

---

# Probabilistic basis decomposition for characterizing temporal dynamics of gene expression

---

Achille Nazaret<sup>\*1,2</sup> Joy Linyue Fan<sup>\*2</sup> Dana Pe'er<sup>3</sup> Elham Azizi<sup>1,2</sup>

## Abstract

Characterization of temporal dynamics in gene expression has the potential to reveal the complex regulatory mechanisms underlying cell state transitions in healthy and disease conditions. However, the ability to compare expression of genes or modules of genes between conditions or cell fates is hindered by the necessity of incorporating the temporal dependency associated with such transitions. We propose a probabilistic factor model for the decomposition of temporal gene expression into a representative set of patterns. We design a scalable inference method to process efficiently thousands of genes that evolve over hundreds of time points. The learned representative patterns reveal the dynamics underlying the data, meanwhile the decomposition of each gene can be used in downstream tasks such as clustering. We demonstrate the utility of the method by uncovering latent patterns and regulatory networks corresponding to leukemogenesis and progression of disease in Acute Myeloid Leukemia.

## 1. Introduction

Recent advances in the temporal analysis of single cell genomic data have enabled novel insights into understanding cell state transitions in normal development, regeneration, and disease. The temporal dimension may be experimentally observed, as in the case of acquiring multiple samples over the course of development or therapy; or computationally derived, via pseudotime or trajectory inference methods such as CellRank (Lange et al., 2022), Wishbone (Setty et al., 2016), and Monocle (Trapnell et al., 2014). The ability to rank cells along a therapeutic or developmental

timeline creates the need for computational methods and statistical metrics that can dissect the ways in which the gene expression landscape is shifting in conjunction. Importantly, identifying gene expression dynamics over time would help uncover gene programs associated with normal or diseased cell fates, which could lead to the discovery of novel markers involved in disease initiation and progression, guiding early disease detection and design of novel therapeutics.

The ability to perform comparisons between temporal gene patterns (Fig. 1), however, is not straightforward. Statistical tests such as common differential gene expression analyses assuming independence between samples or cells are insufficient due to the inability to account for temporal dynamics—two genes may have the same mean expression value, but opposite expression patterns, e.g. ascending versus descending over time. More sophisticated methods such as DPGP (McDowell et al., 2018) and TradeSeq (Van den Berge et al., 2020) are also limited—DPGP utilizes Gaussian Processes, which are computationally expensive and limited in scalability. TradeSeq, on the other hand, performs trajectory-based differential expression by utilizing the assumption that the temporal patterns may be approximated by splines, and relies on built-in denoising that limits user flexibility in upstream preprocessing.

We propose a scalable basis decomposition approach based on a probabilistic factor model for characterization of temporal gene expression patterns. Without making assumptions as to the form of the patterns, our method is able to learn a set of common patterns, called representative basis patterns, across all genes. Individual genes can then be represented as a combination of one or more representative patterns. To ensure scalability, we leverage neural networks to approximate Gaussian Processes in parameterizing the bases.

Since the bases are shared across all genes, comparison of patterns is then easily performed by computing the Euclidean distance between the basis weights. Importantly, this distance takes the temporal dependencies into account, since each basis represents a pattern preserving the order of cells or samples over time. Furthermore, the basis decomposition process allows for the uncovering of shared gene regulatory patterns, which may then be used to inform novel insights into underlying mechanisms determining cell fate.

---

<sup>\*</sup>Equal contribution <sup>1</sup>Department of Computer Science, Columbia University, New York, USA <sup>2</sup>Department of Biomedical Engineering, Columbia University, New York, USA <sup>3</sup>Memorial Sloan Kettering Cancer Center, New York, USA. Correspondence to: Achille Nazaret <achille.nazaret@columbia.edu>.

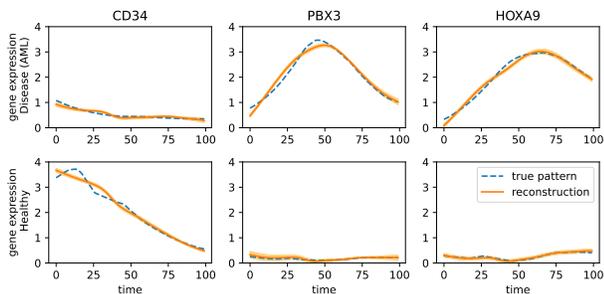


Figure 1: Examples of gene patterns under disease (top) and healthy (bottom) conditions. The model reconstructs the patterns with only  $K = 5$  bases.

