RNA velocity of single cells

Gioele La Manno[1,2], Ruslan Soldatov[3], Amit Zeisel[1,2], Emelie Braun[1,2], Hannah Hochgerner[1,2], Viktor Petukhov[3,4], Katja Lidschreiber[5], Maria E. Kastriti[6], Peter Lönnerberg[1,2], Alessandro Furlan[1], Jean Fan[3], Lars E. Borm[1,2] Zehua Liu[3], David van Bruggen[1], Jimin Guo[3], Xiaoling He[7], Roger Barker[7], Erik Sundström[8], Gonçalo Castelo-Branco[1], Patrick Cramer[5,9], Igor Adameyko[6], Sten Linnarsson[1,2]† and Peter V. Kharchenko[3,10]†

Supplementary Note 2

Considerations for accurate determination and visualization of RNA velocity

Table of Contents

Section 1. Illustrations of theoretical model with simulations

In this section we illustrate the theory discussed in the Supplementary Note 1 section “Theoretical description of RNA velocity”, by plotting solutions to the time-dependent differential equations (1) and (2) under different parameter settings. Figure 1 below shows the solution of the rate equations (solid lines), the master equation (shaded) as well as individual realizations of the stochastic process described the master equation (dots). In each case, we show the behavior of a single gene, as our model does not account for gene interactions.

We first illustrate the result of a step change in the transcription rate α, starting from zero and with abundance zero (Figure 1a). The expected values (given by the rate equations) of unspliced and spliced molecules rise rapidly and converge on the new equilibrium values, with unspliced rising before spliced. However, in any specific realization of this process, there will be stochastic variation around the expectation, given by the master equation (Figure 1a, shaded region) which gives the probability, at any timepoint, of observing n molecules of RNA. When the transcription rate returns to zero, the expected values of both spliced and unspliced mRNA molecules return to zero at an exponential rate.

Viewing this same process as a phase portrait (Figure 1b), reveals how the shapes of unspliced and spliced mRNA expectations are related to the diagonal line representing γ. A single realization of the process, with 500 cells sampled uniformly in time, shows how most cells (observations) appear close to the two steady-state points (corresponding to $\alpha = 20$ and $\alpha = 0$. Intermediate points are rarer, because the approach to the steady state in both cases is exponential.

Next, examining the effect of α (Figure 1c) shows that it sets the position of the steady state along the diagonal given by γ. Thus, if expression levels are regulated by the transcription rate, rather than degradation, steady state equilibria are expected to line up along the diagonal where $\gamma = \frac{u}{s}$ and $\alpha = u$. In contrast, changes to γ change the location of the steady state (Fig. 1d), but it remains true that $\gamma = \frac{u}{s}$ and $\alpha = u$.

More complex scenarios can also be accounted for by allowing α to vary over time in more complex ways. For example, oscillatory gene expression can be described as a transcription rate α that varies according to a trigonometric function, e.g. $\alpha = 25(1 - \cos(t))$, resulting in oscillating abundances of spliced and unspliced molecules (Figure 1e). In this case, there is no strict steady state, but γ can still be obtained as the centerpoint (attractor) of the oscillation.

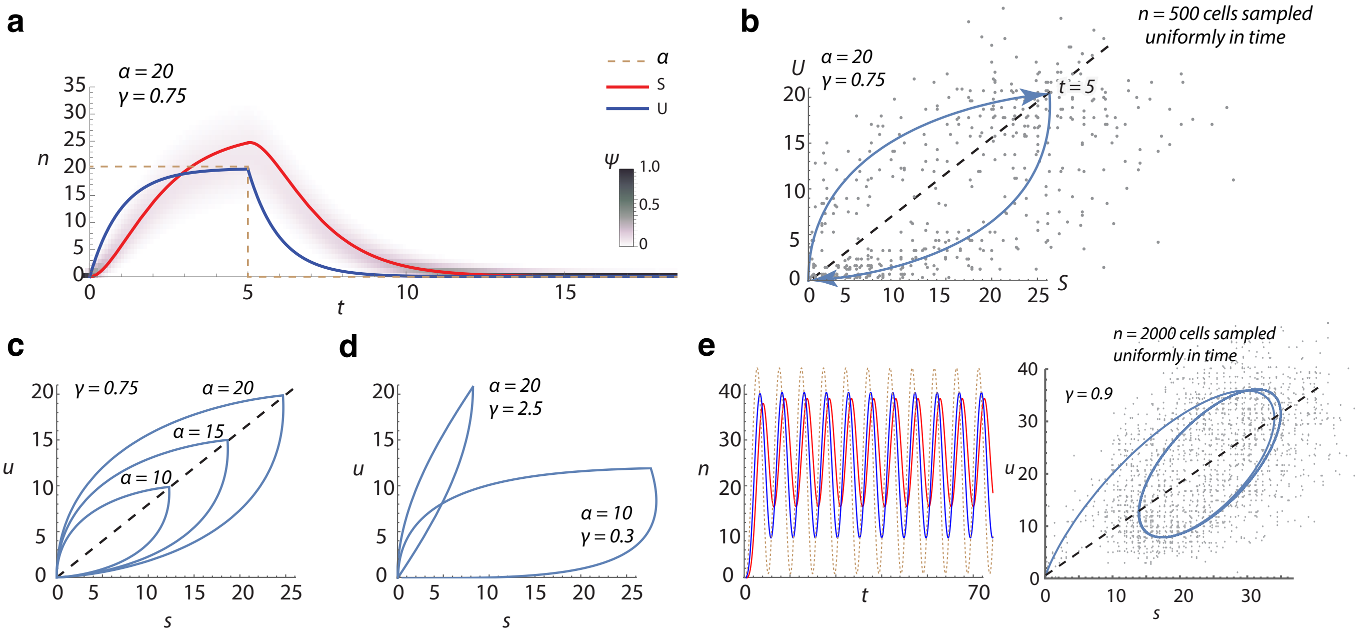

Figure 1. Master equation and model predictions of different dynamics. a. A typical dynamics obtained by our model of transcription as in Fig 1a, here we compute the full probability distribution over the counts described by the master equation $\Psi(\mathrm{s,u,t})$. **b.** A realization of the master equation and the expected value of the u-s relationship shown as a phase portrait. Dots show individual samples (n=500) from the master equation shown in (a), sampled uniformly in time. Jitter was added for clarity, since samples are strictly integers. **c.** The effect of different transcription rates (α) on the u-s dynamics. **d.** Dynamics corresponding to different degradation rates. **e.** Left, the solution of the rate equations for an oscillating transcription rate (*e.g.* like in a biological clock). Right, the phase portrait of the same solution with a realization of Ψ.

Section 2. The rationale behind the extreme quantile fit procedure

To achieve more accurate estimation of gene-specific steady-state coefficient γ, the gene-relative estimates use regression based on the cells found in the extreme quantiles of expression. Specifically, given a quantile value α, the quantile fit uses cells with $i{:}\left\{\frac{s_i}{S} \geq \alpha\right\} \cup \left\{\frac{s_i}{S} \leq 1-\alpha\right\}$, where S is the maximal observed expression magnitude of that gene in the dataset. Alternatively, diagonal quantiles are calculated based on a normalized sum of spliced and unspliced expression magnitude ($x_i = s_i/S + u_i/U$), where U is the maximal unspliced expression. The properties of the quantile fit under several common scenarios are illustrated in Figure 2. In cases when a full cycle of a gene is observed, the regular regression fit and the extreme quantile fit give similar results. The quantile fit results in more exact estimates when only up- or down-regulation of a gene is observed. In more extreme cases, where the gene is not observed in steady state, the quantile estimates will result in more conservative velocity estimates. For instance, in Figure 2c, the quantile fit will report gene as being downregulated, however at lower magnitude relative to the true slope. In contrast, a regular regression fit will show gene as being up-regulated in some regions. The under-estimation of velocity in such partial-observation cases can be corrected using gene-structure model (Supp. Figure 4).

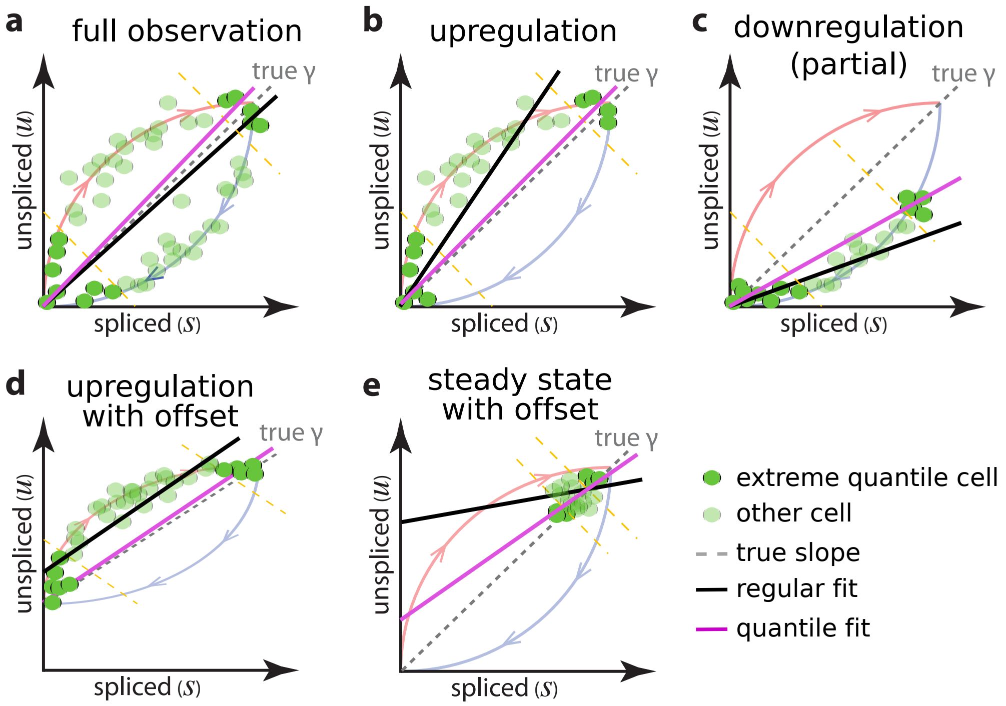

Figure 2. Illustrations of quantile fitting procedure. The schematic drawings illustrate the difference of regular (black) and quantile regression (pink) fits of gamma coefficient under different scenarios, including observation of full up- and down-regulation cycle (a), observations mostly away from steady state (b), far from steady state (c). Fits with non-specific offsets are shown for upregulation and steady-state cases in (d) and (e).

To account for contribution of extraneous transcripts, the fitting model allows for an offset. Several options are available for determining offsets. In the case of regular regression fit, the offset can be determined as a mean of a lower quantile. In case of a quantile fit, the regression model is fit with an intercept, which is then used as gene offset. We note that in steady state situations (Figure 2e), regression fits can produce unstable angles. The quantile fit will result in a positive gamma slope. However, such genes are typically filtered out, because of either low values of gamma or because they fail to meet minimum requirements for correlation between unspliced and spliced abundances (Pearson r>0.05 by default). Even in the cases when such steady state genes are not filtered out, we expect the residuals and the resulting velocity estimates to be randomly distributed among cells, and thus have little impact on the low-dimensional projections.

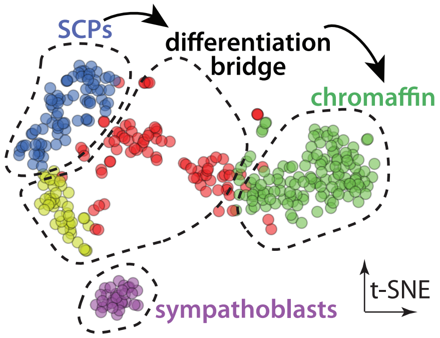

Figure 3. Annotated t-SNE embedding of the chromaffin E12.5 dataset. (n=385 cells)

The alternative procedure for fitting offsets relies on “spanning reads” – reads that cover both exonic and intronic sequence of the gene, and have higher likelihood of originating from the underlying gene as opposed to some extraneous transcript. Such reads are sufficiently abundant in the SMART-seq2 data to allow fitting offset values by contrasting spanning and intronic reads (Figure 4).

