TWO-WAY SPARSITY FOR TIME-VARYING NETWORKS, WITH APPLICATIONS IN GENOMICS

By Thomas E. Bartlett[*,†], and Ioannis Kosmidis[‡,§] and Ricardo Silva[†,§]

Department of Statistics, University College London, WC1E 6BT, UK[†]
Department of Statistics, University of Warwick, Coventry, CV4 7AL, UK[‡]
The Alan Turing Institute, London, NW1 2DB, UK[§]

We propose a novel way of modelling time-varying networks, by inducing two-way sparsity on local models of node connectivity. This two-way sparsity separately promotes sparsity across time and sparsity across variables (within time). Separation of these two types of sparsity is achieved through a novel prior structure, which draws on ideas from the Bayesian lasso and from copula modelling. We provide an efficient implementation of the proposed model via a Gibbs sampler, and we apply the model to data from neural development. In doing so, we demonstrate that the proposed model is able to identify changes in genomic network structure that match current biological knowledge. Such changes in genomic network structure can then be used by neuro-biologists to identify potential targets for further experimental investigation.

1. Introduction. Network models have become an important topic in modern statistics, and the evolution of network structure over time (illustrated in Figure 1) is an important area of study. Network structures that evolve over time naturally occur in a range of applications. Examples of recent applications include evolving patterns of human interaction (Durante et al., 2016) such as in social networks (Sekara, Stopczynski and Lehmann, 2016), time-varying patterns of interaction between genes and their protein-products in biological networks (Alexander et al., 2009; Lebre et al., 2010), and time-varying patterns of connectivity in the brain (Schaefer et al., 2014). However, network models with temporal structure have only recently begun to be studied in detail in statistical research.

An important application area of statistical network models is genomics. Network models are a natural way to describe and analyse patterns of interactions (represented by network edges) between genes and their protein-products (represented by network nodes). An important interaction of this type is gene regulation, in which the protein-product of one gene influences the output level of the protein-product of another gene. Much gene regulation is characteristic of a particular cell type, so that a cell knows its role within the organism. These specific regulatory network structures that are characteristic of particular cell-types

[*]thomas.bartlett.10@ucl.ac.uk

Keywords and phrases: Bayesian inference, sparse statistical models, time-varying networks, genomic networks.

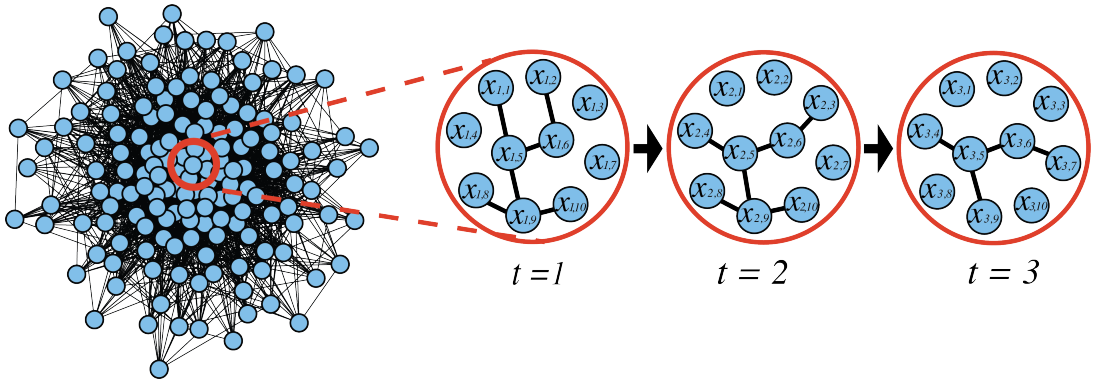

Fig 1: Model of time-varying network structure. Each $x_{t,i}$ represents a class label or continuous variable for node i (e.g., the expression-level of gene i) at time t. The links represent network interactions or dependencies between $x_{1,1}, x_{1,2}, ...$ (e.g. due to gene regulation), which may be different to those between $x_{2,1}, x_{2,2}, ...$ and $x_{3,1}, x_{3,2}, ...$ Hence, these network interactions may vary with time.

are established during embryonic development. Changes in normal gene regulation are also inherent to cancer progression, so that cells 'forget' how they should act, taking on pathological roles (regulatory network re-wiring) (Suvà et al., 2014). However, whilst network models are well established in genomics, historically these models have typically been static, ignoring the fact that genomic processes are inherently time-varying.

There are many examples of recent work on models of time-varying networks. In statistics, this work covers methods based on Markov processes (Crane et al., 2016), on dynamic Erdős-Rényi graphs (Rosengren and Trapman, 2016), and on sparse regression methods (Kolar et al., 2010). It also includes work on time-varying community structure (Zhang, Zhao and Zhang, 2012), on methods extending the stochastic block model (Xu and Hero III, 2013; Matias and Miele, 2016), and related non-parametric graphon-based methods (Pensky, 2016), as well as non-parametric methods for dynamic link prediction (Sarkar and Chakrabarti, 2014) and methods from Bayesian nonparametrics (Palla, Caron and Teh, 2016). Other related work includes sparse graphical models that can take account of different time-points (Kalaitzis et al., 2013).

Motivated by genomics applications, we propose a novel framework for modelling time-varying networks, by inducing *two-way sparsity* on local models of the connectivity of each node to all the others. This is achieved as follows. We start with a regression likelihood function that assumes that observations are mutually independent over time. Dependence is then induced through a novel prior structure that promotes sparsity in a two ways: across time, and within time. This decoupling of the induced sparsity is achieved through a copula specification for the parameters in the likelihood function. Specifically, the regression coefficients for one node across different time-points are jointly distributed according to a Gaussian copula with Laplace marginal distributions. The correlation matrix of the Gaussian copula is formed by assuming that the correlation between time-points decays

with time in a structured, parsimonious way that also ensures its positive definiteness. In this correlation matrix, the only free parameter is the correlation between consecutive time-points, which is given a reverse-exponential prior distribution with support in $[0, 1)$. This prior on the correlation across time discourages large differences in the regression coefficients between consecutive time-points and, as a consequence, also discourages large changes in the inferred structure of the network.

