Variational inference of single cell time series

Bingxian Xu[1,2] and Rosemary Braun[1,2,3,4,5,6,*]

[1]Department of Molecular Biosciences, Northwestern University, Evanston, IL 60208, USA

[2]NSF-Simons National Institute for Theory and Mathematics in Biology, Chicago, IL 60611, USA

[3]Department of Engineering Sciences and Applied Mathematics, Northwestern University, Evanston, IL 60208, USA

[4]Department of Physics and Astronomy, Northwestern University, Evanston, IL 60208, USA

[5]Northwestern Institute on Complex Systems, Northwestern University, Evanston, IL 60208, USA

[6]Santa Fe Institute, Santa Fe, NM 87501, USA

[*]To whom correspondence should be addressed. Email: rbraun@northwestern.edu

Abstract

Time course single–cell RNA sequencing (scRNA-seq) enables researchers to probe genome–wide expression dynamics at the the single cell scale. However, when gene expression is affected jointly by time and cellular identity, analyzing such data — including conducting cell type annotation and modeling cell type–dependent dynamics — becomes challenging. To address this problem, we propose SNOW (SiNgle cell flOW map), a deep learning algorithm to deconvolve single cell time series data into time–dependent and time–independent contributions. We describe our probabilistic framework and demonstrate SNOW using both synthetic and real scRNA-seq data. Our results show that SNOW is able to construct biologically meaningful latent spaces, remove batch effects, and generate realistic time–series at the single–cell level.

Keywords: Single-cell RNA-seq, Variational Inference, Gene expression time series, Batch correction

1 Background

Gene expression is shaped by intrinsic cellular identities and extrinsic environmental conditions. Today, single-cell RNA sequencing (scRNA-seq) technologies enable us to probe how gene expression

changes across cell types under various experimental conditions [1–5], with applications ranging from organ development [6, 7] to cancer progression [8, 9] and more recently to circadian rhythms [10, 11]. To understand the dynamics of these processes, studies have started to directly observe how gene expression profiles change over time via time–coures scRNA-seq profiling [7, 12–14] and a number of methods have been developed to characterize and model scRNA-seq time-series data. For example, Waddington-OT [6] applies unbalanced optimal transport to compute the likelihood of cell state transitions. To gain mechanistic insights, PRESCIENT (Potential eneRgy undErlying Single Cell gradIENTs) [15] constructs a global potential function, $\psi(\mathbf{x})$, and uses $\Delta\psi(\mathbf{x})$ to estimate how gene expression, $\mathbf{x}$, changes over time via the Euler scheme $\mathbf{x}(t+\delta t) = \mathbf{x}(t) - \Delta\psi(\mathbf{x})\,\delta t$. However, this potential function is constructed on the PCA space, which may not represent the relevant geometry and cannot be mapped back to the original gene expression space after the dimensionality is reduced. If the data lie on a nonlinear/curved manifold, such as in the "Swiss roll" example [16] or when gene expression dynamics have a cyclic component, PCs will fail to articulate this coordinate of variation. To overcome this limitation, scNODE [17] uses a variational autoencoder [18] to construct a lower dimensional space with which to find governing equations that recapitulate the observed dynamics.

All the aforementioned methods are some variant of parameterizing a flow that satisfies the optimal transport constraint. For example, PRESCIENT [15] and scNODE [17] minimize the Wasserstein distance between data sampled from one time point and that sampled from an earlier time point that subsequently evolved according to the flow. However, this treatment only constrains the flow on measured time points. To further constrain the flow, methods such as TIGON [19] and TrajectoryNet [20] solve the dynamic optimal transport problem [21] by regularizing the entire path along which the probability densities evolve. This approach is useful in contexts were temporal variation affects all cells, such as during development where cells move smoothly along common paths in a low dimensional space (Figure 1A, top). However, this may not be the best description for systems where cells can act in a highly cell type–specific manner over time. In these cases, the paths they take may not be continuous (Figure 1A, bottom), and may complicate cell type annotations. In this latter situation, it is desirable to remove the effect of time to facilitate cell type annotation, which is usually achieved by integrating and batch–correcting the time points. Since removing temporal variation also removes biologically meaningful dynamics that one may wish to study, further analyses use *non*-integrated data to study the average expression for each cell type

over time. While data integration remains an active field of research [22–25], conclusions drawn from such analyses will depend on the quality of the integration and cell type annotation from the first stage.

To address these challenges, we sought to simultaneously decompose gene expression into time–dependent and time–independent components (Figure 1B). By doing this, we can conduct cell type annotation using the time–independent component, study dynamics without requiring cell type annotation, and project cells forward and backward in time to generate time–series for each individual cell (Figure 1C, top). When cell type labels are provided, one can combine time series generated from individual cells to mitigate the impact of batch effects (Figure 1C, bottom panel). Here, we describe SNOW (SiNgle cell flOW map), an unsupervised probabilistic approach for the annotation, normalization and interpolation of single cell time series data. Our approach parameterizes a zero-inflated negative binomial distribution using latent coordinates computed from the count data. To demonstrate its utility, we show that the latent space constructed by SNOW can capture biologically meaningful structure and map cells collected at one time point to past and future states. By constraining the second derivative of generated time series, SNOW also indirectly removes potential batch effects contaminating the time–series. To our knowledge, SNOW is the only method focusing on the analysis of time series of differentiated cells, in which the effects of time and cell state may be mixed in the data (Figure 1A, bottom).

2 Results

2.1 SNOW algorithm

We aim to achieve a number of things with SNOW. First, we wish to construct a time-independent characterization of the cell state to facilitate cell type annotation. This is achieved by minimizing the Wasserstein distance, a measure of distance between probability distributions, between the prior, $p(z)$, and the latent distribution conditioned on sampling time, $q(z|t)$. Second, we wish to map cells forward and backward in time such that the average of the model–generated gene expression time series across cells matches that of the population average (Figure 1C, top). To increase the smoothness of the interpolated trajectories, we incorporated in the loss function the second derivative of generated time series to penalize high curvature (see Methods for more detail). As a consequence

of this second derivative loss, batch effects in the form of a sudden increase/decrease in expression will be simultaneously removed (Figure 1C, bottom). Third, we wish to infer the sample collection time for an untimed sample, which is an active field of research in chronobiology [26–28]. To do this, we incorporated two additional terms in the loss function: one related to predicting the actual sampling time of each cell, and another related to predicting the sampling time of a cell after being mapped to another time by the model.

To achieve this, SNOW models the expression x_{gc} of gene g in cell c s a sample from a zero-inflated negative binomial (ZINB) distribution $P(x_{gc}|l_c, z_c, t)$ that depends on the observed library size of the cell (l_c), the cell state (z_c), and the time (t) of the observation. The cell state z_c is a low–dimensional vector computed by an encoder network that represents the *time–independent* biological variation contributing to x. To remove the effect of time in constructing the latent representation z_c, we constrain the variational posterior conditioned on time $q(z|t)$ to be close to the prior $z \sim \mathcal{N}(\vec{0}, I)$.

The resulting time–independent latent space z of cell states has a number of appealing uses. It may, if desired, be used to conduct cell type annotation (Figure 1B). By changing t as an input to the decoder while holding the time–independent representation constant, we can generate a gene expression profile of a cell as it might appear at past or future times. In other words, we create an object similar to a flow map, in which the expected expression of the past/future state of any individual cell can be generated.

Details of SNOW's probabilistic framework and loss function are given in the Methods section below.

2.2 SNOW constructs biologically meaningful latent spaces

Time can have a profound impact on single cell data when it contributes to gene expression together with cell state. To see if SNOW can identify time–invariant structures presented in data, we constructed toy datasets that are composed of both a rhythmic component, genes that contain a cell type–specific phase, and a flat component, genes that contain a cell type–specific basal expression (Figure 2A). Details of the data may be found in the Methods. We generated 1000 cells each with an assigned cell-type label and sampling time. The cell type label (which defines the phase of the rhythmic component) and the sampling time of each cell jointly define the expression level of

the rhythmic component. From our simple toy dataset, we observed that the incorporation of the rhythmic component resulted in the generation of tiny clusters on the UMAP [29] plot (Figure 2B), wherein cells segregated by both type and time. As expected, when only the flat component is used, the UMAP plot clustered according to the cell type of each cell (Figure 2B).

As described in the Methods, SNOW constructs a representation of the cell state that is independent of time and can be used for cell type annotation, either by using the latent space directly or by mapping cells to a common time. To test how SNOW performs, we applied it to the aforementioned toy dataset and compared it to scPrisma [30] and scVI [31]. As described in the methods, scPrisma [30] is used to decouple and filter the rhythmic component, while scVI [31] was used to treat the sampling time as batch labels and remove its effect. Of the three methods, SNOW is the only one that consistently captured the cell type information independently of the temporal variation (Figure 2C and Figure S2). Both scPrisma and scVI split the cell types into additional clusters based on time (Figure 2C). We further evaluated the performance of these three algorithms when the cell type dependence was only in the phase, only in the amplitude, or both (Figure S2A). We observed that scPrisma and scVI performed best when we have a cell type dependent amplitude but a cell type *independent* phase (Figure S2A). When the phase varies in a cell type specific manner, either alone or in conjunction with amplitude, scPrisma and scVI's performance degraded. By contrast, SNOW had a near perfect performance in all cases (Figure S2A). We also constructed a toy data that contained only one single cell type (Figure S2B), for which we expect a single cluster once the effect of time is removed. Even in this situation, we observed that neither scPrisma nor scVI could completely remove the temporal effect from the data, and cells continue to cluster by time in the UMAP space (Figure S2B). This observation suggests that the direct enforcement of time–independence imposed by SNOW can remove temporal effect that other methods cannot.

To illustrate the complexity introduced by time in real scRNA-seq datasets, we used UMAP to create lower dimensional embeddings of time–series scRNA-seq data collected from the drosophila clock neurons [10] and the mouse aorta [32], both with existing cell type annotations. We observed that the effect of time strongly drove clustering in the UMAP space (Figure 2D, left column). As illustrated in the top row of Figure 2D, while the UMAP space can separate the smooth muscle cells (SMCs) and fibroblasts, the SMC cluster contains subclusters, each corresponding to different sampling times. This effect is even stronger in the drosophila clock neurons, where the UMAP

projection separates into small, disjoint clusters where each contains cells sampled at a particular point in time, and each such cluster contains cells of different cell types (Figure 2D, second row).

We applied SNOW to both the drosophila and the mouse dataset and observed the successful removal of the temporal effect (Figure 2D, right column). Close examination of the SNOW latent space generated from the drosophila data (Figure 2D, bottom) reveals that we have retained variation attributable to cell type. Adding the original cell-type annotations to the UMAP plot of the SNOW–processed data (Figure 2D, bottom right), we find dorsal neurons ('DN's) located on the top and right side, and lateral neurons ('LN's) on the left (Figure 2D). Interestingly, we observed that a group of dorsal neurons (6:DN1p, 18:DN1p, 19:DN2) and lateral neurons (9:LNd_NPF, 12:LNd) merged into two larger clusters in our latent space (Figure 2D bottom right and Figure S3). On the other hand, we also observed that cluster 14 breaks into at least two smaller clusters (Figure S3). To identify the origin of this discrepancy, we conducted data integration with Seurat [33] (see details in Methods) with the features used to train our model, and made similar observations (Figure S3, S4). As an additional validation, we used SNOW to map all cells onto the same time and generated a similar embedding (S5A). Together, our results suggest that the merging and breaking of clusters in our embedding can be attributed to the small feature set used in the original annotation of cell types.