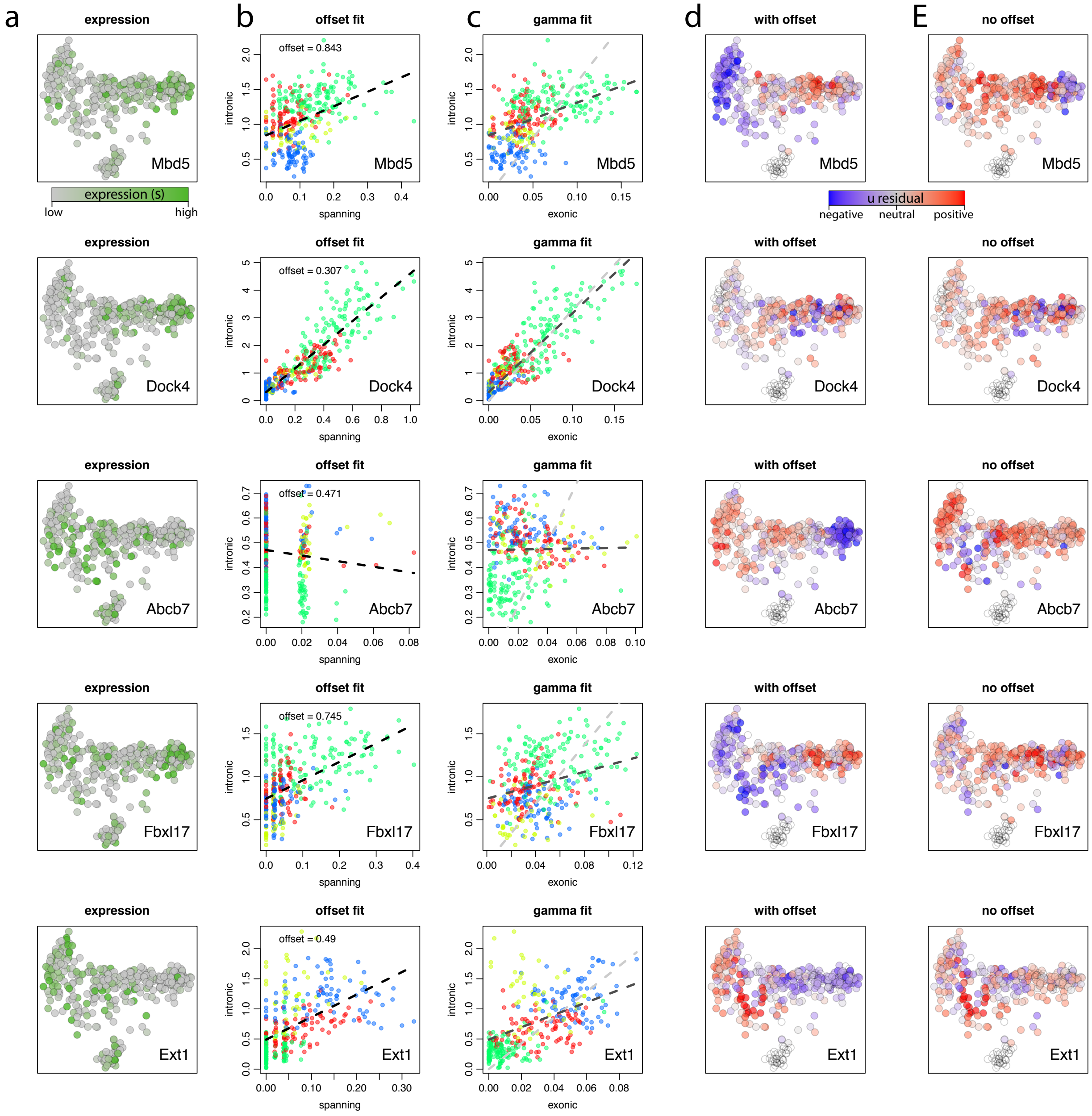

Figure 4. Fitting offset of intronic read counts using spanning reads on the chromaffin E12.5 dataset. Fitting of non-specific unspliced count offset using spanning read counts is shown for five example genes (rows). Gene name is given in the lower right corner of each plot. For each gene (row), the first column **(a)** shows expression (spliced count abundance) of the gene using t-SNE layout (see Figure 3). The second panel **(b)** shows a scatter plot illustrating the observed dependence between the spanning (x axis) and intron-only (y axis) read counts. The dashed line shows the regression fit that is used to determine the y axis intercept (intronic read count offset). The third panel **(c)** shows relationship between exonic and intronic (intron-only) counts. The dashed black line shows a gamma fit using the intronic count intercept determined from panel b, and grey dashed line uses zero intercept. The fourth column **(d)** shows unspliced count residuals (basis of the subsequent velocity estimates) calculated using spanning-read based offset from b. The last column **(e)** shows residuals calculated using default zero-offset. Genes with high offset values were chosen as examples.

Section 3. The range of expression regimes acceptable for the proposed model

To characterize the theoretical range of gene expression regimes and parameter settings under which the implemented velocity estimation procedure can function, we simulated spliced/unspliced RNA dynamics using a wide range of time-dependent parameter settings and compared the *velocyto* estimates with the ground truth. Importantly, to characterize the stability of the *velocyto* estimates beyond the assumed simple models of gene expression behavior, we have abolished the assumption of the constant rates (see Theory section in Supplementary Note 1), simulating data for cases where rates change (smoothly) as a function of time. Such parametrization yields more complex situations that do not allow for perfect fits by the simplified constant-rate model implemented in the *velocyto*. In other words, we have tested how deviations from the basic assumptions break the *velocyto* estimation procedure.

In testing the performance of *velocyto* in recovering correct RNA velocity estimates we covered a wide parameter space. Specifically, we devised a minimal set of nine parameters (Figure 5a) describing dynamic gene behavior. To generate simulations, the values of these parameters were drawn from the prior distributions (Figure 5b) under general assumption of parameter independence, with only exception being scale and variance parameters that were considered to be correlated. The prior distributions were chosen to yield realistic physical scenarios, however as we see from the results some unrealistic border cases were drawn as well. The simulated datasets, composed of 7000 independently sampled genes were processed by the *velocyto* pipeline in the same way as real datasets, including size normalization, dimensionality reduction and cell kNN pooling (feature selection was not performed).

The analysis allowed the identification of the regions of the parameter space where the model is more error prone and others where it performs particularly well (Figure 5). First, we confirmed that scenarios where we observe both up- and down-regulation phases of the gene (going through the steady state) are particularly informative for accurate estimation of velocity direction and magnitude. This is evident from the performance plot marginalized for the pairs of parameters “start-width” and “start-ramp up”: if “ramp up” and “width” are small, then complete up-regulation and down-regulation arcs end up being observed and γ fit performs well. In an opposite scenario, where α starts to raise at time 0 with a slow ramp up, and therefore the system is observed in a constantly accelerating state, it is difficult to estimate magnitude of velocity correctly. Another notable scenario is where we observe only the very beginning of an upregulation process (*i.e.* the “start” parameter is very large). In such scenario, the estimation will be overfit to the initial observations.

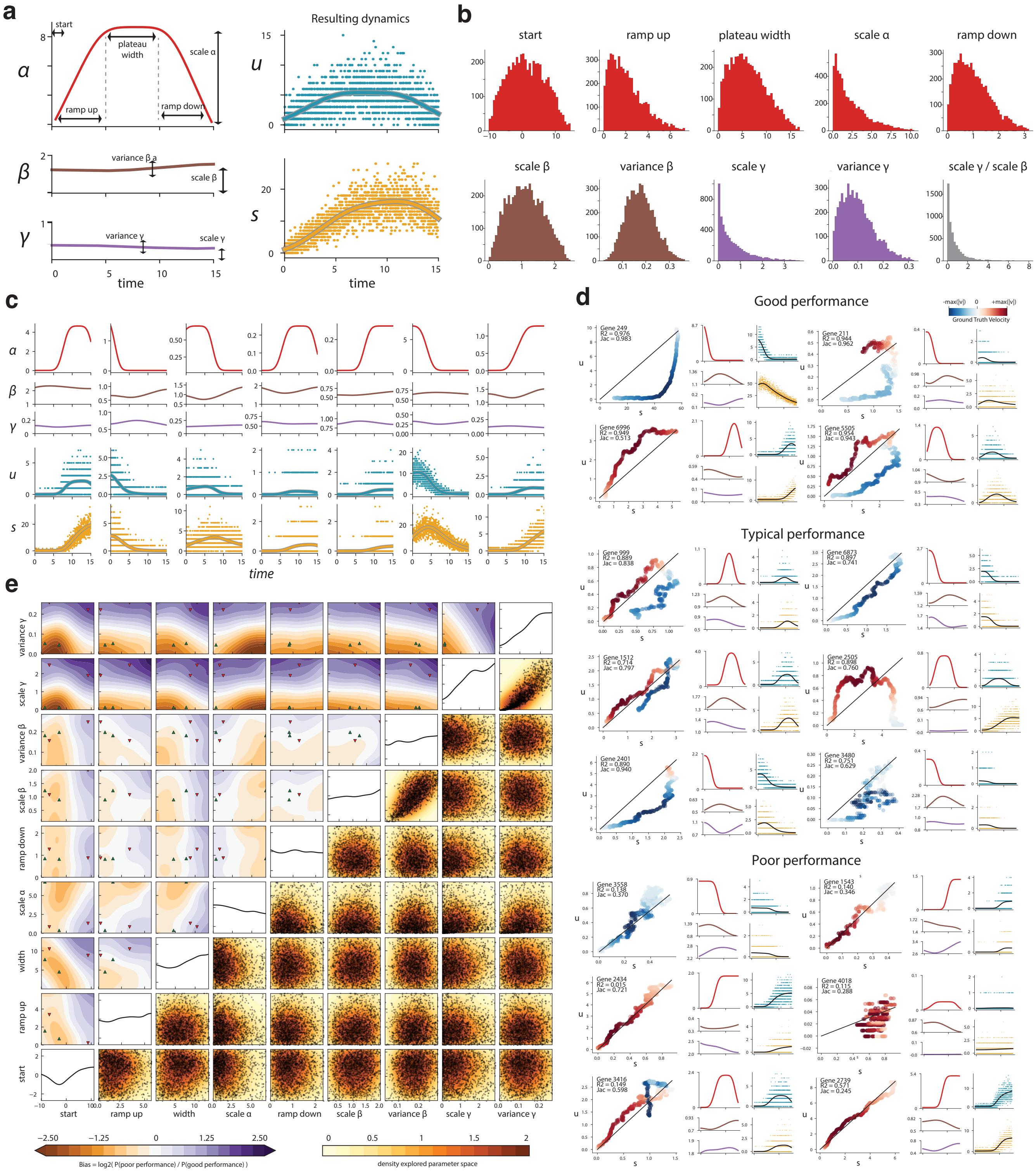

Figure 5. Range of acceptable gene expression model parameters for the *velocyto* pipeline a. Illustration of the simulation framework. On the left, a graphical representation of the 9 parameters used to simulate a wide range of possible dynamics. On the right, the resulting expression dynamic generated by the parameter profiles shown on the left. (n=1500) **b.** Histograms of the parameters that were used to simulate data of 7000 genes. **c.** Randomly selected

examples of dynamics generated by the simulation. The standard *velocyto* analysis pipeline was run on the simulated data and the performance of the algorithm on each simulated gene was ranked. **d.** Randomly drawn examples of high performance (> 80th percentile), typical performance (from 20th to 80th percentile) and low performance (lower than 20th percentile) as determined by correlation coefficient (shown on the left) and fraction of concordant signs (on the right). Plots on the left show spliced and unspliced dynamics, colored by ground truth velocity, on the right the parameters profiles that generate the dynamics. **e.** Systematic performance evaluation on the whole parameter space. Lower triangle showing the parameter space explored. Diagonal is showing the bias marginalized for individual parameters. Upper triangle is showing the bias for different parameter pairs. The red and green symbols mark the values of parameters that generated by two high and low performance genes respectively (the first two of the random selection as shown in (d)).

The simulations showed sensitivity of performance to the magnitude and variance of γ (modeled to be correlated to avoid unrealistic situations). For high γ, the curvature of the phase portrait with respect of the unspliced magnitude is reduced, providing little dynamic range for the velocity residuals and making such estimates sensitive to stochastic fluctuations. This corresponds to border-case scenarios where the spliced transcripts are degraded too rapidly to observe their accumulation. This behavior is worsened by increase of the scale of β and dampened by the increase of α.

