Title

Challenges and progress in RNA velocity: Comparative analysis across multiple biological contexts

Journal

PLOS Computational Biology, 22(6)

ISSN

1553-734X

Authors

Ancheta, Sarah
Dorman, Leah
Le Treut, Guillaume
et al.

Publication Date

2026-06-01

DOI

10.1371/journal.pcbi.1014303

Peer reviewed

PLOS Computational Biology

RESEARCH ARTICLE

Challenges and progress in RNA velocity: Comparative analysis across multiple biological contexts

Sarah Ancheta[1,2*], Leah Dorman[1], Guillaume Le Treut[1], Abel Gurung[1], Greg Huber[1,3], Loïc A. Royer[1], Alejandro Granados[1,4*], Merlin Lange[1,5*‡]

1 Biohub, San Francisco, United States of America, **2** Present address Biological and Medical Informatics Graduate Program, University of California, San Francisco, California, United States of America, **3** University of California, San Francisco, United States of America, **4** Present address Calico Life Sciences LLC, South San Francisco, California, United States of America, **5** Institut de la Vision, Sorbonne Universite´, INSERM, CNRS, Paris, France

‡ Lead contact.
* sarah.ancheta@ucsf.edu (SA); agranado@calicolabs.com (AG), merlin.lange@inserm.fr (ML)

OPEN ACCESS

Citation: Ancheta S, Dorman L, Le Treut G, Gurung A, Huber G, Royer LA, et al. (2026) Challenges and progress in RNA velocity: Comparative analysis across multiple biological contexts. PLoS Comput Biol 22(6): e1014303. https://doi.org/10.1371/journal.pcbi.1014303

Editor: Ilya Ioshikhes, CANADA

Received: May 13, 2025

Accepted: May 6, 2026

Published: June 1, 2026

Data availability statement: All relevant code is available here https://github.com/czbiohub-sf/comparison-RNAVelo. The data sets analyzed in this paper are from previously published research and are publicly available. Zebrafish raw sequencing data are available at NCBI's SRA BioProject PRJNA940501. The raw dataset of pancreatic endocrinogenesis has

Abstract

Single-cell RNA sequencing is revolutionizing our understanding of cell state dynamics, allowing researchers to capture and quantify the transcriptomic profile of a single cell at a specific timepoint. Among the computational techniques used to predict cellular trajectories, RNA velocity has emerged as a predominant tool for modeling transcriptional dynamics. RNA velocity leverages the mRNA maturation process to generate velocity vectors that predict the likely future state of a cell, offering insights into cellular differentiation, aging, and disease progression. Although this technique has shown promise across biological fields, the performance accuracy varies depending on the RNA velocity method and dataset. We established a comparative pipeline and analyzed the performance of five RNA velocity methods on three datasets based on local consistency, method agreement, identification of driver genes, and robustness to sequencing depth. This benchmark provides a resource for scientists to understand the strengths and limitations of different RNA velocity methods.

Author summary

Single-cell RNA sequencing measures gene activity in individual cells, but each cell is only a snapshot in time. RNA velocity uses the balance of unspliced and spliced RNA to estimate where each cell is heading, helping reconstruct cell developmental trajectories. However, different velocity algorithms can yield different arrows on the same data, making results hard to interpret. We built a reproducible comparison pipeline and benchmarked five widely used RNA velocity methods across developmental datasets (mouse pancreas development and two zebrafish embryo datasets). We assessed (i) whether nearby, similar

been deposited under the accession number GSE132188.

Funding: Funding is provided by Biohub San Francisco, USA. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

cells show consistent predicted directions, (ii) how strongly methods agree with one another, (iii) how disagreements propagate to downstream driver gene, and (iv) robustness when sequencing depth is reduced by read downsampling. All methods recovered known trajectories in some settings, but performance varied with biological complexity and read depth, and driver-gene rankings were often method-dependent. We provide practical guidelines and open code to help researchers choose, cross-check, and validate RNA velocity results as hypothesis-generating evidence.

Introduction

Single-cell RNA sequencing (scRNA-seq) has enabled the characterization of thousands of transcriptomic states, defined by distinct gene expression profiles across cells, and many computational methods have been developed to infer the state lineages. While some cell populations exist in equilibrium, others constantly change due to cell differentiation, environmental changes, cell cycle, or disease perturbations [1]. scRNAseq data provides a unique opportunity to investigate cellular trajectories, the sequential transitions between distinct cell states [2–4], and identify the regulatory programs driving these processes by analyzing the dynamic shifts in gene expression patterns that guide these transitions.

Many computational methods exist to infer cellular trajectories from single-cell data, and their performance often varies depending on the type of data, the biological context, and the performance metrics used [5]. One widely used technique, RNA velocity, predicts the future state of a cell based on its mRNA splicing dynamics (Fig 1a).

RNA velocity applies dynamic modeling to scRNA-seq data to predict state transitions between individual cells [6,7]. As the mRNA matures in a cell, introns are removed through the process of splicing, so that a fraction of recently synthesized mRNA molecules exist in their unspliced state, while the rest are processed into their spliced, mature state (Fig 1a, upper panel) [15]. By considering the ratio of spliced and unspliced mRNA measurements, the RNA velocity technique fits a dynamic model to predict the rate of change in the number of mRNA molecules for a specific gene [6]. The rate of change of all genes defines a gradient in the high-dimensional transcriptomic space and predicts the directionality of the molecular states (Fig 1a, lower panel) [7,8].

RNA velocity has been applied to address fundamental questions of cell state transitions in developmental biology [1] and during perturbation [9–13]. Although RNA velocity has been widely adopted by the community, various methods, ranging from linear-based models to deep learning, differ in their sensitivity to the data, often leading to inconsistent or incorrect trajectories [14,15]. Given these limitations, this paper aims to guide researchers in evaluating and choosing the best RNA velocity method for their data, by exploring the differences in velocity vectors yielded by different methods.

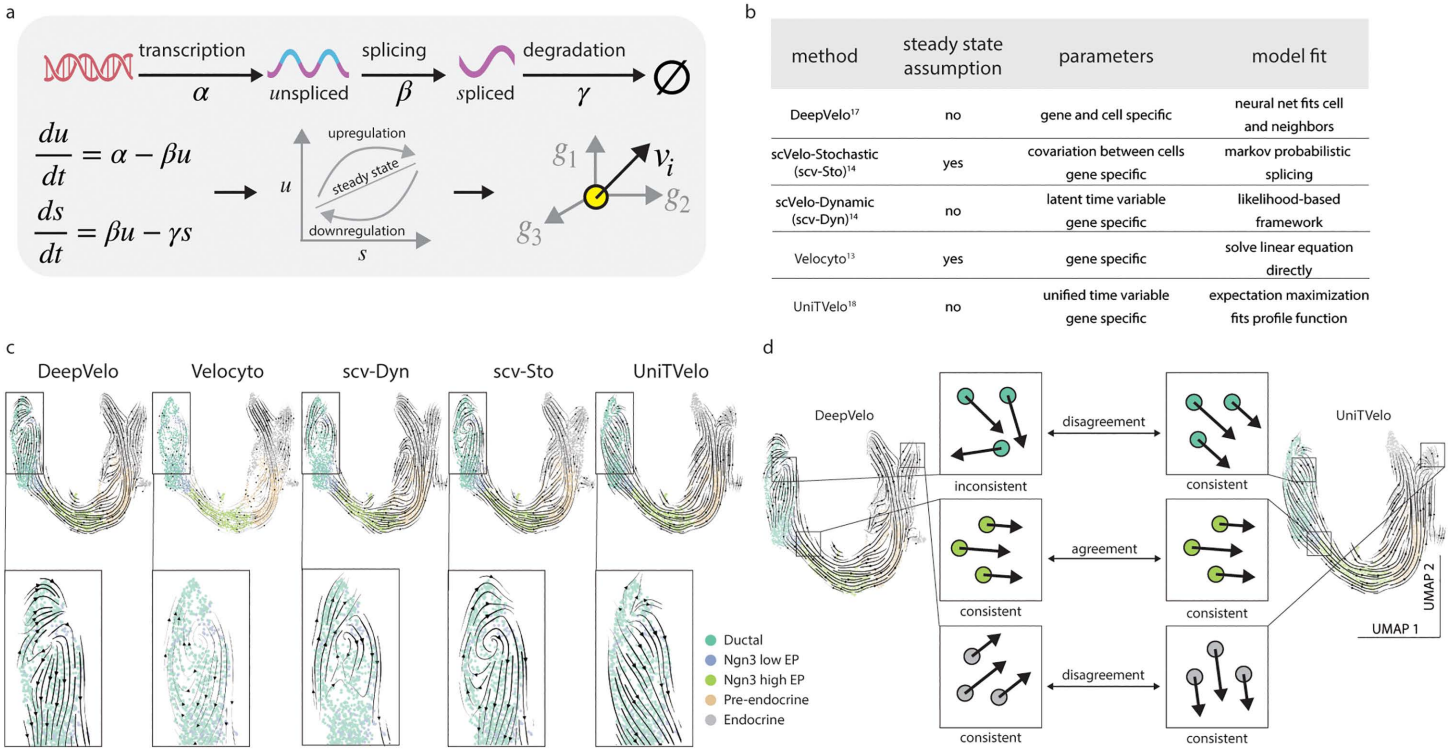

Fig 1. Different RNA velocity methods vary in directionality predictions. a. Overview of the RNA velocity workflow, with the original Velocyto [6] model as an example. The rates of transcription, splicing and degradation are notated as α, β, and γ respectively. All RNA velocity models use spliced (s) and unspliced (u) mRNA counts as model inputs and predict the directionality of a cell in transcriptomic space. **b.** Summary of the five RNA velocity models studied in this work and the methodologies implemented in each model. **c.** UMAP embeddings with RNA velocity predictions of mouse pancreas (n = 3696 cells) across the five different methods, highlighting different directionality predictions in the 'Ductal' cell type (bottom panel). **d.** A schematic describing examples of inconsistency and disagreement in the velocity predictions between different methods: (1) the consistency within the single cells in a neighborhood (the colored dots represent single cells), and (2) the agreement in directionality between methods (black arrows between boxes). The DNA icon is from BioRender.com. See also S1 Fig.

https://doi.org/10.1371/journal.pcbi.1014303.g001

The original RNA velocity model, Velocyto [6], calculates the steady state solution for a series of linear equations which model transcription for each gene as it transitions between two disjoint populations of cells with either active or inactive transcription, with distinct, discrete rates. The method assumes that there is a shared splicing rate (β) across all genes, and a steady-state solution, which expects transcriptional phases to reach a constant ratio of spliced to unspliced molecules (Fig 1b). The model assigns an RNA velocity estimate to each cell based on its deviance from the equilibrium

Finally, the directionality in the cell–cell graph, as seen in the UMAP embedding (Fig 1c), is determined by the similarity of a cell's future transcriptomic state to other cells in gene space, thus creating a directed weighted graph between cells in the dataset [6] (S1A-S1c Fig)

While Velocyto provided a proof-of-principle and initial approximation to understanding the gene expression landscape, its key assumptions do not always hold [14–16]. Not all genes follow the expected behavior of a steady-state model [16], and the results depend heavily on pre-processing decisions [14].

