methods on the aforementioned prediction tasks, we elucidate method performance specific to gene expression modalities or tasks. While intermediate integration approaches such as Seurat v4, SNF, Grassmann joint embedding, integrated diffusion, and PRECISE facilitate increased performance on inferring biological trajectories, simple integration of spliced and unspliced counts through concatenation, sum, or PRECISE achieves increased trajectory inference correlation scores, perturbation classification accuracy, and disease state classification accuracy across most datasets. To this end, integrating multiple gene expression modalities profiled from the same set of cells provides a finer resolution of the transcriptional landscape of development or disease. Thus, an integrated analysis of gene expression modalities may be crucial for the interpretation of dynamic phenotypes.

Methods

Datasets

We evaluated trajectory inference, experimental perturbation, and disease classification performance on ten datasets spanning various biological contexts. For more details on data preprocessing, see Additional file 1: Table S1.

Hematopoiesis differentiation (Nestorowa)

FASTQ files consisting of hematopoietic stem and progenitor cells were accessed from Nestorowa et al. [37] with the accession code GSE81682. FACS labels from broad gating were used to annotate six cell populations along three differentiation lineages: long-term hematopoietic stem cells (LT-HSC), lymphoid multipotent progenitors (LMPP), multipotent progenitors (MPP), megakaryocyte-erythrocyte progenitors (MEP), common myeloid progenitors (CMP), and granulocyte-monocyte progenitors (GMP) (see Additional file 1: Table S3). Individual cell FASTQ files were aligned to the mouse reference genome mm10 with the STAR v2.7.7 aligner. A loom file containing spliced and unspliced molecular counts was obtained using Velocyto v0.17.

Mouse embryonic cell cycle

A dataset of mouse embryonic stem cells undergoing different stages of the cell cycle was accessed from Buettner et al. [36] with the accession code E-MTAB-2805. FACS cell cycle labels from Hoesct flow sorting were used to annotate cells along three phases: G1, S, and G2/M. Individual cell FASTQ files were aligned to the mouse reference genome mm10 with the STAR v2.7.7 aligner. A loom file containing spliced and unspliced molecular counts was subsequently generated with Velocyto v0.17.

NKT cell differentiation

FASTQ files consisting of natural killer T (NKT) cell differentiation was accessed from Engel et al. [38] with the accession code GSE74596. FACS labels were used to annotate the four NKT cell subpopulations: NKT0, NKT1, NKT2, and NKT17 (see Additional file 1: Table S3). Files were aligned to the mouse reference genome mm10 with the STAR v2.7.7 aligner prior to generating a loom file containing spliced and unspliced counts using Velocyto v0.17.

Hematopoiesis differentiation (Olsson)

FASTQ files of mouse hematopoiesis was accessed from Olsson et al. [39] with the accession codes GSE70236, GSE70240, and GSE70244. FACS labels were used to annotate the three subpopulations: lineage negative (LSK) cells, common myeloid progenitors (CMP), and granulocyte monocyte progenitor (GMP) cells (see Additional file 1: Table S3). Individual cell FASTQ files were aligned to the mouse reference genome mm10 with the STAR v2.7.7 aligner prior to generating a loom file containing spliced and unspliced counts using Velocyto v0.17.

LPS stimulation

FASTQ files were accessed from Lane et al. [40] with the accession code GSE94383. Here, a macrophage-like cell line RAW 264.7 was stimulated with lipopolysaccharide (LPS) over 4 time points: 0-min unstimulated, 75-min, 150-min, 300-min post LPS stimulation. Files were aligned to the mouse reference genome mm10 with the STAR v2.7.7 aligner. A loom file containing spliced an unspliced molecular counts was generated with Velocyto v0.17. Following preprocessing, batch effect correction was performed on the libraries.

INFγ stimulation

Aligned BAM files of pancreatic islet cell INFγ stimulation were accessed from Burkhardt et al. [42] with the accession code GSE161465. This dataset consisted of three donors per stimulation condition (control, INFγ stimulated). A loom file containing spliced and unspliced molecular counts was generated for each donor and condition with Velocyto v0.17, then subsequently merged into a single file. Following preprocessing, batch effect correction was performed using the donor labels.

AML chemotherapy

To assess disease progression, aligned BAM files of an individual patient with AML undergoing chemotherapy were accessed from Pollyea et al. [5] with the accession code GSE116481. Condition labels consisted of three timepoints: d0 untreated, d2-, d4- post-Venotoclax and Azacitidine treatment. A loom file containing spliced and unspliced molecular counts for each timepoint was generated with Velocyto v0.17, then merged into a single file. Following preprocessing, batch effect correction was performed on the condition labels.

AML matched diagnosis/relapse

Raw FASTQ files were accessed from Stetson et al. [7] with the accession code GSE126068. In this dataset, PBMCs were collected from 5 patients with AML on the onset of diagnosis and following relapse. FASTQ files were aligned to the human reference genome GRCh38 with the STAR v2.7.7 aligner. A loom file containing spliced and unspliced molecular counts was obtained with Velocyto v0.17. Following preprocessing, batch effect correction was performed using the patient labels.

MS case/control

Aligned BAM files were accessed from Schafflick et al. [6] with the accession code GSE138266. Here, two biological samples were collected from each patient (CSF, PBMCs) with a disease status label (control or MS). Loom files containing spliced and unspliced molecular counts for each patient sample were obtained with Velocyto v0.17. Then, a merged loom file consisting of control and MS patient cells was generated for each sample independently. Following preprocessing, batch effect correction was performed using the patient labels.

Preprocessing

Quality control, normalization, and highly variable gene selection

All scRNA sequencing datasets were quality control filtered according to read depth and distributions of counts. Following empty droplet and doublet removal, dying cells were removed by ensuring less than 20% of total reads were mapped to mitochondrial transcripts. Genes were filtered out if they were expressed in less than five cells or had less than five counts shared between spliced and unspliced matrices. To perform normalization, we estimated size factors for filtered spliced and unspliced count matrices with Scran pooling normalization v1.20.1 [71]. For datasets with an appreciable batch effect, size factors were subsequently scaled according to median normalization of the ratio of average counts between batches with Batchelor v1.8.0; this ensures data is downsampled based upon the batch with the smallest read depth. To restrict the feature space, we selected highly variable genes on `log+1` transformed data by estimating a normalized dispersion measure [72] using the highly variable genes function in Scanpy v1.8.1 (flavor = seurat, minimum mean = 0.012, minimum dispersion = 0.25, maximum mean = 5).