The decoupling of the marginal and dependence structure that is facilitated by the copula specification, and the particular form of the correlation matrix, allow for precise control of marginal priors. This decoupling also makes the adoption of generalisations of the Bayesian lasso, such as the horseshoe (Carvalho, Polson and Scott, 2010), easy to implement in place of the marginal Bayesian lasso prior that we use. The prior dependence among parameters across time can also be viewed as a Bayesian version of the fused lasso (Tibshirani et al., 2005), while within each time-slice we directly utilise existing work on the Bayesian lasso (Park and Casella, 2008). In fact, the proposed modelling framework has the Bayesian lasso as a special case, when the correlation between time-points is set to zero. From a frequentist point of view, the sparsity structure we propose would fall within the remit of the generalised lasso (Tibshirani et al., 2011), which has the fused lasso as a special case (Tibshirani et al., 2005). Bayesian versions of the fused lasso have also been proposed (Kyung et al., 2010; Shimamura et al., 2016). However, a key difference between those methods and the modelling framework we propose, is the formal decoupling of sparsity across time (which the fused lasso induces), from sparsity within time. Importantly, we are able to apply this proposed modelling framework locally to each network node, as previous authors have done (Kolar et al., 2010). Because these local model fits are mutually independent they can easily be carried out sequentially or in parallel, meaning that in practice, we are able to work with large networks of tens of thousands of nodes. The novel prior structure proposed, which enables the time-varying network inference, is also of interest more generally beyond the context of network science. This novel prior structure is relevant in any context where sparse regression with time-varying regression parameters is desirable.

The rest of the paper is structured as follows. In Section 2, we set up notation, and specify the model. Then, in Section 3 we present the results of fitting the model to simulated data, and in Section 4 we present the results of fitting the model to single-cell transcriptome data. Finally, in Section 5, we summarise our findings and discuss their broader context. The Supplementary Material we provide all proofs and derivations, data pre-processing details, and Supplementary Figures, as well as a freely available software implementation of our proposed model and algorithm.

2. Proposed methodology.

2.1. *Data description.* The two-way sparsity that is induced by the proposed modelling framework is motivated by the problem of inferring time-varying structure in genomic

networks. In these networks nodes represent genes: for each node there are observations or measurements of the activity level of the corresponding gene (the 'gene-expression level'). These node-specific observations make up the data-set. The expression-level of a gene is generally influenced by the expression-level of several other genes (in a process called 'gene regulation'). Hence, a natural application for models of time-varying networks is understanding dynamic patterns of gene-regulation in biological processes, such as neural development. Measurements of gene transcript counts are often used as a surrogate for gene expression level in RNA sequencing data, and hence we base our real-data example on single-cell transcriptomic data. We note that single-cell transcriptomic data is a type of single-cell gene-expression data.

Single-cell gene-expression data are ideal for this application, because data from a study of this type will typically be obtained from a heterogeneous mixture of cells, each of which may be at a different point on a trajectory through the biological process under investigation. For example, in the context of neural development, some of these cells may be stem-cells, whereas some may be fully differentiated cells (e.g., neurons), with a whole spectrum of cells in between. Each cell can be thought of as an independent sample from the underlying latent biological process; in this example, that process is neural development. Thus, we can think of the progression of a cell through this process of neural development in terms of a 'developmental trajectory'. The progression along such a developmental trajectory can be quantified in terms of 'developmental time', which is simply a measure of a temporally-ordered progression through the process of cellular development. For each of the cell-samples in the data, no information is available other than its high-dimensional gene-expression measurements. Hence, it is necessary to first infer the 'developmental time' of the cell-samples before fitting any time-varying network model. This leads to an ordered sequence of pseudo-temporal measurements $x_{1,i}, x_{2,i}, ...x_{t,i}, ..., x_{T,i}$ of the log-expression level for gene i. Importantly, the $x_{1,i}, x_{2,i}, ...$etc are taken from different cell samples for each pseudo-time point, and are hence independent. Inference like this is more generally referred to as 'pseudo-time' inference, and several methods exist to carry it out: see for example work by Qiu et al. (2011) and Trapnell et al. (2014).

2.2. *Model overview.* We develop a model for each target-node conditional on all the other nodes, and then we apply this model to several target-nodes of interest. This is different from, for example, the work of Friedman, Hastie and Tibshirani (2008) and Fan, Feng and Wu (2009) who consider the modelling of all nodes jointly. Such a target-node approach has been used previously by Kolar et al. (2010), and it allows the network structure to be inferred independently around each target-node $i \in \{1, ..., p\}$. This strategy has several advantages. Firstly, variable screening can be applied before model fitting. This allows the dimensionality of the problem to be reduced from p of the order of tens of thousands down to p' of the order of a few hundred for each parallel model fit around a target-node, whilst still allowing the global network structure to be estimated over tens of thousands of target-nodes, if required. Our modelling strategy also allows the local network structure

to be estimated around only a small number of target-nodes if required, controlling computational expense, whilst still inferring the connected node-sets from tens of thousands of nodes. Inference is carried out with a sparse linear model, taking the observations for node i at time t as the response, and the observations for all nodes $j \neq i$ at time t as potential predictors. From these potential predictors, the set of predictors 'chosen' by the sparse model fit are then used to infer the network structure. Specifically, we want to infer the network structure around a fixed set of nodes with a set of edges that varies with time. In this scenario, only the patterns of interconnectivity change as the network evolves (Figure 1), which is the scenario most relevant to genomics applications. Such a network can be represented with a time-varying adjacency matrix $\mathbf{A}$, where $A_{i,j,t}$ denotes the absence ($A_{i,j,t} = 0$) or presence ($A_{i,j,t} > 0$) of an edge between nodes i and j at time t. We note that under this scheme, the local model fit (which is responsible for the computational load) does not depend on the network estimation (which takes place subsequently). The inferred network is a particular summary of the posteriors that are obtained from several of our model fits. We propose a model for node-wise regression, and we suggest how to summarise these models over several nodes of a network.

2.3. *Model likelihood.* We assume a likelihood function where observations are mutually independent over time. This is an assumption that is compatible with high-dimensional gene-expression data, where no single cell can be measured at more than one time-point. We note that this implies that observations are independent at different time-points. Let $\mathbf{X}$ represent the full data-set for the nodes shown in Figure 1, with time varying down the rows, and with each node corresponding to a different column. Then, $x_{t,i}$ denotes the value for some node in the system at time $t \in \{1, ..., T\}$, for $i \in \{1, ..., p\}$, and the row-vector $\mathbf{x}_{t,\setminus i}$ denotes the values for the other $p-1$ nodes at time t. We model the dependence of $x_{t,i}$ on $\mathbf{x}_{t,\setminus i}$ as:

$$x_{t,i} = a_i + \mathbf{b}^{(i)}_{t,:}\ \mathbf{x}^{\top}_{t,\setminus i} + \epsilon_{t,i}, \tag{1}$$

where $\mathbf{b}^{(i)}_{t,:}$ is a vector of linear model parameters, and $\epsilon_{t,i} \sim \mathcal{N}\left(0, \tau_i^{-1}\right)$.