We next compared SNOW to scPrisma and scVI on the drosophila dataset. In contrast to the toy datasets, we observed that scPrisma performed worse than scVI in the real datasets (Figure S5B and C), where it removed information regarding cellular identity (Figure S5B); and SNOW continued to outperform both. As a further example, we also applied SNOW to a time series dataset charting the regeneration of mouse lungs subjected to bleomycin-mediated injury [34] (Figure S7) and observed that cells significantly affected by bleomycin in the original gene expression space are now embedded closer to their untreated counterpart in the UMAP space generated from SNOW, revealing their common transcriptomic background (Figure S8). To then examine how bleomycin effects these cell types, one can simply combine the time dependent and time independent component to project the data back to the original gene expression space. In comparison, when we applied scVI to the same dataset, its latent space contained four clusters of cells (Figure S7B).

Finally, we tested whether trajectory inference methods such as Monocle [35–37] could be used to articulate the temporal variation. It should be noted here that trajectory inference is a funda-

mentally different problem than that which we are trying to solve here. In trajectory inference, one assumes that the cells observed at a given time–point may be thought of as observations along a pseudo-temporal trajectory, in which cells transition smoothly from one type to another. There is thus the assumption that transcriptomically similar cells are closer in time as the trajectory evolves from one cell to the next. In SNOW, we make the assumption that we are observing mature cells where the dynamics of gene expression in those cells depends both on time (which we observe directly by taking multiple samples over time) and cell type, without making the assumption that cells that are transcriptomically similar cells are closer in time.

Nevertheless, one might expect that a trajectory inference method would place cells of the same type from adjacent time–points closer to one another than cells of a different type, thereby articulating the trajectory of cells by organizing cells of the same type together and correctly ordering them in time. In the drosophila dataset, we would expect that these methods can first identify the fact that different cell types are traversing different trajectories, and for each cell type, identify the fact that cells are traveling along a circular trajectory. However, as illustrated in Figure S6A, time and cell type jointly affect the transcriptome. Without using algorithms like SNOW to disentangle this joint effect, the trajectory inferred by Monocle captured neither cell types nor the fact that cells are traversing a circular path. Using "batch corrected" data also does not solve the problem. Figure S6B, we conducted trajectory inference with Monocle using its built–in method to conduct batch correction. While this space is now more consistent with cell types, the circular paths cells should take have been destroyed during batch correction, since batches and time are confounded. As a consequence, no substantial trajectories are inferred by Monocle.

2.3 SNOW maps cell forward and backward in time

SNOW generates a latent space that is independent of time and contains a decoder that reconstructs the transcriptome when the latent state and time are both supplied. In principle, then, it is possible to provide the latent space and an *unsampled* time to generate an expression profile of a specific cell at another time–point. To test whether we can produce expression dynamics for each cell that resembles the average of its cell type, we generated de novo time series by concatenating the latent representation of a cell, z, with time, t. Because the concatenated t can differ from the sampling time of the cell, we refer to this as the "pseudo" sampling time. We generated time series using

latent representations of the mouse aorta, which is sampled every six hours for one day, by using 100 equally spaced pseudo sampling times. One might then reasonably ask: if the generated data had in fact been observed data, would the encoder network have correctly identified the time that was used to generate the pseudo sample? By supplying the generated expression profile back to the encoder network, we observed that we are capable of re-inferring the pseudo sampling time of each cell accurately (Figure S9A), with a mean absolute error ($|\tilde{t} - \hat{t}|$) of 0.80 and 0.79 hours for the smooth muscle cells and the fibroblasts respectively (Figure S9B). Overlaying the mean absolute error on its UMAP projection identified no regions with particularly large errors (Figure S9C).

To further validate our approach, we averaged the generated time series for all cells from the same cell type and compared this population average to the experimental data (black lines in Figure 3). Using the well–characterized circadian genes as examples, we observed that while the generated expression time series for both the fibroblast and SMC show a considerable amount of diversity, the population average exhibits clear oscillatory dynamics and matches closely with empirical observation (Figure 3A, B). It is worth noting that no constraint was imposed during the training process to shape the generated population average. This observation suggests that the agreement between the observed and the generated dynamics is consequent of a successful deconstruction of the gene expression into time–dependent and time–independent components.

We next repeated this test on the clock neuron dataset, sampled every 4 hours for two days. As a sanity check, we tested whether SNOW's encoder would recover the pseudo sampling time when the data generated by the decoder was fed back into it. We observed that our model remained competent at "predicting" pseudo sampling times (Figure S10A), with an mean absolute error ranging from 1.5 hours to less than 3 hours. Similar to before, we observed SNOW–generated oscillations in known circadian markers in concordance with experimental observation (Figure 3C, D). Despite the proximity of the 1:DN1p_CNMa cluster and the 2:s_LNv cluster in the UMAP space (Figure 2D, bottom left), we observed the mean expression level of the generated expression time series of *CNMa* to differ by ten fold, suggesting our usage of a fixed latent space standard deviation did not prevent the model from learning the dynamics particular to each cell type.

Interestingly, we found that the quality of the generated time series is tied to the size of the latent standard deviation (σ_z). In the clock neuron dataset, we observed that small σ_z leads to damped oscillation in the long run (Figure S10B). However, this effect is not apparent in the mouse heart

data (Figure S10B), potentially because of its larger sample size, simpler cell type composition, and fewer sampling times.

2.4 SNOW corrects batch effects

While batch effects can be difficult to identify and correct, the fact that samples are related in time provides a potential route of correction by prohibiting abrupt changes of expression, formally achieved by constraining the second derivative of the generated time series. To test whether SNOW can reduce the impact of batch effects in time–course data, we first identified genes that have been potentially affected. We consider a gene to be severely impacted by a batch effect if it is mostly detected only at a single time point. For those that are consistently detected across time points, we assume they are affected if their expression level at a particular time point is much higher than that of the rest (see Methods). With these two criteria, we identified 148 genes within the 1:DN1p_CNMa cluster from the clock neuron data and observed that 117 of them are considered to be features by Seurat [33]. As Seurat identifies features by looking for outliers on a mean–variance plot, it is expected, and alarming, that genes satisfying our criteria will be considered as features. By constructing time series using all sampled 1:DN1p_CNMa cells to span the entirety of the experiment, we observed that the generated signal is unaffected by the outlier samples (Figure 4A).

Interestingly, we observed that a large proportion of the selected genes appear to be impacted by a batch effect at time ZT38. Direct visualization of the expression level of putative batch–affected genes on the UMAP space implies that these genes, which were not originally used as features, may contribute to the disagreement between the original cell type assignment and our latent space. For example, Figure 4B illustrates that cells annotated as 1:DN1p_CNMa neurons that had an elevated expression of batch–affected genes are located away from the main cluster. This suggests that what appears to be batch effect may simply be an artifact of bad cell type annotation. Since we can compute the likelihood of making an observation, if cells considered to be 1:DN1p_CNMa neurons at ZT38 were, in fact, of some other origin, cells collected at ZT38 would stand out from the rest of the time series, but the log–likelihood would not. To test this, we computed the log likelihoods of observing the experimental data and observed that gene-wise log likelihood also shows a sharp drop at the time when gene expression peaks (Figure S11), indicating that the observed expression level has a low probability of occurrence under our statistical model. Computing the log likelihood

of observing the entire cell by summing up the probabilities of observing each gene, we noticed a drop at ZT38 for almost all cell types (Figure 4C, S11C), in agreement with our observation that a large fraction of the identified genes were impacted at ZT38. Additionally, this drop of log likelihood at ZT38 remained even when all cells were pooled together (Figure S11B), suggesting that the expression peaks we observed at ZT38 cannot solely be attributed to cell type assignment.

To summarize, we showed that SNOW can generate time series that are unaffected by outlier samples and that our underlying statistical framework is capable of detecting batch–affected genes.

2.5 SNOW allows unsupervised identification of circadian rhythms in gene expression

The discovery of tissue–specific circadian regulation [38] and advances in single cell technologies have led to studies that report cell-type specific circadian oscillation [10, 11]. While circadian time series conducted on the tissue level can be directly supplied to a number of readily available cycling detection algorithms [39, 40], single cell data requires some special considerations. First, proper cell type annotation requires the removal of all temporal effects. While this can be achieved via data integration, integrated data cannot be used for cycling detection, forcing users to conduct cell type annotation with integrated data but perform cycling detection with "raw" data. Moreover, one needs to choose whether to consider each cell as a replicate or to construct pseudobulk data for each time point. However, considering cells as replicates can be highly computationally inefficient, and it has been shown that constructing pseudobulk profiles can generate false positives, especially for genes with low expression [41].

With SNOW, we can generate a transcriptome-wide time series for each cell by projecting them forward and backward in time, thus enabling us to conduct cycling detection at the single cell level and detect fine expression differences between cell types (Figure S12). For each cell, we can obtain a p value, phase estimate, and estimated amplitude for each gene via harmonic regression on its generated time-series. To test if the p-values obtained from the generated data reflect trends observed in the experimental data, we compared our results to the published list of per–cell–type cycling genes [10]. As demonstrated in Figure S13, genes that were reported to have rhythmic expression in multiple cell types had smaller average (across all cells) p-values and larger average (across all cells) amplitudes.

We then investigated the biological interpretation of the cell–level statistics. Unsurprisingly, known circadian genes *vri* and *tim* had the lowest average p-values across all cells. Overlaying harmonic regression p-values on the UMAP space showed that the core clock genes *vri*, *tim*, *Clk*, and *per* are highly cyclic in all cells (Figure 5A), as expected. We also overlayed the estimated oscillation amplitude and phase for each cell (Figure 5A). Interestingly, we observed high oscillation amplitudes of *vri* and *tim* in all labeled clusters. By contrast, cluster 16, an unnamed cluster, stood out for having a much lower amplitude despite its close proximity to the high–amplitude dorsal neurons on the UMAP space. Additionally, despite the fact that the phases of *vri*, *tim* and *Clk* were reported to be largely identical across cell types in the original (pseudobulk) analysis [10], we observed that SNOW is capable of discerning fine phase differences between clusters on a single cell level (Figure 5A).