## 2. Model

**The data.** The input data of the model is a collection of gene patterns observed over a time or pseudotime axis in different conditions, such as normal, disease, or perturbed (Fig. 1). A pattern  $\mu_{g,c}$  for a gene  $g$  under condition (or trajectory)  $c$  is a function  $t \in [0, 1] \mapsto \mu_{g,c}(t)$  which represents the expected expression of gene  $g$  at time  $t$  in condition  $c$ . This function can be obtained with time-series bulk RNA-seq or trajectory inference methods (Trapnell et al., 2014; Setty et al., 2016), applied in advance to single-cell RNA-seq data. While this paper focuses on applications to gene expression, the patterns can also represent other dynamic features such as chromatin accessibility (measured with ATAC-seq) or protein expression (CITE-seq).

The set of all genes is denoted as  $G$  and the set of conditions as  $C$ . For simplicity of the exposition, we restrict the conditions to  $C = \{\text{healthy}, \text{disease}\}$ . The input data is the collection of functions  $\mathcal{D} = (\mu_{g,c})_{g \in G, c \in C}$ . The data has  $|G| \cdot |C|$  observations, each of which is a function. In the proposed probabilistic model, each pattern  $t \mapsto \mu_{g,c}(t)$  is considered as a single observation. Figure 1 shows examples of gene patterns.

**The model.** In light of commonly used generative models (Blei, 2014), the proposed model is a linear factor model, operating in function space. For gene  $g$  and condition  $c$ , the model associates the data point  $\mu_{g,c}$  with a latent vector  $\beta_{g,c}$  of  $K$  dimensions. Each dimension  $k$  corresponds to a latent basis pattern  $t \mapsto b_k(t)$ . The  $\beta_{g,c,k}$  are coefficients for the function  $\mu_{g,c}$  in this basis. The observations  $\mu_{g,c}$  and the latent basis  $b_k$  are functions. The weights  $\beta_{g,c,k}$  combine the latent basis  $b_k$  to generate the observed function  $\mu_{g,c}$ . That is, informally,  $\mu_{g,c} \approx \sum_{k=1}^K \beta_{g,c,k} b_k$ . Each, basis function  $b_k$  forms a representative pattern shared by multiple genes.

**The weights  $\beta_{g,c,k}$ .** To ensure interpretability of the weights, the model mimics methods like non negative matrix factorization (Lee & Seung, 1999) and draws positive

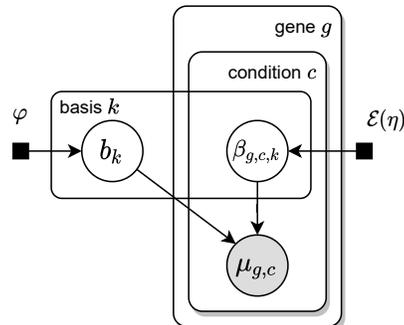


Figure 2: Graphical model for the functional factor model.

weights. With this, a non-zero weight  $\beta_{g,c,k}$  signifies that gene  $g$  in condition  $c$  exhibits the representative pattern  $k$ . Specifically, each weight  $\beta_{g,c,k}$  is drawn independently from an exponential distribution with mean  $\eta$ , denoted as  $\mathcal{E}(\eta)$ , and with density

$$p(\beta|\eta) = \eta e^{-\eta\beta} \mathbb{1}(\beta \geq 0).$$

In term of negative log-likelihood, this prior induces the  $\ell_1$  regularization of the coefficients  $\beta_{g,c,k}$ , with strength  $\eta$ , which will lead to sparsity and even better interpretability of the basis as shown in section 4.

**The function basis  $b_k$ .** In order to sample the basis functions  $b_k$ , we represent them as neural networks and we sample each  $b_k$  by drawing its neural networks parameters. Concretely, each basis function is modeled by a one-dimensional neural network with two hidden layers of 32 units each, followed by the  $\tanh$  activation. The neural network  $b_k$  is of the form

$$b_k : \mathbb{R} \rightarrow \mathbb{R}^{32} \rightarrow \mathbb{R}^{32} \rightarrow \mathbb{R}$$

and its parameters are denoted  $\theta_k$ . Each  $\theta_k$  is sampled from a centered diagonal normal distribution and the variance of each of its coordinates is set to the inverse of the input dimension of the linear layer in which it appears. The study of infinite neural networks in Neal (1996) demonstrates that with wide hidden units (here  $32 \gg 1$ ), and such a prior on the parameters, the induced prior in function space is close to a Gaussian process. Gaussian processes are used in other methods (McDowell et al., 2018), but are hardly scalable. Using neural networks for efficient computations solves the problem. We show samples from this prior in figure 6. The induced prior in function space is denoted by  $\varphi$ .