Batch effect correction

RNA velocity relies on an ordinary differential equation framework to estimate the relationship between two connected modalities, spliced and unspliced mRNA counts [12, 13, 73]. As such, correcting each modality independently may lead to incorrect model fitting and spurious velocity vectors [63]. We evaluated the performance of batch effect correction methods, ComBat [74], mutual nearest neighbors (MNN) [75], and Scanorama [76] on correcting count data simultaneously. These methods were chosen as they directly correct the original gene expression data and were shown to be the top performing methods for recovering cell states [77]. Briefly, we considered two simple approaches for combining the data prior to correction (1) summed spliced and unspliced counts or (2) cell-wise concatenation. To obtain corrected count matrices for summed input data, we followed the batch effect correction approach introduced in Ref. [78],

$$\mathrm{M} = \log(\mathrm{S} + \mathrm{U} + 1) \tag{1}$$

$$\mathrm{R} = \frac{\mathrm{S}}{\mathrm{S} + \mathrm{U}} \tag{2}$$

$$\mathrm{S}_c = \exp(\mathrm{M}_c \cdot \mathrm{R} - 1) \tag{3}$$

$$\mathrm{U}_c = \exp(\mathrm{M}_c \cdot (1 - \mathrm{R}) - 1). \tag{4}$$

Here, S and U represent spliced and unspliced count matrices, respectively. Batch effect correction was performed on the summed total expression matrix, M, to yield a corrected data matrix M_c. Corrected spliced S_c and unspliced U_c counts were then obtained by inverting the log transformation through exponentiation. ComBat was run in python using Scanpy v1.8.1, MNN was run in R using Batchelor v1.8.0, and Scanorama was run in python using Scanorama v1.7.2.

Batch effect correction evaluation

To evaluate batch effect correction methods on combined spliced and unspliced modalities, we consider three metrics for assessing batch effect removal while preserving both biological variation and the unspliced to spliced relationship.

1. *k-nearest neighbor batch effect correction test (kBET):* The kBET algorithm [79] quantifies batch effects by comparing the batch label composition of local random neighborhoods to the overall global label composition through a χ^2 test. Tests are then averaged to obtain an overall rejection rate. To test for batch effects, we perform kBET using a fixed neighborhood size of $k = 10$ neighbors for each correction approach (uncorrected, MNN sum, MNN concatenation, ComBat sum, ComBat concatenation). kBET scores were computed using kBET v0.99.6.
2. *Local Inverse Simpson's Index (LISI):* The LISI score [80] measures the degree of batch label mixing by computing the number of cells that can be drawn from a local neighborhood before a batch label is observed twice. Here, local distances are weighted according to a Gaussian kernel and probabilities are determined by the inverse Simpson's index. LISI returns a diversity score ranging from 1 to the total number of batches. To test for batch label diversity, we compute LISI using a fixed perplexity of 30 for each correction approach (uncorrected, MNN sum, MNN concatenation, ComBat sum, ComBat concatenation). LISI scores were computed using harmonypy.
3. *Pearson correlation of phase space pairwise distances:* The dynamical model of RNA velocity estimates transcriptional dynamics by inferring gene-specific reaction rate and latent parameters through an expectation-maximization framework on the phase space (spliced and unspliced counts) of the data. To quantify how well a batch effect correction approach preserves the unspliced to spliced relationship across all cells, we compared phase space cellular neighborhoods by computing the Pearson correlation of pairwise distances in the phase space for each donor and pairwise distances of the same cells in corrected data. In other words, for each gene we obtain a single correlation score capturing how well cell-cell distances are preserved in the

phase space of corrected data with respect to an individual donor/patient. The distribution of gene correlations measure the overall quality of correction for retaining similar cell distributions for RNA velocity fitting and estimation.

To select a batch effect correction approach, we evaluated correction performance on the each biological condition individually. Furthermore, we took the intersection of genes that were highly variable across all profiled samples (e.g., libraries, donors, patients) to ensure that the data being compared were specific to the biological system under study and that donor-specific variation was removed. For each dataset, we selected the batch effect correction approach that had the best performance across all three metrics (see Additional file 1: Table S1, Fig. S10). One exception was the AML diagnosis/relapse dataset, which contained too few cells for the analysis. Here, we selected ComBat concatenation, as it was the approach that consistently performed well on all other datasets. Once an approach was selected, we performed joint correction on the original full dataset as outlined previously (see the "Preprocessing" section).

RNA velocity estimation

To estimate RNA velocity, we used the dynamical model implementation in Scvelo v0.2.3. More specifically, first order moments of spliced and unspliced counts were computed based on a k-nearest neighbor graph of cells ($k = 10$), constructed by calculating pairwise Euclidean distances between cells based on their first 50 principal components (PCs). The full dynamical model was then solved for all genes to obtain a high dimensional velocity vector for every cell. Given that populations of cells may have different mRNA splicing and degradation kinetics, we performed a likelihood ratio test for differential kinetics on the clusters identified from Leiden community detection (resolution parameter of 1.0) [81]. Clusters of cells that exhibited different kinetic regimes were fit independently and velocity vectors were corrected.

Sketching

To evaluate integration performance on the large-scale datasets, we first performed subsampling with geometric sketching. Geometric sketching [82] is an algorithm that aims to select a representative subset of cells that preserves the overall transcriptional heterogeneity of the full dataset. By approximating the underlying geometry of the data through a plaid covering of equal volume hypercubes, geometric sketching is able to evenly select cells such that rare cell types are sufficiently sampled. We implemented geometric sketching to select a representative subset of cells from both multiple sclerosis case/control datasets. Sketches were constructed from the transcriptional landscape of the mature gene expression data (spliced or moments of spliced), with sketch sizes of approximately 20% of the total data. Sketch indices were then used to subsample all modalities prior to integration and disease state classification evaluation.

Integration methods

Problem formulation

Let $X = \{x_i\}_{i=1}^n$ denote a single-cell dataset consisting of one gene expression modality, where $x_i \in \mathbb{R}^d$ represents a vector of d genes measured in cell i. Given a collection of m gene expression modalities $\{X^m\}_{k=1}^m$ sampled from N individuals, where for sample i there is an associated label y_i, our goal is to identify a biologically meaningful consensus representation, $Z = \{z_i \in \mathbb{R}^p\}_{i=1}^n$ where p represents shared latent features such that $p \leq d$. In this case, we wish to use this consensus representation to build a predictive model to infer biological trajectories or to predict the patient-specific or treatment-induced phenotypic label for sample i, y_i. In this section, we describe the methods selected for integrating two groups of gene expression modalities measured from the same set of cells, either moments of spliced counts with RNA velocity data or normalized and log transformed spliced and unspliced count matrices. For more details on implementation and hyperparameter tuning, see Additional file 1: Table S2.

Unintegrated To evaluate a baseline approach representing unintegrated data, we constructed a k-nearest neighbor graph ($k = 10$) from the top 50 principal components, generated from the normalized and log transformed spliced counts. This is akin to what is traditionally used for downstream single-cell analysis, as outlined by current best practices [51].