The response variable $x_{t,i}$ corresponds to the observations for a 'target' node around which we are modelling the local network structure, whereas the variables represented by $\mathbf{x}_{t,\setminus i}$ correspond to the observations for all the other nodes of the network. To model the whole network, we must fit model (1) around each target-node in turn. We note that here we make an assumption about the existence of a global undirected Markov network (Lauritzen, 1996) that explains the independence constraints in the model. This assumption has also been used previously by Kolar et al. (2010) in an equivalent context. We note that our approach does not enforce hard constraints, such as $\mathbf{b}^{(i)}_{t,j} = \mathbf{b}^{(j)}_{t,i}$. However, it is computationally very expensive to work with a global, coherent model, where such constraints can be enforced. In this work, we have opted to sacrifice some coherence for the sake of computational efficiency. This enables us to estimate quantities of interest in a

computationally-efficient manner through an overparameterized representation of a joint model. It also enables us to focus on a particular subset of nodes of interest without having to go through an overly-expensive computation for the estimation of a global, coherent model.

Using $\mathbf{b}_{:,j}^{(i)}$ to denote the column-vector of model parameters for covariate j for $t \in \{1, ..., T\}$, we collect parameters in matrix $\mathbf{B}^{(i)} = \left[\mathbf{b}_{:,1}^{(i)}, \mathbf{b}_{:,2}^{(i)}, ..., \mathbf{b}_{:,p-1}^{(i)}\right]$. In the next section, we postulate a prior for dependencies within each column j of $\mathbf{B}^{(i)}$, whilst noting that the columns of $\mathbf{B}^{(i)}$ (each corresponding to a different node as covariate) are independent of each other. We also introduce the notation $x_{t,i,k}$ and $\mathbf{x}_{t,\setminus i,k}$ to represent observations of $x_{t,i}$ and $\mathbf{x}_{t,\setminus i}$ for sample $k \in \{1, ..., n_t\}$ at time t.

We denote $\mathbf{x}_{:,i} = [x_{1,i,1}, ..., x_{1,i,n_1}, ..., x_{t,i,1}, ..., x_{t,i,n_t}, ..., x_{T,i,1}, ..., x_{T,i,n_T}]^\top$ and $\mathbf{X}_{:,\setminus i} = [\mathbf{x}_{1,\setminus i,1}^\top, \mathbf{x}_{1,\setminus i,n_1}^\top, ..., \mathbf{x}_{t,\setminus i,1}^\top, ..., \mathbf{x}_{t,\setminus i,n_t}^\top, ..., \mathbf{x}_{T,\setminus i,1}^\top, ..., \mathbf{x}_{T,\setminus i,n_T}^\top]^\top$, where $\mathbf{x}_{:,i}$ is column i of data-matrix $\mathbf{X}$, and $\mathbf{X}_{:,\setminus i}$ is data-matrix $\mathbf{X}$ without column i. Hence, we can write the model likelihood for the target-node i as:

$$P(\mathbf{x}_{:,i}|\mathbf{X}_{:,\setminus i}, \mathbf{B}^{(i)}, a_i, \tau_i) = \prod_{t=1}^{T}\prod_{k=1}^{n_t}\sqrt{\frac{\tau_i}{2\pi}}e^{-\tau_i(x_{t,i,k}-\mathbf{b}_{t,:}^{(i)}\cdot\mathbf{x}_{t,\setminus i,k}^\top - a_i)^2/2}. \tag{2}$$

We note that we consider likelihoods in the form of equation (2) for each target-node i.

2.4. *Priors with decoupled two-way sparsity.* We model the regression coefficients $\mathbf{b}_{:,j}^{(i)}$ across time-points $t = 1, ..., T$ with a Gaussian copula with Laplace marginal distributions, as follows. The elements of $\mathbf{b}_{t,:}^{(i)}$ $(t = 1, ..., T)$ are marginally distributed as $b_{t,j}^{(i)} \sim$ Laplace $(1/\lambda)$, with probability density function $\frac{\lambda}{2}e^{-\lambda|\cdot|}$ and cumulative distribution function $F_{\mathcal{L}}[b_{t,j}^{(i)}]$, for $t \in \{1, ..., T\}$ and $j \in \{1, ..., p-1\}$. Hence, $\Phi^{-1}\left\{F_{\mathcal{L}}[b_{t,j}^{(i)}]\right\}$ follows a Gaussian distribution for $t \in \{1, ..., T\}$ and $j \in \{1, ..., p-1\}$, where Φ is the standard-normal cumulative distribution function. The dependencies between $\Phi^{-1}\left\{F_{\mathcal{L}}[b_{t,j}^{(i)}]\right\}$ and $\Phi^{-1}\left\{F_{\mathcal{L}}[b_{t+1,j}^{(i)}]\right\}$ are then modelled through their joint distribution as:

(3)

$$\begin{bmatrix} \Phi^{-1}\left\{F_{\mathcal{L}}[b_{1,j}^{(i)}]\right\} \\ \Phi^{-1}\left\{F_{\mathcal{L}}[b_{2,j}^{(i)}]\right\} \\ \vdots \\ \Phi^{-1}\left\{F_{\mathcal{L}}[b_{T,j}^{(i)}]\right\} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{\Sigma}_j^{(i)}\right), \text{ with } \mathbf{\Sigma}_j^{(i)} = \begin{bmatrix} 1 & \rho_j^{(i)} & \left(\rho_j^{(i)}\right)^2 & \cdots & \left(\rho_j^{(i)}\right)^T \\ \rho_j^{(i)} & 1 & \rho_j^{(i)} & \cdots & \left(\rho_j^{(i)}\right)^{T-1} \\ \left(\rho_j^{(i)}\right)^2 & \rho_j^{(i)} & 1 & \cdots & \left(\rho_j^{(i)}\right)^{T-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \left(\rho_j^{(i)}\right)^T & \left(\rho_j^{(i)}\right)^{T-1} & \left(\rho_j^{(i)}\right)^{T-2} & \cdots & 1 \end{bmatrix},$$

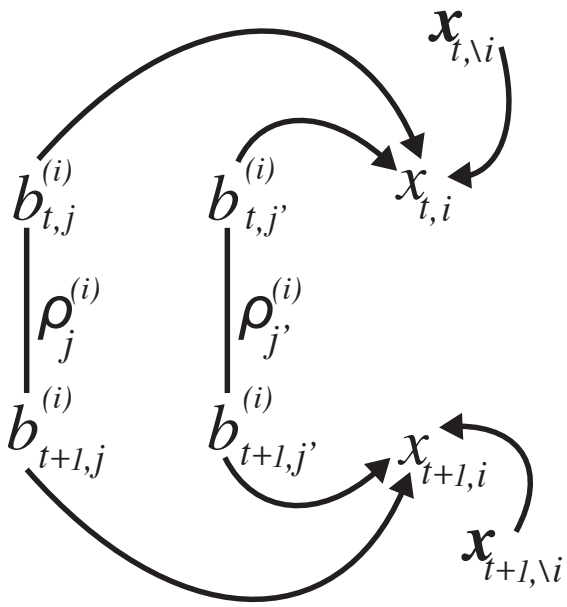

Fig 2: Chain graphical model (Lauritzen, 1996). The diagram shows the dependence of $x_{t,i}$ (the value of the target-node at time t) on $\mathbf{x}_{t,\setminus i}$ (which represents the values of two other nodes j and j′ at time t), and on the corresponding model parameters $b^{(i)}_{t,j}$ and $b^{(i)}_{t,j'}$. Model parameters are correlated across time, such that $\Phi^{-1}\left\{F_{\mathcal{L}}[b^{(i)}_{t,j}]\right\}$ and $\Phi^{-1}\left\{F_{\mathcal{L}}[b^{(i)}_{t+1,j}]\right\}$ have correlation $\rho^{(i)}_j$.