**The observations  $\mu_{g,c}$ .** Finally, the gene pattern  $\mu_{g,c}$  is generated from a distribution parameterized by  $\sum_k \beta_{g,c,k} b_k$ . More specifically,  $\mu_{g,c}$  is sampled from a Gaussian process<sup>1</sup>

<sup>1</sup>Because the Gaussian Process is used here only to define the distribution of the observations, and not to sample an unobserved latent variable, there is no computational difficulty in using it.

with mean  $\sum_k \beta_{g,c,k} b_k$  and with a white Gaussian noise kernel  $(x, x') \mapsto \sigma^2 \delta_{x,x'}$  of variance  $\sigma^2$ .

The generative process is represented graphically in figure 2 and proceeds as follows:

1. For each factor dimension  $k \in \llbracket 1, K \rrbracket$ , draw a basis function  $b_k$  from the function prior:  $b_k \sim \varphi(b_k)$  (that is draw weights  $\theta_k$  according to the prior detailed above)
2. For each gene  $g$ , and condition  $c$  do:

(a) For each factor dimension  $k$ , draw the basis weight  $\beta_{g,c,k} \sim \mathcal{E}(\eta)$

(b) Draw the observed function  $\mu_{g,c}$  from  $\mu_{g,c} \sim \mathcal{GP} \left( \sum_k \beta_{g,c,k} b_k, (x, x') \mapsto \sigma^2 \delta_{x,x'} \right)$ .

For simplicity of notations, the  $\theta_k$  are grouped in parameter  $\theta$ , and the  $\beta_{g,c,k}$  into parameter  $\beta$ .

### 3. Inference

Conditioned on the observed data  $\mathcal{D}$ , the generative model defines a posterior distribution over the global and local latent variables:  $p(\theta, \beta | \mathcal{D})$ . The posterior distribution places probability mass on the basis functions that represent the dominant patterns in the data. In addition, the posterior locates the local weights of each gene expression pattern  $\mu_{g,c}$ . The weights capture how each gene expression pattern exhibits each of the representative patterns. The weights can also be interpreted as a low-dimensional embedding of the gene expression trends, which can then be used for downstream analysis like clustering (see Section 4).

The exact posterior is intractable, so we use variational inference to approximate it (Jordan et al., 1999; Wainwright & Jordan, 2008; Blei et al., 2017). Variational inference defines a family of approximate distributions  $\mathcal{Q}$  over the latent variables, and then attempt to find the member of this family that is the closest to the exact posterior. This way, variational inference transforms the posterior inference task into an optimization problem.

**The variational family.** We use a mean-field approximation for the variational family, such that the variational distribution factorizes as

$$q(\theta, \beta) = \prod_{k,j} q(\theta_{k,j}) \prod_{g,c,k} q(\beta_{g,c,k}).$$

For the approximated posterior over the neural network parameters  $\theta_{k,j}$ , we use a Gaussian posterior with learnable mean and variance. For the basis weights  $\beta_{g,c,k}$ , we use a log-Gaussian posterior with learnable mean and variance.

For a more efficient algorithm, we can also use point estimates of these quantities instead of the Gaussian approximations. It is equivalent to performing a maximum a posteriori (MAP) estimation of the model parameters.

**The evidence lower bound.** In order to locate which approximate distribution  $q \in \mathcal{Q}$  is closest to the exact posterior  $p(\cdot | \mathcal{D})$ , variational inference minimizes the KL divergence between  $q$  and  $p(\cdot | \mathcal{D})$ . This is equivalent to maximizing the following objective function:

$$\mathcal{L}(q) = \mathbb{E}_{q(\theta, \beta)} [\log p(\theta, \beta, \mu) - \log q(\theta, \beta)]$$

also known as the evidence lower bound (ELBO).

With the specific choice of variational family, the ELBO takes the form:  $\mathcal{L}(q) = \mathbb{E}_{q(\theta, \beta)} [\log p(\mu | \theta, \beta)] - KL(q(\theta) || p(\theta | \mathcal{D})) - KL(q(\beta) || p(\beta | \mathcal{D}))$  where  $KL$  is the KL-divergence between distributions.