Concatenation Gene expression data matrices were horizontally concatenated, $\left[X^1 \| X^2\right]$ where $\|$ denotes concatenation, to obtain a merged data matrix with dimensions $n \times 2d$. Principal component analysis (PCA) was performed on the concatenated matrix and a k-nearest neighbors graph ($k = 10$) of cells was ultimately constructed based on the top 50 principal components.

Sum Gene expression data matrices were summed, $X^1 \oplus X^2$ where $\oplus$ denotes element-wise sum, to obtain a merged data matrix with dimensions $n \times d$. PCA was performed on the summed matrix and a k-nearest neighbor graph ($k = 10$) was constructed from the top 50 principal components.

CellRank CellRank [17] computes a joint transition probability matrix through a weighted sum of expression and velocity transition probability matrices as,

$$\mathrm{P} = \lambda \mathrm{P}_v + (1 - \lambda)\mathrm{P}_s \text{ for } \lambda \in [0, 1]. \tag{5}$$

Here, P_v represents the velocity transition matrix, P_s represents the expression similarity transition matrix, and λ is the weight parameter. Importantly, CellRank models cell state transitions using a Markov chain and assumes that (1) the sampled cells cover the entire state-change trajectory and (2) the transitions between cell states are gradual and can be directed according to local velocity vectors. We used CellRank v1.1.0 and performed hyperparameter tuning by varying the weight parameter λ, the measure of velocity similarity (correlation, dot product, or cosine), and the model that determines if velocity uncertainty is propagated in the transition matrix computation (monte-carlo,

dynamical). Given that this approach relies on RNA velocity directionality, integration was only performed using moments of spliced and RNA velocity data.

PRECISE PRECISE [50] was adapted to integrate temporal gene expression modalities. PRECISE first finds a linear subspace of the data by computing principal components for each modality individually, then geometrically aligns components to extract common principal vectors that represent similar weighted combinations of genes. From here, a consensus feature representation is computed by optimizing the match between interpolated sets of features (e.g., expression and velocity). For this analysis, we obtained a lower dimensional latent space by projecting expression data onto (1) the principal vectors (denoted as PRECISE) or (2) the consensus features (denoted as PRECISE consensus). From this shared embedding space, we constructed a k-nearest neighbor graph ($k = 10$). For both approaches, we performed hyperparameter tuning by varying the number of included principal vectors. Given that the principal vectors are rank ordered according to modality similarity, selection is analogous to filtering the data based on shared or unshared information. PRECISE v1.2 was used and modified to include dissimilar components.

Similarity network fusion Similarity network fusion (SNF) [25] constructs a joint graph of cells according to gene expression data modalities using a two-step process. First, a cell affinity graph $\mathcal{G}^m = (\mathcal{V}^m, \mathcal{E}^m)$ is computed for each modality, where $\mathcal{V}^m$ represents cells and edges, $\mathcal{E}^m$, are weighted according to modality-specific similarity using a heat kernel as follows. Here, we compute W_{ij}^m, which gives the specific edge-weight between cells i and j in modality m as,

$$\mathrm{W}_{\mathrm{ij}}^{\mathrm{m}} = \exp\left(-\frac{\|x_i^m - x_j^m\|^2}{\mu\epsilon_{ij}}\right). \tag{6}$$

Specifically, W^m is a $n \times n$ similarity matrix for modality m, μ is a scaling hyperparameter, and ϵ_{ij} is a bandwidth parameter that takes into account local neighborhood sizes. Here, SNF assumes that local similarities are a reliable representation of data and that remote ones can be modeled through graph diffusion on the network. Next, the two individual modality networks are integrated through nonlinear diffusion iterations between each modality to obtain a fused network. Importantly, the network fusion step ensures that the merged graph representation retains edge similarities that are strongly supported by an individual modality in addition to similarities shared across modalities. To compare results to the intermediate embedding integration methods, we modified SNF by constructing a shared embedding from the fused network through eigendecomposition of the unnormalized graph Laplacian L_u. Note that L_u is computed as,

$$\mathrm{L_u} = \mathrm{D} - \mathrm{A}. \tag{7}$$

Here, D is a diagonal degree matrix with i-th diagonal element, $d_i = \sum_j A_{ij}$ and A is the symmetric merged SNF affinity adjacency matrix. Given that eigenvectors of the Laplacian represent frequency harmonics, we selected the eigenvectors corresponding to the K smallest eigenvalues to low pass filter high frequency noise [83]. We then constructed a k-nearest neighbor graph ($k = 10$) for evaluation. We performed hyperparameter

tuning by varying the number of nearest neighbors, the bandwidth scaling parameter μ, and the number of eigenvectors for the merged graph embedding. SNF was implemented using the snfpy v0.2.2 package in python.

Grassmann joint embedding The Grassmann joint embedding approach introduced in Ref. [26] was adapted to construct a shared representative subspace of temporal gene expression information. Similar to SNF, the Grassmann embedding approach begins by constructing affinity matrices to encode similarities between cells i and j in each modality using a heat kernel as,

$$\mathrm{S}_{\mathrm{ij}}^{\mathrm{m}} = \exp\left(-\frac{\|x_i^m - x_j^m\|^2}{2t^2}\right). \tag{8}$$

Here, S^m is a $n \times n$ between-cell similarity matrix for modality m and t is the kernel bandwidth parameter. To prioritize local similarities, the k-nearest neighbors according to the similarity matrix S^m are identified and the similarity matrix is further redefined as,

$$\mathrm{W}_{\mathrm{ij}}^{\mathrm{m}} = \begin{cases} S_{ij}^m, & \text{if } v_j \in \mathcal{N}_i \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Specifically, cell v_i and v_j are connected with an edge with edge weight S_{ij} if the cell is within the set of v_i's neighbors, $\mathcal{N}_i$. Next, low-dimensional linear subspaces are computed through eigendecomposition of the normalized graph Laplacian of each data type. The normalized graph Laplacian L_n^m is formally defined as:

$$\mathrm{L}_n^m = \mathrm{D}^{m^{-\frac{1}{2}}}\left(\mathrm{D}^m - \mathrm{W}^m\right)\mathrm{D}^{m^{-\frac{1}{2}}}. \tag{10}$$

Here, m indexes the data modality and D^m represents a diagonal degree matrix, such that the i-th diagonal element, $d_i^m = \sum_j W_{ij}^m$. Furthermore, A^m is the symmetric Grassmann affinity matrix of modality m. A shared representative subspace from [26] is then ultimately computed as,

$$\mathrm{L}_{mod} = \sum_{k=1}^{m} \mathrm{L}_n^m - \alpha \sum_{k=1}^{m} \mathrm{U}^m \mathrm{U}^{m'}. \tag{11}$$

Here, U^m represents an individual subspace representation and α controls the trade-off between preserving modality-specific structural similarities (in the first term) and minimizing the distance between each subspace representation (in the second term). Lastly, an eigendecomposition of the Laplacian of the joint graph L_{mod} was computed to extract the K eigenvectors corresponding to the first K eigenvalues to represent the merged embedding space. For evaluation, we constructed a k-nearest neighbor graph ($k = 10$) from this shared space. Hyperparameter tuning was performed by varying the number of nearest neighbors and kernel bandwidth parameter t in the affinity graph construction, as well as α, and the number of eigenvectors to include for the merged graph embedding.

