Approximating a linear dynamical system from non-sequential data

Cliff Stein, Pratik Worah

Columbia University and Google Research, NYU and Google Research

Abstract. Given non-sequential snapshots from instances of a dynamical system, we design a compressed sensing based algorithm that reconstructs the dynamical system. We formally prove that successful reconstruction is possible under the assumption that we can construct an approximate clock from a subset of the coordinates of the underlying system.
As an application, we show that our assumption is likely true for genomic datasets, and we recover the underlying nuclear receptor networks and predict pathways, as opposed to genes, that may differentiate phenotypes in some publicly available datasets.

1 Introduction

The problem of reconstructing a noisy linear dynamical system from sequential observations to predict its next step is the basic problem in filtering theory. Here we consider the case when the order of observations has been lost, i.e., they are not necessarily sequential. Such a situation arises often in biological samples, where although one can collect information about cells it is impossible to know perfectly where the cells are in their cell cycle relative to each other. Another area where this problem arises is in reconstructing a linear system learnt over noisy networks, for example in a noisy financial feed where the packets may be disordered or dropped.

In this paper, we formulate a concrete version of the above reconstruction problem and solve it under the assumption that we can construct an approximate clock from the coordinates of the underlying dynamical system. Algorithm 1 in Section 3 is our main theoretical contribution and Theorem 2 in Section 4 upper-bounds the error of the recovered dynamical system using Algorithm 1. The compressed sensing methods used in our paper ensure that reconstruction can be successful even with small sample sizes under sparsity assumptions.

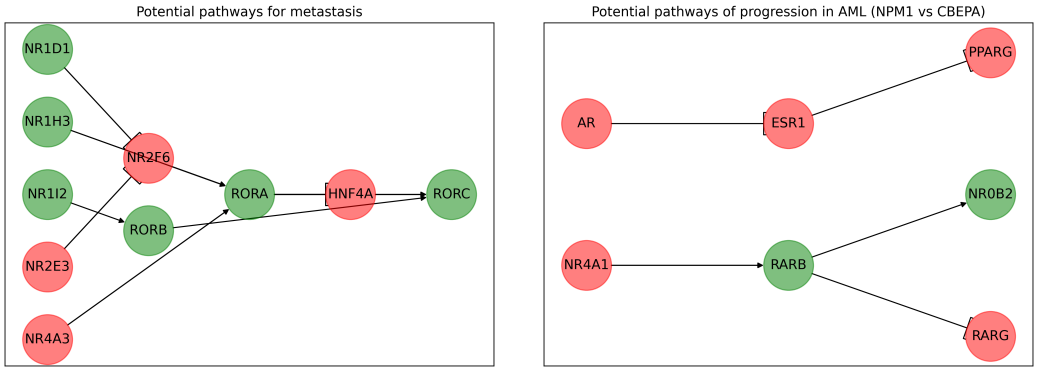
We also show an application of our algorithm to genomic datasets in Section 5. These datasets satisfy both our assumptions: (1) sparsity: since any gene expression *directly* relies on only a small number of genes, and (2) admitting an approximate clock: since cyclin expressions can be used to recover the approximate order in the cell cycle for a given sample (see Figure 3). Moreover, we can potentially use the reconstructed linear dynamical systems to compute genomic pathways that are prominently different between pathological and reference biological samples (see Figure 1). Thus, unlike standard tools (like [LHA14]) which predict genes whose expressions may differentiate between two phenotypes, we can predict pathways that differentiate between two phenotypes.

2 Background

Perhaps the algorithmic line of work closest to our abstract problem is the Kalman filtering problem in the presence of network loss and delays (see [LXL21], [LWLA20] and [NE19], and the references therein). However, in such results they typically assume a independent identical probability of packet drop or delay, or a specific Markov process of interest is chosen to model that. Also, although the idea of computing conditioned Gaussians (for the error upper-bound in Theorem 2) is similar to that used for the analysis of the Kalman filter, the problem studied in our paper differs from the standard filtering problem: (1) we are interested in recovering the underlying linear system and not in predicting the next step of the system, and (2) we are interesting in applications with sparsity (note the connection with compressed sensing).