Recent models have introduced improvements to address these limitations. For example, scVelo introduced a stochastic version (scv-Sto) of the original steady-state model [7] using an approximate stochastic formulation of transcription, splicing, and degradation, estimating parameters with generalized least squares (Fig 1b). The inherent stochasticity in transcription affects overall variation [17]. Additionally, scVelo proposed a dynamic model (scv-Dyn) to address issues in the steady-state

model, including assumptions that the splicing rate is shared across all genes and full splicing dynamics are modeled by transcription, induction and steady-state levels. scv-Dyn fits the gene-specific transcription then processes these by introducing a shared latent time across all cells, representing the cell's internal clock in a biological progression [7] (Fig 1b).

Like scv-Dyn, UniTVelo defines a shared latent time as a cell-specific component to minimize the discrepancy between the directionality of different genes [18] (Fig 1b). Instead of fitting gene-specific splicing functions, UniTVelo [18] implements a general "profile" radial basis function that fits all genes simultaneously, deriving gene-specific splicing parameters, using expectation maximization. Alternatively, DeepVelo uses a graph convolutional network to estimate splicing and degradation rates that are gene and cell-specific [19]. Notably, DeepVelo considers not only a single cell but also its neighbors with similar expression profiles when fitting the model (Fig 1b).

As the five model computational approaches vary in methodology to calculate RNA velocity, their implicit strengths and weaknesses vary as well. Velocyto and scv-Sto incorporate linear regressions in their frameworks, which tend to be stable but may be biased [20]. UniTVelo and scv-Dyn are fit using expectation maximization, which is prone to converging at stationary points that may not be the comprehensive solution [21]. In addition, UniTVelo fits a radial basis function, which is often poorly conditioned when applied to a large dataset as they are very sensitive to small changes in the data [22]. Known limitations of graph convolutional networks, as utilized by DeepVelo, include over-smoothing and sensitivity to changes in graph structure [23]. The differing computational frameworks lead to varying performances (Fig 1b). Not all genes follow the expected behavior of a steady-state model [16].

Evaluating the accuracy of RNA velocity methods is challenging because ground truth trajectories are rarely available [16]. Moreover, the increasing number of computational methods available makes it difficult for scientists to decide the correct workflow for their research (Fig 1b). For example, even in the mouse pancreas, with a well-studied lineage, we observed significant discrepancies in the RNA velocity trajectories generated by different methods (Fig 1c inset, e.g., Ductal cells, a progenitor cell state) [24,25]. Therefore, a general benchmark that compares RNA velocity methods in different contexts is necessary to understand this technology's predictive potential and help scientists choose the best tool to address their questions. We present a comparison of the five RNA velocity methods detailed above (summarized in Fig 1b). These methods are evaluated across three developmental scRNAseq datasets, including a mouse pancreatic development dataset which has traditionally been included as a standard benchmark dataset for many RNA velocity methods [7,25], a single time-point zebrafish 24 hours post-fertilization whole-embryo dataset [26], and a multi-time point zebrafish neuro-mesodermal progenitors (NMP) lineage dataset [26] (S1d-S1g Fig). We selected transcriptionally complex datasets with multiple lineages to understand how these methods can generalize across various contexts. The NMPs and pancreatic datasets have been extensively validated and constitute and provide a well-characterized reference trajectories for our analyses [26,27]. To analyze the RNA velocity methods, we first evaluate the local consistency (L_c) within each method, determining if the velocity vectors are consistent across neighbor cells with high transcriptomic similarity (Fig 1d). Second, we examine method agreement (A_1 and A_2) to assess the landscape's robustness across different RNA velocity methods (Fig 1d). We extend this framework and evaluate the concordance in the downstream identification of driver genes. Finally, we analyze the robustness of each method relative to the number of reads, simulating the sensitivity of RNA velocity methods to sequencing depth. We observe that the smoothness and robustness of RNA velocity landscapes vary significantly depending on the cell type and biological context. We expect our benchmark to provide insight into the strengths and weaknesses of RNA velocity as a tool for understanding cell fate dynamics during differentiation.

Results

Local consistency

We initially evaluated the methods by analyzing the consistency of velocity vectors within neighborhoods, a common metric for evaluating the performance of RNA velocity methods [18,19,28,29]. The molecular state transitions taking place can be represented with a Markov transition matrix between individual cells that considers the directionality and strength

predicted by RNA velocity [7,8,30]. In the Markov transition matrix, each cell has a state transition vector representing the transition probabilities that determine the cell's likely future state. For each cell, we quantified the local consistency as the relative alignment between a cell's state transition vector and those of the most transcriptionally similar cells. We defined the metric L_C as the average cosine similarity between a cell and its 30 nearest neighbors, as each cell type has over 30 cells (Figs 2a, S2) [19]. Under this definition, neighbor cells with transition vectors in inconsistent directions will have a low score, whereas agreement between neighbors will result in higher consistency scores (Fig 1d).

We next calculated the L_C scores for the cells in the zebrafish neuromesodermal progenitor (ZF NMP) lineage dataset for the RNA velocity method scv-Dyn and projected them on the UMAP embedding (Fig 2b). The distribution of L_C scores showed significant differences across cell types and UMAP regions. Cell types from well-characterized developmental lineages, such as the mesodermal-derived cells, showed high consistency, in particular the axial mesoderm (PSM→somites→muscle) [31,32] and the lateral plate mesoderm (Fig 2b). In contrast, those with more complex biological cellular heterogeneity, such as neural cells (see Fig 2b, neural tube and hindbrain), showed the lowest consistency values.

More generally, the L_C distribution showed high heterogeneity and appeared to be cell type-specific (Figs 2b, S3). We then compared the local consistency distributions for all three datasets and across RNA velocity methods (Figs 2c, S3). Some methods, such as UniTVelo, showed high L_C for all cells in the dataset, indicating a high degree of smoothness across the whole velocity graph, independently of the cell type (Figs 2c, S3a, S3c). In contrast, DeepVelo and scv-Sto showed intermediate L_C values with heterogeneous distributions for all three datasets (Figs 2c, 2d, S3a, S3c). We observed a clear trend in L_C across methods and datasets, where scv-Dyn consistently showed lower values, and UniTVelo showed high L_C values across all datasets (Fig 2c).

To understand the heterogeneity in L_C values, we explored the distributions for the cell types in each dataset in more detail. In whole zebrafish embryos at 24 hours post fertilization (ZF embryo 24hpf), the distribution of L_C showed significant differences between cell types (Fig 2d), except for UniTVelo's velocity calculations, which resulted in high L_C values across diverse cell types (Figs 2e, S3b, S3d). For the cell types with the highest L_C values, we observed strong agreement across most methods, suggesting that the RNA velocity signal in these neighborhoods is strong enough to be discernible by different models (Figs 2e, S3b, S3d). Together, these results indicate that the landscape's smoothness and expression distribution in transcriptional space varies depending on cell type.

Given the diverse outcomes from different methods, we grouped the results by creating a consensus score, the average L_C across all methods, which enabled the characterization of high or low agreement for different datasets (Fig 2f). In the pancreas dataset, 92% of cells exhibited consensus L_C above 0.5, indicating high agreement between methods (Fig 2f). Similarly, 74% of cells in ZF embryo 24hpf dataset had a consensus L_C above 0.5, but only 39% of cells are over 0.5 in the ZF NMP dataset (Fig 2f). The low consensus in the ZF NMPs dataset can be explained by the heterogeneity in the age of the cells, as this dataset integrates multiple time points. The consensus L_C could assist in identifying regions where the differentiation signal is strong enough to reconstruct single-cell trajectories based on RNA velocity. On the other hand, lower L_C across all methods could indicate that the velocity signal for the cell type is noisy, their differentiation process is more complex with multiple potential lineages, or the cells are not differentiating (e.g., hindbrain cells in the ZF NMP dataset or neurons in the ZF embryo 24hpf) (Figs 2e, S3b). Overall, we observed that well-defined developmental transitions with low cell diversity have high local consistency, whereas lineages with complex transcriptional diversity show low local consistency. This may also reflect the biological stage of the cells for example, if differentiation occurs on a timescale that is too slow to be inferred reliably from splicing information.

