Genome Biology

METHODOLOGY

Open Access

spVelo: RNA velocity inference for multi-batch spatial transcriptomics data

Wenxin Long[1†], Tianyu Liu[2,3†], Lingzhou Xue[1*] and Hongyu Zhao[2,3*]

†Wenxin Long and Tianyu Liu contributed equally to this work.

*Correspondence: lzxue@psu.edu; hongyu.zhao@yale.edu

[1] Department of Statistics, The Pennsylvania State University, University Park 16802, PA, USA
[2] Department of Biostatistics, Yale University, New Haven 06510, CT, USA
[3] Interdepartmental Program of Computational Biology and Bioinformatics, Yale University, New Haven 06510, CT, USA

Abstract

RNA velocity has emerged as a powerful tool to interpret transcriptional dynamics and infer trajectory from snapshot datasets. However, current methods fail to utilize the spatial information inherent in spatial transcriptomics and lack scalability in multi-batch datasets. Here, we introduce spVelo, a scalable framework for RNA velocity inference of multi-batch spatial transcriptomics data. spVelo supports several downstream applications, including uncertainty quantification, complex trajectory pattern discovery, driver marker identification, gene regulatory network inference, and temporal cell-cell communication inference. spVelo has the potential to provide deeper insights into complex tissue organization and underscore biological mechanisms based on spatially resolved patterns.

Keywords: RNA velocity, Spatial transcriptomics, Transcriptomics dynamics, Trajectory inference, Variational inference

Background

Advances in sequencing technology have facilitated the reconstruction of cellular trajectories, revealing underlying dynamic processes [1–3]. Trajectory inference methods typically order cells along the pseudo-time axes based on similarities in their expression patterns [4–7]. However, traditional trajectory inference methods usually require prior knowledge of initial states or rely on certain assumptions, limiting the reliability and interpretability of these methods [5].

Recently, RNA velocity has become an alternative approach for trajectory inference. RNA velocity describes the rate of expression change for a single gene at a given time point, based on spliced and unspliced counts of messenger RNA (mRNA) [8]. The velocities of genes can then be used to estimate the future transcriptional states of cells, offering a powerful tool for understanding cellular differentiation, lineage tracing, and dynamical processes [9].

Current popular RNA velocity methods make different modeling assumptions. Velocyto [8] used a steady state model, which assumes that each gene undergoes prolonged

induction and repression phases reaching equilibrium, and all genes share a common splicing rate. The likelihood-based dynamical model introduced in scVelo [10] relaxed this steady-state assumption by generalizing to four transcriptional states. scVelo infers the full set of transcriptional parameters and estimates a latent time per cell, per gene by formulating the problem in an expectation-maximization (EM) framework. However, the kinetics are still explained with a deterministic system of linear differential equations with constant kinetic rate parameters. This assumption may not hold in complex biological systems where kinetic parameters can vary substantially among different genes, leading to poor RNA velocity inference in complicated dynamical features such as transcriptional boost [11], lineage-dependent kinetics, and weak unspliced signals [12]. Several methods have been further developed to resolve these limitations: UniTVelo [13] addressed this by modeling spliced gene expression using radial basis function (RBF) instead of ODEs, allowing more flexible gene expression profile modeling, though it still uses a unified latent time. LatentVelo [14] utilized neural ordinary differential equations (neural ODEs [15]) on embedded latent space while performing batch effect correction. The annotated mode of LatentVelo further added cell type information by modifying the prior. veloVI [16] reformulated RNA velocity in a Bayesian deep generative framework, inferring posterior distributions over kinetic parameters and latent cell states, while allowing for gene-specific latent times coupled through a shared low-dimensional representation.

While these methods have been successfully used to infer cellular dynamics [17, 18], they also suffer from several limitations [12, 19]. For example, current RNA velocity inference methods are confined to scRNA-seq data, which only captures the transcriptional profiles, losing the spatial context [20]. Spatial transcriptomics, a rapidly emerging technology, addresses this limitation by measuring the spatial information of gene expression. Spatial resolution determines the relative positions of cells and further reflects the communication and transitory relationships between adjacent cells. Utilizing spatial information can enable better inference of RNA velocity and trajectory, proven by the ablation test in Additional file 1: Fig. S1. Furthermore, current methods are confined to velocity inference in a single batch. This prevents the methods from utilizing the information from the entire dataset, thus failing to capture the global dynamics.

To address these limitations, we present spVelo (**sp**atial **Velo**city inference), a method for estimating RNA velocity in multi-batch spatial transcriptomics data. spVelo combines a Variational AutoEncoder (VAE) [21] for gene expression data with a Graph Attention Network (GAT) [22] for spatial location. By further adding a Maximum Mean Discrepancy (MMD) penalty [23] between latent spaces of different batches, spVelo is able to perform RNA velocity inference in a multi-batch spatial dataset. We compare spVelo with alternative methods using spatial data simulated from mouse pancreas data [24] and real oral squamous cell carcinoma (OSCC) data [25]. spVelo outperforms the previous RNA velocity inference methods for inferring RNA velocity and trajectory. Then, we demonstrate spVelo's ability to perform batch effect correction on RNA velocity [14]. By leveraging the distributions of latent space, spVelo is able to quantify the uncertainty of the inferred latent state. We further show that spVelo can discover complex trajectory patterns, while other methods tend to predict a linear trajectory between cell types. By visualizing predicted phase portraits, spVelo is able to fit the genes' dynamics well.

Additionally, spVelo can select biologically significant state driver markers that are validated through enrichment test using oncogenic gene sets from MSigDB [26, 27]. Finally, we present spVelo's downstream applications, providing new insight into RNA velocity.

Results

spVelo infers RNA velocity for multi-batch spatial transcriptomics data

spVelo first log-normalizes and smooths the data, and then filters uninformative genes based on their contributions to cell development. Utilizing GO analysis in Additional file 1: Fig. S2, we demonstrate that the filtered uninformative genes are less enriched for tumor-related pathways (e.g., cytoplasmic translation, structural molecule activity), compared to other informative genes. spVelo then models unspliced and spliced expression for each gene in a cell as a function of kinetic parameters (transcription, splicing, and degradation rates), latent time, and latent transcriptional state. In each cell, each gene's latent times are tied via a low-dimensional latent variable, following the model assumptions of veloVI [16].

spVelo models the gene expression data with a VAE including two orthogonal encoders. The Multi-Layer Perceptron (MLP) encoder takes the unspliced and spliced expression as input, and outputs the posterior distributions of the latent variable. Then, spVelo uses spatial location proximity and distance between batches as the input for a GAT encoder. By adding up the latent space of the two encoders, spVelo can jointly model the spatial location and gene expression data. Then, by variational posterior inference, spVelo can estimate the kinetic rates and latent time, and then further infer velocity. Additionally, we provide downstream applications including uncertainty quantification, trajectory patterns discovery, state driver markers identification, Gene Regulatory Network (GRN) inference, and temporal cell-cell communication (CCC) inference. A detailed explanation of the spVelo model can be found in the Methods section, and the model architecture is shown in Fig. 1. spVelo improves model performance and provides interpretable results and downstream applications of RNA velocity, suggesting the efficacy of its model design.

spVelo infers accurate velocity and trajectory

We first evaluated the performance of spVelo on a spatial dataset simulated from scRNA-seq pancreas data [24] using scCube [28], and a real OSCC dataset [25]. We compared the performance of velocity with other models, including stochastic mode and dynamical mode of scVelo [10], veloVI [16], standard mode and annotated mode of LatentVelo [14]. Since RNA velocity is defined as the time derivative of gene expression [8], and we cannot directly measure the instantaneous rate of expression change at a single-cell level, ground truth RNA velocities are unknown. As a result, we made use of the known cell type labels to define transition relationships. To evaluate RNA velocity methods in the absence of ground truth, we rely on several criteria that a good velocity field should satisfy: (1) consistency within its local neighborhood; (2) alignment between predicted future gene expression changes and the actual observed transcriptomic changes; and (3) coherence between predicted cell movement direction and observed cell displacement in PCA. We evaluated the performance of all methods based on the velocity confidence score, transition score, and direction score. The velocity confidence score measures the

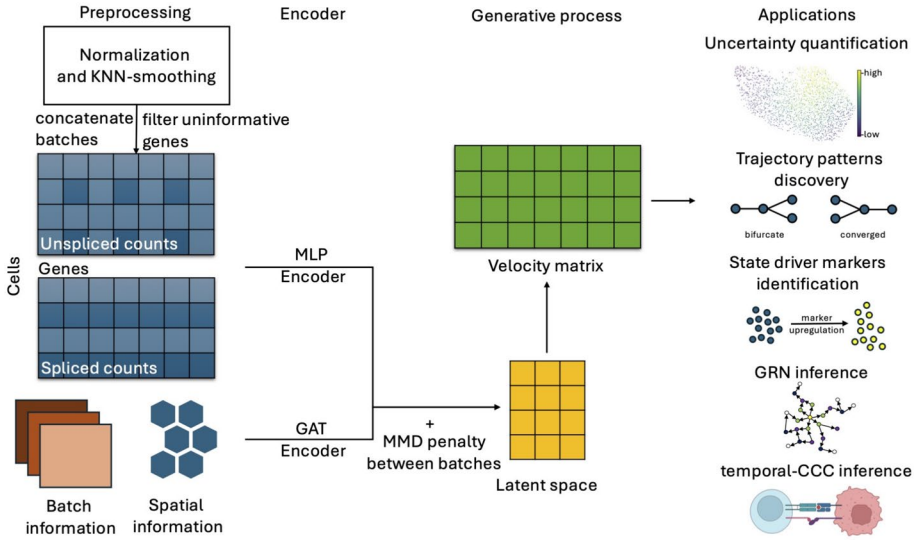

Fig. 1 Overview of spVelo. spVelo jointly models the spatial location and gene expression data by using an MLP encoder to encode information from the expression level, and a GAT encoder to encode spatial and batch information. After posterior inference, the velocity matrix can be used for downstream applications

reliability of inferred velocities, the transition score assesses the probability of true cell-to-cell transition, and the direction score evaluates the consistency of transition directions with known cell type transitions. The three scores are calculated respectively using neighbors of expression data, spatial neighbors in each batch, and mutual nearest neighbors between batches. These metrics are capable of comprehensively evaluating estimated RNA velocity based on the criteria stated previously. Detailed explanations of metrics can be found in the Methods section. Since all methods except LatentVelo are restricted to inferring velocity on a per-batch basis, for fairness, we utilized scGen [29] to correct batch effect prior to applying the velocity inference methods. These methods are denoted as scGen + `<method name>` in Fig. 2. For comparing only the per-batch scores (expr scores and spatial scores), we compared spVelo with both scGen-corrected methods and original per-batch methods. Figure 2a and c show plots of the nine scores for each method by averaging across different seeds and different batches, while Fig. 2b and d show dotplots of only the six per-batch scores for all methods. Here, we did not compare LatentVelo in the simulated pancreas dataset since it reported errors when the input data were in the logcounts format.