For our genomic application, we have used the publicly available datasets [Can12] and [ea13], available at github as well as the cbio portal.[1] The single cell RNA-seq dataset from [BPSTS+19] was

[1] https://www.cbioportal.org



(a) Potential pathways for metastasis. Based on this figure, one prominent pathway could be: NR1I2 up-regulates RORB which up-regulates RORC, whose increased expression is positively correlated with metastasis; based on data in [Can12].

(b) Potential pathways differentiating NPM1 vs CBEPA mutation in AML. Based on this figure, one prominent pathway could be: AR down-regulates ESR1 which down-regulates PPARG, whose decreased expression is negatively correlated with NPM1 mutation; based on data in [ea13].

Fig. 1: Predicted pathways where gene A $\rightarrow$ gene B means that A (indirectly) up-regulates B, while the flat arrowhead indicates (indirect) down-regulation. A green node indicates that the corresponding gene expression positively correlates with the pathological phenotype (metastasis or NPM1 mutation), while a red node indicates positive correlation with the reference phenotype.

used to compare the approximate cyclin clock with latent time [BLP+20], in order to sanity check the basic idea. Note that the problem of trajectory inference, which is a harder problem than our problem of merely recovering the covariance matrix, has been well explored in single cell RNA-seq literature [Sae19]. Finally, it is possible that cyclin expressions themselves may be effected in many situations by the pathology under consideration (for example [GFKH+22]), so while this exact idea may not work in many situations, the basic framework nonetheless should hold.

3 Technical overview

We consider the following concrete problem. Suppose we have a n-dimensional Ornstein-Uhlenbeck process (see for example [Oks13])

$$dx_t = Bx_t dt + AdW_t, \tag{1}$$

where W_t is n-dimensional standard Brownian motion and the *diffusivity* matrix A is not assumed to be known. Note that the diffusivity matrix is the covariance matrix of the increments of x_t and thus requires ordering information for its computation. Suppose that we observe x_t at various time points in $[0, T]$ for large T, but we either do not have the ordering information or do not keep the samples ordered. Then, we want to know whether we can recover the underlying (unknown) $n \times n$ *drift* matrix B, that characterizes our dynamical system.[2]

In general, the answer is negative. However, the main technical result in this paper is to show that if there exists an "approximate clock" and B is sparse, then we actually can approximately recover B. More precisely, when we say that there exists an *approximate clock*, we mean that we know that one or more of the coordinates of x_t, or their linear combination, have positive drift and a relatively smaller diffusivity coefficient, i.e., say that coordinate is denoted $\tau(t)$ and it is modeled by the diffusion:

$$d\tau(t) = \delta dt + \varepsilon dW'_t, \tag{2}$$

where W'_t is standard Brownian motion and δ is positive and much greater than ε^2. Assuming the existence of such a clock, our *main algorithmic contribution* is Algorithm 1, that can recover B from

[2] We have assumed the system is centered about 0 but that is not an issue since the long term average can be computed without knowing the ordering and subtracted from the observations.

the samples under the assumption that B is row sparse, i.e., each row of B has only $s = \log^{O(1)}(n)$ non-zero entries and the number of samples, i.e., t, is larger than $s\log(n)$.

For our algorithm, we do not have a given ϵ and δ as in Equation 2. Rather, we will use the approximate clock to recover an approximation of the diffusivity matrix A, which we denote by $\tilde{A}$. We don't actually need $\tilde{A}$ but rather $\tilde{A}\tilde{A}^T$, and, in Theorem 2 in Section 4 we show that ordering the x_i values using $\tau(t)$ and computing the covariance from the increments can closely approximate the covariance of the increments computed using the ordering based on the actual time.

Algorithm 1 runs in polynomial time as the main steps are computing a covariance matrix and solving the convex optimization problem in Step 8, which is a second order cone program. This convex optimization is a slight variation of basis pursuit in compressed sensing literature (see for example [FR13]) and therefore unique reconstruction is guaranteed under sparsity assumptions. In particular the following uniqueness theorem holds true for the matrix $\tilde{B}$ recovered by Algorithm 1.