In practice, we evaluate the likelihood of the observations  $\log p(\mu | \theta, \beta)$  by evaluating the functions on 100 evenly spaced time points  $t \in T$ , and compute the density induced by the Gaussian process on these 100 points. Because of the form of the Gaussian process, the evaluated points each follow a Gaussian distribution. That is  $\log p(\mu_{g,c} | \beta, (b_k)) = \sum_{t \in T} -\frac{1}{2} \log(2\pi\sigma^2) - (\mu_{g,c}(t) - \sum_k \beta_{g,c,k} b_k(t))^2 / (2\sigma)$ . If the  $q$  are point estimates, the objective function  $\mathcal{L}(q)$  consists of the sum of the mean squared reconstruction error, the L1-regularization of the weights  $\beta$ , and the L2-regularization of the neural networks weights.

## 4. Experiments and applications

We implemented the method in Python using Pyro (Bingham et al., 2019). The practitioner can chose which family  $q$  to use between a Gaussian approximation or a point mass.

### 4.1. Scalability and performances on synthetic data

We test our method on synthetic data for different numbers of gene patterns  $|G|$  and increasingly longer time span  $[0, T]$ . We generate synthetic data by following the simulation proposed in McDowell et al. (2018) (Data simulations). Figure 7 in the appendix shows examples of synthetic gene patterns. We generate 3 datasets, each of them having 5 clusters of genes and 1,050 total genes. D1 spans over 10 time points, D2 over 25 time points, and D3 over 100 time points. The real data contains 3,130 genes over 100 time points and 2 conditions for each gene (effectively 6,260 genes).

**Clustering of genes** The factor model and its inference form an unsupervised method which provides a low dimensional representation for each gene pattern (using the basis weights). We can use these representations for downstream

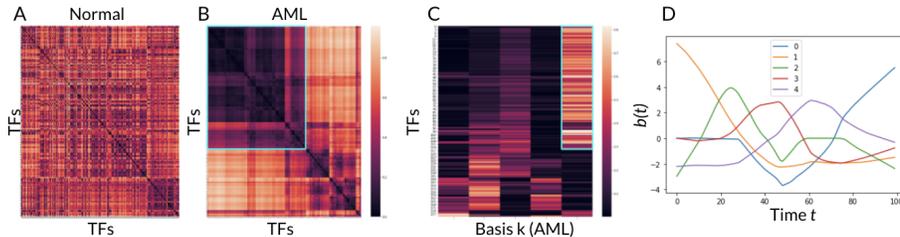


Figure 3: Heatmaps of pairwise distances of TFs in healthy  $c_1$  (A) and AML  $c_2$  (B). Basis weights for TFs in AML  $c_2$  (C). The order of TFs is the same across rows and columns in (A),(B), and rows in (C). Basis patterns (D).

Model	D1 ARI	D2 ARI	D3 ARI
DPGP	0.29	0.19	–
<b>Proposed model</b>	<b>0.95</b>	<b>0.99</b>	<b>1.0</b>

Table 1: ARI score of clustering synthetic datasets. The proposed model outperforms DPGP. The Adjusted Rank Index (ARI) improves for our model when the dataset contains more time points, suggesting that a longer time horizon helps to cluster the genes more accurately. DPGP timed out on 100 time points (D3).

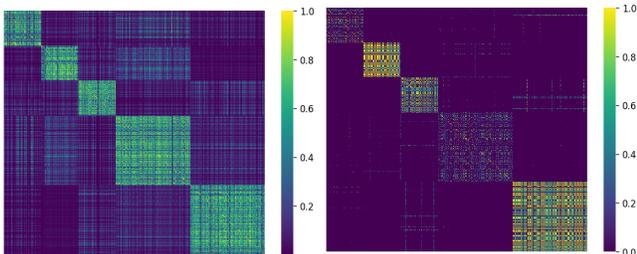


Figure 4: Gene similarity heatmaps returned by our method (left) and DPGP (right) for dataset D1. Each row and column represent a gene, ordered by groundtruth clusters. Similarity between  $g_1$  and  $g_2$  is defined by  $\exp(-\beta_{g_1}^\top \beta_{g_2})$  for our model and by  $\mathbb{P}(g_1, g_2 \text{ are co-clustered})$  for DPGP

tasks such as clustering. We compare our factor model against DPGP (McDowell et al., 2018), a specific method for clustering gene temporal patterns based on Gaussian Processes. In table 1 we show that our unsupervised factor model followed by K-means on the latent space outperforms DPGP on the clustering task. We provide more details in the appendix about the experiment.