and hence the regression coefficients are modelled as a Gaussian copula:

$$F\left[\mathbf{b}^{(i)}_{:,j}\right] = \Phi_p\left[\Phi^{-1}\left\{F_{\mathcal{L}}[b^{(i)}_{1,j}]\right\}, \Phi^{-1}\left\{F_{\mathcal{L}}[b^{(i)}_{2,j}]\right\}, ..., \Phi^{-1}\left\{F_{\mathcal{L}}[b^{(i)}_{T,j}]\right\}; \mathbf{\Sigma}^{(i)}_j\right].$$

The correlation parameter $\rho^{(i)}_j$ is assumed to have a reverse-exponential distribution with support $[0, 1)$ and density

$$f_{\text{rexp}}[\rho^{(i)}_j] \sim k e^{k\rho^{(i)}_j}/(e^k - 1). \tag{4}$$

The structure of $\mathbf{\Sigma}^{(i)}_j$ is such that transformed model parameters at adjacent points in time, such as $\Phi^{-1}\left\{F_{\mathcal{L}}[b^{(i)}_{t,j}]\right\}$ and $\Phi^{-1}\left\{F_{\mathcal{L}}[b^{(i)}_{t+1,j}]\right\}$, have correlation $\rho^{(i)}_j$ (Figure 2). Then, the transformed parameters separated by two time-points have correlation $(\rho^{(i)}_j)^2$, etc. Thus, also denoting the sequence of transformed model parameters $\Phi^{-1}\left\{F_{\mathcal{L}}[b^{(i)}_{1,j}]\right\}$, $\Phi^{-1}\left\{F_{\mathcal{L}}[b^{(i)}_{2,j}]\right\}$,...,$\Phi^{-1}\left\{F_{\mathcal{L}}[b^{(i)}_{t,j}]\right\}$ forms a Markov chain, meaning that $\mathbf{\Sigma}^{(i)}_j$ is guaranteed to be positive-definite for $\rho^{(i)}_j \in [0, 1)$, and by construction

$$b^{(i)}_{t+1,j} \perp b^{(i)}_{t-1,j}, b^{(i)}_{t-2,j}, ... | b^{(i)}_{t,j}. \tag{5}$$

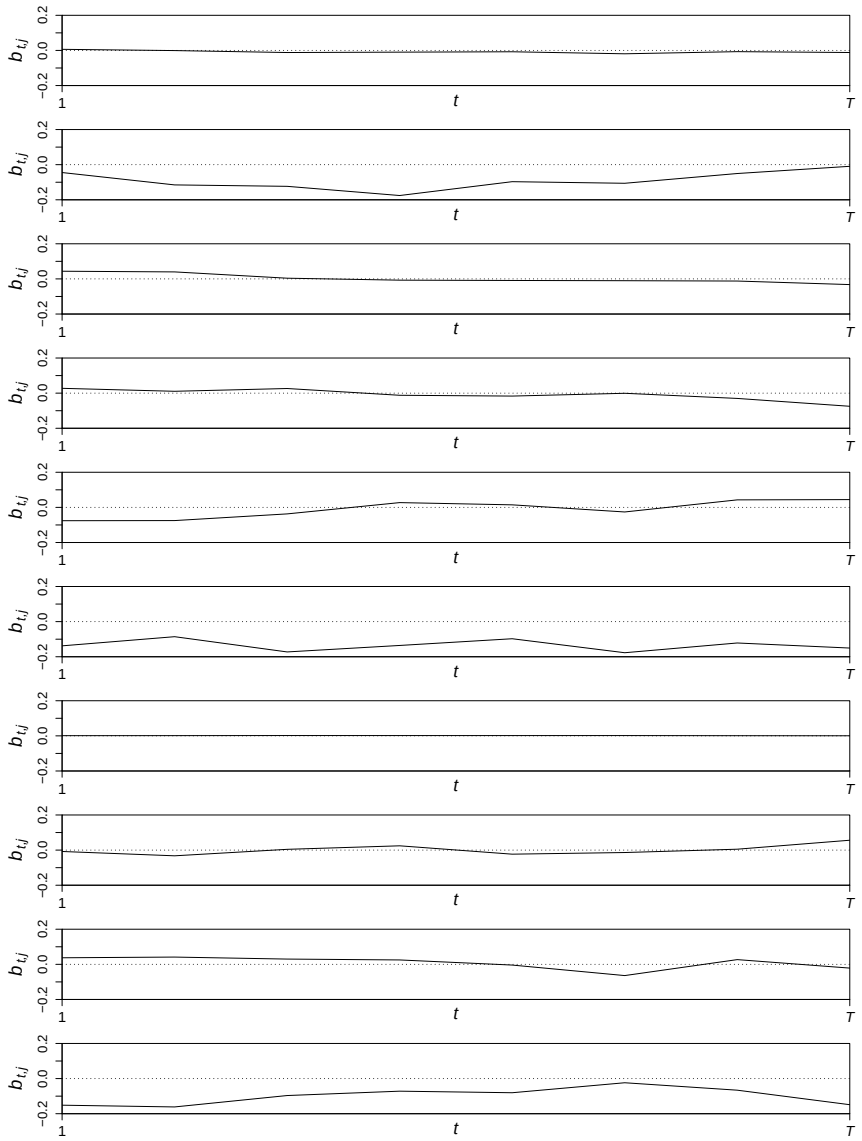

Fig 3: Samples from the prior on $\mathbf{b}_{t,j}$ *plotted against* t*, illustrating their correlation structure over time. These results are with* $T = 8$*,* $\lambda = 20$ *and* $k = 1$*.*

Such a construction for $\mathbf{\Sigma}_j^{(i)}$ discourages differences in the regression coefficients for the same covariate between adjacent time-points, and hence also discourages changes in the network structure over time, resulting in sparsity across time. Then, transforming the $\Phi^{-1}\left\{F_{\mathcal{L}}[b_{t,j}^{(i)}]\right\}$ back to $b_{t,j}^{(i)}$, where the $b_{t,j}^{(i)}$ are marginally Laplace distributed, achieves sparsity within time by discouraging regression coefficients from taking non-zero values,

hence also encouraging discovery of sparse network structures. Figure 3 shows ten samples from our proposed prior on $b_{t,j}^{(i)}$ plotted against t, and demonstrates the correlation structure enforced by the prior over time. We again note that while $b_{t,j}^{(i)}$ and $b_{t+1,j}^{(i)}$ are correlated, $b_{t,j}^{(i)}$ and $b_{t,j'}^{(i)}$ are independent.