Theorem 1. *[DLR18] Let s denote the row sparsity of B, i.e., each row has at most s non-zero coordinates, then for $m = \Omega(\log n)$ large and the sample times for $\{x_{t_1}, ..., x_{t_m}\}$ well separated from each other, then we have with probability $1 - o(1)$ (for large n): $\|\tilde{B} - B\|_1 = 0$, i.e., B can be recovered uniquely. If however, the rows are not exactly s sparse and the ℓ_1 norm of the $n - s$ remaining entries in any row is at most ϵ, then we have: $\|\tilde{B} - B\|_1 \leq \epsilon n$.*

The proof is fairly standard and follows from existing compressed sensing literature (see [FR13] or [DLR18]), so we don't restate it.

Finally, we note that the constraint in Step 7 uses the properties of Ornstein-Uhlenbeck process to recover the drift matrix B from the covariance matrix Σ and the (approximate) covariance matrix of the increments $\tilde{A}\tilde{A}^T$ up to a scalar multiple (that scalar is just the average drift of the approximate clock). Recovery up to a scalar multiple suffices because we are often interested in how various coordinates of x interact relative to each other, as opposed to knowing the exact coefficients.

Algorithm 1 Recover dynamical system

1: Input: Unordered snapshots $\{x_1, ..., x_t \in \mathbb{R}^n; t \in [0, T]\}$ derived from an Ornstein-Uhlenbeck process $dx_t = Bx_t dt + AdW_t$; where the diffusivity matrix A is known up to a constant multiple error as $\tilde{A}$, and B is row sparse but unknown.
2: Output: A recovered matrix $\tilde{B}$ that is close (in ℓ_1 norm) to a scalar multiple of B.
3: ▷ Algorithm starts:
4: Compute the $n \times n$ covariance matrix of the random vectors $\{x_1, ..., x_t\}$, denoted Σ
5: ▷ Note: computing Σ does not require ordering the x_is.
6: ▷ Below, B_i denotes the i^{th} row of B
7: Solve $\tilde{B} := \arg\min_B \sum_{i=0}^{n} \|B_i\|_2 \quad$ s.t. $\Sigma B + B^T\Sigma = -\frac{\tilde{A}\tilde{A}^T}{2}$
8: return $\tilde{B}$

Often it is not enough to recover the drift matrix, but we need to distinguish one linear system from another, where the first is a small perturbation of the second, and one has only few samples of the second at hand. This exact case happens for genomic data samples, where one system is the reference and many samples are available, but another system is some rare genetic condition or disease for which fewer samples are available. Algorithm 2 obtains the *perturbation matrix* P (upto a small error) that distinguishes the second linear system from the first. Note that the algorithm always returns a value for a large enough choice of the noise parameter η and small enough choice of ϵ in Step 8, since that will suffice to make the convex program feasible.

3.1 Application to genomics

As a concrete application, we use the recovered perturbation matrix $\tilde{P}$ to isolate the paths consisting entirely of high weight edges in the directed graph underlying the dynamical system, since these paths should reflect the prominent genomic pathways that differentiate the reference dynamical system from the pathological dynamic system. We note that more standard methods like time series analysis based on Fourier transform of the recovered linear system can also be tried out, but our method below is far

Algorithm 2 Order and recover perturbations between two dynamical systems

1: Input: Unordered snapshots $\{x_1, ..., x_t \in \mathbb{R}^n; t \in [0, T]\}$ and $\{x'_1, ..., x'_{t'} \in \mathbb{R}^n; t \in [0, T']\}$ derived from two Ornstein-Uhlenbeck processes $dx_t = Bx_t dt + AdW_t$ and $dx'_t = B'x_t dt + A'dW_t$; where the diffusivity matrices A and A' are known up to small error, but B and B' are unknown. Moreover, $t' \ll t$ and B and B' are row sparse.
2: Output: A recovered matrix $\tilde{B}$ that is close to B in ℓ_1 norm, and a perturbation matrix $\tilde{P}$ such that $\tilde{B} + \epsilon\tilde{P} \simeq B'$, for a small positive parameter ε.
3: ▷ Algorithm starts:
4: Use Algorithm 1 to compute $\tilde{B}$
5: ▷ The smaller number of samples prevents direct computation of $\tilde{B}'$, so we compute a first order approximation.
6: Compute the $n \times n$ covariance matrix of the random vectors $\{x'_1, ..., x'_{t'}\}$, denoted Σ'
7: ▷ Choose a small noise η so that the following convex program program is feasible
8: Solve $\tilde{P} := \arg\min_P \sum_{i,j=0}^{n} |P_{ij}|$ s.t. $\|(\Sigma - \Sigma')\tilde{B}^T + \tilde{B}(\Sigma - \Sigma') - \epsilon(\Sigma' P^T + P\Sigma')\|_2 \leq \eta$
9: return $\tilde{P}$