We also observed that there are cases where the harmonic regression p-values from flat genes are low, which leads to disagreement between our analysis and the published cycling genes. By looking at the estimated amplitudes, we found that these disagreements can be resolved by using amplitude criteria that exclude cells/clusters with low oscillation amplitudes (Figure S14). For example, *sky*, which was reported to be cycling in the 2:s_LNv and 1:DN1p_CNMa clusters, also appeared to be cycling in two other DN1p clusters (Figure 5B, left panel). While the estimated amplitudes of *sky* from the two DN1p clusters are smaller than that of the 1:DN1p_CNMa cluster, they are similar to that of 2:s_LNv neurons (Figure 5B, right panel). A closer look at the time series generated from the two DN1p groups revealed expression dynamics distinct from that of 1:DN1p_CNMa but similar to that of 2:s_LNv, suggesting *sky* may be cycling in a larger population of dorsal neurons than previously believed (Figure 5C). Another gene, *Ddc*, which was also reported to cycle in the 2:s_LNv neurons, showed high p-values and low amplitudes in our analysis (Figure S15A). Comparing SNOW–generated time series to the experimental observations (Figure S15B) suggests that this may have been a false positive in the original analysis. On the other hand, we observed that two dorsal neuron groups (7:DN1p, 20:DN3) in which *Ddc* was not reported to be cycling originally showed low p-values and high amplitudes in the SNOW generated data (Figure S15B), possibly suggesting a false negative (Figure S15B).

In summary, we showed that SNOW may be used to help the identification of rhythmic genes by first generating time series for each cell, and then conducting cycling detection on a single cell level.

By doing this, cycling detection analysis does not depend on the accuracy of cell type annotation. This suggests that it can be used in combination with traditional analyses that first assign cell types prior to pseudobulking for cycling detection. For example, it can increase the confidence in the identification of cycling genes by confirming that they are rhythmic in the majority of individual cells; detect potential false negatives in the pseudobulk analysis (especially for rare cell types that may not be sampled at all time points); and avoid false-positives by removing potential batch effects. It can also potentially identify subsets of cells of a single type (or a single cluster) that are differentially cycling, an effect that may be missed in analysis where cells of the same type are treated as replicates or pseudobulked for cycling detection.

3 Discussion

We presented SNOW (SiNgle cell flOW map), a deep learning framework for the annotation, normalization, and generation of single–cell time scRNA-seq data. SNOW computes and maximizes the log likelihood of the experimental observations by taking the raw count data as input. The count data is modeled to follow a zero-inflated negative distribution, similar to previous works [31, 42]. SNOW then deconvolves the data internally into time-dependent and time-independent components by minimizing the sliced Wasserstein distance between relevant distributions. To enforce the time independence, we took the innovative approach of computing the sliced Wasserstein distance between $q(z|t)$ and $p(z)$, which scales linearly with the number of cells. This approach is a simpler alternative to computing the maximum mean discrepancy (MMD), which scales quadratically with sample size. The time–independent component can be used readily for cell type annotation, and the time–dependent component can be used to generate artificial time series for individual cells (in a manner similar to a flow map, but without the need or computationally expensive numerical integration). We demonstrated the utility of SNOW by applying it to multiple single–cell datasets with different cell numbers, sampling frequencies, and sequencing depths.

SNOW has a number of advantages. First, most methods for analyzing single cell time series data focus on developmental processes, in which the effect of time and cell type are associated. These methods largely rely on finding an optimal transport map between cells sampled at distinct time points [15, 17, 19, 20]. While such methods are appropriate when cells are gradually transitioning

from one state to another following the same general trajectories (as illustrated in Figure 1A, top), it is difficult to apply them to time series from mature cells where expression changes with time in a cell type–specific manner. SNOW addresses this issue by deconvolving the effect of time and cell type.

Second, we demonstrate that SNOW can be used to identify and eliminate batch effects. By modeling count data with a zero inflated negative binomial distribution, we were able to identify samples from the clock neuron dataset that are likely to be batch-affected by using the estimated probability of observing their gene expression profiles. From these samples, we observed an interesting association between aberrant gene expression and the drop of log likelihood, empirically confirming the validity of our approach. As illustrated in Figure 4A, by generating time series on a single cell level and constraining their second derivatives, the effect of a batch-affected time point can be mitigated.

Third, SNOW is capable of generating time–series data for individual cells, providing a novel avenue for single–cell time–series analysis. Previous work has shown that even seemingly simple tasks such as cycling detection (identifying genes with circadian oscillation) are non-trivial in single–cell data. When cells are considered as replicates, the large number of cells collected from single cell experiments leads to significant drop of p-values, generating false-positive cycling genes. On the other hand, if pseudobulk transcription profiles are used, the effectiveness of the method becomes limited by the number of time points available (usually less than 10) and false positives can arise in genes that are almost never detected but happen to have an average expression that appears sinusoidal in time due to a handful of outlier cells [41]. Here we demonstrated how SNOW can be used to enhance the analysis of circadian signals by conducting cycling detection for individual cells. Since the analysis is done for each cell individually, p-values will not deflate as sample size increases; and since analysis is done for all cells, we maintain a holistic view of gene-level statistics across cells that would be lost by pseudobulking.

It should be noted, of course, that those "per-cell" p-values are only as good as the generated data, and thus should be thought of as "hypothesis generating". The fact that the generated data recapitulates the observed expression data (as illustrated in Figures 3 and 5) suggests that inferences made from the generated data are not unreasonable. Why, then, might someone want to perform the analysis using hypothetical generated data rather than (or in addition to) real observations?

Consider the analysis protocol for the observed data, where the cells observed at each time–point are distinct cells (indeed, the number of cells may even change from one time–point to the next). Since these are just a series of independent observations rather than a true timeseries (where the same individual is followed over time), one must decide which cells to include at each time before doing an analysis. This selection is important, because some cell types may be out of phase with others; if all cells types are selected, the signal may be washed out. However, cell type identification remains an open problem, and any errors will propagate to downstream analyses. As mentioned above, having selected the cells of interest at each time–point, one then has to make the further choice of whether or not to average (pseudobulk) them prior to analysis (a choice which is also not free of complexities, as noted above). Typically, pseudobulking is conducted (as in [10]), yielding a single pseudobulk observation at each time–point that is considered "representative" of that cell-type at that time. The analysis then proceeds on the pseudobulk series as it does for the single per-cell timeseries in our generated data. Clearly, for this approach to be successful, one needs (i) to know the cell-types of interest (either *a priori* or based on clustering of the integrated data) and (ii) to have correctly identified them for the pseudobulking. One also makes the assumption that cells of the same type have similar dynamics, ie, that there is no heterogeneity within a particular cell type.

By contrast, using the SNOW-generated per-cell data does not require the identification of cell types or clusters. This avoids propagating an errors in cell type identification. It also means that we are able to detect heterogeneous dynamics within cell–types, rather than making the implicit assumption that all cells of a given type will behave similarly. This approach can be paired with traditional analyses to increase the confidence in detected cycling genes and potentially identify false positive/negative cyclers from traditional analyses. This capability can be useful in non-circadian contexts as well, such as when one wishes to identify genes that follow a specific pattern during differentiation or drug perturbation.

The ability of SNOW to generate cell–level trajectories also enables the generation of novel hypothesis that would not be possible otherwise. For instance, one could perform analysis shown in Figure 5 and then classify or cluster the cells based not on the level of gene expression, but based on the dynamics (phase and amplitude) of gene expression. One could bring cell identity back into it to ask: do all cells of a particular type have the same dynamics (phase, amplitude), or is there heterogeneity for certain cell types? Other analyses are also possible. For example, SNOW allows

one to identify genes that may mediate the amplitude of circadian oscillation by looking for genes with an expression level that correlates with oscillation amplitude of the core clock genes in each cell. Additionally, one could also imagine supplying SNOW-generated time series data to network reconstruction algorithms to further dissect the details of gene-gene interaction.

Several existing methods bear some similarities to SNOW, with important differences. scVI [31], DCA [42], and many other methods [43–46] are all built on variational autoencoders [18], which are typically trained via the optimization of the evidence lower bound (ELBO). However, as the ELBO only constrains the latent space via the KL divergence (Eq 6), it may generate correlated latent dimensions and fail to enforce the assumption that the prior distribution has identity covariance, enlarging the difference between ELBO and the actual log likelihood, $\log(p(x))$. In this situation, one would fail to generate "realistic" virtual gene expression profiles by passing samples drawn from the prior distribution through the decoder network. While having an "irregular" latent space that fails to match the prior distribution may not impact the performance of the model in other tasks such as clustering and identifying cell types, enforcing independence between the latent dimensions is known to improve model interpretability [47, 48]. To address this, scNODE [17] and scVIS [43] introduced a scaling factor added to the KL divergence term (similar to β-VAEs [47]) to enforce a stronger constraint on the latent space, thereby encouraging a more efficient representation of the data. More recently, various methods have been proposed [48–51] to directly enforce independence between latent dimensions via minimizing $D_{KL}\left(q(z)||\prod_d q(z_d)\right)$. In SNOW, we enforced independence between latent dimensions and alignment with respect to the prior distribution simultaneously by minimizing the sliced Wasserstein distance.

As mentioned in previous sections, we developed SNOW to solve the following problem: in the situation where both cell state and time affects gene expression, removing temporal effects to facilitate cell type annotation also removes biologically meaningful gene expression dynamics. This problem is related to what MrVI [44] attempts to solve by constructing a sample-unaware representation (u) and a sample-aware representation (z), where u is used to conduct cell type annotation and z is used to model how sample related covariates (such as a batch or a time–point) affect gene expression. In some sense, SNOW and MrVI are designed to solve the same problem, except that SNOW specializes in continuous covariates (time) and MrVI in discrete covariates. Our explicit enforcement of statistical independence between the latent space and time, which is absent

in both MrVI and scVI, naturally defines cell state as a time–invariant quantity. By supplying the decoder with time and the time–independent representation of cell type, SNOW can generate data "sampled" from intermediate time points, which cannot happen if time is simply treated as batch label, as it is in MrVI. SNOW also has the additional benefit of enforcing smoothness by constraining the second derivative with respect to time, which is not possible if time is treated as a categorical variable.

We made a few assumptions during the construction of our framework. First, we built the latent representation of each cell as a time–invariant object. For mature cells, we observed that this time–invariant object corresponds to cell type. Biologically speaking, this assumption can hold as long as cells of the same cell type constantly express reliably detectable type–specific marker genes. In developmental systems, this time–invariant object should in turn capture the lineage of each cell if lineage–specific markers are being expressed. However, when gene expression undergoes substantial changes and no lineage or cell type–specific markers are present, our first assumption will be violated. Second, we assumed that gene expression is predominantly affected by two components, namely cell type (or lineage) and time. This assumption implies that our framework is not applicable to developmental systems where bifurcations are present. For example, if a stem cell population differentiated into three distinct cell types at $t = 5$, all expressing the same lineage–specific markers, the decoder cannot generate three distinct set of gene expression profiles when the input lineage (stem cell) and time ($t = 5$) is fixed. In the situation when gene expression of each cell changes along the same non-bifurcating trajectory but with different speed, given our second assumption, the input time for the model should be replaced by the estimated pseudotime of each cell in order to to correctly identify lineage. As SNOW is thus only applicable to a small subset of developmental processes, we recommend using SNOW to analyze dynamics in mature systems (such as the response to a perturbation).