Recent work by Shimamura et al. (2016) that takes a Bayesian approach to generalising the fused lasso could be used similarly to the approach we propose, by modelling the same set of covariates at multiple time-points whilst enforcing smooth changes across time as well as sparsity overall. However, Shimamura et al. (2016) achieve their result by simply multiplying together separate frequentist-inspired priors for smoothness across time and for sparsity. Specifically, they multiply together a Laplace prior to penalise individual non-zero model parameters, with the ultra-sparse negative-exponential-gamma (NEG) prior to penalise non-zero differences in parameters. The Laplace-NEG prior is defined (choosing notation to be consistent with that of our model) as:

$$P(\mathbf{b}_{:,j}^{(i)}) \propto \prod_{t=1}^{T} \text{Laplace}(b_{t,j}^{(i)}|\lambda) \prod_{t=2}^{T} \text{NEG}(b_{t,j}^{(i)} - b_{t-1,j}^{(i)}|\lambda^{\dagger}, \gamma), \tag{6}$$

where the Laplace density is defined as $\frac{\lambda}{2}e^{-\lambda|\cdot|}$, and

$$\text{NEG}(\cdot|\lambda^{\dagger}, \gamma) = \int_0^{\infty}\int_0^{\infty} f_{\mathcal{N}}(\cdot|0, \tau^2) f_{\gamma}(\tau^2|1, 1/\psi) f_{\gamma}(\psi|\lambda^{\dagger}, 1/\gamma^2) d\tau^2 d\psi,$$

where $f_{\mathcal{N}}$ and f_{γ} are the Normal and Gamma densities, respectively. Sampling from the distribution of equation (6) is done by simulating exponential and gamma random variables, which are then used to form the precision matrix of a multivariate normal distribution, as specified by Shimamura et al. (2016). In contrast to the Laplace-NEG prior, the model we propose retains the property that, marginally, each parameter still follows the Bayesian lasso prior (Park and Casella, 2008). In particular, if we set $\rho_j^{(i)} = 0$ (for $j = 1, 2, ..., p-1$), then the model we propose is exactly the same as the Bayesian lasso. This is important because it makes it easier to set priors, including variants of the Bayesian lasso that avoid its well-known shortcomings (see for example the work by Castillo et al. (2015) and van der Pas et al. (2016)). Although we will not consider such variants here, they follow directly by mimicking the construction using the Bayesian lasso.

The novel prior we use on $\rho_j^{(i)}$ is a 'reverse exponential prior' (equation (4)). Figure 4a shows the probability density function of the reverse-exponential prior for different values of hyper-parameter k. Figure 4b then shows heatmaps of the bivariate density distributions of samples from the decoupled-sparsity prior for a parameter j over two time-points, i.e., $\mathbf{b}_{:,j}^{(i)} = [b_{1,j}^{(i)}, b_{2,j}^{(i)}]^{\top}$, for a range of values of λ and k (the corresponding marginal densities are shown in Figures S6 and S7 in the Supplementary Information). For comparison, Figures S3 - S5 in Supplement D show samples from the Laplace-NEG prior as defined in equation (6), for various values of λ (which acts equivalently to λ in our model, controlling sparsity of

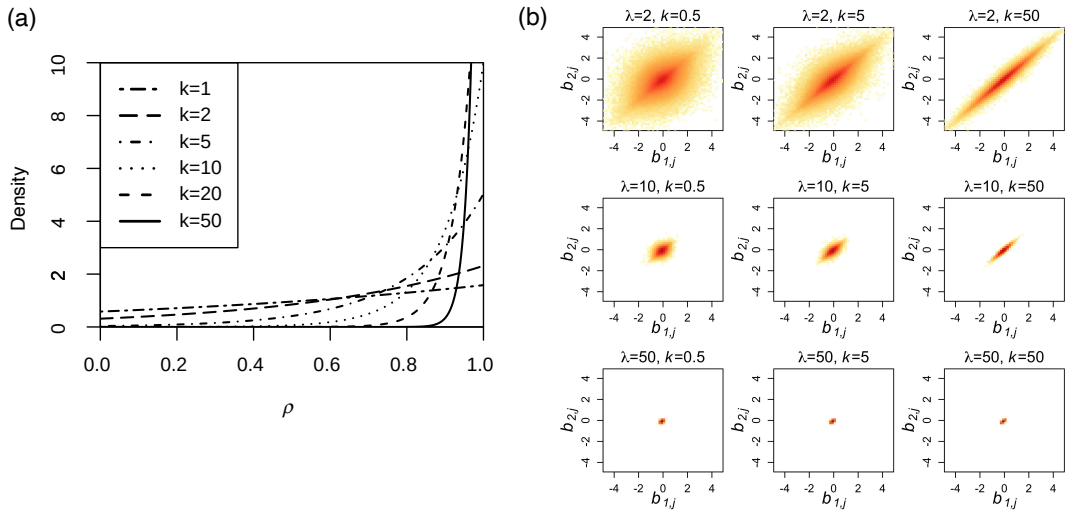

Fig 4: (a) Density function of the reverse-exponential prior. (b) Heatmaps of the bivariate log-densities of prior samples for $\mathbf{b}_{:,j}^{(i)} = [b_{1,j}^{(i)}, b_{2,j}^{(i)}]^\top$.

individual model parameters), and various values of $\lambda^\dagger$ and γ (which both act equivalently to k in our model, controlling sparsity of differences between model parameters). The main difference between these priors is that our decoupled-sparsity prior still marginally follows the Bayesian lasso prior, and is hence a direct generalisation of the Bayesian lasso to this setting with time-varying model parameters. In other words, our prior does what the Laplace-NEG does, but with the added benefit that we generalise the Bayesian lasso.

2.5. *Posterior inference.* The order-1 Markovian relations specified by equation (5) are also computationally attractive, because they result in models with banded precision matrices. From equation (5), and denoting $\theta_{t,j}^{(i)} = \Phi^{-1}\left\{F_{\mathcal{L}}[b_{t,j}^{(i)}]\right\}$ it follows that the partial correlation of $\theta_{t+m,j}^{(i)}$ with $\theta_{t+l,j}^{(i)}$ will be zero for all $|m-l| > 1$. Hence, all entries of the precision matrix $[\mathbf{\Sigma}_j^{(i)}]^{-1}$ will be zero except the diagonal and the elements immediately adjacent to it (i.e., the sub- and super-diagonals). These relationships allow all the entries of this precision matrix to be found easily in terms of $\rho_j^{(i)}$ by solving $[\mathbf{\Sigma}_j^{(i)}]^{-1}\mathbf{\Sigma}_j^{(i)} = \mathbb{I}$, which gives:

$$\left(\left[\mathbf{\Sigma}_j^{(i)}\right]^{-1}\right)_{t,t'} = \begin{cases} 1/(1-[\rho_j^{(i)}]^2), & \text{if } t'=t=1 \text{ or } t'=t=T, \\ (1+[\rho_j^{(i)}]^2)/(1-[\rho_j^{(i)}]^2), & \text{if } t'=t>1 \text{ and } t'=t<T, \\ -\rho_j^{(i)}/(1-[\rho_j^{(i)}]^2), & \text{if } t'=t+1 \text{ or } t'=t-1, \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where $\left(\left[\mathbf{\Sigma}_j^{(i)}\right]^{-1}\right)_{t,t'}$ represents the (t,t') element of the precision matrix $[\mathbf{\Sigma}_j^{(i)}]^{-1}$. A full derivation of equation (7) is given in Supplement A.