simpler, and thus probably more robust to recovery errors, and it suffices to illustrate the pathways that can be recovered.

Intuitively, the matrix P can be thought of as a directed graph with entry P_{ij} reflecting a weighted directed edge from node corresponding to gene j to that for gene i. A *prominent* path is defined to be one such that all its edges have weight higher than some fixed threshold, say θ. For such a prominent path p to potentially reflect an actual underlying genetic pathway that promotes the pathological phenotype, one of the following intuitive condition should hold:

1. If the gene corresponding to the terminal node of p positively correlates with the pathological phenotype then it should be up-regulated by the gene preceding it in the path p; moreover, the same property or property (2) should now hold for the subpath terminating in the penultimate gene, and so on.
2. Otherwise, if the gene corresponding to the terminal node of p negatively correlates with the pathological phenotype then it should be down-regulated by the gene preceding it in the path p; moreover, the same property or property (1) should now hold for the subpath terminating in the penultimate gene, and so on.

This intuition leads to the following algorithm (Algorithm 3) for predicting prominent genomic pathways that may be experimentally verified to confirm that some or all of them lead to the pathological phenotype.

Algorithm 3 Recover Pathways

1: Input: The underlying linear dynamical systems matrices $\tilde{B}$ and $\tilde{B}'$ and the correlations between each of the coordinates (corresponding to genes) and the two phenotypes of interest.
2: Output: A set of pathways of a given length L that are prominently different between the two phenotypes.
3: ▷ Algorithm starts:
4: Compute and sort the list of genes in descending order in terms of the absolute value of their correlation coefficient with the pathological phenotype, denote this list as $\boldsymbol{g}$
5: Compute $C := \text{Diag}(\boldsymbol{g})(\tilde{B}' - \tilde{B})$
6: Fix a positive threshold θ and set $C_{ij} = 0$ if $C_{ij} < \theta$, denote the resulting matrix as Π_θ
7: Compute the set of paths of length L in the graph with adjacency matrix Π_θ and return them.

The results of running Algorithm 3 on the BRCA [Can12] and AML [ea13] datasets to isolate critical differentiating pathways are described in Figure 1. In this figure, we show predicted pathways where gene A $\rightarrow$ gene B means that A (indirectly) up-regulates B, while the flat arrowhead indicates (indirect) down-regulation. A green node indicates that the corresponding gene expression positively correlates with the pathological phenotype (metastasis or NPM1 mutation), while a red node indicates positive correlation with the reference phenotype. In part a, we show potential pathways for metastasis. Based on this figure, one prominent pathway could be: NR1I2 up-regulates RORB which up-regulates

RORC. Data presented in [Can12] shows that the expression of RORC is positively correlated with metastasis, and thus this pathway is a plausible explanation. In part b, we show potential pathways differentiating NPM1 vs CBEPA mutation in AML. Based on this figure, one prominent pathway could be: AR down-regulates ESR1 which down-regulates PPARG. Data in [ea13] shows that a decreased expression of PPARG is negatively correlated with NPM1 mutation.

In the rest of the paper we give and prove our main theorem in Section 4 and then give more details on how we obtain the results in Figure 1 in Section 5.

4 Theoretical bound on error from approximate clock

In this section, we derive an upper-bound on the error of the recovered diffusivity matrix when we use an approximate clock. We will assume that our dynamical system can be modeled by an Ornstein-Uhlenbeck process and it has a coordinate, or a linear combination of coordinates, that admit a positive drift which is larger than the diffusivity. This coordinate will act as an approximate clock. Theorem 2 shows that if we use this approximate clock then the recovered diffusivity is not too different from the diffusivity that would be obtained if we knew the ordering in the dynamical system.