We also acknowledge that SNOW did not consider the effect of growth, which is the focus of a number of existing approaches [15, 20, 25]. It was shown that taking growth into consideration may can improve velocity estimate by removing false cell state transitions [25]. However, in the problem we seek to address, we reason that growth is less relevant because we are not examining dynamics generated by cell state transitions, which can be confounded by cell division. Rather, the dynamics that we study are intrinsic to each cell, such as the self-driven circadian oscillation of the drosophila

clock neurons subtypes.

Nonetheless, we expect our work to be of interest to those studying dynamic processes in complex tissues. Additional features can be easily added into our method to handle more complex datasets, and approaches employed in our work, such as data integration or the enforcement of statistical independence, can also be extracted and adopted for other analyses.

4 Conclusion

Here, we present SNOW, a novel computational approach to characterize single cell time series data through the decomposition of gene expression into a time–dependent and a time–independent space. Our approach allows the unprecedented opportunity of conducting time series analysis on the level of single cells through the generation of cell–level data. On the biological side, SNOW can be used to detect and mitigate batch effects, as well as to generate novel hypotheses and guide the attainment of novel insights. For example, it can be used to identify genes whose expression level correlates with the amplitude of the oscillation of core clock genes to further elucidate the mechanism that underlies the circadian rhythm. On the modeling side, SNOW is an example in which a single model is applied to capture the diverse gene expression dynamics of multiple different cell types. Together, SNOW has the potential to uncover novel biological paradigms in diverse backgrounds as well as driving the development of innovative mathematical approaches to characterize and model complex high-dimensional datasets.

5 Methods

5.1 SNOW's probabilistic framework

SNOW models the count matrix $\mathbf{X} \in \mathbb{R}^{C \times G}$ with a zero-inflated, negative binomial (ZINB) distribution [31, 42], where C and G are the number of cells and genes in the sample, respectively. Without zero-inflation, a given entry within $\mathbf{X}$, X_{cg}, is modeled as:

$$P\left(X_{cg} = y | z_c, t\right) = \frac{\Gamma(y + \theta_{cg})}{\Gamma(y+1)\Gamma(\theta_{cg})} \left(\frac{\theta_{cg}}{\theta_{cg} + \rho_{cg} l_c}\right)^{\theta_{cg}} \left(\frac{\rho_{cg} l_c}{\theta_{cg} + \rho_{cg} l_c}\right)^{y}, \tag{1}$$

where $\Gamma(\cdot)$ is the standard gamma function, z_c the (time-independent) encoded state of $\mathbf{X}_c$ sampled at t, θ_{cg} the gene– and cell–specific inverse dispersion, l_c the library size of cell c, and ρ_{cg} the count fraction of gene g in cell c such that $\sum_i \rho_{ci} = 1$. θ and ρ are optimized using neural networks f_θ and f_ρ respectively.

Zero-inflation is added with the following form:

$$P(X_{cg} = 0|z_c, t) = \underbrace{1 - (f_h(\mathbf{X_c}))}_{\substack{\text{observing zero count} \\ \text{due to dropout}}} + \underbrace{f_h(\mathbf{X_c}) \left(\frac{\theta_{cg}}{\theta_{cg} + \rho_{cg} l_c} \right)^{\theta_{cg}}}_{\text{observing “true” zero}} \tag{2}$$

$$P(X_{cg} = y|z_c, t) = f_h(\mathbf{X_c}) \frac{\Gamma(y + \theta_{cg})}{\Gamma(y+1)\Gamma(\theta_{cg})} \left(\frac{\theta_{cg}}{\theta_{cg} + \rho_{cg} l_c} \right)^{\theta_{cg}} \left(\frac{\rho_{cg} l_c}{\theta_{cg} + \rho_{cg} l_c} \right)^{y}, \tag{3}$$

where $f_h(\cdot)$ is parameterized with a neural network. Since elements of $\mathbf{X}_c \in \mathbb{R}^G$ are conditionally independent of each other given z and t ($\forall i \neq j$, $P(X_{ci}|z, t, X_{cj}) = P(X_{ci}|z, t)$), we can compute the probability of observing the count profile of a particular cell as:

$$P(\mathbf{X}_c = \vec{y}|z_c, t) = \prod_i P(X_{ci} = y_i|z_c, t). \tag{4}$$

Or equivalently:

$$\log\left(P(\mathbf{X}_c = \vec{y}|z_c, t)\right) = \sum_i \log\left(P(X_{ci} = y_i|z_c, t)\right). \tag{5}$$

Our framework allows the generation of “virtual” cells by assuming a Gaussian prior, a commonly

used prior for building variational auto-encoders, as follows:

$$
\begin{aligned}
z_c &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
\rho_c &= f_\rho(z_c, t_c) \\
\theta_c &= f_\theta(z_c, t_c) \\
w_c &\sim \text{Gamma}\left(\theta_c, \frac{\rho_c}{\theta_c}\right) \\
y_c &\sim \text{Poisson}(l_c w_c) \\
h_c &\sim \text{Bernoulli}\big(f_h(z_c, t_c)\big) \\
x_{cg} &= \begin{cases} y_{cg}, & \text{if } h_{cg} = 1 \\ 0, & \text{otherwise} \end{cases}
\end{aligned}
$$

where z_c is the time-independent latent representation of a cell; $\rho_c \in [0, 1]$ is the normalized expression profile (or count fraction) enforced by using a softmax activation function in the last layer of f_ρ; $x_c \in \mathbb{N}^G$ is the count profile of the virtual cell; and l_c is the observed the library size. The Gamma-Poisson process generates $y_c \in \mathbb{N}^G$ following a negative binomial distribution with mean $\rho_c l_c$, while h_c is a binary vector that represents dropouts. f_ρ, f_θ, and f_h are neural networks that map the latent space and time back to the full gene space, $\mathbb{R}^D \times \mathbb{R}^1_+ \to \mathbb{N}^G$.

5.2 SNOW loss function

A number of methods have used variational autoencoders (VAEs) [18] to model count data from single-cell RNA seq [17, 31, 43]. All have used loss functions reminiscent of the evidence lower bound (ELBO), which constrains the shape of the latent space $q(z)$ indirectly via the KL-divergence term:

$$
\text{ELBO} = \mathbb{E}_{z \sim q(z|x)} \log p(x|z) - D_{KL}\left(q(z|x) || p(z)\right) . \tag{6}
$$

(See derivation of ELBO in Supplement.) In the above expression, $p(z)$, the prior distribution of the representations z, has been chosen for convenience to be $\mathcal{N}(\vec{0}, I)$ and $q(z)$ is the variational posterior distribution of z constructed by the encoder network. The KL-divergence term provides the model some level of robustness, as it essentially requires points near z in the latent space to be decoded

into similar objects. However, as the dimensionality of the data grows, the log-likelihood term of the ELBO will dominate over the regularizing KL-divergence term. While this is unaccounted for in scVI [31], both scNODE [17] and scVIS [43] incorporate scaling factors to maintain the strength of the regularization of the latent space. By definition, maximizing ELBO leads to the maximization of the marginal log likelihood ($\log p(x)$),

$$\log p(x) = \text{ELBO} + D_{KL}\left(q(z|x)||p(z|x)\right) \ . \tag{7}$$

When $D_{KL}\left(q(z|x)||p(z|x)\right) = 0$, or equivalently $q(z|x) = p(z|x)$, the ELBO will be equal to the marginal log likelihood of x and $p(z) = \int q(z|x)p(x)dx = q(z)$. However, when the ELBO is not tight, its optimization can lead to an enlargement of the approximation error, $D_{KL}(p(z|x)||(q(z|x))$. To account for this, SNOW regularizes the latent space directly by minimizing the distance between the latent distribution ($q(z)$) and the prior ($p(z)$) as measured by the Wasserstein distance. Briefly, in addition to the log likelihood term, the SNOW loss function begins with two main regularization terms, the former of which regularizes the latent space and the latter of which enables predictions of the sampling time:

$$-\log\left(p(x|z)\right) + \lambda_z \mathcal{L}_z + \underbrace{\lambda_t ||t - \hat{t}||_2}_{\substack{\text{predicting} \\ \text{sampling time}}} \ . \tag{8}$$

In the above expression, $\mathcal{L}_z$ regularizes the latent space $q(z)$ and enforces time–independence via:

$$\mathcal{L}_z = \underbrace{W_2\left(q(z), \mathcal{N}(\vec{0}, I)\right)}_{\substack{\text{regularizing the shape} \\ \text{of the latent space}}} + \underbrace{\sum_i W_2\left(q(z|t=i), \mathcal{N}(\vec{0}, I)\right)}_{\text{ensuring latent–space time–independence}} \ , \tag{9}$$

where $W_2(q, p)$ denotes the Wasserstein-2 distance between distributions p and q. This regularization enables the generation of a "virtual" cell when z is sampled from $\mathcal{N}(\vec{0}, I)$. To ensure our model can generate proper "synthetic" cells sampled from different time points, we enforced two things. First, the time–independent components of the "synthetic" cells should follow the same distribution as that of the real cells (a Gaussian distribution). Second, the sampling time of the "synthetic" cells

should remain predictable. To achieve this, we therefore impose:

$$\mathcal{L}_{\tilde{t}} = \underbrace{\lambda_{z,\tilde{t}} W_2 \left(\mathcal{N}(\vec{0}, I), q(z|x(\tilde{t}) \right)}_{\text{latent space preservation}} + \lambda_{\tilde{t}} ||\tilde{t} - \hat{t}||_2, \tag{10}$$

where $\tilde{t}$ is the sampling time of the "synthetic" cells. And finally, we constrain the second derivative of the generated time series to enforce smoothness:

$$\mathcal{L}_s = \sum_{i=1}^{G} \frac{1}{\bar{x}_i} \left|\left| \frac{d^2 x_i}{dt^2} \right|\right|_\infty , \tag{11}$$

where G is the number of genes and $\bar{x}_i$ is the average of x_i over all generated time points. In practice, we find that computing $\mathcal{L}_s$ for a randomly selected gene, r, in each training loop to be computationally cheaper and sufficient to generate smooth time series, giving the final form of our loss function:

$$\mathcal{L} = -\log\left(p(x|z)\right) + \lambda_z \mathcal{L}_z + \lambda_t ||t - \hat{t}||_2 + \mathcal{L}_{\tilde{t}} + \lambda_s \frac{1}{\bar{x}_r} \left|\left| \frac{d^2 x_r}{dt^2} \right|\right|_\infty , \tag{12}$$

which preserves the latent space distribution, its time independence, and ensures the smoothness of the generated time series.