Figure 4 shows the gene-gene similarity matrices computed respectively by our method and by DPGP.

**Scalability** Because of its functional formulation with neural networks, our proposed inference method can scale to a very large number of time points and genes. In table 2 we show the running time of our method, and of DPGP. Both

methods were run on a 2019 MacBook Pro with an 8-Core Intel Core i9 CPU. We notice that the number of time-points doesn’t affect the number of gradient updates per second of our method (because these updates are batched), whereas more time points can actually accelerate convergence. As a consequence, our method scales well with number of time points whereas DPGP doesn’t.

Model	D1 $T = 10$	D2 $T = 25$	D3 $T = 100$
DPGP	2h18m	6h06m	timeout
<b>Proposed model</b>	<b>2m18s</b>	<b>2m23s</b>	<b>1m45s</b>
<b>Proposed model (it/s)</b>	31	30	28

Table 2: Execution time of DPGP and of our model’s inference on the synthetic datasets. Our method’s number of gradient updates per second is almost invariant to the number of time points due to tensor multiplication acceleration. When the dataset grows larger, the convergence can even happen earlier.

## 4.2. Discover gene patterns in AML data

We apply the proposed method to single cell data derived from bone marrow biopsies of a patient with Acute Myeloid Leukemia (AML) and a healthy individual as control, sorting for stem and progenitor (CD34+) cells. Denoised expression and pseudotime are computed using our own trajectory inference method (in development), but any existing framework can be used (e.g. Gayoso et al. (2021)). The dataset consists of 3,130 genes, each having a healthy pattern (condition  $c_1$ ) and an AML pattern (condition  $c_2$ ).

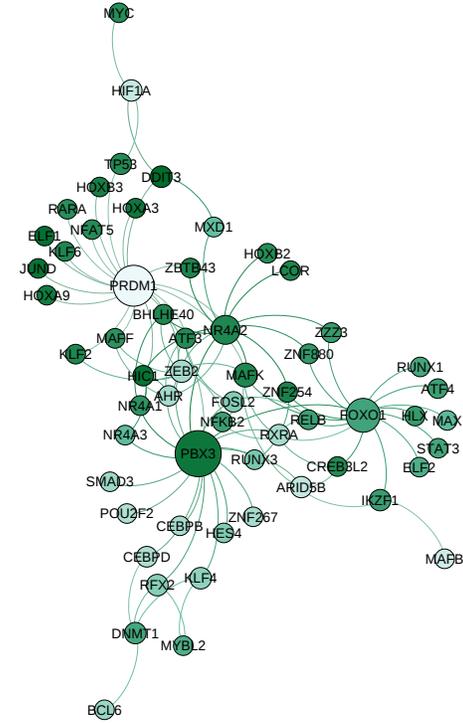
We fit the model by fixing the number of representative basis patterns to  $K = 5$ , which is large enough to obtain a good reconstruction error of the gene patterns. Figure 1 shows the patterns of three genes in the two conditions of healthy hematopoiesis (bottom) and leukemia (disease AML; top).

**Application for AML analysis** To investigate regulatory mechanisms underlying leukemia, we compare all transcription factors (TFs) patterns against each other by computing

the pairwise distance between their basis weights. Between genes  $i$  and  $j$  in condition  $c$  we compute the normalized Euclidean distance  $d(\beta_{i,c}, \beta_{j,c}) = \sqrt{\sum_{k=1}^K (\bar{\beta}_{i,c,k} - \bar{\beta}_{j,c,k})^2}$ , where  $\bar{\beta}_{i,c,k} = \frac{\beta_{i,c,k}}{\sum_{k=1}^K \beta_{i,c,k}}$ . Visual inspection of the results indicates the presence of a large block of highly similar transcription factors in the AML patient (Fig. 3B) that is not present in the healthy control (Fig. 3A). This block of transcription factors may indicate an increase in TF co-regulation that is gained with the onset of disease. Interestingly, when we observe the basis weights of each TF (Fig. 3C), it is apparent that the block of interest is highly weighted in basis 4. Inspection of the basis patterns (Fig. 3D) reveals that basis 4 increases towards the middle and end of the pseudotime, which suggests some role in disease progression. To validate these observations, we perform Gene Set Enrichment Analysis (GSEA), ranking genes by their weights in basis 4 (Supp. Fig. 6). Among the most enriched results are mTOR, TNFa, hypoxia, TGF-B, and p53, which are well-known oncogenic pathways. Together, these results indicate that our basis decomposition method results in interpretable basis patterns, with basis 4 being highly implicated in cancer onset.