The model parameters $\mathbf{B}^{(i)}$ can be sampled directly from multivariate Normal distributions, without needing the intermediate transformation to the marginally Laplace-distributed variables described in Section 2.4. This can be achieved with an algebraic manipulation which is an extension from the Bayesian lasso, as follows. The Laplace distribution can be written as an uncountable mixture of zero-mean Normal distributions, with the variances of the mixture components distributed as $\text{Exp}(\frac{\lambda^2}{2})$ (Andrews and Mallows, 1974; Park and Casella, 2008). Specifically,

$$P(b_{t,j}^{(i)}|\lambda) = \frac{\lambda}{2}e^{-\lambda|b_{t,j}^{(i)}|} = \int_0^\infty P(b_{t,j}^{(i)}, s_j^{(i)}|\lambda)ds_j^{(i)},$$

where

$$P(b_{t,j}^{(i)}, s_j^{(i)}|\lambda) = \frac{1}{\sqrt{2\pi s_j^{(i)}}}e^{-[b_{t,j}^{(i)}]^2/[2s_j^{(i)}]}\frac{\lambda^2}{2}e^{-\lambda^2 s_j^{(i)}/2},$$

for $s_j^{(i)} \sim \text{Exp}(\frac{2}{\lambda^2})$. This says that we will achieve $b_{t,j}^{(i)}$ being marginally Laplace distributed by sampling these $s_j^{(i)}$ from the $\text{Exp}(\frac{2}{\lambda^2})$ prior, and then sampling the $b_{t,j}^{(i)}$ from zero-mean Normal distributions with variances $s_j^{(i)}$. Hence

$$P(b_{t,j}^{(i)}|s_j^{(i)}) = \frac{1}{\sqrt{2\pi s_j^{(i)}}}e^{-[b_{t,j}^{(i)}]^2/[2s_j^{(i)}]},$$

and so $\mathbf{b}_{:,j}^{(i)}$ has the same Normal distribution as $\Phi^{-1}\left\{F_{\mathcal{L}}[b_{t,j}^{(i)}]\right\}$ but with the variances and covariances scaled up by $s_j^{(i)}$, with $s_j^{(i)} \sim \text{Exp}(\frac{2}{\lambda^2})$. Therefore, also referring back to equation (3), it follows that

$$P(\mathbf{b}_{:,j}^{(i)}, s_j^{(i)}|\rho_j^{(i)}, \lambda) = \frac{\lambda^2}{2}e^{-\lambda^2 s_j^{(i)}/2}\frac{1}{(2\pi)^{T/2}[s_j^{(i)}]^{1/2}|\mathbf{\Sigma}_j^{(i)}|^{1/2}}e^{-\mathbf{b}_{:,j}^{(i)\top}[s_j^{(i)}]^{-1}[\mathbf{\Sigma}_j^{(i)}]^{-1}\mathbf{b}_{:,j}^{(i)}/2}.$$

To make sampling easier, at this stage we let $s_j^{(i)} = [\nu_j^{(i)}]^{-1}$, leading to the density

$$\begin{aligned} P(\mathbf{b}_{:,j}^{(i)}, \nu_j^{(i)}|\rho_j^{(i)}, \lambda) &= \frac{1}{[\nu_j^{(i)}]^2}\frac{\lambda^2}{2}e^{-\lambda^2/(2\nu_j^{(i)})}\frac{[\nu_j^{(i)}]^{1/2}}{(2\pi)^{T/2}|\mathbf{\Sigma}_j^{(i)}|^{1/2}}e^{-\mathbf{b}_{:,j}^{(i)\top}\nu_j^{(i)}[\mathbf{\Sigma}_j^{(i)}]^{-1}\mathbf{b}_{:,j}^{(i)}/2} \\ &= \frac{\lambda^2}{2}e^{-\lambda^2/(2\nu_j^{(i)})}\frac{[\nu_j^{(i)}]^{-3/2}}{(2\pi)^{T/2}|\mathbf{\Sigma}_j^{(i)}|^{1/2}}e^{-\mathbf{b}_{:,j}^{(i)\top}[\mathbf{\Sigma}_j^{(i)}]^{-1}\mathbf{b}_{:,j}^{(i)}\nu_j^{(i)}/2}, \end{aligned} \tag{8}$$

where the extra factor of $1/[\nu_j^{(i)}]^2$ is the factor $|d\{[\nu_j^{(i)}]^{-1}\}/d\nu_j^{(i)}|$ due to the change of variable. Assuming that the model will be fit to data standardised to have unit variance, we set the prior on the intercept as $a \sim \mathcal{N}(0,1)$, and we set the prior on the model precision as $\tau_i \sim \text{Gamma}(1,1)$ (which has prior mean 1, with 95% of the prior mass between 0.025 and 3.7, which we believe is reasonable for these data). Now combining equation (8) with these prior specifications, and $P(\rho_j^{(i)}|k) = \frac{k}{e^k-1}e^{k\rho_j^{(i)}}$ (for $0 \leq \rho_j^{(i)} \leq 1$), as well as with the model likelihood (equation (2)), we get:

$$P(\mathbf{x}_{:,i}, \mathbf{B}^{(i)}, \boldsymbol{\rho}^{(i)}, \boldsymbol{\nu}^{(i)}, a_i, \tau_i | \mathbf{X}_{:,\setminus i}, \lambda, k) = \left\{ \prod_{t=1}^{T} \prod_{k=1}^{n_t} \sqrt{\frac{\tau_i}{2\pi}} e^{-\tau_i \left(x_{t,i,k} - \mathbf{b}_{t,:}^{(i)} \cdot \mathbf{x}_{t,\setminus i,k}^{\top} - a_i\right)^2/2} \right\}$$

$$\frac{1}{\sqrt{2\pi}} e^{-\{\tau_i + a_i^2/2\}} \prod_{j=1}^{p-1} \left\{ \frac{k}{e^k - 1} e^{k\rho_j^{(i)}} \frac{\lambda^2}{2} e^{-\lambda^2/(2\nu_j^{(i)})} \frac{[\nu_j^{(i)}]^{-3/2}}{(2\pi)^{T/2} |\boldsymbol{\Sigma}_j^{(i)}|^{1/2}} e^{-\mathbf{b}_{:,j}^{(i)\top} [\boldsymbol{\Sigma}_j^{(i)}]^{-1} \mathbf{b}_{:,j}^{(i)} \nu_j^{(i)}/2} \right\}. \tag{9}$$

Following equation (9), posterior sampling for the model described in Sections 2.3 and 2.4 can be implemented through a Gibbs sampler with the steps given in Algorithm 1. We note that Algorithm 1 has a relatively low computational cost, because each of the steps (with the exception of step 4) involves sampling from a known distribution for which the parameters can be easily calculated. Then for step 4, we can simply use a slice-sampler to sample $\rho_j^{(i)}$, which has finite support $\rho_j^{(i)} \in [0,1)$. The full derivations of each step of Algorithm 1 appear in Supplement B.