In practice, we simplify the calculation by replacing the Wasserstein-2 distance W_2 with a more computationally tractable form, the sliced Wasserstein distance ($\hat{W}_2$) [52], defined as:

$$\hat{W}_2(p, q) = \int_{\omega \in \Omega} W_2 \left(p^{(\omega)}, q^{(\omega)} \right) d\omega, \tag{13}$$

where the distributions $p^{(\omega)}$ and $q^{(\omega)}$ can be generated by first sampling from p and q directly before projecting them in a random direction, ω, sampled uniformly from the unit sphere $\hat{\Omega}$. Given a set of data points $\{x_i\}_{i=1}^n$ with an unknown underlying distribution $q(x)$, the sliced-Wasserstein distance with respect to a known distribution, such as the standard normal, can be easily computed as:

$$\hat{W}_2(\mathcal{N}(\vec{0}, I), q(x)) = \frac{1}{|\hat{\Omega}| n} \sum_{\omega \in \hat{\Omega}} ||\omega^\top X - \omega^\top Y||_2, \tag{14}$$

where $y \sim \mathcal{N}(\vec{0}, I)$ and we assume the columns of X and Y are sorted such that elements of both

$\omega^\top X$ and $\omega^\top Y$ are arranged in ascending/descending order.

5.3 Neural network optimization

By default, SNOW uses a 3–layer encoder neural network with 256 fully connected neurons per layer and ReLU activation to project count data onto a 32 dimensional latent space (z_c). Subsequently, z_c and t were used as input to individual neural networks (f_ρ, f_θ and f_h) with the same structure as the encoder network to generate the count fraction, inverse dispersion and dropout probability. To ensure that f_h generates probabilities, its last layer is activated by a sigmoid function so that its output ranges from 0 to 1. We further clamped the dropout probability between 0.01 and 0.99 to prevent the appearance of $\log(0)$. As mentioned above, the last layer of f_ρ is activated by a softmax function to enforce the sum of its output. During each training loop, we focus only on a randomly selected small subset of the data, by default 300. Everything within the loss function is computed from information contained within this subset of 300 cells, which enables our method to be applied to larger datasets in a memory–efficient manner.

In all test cases, the optimization of the model parameters was done with the ADAM [53] optimizer as implemented by pytorch [54] with a learning rate of 0.0005, $\beta_1 = 0.8$, $\beta_2 = 0.9$, and a weight decay of 0.0001. No scheduler was used to change the learning rate during the training process.

5.4 Construction of toy datasets

To test SNOW with data where the dynamics are known, we constructed a "toy" dataset containing two types of genes: "flat" genes and "rhythmic" genes. Each cell is randomly assigned a sampling time, t, from $[0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48]$. We simulated the expression of 1000 cells. In Figure 2, we simulated 120 rhythmic genes and 30 flat genes. In Figure S2, we simulated 50 rhythmic genes and 100 flat genes. For the ith flat gene in cell type j, its expression level at time t ,$x_i(t)$, is defined as:

$$x_i(t_i) = \texttt{round}(b_{ij} + \epsilon_f), \tag{15}$$

where ϵ is added noise, b_{ij} is the basal expression of the flat gene i in cell type j and $\texttt{round}(\cdot)$ maps its input to the nearest integer. For the ith rhythmic gene in cell type j, its expression level at time

t is defined as:

$$x_i(t_i) = \texttt{round}\left(a_{ij}\left(\cos(2\pi\frac{t_i - \phi_{ij}}{24}) + \epsilon_c + 1\right)\right), \tag{16}$$

where a_{ij}, ϕ_{ij} is the amplitude and phase of gene i in cell type j, and the $+1$ term lifts the cosine wave to have a zero minimum. Amplitudes a_{ij} and baseline expressions b_{ij} were chosen uniformly at random on $[0, 20]$, and phases ϕ_{ij} were chosen uniformly at random on $[0, 2\pi)$. Noise levels for the flat and cycling genes were drawn on $\epsilon_f \sim \mathcal{N}(0, 2)$ and $\epsilon_c \sim \mathcal{N}(0, 0.1)$, respectively. In Figure 2, a_{ij} is only gene dependent thus it can be reduced to a_i. In Figure S2A, we considered the situation when only a, only ϕ, or both a and ϕ are cell type dependent. Negative values are thresholded to 0.

5.5 Application datasets

We applied SNOW to three sc-RNAseq datasets, including a circadian drosophila clock neuron dataset, a circadian mouse aorta dataset, and a lung regeneration dataset.

The circadian drosophila clock neuron dataset The drosophila clock neuron dataset [10] (mean UMI/cell = 20060) was collected from *Drosophila* clock neurons every four hours with two replicates (12 time points in total) under both light-dark (LD) and dark-dark (DD) cycles. We focused our analysis on cells subject to the LD cycle, which contains 2325 cells. Count data was downloaded from the Gene Expression Omnibus under the accession code GSE157504 and the relevant metadata from https://github.com/rosbashlab/scRNA_seq_clock_neurons. Data integration was conducted using the IntegrateData function from Seurat [22] with ndim = 1:50, and k.weight=100. The resulting counts were used as input to the model.

The circadian mouse aorta dataset The mouse aorta dataset [32] (mean UMI/cell = 14181) was collected every 6 hours (4 time points in total) under LD conditions, with a total of 21998 cells. H5ad files of the smooth muscle cells (SMC) and fibroblasts were downloaded from https://www.dropbox.com/sh/tl0ty163vyg265i/AAApt14eybExMMPK7VVDmfvga. Raw counts were used as input to the model.

The lung regeneration dataset The lung regeneration dataset [34] (mean UMI/cell = 1585) was collected every day for two weeks (day 1 through day 14), and on day 21, 28, 36 and 54. We used AT2

cells, cilliated cells and club cells because they are activated after bleomycin treatment, resulting in a total of 24383 cells. Gene expression data were downloaded from https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE141259 along with the associated metadata. Raw counts were used as input to the model.

5.6 Application of compared algorithms

scVI [31] scVI is implemented following https://docs.scvi-tools.org/en/stable/tutorials/notebooks/scrna/harmonization.html. Raw count data is used as input to scVI, and the sampling time of each cell is used as the batch key. We chose a ZINB to model gene expression and set network hidden unit to 256 and the dimension of the latent space to be 32 to match that of SNOW.

scPrisma [30] scPrisma is implemented following https://github.com/nitzanlab/scPrisma/blob/master/tutorials/cpu/tutorial_prior_knowledge_linear_and_cyclic_cpu.ipynb using the `filtering_cyclic_torch` function. The resulting filtering matrix $\mathbf{F}$, which has the same dimension as the data matrix ($\mathbf{X}$), is used to filter the data matrix by element-wise multiplication.

Monocle [35–37] Monocle3 is implemented following https://cole-trapnell-lab.github.io/monocle3/docs/trajectories/ using default parameters.

5.7 Criteria for identifying batch effects

To identify genes potentially affected by batch effects, we looked for two types of patterns: spurious expression and spurious detection. We consider a gene to have spurious expression if its maximum normalized expression at one time point is five times higher than its average over all time points; and we consider a gene to have spurious detection if its maximum capture rate (number of cells that contain a said gene over the total number of cells collected at this time point) at one time point is five times greater than its mean capture rate over all time points. Here, we used the empirical ρ as normalized expression. To exclude genes that are almost never detected, we only used those with an average normalized expression (across all time points) greater than 0.00001 and an average capture rate (across all time points) over 5%. In the clock neuron dataset, this analysis identifies 124 genes with unusual capture rates and 24 genes with unusual expression, with zero intersection.

5.8 Detecting circadian behavior on a single cell level

To conduct cycling detection for each individual cell, we generated pseudo samples for each cell by concatenating the time independent representation of the cell state, z, with time, t. Using the SNOW decoder, we generated time series comprising 24 time points spanning 48 hours. We then conducted harmonic regression on these time series, resulting in a p value, a phase estimate, and an amplitude estimate for each gene from each cell.

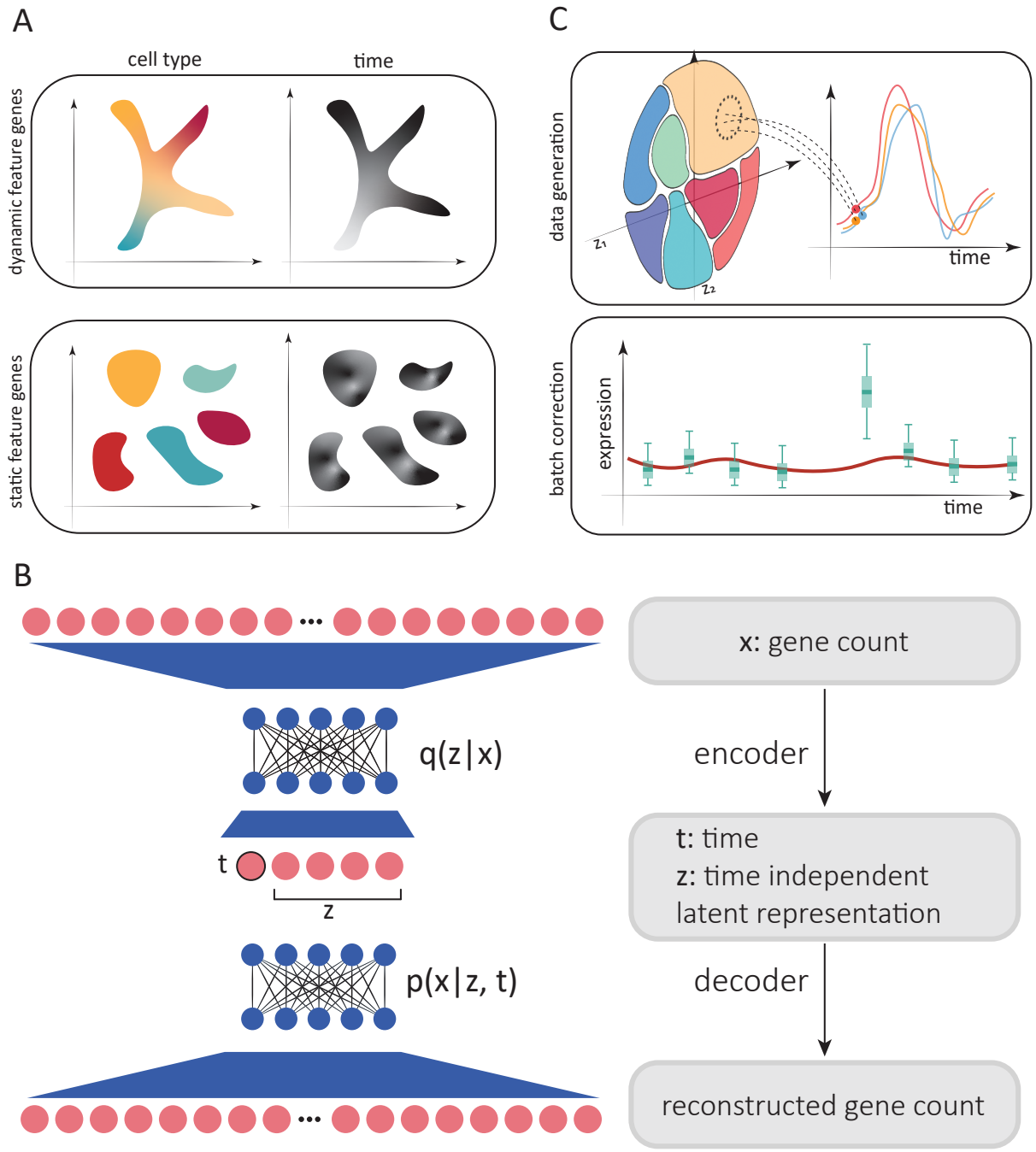

Figure 1: SNOW overview. A: Two possible scenarios of temporal effects in scRNA-seq time–series data. Top: Cell states and time are related, as earlier cell states transition into new cell states (such as during development). Bottom: Discrete cell states exhibit cell type–specific dynamics (such as circadian dynamics in mature cells). B: Simplified architecture of employed neural network. Count data is compressed and deconvolved into time–dependent and time–independent components. C: Top: Generation of synthetic per–cell time series by sampling from the time–independent latent space and then modifying the time–dependent component to project cells forward and backward in time. Bottom: Batch effect correction and imputation by constraining the second derivative of generated time series.