Simulation details:

For each realization of the nine parameters, the system of differential equations was solved by numerical integration using the function *scipy.integrate.odeint* a python interface to LSODE (Livermore Solver for Ordinary Differential equation). Given the analytical results on the master equation provided by the theory (see Supplementary Note 1, theory section), we could draw a realization of the dynamical system by simply drawing a sample from a Poisson distribution with expectation equal to the solution of differential equations. In this way we simulated the spliced and unpliced expression of 3000 cells and 7000 genes. This data was used as input of a standard *velocyto* analysis pipeline. After velocity estimation we used two different scores to evaluate the performance of *velocyto* compared to the ground truth velocity: (1) the correlation coefficient of the estimated velocity and ground truth velocity and (2) the fraction of concordant signs:

$$Jac_g = \frac{\sum_{cell=1}^{N} f(\hat{v}_{cell,g}, v_{cell,g})}{N}$$

$$where \quad f(x,y) = \begin{cases} 0 & sgn(x) \neq sgn(y) \\ 1 & sgn(x) = sgn(y) \end{cases}$$

All the genes were ranked using these two scores and examples were randomly chosen from the top 20% and bottom 20% scoring genes. To evaluate the effect of different parameters on the estimation performance we defined two set of genes on the basis of the velocity estimation matching with the ground truth, those were defined as the set of genes whose both scores ranked above the 75th ("good performing" genes) or below the 25th percentile and bottom 25th percentile ("poorly performing" genes). To generate the plots shown in Figure 5, the density in parameter space of both good performing and poorly performing genes was determined by kernel density estimation (*scikit-learn* implementation) and the log-ratio of the densities of the two groups was reported as a measure of the increase tendency of estimating the velocity poorly.

Section 4. Gene-specific velocity estimation failures and mechanisms

As illustrated by Supp. Figure 8, while velocity estimates for most genes show positive correlation with the empirically-estimated gene expression derivatives, the velocity estimation procedure fails for a smaller subset of genes. While some of this may be attributed to complex dynamics regulating nascent transcription, splicing, and degradation (as in Figure 4d of the main manuscript), we can identify specific classes of genes for which the velocity estimation ends up being inaccurate.

Some genes show strikingly different gamma coefficients within different populations (see also Supp. Figure 3). Most such variation is observed between very different tissues, however, such examples can be found even within closely related subpopulatoins, such as the ones captured in the chromaffin differentiation or hippocampus development datasets. Some such occurrences are tied to shifts in expression of alternative splice isoforms between the measured subpopulations (Figure 6a-f, Figure 7a-d), which would directly impact the ratio of unspliced and spliced molecules. In other cases, however, no obvious alternative splicing difference can be detected (Figure 7g,h), suggesting that other mechanisms, such as subpopulation-specific control of the degradation rates may be in play.

The unspliced/spliced ratios can also be skewed by presence of extraneous transcripts, such as non-coding RNAs that can be found in many intronic regions (Figure 6g,h). If the expression of such extraneous transcripts remains constant or is randomized with respect to the underlying biological process, then their contribution can often be controlled for by the offset parameter. However, if the extraneous transcripts are also differentially regulated throughout the measured biological process, that can lead to erroneous velocity estimates. In many cases, presence of a high-expressing extraneous transcript will result in an atypical phase portrait and a low overall correlation between unspliced and spliced signals. Such low-correlated genes are filtered out by the *velocyto* pipeline by default.

A different class of errors is associated with velocity estimates of genes observed far away from their steady state. These are typically genes that are either induced very late in the observed trajectories, and thus are seen only with increased unspliced/spliced ratios, or genes that are already being actively downregulated in the earliest parts of the observed trajectories, and are hence seen with low or absent unspliced abundance. Examples of such genes can be found in most datasets (Figure 6i-l, Figure 7i). For such difficult cases, we show that equilibrium slopes gamma can be estimated based on gene structure parameters (Supp. Figure 4).

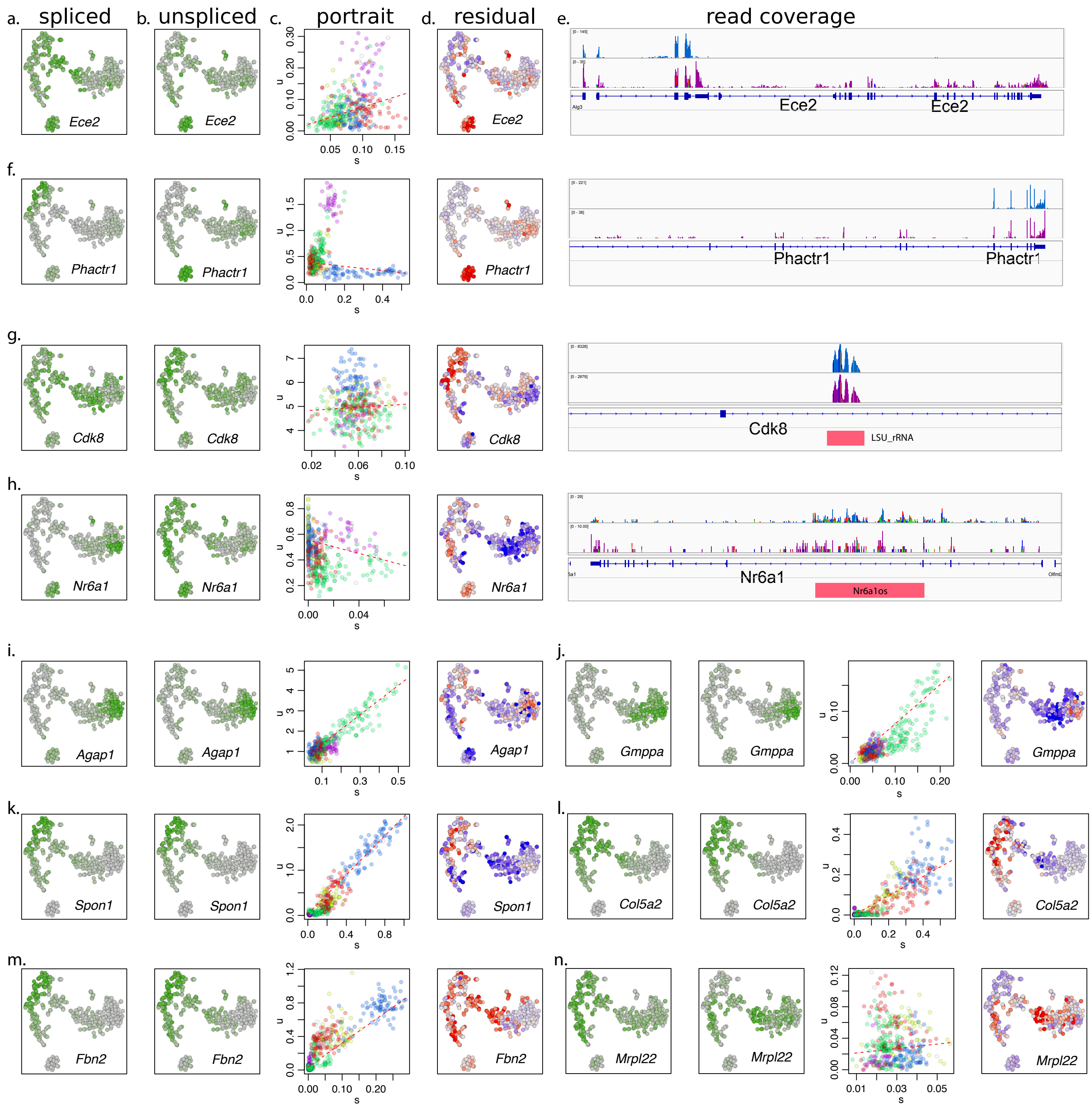

Figure 6. Examples of genes representing different modes of velocity estimation errors. Taken from the chromaffin differentiation E12.5 dataset, for each example gene, four panels show patterns of spliced (a) and unspliced (b) expression, phase portrait with gamma fit (c), and the residual (d). Browser screenshot (e) showing read intensity profiles for two subpopulations (blue – SCPs, purple – sympathoblasts) are shown for some genes. *Ece2* and *Phactr1* (f) examples show impact of alternative splicing, which results in very different gamma coefficients for different subpopulations of cells, violating the assumptions of the model. (g,h) show examples of extraneous transcripts (highlighted with red blocks on the browser screenshots) that increase offset and distort the phase portrait. Note that such genes would be normally filtered out because of the poor *u* vs. *s* correlation. (i,j) show examples of late genes, observed far away from the steady state point. (k,l) show examples of early genes that

are also observed only far from the steady state. (m,n) show complex examples that do not fall into well-defined failure categories.

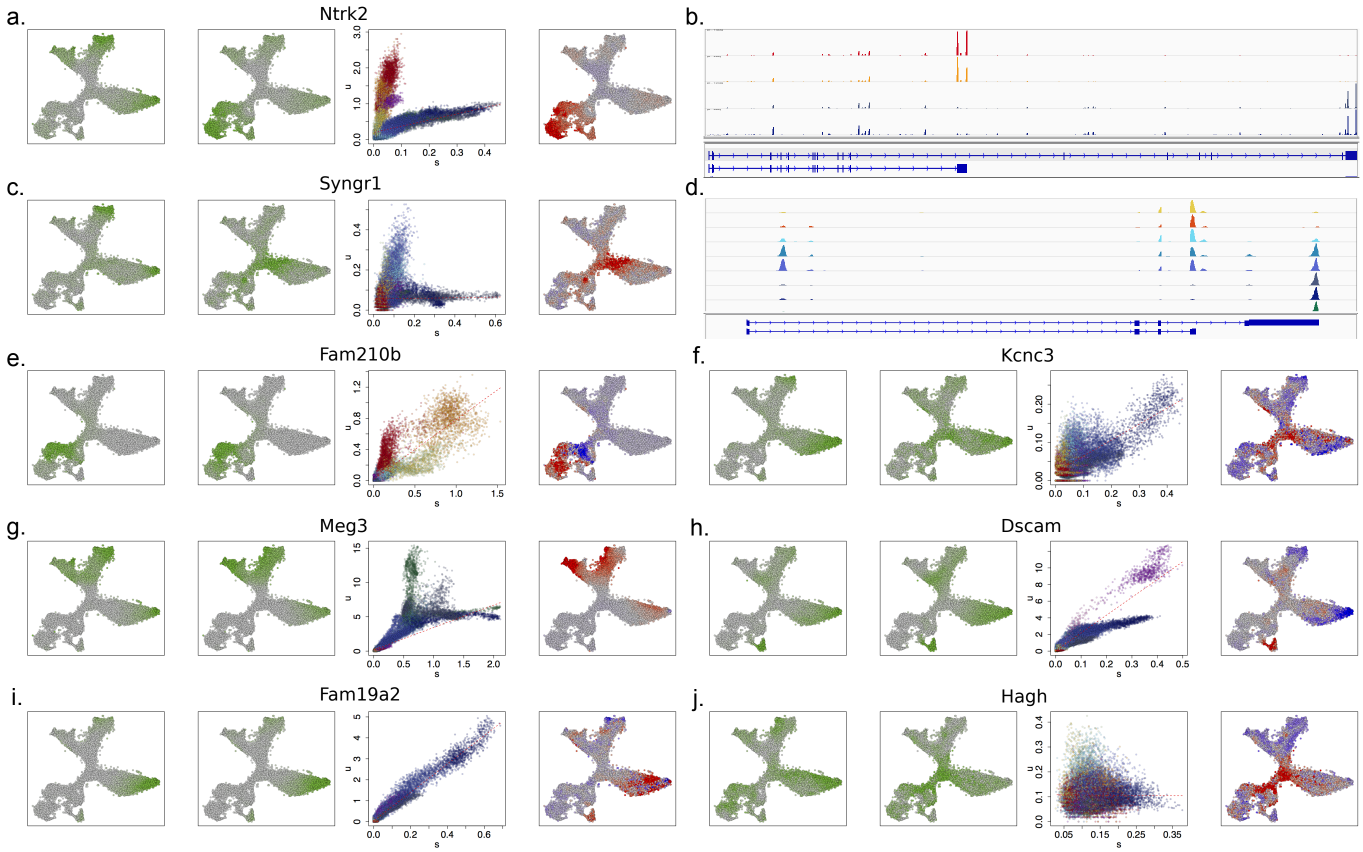

Figure 7. Examples of different types of velocity estimation errors on the hippocampal dataset. (a,b) Example of a gene exhibiting different gamma slopes within different populations, driven by alternative 3'UTR usage. **(c,d)** A gene exhibiting gradual shift of the dominant 3'UTR during development, which also manifests itself as two distinct gamma slopes on the phase portrait. **(e,f)** Genes showing mixed or opposing phase portrait curvatures. **(g,h)** Examples of multiple slope trends in the phase portrait, that do not appear to be explained by an obvious alterative splicing pattern. **(i)** An example of a late-expressing gene, observed far from the steady state. **(j)** Example of a gene with uncorrelated spliced-unspliced pattern. Such genes are normally filtered out by the *velocyto* pipeline.

Section 5. Illustrations of different velocity fits and visualizations on chromaffin data

Various corrections can be considered when estimating multi-dimensional velocity vectors, and visualizing them on two-dimensional plots. In this section we use chromaffin differentiation E12.5 dataset to illustrate the results of different such procedures. Visualization of velocity estimates in PCA space are shown in Figure 8, including estimates based on individual cells (without pooling of information from neighboring cells), with k-nearest gene clustering, and with different ways of estimating gene-specific offsets. The velocity pattern is generally robust for all these estimates.