ALGORITHM 1. *A Gibbs sampler with the following steps:*

1) **Sample:** a_i *from:* $P(a_i|\mathbf{x}_{:,i}, \mathbf{X}_{:,\setminus i}, ...) \propto f_{\mathcal{N}}(a_i|\mu_a, \sigma_a) = g_a(a_i)$,

where $f_{\mathcal{N}}$ *is the Normal density,* $\sigma_a^{-2} = 1 + n\tau_i$ *and* $\mu_a = \sigma_a^2 \tau_i \sum_{t=1}^{T} \sum_{k=1}^{n_t} \{x_{t,i,k} - \mathbf{b}_{t,:}^{(i)} \cdot \mathbf{x}_{t,\setminus i,k}^{\top}\}$.

2) **Sample:** τ_i *from:* $P(\tau_i|\mathbf{x}_{:,i}, \mathbf{X}_{:,\setminus i}, ...) \propto f_\gamma(\tau_i|k_\tau, \theta_\tau) = g_\tau(\tau_i)$,

where f_γ *is the density of the gamma distribution with* $k_\tau = 1 + \frac{\sum_{t=1}^{T} n_t}{2}$ *and*

$$\theta_\tau = 1/\{1 + \sum_{t=1}^{T} \sum_{k=1}^{n_t} (x_{t,i,k} - \mathbf{b}_{t,:}^{(i)} \cdot \mathbf{x}_{t,\setminus i,k}^{\top} - a_i)^2/2\}.$$

3) **Sample:** $\nu_j^{(i)}$ *from:* $P(\nu_j^{(i)}|\mathbf{x}_{:,i}, \mathbf{X}_{:,\setminus i}, ...) \propto f_{IG}(\nu_j^{(i)}|\mu_\nu, \lambda_\nu) = g_{\nu_j}(\nu_j^{(i)})$,
where f_{IG} *is the density of the inverse Gaussian distribution with parameters*

$\lambda_\nu = \lambda^2$ *and* $\mu_\nu = \lambda \Big/ \sqrt{\mathbf{b}_{:,j}^{(i)\top} [\mathbf{\Sigma}_j^{(i)}]^{-1} \mathbf{b}_{:,j}^{(i)}}$.

4) **Sample:** $\rho_j^{(i)}$ *from:*

$$P(\rho_j^{(i)} | \mathbf{x}_{:,i}, \mathbf{X}_{:,\setminus i}, ...) \propto e^{k\rho_j^{(i)}} \frac{1}{|\mathbf{\Sigma}_j^{(i)}|^{1/2}} e^{-\mathbf{b}_{:,j}^{(i)\top} \nu_j^{(i)} [\mathbf{\Sigma}_j^{(i)}]^{-1} \mathbf{b}_{:,j}^{(i)}/2} = g_{\rho_j}(\rho_j^{(i)}).$$

5) **Sample:** $\mathbf{b}_{:,j}^{(i)}$ *from:* $P(\mathbf{b}_{:,j}^{(i)} | \mathbf{x}_{:,i}, \mathbf{X}_{:,\setminus i}, ...) \propto f_{\mathcal{N}}(\mathbf{b}_{:,j}^{(i)} | \widetilde{\mathbf{m}}_j^{(i)}, \widetilde{\mathbf{\Sigma}}_j) = \widetilde{g}_{\mathbf{b}_j}(\mathbf{b}_{:,j}^{(i)})$,

where $f_{\mathcal{N}}$ *is the multivariate Normal density,* $[\widetilde{\mathbf{\Sigma}}_j^{(i)}]^{-1} = \nu_j^{(i)} [\mathbf{\Sigma}_j^{(i)}]^{-1} + [\mathbf{V}_j^{(i)}]^{-1}$, *and* $\widetilde{\mathbf{m}}_j^{(i)} = \widetilde{\mathbf{\Sigma}}_j^{(i)} [\mathbf{V}_j^{(i)}]^{-1} \mathbf{m}_j^{(i)}$, *where the* t^{th} *element of the vector* $\mathbf{m}_j^{(i)}$ *is*

$$m_{t,j}^{(i)} = \sum_{t=1}^{T} \sum_{k=1}^{n_t} x_{t,j,k} \left\{ x_{t,i,k} - \mathbf{b}_{t,\setminus j}^{(i)} (\mathbf{x}_{t,\setminus i,k})_{\setminus j}^{\top} - a_i \right\} \Big/ \sum_{t=1}^{T} \sum_{k=1}^{n_t} x_{t,j,k}^2,$$

where $\mathbf{b}_{t,\setminus j}^{(i)}$ *and* $(\mathbf{x}_{t,\setminus i,k})_{\setminus j}$ *represent* $\mathbf{b}_{t,:}^{(i)}$ *and* $\mathbf{x}_{t,\setminus i,k}$ *without the* j^{th} *elements, respectively, and* $\mathbf{V}_j^{(i)}$ *is a diagonal matrix, with the* t^{th} *diagonal element equal to* $1/\{\tau_i \sum_{t=1}^{T} \sum_{k=1}^{n_t} x_{t,j,k}^2\}$.

3. Simulation study. In this section, we present the results from a simulation study, to test how accurately our model can recover network structure which we know in advance. We generate simulated data with structure that we expect to be typical of real data (Nowakowski et al., 2017; Mayer et al., 2019), and then fit the proposed model to the simulated data. To generate the data, the observations $x_{t,i}$ for each node i are generated such that they follow a mean time-series of one of four types (illustrated in Figure 5), as follows:

(a) Monotonic; decreasing to no signal.
(b) Monotonic; increasing from no signal.
(c) Maximum: increasing from and decreasing to no signal.
(d) Null: random noise.

Types (a) and (b) represent node-types of interest to the biological setting, as follows. Type (a) corresponds to genes that are activated (i.e., $x_{t,i} > 0$) early in the time-series before becoming de-activated (as we would expect of genes which are important for stem-like cell identity). Type (b) corresponds to genes which only become activated later in the time-series (as we would expect of genes which are important for the identity of mature cells, such as neurons). Types (c) and (d) make the simulated data closer to what we would expect of the real data, by mixing in nodes with other sorts of signals: type (c) corresponds to genes which are active in the middle of the time-series only, and type (d) are null nodes (with random activation).