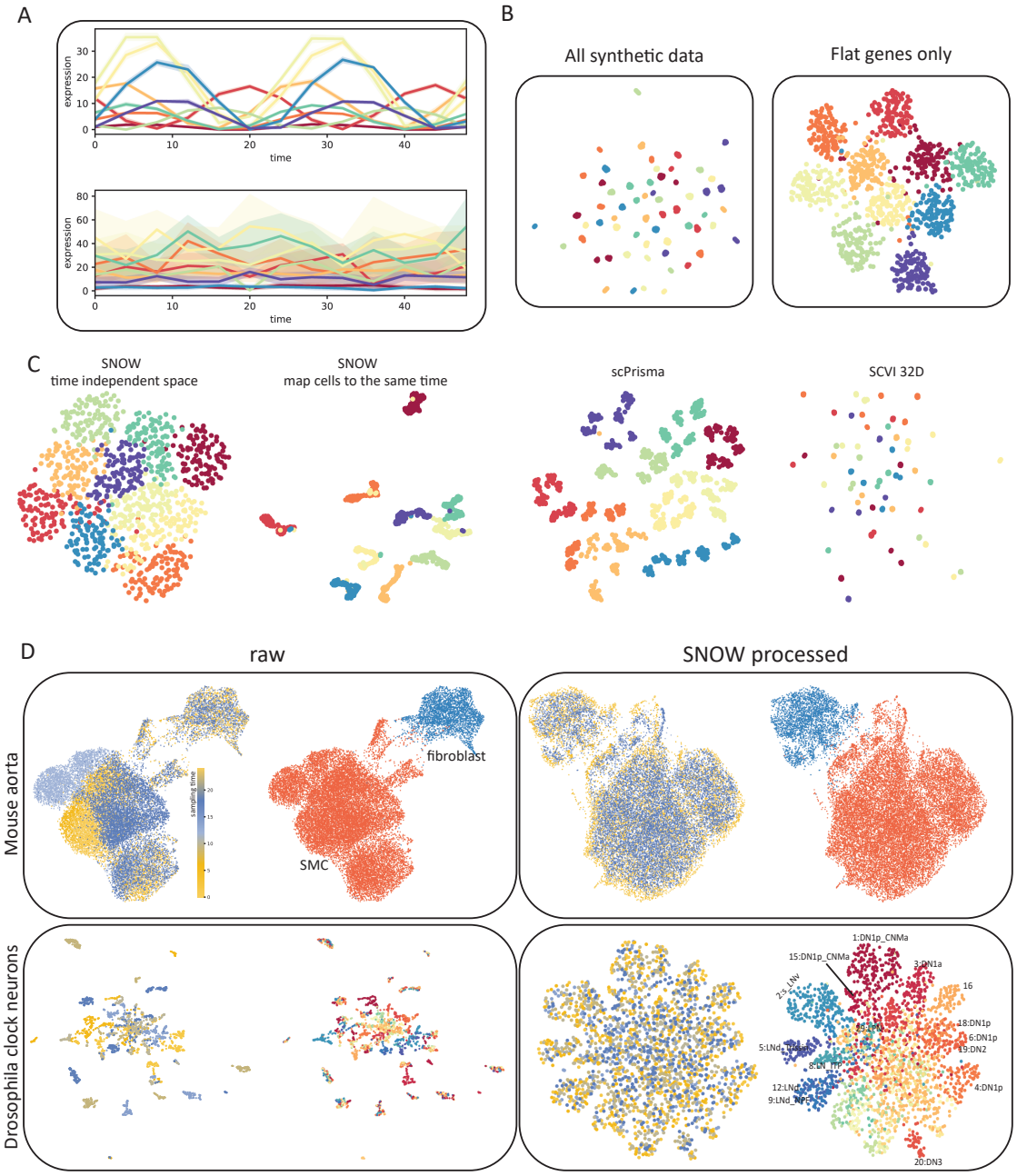

Figure 2: SNOW removes temporal effects while preserving biologically meaningful structure. A: Example toy data that consists of a rhythmic and a flat component, colors denote cell types. B: UMAP projection of the toy data using all genes and using only the flat component. C: Low-dimensional projections of the data generated by SNOW, scPrisma and scVI. Colors denote cell types D: Shown are UMAP plots of a mouse aorta dataset (top) and a drosophila neuron data dataset (bottom) using unprocessed (left) and SNOW-processed (right) data. Within each panel, the UMAP plots are colored according to annotated cell type (right image) and sampling time (left image). In the mouse aorta data without correction (top left), time separates the smooth muscle cell (SMC) cluster (orange) into subclusters. In the SNOW embedding of the same data (top right), the temporal effect has been removed and the SMCs and fibroblasts remain separated (top right). In the drosophila neuron dataset, UMAP shows clusters strongly dominated by time in the unprocessed data (bottom left), but by cell type in the processed data (bottom right). Cell type annotations by Ma et al. [10] are shown.

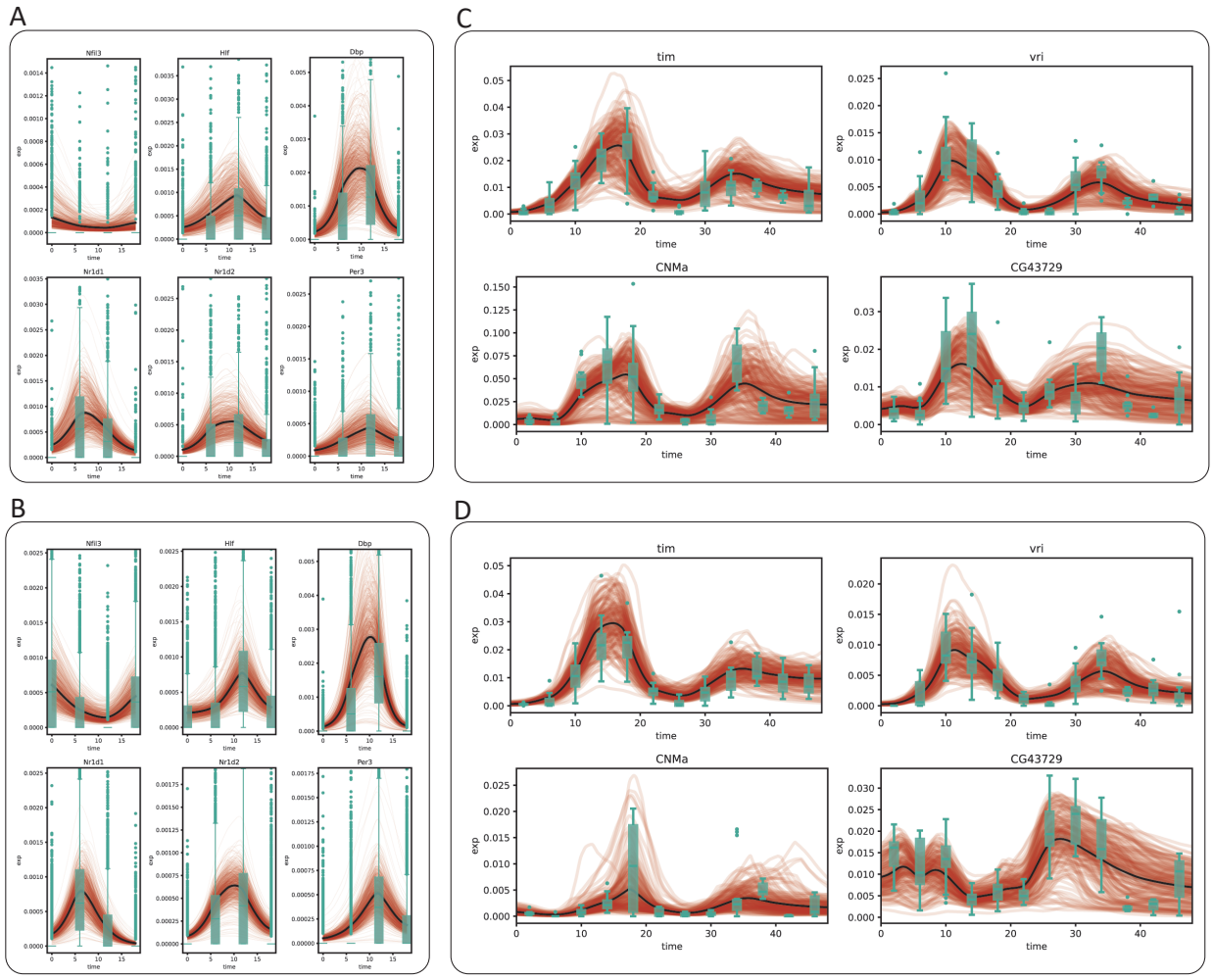

Figure 3: SNOW generates cell–level expression time series. Model generated cell-level expression time series (red), average of model generated time series (black) and observed gene expression (green boxplots) for the mouse aorta (A: mouse fibroblast), B: mouse SMC) and drosophila clock neuron (C: dorsal neurons, D: lateral neurons) datasets.