Theorem 2. *Assuming that the clock coordinate $\tau(t)$ is independent of the remaining coordinates of x_t, the diffusivity matrix $\tilde{A}\tilde{A}^T$ computed using the approximate clock $\tau(t)$ is a constant multiple of the actual diffusivity, i.e.,*

$$\tilde{A}\tilde{A}^T = \lim_{h\to 0} \mathbb{E}\left[\frac{1}{T}\int_0^T \langle (x(\tau(t+h)) - x(\tau(t))), (x(\tau(t+h)) - x(\tau(t)))\rangle d\tau(t)\right] \simeq \delta A A^T, \tag{3}$$

for $\varepsilon \ll \delta$, which is the drift of the clock.[3]

Proof. Note that

$$\tilde{A}\tilde{A}^T = \lim_{h\to 0} \mathbb{E}\left[\frac{1}{T}\int_0^T \langle (x(\tau(t+h)) - x(\tau(t))), (x(\tau(t+h)) - x(\tau(t)))\rangle d\tau(t)\right] \tag{4}$$

by definition of x_t. By the tower property of conditional expectation and switching the order of integrals:

$$\mathbb{E}\left[\frac{1}{T}\int_0^T \langle dx(\tau(t)), dx(\tau(t))\rangle\right] = \mathbb{E}\left[\frac{1}{T}\int_0^T \mathbb{E}\left[\langle dx(\tau(t)), dx(\tau(t))\rangle \middle| \tau(t)\right]\right] \tag{5}$$

Recall that,

$$dx(t) = Bx(t)dt + AdW_t \tag{6}$$
$$d\tau(t) = \delta dt + \varepsilon dW'_t, \tag{7}$$

where W'_t and W_t are independent standard Brownian motions in $\mathbb{R}$ and $\mathbb{R}^n$ respectively. The inner conditional expectation can be evaluated in terms of $\tau(t)$ as follows:

$$\mathbb{E}\left[\langle dx(\tau(t)), dx(\tau(t))\rangle \middle| \tau(t)\right] = \mathbb{E}\left[\langle Bx(\tau)d\tau + AdW_\tau, Bx(\tau)d\tau + AdW_\tau\rangle \middle| \tau(t)\right]. \tag{8}$$

Note that we are conditioning a Gaussian with another Gaussian, so the result is a Gaussian variable. The quadratic variation term in the RHS expectation is a sum of three types of terms that can be calculated using:

$$\mathbb{E}\left[\langle AdW_\tau, AdW_\tau\rangle\right] = AA^T d\tau(t) \tag{9}$$
$$\mathbb{E}\left[\langle Bx(\tau)d\tau, Bx(\tau)d\tau\rangle\right] = \varepsilon^2 Bx(\tau)x(\tau)^T B^T dt \tag{10}$$
$$\mathbb{E}\left[\langle AdW_\tau, Bx(\tau)d\tau\rangle\right] = 0. \tag{11}$$

[3] Note that one can skirt around the issue of negative values of τ by assuming that $\tau(0)$ is large, that will ensure τ remains positive with high probability.

The last equality follows because we have assumed that W'_t and W_t are independent. However, that correlation can be calculated explicitly as well, if needed.

Therefore,

$$\int_0^T \mathbb{E}\left[\langle dx(\tau(t)), dx(\tau(t))\rangle \middle| \tau(t)\right] = \int_0^T \left(AA^T d\tau(t) + \varepsilon^2 Bx(\tau)x(\tau)^T B^T dt\right) \tag{12}$$

Note that

$$\mathbb{E}\left[\frac{1}{T}\int_0^T AA^T d\tau(t)\right] = \mathbb{E}\left[\frac{1}{T}\int_0^T \delta AA^T dt\right], \tag{13}$$

since the integral with dW'_t is a martingale with zero mean. Therefore, if $\varepsilon \ll \delta$ then

$$\mathbb{E}\left[\frac{1}{T}\int_0^T \langle dx(\tau(t)), dx(\tau(t))\rangle\right] \simeq \frac{\delta}{T}\int_0^T AA^T dt. \tag{14}$$

□

If the correlations between the randomness in $\tau(t)$ and other coordinates is significant then we can evaluate Equation 11 as follows.