Method agreement. Though local consistency with a method is an important metric in evaluating RNA velocity methods, alternatively, one can ask if the vector predictions from different velocity methods agree. We compare the cell-cell transition matrices from each method directly, in the shared cell-cell space of each dataset. Agreement between methods can help to identify lineages and cellular states with stronger velocity signals that correlate with biological relevance. We analyzed the agreement between methods using two approaches: (1) comparing the directionality of each cell's vector

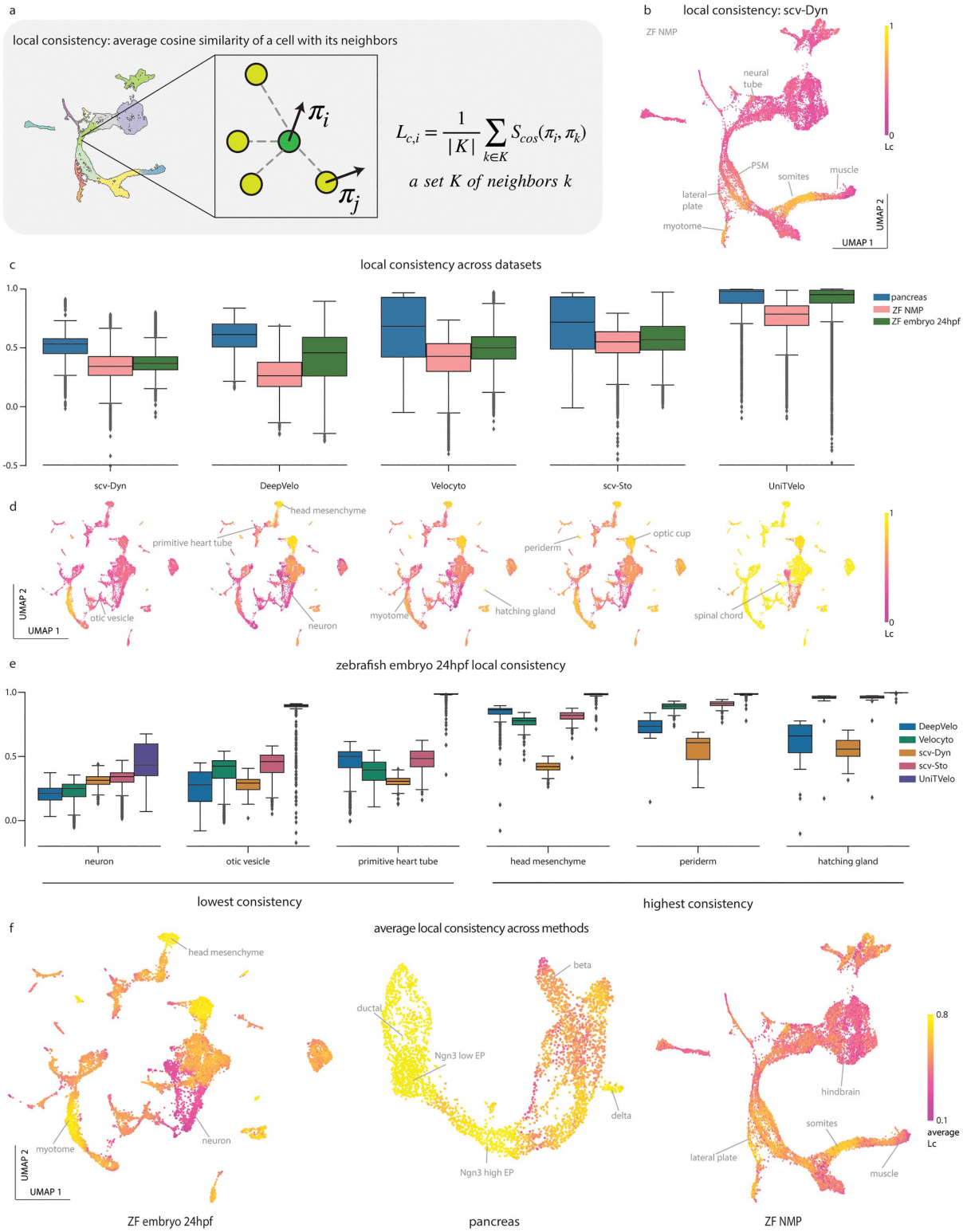

Fig 2. RNA velocity local consistency in cell neighborhoods. a. The local consistency quantifies the average velocity transition vector alignment of a cell with its nearest neighbors. Each dot is a cell, and the local consistency is the average of the cosine similarities between the target cell and each of its nearest neighbors (K = set of 30 neighbors **k)**. **b.** Single-cell local consistency of scv-Dyn projected into the UMAP embedding for the ZF NMP

dataset (n = 16035 cells). **c.** Local consistency distributions for each RNA velocity method across the three datasets. **d.** UMAP embeddings for zebrafish whole-embryos 24hpf (n = 12914 cells) colored by the single-cell local consistency calculated for each RNA velocity method. The labels highlight cell populations with high or low consistency. **e.** Local consistency distributions for the three cell types with the highest and three with the lowest average local consistency in the zebrafish embryos 24hpf dataset. **f.** UMAPs colored by the average (consensus) local consistency across all methods, for the three datasets. The labels highlight cell populations with high or low consensus consistency. *See also* S2 Fig.

https://doi.org/10.1371/journal.pcbi.1014303.g002

predictions for each pair of methods (Fig 3a - 1) and (2) comparing the direction of each method to the 'median vector,' the aggregate central vector calculated by taking the median across all the methods (Fig 3a – 2 – the median vector serving as ground truth). The metric A_1 is defined as the landscape's agreement across pairs of different methods (Figs 3a, right, S4-S6). More specifically, the agreement (A_1) quantifies the cosine similarity between a pair of transition probability vectors obtained from different methods for the same cell. In comparing DeepVelo and Velocyto in the ZF NMP dataset, levels of agreement varied by cell type.

We projected the A_1 distribution on the UMAP embedding and observed high agreement in the mesodermal lineages and lower agreement in the hindbrain cells, consistent with patterns of developmental heterogeneity [26] (see Results Local Consistency, Fig 3b). The mesodermal lineage showed high agreement only in the early stages of differentiation, but as cells differentiate into muscle, the agreement across methods dropped to almost zero (Fig 3b). The A_1 (DeepVelo vs. Velocyto) scores (Fig 3b, histogram) over all cells were widely distributed, indicating heterogeneity in the agreement between the two methods, with patterns of low and high agreement similar to those found for local consistency in the ZF NMP dataset (i.e., highest in somites, lowest in the hindbrain) (Fig 2b).

Next, we investigated the agreement across methods for cell types in the ZF NMP dataset and observed consensus across most methods. To examine the method agreement for different cell types more closely, we computed the mean A_1 for each cell type across all pairs of methods (Fig 3c). Analysis of the agreement between methods by cell type revealed varying levels of agreement depending on the method and cell population (S4-S6 Figs). The lateral plate, somites, and endoderm exhibited concordance among all methods except for UniTVelo (Fig 3c). Examining the UMAP for the ZF NMP dataset revealed that UniTVelo had predicted the opposite direction to the known biological trajectory, going from the differentiated cell type towards the progenitor [26,33] (S1b Fig). For three cell types (notochord, endoderm, and hindbrain) scv-Dyn and DeepVelo had a low agreement, which is correlated with cell type diversity and transcriptomic complexity (Figs 3c, S7b).

To expand the investigation of method agreement on a global scale and evaluate the systematic disagreement across methods, we compared each method to the median transition vector of each cell. The median transition vector is calculated as the median of the individual vectors generated by all five methods for the cell, generating an aggregate vector summarizing the central vector found across all methods (Fig 3a, right). We then computed the cosine similarity of each method's transition vector with the cell's derived median vector (A_2) (Fig 3a). The A_2 metric, therefore, provides a measure of how well each method agrees with the consensus prediction across all methods. For the ZF NMP dataset, we noticed that UniTVelo systematically disagreed across many cell types, whereas DeepVelo had a scattering of cells with low A_2, mixed with higher values (Fig 3d). Velocyto, scv-Dyn, and scv-Sto had high cosine similarity with the median vector (S7c, S7d Fig).

When applying the analysis to the three datasets, we found method agreement with the median vector varied depending on the dataset, except for scv-Dyn and scv-Sto (Fig 3e). DeepVelo had high levels of agreement for the pancreas dataset, but lower levels of agreement in the zebrafish datasets (pancreas: median $A_2 = 0.894$, ZF NMP: median $A_2 = 0.667$, ZF embryo 24hpf: median $A_2 = 0.726$) (Fig 3e). The agreement for UniTVelo followed a similar pattern, with much lower levels of agreement in the zebrafish datasets (pancreas: median $A_2 = 0.848$, ZF NMP: median $A_2 = 0.305$ and ZF embryo 24hpf: median $A_2 = 0.392$) (Fig 3e). This may be due to the much bigger size of the zebrafish datasets, as the underlying radial basis function being fit in the method is prone to inaccuracy when utilized for large-scale data [22]. The