Dotplots in Fig. 2a–d demonstrate that spVelo ranks high when compared to all methods, especially in the direction score, which is the most important score for evaluating velocity's performance in trajectory inference. Overall, spVelo consistently achieves the highest average scores across all datasets, as illustrated in the final column. This highlights spVelo's ability to accurately capture the underlying cellular dynamics. All scores are visualized in Additional file 1: Fig. S3 and Additional file 1: Fig. S4. We further performed an ablation test to remove spatial information from our model. Results are visualized in Additional file 1: Fig. S1 and reveal that the integration of spatial information during model training significantly improves the performance of velocity and trajectory inference.

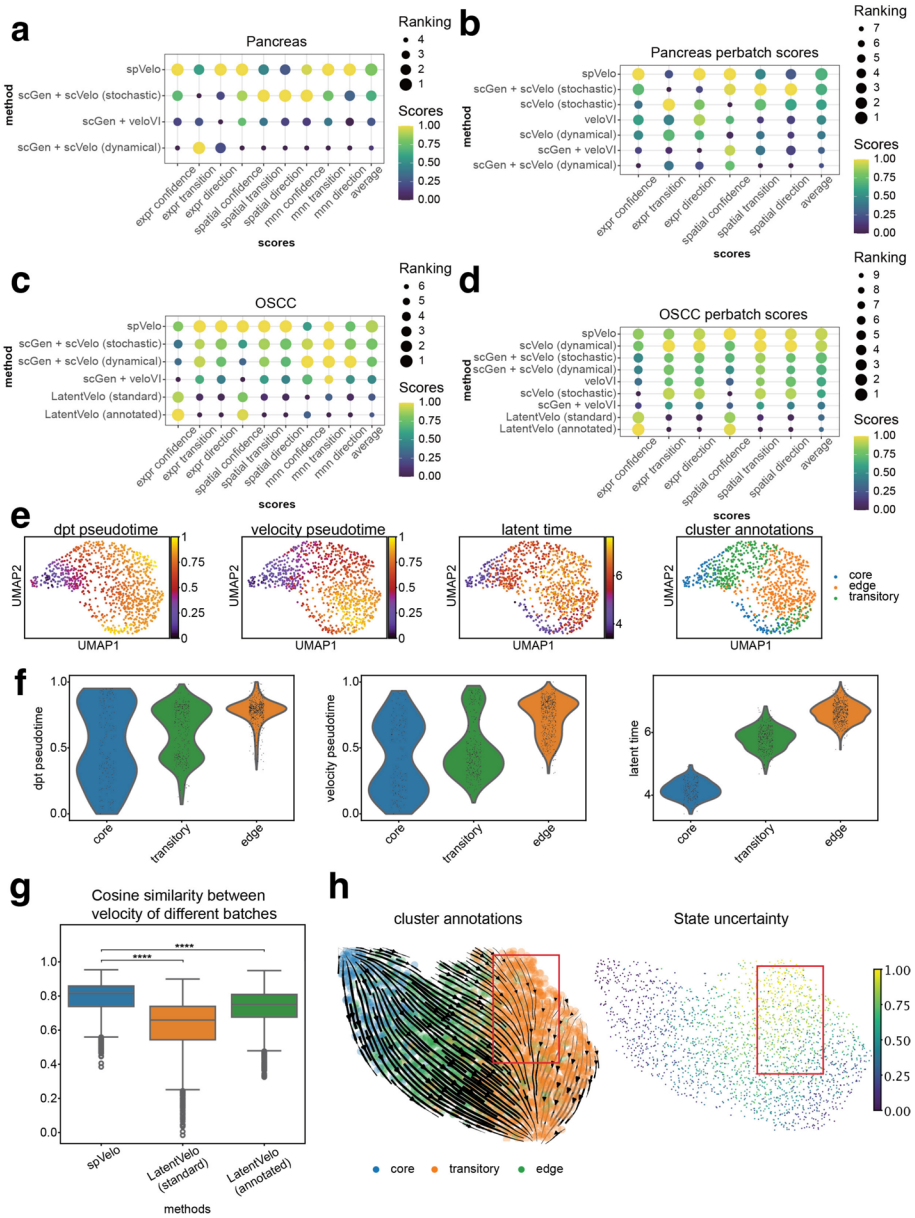

Fig. 2 Compare results for simulated pancreas dataset and OSCC dataset. **a** Dotplot of comparing all scores in simulated pancreas dataset. Each score is minmax-scaled across all methods. **b** Dotplot of comparing only per-batch scores in simulated pancreas dataset. Each score is minmax-scaled across all methods. **c** Dotplot of comparing all scores in the OSCC dataset. Each score is minmax-scaled across all methods. **d** Dotplot of comparing only per-batch scores in OSCC dataset. Each score is minmax-scaled across all methods. **e** Pseudo-time scatter plot of latent time inferred by spVelo, compared with DPT pseudo-time and velocity pseudo-time. **f** Pseudo-time violin plot of latent time inferred by spVelo, compared with DPT pseudo-time and velocity pseudo-time. **g** Comparison of cosine similarity between the velocity of different batches in MNN graph. **h** Streamline plot of trajectory and scatter plot of quantified uncertainty for sample 9 of OSCC dataset. The red frame in the streamline plot indicates the lineage with high uncertainty cells

Furthermore, we examined the latent time estimated by spVelo and compared it with pseudo-time inferred using Diffusion Pseudo-Time (DPT) [30] and pseudo-time inferred using diffusion-based random walk on RNA velocity matrix. The results are shown as the scatter plots and violin plots in Fig. 2e and f, and all other results are shown in

Additional file 1: Fig. S5. The plots reveal that our inferred latent time is distinct between different cell types and better matches with the ground truth.

To evaluate the ability of spVelo to correct batch effect in RNA velocity inference, we calculated the cosine similarity between the velocity of mutual nearest neighbor cells in different batches. The comparison result to LatentVelo is visualized in Fig. 2g. The boxplot reveals that spVelo infers significantly more coherent velocity than LatentVelo. To quantitatively show the contributions of spVelo's MMD penalty to velocity coherence across batches, we further performed ablation studies by removing spVelo's MMD penalty. More details can be found in Additional file 1: Text S1.1 and results are visualized in Additional file 1: Fig. S6.

This shows that, with the MMD penalty between latent space of different batches, spVelo is able to infer more coherent velocity between batches. The coherence in velocity may also facilitate more accurate trajectory inference, since the aligned velocities better reflect the true underlying biological processes rather than noise.

Following the suggestions of reviewers, we further considered proving spVelo's consistency across spatial datasets, despite differences in resolution and platform design. We compared all methods on a new stereo-seq mousebrain dataset [31], processed with bin size 60. More details can be found in Additional file 1: Text S2 and Additional file 1: Table S1. Then, we also compared spVelo with existing spatially aware velocity inference methods, including STT [32], SIRV [33], scGen+STT, and scGen+SIRV. The results can be found in Additional file 1: Text S2 and Additional file 1: Table S2.

spVelo quantifies uncertainty for cell state

Since spVelo is a generative model, the distribution of its latent space can be used for uncertainty quantification. Inspired by VeloVAE [34], we calculated differential entropy on the variance of the latent space. Since the latent space is a low-dimension representation of cells, the differential entropy can be used as the uncertainty measurement for cell state [35], where higher differential entropy indicates a higher uncertainty score.

We visualized the streamline plot of trajectory and the scatter plot of quantified uncertainty for sample 9 of the OSCC dataset in Fig. 2h. Results of other samples are visualized in Additional file 1: Fig. S7. The plots reveal that some edge cells show higher uncertainty levels. These cells are mostly located at the starting area of the lineage in the red frame, suggesting heterogeneity in the edge cells. This observation also matches with the interpretation in VeloVAE that multi-potent progenitor cells have higher cell state uncertainty [34]. To further prove high-uncertainty regions reflect meaningful biological heterogeneity rather than model instability, we conducted pathway enrichment analysis using GSEA prerank. We divided cells from batch 9 into high- and low-uncertainty groups (top and bottom 50%) and identified differentially expressed genes using the Wilcoxon test. For the high-uncertainty group, we constructed a ranked gene list based on the Wilcoxon test statistic and applied GSEA prerank with MSigDB (C2 collection). The result is visualized in Additional file 1: Fig. S8. The pathways chosen in this figure are EMT- and plasticity-associated gene sets, which play a central role in driving cancer cell metastasis and lead to different signaling patterns and therapeutic responses [36]. According to the figure, genes with lower Wilcoxon test statistics (i.e., differentially expressed genes in low-uncertainty cells) are significantly enriched in the pathways,

supporting the interpretation that high-uncertainty cells are more heterogeneous and potentially multi-potent. As a result, the uncertainty quantification from spVelo allows researchers to identify and examine the regions with high variability, and further understand intricate biological mechanisms.

spVelo discovers complex trajectory patterns

In this section, we investigated the trajectory inferred using velocity from different methods. From Fig. 3a, spVelo inferred a bifurcate trajectory from sample 12 of the OSCC dataset. To validate the inferred bifurcate trajectory, we visualized how spliced expression varies along with the latent time inferred by spVelo in scatter plots. Velocity clusters were calculated by using Leiden clustering [37] on the inferred velocity matrix. Expression data and latent time were calculated by averaging the top five markers of edge (1) cells and edge (2) cells. From the visualized scatter plots in Fig. 3c, markers of edge (1) are upregulated in the first lineage (core (1), transitory (1), and edge (1) cells), while markers of edge (2) are upregulated in the second lineage (core (1), transitory (2), and edge (2) cells). For distinct comparison, we fitted two lines to the two lineages in the first scatter plot. The *t*-test between the slopes of the two lines shows the statistical significance of the difference between the two lineages, thereby validating the bifurcate trajectory inferred by spVelo.