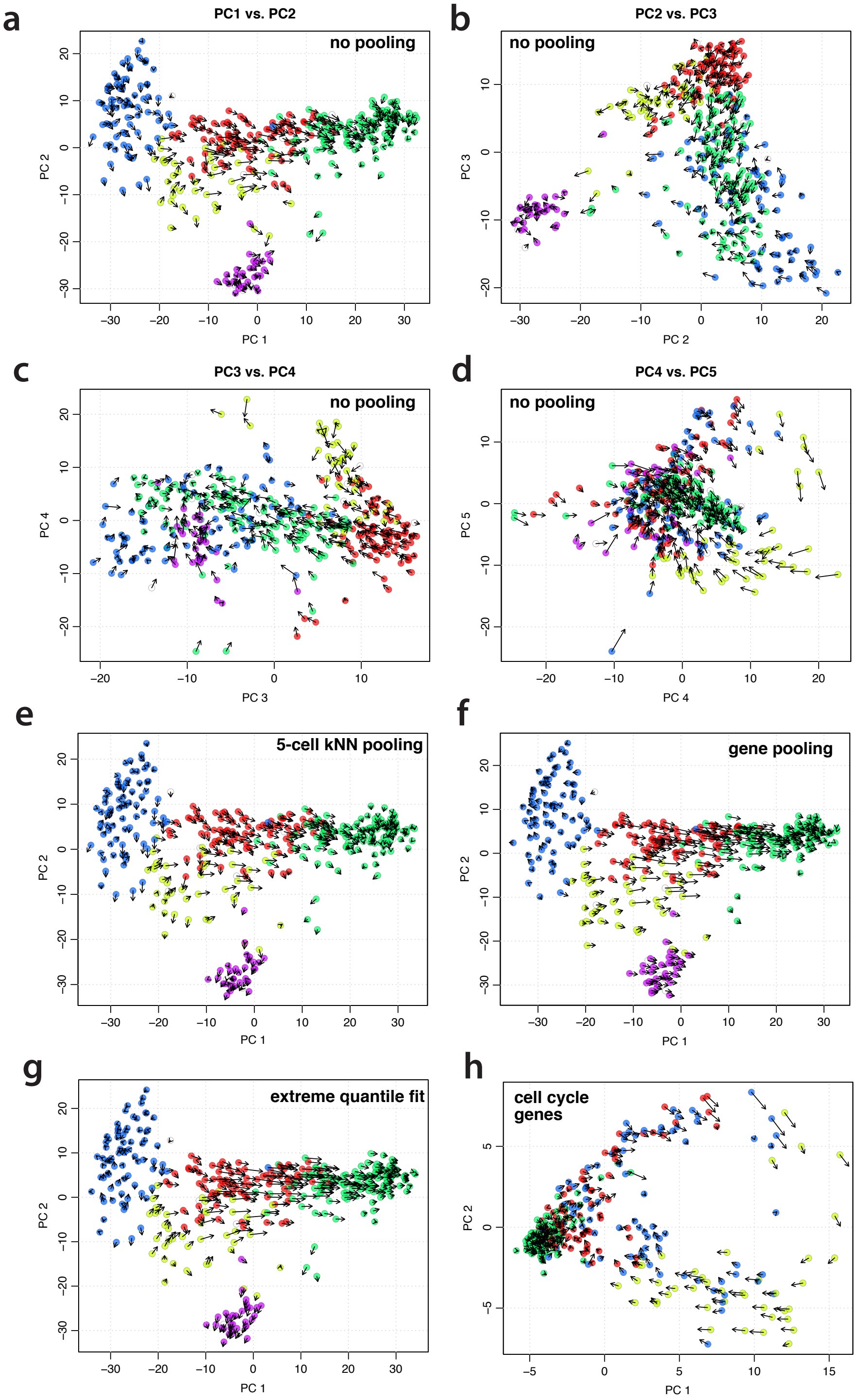

Figure 8. PCA visualization of chromaffin E12.5 velocities. a-d. Projections on the first five principal components are shown. Please refer to Figure 3 for the annotation of the subpopulations. The velocity was estimated using gene-relative fit for individual cells (*i.e.* without cell or gene pooling). Overall, PC1 captures the main chromaffin differentiation axis, PC2 captures separation between sympathoblasts and other cell types, PC3 separates bridge-specific (red) cells from others, PC4 and PC5 together capture cell cycle signature of the cycling bridge cells (yellow) - as seen in the last (PC4 vs. PC5) panel. **e.** Gene-relative velocity estimates are shown with k=5 cell kNN read pooling. **f.** Velocities estimated pooling reads across neighboring cells (kcells=5) and well-correlated genes (kgenes=20). **g.** Velocity estimates, with γ slope and offset fit using only cells within the top/bottom 2% expression quantile of each gene. As such approach works robustly on smoothed data, cell kNN pooling (kcells=5) was used in calculating the estimates. **h.** To emphasize cell cycle trends, velocity estimates from the previous panel were subset to include only cell cycle-related. The genes were selected using GO annotations. The resulting observed and extrapolated states were visualized by projecting on the first two PCs. Sympathoblast cells, which also undergo cell cycle within the dataset, were excluded from this visualization.

While PCA space allows for straightforward projection of the velocity vectors, PCA embeddings are generally not effective for visualizing samples with high complexity of subpopulations. Joint embedding of current and extrapolated cell states using t-SNE can be effective for some datasets (see Figure 9), however in some cases can be sensitive to gene-specific errors in velocity estimation. For that reason, we have devised a neighborhood-based projection procedure that can be used to visualize velocity on pre-defined embeddings (see Figure 2h,i of the main manuscript, as well as Section 11 below).

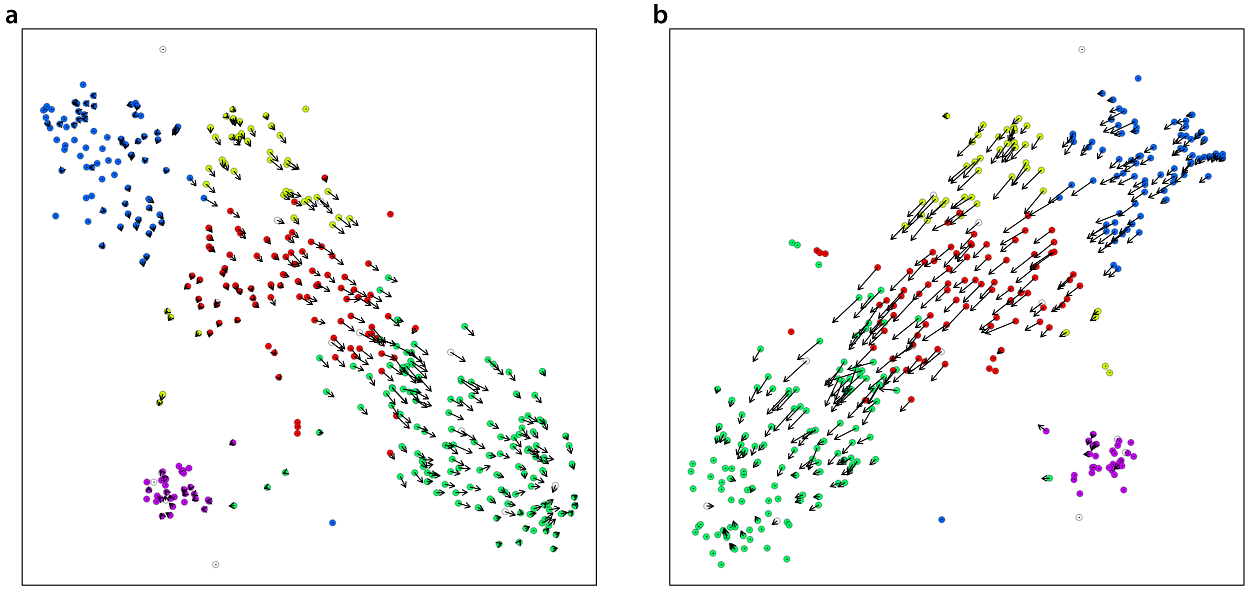

Figure 9. Joint t-SNE visualization of observed and extrapolated chromaffin E12.5 cells. a. The chromaffin E12.5 velocities estimated using gene-relative fit, with *k=5* cell pooling are shown by joint embedding of observed (circles) and extrapolated cells (end of arrows) using t-SNE. **b.** Analogous joint t-SNE embedding for the chromaffin E12.5 velocities estimated using structure-based model.

Section 6. Extrapolation distance and interpretation of velocity magnitude

The RNA velocity estimates the first time derivative of the expression state, and a linear extrapolation is used to estimate the state of the cell short period of time into the future. In general, the time at which such extrapolation will be effective depends on the curvature of the manifold that the underlying biological process is following. For instance, in the circadian cycle examples shown in Figure 1h of the main manuscript, it is noticeable that the extrapolated states lie on tangent lines and lag behind a circular shape of the circadian trajectory. While the effective extrapolation time will vary depending on the biological process, we used a simple chromaffin linear differentiation trajectory to estimate the extrapolation time for that particular case (Figure 10). To do so, we identified the pseudotime difference between each cell and the cell most closely resembling the extrapolated cell. Scaling this distribution by the experimentally-determined 14 hour total chromaffin differentiation time, we obtained the distribution shown in the Figure 2g of the main manuscript.

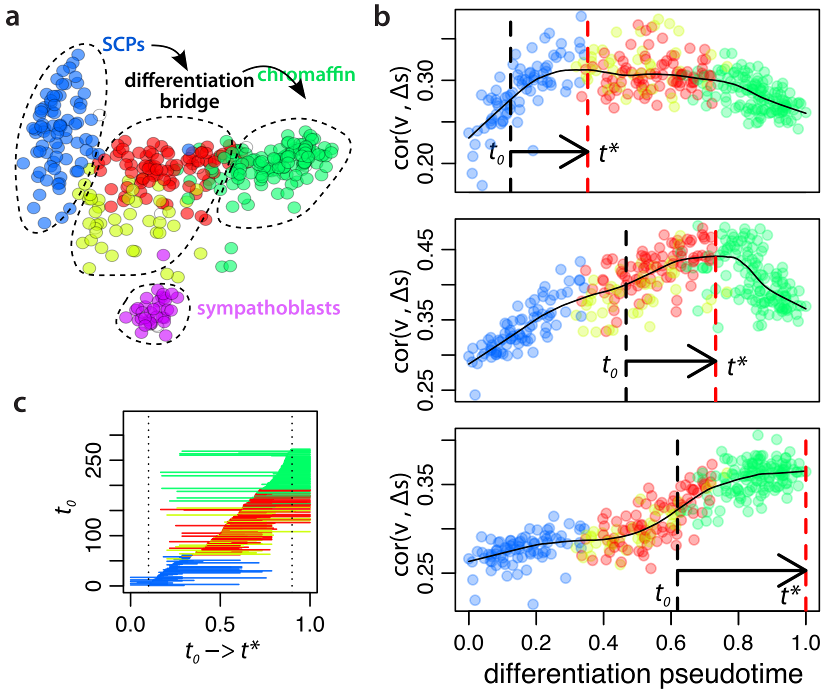

Figure 10. Effective extrapolation distance for chromaffin differentiation. a. PCA projection of E12.5 dataset showing, as a reference, major subpopulations in the chromaffin differentiation (same as in Figure 2a of the main manuscript). **b.** Optimal extrapolation distance along the chromaffin differentiation trajectory. The plots show correlation between the velocity vector and cell expression difference vector (y axis) for the cells ordered by chromaffin differentiation pseudotime (x axis). Correlation profiles for three example cells are shown, with pseudotime of each cell (t_0) and pseudotime of the maximal correlation (t^*) marked by the black and red dashed lines, respectively. **c.** The optimal extrapolation distances (from t_0 to t^*, x axis) are shown for all of the cells along the chromaffin differentiation pseudotime (y axis). The distribution of these distances is shown in the Figure 2g of the main manuscript. The cells at the extreme of the pseudotime (beyond the 10% thresholds marked by vertical dashed lines on the current plot) were excluded, as estimation of pseudotime within such extremes is not expected to be robust. For the Figure 2g of the main manuscript, the pseudotime time differences were translated into real hours, based on the 14 hour total chromaffin differentiation time (see Figure 12). Even though we have trimmed 10% populations on each side of the measured chromaffin time course, we have not adjusted the 14 hour total differentiation time accordingly, as the trimmed populations likely represent static subpopulations that would not be captured in the 14 hour window. Applying such adjustment, would reduce the estimated mean effective timescale to 1.7 hours.