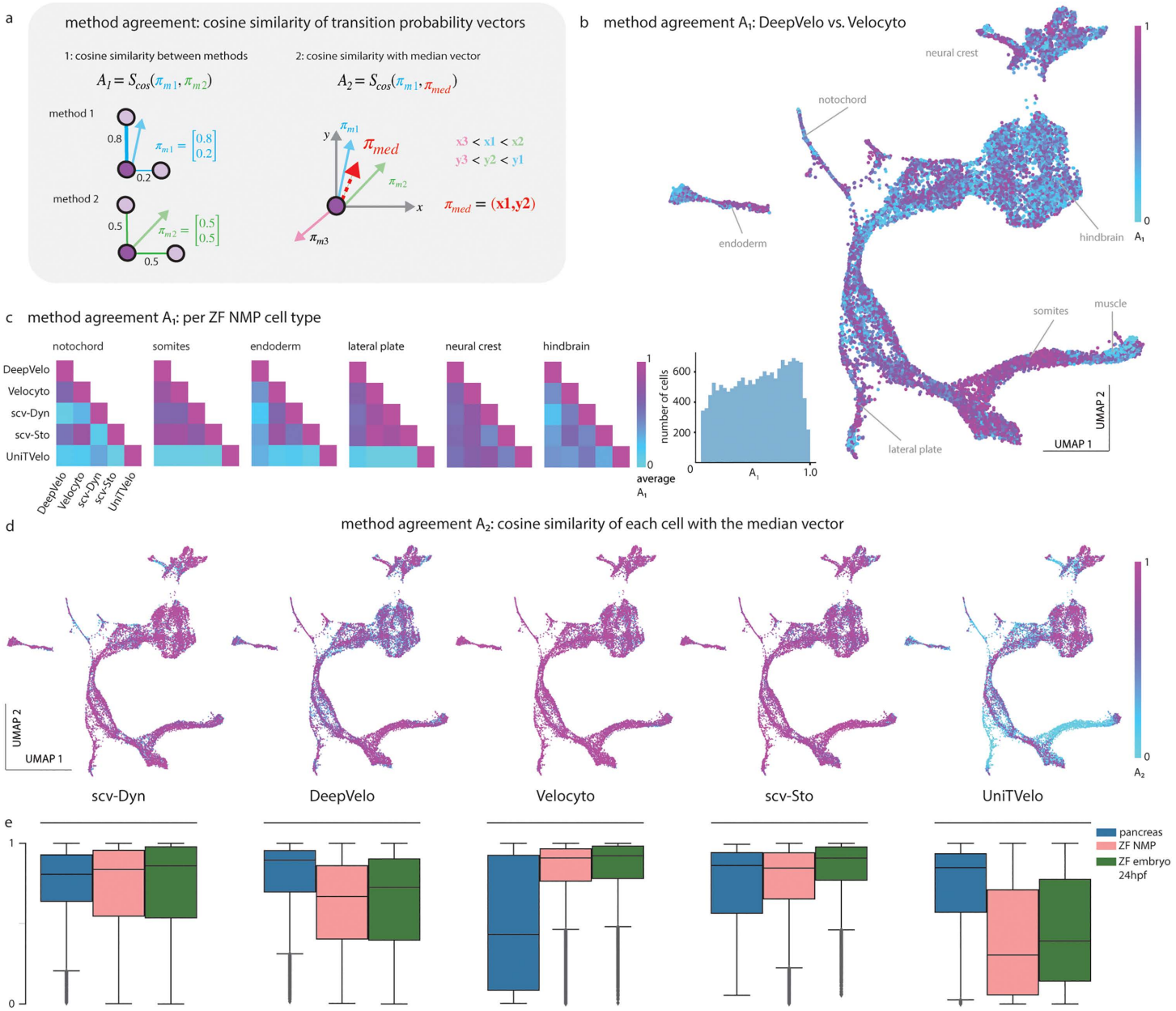

Fig 3. Comparing single-cell velocity agreement between methods. a. Each RNA velocity method yields a cell-cell transition graph. Part (1) of the schematic shows different transition probability vectors for the same cell obtained from different methods. A transition vector is defined by the transition probabilities between cell states in the graph. The method agreement (1) metric quantifies the similarity between cell transition vectors across datasets using cosine similarity. Part (2) of the schematic illustrates the median (central) vector for each cell, computed by taking the median of all transition vectors across methods. The method agreement (2) metric quantifies the similarity of cell transition vectors from each method as compared to the median vector by using cosine similarity. **b.** UMAP embedding for the ZF NMP dataset with the method agreement (1) between DeepVelo and Velocyto. The labels highlight cell populations with high or low agreement between the two methods. The histogram shows the distribution of method agreement values for all the cells. **c.** Pairwise comparisons for six cell types from the ZF NMP dataset across all methods. The heatmap shows the mean method agreement across individual cells within a cell type for each pair of methods. **d.** UMAP embeddings for ZF NMP for each RNA velocity method, colored by each method's agreement (2) with the median vector. **e.** Distribution of method agreement (2), each method's agreement with the median vector for the three datasets. *See also* S3, S4, S5, S6 and S7 Figs.

https://doi.org/10.1371/journal.pcbi.1014303.g003

high agreement of the deep learning-based method DeepVelo on the pancreas (a commonly tested dataset for developing RNA velocity methods) may indicate overtraining [19]. The opposite pattern was seen in Velocyto, where the score was low on the pancreas dataset (median $A_2 = 0.432$), and the method achieved the highest agreement across all methods on the zebrafish datasets (ZF NMP: median $A_2 = 0.908$, ZF embryo 24hpf: median $A_2 = 0.922$) (Fig 3e). Altogether, the variation in agreement across datasets and methods underscores the importance of implementing and comparing predictions across multiple methods when interpreting RNA velocity. On the more transcriptionally complex zebrafish datasets, Velocyto, scv-Sto and scv-Dyn had stronger performances. The observed differences in performance indicate that although the preprocessing is similar across all methods excluding UniTVelo, the method predictions diverge enough to disagree in the transition matrix construction. However, the method agreement does not mean that the method's predictions are correct. Disagreement among methods indicates that further investigation of the data and lineage in the particular subset is needed, and could either be due to biological complexity, noise in the data, or incorrect predictions in the technical approach.

Downstream: Overlap of driver genes

We next explored how the disagreements between methods are propagated in downstream analysis by evaluating the overlap in macrostates, defined as regions in the transcriptional landscape cells are unlikely to transition out of, and top driver genes in the pancreas dataset, the most well-studied lineage among the datasets we evaluated. We utilized CellRank to identify macrostates and driver genes, i.e., genes whose expression highly correlates with a specific trajectory or lineage with a velocity kernel generated from each method [8] (see Methods). CellRank estimates absorption probabilities (i.e., probability of a cell fate trajectory towards a particular terminal state) using ensembles of random walks. Genes are classified as drivers if they are systematically highly expressed in cells that are more likely to differentiate towards a given terminal state [8] (see Methods). CellRank estimates absorption probabilities (i.e., probability of a cell fate trajectory towards a particular terminal state) using ensembles of random walks. Driver genes are identified by computing the Pearson correlation of gene expression with the cell fate probabilities associated with each specific terminal state [8] (see Methods).

While the macrostates identified by CellRank generally agreed across methods and corresponded to the Leiden clusters and cell type annotations (initial cluster Ductal, terminal Alpha, Beta, Delta, and Epsilon), we found that some states didn't agree across methods (Figs 4a, 4b, S8a, see Delta, Ngn3 low EP, Ductal 5).

In exploring the role of driver genes, we focused on the terminal state Beta, which was robustly identified by CellRank in all methods (Fig 4b). We looked at the overlap among the identified top 100 driver genes across the methods.

Strikingly, we found only one gene, Pdx1, at the intersection of all methods (Fig 4c). Pdx1 is an essential transcription factor and master regulator in the development and maintenance of Beta cells [34]. While it is encouraging that all methods agreed on Pdx1, it is notable that only one gene appeared in the intersection. Ideally, the identification of driver genes using RNA velocity will identify genes that are more highly correlated with trajectories, rather than marker genes of the terminal state.

Interestingly, the different methods clustered into two groups with regard to the driver genes identified (Fig 4c). The first group (DeepVelo, scv-Sto and Velocyto, agreed on 79% of driver genes) (Fig 4c); whereas the second group (scv-Dyn and UniTVelo) agreed on 55% of their predicted driver genes (Fig 4c). We next computed a (GO) analysis on the driver genes for the first two columns of the histogram Fig 4c (S9 Fig). The pathway analysis reveals that the driver genes are involved in pathways related to development (pancreatic but not only) and cell transcription as expected (S9 Fig). scv-Dyn and UniTVelo both employ a latent time component in their models, so cells are assigned an order with the assumption that there is an underlying biological process, which may result in the identification of similar driver genes for the core process. When conducting an analysis of the driver genes for the Epsilon terminal state, we found similarly low levels of overlap, with three genes at the intersection of all methods (S8b Fig). The general lack of agreement in driver genes

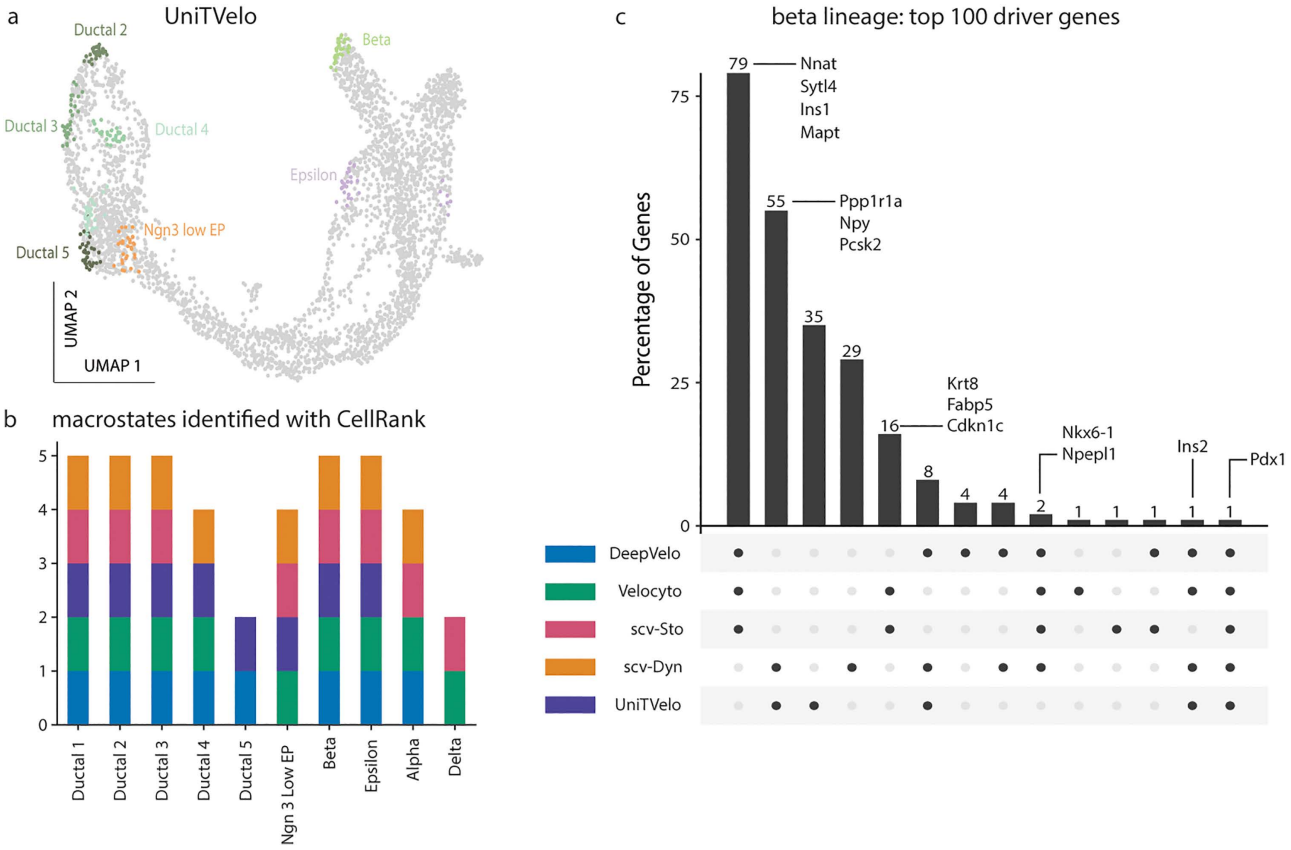

Fig 4. Comparing driver genes predicted from different methods. a. Macrostates identified by CellRank based on the velocity kernel from UniTVelo in the pancreas dataset (see Methods). **b.** A summary of the macrostates identified by CellRank for different RNA velocity methods. The x-axis shows the macrostate label. The y-axis shows the number of methods in which the macrostate was identified by CellRank. **c.** A histogram showing the top 100 genes for the Beta lineage from each method. The bars indicate the number of genes that appear at the intersection of the methods. *See also* S8 and S9 Fig.

https://doi.org/10.1371/journal.pcbi.1014303.g004

emphasizes the importance of considering multiple methods when making trajectory predictions with RNA velocity and testing these predictions using experimental perturbations or other validation methods. Given the low overlap of predicted driver genes across methods, we caution that RNA-velocity-based driver-gene rankings are method-sensitive and should be treated as hypothesis-generating. Cross-method concordance can serve as a conservative robustness check to prioritize candidates for follow-up, but does not establish causality.