Lemma 1. *Let $\rho \in \mathbb{R}^{1\times n}$ be the covariance matrix between W'_t and W_t, which are the Brownian motions in the definitions of $\tau(t)$ and x_t respectively; then the quadratic variation below evaluates as:*

$$\mathbb{E}\left[\langle AdW_\tau, Bx(\tau)d\tau\rangle\right] = \varepsilon A\rho x(t)^T B^T dt \tag{15}$$

The proof follows immediately from the definition quadratic variation. Therefore, following a similar proof as Theorem 2, the error to $\tilde{A}\tilde{A}^T$ from Theorem 2 is changed by an additive factor that is $O(\varepsilon \max\{\rho\})$. So when all coordinates of $\varepsilon\rho$ are small, the characterization of $\tilde{A}\tilde{A}^T$ in Theorem 2 continues to hold.

5 Application: RNA-seq data

In this section, we describe an application of our algorithms on publicly available genomic datasets. While our main result is in Figure 1 which is the output of Algorithm 3, where the latter predicts potential nuclear receptor pathways that differentiate two clusters of genomic data; we do need to compute the drift matrix for that and also verify in one setting that the approximate cyclin expression based clock can be used for our purposes. In this section, we illustrate the intermediate outputs for both these problems.

Although our example applications are based on RNA-seq data from [Can12] and [ea13] and use only a small set of nuclear receptors for computational tractability reasons, the output illustrates our basic approach well. [4]

5.1 Recovering dynamical systems from RNA-seq datasets

Recall that, often biological data-samples do not come with timestamps, for example, RNA-seq data can be modeled to be derived from a Markov process, but it is often impossible to say, with perfect accuracy, whether a sequenced cell precedes or succeeds another sequenced cell in cell cycle time. For single cell data this has led to a substantial number of algorithms and literature that tries to order cells by their latent time [BLP+20] or some equivalent of it, and those results are surveyed in [Sae19].

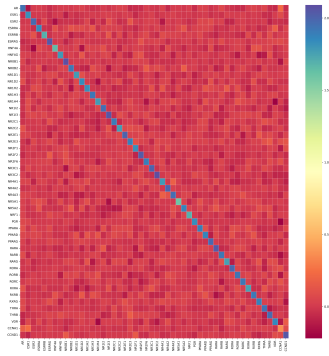
We used Algorithms 1 and 2 to compute the underlying linear dynamical system matrix: $\tilde{B}$ and the perturbations: $\tilde{P}$, and our results are in the Figure 2. They were obtained using RNA-seq data in publicly available data-sets [Can12] and [ea13]. Since the data-sets are unordered we use a linear combination of Cyclin A and D expression levels, as our approximate clock, to order the data-points by cell age. Essentially, if we were to obtain a random sample of two cells from a tissue and found the difference, i.e., cyclin A - cyclin D levels, was relatively higher for sample 1 over sample 2 than we

[4] The python code for this section is available at: https://github.com/cliffstein/recomb_pathways

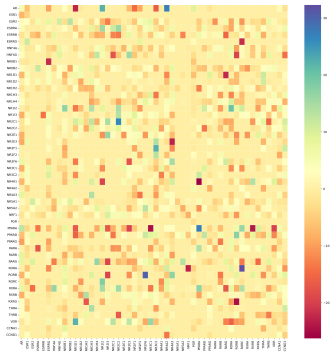
know that sample 1 was more advanced in the cell cycle than sample 2 (with some error). Thus we can use their difference in their expression as our approximate clock. Note that we can't verify how good this clock is, since we don't have access to any ground truth clock for [Can12] and [ea13] data-sets that orders cells by their relative cell cycle stage. However, in the next subsection, we do compare scVelo latent time against this kind of an approximate cyclin clock in murine pancreatic tissue using data from [BPSTS+19].