To confirm that our approach was able to capture differences in the magnitude of transcription velocity, we examined another mouse chromaffin differentiation dataset, taken at a later developmental time point (E13.5). The resulting velocities recapitulate chromaffin differentiation in a way similar to the earlier time point (Figure 11), however showing lower apparent velocity magnitude for the chromaffin bridge (red and yellow clusters).

Direct comparison of unspliced / spliced abundances between different subpopulations confirmed statistically significant decrease in predicted velocity in E13.5 time point compared to E12.5 (Figure 12c). To confirm this, we quantified the relative abundance of *Sox10*+ Schwann cell precursors, *Htr3a*-GFP+ bridge cells, and *Th*+ chromaffin cells in tissue sections. Indeed, we found that the developmental dynamics of chromaffin cell production slowed down at E13.5 as compared to E12.5 based on the ratio of progenitors and resulting TH+ cells, consistent with lower predicted velocity (Figure 12).

Mouse lines

For all experiments, the day the plug was detected was considered as E0.5. All animal work was permitted by the Ethical Committee on Animal Experiments (Stockholm North committee) and conducted according to The Swedish Animal Agency's Provisions and Guidelines for Animal Experimentation recommendations. *Htr3a*^{EGFP} animals were received from MMRRC and provided by J. Hjerling-Leffler laboratory (Karolinska Institutet, Sweden) (https://www.mmrrc.org/catalog/sds.php?mmrrc_id=273).

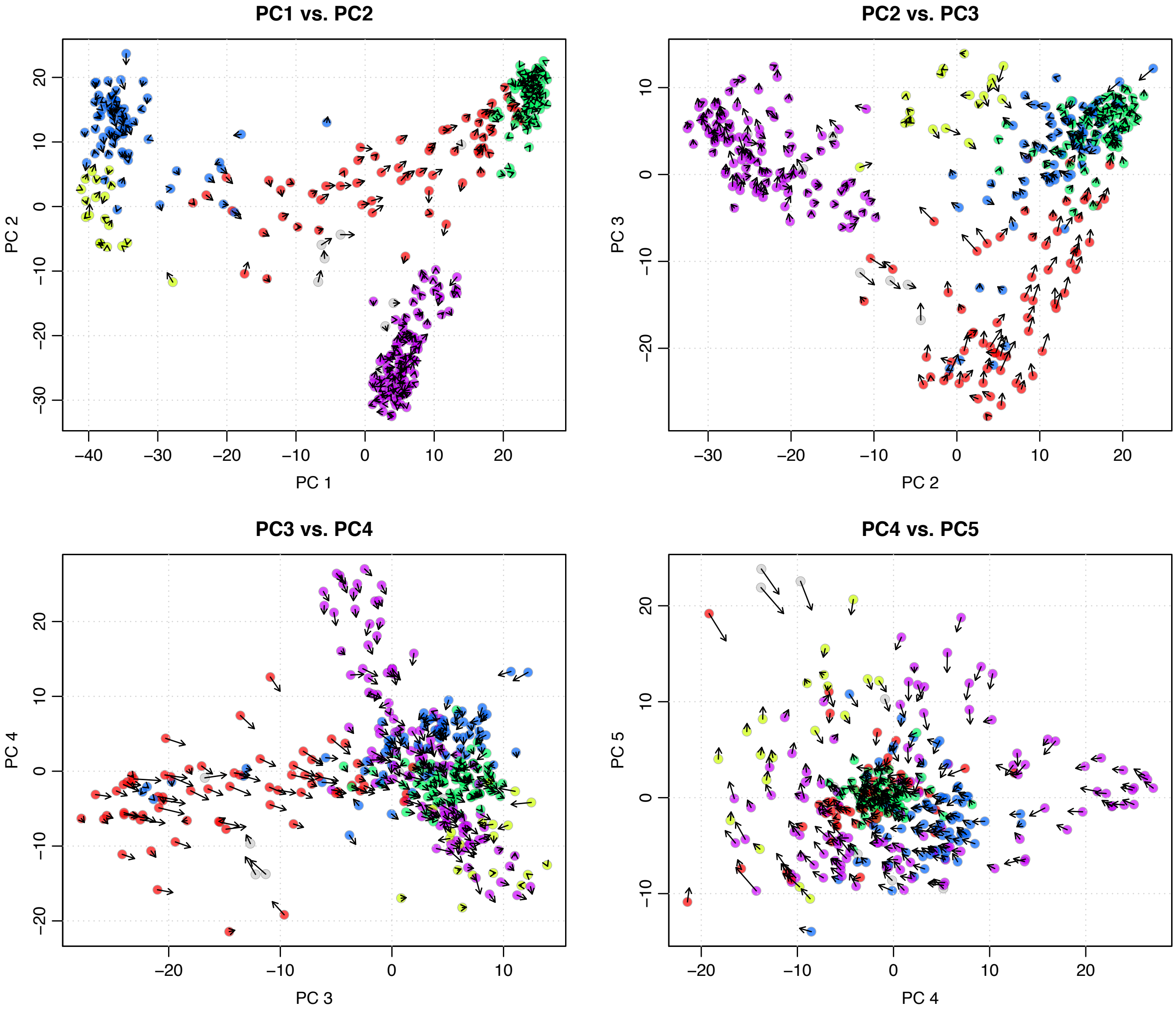

Figure 11. PCA visualization of chromaffin E13.5 velocities using estimated using gene-relative model. PCA projections are used to show E13.5 chromaffin dataset velocities (n=362 cells), as estimated by the gene-relative model with k=5 cell kNN pooling. Projections onto the first five PCs are shown. The cell clusters are colored using the same color scheme as for E12.5 dataset (see Figure 2a of the main manuscript).

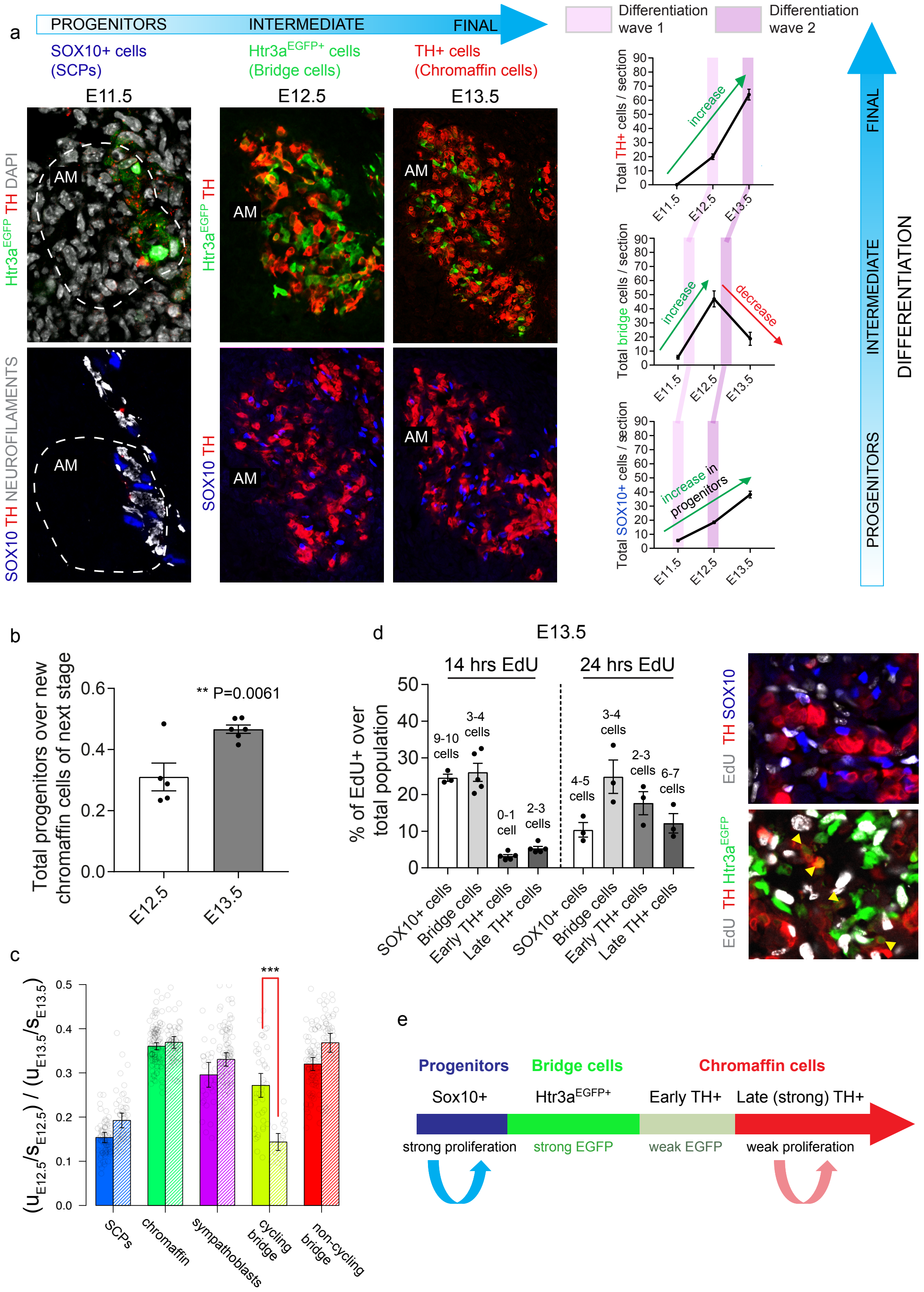

Figure 12. Analysis of developmental dynamics in early adrenal medulla. a. Immunofluorescence of the developing adrenal medulla during embryonic stages E11.5 to E13.5. During this period of development, there is a continuous differentiation of TH+ chromaffin cells from intermediate Htr3a-EGFP+ bridge cells that is in turn formed from the only massively proliferating Sox10+ SCP progenitors. It is possible to observe the dynamics of

differentiation at different developmental days by tracking the numbers of Sox10+ progenitors, bridge and differentiated chromaffin cells. The panels on the left show immunohistochemistry analysis of SOX10, Htr3a-EGFP and TH at different developmental stages of adrenal medulla. In all three stages, a minimum of N=3 embryos were analysed, from two independent litters (N=3 for E11.5, N= 5 for E12.5 and N=6 for E13.5). The graphs of the right show how the ratios of progenitors, intermediate cells and differentiated cells are changing over time, in each case as mean±SEM. The graph shows the transition from the first "wave" of SCP progenitors to the first population of differentiated chromaffin cells (first detected at E12.5), as well as that of the second "wave" (generated from SCPs from E12.5 and detected at E13.5) as depicted by the two tracers in the plot shown in two hues of pink. Note that at E13.5 the pool of bridge cells decreases as compared to the pool of bridge cells at to E12.5, whereas Sox10+ progenitors keep increasing their numbers from E12.5 to E13.5. This supports the decreased transition of SCP progenitors into bridge state between E12.5 and E13.5 as compared to the same transition between E11.5 to E12.5 especially taking into account that the speed of accumulation of mature TH+ cells changes only slightly. **b.** Proportion of SOX10+ progenitors over generated TH+ chromaffin cells per next developmental stage (TH+ cells accumulated at the previous developmental stages were subtracted). Note that bigger numbers of Sox10+ progenitors generate proportionally less TH+ chromaffin cells at E13.5 as compared to E12.5. Data were collected from N=3 for E11.5 embryos, N=5 for E12.5 and N=6 for E13.5, and represented as mean±SEM (E12.5: 0.3103±0.0455, E13.5: 0.4664±0.0138). Statistical significance was calculated using a two-tailed unpaired t-test with 95% confidence level. **c.** The barplots compare the ratio of total unspliced and spliced mRNA molecules between E12.5 (solid bars) and E13.5 (shaded bars) time points. The bars and whiskers show mean±SEM. Statistically significant ($p<10^{-5}$, two-sided *t* test) decrease in the unspliced/spliced molecule count ratio is observed for the cycling subpopulation (yellow) of the chromaffin bridge at E13.5, indicating lower cell expression velocity. **d.** Measurements of EdU incorporation and retaining in various populations of adrenal medulla 14 and 24 hours after the single pulse. The analysis stage is E13.5. Data were collected from N=5 E13.5 embryos which received EdU 14 hrs prior to collection, and N=5 for E13.5 which received EdU 14 hrs prior to collection, and represented as mean±SEM (E13.5 - 14 hrs EdU: SOX10+ cells=24.580±0.989, bridge cells=26.060±2.449, early TH+ cells=3.183±0.460, late TH+ cells=5.305±0.557, E13.5 - 24 hrs EdU: SOX10+ cells=10.370±1.980, bridge cells=24.860±4.541, early TH+ cells=17.640±3.145, late TH+ cells=12.160±2.630). Note that the very first TH+ cells that retain both weak GFP and EdU (immediate progeny of Htr3a-EGFP+ bridge cells) are identified in the tissue 14 hours after EdU injection, which suggests the minimal time of the trajectory from Sox10+ proliferative SCPs to differentiated TH+ chromaffin cells. Yellow arrowheads in immunohistochemistry panel point at EGFP-retaining TH+ cells. **e.** Schematic explanation of differentiation progression in chromaffin cell lineage. Note that Sox10+ SCPs proliferate strongly. At the same time, very few independently dividing cells were detected in more mature GFP-/TH+ population of chromaffin cells 4 hours after EdU pulse (data not shown).