Robustness to sequencing depth

The robustness of the RNA velocity methods to changes in sequencing depth was analyzed to evaluate the sensitivity of the methods. To simulate different levels of depth, we subset the ZF embryo 24hpf dataset, randomly selecting different proportions of reads (2, 5, 12, 25, 50, 80, 95 and 98%, each repeated five times), computed each velocity method and evaluated the robustness of the prediction as compared to the full dataset (Fig 5a, S10-S11 Figs).

The impact on the velocity predictions can be observed at a high level, as trajectories vary with different proportions of reads. When computed by DeepVelo with 2% of the reads, the velocity flow for the myotome is in the opposing direction

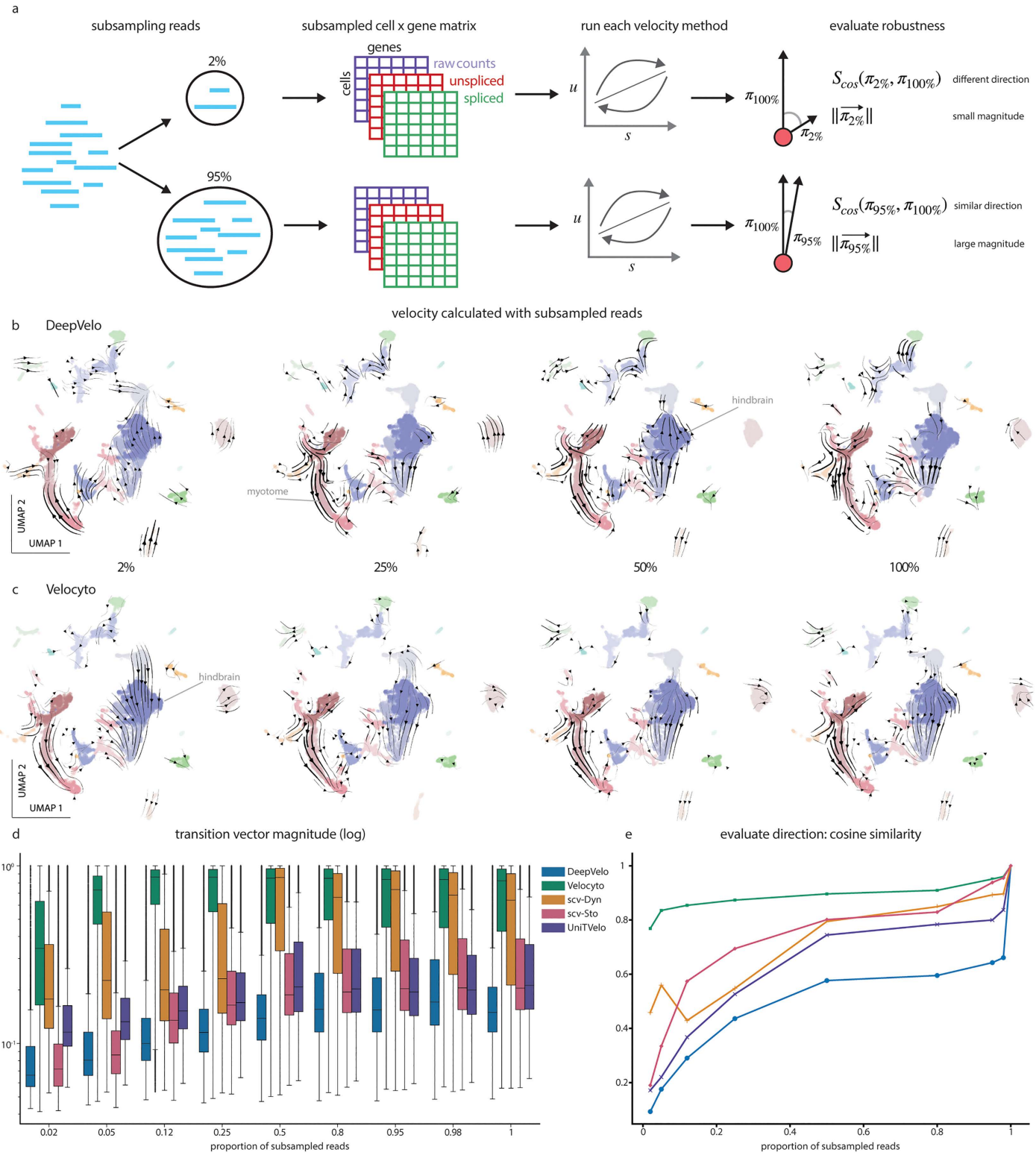

Fig 5. Robustness of methods to sequencing depth. a. Sequencing depth is simulated by taking a randomized subset of the reads, including 2, 5, 12, 25, 50, 80, 95, and 98%. The input matrices, including raw, spliced, and unspliced counts, are then derived from the reads. All RNA velocity methods

are run on the subset, and we evaluate the robustness based on the vector's magnitude and direction as compared with the predictions from 100% of the reads. **b.** Illustration of UMAP embeddings of the ZF whole-embryo 24hpf with RNA velocity predictions from DeepVelo, calculated from subsets 2, 25, 50, and 100% of the reads. The labels highlight cell populations with directionality disagreement between subsets. **c.** UMAP embeddings of the ZF whole-embryo 24hpf with RNA velocity predictions from Velocyto, calculated from subsets 2, 25, 50, and 100% of the reads. The labels highlight cell populations with directionality disagreement between subsets. **d.** Distribution of transition vector magnitudes across all subsets for the five RNA velocity methods for the ZF whole-embryo 24hpf dataset. **e.** Comparison of directionality robustness for each method, determined by calculating the cosine similarity between the transition vector from the subset and the directionality of the transition vector derived from 100% of the reads. This calculation was averaged across all cells in the ZF whole-embryo 24hpf dataset. *See also* S10, S11, S12 and S13 Figs.

https://doi.org/10.1371/journal.pcbi.1014303.g005

as compared to all increased subsets (Figs 5b, S10a). With Velocyto, the flow through the hindbrain appears to become more complex as the proportion of reads increases (Figs 5c, S10c).

To evaluate the sensitivity of each method to sequencing depth, we examined the magnitude and direction of the transition vectors. As expected across all methods, the magnitude of the transition vectors generally increased with the number of reads (Fig 5d). The transition vectors from Velocyto had the largest magnitudes across all levels, and scv-Dyn had similarly high magnitudes starting at 50% of the original sequencing depth (Fig 5d). The transition vector magnitudes for scv-Sto, UniTVelo, and DeepVelo remained much smaller (Fig 5d).

In comparing the magnitudes derived from subsets to those calculated from 100% of the data, the vector magnitudes converged for scv-Sto and Velocyto, whereas DeepVelo, scv-Dyn, and UniTVelo maintained low levels of correlation with the magnitudes from the full reads (S12a Fig). Together, the data indicates that more reads lead to larger transition vectors which indicate stronger directionality predictions, and that the vector magnitudes for Velocyto and scv-Sto are more robust to read numbers, consistent with the more stable linear models which they both utilize (S12a Fig).

When computing the analysis, we repeated the subsampling five times at each proportion level. We compared the variance in the replicates as the average pairwise cosine similarity of the transition vectors for each cell. Across all methods, scv-Dyn had the least overall variance, with the highest median similarity scores across the dataset. As the percentage of the reads increased to 98%, the predictions across all methods became more consistent, with the median cosine similarity of scv-Dyn, scv-Sto, Velocyto, and UniTVelo reaching values above 0.98. DeepVelo reached a maximum cosine similarity around 0.83, as its variance remained high (S13 Fig). The performance of DeepVelo relies on a stable graph structure, which may be sensitive to small changes in the data.

We evaluated the directionality robustness of the transition vectors by computing the cosine similarity between each cell's predicted transition vector from the subset reads and the predicted vector from the full reads. Velocyto was the most robust, with the largest increase at 5% of the reads to a cosine similarity around 0.85, after which additional reads provided a small amount of improvement (Fig 5e). All other methods plateau at lower values, with a jump at the end between 98% and 100% of the reads and reaching their stable points around 50% of the reads (Fig 5e). DeepVelo was the lowest, reaching a plateau of ~0.5, while scv-Sto and scv-Dyn were just above UniTVelo (at 0.7) (Fig 5e). We checked that the magnitude did not affect the evaluation of directionality and observed that the variance was smallest for Velocyto (S12b, S12c Fig). The directionality of the velocity predictions from Velocyto are the most robust to simulated lower sequencing depth. In datasets with high transcriptional diversity and low levels of reads, DeepVelo may yield inconsistent results, as the model depends on a stable graph structure. Velocyto will be most consistent for datasets with low sequencing depth, followed by scv-Sto.

Discussion

We evaluated the performance of five RNA velocity methods on three developmental datasets by analyzing their local consistency, method agreement, overlap of driver genes, and robustness to sequencing depth. Collectively, the RNA velocity methods identified known biological trajectories and important driver genes, with each method displaying varying levels of performance depending on the dataset and evaluation metric.

a

method	local consistency	method agreement	reads robustness
DeepVelo	✓	✓	✗
scVelo-Stochastic (scv-Sto)	✓	✓	✓
scVelo-Dynamic (scv-Dyn)	✗	✓	✓
Velocyto	✓	✓	✓
UniTVelo	✓	✗	✓

✓ high performance
✓ medium performance
✗ low performance

Fig 6. Guidelines to choose an RNA Velocity Method. Table of metrics used to evaluate each RNA velocity method, with color indicating high (green), medium (yellow) or low (red) performance of the method for each metric. Local consistency indicated is the median local consistency for each method across the median for each dataset: high (0.8-1), medium (0.5-0.8), and low (0-0.5). Method agreement is the median agreement with the center vector for each method, across the median for each dataset: high (0.8-1), medium (0.5-0.8), low (0-0.5). Reads robustness is based on the cosine similarity plateau value for the directionality of each method: high (0.8-1), medium (0.5-0.8), low (0-0.5).

https://doi.org/10.1371/journal.pcbi.1014303.g006

This paper does not aim to claim which method is superior; instead, it seeks to equip scientists with guidance and insights into the various outcomes of different RNA Velocity methods. We observed many differences in method outcomes across each of the metrics and therefore recommend that RNA Velocity is used as a hypothesis generation tool, to explore directionalities, lineages and driver genes for further experimental validation. The variety of metrics posed here aim to, enable scientists to choose an approach that best suits their dataset. Our research emphasizes the importance of implementing a method that best fits the dataset, as we observed the varying levels of performance based on transcriptional diversity and sequencing depth and encourages the exploration of outcomes from multiple approaches when identifying trajectories for further experimentation. None of the methods perform highly in the three evaluated parameters (Fig 6). Therefore, we always recommend validating RNA velocity predictions. Based on the outcome of the evaluated metrics summarized in Fig 6, the original RNA velocity model Velocyto is high-performing in two categories and has medium performance for local consistency. As the original method from which the others are expanding, Velocyto is the most consistent in our comparison framework. Local consistency varied across cell types and methods (Fig 2c, 2e). For cell types with high local consistency, all five methods showed similar results, suggesting that the signal in the data was high enough to be identified by different models. In some terminal state cell types (i.e., muscle, hindbrain, neural tube), we observed low consistency, which may be due to noisy measurements or heterogeneity in subpopulations [16] (summarized in Fig 6). Many methods benchmark their performance using metrics to measure local consistency [19,28,29], and our findings highlight the value of incorporating different analytical perspectives.