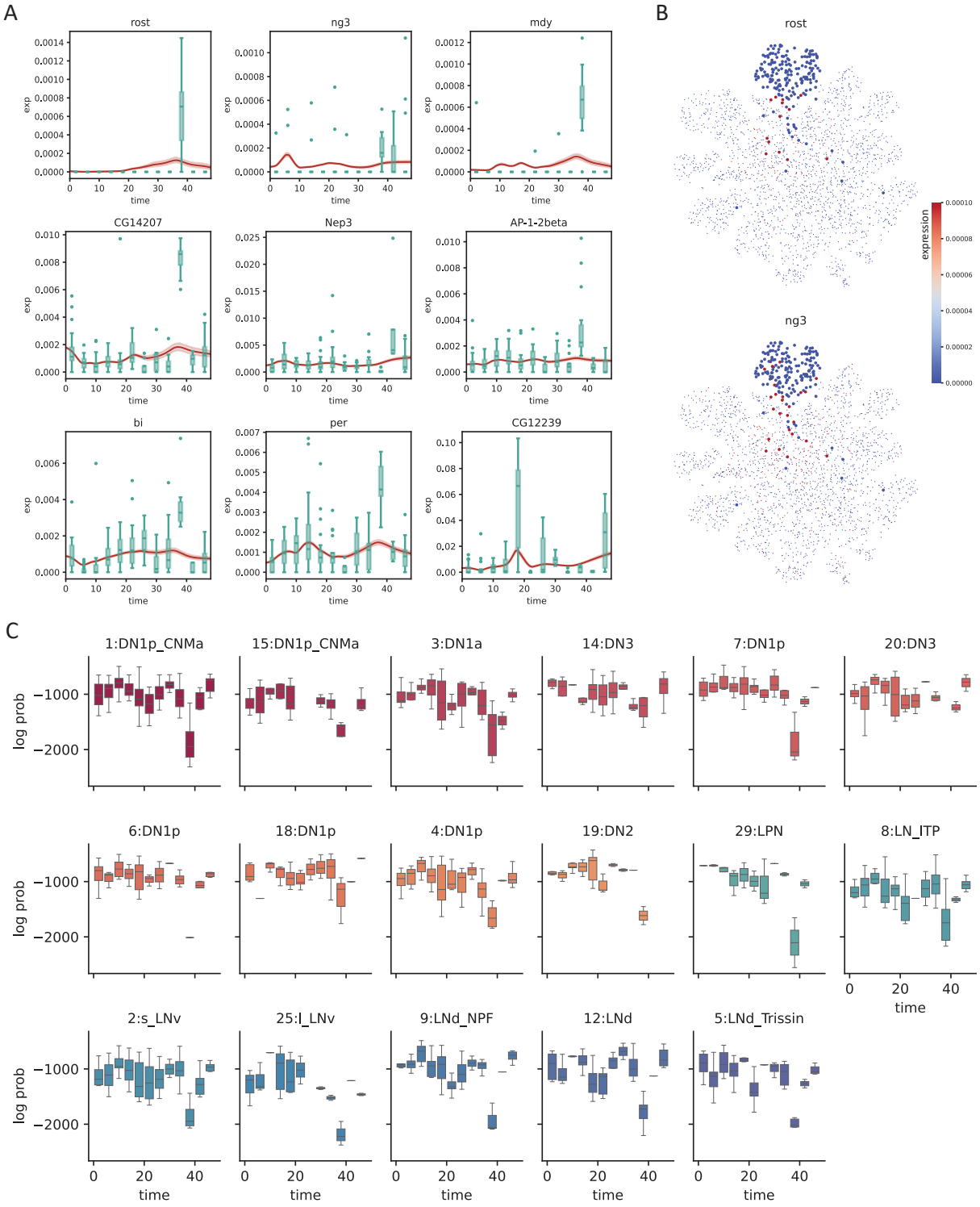

Figure 4: SNOW corrects potential batch effects. A: Examples showing SNOW–generated time series (average across cells shown as red lines, with shaded 95%CI) and the experimental observation (green boxplots). B: Gene expression of batch–affected genes overlayed on top of the UMAP projection of the clock neuron dataset. Cells belonging to the 1:DN1p_CNMa neuron cluster are plotted to be bigger. C: Box plots showing the log probability of observing each cell for the named clusters at each time–point.

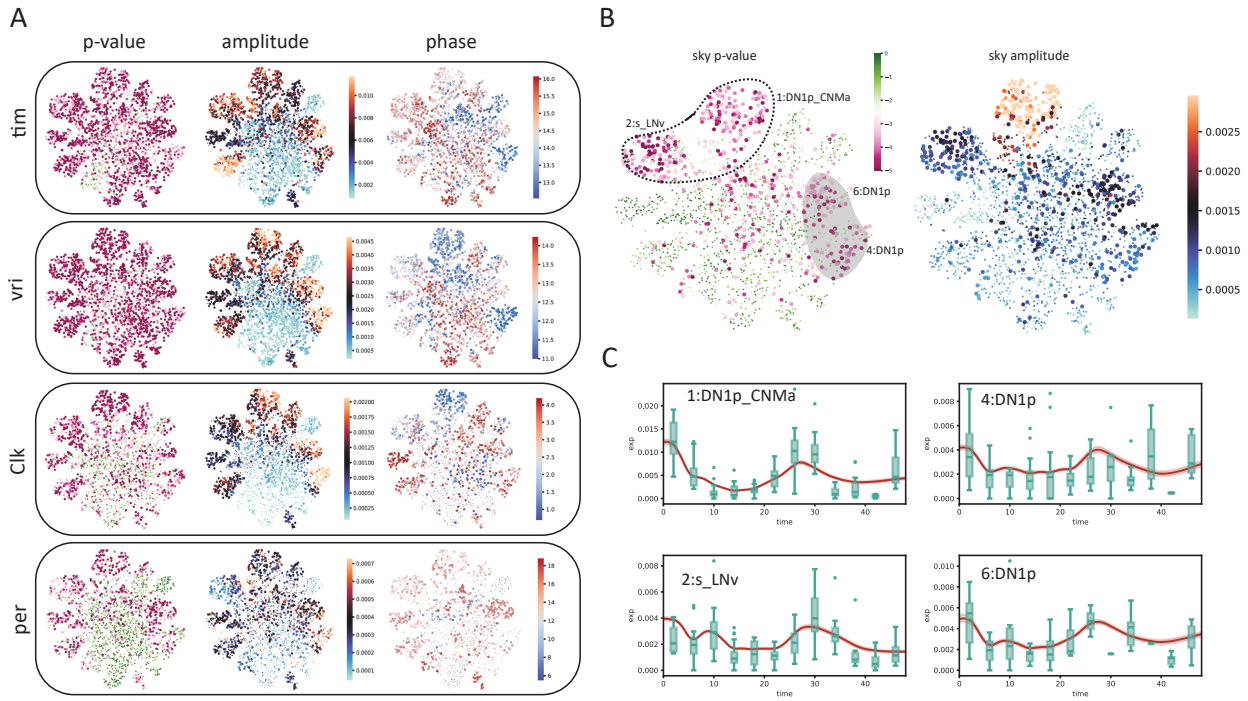

Figure 5: SNOW allows cycling detection at the single–cell level. A: Estimated p-values, amplitudes and phases (in hours) of known circadian genes (*tim*, *vri*, *Clk*, *per*) overlayed on the UMAP projection of the clock neuron dataset. Cells with $p > 0.001$ or amplitude smaller than 0.0001 were made small for better visualization. The p-value color scale is the same as panel B. B: Estimated p-values and amplitude of *sky*. The circled region indicates agreement between our analysis and that of Ma et al., and the shaded region indicates disagreements. C: SNOW–generated time series (average across cells shown as red lines with shaded 95%CI) and experimental observation (green boxes) of *sky* from cells within the circled and shaded region and panel B.

6 Acknowledgements

This work was supported by NSF grant DMS-1764421, Simons Foundation grant 597491, and NIH grant R01AG068579.

7 Code and Data Availability

Code for our analysis is available on bitbucket (https://bitbucket.org/biocomplexity/snow/src/main/).

References

[1] Stefan Peidli, Tessa D. Green, Ciyue Shen, Torsten Gross, Joseph Min, Samuele Garda, Bo Yuan, Linus J. Schumacher, Jake P. Taylor-King, Debora S. Marks, Augustin Luna, Nils Blüthgen, and Chris Sander. scPerturb: harmonized single-cell perturbation data. *Nature Methods*, January 2024.

[2] Alexandre F. Aissa, Abul B. M. M. K. Islam, Majd M. Ariss, Cammille C. Go, Alexandra E. Rader, Ryan D. Conrardy, Alexa M. Gajda, Carlota Rubio-Perez, Klara Valyi-Nagy, Mary Pasquinelli, Lawrence E. Feldman, Stefan J. Green, Nuria Lopez-Bigas, Maxim V. Frolov, and Elizaveta V. Benevolenskaya. Single-cell transcriptional changes associated with drug tolerance and response to combination therapies in cancer. *Nature Communications*, 12(1):1628, March 2021.

[3] Matthew T. Chang, Frances Shanahan, Thi Thu Thao Nguyen, Steven T. Staben, Lewis Gazzard, Sayumi Yamazoe, Ingrid E. Wertz, Robert Piskol, Yeqing Angela Yang, Zora Modrusan, Benjamin Haley, Marie Evangelista, Shiva Malek, Scott A. Foster, and Xin Ye. Identifying transcriptional programs underlying cancer drug response with TraCe-seq. *Nature Biotechnology*, 40(1):86–93, January 2022.

[4] Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P. Fulco, Livnat Jerby-Arnon, Nemanja D. Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, Britt Adamson, Thomas M. Norman, Eric S. Lander, Jonathan S. Weissman, Nir Friedman, and Aviv Regev. Perturb-Seq: Dissecting Molecular Circuits with Scalable Single-Cell RNA Profiling of Pooled Genetic Screens. *Cell*, 167(7):1853–1866.e17, December 2016.

[5] Britt Adamson, Thomas M. Norman, Marco Jost, Min Y. Cho, James K. Nuñez, Yuwen Chen, Jacqueline E. Villalta, Luke A. Gilbert, Max A. Horlbeck, Marco Y. Hein, Ryan A. Pak, Andrew N. Gray, Carol A. Gross, Atray Dixit, Oren Parnas, Aviv Regev, and Jonathan S. Weissman. A Multiplexed Single-Cell CRISPR Screening Platform Enables Systematic Dissection of the Unfolded Protein Response. *Cell*, 167(7):1867–1882.e21, December 2016.

[6] Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh

Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, Lia Lee, Jenny Chen, Justin Brumbaugh, Philippe Rigollet, Konrad Hochedlinger, Rudolf Jaenisch, Aviv Regev, and Eric S. Lander. Optimal-Transport Analysis of Single-Cell Gene Expression Identifies Developmental Trajectories in Reprogramming. *Cell*, 176(4):928–943.e22, February 2019.

[7] Aimée Bastidas-Ponce, Sophie Tritschler, Leander Dony, Katharina Scheibner, Marta Tarquis-Medina, Ciro Salinno, Silvia Schirge, Ingo Burtscher, Anika Böttcher, Fabian Theis, Heiko Lickert, and Mostafa Bakhti. Massive single-cell mRNA profiling reveals a detailed roadmap for pancreatic endocrinogenesis. *Development*, page dev.173849, January 2019.

[8] Anoop P. Patel, Itay Tirosh, John J. Trombetta, Alex K. Shalek, Shawn M. Gillespie, Hiroaki Wakimoto, Daniel P. Cahill, Brian V. Nahed, William T. Curry, Robert L. Martuza, David N. Louis, Orit Rozenblatt-Rosen, Mario L. Suvà, Aviv Regev, and Bradley E. Bernstein. Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science*, 344(6190):1396–1401, June 2014.