Once we have the approximate clock using cyclin expressions, we can order the cell samples to derive the covariance matrix of increments $\tilde{A}\tilde{A}^T$. Finally, we model the logarithmic RNA-seq expression levels in the data-sets [Can12] and [ea13] as an Ornsetin-Uhlenbeck process of the form in Equation 2, and use Algorithm 2 to derive the matrices $\tilde{B}$ and $\tilde{P}$ in Figure 2.

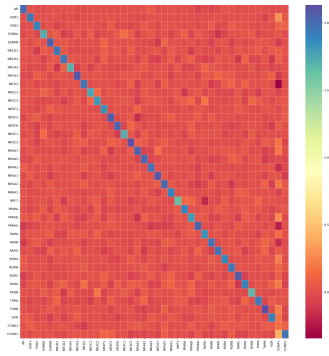
Given the ability of linear dynamical systems to capture fairly complicated dynamics such a model should be general enough to capture many of the intricacies of gene expression pathways.



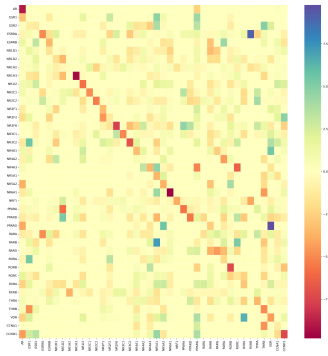
(a) Underlying dynamical system for non-metastatic cases



(b) Perturbations towards metastasis



(c) Underlying dynamical system for CBEPA mutation



(d) Perturbations towards NPM1 mutation

Fig. 2: Recovered dynamical system and perturbation for breast cancer and AML datasets.

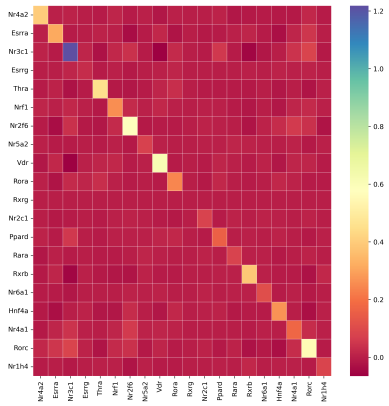
5.2 The cyclin clock

In this subsection, we verify that the error in the computation of the diffusivity matrix is indeed small by computing it in two ways using the scRNA-seq data from [BPSTS+19]:

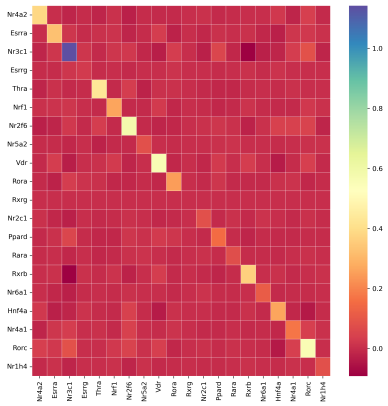
1. First, we compute the diffusivity matrix $\tilde{A}$ using an approximate clock that equals the difference in expression level between two cyclin expressions, say Cyclins I and D, i.e., $\tau(c) :=$ Cyclin I$-$Cyclin D for a given cell c. Cyclin I and D represent the optimal choice based on the correlations in Figure 3.

2. Next, we compute the diffusivity matrix $\tilde{A}$ by using the latent time function in scVelo as our approximate clock.

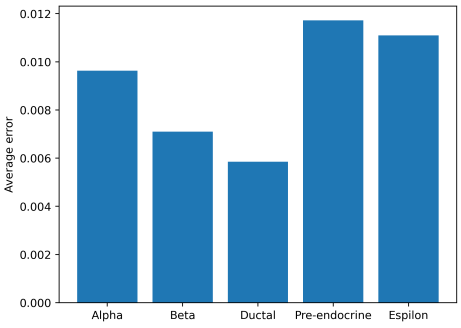
We use the total RNA expression levels data for alpha, beta, epsilon, pre-endocrine and ductal cells in the murine pancreatic single cell RNA-seq dataset [BPSTS+19] for this purpose. As can be seen in Figure 3, the error measured as the average entry-wise absolute difference of the computed diffusivity matrix is indeed small, much smaller than the variances (diagonals in the heatmaps), thus leading some amount of credence to our overall methods above.