Integrated diffusion Integrated diffusion [24] combines data modalities by computing a joint data diffusion operator. First, individual modalities are locally denoised by

performing a truncated singular value decomposition (SVD) on local neighborhoods determined through spectral clustering. Next a symmetric diffusion operator is constructed for each denoised modality, and spectral entropy is used to determine the number of diffusion time steps to take. By taking the reduced ratio of information, the joint diffusion operator P_j is computed as:

$$\mathrm{P}_j = \mathrm{P}_1^{\mathrm{t}_1} \cdot \mathrm{P}_2^{\mathrm{t}_2}. \tag{12}$$

Here, P_1 and P_2 represent individual modality diffusion operators (e.g., expression and velocity) and t_1 and t_2 represent the reduced ratio of diffusion time steps, respectively. By powering transition probability matrices independently, this captures both modality-specific information, while allowing the random walk to jump between data types for merging. Lastly, the joint diffusion operator is powered using the same spectral entropy measure. It is important to note that approach assumes that high frequency signals may be low pass filtered; thus, choice of t can be crucial, as it can either effectively denoise data or remove important variation and lead to oversmoothing. We eigendecomposed the diffused joint operator and selected the eigenvectors corresponding to the K largest eigenvalues to obtain a merged lower dimensional representation. We then constructed a k-nearest neighbor graph ($k = 10$). Hyperparameter tuning was performed by varying the number of clusters for local denoising, the number of nearest neighbors in affinity graph construction, and the number of included eigenvectors.

Multi-Omics Factor Analysis v2 Multi-Omics Factor Analysis v2 (MOFA+) [30] merges data modalities through a statistical matrix factorization approach. In particular, MOFA+ decomposes each data modality as:

$$\mathrm{X}^m = \mathrm{Z}\mathrm{W}^{m^{\mathrm{T}}} + \epsilon^m. \tag{13}$$

Here, X^m denotes the matrix of observations for the modality m, Z denotes the shared factor matrix, W^m denotes the weight matrix for modality m, and ϵ^m denotes the residual noise for modality m. The model is formulated in a probabilistic Bayesian setting with sparsity priors and hierarchical variance regularization. We implement MOFA+ using the mofapy2 v0.6.4 package in python, assuming a Gaussian likelihood model where the residuals are normally distributed and selected the top 50 factors. For evaluation, we constructed a k-nearest neighbor graph ($k = 10$) from this shared space.

Seurat v4 Seurat v4 [49] constructs a joint graph of cells according to weighted nearest neighbor graph approach. First, individual k-nearest neighbor graphs are constructed for each modality to obtain local modality-specific neighborhoods. Then, using a two-step process, cell specific-modality weights are learned in order to determine the relative information content of each data type within the cell by: (1) computing within and cross modality predictions based upon modality-specific local neighborhoods and (2) computing the similarity between predicted and actual molecular profiles. Lastly, an integrated k-nearest neighborhood graph is ultimately constructed according to a similarity

metric defined by the weighted average of modality affinities. For this analysis, we constructed individual k-nearest neighbor graphs ($k = 10$) from the top 50 principal components. Seurat v4 integration was implemented in Seurat v4.1.1 using the FindMultiModalNeighbors function in R.

Evaluation

Trajectory inference

To quantify how well incorporation of unspliced counts or RNA velocity recapitulates the underlying biological trajectory, we compared predicted trajectories to a ground truth reference using the metrics implemented in the R suite Dynverse [56]. Reference trajectories were curated from the literature [36–39, 59], with cell groups, connections, and root cluster provided by the authors of the original study. We note that cell population annotations were externally determined through cell surface protein measurements and not from unsupervised clustering on the expression data.

To obtain predicted trajectories from integrated data, we performed trajectory inference using two approaches that were shown to outperform other methods for inference of complex or tree differentiation trajectories [56]. First, we evaluated trajectory inference on the integrated graphs from an integration strategy using partition-based graph abstraction [57] followed by diffusion pseudotime [58]. Predicted trajectories consisted of two main attributes: (1) a trajectory network, where nodes represent FACS cell groups and edges connect populations based on PAGA inferred connectivity generated from the integrated or unintegrated k-nearest neighbor graph and (2) a list of cellular percentages representing a cell's relative position between groups. Here, cellular percentages were determined from diffusion pseudotime using 20 diffusion map components. For each integration approach, we computed predicted trajectories for ten random root cells selected from the annotated root cluster. As a secondary approach, we also evaluated trajectory inference with Slingshot [60]. Here, the trajectory network consisted of the cluster minimum spanning tree and cellular percentages were determined from pseudotime estimated from the integrated or unintegrated embedding from an integration approach. Therefore, performance was evaluated for the integration methods that infer a joint latent space after specifying the annotated root cluster as the starting population.

To evaluate a method's performance on inferring developmental gene expression dynamics from integrated or unintegrated data, we compared reference and predicted trajectories using two metrics previously described in Ref. [56]: cell distance correlation and feature importance score correlation.

1. *Cell distance correlation* C_{corr}: Geodesic distances represent the shortest path distance between two cells on a nearest neighbor graph of the data [84]. To estimate a measure of the correlation of between-cell distances between reference and predicted trajectories, geodesic distances were computed between cells on a trajectory graph. The cell distance correlation is defined as the Spearman rank correlation between the geodesic cell distances of both trajectories.
2. *Feature importance score correlation* F_{corr}: To assess whether the same temporally expressed genes were found in the predicted trajectory as in the reference, a random

forest regression framework was used to predict the expression values of each gene based on geodesic distances of each cell to each cell state cluster. The feature importance score correlation is defined as the Pearson correlation between the reference and predicted scores.

To obtain an overall trajectory inference correlation score reflective of high cell and feature similarity, we compute the harmonic mean of both correlation metrics as,

$$\mathrm{TI}_{\mathrm{corr}} = 2 \cdot \frac{\mathrm{C}_{\mathrm{corr}} \cdot \mathrm{F}_{\mathrm{corr}}}{\mathrm{C}_{\mathrm{corr}} + \mathrm{F}_{\mathrm{corr}}}. \tag{14}$$

Classification

Label propagation To quantitatively compare integration methods on disease state prediction, we aimed to implement an approach that would use the underlying integrated or unintegrated graph structure. Label propagation [62] is a semi-supervised learning algorithm that uses iterative diffusion processes to predict the labels of unlabeled nodes. The output of this algorithm is a probability distribution of labels for every cell. We implemented label propagation to predict stimulation condition or disease status labels as follows.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, y = \{y_i\}_{i=1}^{n})$ denote a graph of n cells comprising the nodes ($\mathcal{V}$) generated from an integration approach and the set $\mathcal{E}$ edges encoding between-cell similarities. Similarly, a particular y_i gives a phenotypic label for cell i (e.g., patient disease status). Let $y' = (y_l, y_u)$ denote a vector consisting of a training subset of cells that are labeled $y_l = \{y_j\}_{j=1}^{m}$ where $y_j \in y$ and $m < n$, and a test subset of cells that are unlabeled, $y_u = \{0\}^{n-m}$. Given $\mathcal{G}$ and y', our goal is to assign a label to the unlabeled cells and the corresponding entries of $y's$. To do so, we perform the following approach.