Additionally, for sample 4 of the OSCC dataset, spVelo inferred a converged trajectory as shown in Fig. 3b. The clustered results indicated three edge sub-types. Similarly, we visualized the scatter plots of averaged spliced expression and latent time in Fig. 3d. However, upon closer examination, the expression patterns of edge (2) are more consistent with transitory (2) cells, since they transition into edge (3). As a result, we re-annotated edge (2) into transitory (2) and presented the scatter plots after re-annotation in the lower half of Fig. 3e. In the left panel of Fig. 3e, the first lineage (core (1), transitory (1), and edge (1) cells) expresses edge1 markers at a higher level, while the second lineage (transitory (2) and edge (3) cells) expresses at a lower level. The right panel of Fig. 3e shows the opposite for edge3 markers. We further performed K-means clustering with the concatenation of the latent time matrix and gene expression matrix as input and `n_clusters` set as 3. From the visualization in Fig. 3f, previous edge (2) cells should be separated from edge (3) cells. As a result, this updated information aligns the cell classifications with expression dynamics and more accurately reflects the cell type transitions, further supporting spVelo's capability in identifying complex cellular dynamics and refining cell type classifications.

The trajectory plots of all OSCC samples on UMAP embedding are visualized in Additional file 1: Fig. S7 and trajectory plots on spatial coordinates are visualized in Additional file 1: Fig. S9. The trajectory plots of simulated pancreas dataset are visualized in Additional file 1: Fig. S10.

spVelo improves genes' fit and selects biologically important state driver markers

Multiple rate kinetics (MURK) genes are defined as genes with transcriptional boosts [11]. Their expression levels increase rapidly during specific cellular states. Models with simple assumptions may fail to capture their complex dynamics. These upregulating boosts would lead to downregulation estimations, and may further lead to reversed

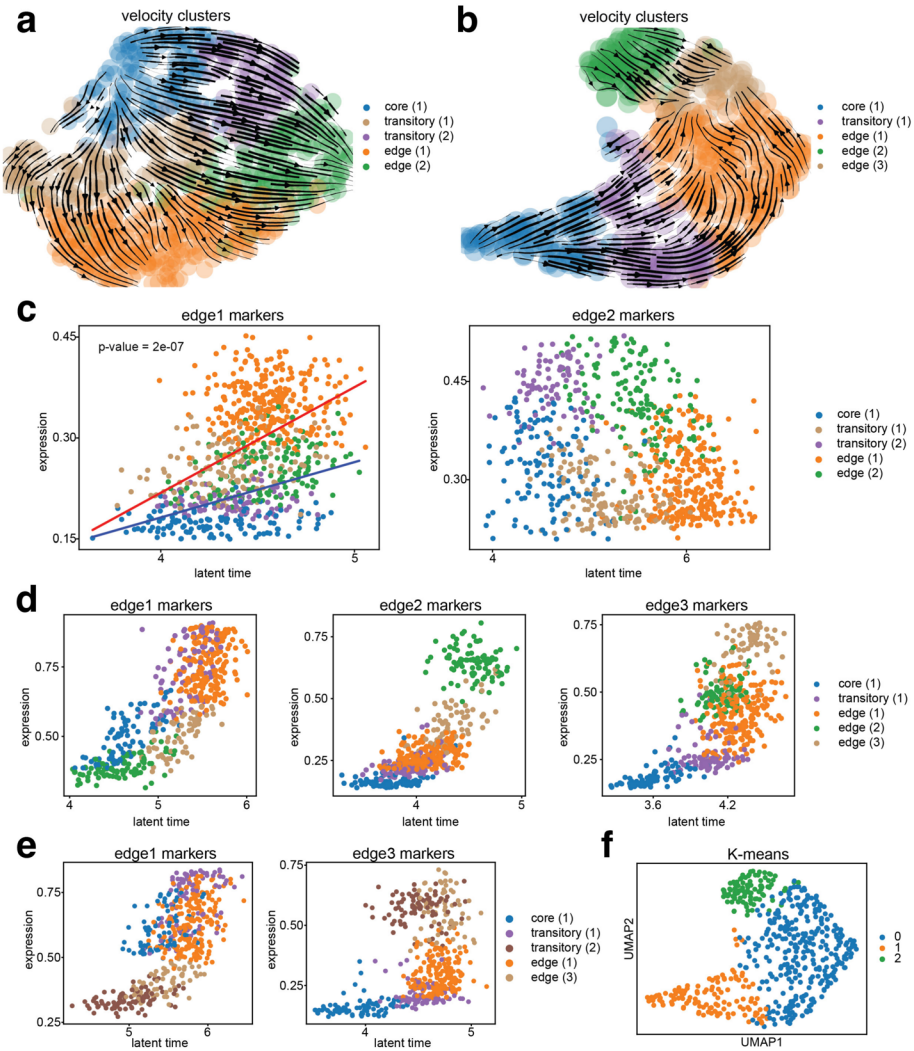

Fig. 3 spVelo discovers complex trajectory patterns. **a** UMAP of bifurcate trajectory in sample 12 from the OSCC dataset. **b** UMAP of converged trajectory in sample 4 from OSCC dataset before re-annotation. **c** Scatter plot of how spliced expression varies along with the latent time inferred by spVelo in sample 12. Each dot represents a cell, and expression and latent time are calculated by averaging the top five markers of edge (1) cells and edge (2) cells. Linear regression lines are fitted for each lineage in the first scatter plot, with a *p* value indicating the significance of slope difference. **d** Scatter plot of how spliced expression varies along with the latent time inferred by spVelo before re-annotation in sample 4. Each dot represents a cell, and expression and latent time are calculated by averaging the top five markers of edge (1), edge (2), and edge (3) cells. **e** Scatter plot of how spliced expression varies along with the latent time inferred by spVelo after re-annotation in sample 4. Each dot represents a cell, and expression and latent time are calculated by averaging the updated top five markers of edge (1) and edge (3) cells. **f** UMAP of K-means clustering

estimations of cellular transitions [12]. Possible solutions include manually removing the MURK genes that violate the model assumption [11]. However, this removal risks the loss of biologically informative genes that are crucial for velocity and trajectory inference.

To address this limitation, we evaluated the capacity of spVelo in inferring the kinetic rates of MURK genes. In Fig. 4a, we visualized phase portraits of five MURK genes from the OSCC dataset, showing the robustness of spVelo in capturing the non-linear

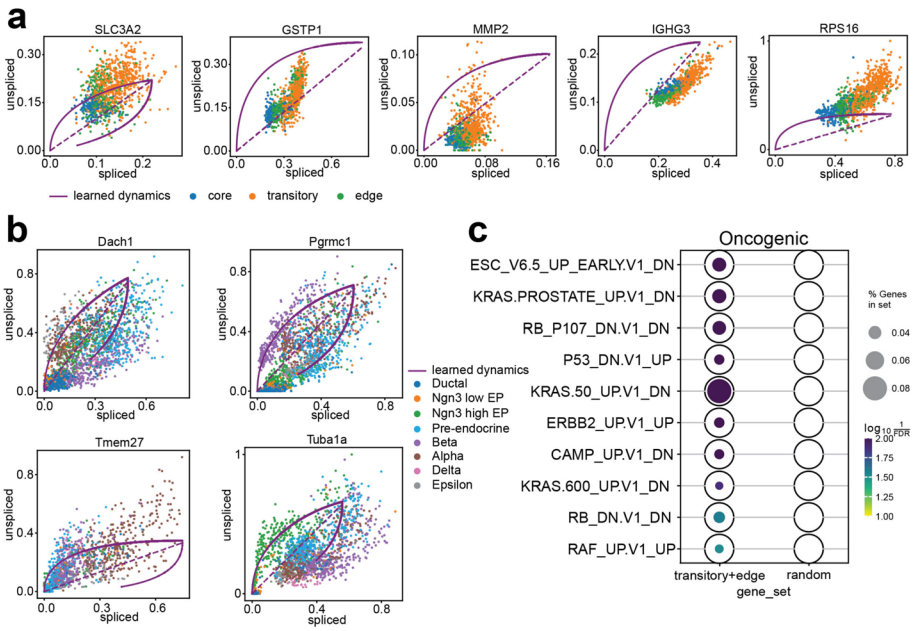

Fig. 4 spVelo fits genes' dynamics well. **a** Phase portraits of five MURK genes from the OSCC dataset. **b** Phase portraits of state driver markers selected from the simulated pancreas dataset. **c** spVelo selects biologically significant state driver markers, verified by gene set enrichment analysis using MSigDB

dynamics and estimating complex kinetics. By fitting the MURK genes, spVelo provides a more accurate representation of the underlying biological process. We also visualized phase portraits of state driver markers selected from the simulated pancreas dataset in Fig. 4b. This further demonstrates spVelo's ability to accurately fit genes' dynamics.