In further refining our analyses, we highlight some key TFs pairs with high gains or losses in co-regulation. For this, we evaluate for each pair  $(i, j)$  of TFs, how the pairwise distance  $d(\beta_{i,c}, \beta_{j,c})$  changes between  $c = c_1$  and  $c = c_2$ . Using this information, we construct a graph where nodes represent the TFs found among the pairs with the greatest differences in pairwise distance, and edges indicate a gained or lost relationship between two nodes (Fig. 4). We color each node by a measure of how much the associated TFs is disrupted between the two conditions  $c_1$  and  $c_2$ . We define the disruption score as  $d(\beta_{i,c_1}, \beta_{i,c_2}) = \sqrt{\sum_{k=1}^K (\bar{\beta}_{i,c_1,k} - \bar{\beta}_{i,c_2,k})^2}$ . By defining both a disruption score as well as a pairwise distance, we are able to assess not only how a TF is itself changing between conditions (with  $d(\beta_{i,c_1}, \beta_{i,c_2})$ ), but also how its interactions with other TFs evolves (from  $d(\beta_{i,c_1}, \beta_{j,c_1})$  to  $d(\beta_{i,c_2}, \beta_{j,c_2})$ ).

In Figure 4, we notice four main hubs of TFs, centered around PBX3, NR4A2, FOXO1, and PRDM1. Importantly, all four of these TFs are implicated in the onset of AML: PBX3 is a known cofactor of HOXA9 in leukemogenesis (Li et al., 2013) and is significantly correlated with poor clinical outcome in AML patients (Guo et al., 2017); the NR4A family of genes is known to function in suppressing the onset of myeloid leukemias (Boulet et al., 2022); FOXO1 has been shown to lead to AML through the induction of leukemogenesis in HSCs via the Notch signaling pathway (Kode et al., 2016); and PRDM1 has been found to be up-regulated in RUNX1-mutated early-stage AML (Silva et al., 2009). Interestingly, PRDM1 has a low disruption



- fall, P., and Goodman, N. D. Pyro: Deep universal probabilistic programming. *The Journal of Machine Learning Research*, 20(1):973–978, 2019.
- Blei, D., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112:859–877, 2017.
- Blei, D. M. Build, compute, critique, repeat: Data analysis with latent variable models. *Annual Review of Statistics and Its Application*, 1:203–232, 2014.
- Boulet, S., Le Corre, L., Odagiu, L., and Labrecque, N. Role of nr4a family members in myeloid cells and leukemia. *Current Research in Immunology*, 3:23–36, 2022.
- Gayoso, A., Lopez, R., Xing, G., Boyeau, P., Wu, K., Jayasuriya, M., Melhman, E., Langevin, M., Liu, Y., Samarán, J., Misrachi, G., Nazaret, A., Clivio, O., et al. Scvi-tools: A library for deep probabilistic analysis of single-cell omics data. *bioRxiv*, 2021.
- Guo, H., Chu, Y., Wang, L., Chen, X., Chen, Y., Cheng, H., Zhang, L., Zhou, Y., Yang, F.-c., Cheng, T., et al. Pbx3 is essential for leukemia stem cell maintenance in mll-rearranged leukemia. *International journal of cancer*, 141(2):324–335, 2017.
- Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.
- Kode, A., Mosialou, I., Manavalan, S. J., Rathinam, C. V., Friedman, R. A., Teruya-Feldstein, J., Bhagat, G., Berman, E., and Kousteni, S. Foxo1-dependent induction of acute myeloid leukemia by osteoblasts in mice. *Leukemia*, 30(1):1–13, 2016.
- Lange, M., Bergen, V., Klein, M., Setty, M., Reuter, B., Bakhti, M., Lickert, H., Ansari, M., Schniering, J., Schiller, H. B., et al. Cellrank for directed single-cell fate mapping. *Nature methods*, pp. 1–12, 2022.
- Lee, D. D. and Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755): 788–791, 1999.
- Li, Z., Zhang, Z., Li, Y., Arnovitz, S., Chen, P., Huang, H., Jiang, X., Hong, G.-M., Kunjamma, R. B., Ren, H., He, C., Wang, C.-Z., Elkahlon, A. G., Valk, P. J. M., Döhner, K., Neilly, M. B., Bullinger, L., Delwel, R., Löwenberg, B., Liu, P. P., Morgan, R., Rowley, J. D., Yuan, C.-S., and Chen, J. PBX3 is an important cofactor of HOXA9 in leukemogenesis. *Blood*, 121(8):1422–1431, 02 2013. ISSN 0006-4971. doi: 10.1182/blood-2012-07-442004. URL <https://doi.org/10.1182/blood-2012-07-442004>.
- McDowell, I. C., Manandhar, D., Vockley, C. M., Schmid, A. K., Reddy, T. E., and Engelhardt, B. E. Clustering gene expression time series data using an infinite gaussian process mixture model. *PLoS computational biology*, 14(1), 2018.
- Neal, R. M. Priors for infinite networks. In *Bayesian Learning for Neural Networks*, pp. 29–53. Springer, 1996.
- Setty, M., Tadmor, M. D., Reich-Zeliger, S., Angel, O., Salame, T. M., Kathail, P., Choi, K., Bendall, S., Friedman, N., and Pe’er, D. Wishbone identifies bifurcating developmental trajectories from single-cell data. *Nature biotechnology*, 34(6):637–645, 2016.
- Silva, F. P., Swagemakers, S. M., Erpelinck-Verschueren, C., Wouters, B. J., Delwel, R., Vrieling, H., van der Spek, P., Valk, P. J., and Giphart-Gassler, M. Gene expression profiling of minimally differentiated acute myeloid leukemia: M0 is a distinct entity subdivided by runx1 mutation status. *Blood, The Journal of the American Society of Hematology*, 114(14):3001–3007, 2009.
- Trapnell, C., Cacchiarelli, D., Grimsby, J., Pokharel, P., Li, S., Morse, M., Lennon, N. J., Livak, K. J., Mikkelsen, T. S., and Rinn, J. L. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature biotechnology*, 32(4):381–386, 2014.
- Van den Berge, K., Roux de Bézieux, H., Street, K., Saelens, W., Cannoodt, R., Saeys, Y., Dudoit, S., and Clement, L. Trajectory-based differential expression analysis for single-cell sequencing data. *Nature communications*, 11(1):1–13, 2020.
- Wainwright, M. J. and Jordan, M. I. *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers Inc, 2008.