1 Stratified random sampling is used to assign cells to a training or test set; this ensures that the original ratio of class labels (e.g., AML diagnosis or relapse) remains the same as in the full dataset.
2 Initialize algorithm on the training set to predict the labels of the masked test set. Each node has a label y'_i, and edge weight w_{ij} representing the strength of similarity between nodes i and j. Here, larger weights indicate a higher probability of cell i propagating its label y'_i to cell j.
3 Labels are iteratively updated through diffusion, where D is a diagonal degree matrix with i'th diagonal element $d_i = \sum_j W_{ij}$ as,

$$y'^{(t+1)} \leftarrow \mathrm{D}^{-1}\mathrm{W}y'^{(t)}. \tag{15}$$

4 Row normalize labels y' to maintain a probability distribution.
5 Training labels are clamped after each iteration as,

$$y_l^{(t+1)} \leftarrow y_l^{(t)}. \tag{16}$$

6 Iterations are repeated until convergence, with a threshold $\delta = 0.001$, such that,

$$|y'^{(t)} - y'^{(t-1)}| < \delta. \tag{17}$$

7 Class labels are assigned to every node by taking the label with the maximum probability.

We repeated this procedure for ten random training initializations to obtain a set of predicted labels for each integration approach.

Support vector machine (SVM) The support vector machine (SVM) [85] is a supervised learning algorithm that constructs hyperplanes in the high dimensional data to separate classes. We implemented SVM as a secondary classification approach for predicting perturbation response or disease status labels from the individual or joint embedding space (e.g., PCA, diffusion embedding). Specifically, nested 10-fold cross validation was performed using stratified random sampling to assign cells to either a training or test set. SVM hyperparameters were tuned over a grid search within each fold prior to training the model and labels were subsequently predicted from the test data.

Metrics To quantify stimulation condition and disease status classification performance, we compared predicted labels to ground truth annotations using three metrics: F1 score, balanced accuracy (acc_b), and area under the receiver operator curve (AUC). The F1 score measures the harmonic mean of precision and recall as,

$$\text{F}_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{18}$$

Balanced accuracy represents the average of sensitivity (true positive rate) and specificity (true negative rate). When predicting more than two labels (e.g., disease progression), we computed the mean sensitivity for all classes.

$$\text{acc}_\text{b} = \frac{\text{sensitivity} + \text{specificity}}{2} \tag{19}$$

Lastly, area under the receiver operator curve was computed using the soft probability assignments. For the multi-class case, each class label was compared to the remaining in an all vs. rest approach, then averaged. All of these metrics return a value between 0 and 1, where 1 indicates predicted labels were in perfect accordance to the ground truth annotations.

Aggregate scores

To rank methods for each prediction task, we compute aggregate scores by taking the mean of scaled method scores across datasets. More specifically, we first define an overall method score per dataset as the median of each metric. Method scores were subsequently min-max scaled to ensure datasets were equally weighted prior to computing the average.

Runtime scalability

To compare the runtime efficiency for each integration strategy, we randomly downsampled the INFγ stimulation dataset by (1) varying the number of cells (250, 500, 1000, 2500, 5000, 7500, 10,000) while keeping the number of features constant (1000) or (2) varying the number of features (250, 500, 1000, 1500, 2000, 2500, 3000) while keeping the number of cells constant (1000). All of the integration strategies were run on a Linux server allocating 16 cores and 32GB of memory (Intel Xeon E5-2680 v3 processors). MOFA+ was evaluated with GPU mode enabled (1 GPU Nvidia GeForce GTX1080). Moreover, elapsed time was averaged over 5 trials of random sampling prior to integration with moments of spliced and RNA velocity modalities. Of note, we used the native implementation of software when available (CellRank, SNF, MOFA+, and Seurat v4) or we reimplemented the code in Python when modified or unavailable (concatenation, sum, PRECISE, Grassmann joint embedding, and integrated diffusion).

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13059-022-02749-0.

Additional file 1. Supplementary Tables S1-S3 and Supplementary Figures S1-17.

Additional file 2. Review history.

Acknowledgements
The authors would like to thank Logan Whitehouse and Tarek Zikry for their thoughtful discussions related to this work.

Peer review information
Stephanie McClelland was the primary editor of this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Review history
The review history is available as Additional file 2.

Authors' contributions
JSR, NS, and JEP conceptualized and designed the study. JSR performed the data preprocessing, benchmarking, evaluation, and analysis. JSR wrote the manuscript with input from all authors. All authors read and approved of the final manuscript.

Funding
This work was supported by the National Institutes of Health, F31-HL156433 (JSR), 5T32-GM067553 (JSR), DP2-HD091800 (JEP), R01-GM138834 (JEP), and NSF CAREER Award 1845796 (JEP).

Availability of data and materials
The raw publicly available single-cell RNA sequencing datasets downloaded and used in this study are available in the Gene Expression Omnibus (GEO; https://www.ncbi.nlm.nih.gov/geo/) repository, under the accession codes GSE81682 for hematopoiesis differentiation (Nestorowa) [86]; GSE74596 for NKT cell differentiation [87]; GSE70236, GSE70240, and GSE70244 for hematopoiesis differentiation (Olsson) [88]; GSE94383 for LPS stimulation [89]; GSE161465 for INFγ stimulation [90]; GSE116481 for AML chemotherapy [91]; GSE126068 for AML diagnosis/relapse [92]; and GSE138266 for MS case/control PBMC and CSF datasets [93] and in the European Nucleotide Archive (ENA; https://www.ebi.ac.uk/arrayexpress/experiments/) repository, under accession numbers E-MTAB-2805 for mouse embryonic cell cycle [94] datasets, respectively. Loom files and preprocessed data are available in the Zenodo repository https://doi.org/10.5281/zenodo.6587903 [95]. Source code including all functions for preprocessing, integration, and evaluation are publicly available at www.github.com/jranek/EVI [96] and in the Zenodo repository [97]. Source code is released under the MIT license.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 1 March 2022 Accepted: 16 August 2022
Published online: 05 September 2022