Furthermore, we examined the biological significance of state driver markers selected by spVelo. Based on the velocity estimation, state driver markers are defined as genes pivotal in driving cellular state transitions. Here we utilized a *t*-test on the estimated velocity matrix to select state driver markers and used oncogenic gene sets from MsigDB [26, 27] for gene set enrichment analysis (GSEA). We visualized the GSEA results through a dotplot in Fig. 4c. The first column of the dotplot is state-driver markers selected from transitory and edge cells, and the second column is the same number of randomly selected genes from the dataset, serving as a control group. The dotplot demonstrates that the state driver markers are significantly enriched in oncogenic pathways compared to the random gene set, proving spVelo's ability to select state driver markers that play a crucial role in cancer progression. These state driver markers can potentially serve as targets for therapeutic intervention.

spVelo infers gene regulatory networks by in silico gene deletion

Gene regulatory network (GRN) inference is a popular area since it is critical for understanding transcription. Traditional GRN inference methods largely rely on static gene expression data [38, 39]. However, gene regulation is a highly complex and dynamic process, so traditional methods may be unable to capture the true underlying regulatory relationships and lead to false-positive and false-negative predictions. For example, for the co-expression methods, expression levels of genes may not correlate with those of

their regulating TFs due to the time delay between TF binding and expression accumulation [40, 41]. Here we present spVelo's downstream application in GRN inference. By integrating RNA velocity information in GRN inference, spVelo can predict future gene expression change, revealing true causal relationships. Inspired by [42], we employed an in silico gene deletion approach. We inferred the velocity before and after removing *EGFR*, a gene known for prompting OSCC cell proliferation, metastasis, invasion, and apoptosis resistance [25, 43, 44]. To quantify the impact of *EGFR* deletion, we calculated the gene-wise cosine similarity between the two velocity matrices obtained before and after in silico perturbation. The comparison between EGFR target genes and target genes of other genes is visualized in Fig. 5a. The boxplot reveals that direct *EGFR* targets (defined by the transcription factor target gene sets from MsigDB [26, 27]) are more impacted by the in silico deletion of *EGFR* compared to other target genes. The results suggest that with in silico perturbation, spVelo may identify regulatory relationships and enable the identification of critical genes driving biological processes, thus contributing to understanding the mechanisms underlying disease progression.

spVelo enables temporal cell-cell communication inference

Inspired by CytoSignal and VeloCytoSignal [45], we inferred cell-cell communication (CCC) and temporal CCC using spVelo. Detailed steps of CCC inference can be found

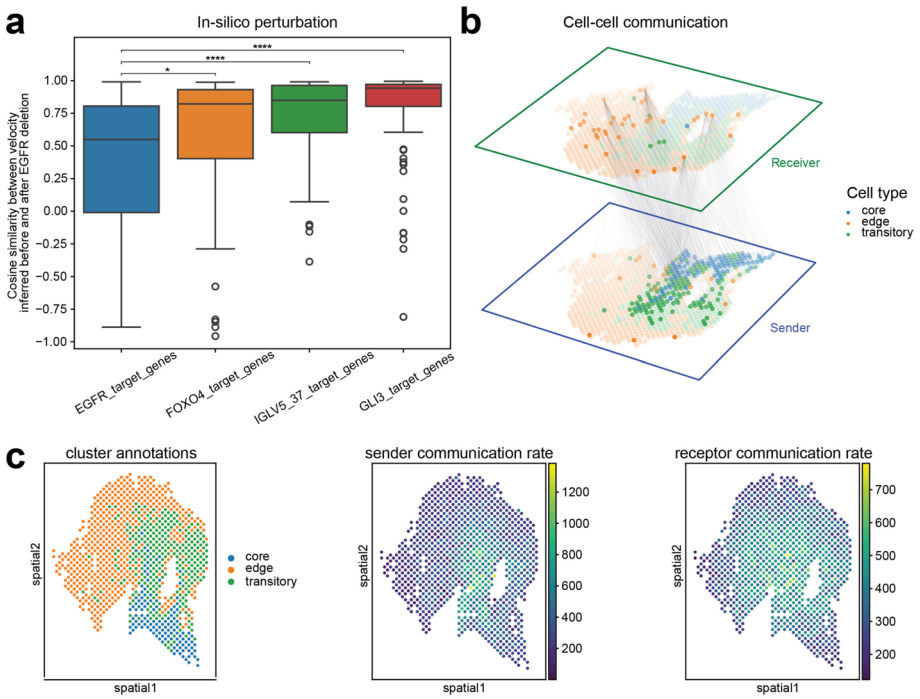

Fig. 5 Downstream applications of spVelo. **a** The Y-axis is the cosine similarity calculated of each gene's velocity before and after in silico perturbation. The boxplot compares the cosine similarity between EGFR target genes and other target genes. **b** 3D plot of inferred cell-cell communication. The opacity of each line is proportional to the cell-cell communication score of the corresponding sender and receiver cells. For clarity and interpretability of the plot, we only use the top 0.1% of the scores to visualize. **c** Temporal cell-cell communication inferred with velocity from spVelo. From left to right: spatial scatter plot of sample 2 from OSCC, scatter plot with sender communication rate, scatter plot with receptor communication rate

in the Methods section. Here we used the ligand-receptor gene pair (*ANXA1*, *EGFR*) for CCC inference. The inferred spot-level CCC is visualized in Fig. 5b, where lines between sender and receptor cells indicate communications between them. From Fig. 5b, few core cells are receptors. Additionally, in receivers, edge cells communicate with greater numbers of senders and higher communication scores; while in senders, core and transitory cells have more outgoing lines. These conclusions are consistent with the cell transition ground truth provided by [25].

Given the significance of CCC in dynamical processes, we quantified spatial-temporal changes in signaling activities to understand the role of CCC in cell state transition. Previous methods use samples sequenced at different time points or estimate pseudo-times from RNA-seq datasets to infer temporal CCC [46]. However, even if we detect the mRNAs for ligands and receptors, that does not guarantee that the cells are communicating at that moment. Proteins might still be missing, inactive, stored, or taking time to build up [47]. By incorporating RNA velocity, spVelo can overcome this problem by predicting expression change in receptor or downstream genes, therefore showing whether cells are actively communicating. Here we inferred temporal CCC and visualized the sender and receptor communication rate in Fig. 5c, and the other results are shown in Additional file 1: Fig. S11. Figure 5c reveals that sender communication rates are higher in core and transitory cells, while receptor communication rates are higher in transitory and edge cells. The enrichment of sender cells in core and transitory state might imply the higher proportion of cancer stem cells, which aligns well with one scRNA-seq study in OSCC samples [48] The higher receptor communication rate in the edge state represents more stable cancer development progress, which also aligns well with research focusing on late-stage cancer [49]. As a result, this result aligns with ground truth, demonstrating that spVelo effectively captures temporal dynamics in cell-cell communications. This helps elucidate the signaling networks in both static and developmental contexts, enabling researchers to better understand the timing of critical cellular interactions.

Conclusions

RNA velocity has emerged as a new approach for inferring cellular trajectory and understanding dynamical processes. Meanwhile, spatially resolved transcriptomics combines gene expression with spatial context, offering insights into cellular architectures. However, existing RNA velocity methods fail to utilize these spatial insights, particularly in large-scale, multi-batch datasets. Here, we introduce spVelo, a novel RNA velocity inference method for multi-batch spatial transcriptomics datasets. Our extensive analysis proves its accuracy and interpretability in velocity and trajectory inference.

Existing methods exhibit several limitations when applied to large-scale spatial datasets. All methods are developed for scRNA-seq and are unable to utilize the spatial information. Among the compared methods, scVelo suffers from strict assumptions and simple modeling, making it unable to capture complex dynamics. This results in oversimplified or inaccurate trajectory inference. On the other hand, veloVI presents a complex VAE-based model with a time-dependent transcriptional rate. However, it fails to infer RNA velocity from multi-batch datasets. LatentVelo is scalable to multi-batch datasets by incorporating

batch information into its model, yet fails to infer coherent velocity between batches and infers an inaccurate trajectory.

spVelo overcomes the above limitations. With its design of combining VAE with GAT, spVelo is capable of leveraging the information from both spatial location and expression data. Additionally, by introducing an MMD penalty between batches, spVelo can infer coherent velocity from multi-batch datasets. Consequently, spVelo more accurately infers velocity and trajectory from large-scale datasets, effectively capturing the underlying dynamics of tissues.

We further provided downstream applications utilizing the velocity inferred by spVelo. Firstly, we demonstrated that the generative modeling of spVelo enables interpretable uncertainty quantification. Secondly, we discovered complex trajectory patterns and further discovered possible cell type refinement. Thirdly, we selected state driver markers and proved their biological significance. Fourthly, we inferred the Gene Regulatory Network utilizing an in silico gene deletion approach. Finally, we inferred temporal cell-cell communications that are consistent with the ground truth. Therefore, RNA velocity inferred by spVelo offers new biological insight into cellular dynamics and exhibits great promise for future explorations.

Methods

Problem definition

In the RNA velocity inference problem, we denote the spliced expression matrix as $S^{N\times G}$ and the unspliced expression matrix as $U^{N\times G}$, where N represents the number of cells and G represents the number of genes. We use $X^{N\times 2}$ to represent the spatial locations of the cells. With these as input, spVelo aims to learn a model M, which can infer the cell-by-gene velocity matrix as $V^{N\times G} = M(S, U, X)$. The model can simultaneously infer cell-gene-specific latent time t_{ng}, transcriptional state k, and kinetic rates including gene-state-specific transcription rate α_{gk}, gene-specific splicing rate β_g, and gene-specific degradation rate γ_g. Here transcriptional state $k \in \{1, 2, 3, 4\}$, where $k = 1$ indicates induction, $k = 2$ indicates the induction steady state, $k = 3$ indicates repression, and $k = 4$ indicates the repression steady state.

spVelo model specification

Following [10] and [16], spVelo assumes that for each gene, cells first go through an induction state where spliced and unspliced expression increases. Then, cells reach an induction steady state, and then at a switching time, the system switches to a repression state where spliced and unspliced expression decreases. Finally, cells reach a repression steady state with no expression.