When evaluating the methods compared to the median vector as computed across all methods, the inconsistencies for different datasets were apparent. Because the pancreas dataset is often used as a benchmark dataset for RNA velocity methods, the high performance of other methods over Velocyto (the original method) could indicate improvement in the field or could indicate overfitting [19,35]. In previous benchmarking studies, high performance on the pancreas dataset is observed across methods, perhaps because it is a more straightforward and well characterized lineage. We recapitulate this pattern in our findings. UniTVelo had low method agreement for both zebrafish datasets yet high local consistency, which is likely due to over-smoothing as the method fits a single profile function for all cells and genes and utilizes a unified latent time to infer dynamics in a sample [18] (Fig 6). As UniTVelo fits a radial basis function, which can become unstable when interpolating a large amount of data, we suspect that its strong performance on the pancreas may be due to the much smaller size of the dataset (n = 3696 cells) compared to the zebrafish datasets (n = 16035 and n = 12914 cells) [22]. The deep learning-based method, DeepVelo, had lower method agreement in the zebrafish datasets as compared to the pancreas (Fig 3e). We suspect that the default parameters were optimized for a specific training set, and that with parameter optimization the methods may perform more accurately, as deep learning models are more complex [36].

Performance highly varies based on hyperparameter choices, and all methods are vulnerable to adjustments that yield results that align with prior assumptions. Therefore, we recommend that predictions are used as an exploratory tool.

It is important to note that the method agreement test developed in the manuscript serves as a measure of how consistent different methods are with one another on a given dataset. High agreement can arise, in regions where the underlying lineage structure is relatively straightforward, so that multiple approaches converge on a similar solution. Conversely, low agreement can reflect several possibilities: (1) the true lineage structure is complex or branching, (2) the splicing information is noisy or incomplete, finally (3) one or more of the methods is making incorrect predictions. Therefore, we consider disagreement in method direction as a flag for further investigation of that trajectory (evaluation of biological plausibility, additional analyses...) rather than direct evidence that any particular method is inferior. The downstream analysis identification of driver genes was sensitive to the differences in velocity calculations between methods, emphasizing the need to include multiple RNA velocity predictions before making decisions about further experimentation.

The levels of robustness to the sequencing depth, as simulated by downsampling the number of reads, varied depending on the model type of each method. The deep learning-based approach DeepVelo was more sensitive to the number of reads, as the graph computational network model is highly variable as the graph structure shifts with the changing transcriptomic information [23]. Velocyto was the most robust to the number of reads; as a linear regression-based model, its performance is more stable [20] (Fig 6). For scientists who would like to implement RNA velocity, we considering recommend sequencing depth when choosing the best method for their data.

Several limitations apply to the findings reported here: 1) the number of datasets is limited and only includes two organisms with vastly different numbers of cells, 2) our study is not comprehensive of all current RNA velocity methods and includes 5 of the top 10 cited methods, as our attempts to include several other methods did not succeed due to package dependencies and deprecated software, and 3) all comparisons rely on default parameters as suggested by the authors as we do not explore method optimization. We recognize that we are not providing a comprehensive review of all RNA velocity methods. But we believe our framework provides an initial approach to address these questions and to apply to newly released technics, especially deep-learning based methods. We provide the code for each of the metrics and analyses discussed in a GitHub repository, with the link in the Methods section.

With the current restraints of scRNA-seq, short-read mRNA sequencing alone might be insufficient to describe the differentiation dynamics of these cell types and additional “omic” modalities could improve the modeling of velocities [16]. For example, recent approaches combine RNA sequencing with ATAC-seq to jointly model RNA velocity [37]. Additional single-cell techniques, such as long-read sequencing and methods to detect ambient RNA in single-cell sequencing, could enhance RNA velocity models [38,39]. The ability of long-read sequencing to capture all spliced reads and improvements in genome annotation may provide additional transcriptomic information needed to more accurately predict the future state of a cell with RNA velocity. Similarly, identifying RNA contamination via ambient RNA detection methods in single-cell sequencing can enhance the quality of scRNA-seq data. Together, these technical advancements could refine the estimation of splicing parameters and lead to more precise velocity predictions. As velocity is a predictive with a range of predicted outcomes depending on the method, utilizing a range of approaches can provide insights into trajectories in the transcriptomic space to help us understand the underlying biological processes and provide directions for further exploration.

Methods

Code availability. All relevant code is available here https://github.com/czbiohub-sf/comparison-RNAVelo.

Data availability. The data sets analyzed in this paper are from previously published research and are publicly available. Zebrafish raw sequencing data are available at NCBI’s SRA BioProject PRJNA940501. The raw dataset of pancreatic endocrinogenesis has been deposited under the accession number GSE132188.

Single-cell quality control and RNA velocity method implementation. The mouse pancreatic developmental dataset is available through scVelo [7,25]. The ZF NMP and the ZF embryo 24hpf datasets are subsets of the

Zebrahub data [26]. The ZF NMP dataset contains multiple timepoints and embryos, and the ZF embryo 24hpf also contains multiple embryos. The datasets have no batch correction. For each RNA velocity method, we followed the workflow and preprocessing as outlined in each method's tutorial. We executed the same preprocessing for Velocyto, scv-Sto, scv-Dyn, and DeepVelo. For each dataset, we normalized the raw counts using scVelo v0.2.5. Genes with fewer than 20 total detected counts were excluded. Gene counts were normalized by dividing by the total counts per cell and multiplying by the median total counts per cell. The top 2,000 highly variable genes were then log normalized using the functions scvelo.pp_filter_genes, scvelo.pp.normalize_per_cell, scvelo.pp.filter_genes_dispersion, and scvelo.pp.log1p respectively. Utilizing the scVelo pipeline, we computed first and second moments for the velocity estimation, with 30 principal components and 30 nearest neighbors (sc.pp.neighbors, scvelo.pp.moments). The scVelo v0.2.5 Deterministic mode recapitulates the Velocyto steady-state model. We ran this model with scvelo.tl.velocity mode = 'deterministic' using default parameters. For scv-Sto and scv-Dyn, we ran scvelo.tl.velocity (scVelo v0.2.5) with mode = 'stochastic' and mode = 'dynamic' with default parameters. We ran DeepVelo [19] (0.2.5rc1) with the default configuration. For UniTVelo [18] (v0.2.5.2), all preprocessing is part of the model, and we ran the model using its default parameters. We input precomputed Leiden clusters for the input parameter 'cluster,' as calculated with scanpy for the zebrafish datasets [26]. For the pancreas dataset, we input the cell type parameter, as given with the dataset, as the 'cluster' input.

Velocity counts and generation of the velocity graph. For each of the RNA velocity methods, a 'velocity' prediction was generated. This yields a matrix in which each cell has a velocity vector with directionality in the transcriptomic space, and a 'velocity_graph,' a cell state–cell state graph. To compare the methods directly, we used the scVelo v0.2.4 function scv.utils.get_transition_matrix to compute cell state–cell state transition probabilities from the velocity graph, based on the similarity of a cell's predicted future state to the profile of cells observed in the sample. We input the dataset and each method's 'velocity_graph' variable to get the transition matrix.

Consistency within single-cell neighborhoods. For each method, we calculated the local neighborhood consistency (L_C) as shown in DeepVelo [19]. We calculate $L_{C,i} = \frac{1}{30}\sum_{k\in K} S_{cos(i,k)}$, where $L_{C,i}$ indicates the local consistency of the cell i, and K indicates the set of the 30 nearest neighbors of k the cell. We computed the cosine similarity between the cell state transition vectors of each cell to its neighbors, using np.inner and np.linalg.norm from numpy v1.23.5. L_C is defined as the average cosine similarity across all 30 nearest neighbors. We used the local neighborhood calculation for each RNA velocity method, and for all datasets. To identify cell types with higher or lower consistency, we grouped cells together by Leiden cluster, and labeled each Leiden cluster with the most prominent cell type represented. To evaluate the local neighborhood consistency across all methods, we took the average L_C across all RNA velocity methods for each single cell.

Transition vector agreement between methods. We calculated the pairwise method agreement by computing the cosine similarity between two transition vectors for the cell from transition matrices created by two different RNA velocity methods: $A_1 = S_{cos}\left(v_{i,M1}, v_{i,M2}\right)$, where $S_{cos(v_{i,M1},v_{i,M2})}$ indicates the cosine similarity between state transitions vectors for cell i, with pairs of methods M1

and M2 for each cell. The cosine similarity is defined as the dot product of the vectors, divided by the product of the norm of each vector, which we implemented using np.inner and np.linalg.norm with numpy v1.23.5. For each pair of methods and each cell, this yields a method agreement score A_1.

With the transition matrix from each method, we calculate the central vector for each cell as the median transition vector across all methods. We then compute the agreement with the central vector for each individual method as the cosine similarity $A_1 = S_{cos}\left(v_{i,M1}, v_{i,Med}\right)$ of the transition vector $v_{i,M1}$ (for the method M1 and cell i), with the median vector for the cell i, $v_{i,Med}$. **Overall the method agreements are intended as relative measures of cross-method consistency.**

Computing macrostates and driver genes with CellRank. In the pancreas dataset, we used the variable 'velocity,' the velocity for each gene and cell generated from each RNA velocity method, to create the velocity kernel from CellRank [8] v2.0.0, and we then computed a transition matrix with CellRank's default parameters, including model='deterministic,'

similarity='correlation,' and softmax_scale=6.925. The velocity kernel was generated with the parameters backward=False and vkey='velocity.' Next, we followed the CellRank pipeline and generated macrostates, defined the terminal states, and computed lineage drivers. We fit the pancreas cluster_key = 'clusters' as the cell type variable, and we set the number of macrostates to 8.