References

1. Farrell JA, Wang Y, Riesenfeld SJ, Shekhar K, Regev A, Schier AF. Single-cell reconstruction of developmental trajectories during zebrafish embryogenesis. Science. 2018;360(6392):eaar3131. https://doi.org/10.1126/science.aar3131.
2. Crosse El, Gordon-Keylock S, Rybtsov S, Binagui-Casas A, Felchle H, Nnadi NC, et al. Multi-layered Spatial Transcriptomics Identify Secretory Factors Promoting Human Hematopoietic Stem Cell Development. Cell Stem Cell. 2020;27(5):822-39.e8.
3. Fawkner-Corbett D, Antanaviciute A, Parikh K, Jagielowicz M, Gerós AS, Gupta T, et al. Spatiotemporal analysis of human intestinal development at single-cell resolution. Cell. 2021;184(3):810-26.e23.
4. Kaufmann M, Evans H, Schaupp AL, Engler JB, Kaur G, Willing A, et al. Identifying CNS-colonizing T cells as potential therapeutic targets to prevent progression of multiple sclerosis. Med (N Y). 2021;2(3):296-312.e8.
5. Pollyea DA, Stevens BM, Jones CL, Winters A, Pei S, Minhajuddin M, et al. Venetoclax with azacitidine disrupts energy metabolism and targets leukemia stem cells in patients with acute myeloid leukemia. Nat Med. 2018;24(12):1859–66.
6. Schafflick D, Xu CA, Hartlehnert M, Cole M, Schulte-Mecklenbeck A, Lautwein T, et al. Integrated single cell analysis of blood and cerebrospinal fluid leukocytes in multiple sclerosis. Nat Commun. 2020;11(1):247.
7. Stetson LC, Balasubramanian D, Ribeiro SP, Stefan T, Gupta K, Xu X, et al. Single cell RNA sequencing of AML initiating cells reveals RNA-based evolution during disease progression. Leukemia. 2021;35(10):2799–812.
8. Weinreb C, Wolock S, Tusi BK, Socolovsky M, Klein AM. Fundamental limits on dynamic inference from single-cell snapshots. Proc Natl Acad Sci U S A. 2018;115(10):E2467-76.
9. Teschendorff AE, Feinberg AP. Statistical mechanics meets single-cell biology. Nat Rev Genet. 2021;22(7):459–76.
10. Kiselev VY, Andrews TS, Hemberg M. Challenges in unsupervised clustering of single-cell RNA-seq data. Nat Rev Genet. 2019;20(5):273–82.
11. Tritschler S, Büttner M, Fischer DS, Lange M, Bergen V, Lickert H, et al. Concepts and limitations for learning developmental trajectories from single cell genomics. Development. 2019;146(12):dev170506. Published 2019 Jun 27. https://doi.org/10.1242/dev.170506.
12. La Manno G, Soldatov R, Zeisel A, Braun E, Hochgerner H, Petukhov V, et al. RNA velocity of single cells. Nature. 2018;560(7719):494–8.
13. Bergen V, Lange M, Peidli S, Wolf FA, Theis FJ. Generalizing RNA velocity to transient cell states through dynamical modeling. Nat Biotechnol. 2020;38(12):1408–14.
14. Xia C, Fan J, Emanuel G, Hao J, Zhuang X. Spatial transcriptome profiling by MERFISH reveals subcellular RNA compartmentalization and cell cycle-dependent gene expression. Proc Natl Acad Sci U S A. 2019;116(39):19490–9.
15. Gorin G, Svensson V, Pachter L. Protein velocity and acceleration from single-cell multiomics experiments. Genome Biol. 2020;21(1):39.
16. Li C, Virgilio M, Collins KL, Welch JD. Single-cell multi-omic velocity infers dynamic and decoupled gene regulation. bioRxiv. 2021. https://doi.org/10.1101/2021.12.13.472472.
17. Lange M, Bergen V, Klein M, Setty M, Reuter B, Bakhti M, et al. CellRank for directed single-cell fate mapping. Nat Methods. 2022;19(2):159–70.
18. Qiu X, Rahimzamani A, Wang L, Ren B, Mao Q, Durham T, et al. Inferring Causal Gene Regulatory Networks from Coupled Single-Cell Expression Dynamics Using Scribe. Cell Syst. 2020;10(3):265-74.e11.
19. Weng G, Kim J, Won KJ. VeTra: a tool for trajectory inference based on RNA velocity. Bioinformatics. 2021;37(20):3509–13.
20. Tong A, Huang J, Wolf G, van Dijk D, Krishnaswamy S. TrajectoryNet: A Dynamic Optimal Transport Network for Modeling Cellular Dynamics. Proc Mach Learn Res. 2020;119:9526–36.
21. Zhang Z, Zhang X. Inference of high-resolution trajectories in single-cell RNA-seq data by using RNA velocity. Cell Rep Methods. 2021;1(6):100095.
22. Atta L, Sahoo A, Fan J. VeloViz: RNA velocity informed embeddings for visualizing cellular trajectories. Bioinformatics. 2021;38(2):391–6.
23. Subramanian I, Verma S, Kumar S, Jere A, Anamika K. Multi-omics data integration, interpretation, and its application. Bioinform Biol Insights. 2020;14:1177932219899051.
24. Kuchroo M, Godavarthi A, Wolf G, Krishnaswamy S. Multimodal data visualization, denoising and clustering with integrated diffusion. arXiv. 2021. https://doi.org/10.48550/arXiv.2102.06757.
25. Wang B, Mezlini AM, Demir F, Fiume M, Tu Z, Brudno M, et al. Similarity network fusion for aggregating data types on a genomic scale. Nat Methods. 2014;11(3):333–7.
26. Ding H, Sharpnack M, Wang C, Huang K, Machiraju R. Integrative cancer patient stratification via subspace merging. Bioinformatics. 2019;35(10):1653–9.
27. Wang B, Zhu J, Pierson E, Ramazzotti D, Batzoglou S. Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. Nat Methods. 2017;14(4):414–6.
28. Ramazzotti D, Lal A, Wang B, Batzoglou S, Sidow A. Multi-omic tumor data reveal diversity of molecular mechanisms that correlate with survival. Nat Commun. 2018;9(1):4453.
29. Argelaguet R, Velten B, Arnol D, Dietrich S, Zenz T, Marioni JC, et al. Multi-Omics Factor Analysis-a framework for unsupervised integration of multi-omics data sets. Mol Syst Biol. 2018;14(6): e8124.
30. Argelaguet R, Arnol D, Bredikhin D, Deloro Y, Velten B, Marioni JC, et al. MOFA+: a statistical framework for comprehensive integration of multi-modal single-cell data. Genome Biol. 2020;21(1):111.
31. Lock EF, Hoadley KA, Marron JS, Nobel AB. Joint and individual variation explained (jive) for integrated analysis of multiple data types. Ann Appl Stat. 2013;7(1):523–42.
32. Chalise P, Ni Y, Fridley BL. Network-based integrative clustering of multiple types of genomic data using non-negative matrix factorization. Comput Biol Med. 2020;118: 103625.