By solving the ordinary differential equations [10], the estimated unspliced and spliced abundance at time t_{ng} for cell n and gene g is defined as:

$$\bar{u}^{(g)}(t_{ng}, k) := u^0_{gk} e^{-\beta_g (t_{ng} - t^0_{gk})} + \frac{\alpha_{gk}}{\beta_g}\left(1 - e^{-\beta_g (t_{ng} - t^0_{gk})}\right), \tag{1}$$

$$\bar{s}^{(g)}(t_{ng}, k) := s^0_{gk} e^{-\gamma_g \tau} + \frac{\alpha_{gk}}{\gamma_g}\left(1 - e^{-\gamma_g (t_{ng} - t^0_{gk})}\right) + \frac{\alpha_{gk} - \beta_g u^0_{gk}}{\gamma_g - \beta_g}\left(e^{-\gamma_g (t_{ng} - t^0_{gk})} - e^{-\beta_g (t_{ng} - t^0_{gk})}\right), \tag{2}$$

where t_{gk}^0 denotes the initial time of the system in state k. u_{gk}^0 and s_{ng}^0 denotes the estimated initial unspliced and spliced expression of gene g in state k, i.e., $u_{gk}^0 = \bar{u}^{(g)}\left(t_{gk}^0, k\right)$ and $s_{gk}^0 = \bar{s}^{(g)}\left(t_{gk}^0, k\right)$.

Transcription rate α is assumed to be time-dependent with parameters $\alpha_0, \alpha_1, \lambda_\alpha$:

$$\alpha^{(k)}(t) = \begin{cases} \alpha_1 - (\alpha_1 - \alpha_0)e^{-\lambda_\alpha t}, & k \in \{1,2\}, \\ 0, & k \in \{3,4\}. \end{cases} \tag{3}$$

For future conciseness, we still write the gene-state-specific transcription rate $\alpha_g^{(k)}(t)$ as α_{gk}.

For $k = 1$ (induction state), we have $u_{g1}^0 = 0$, $s_{g1}^0 = 0$, $\alpha_{g1} > 0$, and $t_{g1}^0 = 0$ by definition. Thus (6) and (7) can be simplified into

$$\bar{u}^{(g)}(t_{ng}, k=1) := \frac{\alpha_{g1}}{\beta_g}\left(1 - e^{-\beta_g t_{ng}}\right), \tag{4}$$

$$\bar{s}^{(g)}(t_{ng}, k=1) := \frac{\alpha_{g1}}{\gamma_g}\left(1 - e^{-\gamma_g t_{ng}}\right) + \frac{\alpha_{g1}}{\gamma_g - \beta_g}\left(e^{-\gamma_g t_{ng}} - e^{-\beta_g t_{ng}}\right). \tag{5}$$

For $k = 2$ (induction steady state), the unspliced and spliced expression is defined as the limit of the induction state as time approaches ∞:

$$\bar{u}^{(g)}(t_{ng}, k=2) := \lim_{t_{ng}\to\infty} \bar{u}^{(g)}(t_{ng}, k=1) = \frac{\alpha_{g1}}{\beta_g}, \tag{6}$$

$$\bar{s}^{(g)}(t_{ng}, k=2) := \lim_{t_{ng}\to\infty} \bar{s}^{(g)}(t_{ng}, k=1) = \frac{\alpha_{g1}}{\gamma_g}. \tag{7}$$

For $k = 3$ (repression state), we have $\alpha_{g3} = 0$ and $t_{g1}^0 = t_g^s$, where t_g^s is the gene-specific switching time from the induction phase to the repression phase. Thus (6) and (7) can be expressed as

$$\bar{u}^{(g)}(t_{ng}, k=3) := u_{g3}^0 e^{-\beta_g (t_{ng} - t_{g3}^0)}, \tag{8}$$

$$\bar{s}^{(g)}(t_{ng}, k=3) := s_{g3}^0 e^{-\gamma_g (t_{ng} - t_{g3}^0)} - \frac{\beta_g u_{g3}^0}{\gamma_g - \beta_g}\left(e^{-\gamma_g (t_{ng} - t_{g3}^0)} - e^{-\beta_g (t_{ng} - t_{g3}^0)}\right). \tag{9}$$

Similarly, $k = 4$ (repression steady state) is defined as the limit of the repression state, resulting in

$$\bar{u}^{(g)}(t_{ng}, k=4) := 0, \tag{10}$$

$$\bar{s}^{(g)}(t_{ng}, k=4) := 0. \tag{11}$$

spVelo generative process

The generative modeling of spVelo combines a Variational AutoEncoder (VAE) [21] inspired by [16], with a Graph Attention Network (GAT) [22]. We explored several types of Graph Neural Networks (GNNs) for modeling cell-cell relationships, including GCN [50] and GraphSAGE [51]. Among them, GAT consistently showed the best performance, which motivated us to adopt GAT as the backbone of our model. More details can be found in Additional file 1: Text S1.2, and comparison results can be found in Additional file 1: Fig. S12.

We assume the following generative process to model the underlying dynamics of the unspliced expression u_{ng} and spliced expression s_{ng}:

For each cell n and gene g, we use a low-dimensional latent variable z_n to summarize the latent state of each cell (default $d = 10$). z_n is the sum of the latent space from VAE and GAT, modeling both expression data and spatial location. Let

$$z_n^{VAE} \sim \text{Normal}(0, I_d), \tag{12}$$

$$z_n^{GAT} = \text{GAT}(z_n^{VAE}, e), \tag{13}$$

$$z_n = z_n^{VAE} + z_n^{GAT}, \tag{14}$$

where e denotes the edges input to GAT. In GAT modeling, z_n^{GAT} is constructed based on a graph structure where edges represent relationships between cells. The edges are composed of two parts: The first part of the edges is calculated using k Nearest Neighbors (kNN) on the spatial coordinates. We compute the edges in each batch and concatenate across all batches. The second part of the edges is calculated across different batches using Mutual Nearest Neighbors (MNN) on the expression data. The distance of MNN is defined as the optimal transport (OT) matrix, quantifying the correspondence between samples in different batches [52]. The metric cost matrix in the OT problem is calculated as the Euclidean distance between batches. By combining the two parts of the edges, the GAT module effectively captures spatial information together with relationships between batches. The number of neighbors for both parts is set as 15. More details of tuning weights between spatial and mnn edges can be found in Additional file 1: Text S1.3, and comparison results can be found in Additional file 1: Fig. S13.

We then use a Dirichlet distribution to model state assignment probability π_{ng}. The settings are based on veloVI. We further performed ablation studies for Dirichlet prior distribution parameters in Additional file 1: Text S1.4 and visualized results in Additional file 1: Fig. S14. The state k_{ng} is then defined as the state with the highest state assignment probability.

$$\pi_{ng} \sim \text{Dirichlet}(0.25, 0.25, 0.25, 0.25), \tag{15}$$

$$k_{ng} \sim \text{Categorical}(\pi_{ng}). \tag{16}$$

Latent time t_{ng} is modeled as a state-specific function of latent state z_n:

$$\rho_{ng}^{(k)} = [h_k(z_n)]_g, \tag{17}$$

$$t_{ng}^{(k)} = \begin{cases} \rho_{ng}^{(1)} t_g^s & \text{if } k = 1, \\ (t_{max} - t_g^s) \times \rho_{ng}^{(3)} + t_g^s & \text{if } k = 3, \end{cases} \tag{18}$$

where $t_{max} := 20$ fixes the time scale across genes. $h_k : \mathbb{R}^d \to (0,1)^G$ is parameterized as a state-specific fully connected neural network.

Finally, we assume the observed expression data are sampled from normal distributions as

$$u_{ng} \sim \text{Normal}\left(\bar{u}^{(g)}(t_{ng}^{(k_{ng})}, k_{ng}), (c_k \sigma_g^u)^2\right), \tag{19}$$

$$s_{ng} \sim \text{Normal}\left(\bar{s}^{(g)}(t_{ng}^{(k_{ng})}, k_{ng}), (c_k \sigma_g^s)^2\right), \tag{20}$$

where c_k is a state-dependent scaling factor on the variance. As default, $c_k = 1$ for $k = 1, 2, 3$ except for $c_4 = 0.1$ in the repression steady state.

spVelo posterior inference

Variational posterior Let θ be the set of parameters including kinetic rates (α, β, γ), switching time t^s, and neural network parameters. We use variational inference [21] to approximate the posterior distribution. The posterior distribution is posited as

$$q_\phi(z, \pi \mid u, s) := \prod_{n=1}^{N} q_\phi(z_n \mid u_n, s_n) \prod_{g=1}^{G} q_\phi(\pi_{ng} \mid z_n), \tag{21}$$

where dependencies are specified using neural networks with parameter set ϕ.