## A. Experiment on synthetic data

For the synthetic data experiments, we used the probabilistic model with  $K = 10$  factors. The synthetic data has 5 clusters. We ran the inference with a point mass posterior for the neural network weights and a Gaussian approximation for the basis weights.

We performed K-means on the factor model latent space for the downstream clustering. To mimic DPGP, which uses a Dirichlet Process to automatically find the number of clusters, we used a standard selection procedure to pick automatically a number of clusters for K-means. More precisely, we ran K-means for a wide range of  $K$  (from 2 to 20) and selected automatically the  $K$  producing the clustering with the highest silhouette score.

Finally, we used early stopping to stop the inference procedure and produce a clustering. We stopped the inference whenever the reconstruction loss stopped improving for 200 iterations.

## B. Additional figures

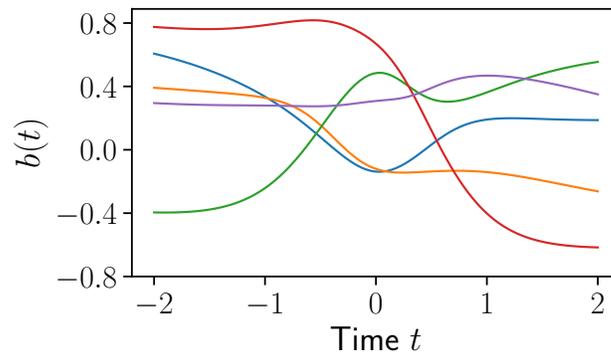


Figure 6: Five samples of one-dimensional neural networks from the prior over functions used by the model. The diversity of the samples demonstrates the relevance of the prior to generate basis functions that will capture key patterns of the genes evolution.