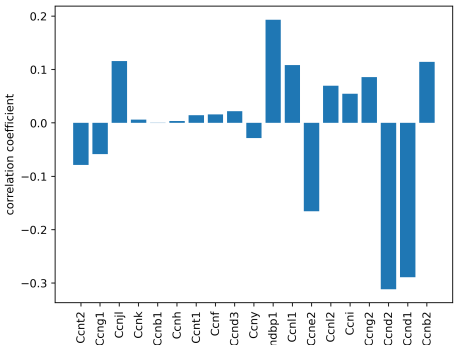
(a) Dynamical system obtained from latent time computation



(b) Dynamical system obtained from Cyclin clock



(c) Average entry-wise difference between the recovered dynamical system matrix for various cell clusters



(d) Correlations between various cyclin expressions and scVelo latent time

Fig. 3: Effectiveness of Cyclin clock

6 Conclusion

We have demonstrated that it is possible to approximately order the snapshots of a linear dynamical system and reconstruct it from few samples, by combining compressed sensing and filtering methods. We have shown how it can be practically useful in identifying prominent genomic pathways that differentiate two classes of samples in a couple of TCGA data-sets. Due to lack of resources and domain expertise we have not been able to verify the predictions ourselves. However, domain experts should be able to generate and verify such predictions to further their basic understanding of the genomic pathways underlying the conditions.

7 Acknowledgement

The authors are grateful to Dr. Jacob Glass for helpful discussion and comments.

References

BLP+20. Volker Bergen, Marius Lange, Stefan Peidli, F Alexander Wolf, and Fabian Theis. Generalizing RNA velocity to transient cell states through dynamical modeling. *Nature Biotechnology*, 38:1408–1414, 2020.

BPSTS+19. Aimée Bastidas-Ponce, Leander Dony Sophie Tritschler, Katharina Scheibner, Marta Tarquis-Medina, Ciro Salinno, Silvia Schirge, Ingo Burtscher, Anika Böttcher, Fabian J Theis, Heiko Lickert, and Mostafa Bakht. Comprehensive single cell mRNA profiling reveals a detailed roadmap for pancreatic endocrinogenesis. *Development*, 146(12):dev173849, 2019.

Can12. Cancer Genome Atlas Network: Daniel C Koboldt et al. Comprehensive molecular portraits of human breast tumours. *Nature*, 490(7418):61–70, 2012.

DLR18. Sjoerd Dirksen, Guillaume Lecue, and Holger Rauhut. On the gap between rip properties and sparse recovery conditions. *IEEE Transactions on Information Theory*, 64(8):5478–5487, 2018.

ea13. Cancer Genome Atlas Research Network: TJ Ley et al. Genomic and epigenomic landscapes of adult de novo acute myeloid leukemia. *N Engl J Med.*, 368(22):2059–74, 2013.

FR13. Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. 2013.

GFKH+22. Soudeh Ghafouri-Fard, Tayyebeh Khoshbakht, Bashdar Mahmud Hussen, Peixin Dong, Nikolaus Gassler, Mohammad Taheri, Aria Baniahmad, and Nader Akbari Dilmaghani. A review on the role of cyclin dependent kinases in cancers. *Cancer Cell International*, 22, 2022.

LHA14. Michael Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(550), 2014.

LWLA20. Dan Liu, Zidong Wang, Yurong Liu, and Fuad Alsaadi. Extended kalman filtering subject to random transmission delays: Dealing with packet disorders. *Information Fusion*, 60:80–86, 2020.

LXL21. Hongjiu Yangand Hui Li, Yuanqing Xia, and Li Li. Distributed Kalman Filtering Over Sensor Networks With Transmission Delays. *IEEE Transactions on Cybernetics*, 51(11):5511–5521, 2021.

NE19. Akram Nikfetrat and Reza Mahboobi Esfanjani. Self-tuning Kalman filter for compensation of transmission delay and loss in line-of-sight guidance. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 233(11):4191–4201, 2019.

Oks13. Bernt Oksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer, Berlin, 2013.

Sae19. Saelens, W. and Cannoodt, R. and Todorov, H. et al. A comparison of single-cell trajectory inference methods. *Nature Biotechnology*, 37:547–554, 2019.