[9] Itay Tirosh, Benjamin Izar, Sanjay M. Prakadan, Marc H. Wadsworth, Daniel Treacy, John J. Trombetta, Asaf Rotem, Christopher Rodman, Christine Lian, George Murphy, Mohammad Fallahi-Sichani, Ken Dutton-Regester, Jia-Ren Lin, Ofir Cohen, Parin Shah, Diana Lu, Alex S. Genshaft, Travis K. Hughes, Carly G. K. Ziegler, Samuel W. Kazer, Aleth Gaillard, Kellie E. Kolb, Alexandra-Chloé Villani, Cory M. Johannessen, Aleksandr Y. Andreev, Eliezer M. Van Allen, Monica Bertagnolli, Peter K. Sorger, Ryan J. Sullivan, Keith T. Flaherty, Dennie T. Frederick, Judit Jané-Valbuena, Charles H. Yoon, Orit Rozenblatt-Rosen, Alex K. Shalek, Aviv Regev, and Levi A. Garraway. Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science*, 352(6282):189–196, April 2016.

[10] Dingbang Ma, Dariusz Przybylski, Katharine C Abruzzi, Matthias Schlichting, Qunlong Li, Xi Long, and Michael Rosbash. A transcriptomic taxonomy of Drosophila circadian neurons around the clock. *eLife*, 10:e63056, January 2021.

[11] Shao'ang Wen, Danyi Ma, Meng Zhao, Lucheng Xie, Qingqin Wu, Lingfeng Gou, Chuanzhen Zhu, Yuqi Fan, Haifang Wang, and Jun Yan. Spatiotemporal single-cell analysis of gene ex-

pression in the mouse suprachiasmatic nucleus. *Nature Neuroscience*, 23(3):456–467, March 2020.

[12] Diego Calderon, Ronnie Blecher-Gonen, Xingfan Huang, Stefano Secchia, James Kentro, Riza M. Daza, Beth Martin, Alessandro Dulja, Christoph Schaub, Cole Trapnell, Erica Larschan, Kate M. O'Connor-Giles, Eileen E. M. Furlong, and Jay Shendure. The continuum of *Drosophila* embryonic development at single-cell resolution. *Science*, 377(6606):eabn5800, August 2022.

[13] Daniela J. Di Bella, Ehsan Habibi, Robert R. Stickels, Gabriele Scalia, Juliana Brown, Payman Yadollahpour, Sung Min Yang, Catherine Abbate, Tommaso Biancalani, Evan Z. Macosko, Fei Chen, Aviv Regev, and Paola Arlotta. Molecular logic of cellular diversification in the mouse cerebral cortex. *Nature*, 595(7868):554–559, July 2021.

[14] Jeffrey A. Farrell, Yiqun Wang, Samantha J. Riesenfeld, Karthik Shekhar, Aviv Regev, and Alexander F. Schier. Single-cell reconstruction of developmental trajectories during zebrafish embryogenesis. *Science*, 360(6392):eaar3131, June 2018.

[15] Grace Hui Ting Yeo, Sachit D. Saksena, and David K. Gifford. Generative modeling of single-cell time series with PRESCIENT enables prediction of cell trajectories with interventions. *Nature Communications*, 12(1):3222, May 2021.

[16] Yiting Tsai, Qiugang Lu, Lee Rippon, Siang Lim, Aditya Tulsyan, and Bhushan Gopaluni. Pattern and Knowledge Extraction using Process Data Analytics: A Tutorial. *IFAC-PapersOnLine*, 51(18):13–18, 2018.

[17] Jiaqi Zhang, Erica Larschan, Jeremy Bigness, and Ritambhara Singh. scNODE : Generative Model for Temporal Single Cell Transcriptomic Data Prediction. preprint, Bioinformatics, November 2023.

[18] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. 2013. Publisher: arXiv Version Number: 11.

[19] Yutong Sha, Yuchi Qiu, Peijie Zhou, and Qing Nie. Reconstructing growth and dynamic

trajectories from single-cell transcriptomics data. *Nature Machine Intelligence*, 6(1):25–39, November 2023.

[20] Alexander Tong, Jessie Huang, Guy Wolf, David van Dijk, and Smita Krishnaswamy. TrajectoryNet: A Dynamic Optimal Transport Network for Modeling Cellular Dynamics. 2020. Publisher: arXiv Version Number: 2.

[21] Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, January 2000.

[22] Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology*, 36(5):411–420, May 2018.

[23] Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with Harmony. *Nature Methods*, 16(12):1289–1296, December 2019.

[24] Xiaokang Yu, Xinyi Xu, Jingxiao Zhang, and Xiangjie Li. Batch alignment of single-cell transcriptomics data using deep metric learning. *Nature Communications*, 14(1):960, February 2023.

[25] Lu Qin, Guangya Zhang, Shaoqiang Zhang, and Yong Chen. Deep Batch Integration and Denoise of Single-Cell RNA-Seq Data. *Advanced Science*, 11(29):2308934, August 2024.

[26] Yitong Huang and Rosemary Braun. Platform-independent estimation of human physiological time from single blood samples. *Proceedings of the National Academy of Sciences*, 121(3):e2308114120, January 2024.

[27] Rosemary Braun, William L. Kath, Marta Iwanaszko, Elzbieta Kula-Eversole, Sabra M. Abbott, Kathryn J. Reid, Phyllis C. Zee, and Ravi Allada. Universal method for robust detection of circadian state from gene expression. *Proceedings of the National Academy of Sciences*, 115(39), September 2018.

[28] Ron C. Anafi, Lauren J. Francey, John B. Hogenesch, and Junhyong Kim. CYCLOPS reveals human transcriptional rhythms in health and disease. *Proceedings of the National Academy of Sciences*, 114(20):5312–5317, May 2017.

[29] Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. 2018. Publisher: arXiv Version Number: 3.

[30] Jonathan Karin, Yonathan Bornfeld, and Mor Nitzan. scPrisma infers, filters and enhances topological signals in single-cell data using spectral template matching. *Nature Biotechnology*, 41(11):1645–1654, November 2023.

[31] Romain Lopez, Jeffrey Regier, Michael B. Cole, Michael I. Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058, December 2018.

[32] Benjamin J. Auerbach, Garret A. FitzGerald, and Mingyao Li. Tempo: an unsupervised Bayesian algorithm for circadian phase inference in single-cell transcriptomics. *Nature Communications*, 13(1):6580, November 2022.

[33] Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M. Mauck, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive Integration of Single-Cell Data. *Cell*, 177(7):1888–1902.e21, June 2019.

[34] Maximilian Strunz, Lukas M. Simon, Meshal Ansari, Jaymin J. Kathiriya, Ilias Angelidis, Christoph H. Mayr, George Tsidiridis, Marius Lange, Laura F. Mattner, Min Yee, Paulina Ogar, Arunima Sengupta, Igor Kukhtevich, Robert Schneider, Zhongming Zhao, Carola Voss, Tobias Stoeger, Jens H. L. Neumann, Anne Hilgendorff, Jürgen Behr, Michael O'Reilly, Mareike Lehmann, Gerald Burgstaller, Melanie Königshoff, Harold A. Chapman, Fabian J. Theis, and Herbert B. Schiller. Alveolar regeneration through a Krt8+ transitional stem cell state that persists in human lung fibrosis. *Nature Communications*, 11(1):3559, July 2020.

[35] Cole Trapnell, Davide Cacchiarelli, Jonna Grimsby, Prapti Pokharel, Shuqiang Li, Michael Morse, Niall J Lennon, Kenneth J Livak, Tarjei S Mikkelsen, and John L Rinn. The dynamics

and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology*, 32(4):381–386, April 2014.

[36] Xiaojie Qiu, Qi Mao, Ying Tang, Li Wang, Raghav Chawla, Hannah A Pliner, and Cole Trapnell. Reversed graph embedding resolves complex single-cell trajectories. *Nature Methods*, 14(10):979–982, October 2017.

[37] Junyue Cao, Malte Spielmann, Xiaojie Qiu, Xingfan Huang, Daniel M. Ibrahim, Andrew J. Hill, Fan Zhang, Stefan Mundlos, Lena Christiansen, Frank J. Steemers, Cole Trapnell, and Jay Shendure. The single-cell transcriptional landscape of mammalian organogenesis. *Nature*, 566(7745):496–502, February 2019.

[38] Ray Zhang, Nicholas F. Lahens, Heather I. Ballance, Michael E. Hughes, and John B. Hogenesch. A circadian gene expression atlas in mammals: Implications for biology and medicine. *Proceedings of the National Academy of Sciences*, 111(45):16219–16224, November 2014.

[39] Paul F. Thaben and Pål O. Westermark. Detecting Rhythms in Time Series with RAIN. *Journal of Biological Rhythms*, 29(6):391–400, December 2014.

[40] Michael E. Hughes, John B. Hogenesch, and Karl Kornacker. JTK_cycle: An Efficient Nonparametric Algorithm for Detecting Rhythmic Components in Genome-Scale Data Sets. *Journal of Biological Rhythms*, 25(5):372–380, October 2010.

[41] Bingxian Xu, Dingbang Ma, Katharine Abruzzi, and Rosemary Braun. Detecting rhythmic gene expression in single-cell transcriptomics. *Journal of Biological Rhythms*, 39(6):581–593, 2024.

[42] Gökcen Eraslan, Lukas M. Simon, Maria Mircea, Nikola S. Mueller, and Fabian J. Theis. Single-cell RNA-seq denoising using a deep count autoencoder. *Nature Communications*, 10(1):390, January 2019.

[43] Jiarui Ding, Anne Condon, and Sohrab P. Shah. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nature Communications*, 9(1):2002, May 2018.

[44] Pierre Boyeau, Justin Hong, Adam Gayoso, Martin Kim, José L. McFaline-Figueroa, Michael I. Jordan, Elham Azizi, Can Ergen, and Nir Yosef. Deep generative modeling of sample-level heterogeneity in single-cell genomics, October 2022.

[45] Haoxiang Gao, Kui Hua, Xinze Wu, Lei Wei, Sijie Chen, Qijin Yin, Rui Jiang, and Xuegong Zhang. Building a learnable universal coordinate system for single-cell atlas with a joint-VAE model. *Communications Biology*, 7(1):977, August 2024.

[46] Kai Cao, Qiyu Gong, Yiguang Hong, and Lin Wan. A unified computational framework for single-cell data integration with optimal transport. *Nature Communications*, 13(1):7419, December 2022.

[47] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2016.

[48] Hyunjik Kim and Andriy Mnih. Disentangling by Factorising, July 2019. arXiv:1802.05983 [cs, stat].

[49] Shuyang Gao, Rob Brekelmans, Greg Ver Steeg, and Aram Galstyan. Auto-Encoding Total Correlation Explanation, February 2018. arXiv:1802.05822 [cs, stat].

[50] Ricky T. Q. Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating Sources of Disentanglement in Variational Autoencoders, April 2019. arXiv:1802.04942 [cs, stat].

[51] Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, N. Siddharth, Brooks Paige, Dana H. Brooks, Jennifer Dy, and Jan-Willem van de Meent. Structured Disentangled Representations, December 2018. arXiv:1804.02086 [cs, stat].

[52] Ishan Deshpande, Ziyu Zhang, and Alexander Schwing. Generative Modeling using the Sliced Wasserstein Distance. 2018. Publisher: arXiv Version Number: 1.

[53] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017. arXiv:1412.6980 [cs].

[54] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library, December 2019. arXiv:1912.01703 [cs, stat].