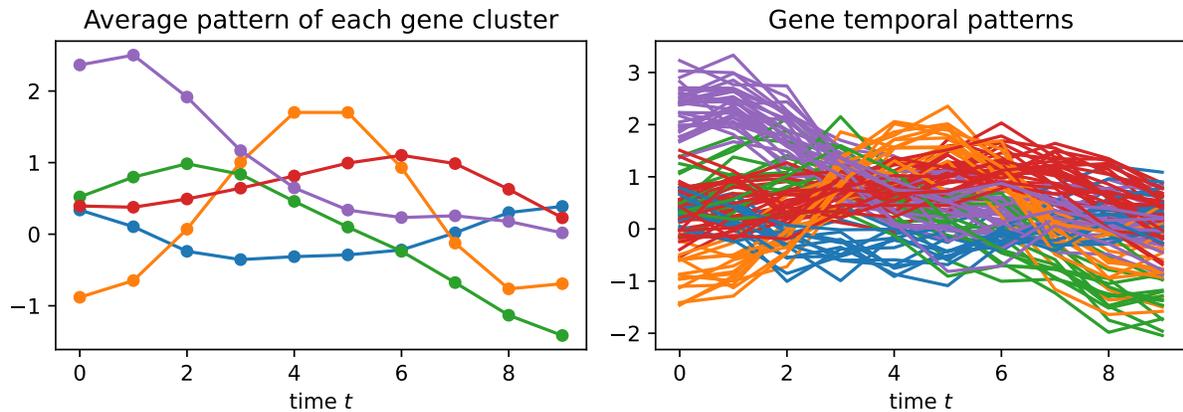


Figure 7: Example of synthetic data generated following the procedure in McDowell et al. (2018) (5 clusters, 10 time points, total of 105 genes)

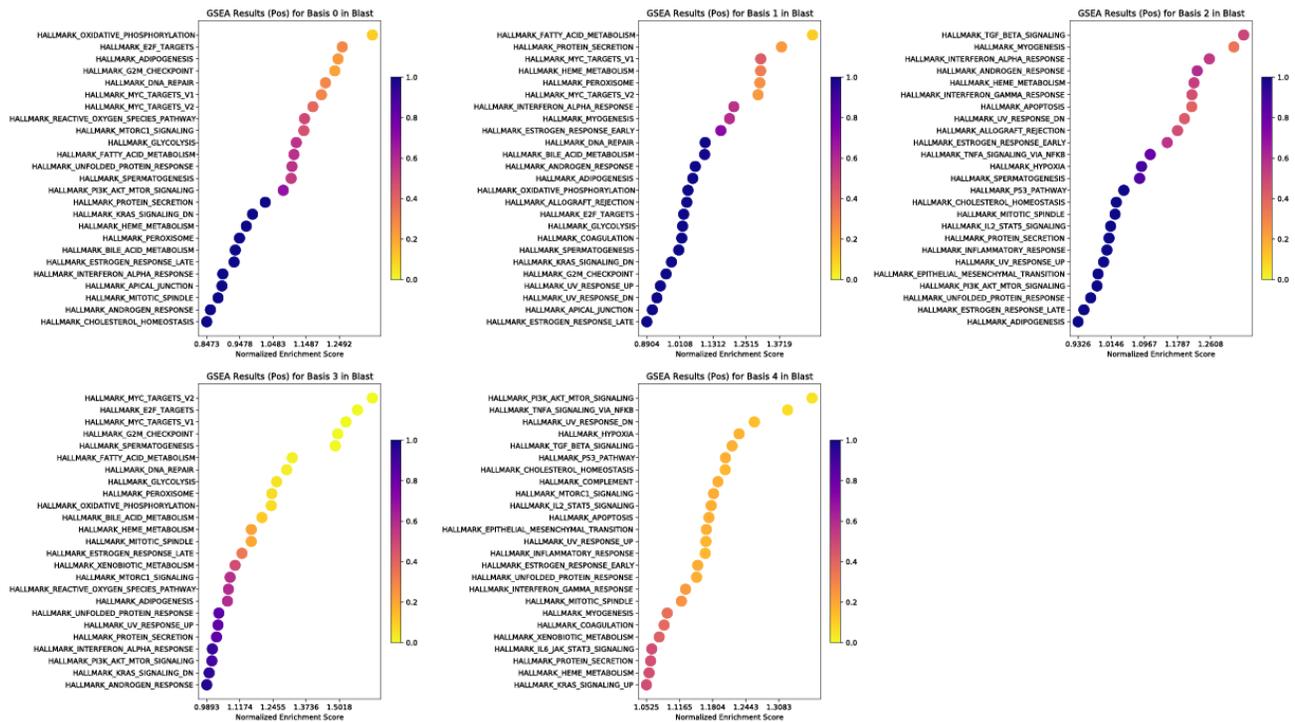


Figure 8: Gene Set Enrichment Analysis (GSEA) on the representative basis patterns in AML. The x-axis represents the normalized enrichment score for each pathway, and the color gradient indicates the false discovery rate (FDR). Notably, basis 4 contains a high number of inflammatory and cancer-associated pathways.