Based on the macrostates generated, we set the terminal states to pancreatic cell types 'Beta', 'Alpha', 'Delta', and 'Epsilon' if identified. We then computed the lineage drivers for the Beta lineage, as the terminal state Beta was found in all methods.

In CellRank, the lineage driver genes were identified by the correlation of gene expression with the fate probabilities of the cells from the lineage. To find the overlapping driver genes between velocity methods, we selected the top 100 driver genes with the highest correlation to the Beta lineage from each method, and we utilized UpSet [40] to visualize the overlap between all groupings of methods.

To further investigate the two major overlapping groups of genes, we conducted a gene ontology analysis with Enrichr [41] utilizing gene set libraries 'BioCarta_2016,' 'GO_Biological_Process_2021,' 'KEGG_2019_Mouse,' 'Mouse_Gene_Atlas,' 'Panther_2016,' 'Reactome_2022,' 'Reactome_2016,' and 'WikiPathways_2019_Mouse.' We filter the resulting pathways based on adjusted p-value (less than 0.05) and number of genes (at least two overlapping genes). We plotted the top 20 pathways, by p-value, for each group of genes.

Sampling reads to simulate robustness to sequencing depth. To simulate robustness, we subsampled the reads, starting from the bam files for the zebrafish embryo 24 hours post fertilization. The full dataset contains four zebrafish embryos. For each embryo, we utilized samtools to subset a percentage (2, 12, 25, 50, 80, 95, and 98%) of the total reads, creating 5 randomly sampled replicates for each percent. We then ran Velocyto [6] v. 0.17.17 on each subsampled bam file for each embryo, with the reference genome used for the original dataset [26]. We combined the resulting count matrices for the four fish at each subsampling percentage and iteration and ran the five RNA velocity methods as outlined earlier in the methods on each of the resulting anndata created from subsampled reads.

We analyzed the directionality and robustness by comparing the resulting transition matrices, to the true (100% of the data) matrix, and subset the cell-cell transition matrix to the overlapping cells matching the subset transition matrix. We utilized the same workflow outlined in the method section 'velocity counts and generation of the velocity graph' to generate the matrix. We then computed the magnitude of the transition vector for each cell (np.linalg.norm), and the velocity vector (generated by RNA velocity in transcriptomic space). To analyze the directionality, we computed the cosine similarity for each cell between its transition vector generated by a proportion of the reads and the transition vector generated by 100% of the reads, repeated for each replicate. The results were averaged across cells and replicates before plotting. To plot the direction of the velocity arrows, we projected each subset's velocity vectors onto the UMAP created with the entire dataset, using the coordinates paired with each cell's location in the plot.

To ensure the evaluation of directionality was not largely affected by vectors with a small magnitude, we also computed a min/max weighting for each cosine similarity value, multiplying by the minimum magnitude of the two vectors and dividing by the maximum magnitude of the vectors.

We compared the variability within the five randomly sampled replicates at each proportion level by computing the cosine similarity of the transition vectors generated from each unique pair of replicates (totaling 10 pairs, for each of the five methods). To analyze the variance in directionality, we took the mean cosine similarity across the ten pairs of transition vector predictions for each cell and repeated the computation for each method, at each level of subset percentage.

Supporting information

S1 Fig. RNA Velocity methods and datasets, associated with Fig 1. a. RNA velocity UMAP projections for five methods, implemented in the pancreas dataset. b. RNA velocity UMAP projections for five methods, implemented in the zebrafish NMP (ZF NMP) dataset. c. RNA velocity UMAP projections for five methods, implemented in the zebrafish full embryo

24 hours post fertilization (ZF embryo 24hpf) dataset. d. ZF NMP UMAP colored by cell type annotations. e. ZF embryo 24hpf UMAP colored by cell type annotations. f. Pancreas UMAP colored by cell type annotations. g. Table of datasets used in the paper, including information about the organism, biological context, number of cells, and author reference.
(TIF)

S2 Fig. Number of cells per cell type for each dataset, associated with Fig 2. a. Number of cells per cell type for the ZF 24hpf whole-embryo dataset. b. Number of cells per cell type for the ZF NMP dataset. c. Number of cells per cell type for the pancreas dataset.
(TIF)

S3 Fig. Local consistency across the ZF NMP and pancreas datasets, associated with Fig 2. a. UMAP embeddings for ZF NMP dataset colored by the single-cell local consistency for each RNA velocity method. b. Local consistency distributions for the three top and bottom cell types from the ZF NMP dataset, as ranked by average local consistency. c. UMAP embeddings for pancreas dataset colored by the single-cell local consistency for each RNA velocity method. d. Local consistency distributions for the three top and bottom cell types from the pancreas dataset, as ranked by average local consistency.
(TIF)

S4 Fig. Pairwise Method Agreement UMAPs for ZF embryo 24hpf, associated with Fig 3. UMAP embedding for the ZF embryo 24hpf dataset with all unique pairs of method agreement.
(TIF)

S5 Fig. Pairwise Method Agreement UMAPs for pancreas, associated with Fig 3. UMAP embedding for the pancreas dataset with all unique pairs of method agreement.
(TIF)

S6 Fig. Pairwise Method Agreement UMAPs for ZF NMP, associated with Fig 3. UMAP embedding for the ZF NMP dataset with all unique pairs of method agreement.
(TIF)

S7 Fig. Additional pairwise method agreement distributions and agreement with median vector, associated with Fig 3. a. Boxplots with distributions of pairwise method agreement for each pair of method for each of the three datasets. b. Pairwise comparisons for six cell types from the ZF embryo 24hpf dataset across all methods. The heatmap shows the median method agreement across individual cells within a cell type for each pair of methods. c. UMAP embeddings for the pancreas for each RNA velocity method, colored by each method's agreement (2) with the median vector. d. UMAP embeddings for ZF NMPs for each RNA velocity method, colored by each method's agreement (2) with the median vector.
(TIF)

S8 Fig. Overlap of Driver Genes and CellRank Terminal States, associated with Fig 4. a. Macrostates identified by CellRank for scv-Sto, Velocyto, DeepVelo and scv-Dyn in the pancreas dataset. CellRank identified different macrostates for each dataset depending on the model's predictions (see Methods). b. Venn diagram with the percentage overlap across all methods and select groups indicated of the top 100 driver genes for the Epsilon lineage. Genes identified across all methods as top driver genes are labeled.
(TIF)

S9 Fig. Pathway analysis of overlapping genes identified within the top 100 driver genes for the beta lineage of the pancreas found for each velocity method, associated with Fig 4. a. Gene ontology of the 79 genes overlapping

for the beta lineage identified by DeepVelo, Velocyto, and scv-Sto. b. Gene ontology of the 55 genes overlapping for the beta lineage identified by UniTVelo and scv-Dyn.
(TIF)

S10 Fig. Velocity UMAPs with Subsets of Reads for DeepVelo, scv-Dyn and Velocyto, associated with Fig 5. a. UMAP embeddings of the ZF 24hpf whole-embryo with RNA velocity predictions from DeepVelo, calculated from subsets 2, 5, 12, 25, 50, 80, 95 and 100% of the reads. b. UMAP embeddings of the ZF 24hpf whole-embryo with RNA velocity predictions from scv-Dyn, calculated from subsets 2, 5, 12, 25, 50, 80, 95 and 100% of the reads. c. UMAP embeddings of the ZF 24hpf whole-embryo with RNA velocity predictions from Velocyto, calculated from subsets 2, 5, 12, 25, 50, 80, 95 and 100% of the reads.
(TIF)

S11 Fig. Velocity UMAPs with Subsets of Reads for scv-Sto and UniTVelo, associated with Fig 5. a. UMAP embeddings of the ZF 24hpf whole-embryo with RNA velocity predictions from scv-Sto calculated from subsets 2, 5, 12, 25, 50, 80, 95 and 100% of the reads. b. UMAP embeddings of the ZF 24hpf whole-embryo with RNA velocity predictions from UniTVelo, calculated from subsets 2, 5, 12, 25, 50, 80, 95 and 100% of the reads.
(TIF)

S12 Fig. Robustness comparison of magnitude and direction for subset reads, associated with Fig 5. a. Scatterplots comparing the magnitude of the transition vector calculated from the subset reads (x-axis) vs. 100% of the reads (y-axis) for each method. The columns correspond to different subsets, with increasing proportions of reads (2, 5, 12, 25, 50, 80, 95, 98%), and the rows correspond to different methods (DeepVelo, scv-Dyn, scv-Sto, UniTVelo, Velocyto). b. Comparison of directionality robustness for each method weighted by the min/max, calculated as the cosine similarity of the transition vector from the subset with the directionality from the transition vector calculated with 100% of the reads. We multiply each similarity score by the minimum magnitude of the two vectors divided by the maximum magnitude. The line plot shows the averaged value across all cells and subset iterations in the ZF 24hpf whole-embryo dataset. c. Median and InterQuartile Range (IQR - 25% and 75% percentile) of the cosine similarity between the transition vector from the subset with the directionality from the transition vector calculated with 100% of the reads, calculated across all cells and subset iterations in the ZF embryo 24hpf dataset.
(TIF)

S13 Fig. Variance in directionality in replicate subsamples, associated with Fig 6. a. We plot the median and the 25% and 75% percentile (range indicated by the gray bars) of the mean pairwise cosine similarity scores across the ten unique replicate pairs of transition vectors for each cell and repeat for each method, at each proportion level of subset reads (5, 12, 25, 50, 80, 90, 95, 98%). b. Boxplot distribution of mean pairwise cosine similarity scores across proportion levels of subset reads for each method separately. c. UMAPs where each cell is colored by the mean pairwise cosine similarity of transition vectors across the ten unique pairs of replicates. The columns correspond to different subsets, with increasing proportions of reads (2, 12, 25, 50, 90, 98%), and the rows correspond to different methods (DeepVelo, scv-Dyn, scv-Sto, UniTVelo, Velocyto).
(TIF)

Acknowledgments

We thank the Data Science and Royer teams of the Biohub for their helpful discussion. We are grateful for the review and feedback from Sandy Schmid, Jordão Bragantini, Joan Wong, and Yang-Joon Kim. We thank the Biohub and its donors, Priscilla Chan and Mark Zuckerberg, for funding this work. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Author contributions

Conceptualization: Sarah Ancheta, Alejandro Granados, Merlin Lange.