33. Velten B, Braunger JM, Argelaguet R, Arnol D, Wirbel J, Bredikhin D, et al. Identifying temporal and spatial patterns of variation from multimodal data using MEFISTO. Nat Methods. 2022;19(2):179–86.
34. Gundersen G, Ash JT, Engelhardt BE. End-to-end training of deep probabilistic CCA on paired biomedical observations. http://proceedings.mlr.press/v115/gundersen20a/gundersen20a.pdf. Accessed 27 Jan 2022.
35. Zeng T, Dai H. Single-cell RNA sequencing-based computational analysis to describe disease heterogeneity. Front Genet. 2019;10:629.
36. Buettner F, Natarajan KN, Casale FP, Proserpio V, Scialdone A, Theis FJ, et al. Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. Nat Biotechnol. 2015;33(2):155–60.
37. Nestorowa S, Hamey FK, Pijuan Sala B, Diamanti E, Shepherd M, Laurenti E, et al. A single-cell resolution map of mouse hematopoietic stem and progenitor cell differentiation. Blood. 2016;128(8):e20-31.
38. Engel I, Seumois G, Chavez L, Samaniego-Castruita D, White B, Chawla A, et al. Innate-like functions of natural killer T cell subsets result from highly divergent gene programs. Nat Immunol. 2016;17(6):728–39.
39. Olsson A, Venkatasubramanian M, Chaudhri VK, Aronow BJ, Salomonis N, Singh H, et al. Single-cell analysis of mixed-lineage states leading to a binary cell fate choice. Nature. 2016;537(7622):698–702.
40. Lane K, Van Valen D, DeFelice MM, Macklin DN, Kudo T, Jaimovich A, et al. Measuring signaling and RNA-Seq in the same cell links gene expression to dynamic patterns of NF-κB activation. Cell Syst. 2017;4(4):458-69.e5.
41. Dorrington MG, Fraser IDC. NF-κB signaling in macrophages: dynamics, crosstalk, and signal integration. Front Immunol. 2019;10:705.
42. Burkhardt DB, Stanley JS 3rd, Tong A, Perdigoto AL, Gigante SA, Herold KC, et al. Quantifying the effect of experimental perturbations at single-cell resolution. Nat Biotechnol. 2021 May;39(5):619–29.
43. Lopes M, Kutlu B, Miani M, Bang-Berthelsen CH, Størling J, Pociot F, et al. Temporal profiling of cytokine-induced genes in pancreatic β-cells by meta-analysis and network inference. Genomics. 2014;103(4):264–75.
44. Huntly BJP, Gilliland DG. Leukaemia stem cells and the evolution of cancer-stem-cell research. Nat Rev Cancer. 2005;5(4):311–21.
45. Compston A, Coles A. Multiple sclerosis. Lancet. 2008;372(9648):1502–17.
46. Picard M, Scott-Boyer MP, Bodein A, Périn O, Droit A. Integration strategies of multi-omics data for machine learning analysis. Comput Struct Biotechnol J. 2021;19:3735–46.
47. Lake BB, Chen S, Sos BC, Fan J, Kaeser GE, Yung YC, et al. Integrative single-cell analysis of transcriptional and epigenetic states in the human adult brain. Nat Biotechnol. 2018;36(1):70–80.
48. Mo Q, Li R, Adeegbe DO, Peng G, Chan KS. Integrative multi-omics analysis of muscle-invasive bladder cancer identifies prognostic biomarkers for frontline chemotherapy and immunotherapy. Commun Biol. 2020;3(1):784.
49. Hao Y, Hao S, Andersen-Nissen E, Mauck WM 3rd, Zheng S, Butler A, et al. Integrated analysis of multimodal single-cell data. Cell. 2021;184(13):3573-87.e29.
50. Mourragui S, Loog M, van de Wiel MA, Reinders MJT, Wessels LFA. PRECISE: a domain adaptation approach to transfer predictors of drug response from pre-clinical models to tumors. Bioinformatics. 2019;35(14):i510-9.
51. Luecken MD, Theis FJ. Current best practices in single-cell RNA-seq analysis: a tutorial. Mol Syst Biol. 2019;15(6): e8746.
52. Stahlschmidt SR, Ulfenborg B, Synnergren J. Multimodal deep learning for biomedical data fusion: a review. Brief Bioinform. 2022;23(2):bbab569. https://doi.org/10.1093/bib/bbab569.
53. Fukui A, Park DH, Yang D, Rohrbach A, Darrell T, Rohrbach M. Multimodal compact bilinear pooling for visual question answering and visual grounding. arXiv. 2016. https://doi.org/10.48550/arXiv.1606.01847.
54. Kharchenko PV, Silberstein L, Scadden DT. Bayesian approach to single-cell differential expression analysis. Nat Methods. 2014;11(7):740–2.
55. Torregrosa G, Garcia-Ojalvo J. Mechanistic models of cell-fate transitions from single-cell data. Curr Opin Syst Biol. 2021;26:79–86.
56. Saelens W, Cannoodt R, Todorov H, Saeys Y. A comparison of single-cell trajectory inference methods. Nat Biotechnol. 2019;37(5):547–54.
57. Wolf FA, Hamey FK, Plass M, Solana J, Dahlin JS, Göttgens B, et al. PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. Genome Biol. 2019;20(1):59.
58. Haghverdi L, Büttner M, Wolf FA, Buettner F, Theis FJ. Diffusion pseudotime robustly reconstructs lineage branching. Nat Methods. 2016;13(10):845–8.
59. Laurenti E, Göttgens B. From haematopoietic stem cells to complex differentiation landscapes. Nature. 2018;553(7689):418–26.
60. Street K, Risso D, Fletcher RB, Das D, Ngai J, Yosef N, et al. Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. BMC Genomics. 2018;19(1):477.
61. Lei Y, Tang R, Xu J, Wang W, Zhang B, Liu J, et al. Applications of single-cell sequencing in cancer research: progress and perspectives. J Hematol Oncol. 2021;14(1):91.
62. Zhu X, Ghahramani Z. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University. 2002.
63. Bergen V, Soldatov RA, Kharchenko PV, Theis FJ. RNA velocity-current challenges and future perspectives. Mol Syst Biol. 2021;17(8):e10282.
64. Gorin G, Fang M, Chari T, Pachter L. RNA velocity unraveled. bioRxiv. 2022. https://doi.org/10.1101/2022.02.12.480214.
65. Soneson C, Srivastava A, Patro R, Stadler MB. Preprocessing choices affect RNA velocity results for droplet scRNA-seq data. PLoS Comput Biol. 2021;17(1):e1008585.
66. Levine JH, Simonds EF, Bendall SC, Davis KL, Amir EAD, Tadmor MD, et al. Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells that Correlate with Prognosis. Cell. 2015;162(1):184–97.
67. Dann E, Henderson NC, Teichmann SA, Morgan MD, Marioni JC. Differential abundance testing on single-cell data using k-nearest neighbor graphs. Nat Biotechnol. 2021;40(2):245–53.
68. Lun ATL, Richard AC, Marioni JC. Testing for differential abundance in mass cytometry data. Nat Methods. 2017;14(7):707–9.