EdU incorporation and analysis

14 hrs or 24 hrs prior to embryo collection, pregnant females received an intraperitoneal injection of EdU (50 µg/g of body weight). EdU was visualized using the Click-iT EdU Alexa Fluor 647 Imaging Kit (Life Technologies) according to manufacturer's instructions.

Immunohistochemistry

Immunohistochemistry was performed as previously described[22]. Briefly, embryos were collected and fixed in 4% paraformaldehyde in PBS (pH 7.4) at 4°C for 5 hours. Samples were washed in PBS at 4°C for one hour and cryoprotected by incubating at 4°C overnight in 30% sucrose in PBS. Tissue samples were subsequently embedded in OCT and frozen at -20°C. Tissue samples were sectioned at 14 µm and frozen at -20°C after drying at RT for at least one hour. Antigen retrieval was performed by immersing the sections in 1x Target Retrieval Solution (Dako, S1699) in water for 20 min, pre-heated at 80°C. Sections were washed three times in PBS containing 0.1% Tween-20 (PBSt), incubated at 4°C overnight with primary antibodies diluted in PBSt and coverslipped with parafilm. Finally, sections were washed in PBSt and incubated with secondary antibodies diluted in PBSt at RT for one hour, washed again three times in PBSt and mounted using Fluorescent mounting medium (Dako, #S3023).

Primary antibodies

Goat anti-GFP (1:500, Abcam, #ab6662), mouse anti-Neurofilaments (1:100, clone 2H3, DSHB), goat anti-SOX10 (1:500, Santa-Cruz, #sc-17342), mouse anti-SOX10 (1:500, Santa-Cruz, #sc-374170), rabbit anti-TH (1:1000, Pel-Freez Biologicals, #P40101-150).

DAPI (Thermo Fisher Scientific, 1:10,000, #D1306) was diluted in PBS and applied on sections for 20 min at 20–25 °C, after immunohistochemistry.

For detection of the primary antibodies, secondary antibodies raised in donkey and conjugated with Alexa-488, -555 and -647 fluorophores were used (1:1000, Molecular Probes, ThermoFisher Scientific).

Microscopy

Images were acquired using LSM 710 and LSM 780 Zeiss confocal microscopes equipped with 20x, 40x and 63x objectives. Images were acquired in the .lsm format and processed with ImageJ or IMARIS (8.0).

Section 7. Visualizing cell diffusion trajectories over longer time scales

To extrapolate the movement of the cell over longer periods of time, one can assume that the underlying biological process is ergodic – that is all of its properties and intermediate transition states can be observed given that sufficiently large number of cells has been measured. One such extrapolation approach is to approximate the shape of the expression manifold using k nearest neighbor graphs, and then track velocity-biased diffusion of cells within this graph. In a discrete setting this can be done as a simulation of a Markov process with transition probabilities biased by the estimated velocity vectors for each cell (Figure 13).

Modeling of cell trajectories was performed based on a Euclidean transition probability matrix, as it provides better control over distant transitions by allowing to explicitly describe the drop off in the transition probability with the increasing expression distance.

$$\boldsymbol{P}_{ij} = K_{\sigma}\left(\hat{\boldsymbol{s}}_{t,i}, \hat{\boldsymbol{s}}_{0,j}\right)$$

where $\boldsymbol{s}_{0,j}$ is the (size-normalized) observed spliced expression state of a cell j , and $\boldsymbol{s}_{t,i}$ is the extrapolated state of the cell i at a time t. $\hat{\boldsymbol{s}}$ designates a projection of vector $\boldsymbol{s}$ onto the first 30 principal components. $\sigma = 2.5$ was used. The background transition probability capturing the observed cell similarities was calculated as:

$$\boldsymbol{B}_{ij} = K_\sigma\left(\hat{\boldsymbol{s}}_{0,i}, \hat{\boldsymbol{s}}_{0,j}\right)$$

Transition probabilities between cells were restricted to $k = 40$ nearest neighbors by setting other values to 0. To correct for local cell density, the rows of each matrix was multiplied by the $diag(\boldsymbol{B}^b)$. The matrices were row-normalized to unity. The probability of a cell i at a discrete time t was estimated as $\boldsymbol{P}^t$. Most likely position of each cell after $t_f = 500$ were estimated as the maximum likelihood positions. A trajectory p_i for cell i was determined as a path maximizing the total log likelihood:

$$argmax_{p_i} \sum_{t=0}^{t_f} log\left(\boldsymbol{P}^t{}_{i,p_i(t)}\right)$$

where $p_i(t)$ is the predicted trajectory position of the cell i at a time t . To determine prevalent trajectories within a population, individual trajectories p_i were clustered using manhattan distance measuring the difference in the set of cells covered by each path using k-means clustering. 10 clusters were used. The cluster medoids were visualized using spline smoothing.

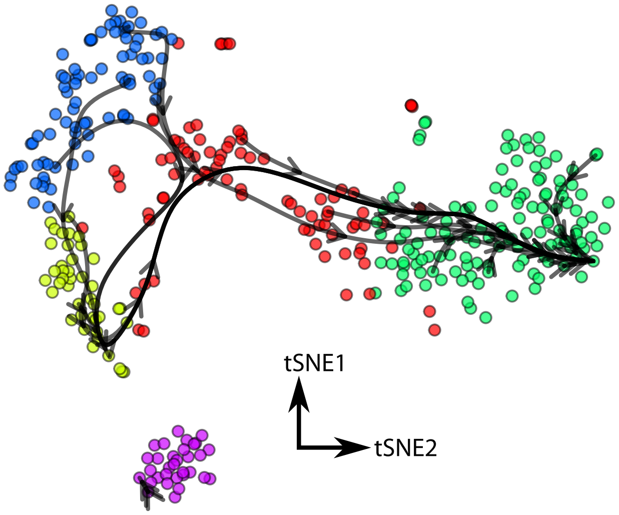

Figure 13. Predicted cellular trajectories for chromaffin E12.5 dataset. Cell diffusion was modeled by a Markov process with transition probabilities determined based on the velocity estimates. Trajectories were simulated for each cell and clustered into 10 clusters. The centroid trajectories of each cluster are shown, using spline smoothing.

Section 8. Uncertainty of velocity projections with respect to the exact gene and cell set

To evaluate to what extent the velocity estimates are driven by specific genes or specific cells, we have performed velocity estimation under bootstrap sampling of cells (or genes). Performing multiple rounds of such bootstrapping we assessed variability of the resulting projections (under different neighborhood size parameters, see Section 11 for further discussion of that parameter). Overall, we find that velocity directions are stable, showing very low sensitivity to the exact set of cells and genes (Figure 14).

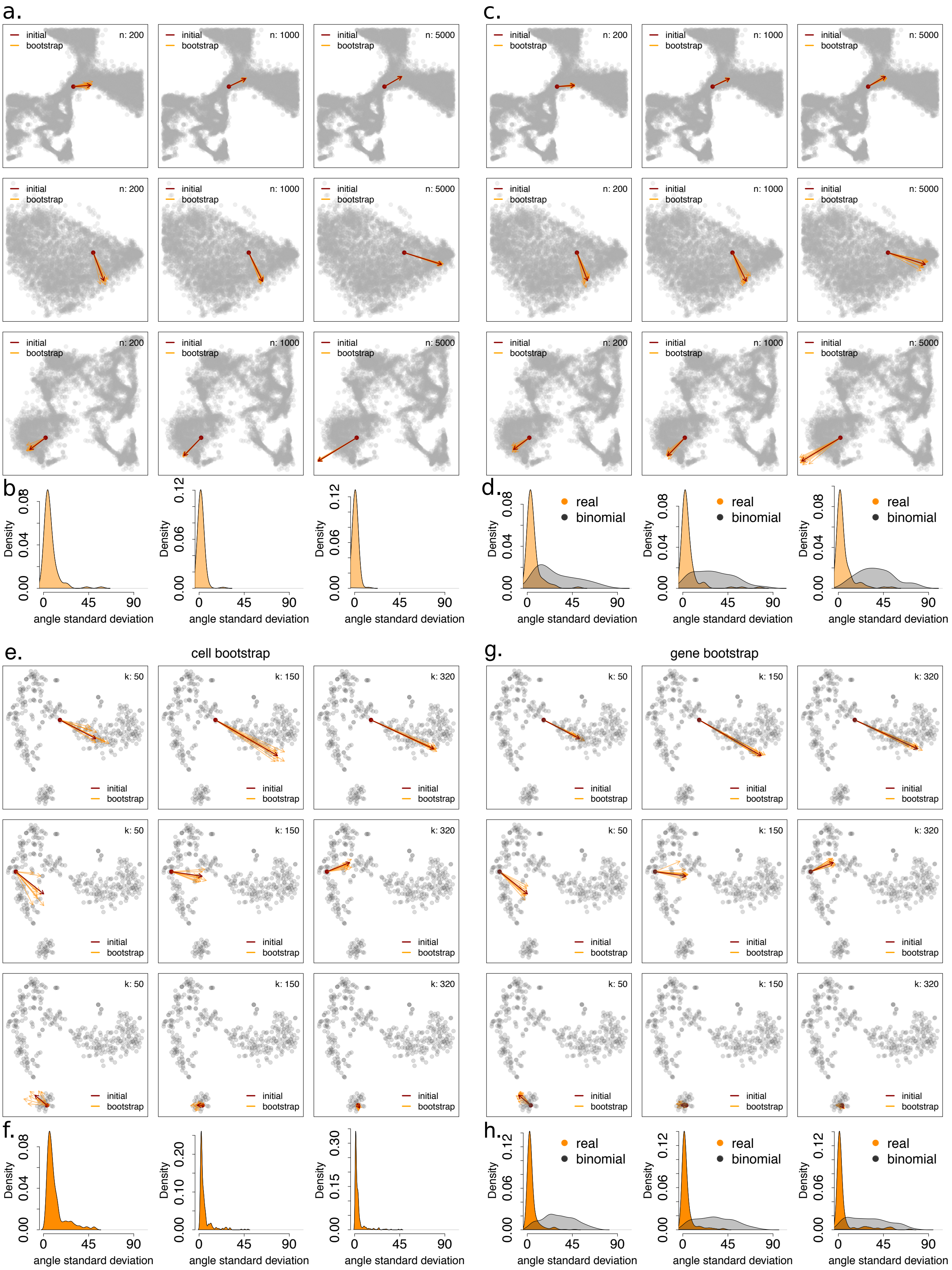
a.
initial
bootstrap
n: 200
n: 1000
n: 5000
c.
b.
Density
0.00
0.04
0.08
0.06
0.12
0
45
90
angle standard deviation
d.
real
binomial
e.
cell bootstrap
k: 50
k: 150
k: 320
g.
gene bootstrap
f.
0.10
0.20
0.15
0.30
h.

Figure 14. Sensitivity of velocity estimates to the exact gene and cell set. a. Examples of the projected velocity arrow directions for multiple cell bootstrapping rounds are shown for the three example cells from the hippocampal dataset (upper, middle and bottom set of panels). The effect of the neighborhood size visualization parameter (n) is also shown (left: n=200, middle: n=1000, right: n=5000). **b.** Distribution of standard deviations of cell arrow angles for 100 random cells (x-axis is shows angle in radians). Standard deviation of the projected velocity arrow angle is estimated based on cell bootstrap ensemble of arrows. The distributions are shown for three values of the neighborhood size parameter (left: n=200, middle: n=1000, right: n=5000). **c.** Examples of velocity arrow directions obtained after gene bootstrapping. **d.** Distribution of standard deviations of velocity arrow angles for 100 cells of gene bootstrap-based arrow ensembles (orange). Standard deviations of random (binomial) velocity estimates are shown for comparison (grey). **e-h.** Analogous panels showing example velocity projection uncertainty and dataset-wide velocity arrow angle variance under random cell (e,g) and gene (f,h) bootstrapping results are shown for the chromaffin E12.5 dataset.