Integrating over the choice of transcriptional state k_{ng}, the likelihoods for spliced and unspliced transcript abundances are Gaussian mixture models:

$$p_\theta(u_{ng} \mid z_n, \pi_n) = \sum_{k_{ng} \in \{1,2,3,4\}} \pi_{ngk_{ng}} \text{Normal}\left(\bar{u}^{(g)}\left(t_{tng}^{(k_{ng})}, k_{ng}\right), (c_k \sigma_g^u)^2\right) \tag{22}$$

$$p_\theta(s_{ng} \mid z_n, \pi_n) = \sum_{k_{ng} \in \{1,2,3,4\}} \pi_{ngk_{ng}} \text{Normal}\left(\bar{s}^{(g)}\left(t_{tng}^{(k_{ng})}, k_{ng}\right), (c_k \sigma_g^s)^2\right) \tag{23}$$

Optimization The objective function is composed of three terms

$$\mathcal{L}_{\text{velo}}(\theta, \phi; u, s) = \mathcal{L}_{\text{elbo}}(\theta, \phi; u, s) + \lambda \mathcal{L}_{\text{switch}}(\theta; u, s) + \lambda \mathcal{L}_{\text{batch}}(z), \tag{24}$$

where $\mathcal{L}_{\text{elbo}}$ is the negative evidence lower bound [53] of $logp_\theta(u, s)$, $\mathcal{L}_{\text{switch}}$ is a penalty that regularizes the location of transcriptional switch in the phase portrait, and $\mathcal{L}_{\text{batch}}$ is an MMD penalty that regularizes the latent space between different batches. As default, the penalty weight $\lambda = 2$. In more detail, we denote b_1, b_2 as a pair of different batch IDs, z_b as the latent space of batch b, and u^* and s^* as the median unspliced and spliced expression for each gene,

$$\begin{aligned}\mathcal{L}_{\text{elbo}}(\theta,\phi;u,s) = \sum_n & -\mathbb{E}_{q_\phi(z_n,\pi_n|u_n,s_n)}[\log p_\theta(u_n,s_n \mid z_n,\pi_n)] \\ & + \mathrm{KL}\big(q_\phi(z_n \mid u_n,s_n) \,\|\, p(z)\big) \\ & + \mathbb{E}_{q_\phi(z_n|u_n,s_n)}\left[\sum_g \mathrm{KL}\big(q_\phi(\pi_{ng} \mid z_n) \,\|\, p(\pi_{ng})\big)\right],\end{aligned} \tag{25}$$

$$\mathcal{L}_{\text{switch}}(\theta;u,s) = \sum_g \left(\left(u^0_{g3} - u^*_g\right)^2 + \left(s^0_{g3} - s^*_g\right)^2\right), \tag{26}$$

$$\mathcal{L}_{\text{batch}}(z) = \sum_{b_1,b_2} \mathrm{MMD}^2(z_{b_1}, z_{b_2}), \tag{27}$$

$$\mathrm{MMD}^2(U,V) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{i'=1}^{n} k(u_i,u_{i'}) - \frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} k(u_i,v_j) + \frac{1}{m^2}\sum_{j=1}^{m}\sum_{j'=1}^{m} k(v_j,v_{j'}). \tag{28}$$

Here $k(x, y)$ denotes a Gaussian kernel, i.e., $k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$, where σ is a bandwidth parameter and $\|x - y\|$ is the Euclidean distance between x and y.

To optimize $\mathcal{L}_{\text{velo}}$, we use stochastic gradients [21] and Adam optimizer with weight decay [54]. We set the number of epochs as 2000. We present our results of hyper-parameter tuning in Additional file 1: Fig. S15.

Velocity inference After fitting the parameters, the cell-gene-specific state assignment is calculated as the posterior mean:

$$\tilde{\pi}_{ng} = \mathbb{E}_{q_\phi(z_n|u_n,s_n)}\left[\mathbb{E}_{q_\phi(\pi_{ng}|z_n)}\left[\pi_{ng}\right]\right]. \tag{29}$$

The cell-gene-specific latent time is calculated as

$$\tilde{t}^{(k_{ng})}_{ng} = \mathbb{E}_{q_\phi(z_n|u_n,s_n)}\left[\mathbb{E}_{q_\phi(\pi_{ng}|z_n)}\left[t^{(k_{ng})}_{ng}\right]\right]. \tag{30}$$

RNA velocity is calculated as a function of the variational posterior

$$v^{(g)}\left(t^{(k)},k\right) := \left.\frac{d\bar{s}^{(g)}(t,k)}{dt}\right|_{t^{(k)}} = \beta_g \bar{u}^{(g)}\left(t^{(k)},k\right) - \gamma_g \bar{s}^{(g)}\left(t^{(k)},k\right). \tag{31}$$

Uncertainty quantification

Uncertainty of the latent state is calculated as the differential entropy of the latent space:

$$h(z) = \frac{1}{2}\log\left((2\pi e)^d \det(\Sigma)\right), \tag{32}$$

where d is the dimension (default as 10) and Σ is the variance matrix of the latent space.

Temporal cell-cell communication inference

The spatial interaction score is defined as the co-expression of ligand and receptor genes within close spatial proximity. Here we select a ligand-receptor gene pair from OmniPath [55] and denote the spliced expression matrix as S, and denote a pair of ligand and receptor genes as l and r.

For cells i and j, we calculate the LRscore as:

$$LRscore(i,j) = S_{il} \times S_{jr} \times \mathbb{I}\{d_{ij} < q\}. \tag{33}$$

For cell types A and B, we calculate the LRscore as:

$$LRscore(A,B) = \sum_{i \in C_A} \sum_{j \in C_B} S_{il} \times S_{jr} \times \mathbb{I}\{d_{ij} < q\}, \tag{34}$$

where C_A refers to all cells in cell type A, and S_{il} refers to the expression value of gene l in cell i. In the indicator function, d_{ij} refers to the Euclidean distance between the spatial location of cell i and cell j, and q refers to a user-defined threshold, set as 30. After calculating scores between cell types, we randomly permuted cell types 50 times and performed False Discovery Rate (FDR) correction.

The spatial-temporal interaction score is defined as the time derivative of LRscore and calculated as follows:

$$\begin{aligned} LRvelo(i,j) =& \frac{d\, LRscore(i,j)}{dt} = \left[S_{il} \times \frac{d\, S_{jr}}{dt} + \frac{d\, S_{il}}{dt} \times S_{jr} \right] \times \mathbb{I}\{d_{ij} < q\} \\ =& \left(S_{il} \times V_{jr} + V_{il} \times S_{jr} \right) \times \mathbb{I}\{d_{ij} < q\}, \end{aligned} \tag{35}$$

where V refers to the inferred velocity matrix.

Metrics explanations

To evaluate the performance of inferred velocity, we calculated three different types of scores, inspired by VeloAE [56]. For each pair of cell types (A, B), the scores are calculated for the boundary scores, referring to cells of cell type A with cell type B in the neighborhood, i.e., $C_{A \to B} = \{c \in C_A \mid \exists c' \in C_B \cap N(c)\}$. Here C_A denotes all the cells of cell type A and $N(c)$ denotes the neighbor cells of c.

1. Confidence score: Confidence score for cell c from cell type A with regard to cell type B is defined as

$$Confidence(c) = \frac{1}{|c' \in C_B \cap N(c)|} \sum_{c' \in C_B \cap N(c)} \frac{V_c \cdot V_{c'}}{\|V_c\| \cdot \|V_{c'}\|}, \tag{36}$$

where V_c is the velocity vector of cell c. This is calculated using `scv.tl.velocity_confidence`. Then, the confidence score for cell type A is calculated as the average of $Confidence(c)$ for all $c \in C_{A \to B}$. It summarizes the consistency of the inferred velocity, and a higher confidence score represents better consistency.

2. Transition score: Transition score for cell c from cell type A with regard to cell type B is defined as

$$Transition(c) = \frac{1}{|c' \in C_B \cap N(c)|} \sum_{c' \in C_B \cap N(c)} \tilde{\pi}_{cc'}. \tag{37}$$

Here $\tilde{\pi}_{cc'}$ denotes the cell-to-cell transition probabilities calculated from the velocity graph $\pi_{cc'}$ with row-normalization z_c and kernel width σ. This is calculated using `scv.tl.velocity_graph` and `scv.utils.get_transition_matrix`.

$$\pi_{cc'} = \cos \angle (S_{c'} - S_c, V_c) = \frac{(S_{c'} - S_c) \cdot V_c}{\|S_{c'} - S_c\| \|V_c\|}, \tag{38}$$

$$\tilde{\pi}_{cc'} = \frac{1}{z_c} \exp(\pi_{cc'}/\sigma), \tag{39}$$

where S_c refers to the spliced gene expression of cell c. Transition score for cell type A is calculated as the average of $Transition(c)$ for all $c \in C_{A \to B}$, measuring how well the corresponding change in gene expression matches the predicted change. A higher transition score represents a better match.

3. Direction score: Direction score for cell c from cell type A with regard to cell type B is defined as

$$Dir(c) = \frac{1}{|c' \in C_B \cap N(c)|} \sum_{c' \in C_B \cap N(c)} \frac{(x_{c'} - x_c) \cdot \tilde{v}_c}{\|x_{c'} - x_c\| \|\tilde{v}_c\|}. \tag{40}$$

Here x_c and $x_{c'}$ are vectors representing cells c and c' in a low-dimensional Principal Component Analysis (PCA) space via [57] (number of principal components default as 30). $x_{c'} - x_c$ is the displacement in this space, and $\tilde{v}_c$ is the projection of velocity into PCA space, calculated using `scv.tl.velocity_embedding`. Denoting $\tilde{\pi}_{cc'}$ as the transition probability matrix, we have

$$\tilde{v}_c = \mathbb{E}_{\tilde{\pi}_c}[\frac{x_{c'} - x_c}{\|x_{c'} - x_c\|}] = \sum_{c' \neq c} \left(\tilde{\pi}_{cc'} - \frac{1}{n}\right) \frac{x_{c'} - x_c}{\|x_{c'} - x_c\|}. \tag{41}$$

Direction score for cell type A is calculated as the average of $Dir(c)$ for all $c \in C_{A \to B}$, measuring how well the corresponding change in PCA embedding matches the predicted change. A higher direction score represents a better match.

With ground truth cell type transition information as input, the confidence scores are calculated as the average score of all correct cell type transition pairs, while transition scores and direction scores are calculated by averaging scores of correct cell type transition pairs while incorporating a penalty for incorrect transitions by using their negated scores. The correct cell type pairs are defined as a pair of cell types with known transition relationships from the first cell type to the second. For the simulated pancreas dataset, the list of cell type pairs is defined as [('Ductal', 'Ngn3 low EP'), ('Ngn3 low EP', 'Ngn3 high EP'), ('Ngn3 high EP', 'Pre-endocrine'), ('Pre-endocrine', 'Delta'), ('Pre-endocrine', 'Beta'), ('Pre-endocrine', 'Epsilon'), ('Pre-endocrine', 'Alpha')], as defined in [10, 24]. For the OSCC dataset, the list of cell type pairs is defined as [('core', 'transitory'), ('transitory', 'edge'), ('core', 'edge')], as defined in [25]. More discussions of the metrics can be found in Additional file 1: Text S3.