69. Pratapa A, Jalihal AP, Law JN, Bharadwaj A, Murali TM. Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. Nat Methods. 2020;17(2):147–54.
70. Tedesco M, Giannese F, Lazarević D, Giansanti V, Rosano D, Monzani S, et al. Chromatin Velocity reveals epigenetic dynamics by single-cell profiling of heterochromatin and euchromatin. Nat Biotechnol. 2021;40(2):235–44.
71. Lun ATL, Bach K, Marioni JC. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. Genome Biol. 2016;17:75.
72. Satija R, Farrell JA, Gennert D, Schier AF, Regev A. Spatial reconstruction of single-cell gene expression data. Nat Biotechnol. 2015;33(5):495–502.
73. Zeisel A, Köstler WJ, Molotski N, Tsai JM, Krauthgamer R, Jacob-Hirsch J, et al. Coupled pre-mRNA and mRNA dynamics unveil operational strategies underlying transcriptional responses to stimuli. Mol Syst Biol. 2011;7:529.
74. Johnson WE, Li C, Rabinovic A. Adjusting batch effects in microarray expression data using empirical Bayes methods. Biostatistics. 2007;8(1):118–27.
75. Haghverdi L, Lun ATL, Morgan MD, Marioni JC. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. Nat Biotechnol. 2018;36(5):421–7.
76. Hie B, Bryson B, Berger B. Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. Nat Biotechnol. 2019;37(6):685–91.
77. Luecken MD, Büttner M, Chaichoompu K, Danese A, Interlandi M, Mueller MF, et al. Benchmarking atlas-level data integration in single-cell genomics. Nat Methods. 2022;19(1):41–50.
78. Lab H. Batch effects in scRNA velocity analysis. https://www.hansenlab.org/velocity_batch. Accessed 16 Feb 2022.
79. Büttner M, Miao Z, Wolf FA, Teichmann SA, Theis FJ. A test metric for assessing single-cell RNA-seq batch correction. Nat Methods. 2019;16(1):43–9.
80. Korsunsky I, Millard N, Fan J, Slowikowski K, Zhang F, Wei K, et al. Fast, sensitive and accurate integration of single-cell data with Harmony. Nat Methods. 2019;16(12):1289–96.
81. Traag VA, Waltman L, van Eck NJ. From Louvain to Leiden: guaranteeing well-connected communities. Sci Rep. 2019;9(1):5233.
82. Hie B, Cho H, DeMeo B, Bryson B, Berger B. Geometric sketching compactly summarizes the single-cell transcriptomic landscape. Cell Syst. 2019;8(6):483-93.e7.
83. Shuman DI, Narang SK, Frossard P, Ortega A, Vandergheynst P. The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains. arXiv. 2012. https://doi.org/10.48550/arXiv.1211.0053.
84. Tenenbaum JB, de Silva V, Langford JC. A global geometric framework for nonlinear dimensionality reduction. Science. 2000;290(5500):2319–23.
85. Cortes C, Vapnik V. Support-vector networks. Mach Learn. 1995;20(3):273–97.
86. Nestorowa S, Hamey FK, Pijuan Sala B, Diamanti E, Shepherd M, Laurenti E, et al. A single-cell resolution map of mouse hematopoietic stem and progenitor cell differentiation. Datasets. Gene Expression Omnibus. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE81682. Accessed 9 Sept 2021.
87. Engel I, Seumois G, Chavez L, Samaniego-Castruita D, White B, Chawla A, et al. Innate-like functions of natural killer T cell subsets result from highly divergent gene programs. Datasets. Gene Expression Omnibus. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE74596. Accessed 11 May 2022.
88. Olsson A, Venkatasubramanian M, Chaudhri VK, Aronow BJ, Salomonis N, Singh H, et al. Single-cell analysis of mixed-lineage states leading to a binary cell fate choice. Datasets. Gene Expression Omnibus. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE70236, https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE70240, https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE70244. Accessed 11 May 2022.
89. Lane K, Van Valen D, DeFelice MM, Macklin DN, Kudo T, Jaimovich A, et al. Measuring Signaling and RNA-Seq in the Same Cell Links Gene Expression to Dynamic Patterns of NF-κB Activation. Datasets. Gene Expression Omnibus. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE94383. Accessed 7 Oct 2021.
90. Burkhardt DB, Stanley JS 3rd, Tong A, Perdigoto AL, Gigante SA, Herold KC, et al. Quantifying the effect of experimental perturbations at single-cell resolution. Datasets. Gene Expression Omnibus. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE161465. Accessed 7 Oct 2021.
91. Pollyea DA, Stevens BM, Jones CL, Winters A, Pei S, Minhajuddin M, et al. Venetoclax with azacitidine disrupts energy metabolism and targets leukemia stem cells in patients with acute myeloid leukemia. Datasets. Gene Expression Omnibus. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE116481. Accessed 7 Oct 2021.
92. Stetson LC, Balasubramanian D, Ribeiro SP, Stefan T, Gupta K, Xu X, et al. Single cell RNA sequencing of AML initiating cells reveals RNA-based evolution during disease progression. Datasets. Gene Expression Omnibus. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE126068. Accessed 7 Oct 2021.
93. Schafflick D, Xu CA, Hartlehnert M, Cole M, Schulte-Mecklenbeck A, Lautwein T, et al. Integrated single cell analysis of blood and cerebrospinal fluid leukocytes in multiple sclerosis. Datasets. Gene Expression Omnibus. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE138266. Accessed 7 Oct 2021.
94. Buettner F, Natarajan KN, Casale FP, Proserpio V, Scialdone A, Theis FJ, et al. Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. Datasets. European Nucleotide Archive. https://www.ebi.ac.uk/arrayexpress/experiments/E-MTAB-2805. Accessed 16 Nov 2021.
95. Ranek JS, Stanley N, Purvis JE. Preprocessed datasets for temporal gene expression integration. Zenodo. 2022. https://doi.org/10.5281/zenodo.6587903.
96. Ranek JS, Stanley N, Purvis JE. Expression and Velocity integration (EVI). GitHub. https://github.com/jranek/EVI. Accessed 27 May 2022.
97. Ranek JS, Stanley N, Purvis JE. Integrating temporal single-cell gene expression modalities for trajectory inference and disease prediction. Source code Zenodo. 2022. https://doi.org/10.5281/zenodo.6986191.

Publisher's Note