Section 9. Estimates of velocity on random data

To assess the biases that may be introduced by the velocity estimation and visualization procedures, we examined the velocities generated under different *null* background distributions, where we do not expect to see pronounced velocity. Randomization of the data was performed using three different schemes. A naïve scheme where the residuals from the gamma fit where randomized among samples, in particular the values of $\Delta s = vt$ were permuted for each gene independently and sign was randomly flipped, velocity projection was then calculated as usual. A binomial-based randomization first modeled the expected expression noise (expression variance *v*) as a function of mean expression magnitude of the gene using $\log(v) \sim \log(s)$. The expected unspliced intensity $\mu_{c,g}$ for each gene g in each cell *c* was sampled as:

$$\mu_{c,g} \sim s_{c,g} * N(0, v_g) * \xi * \gamma_g$$

$$\gamma_g \sim \exp\,[N(0{,}0.5)]$$

where $\xi = 0.2$ (a ratio of random noise variance to the dataset-wide gene variance), and γ_g is the randomly sampled equilibrium slope for a given gene g. After that, the observed unspliced counts for each gene ($u_{c,g}$) were sampled using binomial distribution with the number of trials equal to empirically observed size of a cell, and probability being equal to $\mu_{c,g}$.

The binomial-based randomization yielded low velocity field in two small patches of the embedding – a residual effect likely driven by size normalization and t-SNE projection procedures. We note that such residual signal is unstable under gene bootstrap sampling (Figure 14d,h). To probe to which extent these projected velocity directions corresponded to the high dimensional velocity estimates, we scaled the arrow by a trimmed cosine projection of the high-dimensional velocity vector onto the expected expression shift, as calculated from the transition probabilities.

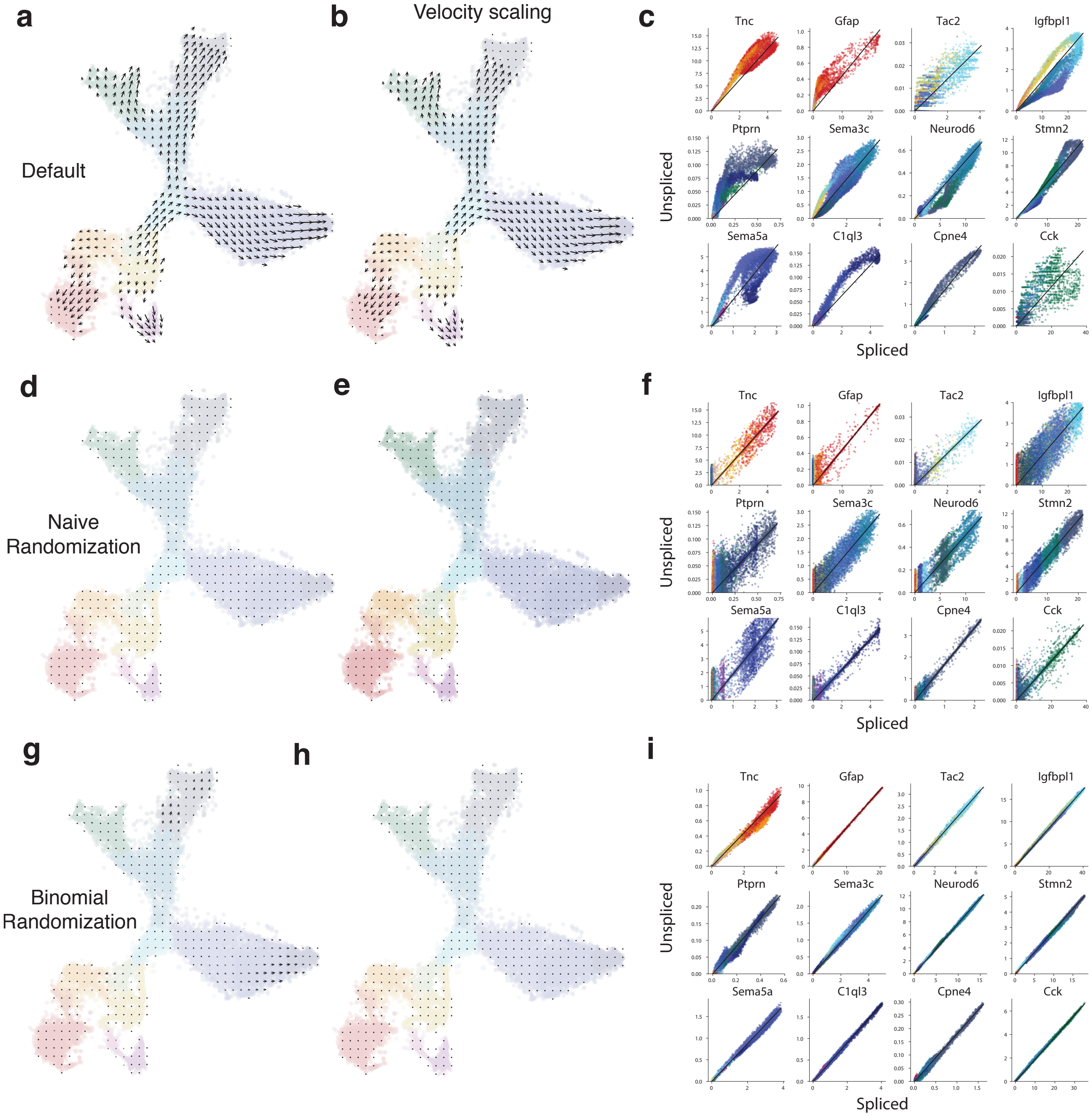

Figure 15. **Velocity estimates for randomized data for the mouse hippocampus dataset. a-c.** Velocity field and phase portraits in the dentate gyrus dataset, along with phase portraits of sample genes (c). **d-f** The results of a naïve randomization procedure where residuals of the gamma fit were reassigned to the cells randomly. **g-i** The generation of simulated unspliced molecule counts using a bionomial model that assumes no velocity information (only noise) is contained in the unspliced counts. The velocity field on the right columns (b, e and h) shows velocities after a rescaling that takes in consideration how well the represented velocity arrow summarizes the high dimensional velocity (see details above).

Specifically, we calculated a scaling factor h as follows:

$$h = \begin{cases} 0 \;\; if \;\; \frac{\boldsymbol{v}\hat{\boldsymbol{v}}}{\|\hat{\boldsymbol{v}}\|} < 0 \\ 1\, if \;\; \frac{\boldsymbol{v}\hat{\boldsymbol{v}}}{\|\hat{\boldsymbol{v}}\|} > 1 \\ \frac{\boldsymbol{v}\hat{\boldsymbol{v}}}{\|\hat{\boldsymbol{v}}\|} \;\; otherwise \end{cases}$$

$$where$$

$$\hat{\boldsymbol{v}} = P\boldsymbol{s} - U\boldsymbol{s} \qquad P_{ij} = \begin{cases} p(i \rightarrow j)\; if\; j\; \in kNN(i) \\ \;\;\;\; 0 \qquad\quad otherwise \end{cases} \qquad U_{ij} = \begin{cases} \frac{1}{k}\; if\; j\; \in kNN(i) \\ 0 \qquad otherwise \end{cases}$$

After this "velocity scaling" correction the randomized samples did not show any noticeable velocities, indicating that they were driven by correlation of low-magnitude components. At the same time, applying the same scaling strategy to the real data did not have a noticeable impact (Figure 15).

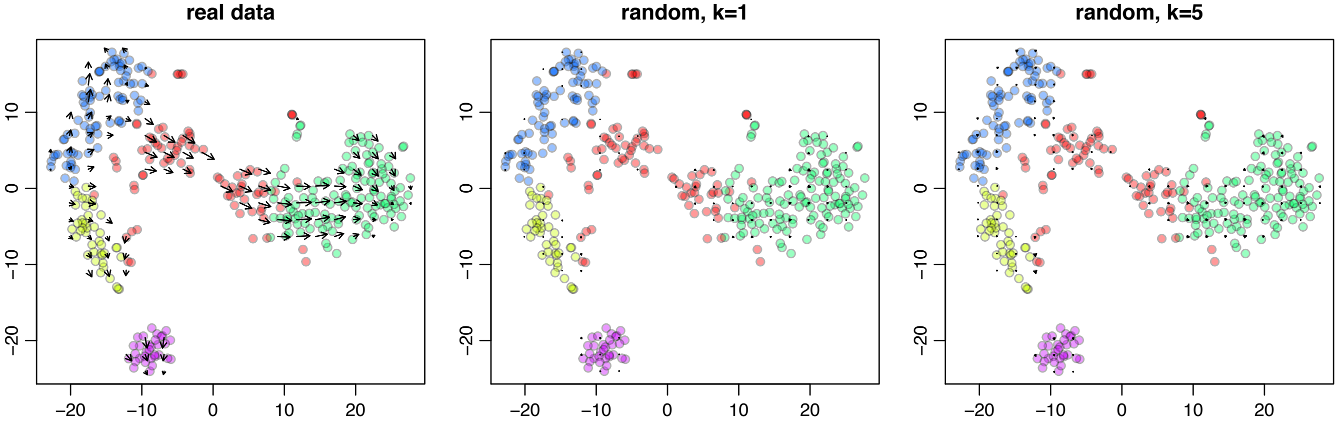

Figure 16. Velocity estimates for randomized data for the chromaffin E12.5 dataset. The left panel shows grid view of velocities predicted for the real chromaffin E12.5 dataset. The subsequent two panels show grid visualization of velocities for binomial random data, without cell kNN smoothing (k=1), and with kNN smoothing (k=5).

Section 10. Sensitivity to estimation parameters

To examine robustness of the velocity estimates to the variations in gene filtering and other parameters, we have evaluated variation in the direction and magnitude of the projected velocity vector field under different parameter variations on the hippocampus dataset (Figure 17).

Sensitivity analysis was performed varying one parameter at a time for the set of parameters shown in the figure (Figure 16). Change in the velocity vector filed (as computed on a grid) were summarized in two scores reported taking into account the change in direction and magnitude respectively. The scores were computed as:

$$score1 = var(\Delta\theta)$$

$$score2 = corr(\boldsymbol{m_{ref}}, \boldsymbol{m_{test}})$$

where $\mathbf{m}$ is the vector of magnitudes associated at every point $\boldsymbol{w}$ of the grid that is used to visualize the field: $m_w = \|v(w)\| = \sqrt{v(w)_x^2 + v(w)_y^2}$