From the equations, the three scores are all calculated based on local neighborhoods. We compute the kNN graph, spatial graph, and MNN graph respectively, incorporating different information in model comparison. As default, the neighbor size is set as 30.

Inspired by LatentVelo [14], we also measure the cosine similarity of MNN cells in different batches to evaluate batch effect correction of RNA velocity. Let C_b be all the cells in batch b and $N^{MNN}(c)$ be MNN of cell c, the velocity coherence score for cell c is defined as:

$$Coh(c) = \frac{1}{|B|(|B|-1)} \sum_{b_1=1}^{B} \sum_{b_2 \neq b_1} \frac{v_{b_1} \cdot v_{b_2}}{\|v_{b_1}\| \|v_{b_2}\|}, \tag{42}$$

$$v_b = \frac{1}{|c' \in C_b \cap N^{MNN}(c)|} \sum_{c' \in C_b \cap N^{MNN}(c)} V_{c'}, \tag{43}$$

where B denotes the set of batches in the dataset and (b_1, b_2) denotes a pair of different batch IDs. Then, the final velocity coherence score is calculated as the average of 100 randomly selected cells.

Baseline model explanations

In the model comparison process, we consider eight baseline methods (settings) in total for comparison, including standard and annotated mode of LatentVelo, stochastic and dynamical mode of scVelo, veloVI, and scGen-corrected scVelo and veloVI. The order of these methods (settings) is random.

LatentVelo [14] uses a VAE that embeds unspliced and spliced abundances of RNA into the latent space, and dynamics on the latent space are described as a neural ODE. By learning a shared latent space for multiple batches, LatentVelo enables batch effect correction from a dynamic view. The annotated mode of LatentVelo incorporates cell type information by modifying the prior.

The stochastic mode of scVelo [10] treats transcription, splicing, and degradation as probabilistic events and approximates the Markov process using moment equations. By using both first- and second-order moments, scVelo (stochastic) can utilize both relationships and covariation between unspliced and spliced mRNA abundances. The dynamical mode of scVelo solves the ODEs with a likelihood-based expectation-maximization framework, iteratively estimating the parameters of kinetic rates, transcriptional state, and cell-internal latent time.

veloVI [16] treats unspliced and spliced abundances of RNA for each gene as a function of kinetic parameters, latent time, and latent transcriptional state. It further treats latent time as tied via a low-dimensional latent variable. veloVI uses a VAE architecture and outputs a posterior distribution over estimated velocity.

For batch effect correction settings, since current RNA velocity methods require cell-by-gene spliced and unspliced counts as input, only batch effect correction methods that return a corrected and reconstructed gene matrix can be used. As a result, we used scGen [29] for batch effect correction, as recommended by scIB [58].

In the scGen-corrected models, we followed the approach taken by [14, 59]. Since we need to simultaneously correct spliced and unspliced counts, we perform batch

effect correction on the sum of these counts. Denote the spliced and unspliced counts as S and U, we define the sum matrix as $M = S + U$, and the ratio matrix as $R = \frac{S}{S+U}$. scGen batch effect correction is performed on log-normalized M with the default settings, and we get the corrected matrix $\tilde{M}$. To recover corrected spliced and unspliced expression, we multiply $\tilde{M}$ with R or $1 - R$.

$$S_{corrected} = \tilde{M} \times R \tag{44}$$

$$U_{corrected} = \tilde{M} \times (1 - R) \tag{45}$$

Then, RNA velocity is estimated as before.

Experiment design

For the simulated dataset, we followed the tutorial from scCube [28] and generated random spatial patterns for cell types with a reference-free strategy. We also considered scDesign3 [60] for the simulation. Extra analysis of data simulation can be found in Additional file 1: Text S4 and Additional file 1: Fig. S16. We used the scRNA-seq pancreas dataset [24] for this simulation. For the real OSCC dataset, we filtered all noncancer (nc) cells, following the preprocessing step in [25].

For both the simulated pancreas dataset and the real OSCC dataset, we followed the pre-processing guidelines from scVelo [10]. We normalized the count matrices to the median of total molecules across cells and filtered genes with less than 20 expressed counts commonly for spliced and unspliced mRNA, followed by log-transforming the data and selecting the top 2000 highly variable genes. Then, we calculated a nearest neighbor graph (with 30 neighbors) based on Euclidean distances in principal component analysis space (with 30 principal components) on spliced logcounts. We computed first- and second-order moments (means and uncentered variances) for each cell across its 30 nearest neighbors.

Following [16], we min-max scaled the unspliced and spliced expression to the unit interval and applied the steady-state scVelo model. Finally, we filtered the genes with negative steady-state ratio and R^2 statistic below a user-defined threshold (default as 0.2). We further performed ablation studies for the R^2 threshold in Additional file 1: Text S1.5 and visualized results in Additional file 1: Fig. S17. Then, the remaining genes are used for velocity inference.

In model comparison, we followed the tutorials of all methods. To prove the scalability of spVelo on larger datasets, we further performed simulation for two different conditions, including large number of slices and large number of cells per slice. More details can be found in Additional file 1: Text S5 and results are visualized in Additional file 1: Fig. S18.

Supplementary information

The online version contains supplementary material available at https://doi.org/10.1186/s13059-025-03701-8.

Additional file 1. Contains supplementary texts, tables and figures.

Additional file 2. Contains information and download links of datasets, and Supplementary tables.

Acknowledgements
We thank Hanshu Yu for providing suggestions from a biological perspective.

Peer review information
Claudia Feng was the primary editor of this article and managed its editorial process and peer review in collaboration with the rest of the editorial team. The peer-review history is available in the online version of this article.

Authors'contributions
T.L. and W.L. designed this study. W.L. and T.L. designed the model. W.L. ran all the experiments. W.L., T.L., L.X., and H.Z. wrote the manuscript. L.X. and H.Z. supervised this work. All authors read and approved the final manuscript.

Funding
Supported in part by NIH grants U01 HG013840, U24 HG012108, P50 CA196530, and R01 GM152812.

Data availability
We summarize the sources and statistics of all datasets we used in Additional file 2. All the public datasets can be accessed based on the links in this file. The OSCC dataset is available at GEO with the accession number GSE208253 (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE208253) [61]. The pancreas dataset used for simulation is available at GEO with accession number GSE132188 (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE132188) [62]. We downloaded this dataset from https://scvelo.readthedocs.io/en/stable/VelocityBasics.html. The stereo-seq mousebrain dataset is downloaded from the Spateo package (https://github.com/aristoteleo/spateo-tutorials, https://www.dropbox.com/s/wxgkim87uhpaz1c/mousebrain_bin60_clustered.h5ad?dl=1) [63].
We relied on Yale High-performance Computing Center (YCRC) and utilized one NVIDIA A5000 GPU with up to 30 GB of RAM for model training. The source codes of spVelo and the simulation code for the pancreas dataset can be found at Github (https://github.com/VivLon/spVelo) [64] and Zenodo (https://zenodo.org/records/15343924) [65]. We follow the MIT license for usage. The simulated pancreas dataset has also been deposited at Zenodo.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 21 February 2025 Accepted: 16 July 2025
Published online: 11 August 2025

References

1. Tanay A, Regev A. Scaling single-cell genomics from phenomenology to mechanism. Nature. 2017;541(7637):331–8.
2. Moncada R, Barkley D, Wagner F, Chiodin M, Devlin JC, Baron M, et al. Integrating microarray-based spatial transcriptomics and single-cell RNA-seq reveals tissue architecture in pancreatic ductal adenocarcinomas. Nat Biotechnol. 2020;38(3):333–42.
3. Alon S, Goodwin DR, Sinha A, Wassie AT, Chen F, Daugharthy ER, et al. Expansion sequencing: spatially precise in situ transcriptomics in intact biological systems. Science. 2021;371(6528): eaax2656.
4. Trapnell C, Cacchiarelli D, Grimsby J, Pokharel P, Li S, Morse M, et al. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. Nat Biotechnol. 2014;32(4):381–6.
5. Cannoodt R, Saelens W, Saeys Y. Computational methods for trajectory inference from single-cell transcriptomics. Eur J Immunol. 2016;46(11):2496–506.
6. Saelens W, Cannoodt R, Todorov H, Saeys Y. A comparison of single-cell trajectory inference methods. Nat Biotechnol. 2019;37(5):547–54.
7. Setty M, Kiseliovas V, Levine J, Gayoso A, Mazutis L, Pe'Er D. Characterization of cell fate probabilities in single-cell data with Palantir. Nat Biotechnol. 2019;37(4):451–60.
8. 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.
9. Lange M, Bergen V, Klein M, Setty M, Reuter B, Bakhti M, et al. Cell rank for directed single-cell fate mapping. Nat Methods. 2022;19(2):159–70.
10. 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.
11. Barile M, Imaz-Rosshandler I, Inzani I, Ghazanfar S, Nichols J, Marioni JC, et al. Coordinated changes in gene expression kinetics underlie both mouse and human erythroid maturation. Genome Biol. 2021;22:1–22.
12. Bergen V, Soldatov RA, Kharchenko PV, Theis FJ. RNA velocity–current challenges and future perspectives. Mol Syst Biol. 2021;17(8):e10282.
13. Gao M, Qiao C, Huang Y. Unitvelo: temporally unified RNA velocity reinforces single-cell trajectory inference. Nat Commun. 2022;13(1): 6586.