Data curation: Sarah Ancheta, Leah Dorman.

Formal analysis: Sarah Ancheta, Leah Dorman, Guillaume Le Treut, Abel Gurung, Alejandro Granados.

Funding acquisition: Greg Huber, Loïc A. Royer, Merlin Lang.

Investigation: Sarah Ancheta, Leah Dorman, Abel Gurung.

Methodology: Sarah Ancheta, Leah Dorman, Guillaume Le Treut, Abel Gurung, Loic Alain Royer, Alejandro Granados.

Project administration: Alejandro Granados, Merlin Lange.

Resources: Loïc A. Royer, Greg Huber, Merlin Lange.

Software: Sarah Ancheta, Leah Dorman, Guillaume Le Treut, Abel Gurung, Alejandro Granados.

Supervision: Alejandro Granados, Merlin Lange.

Visualization: Sarah Ancheta, Abel Gurung, Alejandro Granados, Merlin Lange.

Writing – original draft: Sarah Ancheta, Alejandro Granados, Merlin Lange.

Writing – review & editing: Sarah Ancheta, Merlin Lange.

References

1. Wagner DE, Klein AM. Lineage tracing meets single-cell omics: opportunities and challenges. Nat Rev Genet. 2020;21(7):410–27. https://doi.org/10.1038/s41576-020-0223-2 PMID: 32235876
2. 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. https://doi.org/10.1186/s12864-018-4772-0 PMID: 29914354
3. 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. https://doi.org/10.1038/nbt.2859 PMID: 24658644
4. Haghverdi L, Büttner M, Wolf FA, Buettner F, Theis FJ. Diffusion pseudotime robustly reconstructs lineage branching. Nat Methods. 2016;13(10):845–8. https://doi.org/10.1038/nmeth.3971 PMID: 27571553
5. Saelens W, Cannoodt R, Todorov H, Saeys Y. A comparison of single-cell trajectory inference methods. Nat Biotechnol. 2019;37(5):547–54. https://doi.org/10.1038/s41587-019-0071-9 PMID: 30936559
6. 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. https://doi.org/10.1038/s41586-018-0414-6 PMID: 30089906
7. 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. https://doi.org/10.1038/s41587-020-0591-3 PMID: 32747759
8. 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. https://doi.org/10.1038/s41592-021-01346-6 PMID: 35027767
9. Wyler E, Franke V, Menegatti J, Kocks C, Boltengagen A, Praktiknjo S, et al. Single-cell RNA-sequencing of herpes simplex virus 1-infected cells connects NRF2 activation to an antiviral program. Nat Commun. 2019;10(1):4878. https://doi.org/10.1038/s41467-019-12894-z PMID: 31653857
10. Kimmel JC, Yi N, Roy M, Hendrickson DG, Kelley DR. Differentiation reveals latent features of aging and an energy barrier in murine myogenesis. Cell Rep. 2021;35(4):109046. https://doi.org/10.1016/j.celrep.2021.109046 PMID: 33910007
11. Aissa AF, Islam ABMMK, Ariss MM, Go CC, Rader AE, Conrardy RD, et al. Single-cell transcriptional changes associated with drug tolerance and response to combination therapies in cancer. Nat Commun. 2021;12(1):1628. https://doi.org/10.1038/s41467-021-21884-z PMID: 33712615
12. Cuevas-Diaz Duran R, González-Orozco JC, Velasco I, Wu JQ. Single-cell and single-nuclei RNA sequencing as powerful tools to decipher cellular heterogeneity and dysregulation in neurodegenerative diseases. Front Cell Dev Biol. 2022;10:884748. https://doi.org/10.3389/fcell.2022.884748 PMID: 36353512
13. Rossi G, et al. Capturing cardiogenesis in gastruloids. Cell Stem Cell. 2021;28:230-240.e6.
14. Gorin G, Fang M, Chari T, Pachter L. RNA velocity unraveled. PLoS Comput Biol. 2022;18(9):e1010492. https://doi.org/10.1371/journal.pcbi.1010492 PMID: 36094956
15. Zheng SC, Stein-O'Brien G, Boukas L, Goff LA, Hansen KD. Pumping the brakes on RNA velocity by understanding and interpreting RNA velocity estimates. Genome Biol. 2023;24(1):246. https://doi.org/10.1186/s13059-023-03065-x PMID: 37885016

16. Bergen V, Soldatov RA, Kharchenko PV, Theis FJ. RNA velocity-current challenges and future perspectives. Mol Syst Biol. 2021;17(8):e10282. https://doi.org/10.15252/msb.202110282 PMID: 34435732
17. Elowitz MB, Levine AJ, Siggia ED, Swain PS. Stochastic gene expression in a single cell. Science. 2002;297(5584):1183–6. https://doi.org/10.1126/science.1070919 PMID: 12183631
18. Gao M, Qiao C, Huang Y. UniTVelo: temporally unified RNA velocity reinforces single-cell trajectory inference. Nat Commun. 2022;13(1):6586. https://doi.org/10.1038/s41467-022-34188-7 PMID: 36329018
19. Cui H, Maan H, Vladoiu MC, Zhang J, Taylor MD, Wang B. DeepVelo: deep learning extends RNA velocity to multi-lineage systems with cell-specific kinetics. Genome Biol. 2024;25(1):27. https://doi.org/10.1186/s13059-023-03148-9 PMID: 38243313
20. Hastie T, Tibshirani R, Friedman J. The elements of statistical learning. New York, NY: Springer; 2009. https://doi.org/10.1007/978-0-387-84858-7
21. Xu J, Hsu DJ, Maleki A. Global analysis of expectation maximization for mixtures of two Gaussians. In: Advances in neural information processing systems. Curran Associates, Inc.; 2016.
22. Neamtu M. Splines on surfaces. In: Farin G, Hoschek J, Kim M-S, editors. Handbook of computer aided geometric design. Amsterdam: North-Holland; 2002. 229–53. https://doi.org/10.1016/B978-044451104-1/50010-1
23. Job S. Towards causal classification: A comprehensive study on graph neural networks. 2024. http://arxiv.org/abs/2401.15444
24. Bonner-Weir S, Toschi E, Inada A, Reitz P, Fonseca SY, Aye T, et al. The pancreatic ductal epithelium serves as a potential pool of progenitor cells. Pediatr Diabetes. 2004;5 Suppl 2:16–22. https://doi.org/10.1111/j.1399-543X.2004.00075.x PMID: 15601370
25. 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. https://doi.org/10.1242/dev.173849 PMID: 31160421
26. Lange M. Zebrahub – multimodal zebrafish developmental atlas reveals the state-transition dynamics of late-vertebrate pluripotent axial progenitors. 2023. https://doi.org/10.1101/2023.03.06.531398
27. Magnuson MA, Osipovich AB. Pancreas-specific cre driver lines and considerations for their prudent use. Cell Metab. 2013;18:9–20.
28. Qiao C, Huang Y. Representation learning of RNA velocity reveals robust cell transitions. Proc Natl Acad Sci U S A. 2021;118(49):e2105859118. https://doi.org/10.1073/pnas.2105859118 PMID: 34873054
29. Li J, Pan X, Yuan Y, Shen HB. TFvelo: gene regulation inspired RNA velocity estimation. 2023. https://doi.org/10.1101/2023.07.12.548785
30. Li T, Shi J, Wu Y, Zhou P. On the mathematics of RNA velocity I: theoretical analysis. CSIAM-AM. 2021;2(1):1–55. https://doi.org/10.4208/csiam-am.so-2020-0001
31. Bayner JS. Developmental biology. Eleventh edition. Yale J. Biol. Med. 2017. 90, 697–69.
32. Fior R, Maxwell AA, Ma TP, Vezzaro A, Moens CB, Amacher SL, et al. The differentiation and movement of presomitic mesoderm progenitor cells are controlled by Mesogenin 1. Development. 2012;139(24):4656–65. https://doi.org/10.1242/dev.078923 PMID: 23172917
33. Sambasivan R, Steventon B. Neuromesodermal progenitors: a basis for robust axial patterning in development and evolution. Front Cell and Develop Biol. 2021;8.
34. Ebrahim N, Shakirova K, Dashinimaev E. PDX1 is the cornerstone of pancreatic β-cell functions and identity. Front Mol Biosci. 2022;9:1091757. https://doi.org/10.3389/fmolb.2022.1091757 PMID: 36589234
35. Marot-Lassauzaie V, Bouman BJ, Donaghy FD, Demerdash Y, Essers MAG, Haghverdi L. Towards reliable quantification of cell state velocities. PLoS Comput Biol. 2022;18(9):e1010031. https://doi.org/10.1371/journal.pcbi.1010031 PMID: 36170235
36. 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. https://doi.org/10.1038/s41592-021-01336-8 PMID: 34949812
37. Li C, Virgilio MC, Collins KL, Welch JD. Multi-omic single-cell velocity models epigenome-transcriptome interactions and improves cell fate prediction. Nat Biotechnol. 2023;41(3):387–98. https://doi.org/10.1038/s41587-022-01476-y PMID: 36229609
38. Wu S, Schmitz U. Single-cell and long-read sequencing to enhance modelling of splicing and cell-fate determination. Comput Struct Biotechnol J. 2023;21:2373–80. https://doi.org/10.1016/j.csbj.2023.03.023 PMID: 37066125
39. Zhang C, Fang Y, Chen W, Chen Z, Zhang Y, Xie Y, et al. Improving the RNA velocity approach with single-cell RNA lifecycle (nascent, mature and degrading RNAs) sequencing technologies. Nucleic Acids Res. 2023;51(22):e112. https://doi.org/10.1093/nar/gkad969 PMID: 37941145
40. Lex A, Gehlenborg N, Strobelt H, Vuillemot R, Pfister H. UpSet: visualization of intersecting sets. IEEE Trans Vis Comput Graph. 2014;20(12):1983–92. https://doi.org/10.1109/TVCG.2014.2346248 PMID: 26356912
41. Chen EY, Tan CM, Kou Y, Duan Q, Wang Z, Meirelles GV, et al. Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool. BMC Bioinformatics. 2013;14:128. https://doi.org/10.1186/1471-2105-14-128 PMID: 23586463