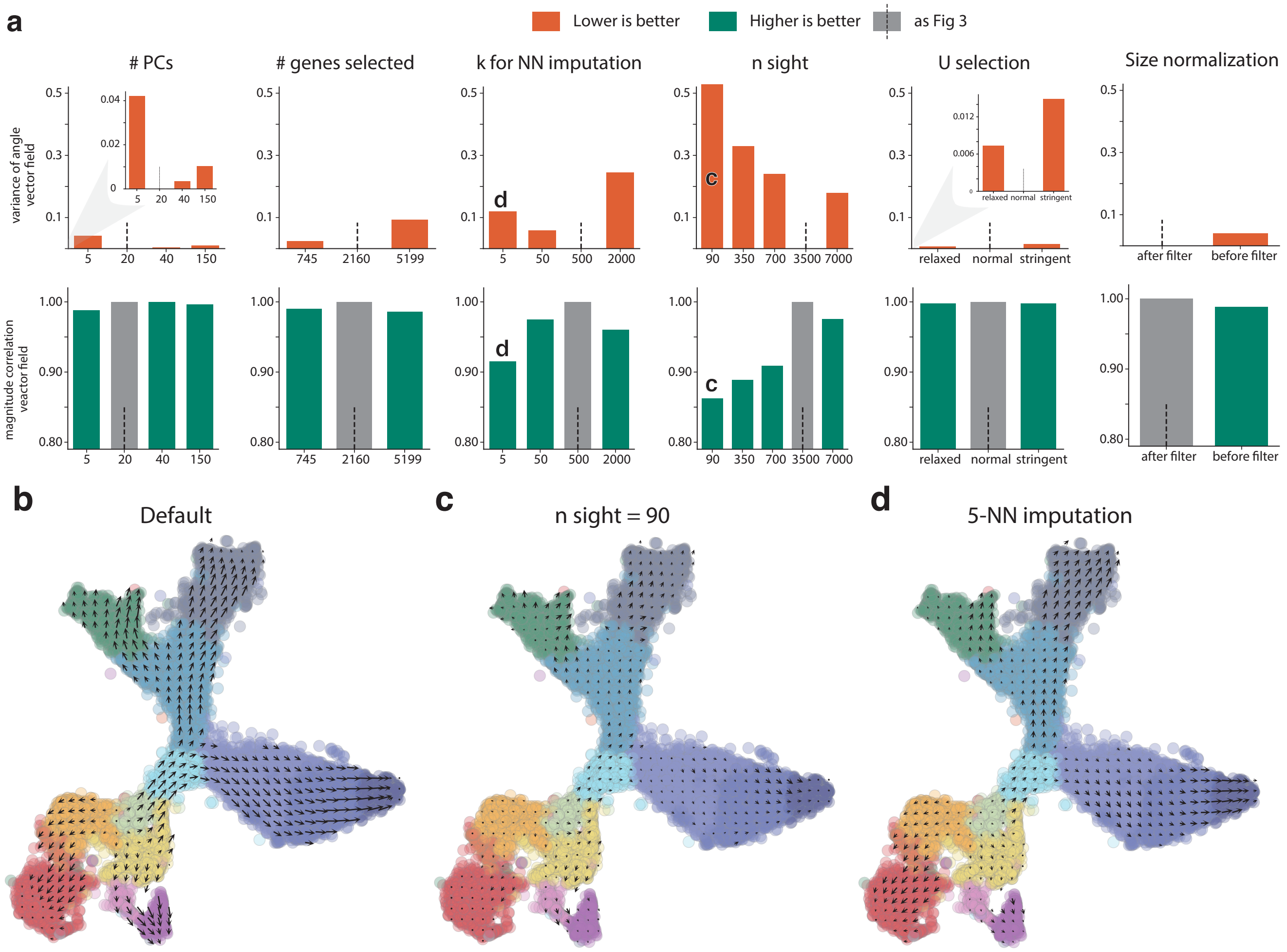

Figure 17. Sensitivity of velocity field estimation to different estimation parameters. **a.** The deviation of the velocity field from the default reference (Figure 3 of the main manuscript) is shown for variation of different parameters. Two summary scores are provided: the average variance of the projected velocity arrow direction, and the correlation of arrow magnitudes. The bars labeled with letters have corresponding velocity field shown below. Notice that the input parameter that influence the estimation the most is the "n sight" parameter, which defines the

size of the neighborhood size taken in consideration when projecting the velocity from high dimensional space onto pre-defined low-dimensional cell embeddings. The angles are measured in radians. **b-d.** Illustration of the velocity projections with default (b) and altered (c,d) parameter settings.

Section 11. Uncertainty and limitations of neighborhood-based velocity projections

The "n sight" parameter, defining the size of the neighborhood used for projecting the velocity onto pre-defined embeddings appears to be the most sensitive parameter (Figure 17, Figure 18).

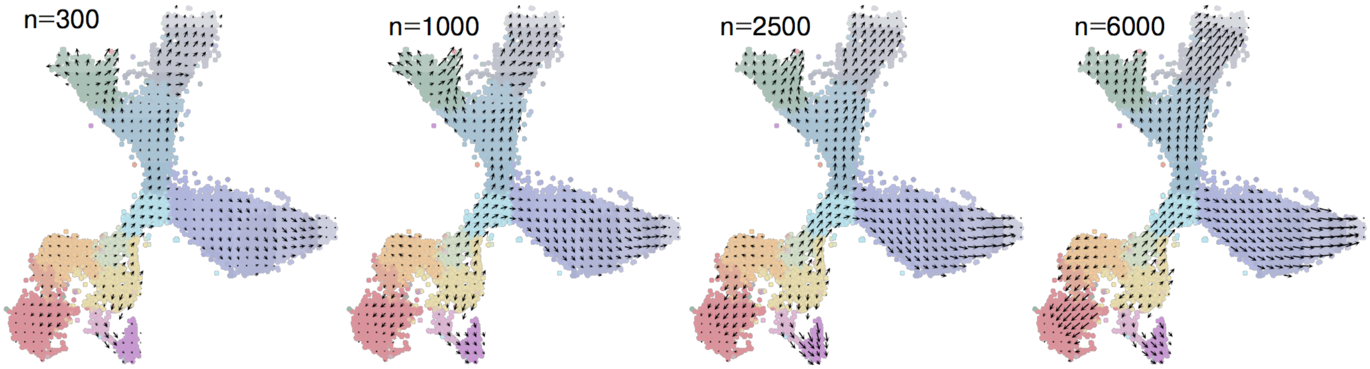

Figure 18. Projections of hippocampus velocity estimates using different neighborhood sizes.

Projections of velocity onto embeddings without gene-defined axes is generally challenging. The approach implemented in *velocyto* (see Methods and Supplementary Note 1) relies on looking at a neighborhood around each cell, examining expression state (spliced) differences with different cells in the neighborhood, and drawing a velocity arrow in the direction of expected cell shift after accounting for the cell density (see Supplementary Note 1). Specifically, the procedure calculates the difference between expected cell transition direction based on velocities and the direction based on the even transition probability (which will point towards the center of mass, see Figure 19a).

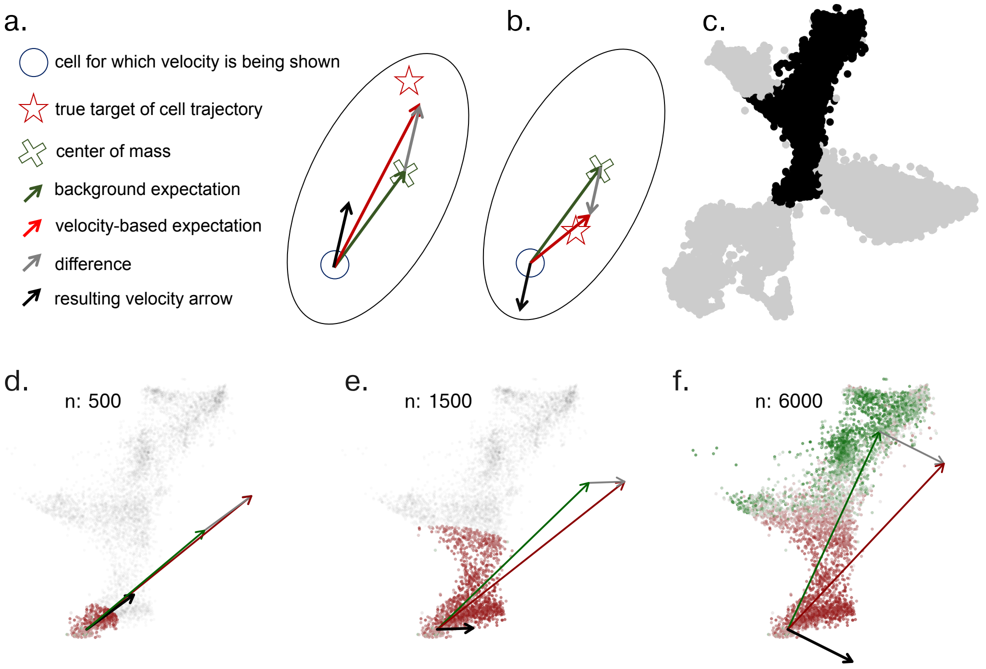

Figure 19. Edge effect in neighborhood-based projections. a. A straightforward example of a cell velocity projection is shown, with the neighborhood of the cell

pictured as an oval, and positions of the cell velocity target and center of mass indicated by star and cross symbols, respectively. The difference between velocity-biased (red arrow) and flat (green arrow) transition probability direction expectations defines the direction and the magnitude of the velocity arrow (grey arrow shows the original difference, black arrow shows the same vector shifted into the position of the cell). **b.** An edge effect, where the center of mass is located further away from the cell target (i.e. a very large cell neighborhood was used). In such case, the resulting velocity arrow will point in the direction opposite of the true cell trajectory target. **c.** Region of the hyppocampus manifold (see Figure 3 of the main manuscript) being analyzed is highlighted in black. **d-f.** An example of this effect on the real data, shown for increasing neighborhood sizes. All arrow sizes were scaled by a constant factor for the purposes of this visualization.

The neighborhood-based procedure has several biases and unintuitive effects. First, if the neighborhood size is larger than the distance to the ultimate trajectory target, it is possible to have edge effects that will rotate or even invert the direction of the projection away from its intended target (Figure 19). Second, the measured subpopulations may contain several distinct subpopulations that share some expression similarity with the direction of the predicted velocity extrapolation. As the velocity projection is based on the expected transition direction, such "multiple attractors" can result in a rotation of the velocity directions (Figure 20).

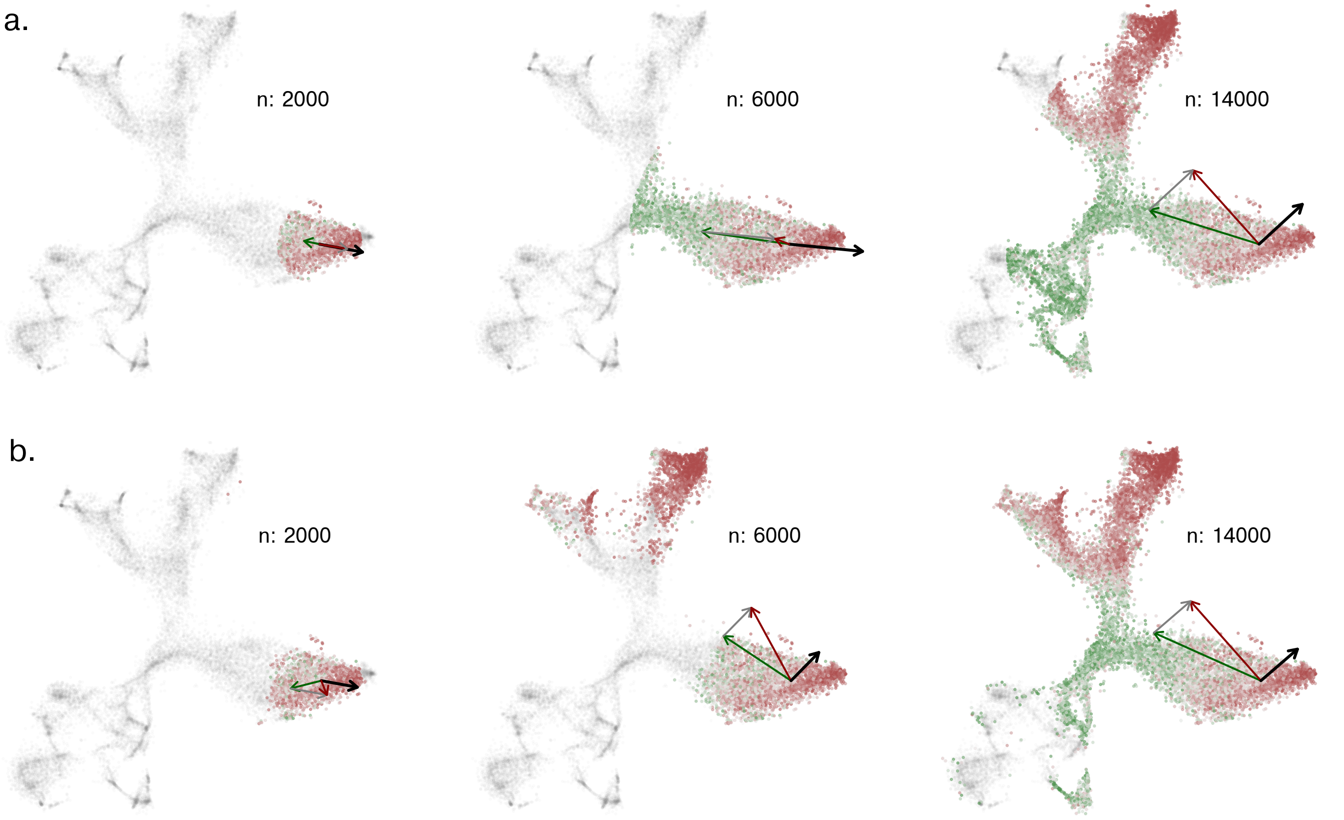

Figure 20. Effect of multiple attractors in the hippocampus dataset. a. The effect of multiple attractors is shown for a single cell in the hippocampus dataset. At low neighborhood sizes, the projection points correctly towards the terminally-differentiated end of the corresponding trajectory arm. However, as the neighborhood size increases, other maturing neuronal populations (CA2/4 and CA3) come into view. As the tips of these branches also represent maturing neurons, they exhibit strong expression similarity to velocity direction of the chosen cell, and end up

rotating the expected transition direction towards them. **b.** The neighborhood can be defined using high-dimensional cell-cell distance, in which case, the terminally differentiating tips of other neuronal branches will come into view even faster as the neighborhood expands, rotating the velocity projection sooner.

Finally, it is important to point out that as velocity vectors are estimated in high-dimensional space, their projections onto low-dimensional space can create situations where different populations appear to be in intersecting or diverging course, whereas in high-dimensional their trajectories never cross or come near each other.