14. Farrell S, Mani M, Goyal S. Inferring single-cell transcriptomic dynamics with structured latent gene expression dynamics. Cell Rep Methods. 2023;3(9):100581.
15. Chen RT, Rubanova Y, Bettencourt J, Duvenaud DK. Neural ordinary differential equations. In Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18). Red Hook: Curran Associates Inc.; 2018. p. 6572–83.
16. Gayoso A, Weiler P, Lotfollahi M, Klein D, Hong J, Streets A, et al. Deep generative modeling of transcriptional dynamics for RNA velocity analysis in single cells. Nat Methods. 2024;21(1):50–9.
17. Clark BS, Stein-O'Brien GL, Shiau F, Cannon GH, Davis-Marcisak E, Sherman T, et al. Single-cell RNA-seq analysis of retinal development identifies NFI factors as regulating mitotic exit and late-born cell specification. Neuron. 2019;102(6):1111–26.
18. Mahdessian D, Cesnik AJ, Gnann C, Danielsson F, Stenström L, Arif M, et al. Spatiotemporal dissection of the cell cycle with single-cell proteogenomics. Nature. 2021;590(7847):649–54.
19. Gorin G, Fang M, Chari T, Pachter L. RNA velocity unraveled. PLoS Comput Biol. 2022;18(9): e1010492.
20. Fang S, Chen B, Zhang Y, Sun H, Liu L, Liu S, et al. Computational approaches and challenges in spatial transcriptomics. Genomics Proteomics Bioinforma. 2023;21(1):24–47.
21. Kingma DP. Auto-encoding variational bayes. 2013. arXiv preprint arXiv:1312.6114.
22. Velickovic P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y, et al. Graph attention networks. Stat. 2017;1050(20):10–48550.
23. Gretton A, Borgwardt K, Rasch M, Schölkopf B, Smola A. A kernel method for the two-sample-problem. In Proceedings of the 20th International Conference on Neural Information Processing Systems (NIPS'06). Cambridge: MIT Press; 2006. p. 513–20.
24. Bastidas-Ponce A, Tritschler S, Dony L, Scheibner K, Tarquis-Medina M, Salinno C, et al. Comprehensive single cell mRNA profiling reveals a detailed roadmap for pancreatic endocrinogenesis. Development. 2019;146(12): dev173849.
25. Arora R, Cao C, Kumar M, Sinha S, Chanda A, McNeil R, et al. Spatial transcriptomics reveals distinct and conserved tumor core and edge architectures that predict survival and targeted therapy response. Nat Commun. 2023;14(1): 5029.
26. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci. 2005;102(43):15545–50.
27. Liberzon A, Birger C, Thorvaldsdóttir H, Ghandi M, Mesirov JP, Tamayo P. The molecular signatures database hallmark gene set collection. Cell Syst. 2015;1(6):417–25.
28. Qian J, Bao H, Shao X, Fang Y, Liao J, Chen Z, et al. Simulating multiple variability in spatially resolved transcriptomics with scCube. Nat Commun. 2024;15(1): 5021.
29. Lotfollahi M, Wolf FA, Theis FJ. Scgen predicts single-cell perturbation responses. Nat Methods. 2019;16(8):715–21.
30. Haghverdi L, Büttner M, Wolf FA, Buettner F, Theis FJ. Diffusion pseudotime robustly reconstructs lineage branching. Nat Methods. 2016;13(10):845–8.
31. Qiu X, Zhu DY, Lu Y, Yao J, Jing Z, Min KH, et al. Spatiotemporal modeling of molecular holograms. Cell. 2024;187(26):7351–73.
32. Zhou P, Bocci F, Li T, Nie Q. Spatial transition tensor of single cells. Nat Methods. 2024;21(6):1053–62.
33. Abdelaal T, Grossouw LM, Pasterkamp RJ, Lelieveldt BP, Reinders MJ, Mahfouz A. SIRV: spatial inference of RNA velocity at the single-cell resolution. NAR Genom Bioinform. 2024;6(3): lqae100.
34. Gu Y, Blaauw D, Welch JD. Bayesian inference of RNA velocity from multi-lineage single-cell data. bioRxiv. 2022:2022–07.
35. Garbaczewski P. Differential entropy and dynamics of uncertainty. J Stat Phys. 2006;123:315–55.
36. Foroutan M, Cursons J, Hediyeh-Zadeh S, Thompson EW, Davis MJ. A transcriptional program for detecting TGFβ-induced EMT in cancer. Mol Cancer Res. 2017;15(5):619–31.
37. Traag VA, Waltman L, Van Eck NJ. From Louvain to Leiden: guaranteeing well-connected communities. Sci Rep. 2019;9(1):1–12.
38. Moerman T, Aibar Santos S, Bravo González-Blas C, Simm J, Moreau Y, Aerts J, et al. GRNBoost2 and Arboreto: efficient and scalable inference of gene regulatory networks. Bioinformatics. 2019;35(12):2159–61.
39. Bravo González-Blas C, De Winter S, Hulselmans G, Hecker N, Matetovici I, Christiaens V, et al. Scenic+: single-cell multiomic inference of enhancers and gene regulatory networks. Nat Methods. 2023;20(9):1355–67.
40. Zhu Z, Pilpel Y, Church GM. Computational identification of transcription factor binding sites via a transcription-factor-centric clustering (TFCC) algorithm. J Mol Biol. 2002;318(1):71–81.
41. Popp AP, Hettich J, Gebhardt JCM. Altering transcription factor binding reveals comprehensive transcriptional kinetics of a basic gene. Nucleic Acids Res. 2021;49(11):6249–66.
42. Theodoris CV, Xiao L, Chopra A, Chaffin MD, Al Sayed ZR, Hill MC, et al. Transfer learning enables predictions in network biology. Nature. 2023;618(7965):616–24.
43. Szturz P, Vermorken JB. Management of recurrent and metastatic oral cavity cancer: raising the bar a step higher. Oral Oncol. 2020;101: 104492.
44. Tan Y, Wang Z, Xu M, Li B, Huang Z, Qin S, et al. Oral squamous cell carcinomas: state of the field and emerging directions. Int J Oral Sci. 2023;15(1): 44.
45. Liu J, Manabe H, Qian W, Wang Y, Gu Y, Chu AKY, et al. CytoSignal detects locations and dynamics of ligand-receptor signaling at cellular resolution from spatial transcriptomic data. bioRxiv. 2024:2024–03.
46. Park C, Mani S, Beltran-Velez N, Maurer K, Huang T, Li S, et al. A bayesian framework for inferring dynamic intercellular interactions from time-series single-cell data. Genome Res. 2024;34(9):1384–96.
47. Armingol E, Officer A, Harismendy O, Lewis NE. Deciphering cell-cell interactions and communication from gene expression. Nat Rev Genet. 2021;22(2):71–88.
48. Conde-Lopez C, Marripati D, Elkabets M, Hess J, Kurth I. Unravelling the complexity of HNSCC using single-cell transcriptomics. Cancers (Basel). 2024;16(19): 3265.

49. Li M, Wei Y, Huang W, Wang C, He S, Bi S, et al. Identifying prognostic biomarkers in oral squamous cell carcinoma: an integrated single-cell and bulk RNA sequencing study on mitophagy-related genes. Sci Rep. 2024;14(1): 19992.
50. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. 2016. arXiv preprint arXiv:1609.02907.
51. Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Red Hook: Curran Associates Inc.; 2017. p. 1025–35.
52. Bonneel N, Van De Panne M, Paris S, Heidrich W. Displacement interpolation using Lagrangian mass transport. ACM Trans Graph. 2011;30(6):1–12. https://doi.org/10.1145/2070781.2024192.
53. Blei DM, Kucukelbir A, McAuliffe JD. Variational inference: a review for statisticians. J Am Stat Assoc. 2017;112(518):859–77.
54. Kingma DP. Adam: a method for stochastic optimization. 2014. arXiv preprint arXiv:1412.6980.
55. Türei D, Korcsmáros T, Saez-Rodriguez J. Omnipath: guidelines and gateway for literature-curated signaling pathway resources. Nat Methods. 2016;13(12):966–7.
56. Qiao C, Huang Y. Representation learning of RNA velocity reveals robust cell transitions. Proc Natl Acad Sci. 2021;118(49):e2105859118.
57. Maćkiewicz A, Ratajczak W. Principal components analysis (PCA). Comput Geosci. 1993;19(3):303–42.
58. Luecken MD, Büttner M, Chaichoompu K, Danese A, Interlandi M, Müller MF, et al. Benchmarking atlas-level data integration in single-cell genomics. Nat Methods. 2022;19(1):41–50.
59. Ranek JS, Stanley N, Purvis JE. Integrating temporal single-cell gene expression modalities for trajectory inference and disease prediction. Genome Biol. 2022;23(1): 186.
60. Song D, Wang Q, Yan G, Liu T, Sun T, Li JJ. scDesign3 generates realistic in silico data for multimodal single-cell and spatial omics. Nat Biotechnol. 2024;42(2):247–52.
61. Arora R, Cao C, Kumar M, Sinha S, Chanda A, Samuel D, et al. Spatial transcriptomics reveals distinct and conserved tumor core and edge architectures that predict survival and targeted therapy response. Gene Expression Omnibus; 2023. Datasets. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE208253. Accessed 1 Jan 2025.
62. Bastidas-Ponce A, Tritschler S, Leander D, Scheibner K, Tarquis-Medina M, Salinno C, et al. Comprehensive single cell mRNA profiling reveals a detailed roadmap for pancreatic endocrinogenesis. Gene Expression Omnibus; 2019. Datasets. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE132188. Accessed 1 Jan 2025.
63. Qiu X, Zhu DY, Lu Y, Yao J, Jing Z, Min KH, et al. Spatiotemporal modeling of molecular holograms. 2024. Datasets. https://www.dropbox.com/s/wxgkim87uhpaz1c/mousebrain_bin60_clustered.h5ad?dl=1. Accessed 1 Jan 2025.
64. Long W, Liu T, Xue L, Zhao H. spVelo: RNA velocity inference for multi-batch spatial transcriptomics data. 2024. Github. https://github.com/VivLon/spVelo. Accessed 1 Jan 2025.
65. Long W, Liu T, Xue L, Zhao H. spVelo: RNA velocity inference for multi-batch spatial transcriptomics data. Zenodo; 2025. https://doi.org/10.5281/zenodo.15343923. Accessed 1 Jan 2025.

Publisher's Note