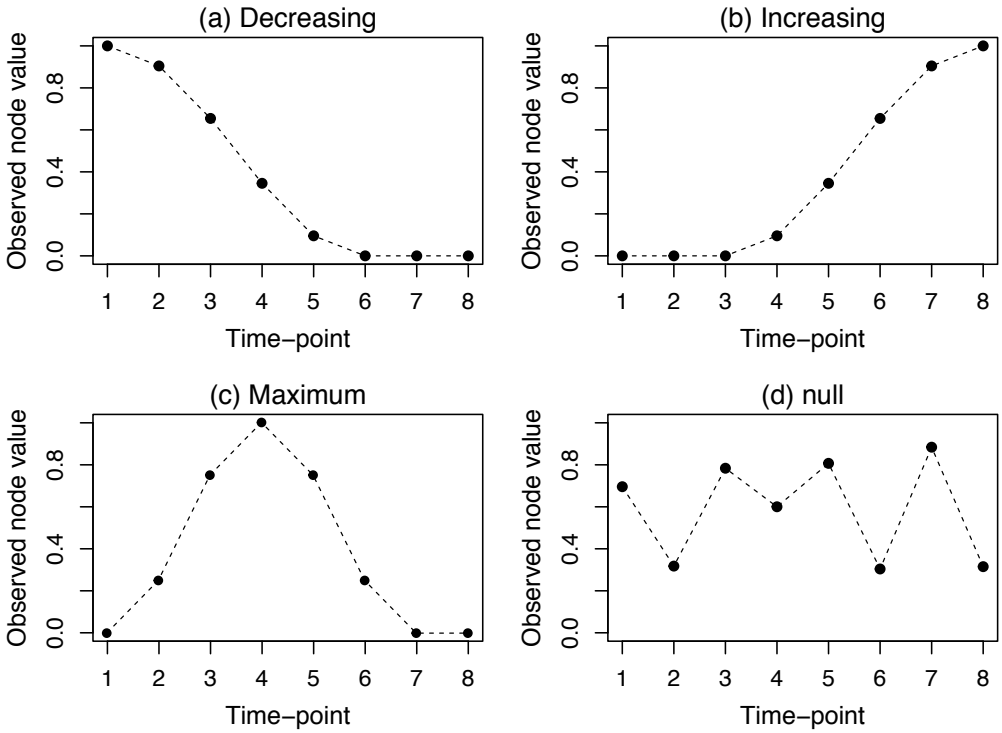

Fig 5: Simulated time-series of sampled observations at four types of network nodes.

After generating each characteristic mean time-series according to fixed types (a)-(d), we then time-stretch the particular characteristic mean time-series chosen for each node i by a random amount. We do this to reflect the fact that developmental events (as represented by gene-expression measurements) occur at different times in different cells. We achieve this effect by changing the length of the time-series to have a random but uniformly-distributed period $T' \sim \mathcal{U}[T-3, T]$, before zero-padding to return the time-series to its original period T. This gives a mean profile $x_{t,i}$, $t = 1, ..., T$ for each node that is distinct from all other nodes of the same type. We then generate observations $x_{t,i,k}$ for each node according to equation (1) based on these mean profiles, also setting the intercept parameter a_i to 0, and sampling via the Markov chain described in Algorithm 2:

ALGORITHM 2. *A Markov chain:*
Loop*: t in $1:T$*
  $X_{t,:,1} \leftarrow 0$ *// Initialize Markov chain at 0*
  Loop*: r in $2:R$*
    $\mathbf{S} \leftarrow \mathbf{X}_{t,:,r-1}$
    Loop*: i in $1:p$*

Sample: $S_i \sim N(\mathbf{S}_{\setminus i}.\mathbf{b}^{(i)}_{t,:}, \sigma^2)$
end loop
$\mathbf{X}_{t,:,r} \leftarrow \mathbf{s}$
end loop
end loop

where $\mathbf{X}$ is a $T \times p \times R$ array containing the sampled data, $\mathbf{S}$ is a vector of length p which temporarily stores intermediate results, and the elements of $\mathbf{b}^{(i)}$ are specified as:

$$b^{(i)}_{t,j} = \begin{cases} 1/p', & \text{if nodes } i \text{ and } j \text{ are of the same type} \\ 0, & \text{otherwise} \end{cases}$$

where p' is the number of nodes j of the same type as i. The number of MCMC samples in the Markov chain specified in Algorithm 2 is given by the variable R: we use $R = 10^4$, and after thinning to take one sample in every 100, we choose the final 25 (thinned) samples to pass forward to the model fitting after adding the mean characteristic profiles. That is, we have 25 samples per time-point, i.e., $n_t = 25$, where autocorrelation analysis and an experimentation with burn-in times show no evidence against them being independently and identically distributed at each time-point group. We note that in the simulation we specify means that vary with time but our model has constant mean. In practice this does not make any difference as the data are always standardised before model fitting, but the user can easily make the intercept time-varying if this is a concern. The procedure for generating the simulated data is also illustrated in Figure 6.

We generate each time-series with $T = 8$, $n_t = 25$ (constant for all values of t), and $p' = 10$, adding noise with standard deviations $\tau_i^{-1/2} = \tau^{-1/2} \in \{0.1, 0.2, 0.3\}$. Then, we apply Algorithm 1, and calculate each $\hat{b}^{(i)}_{t,j}$ from the median of the corresponding posterior. We infer an edge between nodes i and j if $\hat{b}^{(i)}_{t,j} \neq 0$, after thresholding the $\hat{b}^{(i)}_{t,j}$ to remove trivially small values, i.e., if $|\hat{b}^{(i)}_{t,j}| \geq \phi$. We generate ROC (receiver-operator characteristic) curves as this threshold ϕ is decreased to 0 from $\max|\hat{b}^{(i)}_{t,j}|$ (for $t \in \{1, ..., T\}$ and all j). We generate these curves from the true-positives (TP) and false-positives (FP) which we calculate from the ground-truth network edges $b^{(i)}_{t,j}$ and estimated network edges $\hat{b}^{(i)}_{t,j}$ as follows:

$$|\hat{b}^{(i)}_{t,j}| > 0 \text{ for } |b^{(i)}_{t,j}| > 0 \implies \text{TP}$$
$$\text{and} \quad |\hat{b}^{(i)}_{t,j}| > 0 \text{ for } |b^{(i)}_{t,j}| = 0 \implies \text{FP}$$

We generate an average ROC curve over 1000 repetitions of this procedure, and then calculate an AUC (area under curve) statistic for this average ROC curve.

We assess the performance of our full model using both sparsity within and sparsity across time, compared with the scenarios when one of these priors is excluded from the

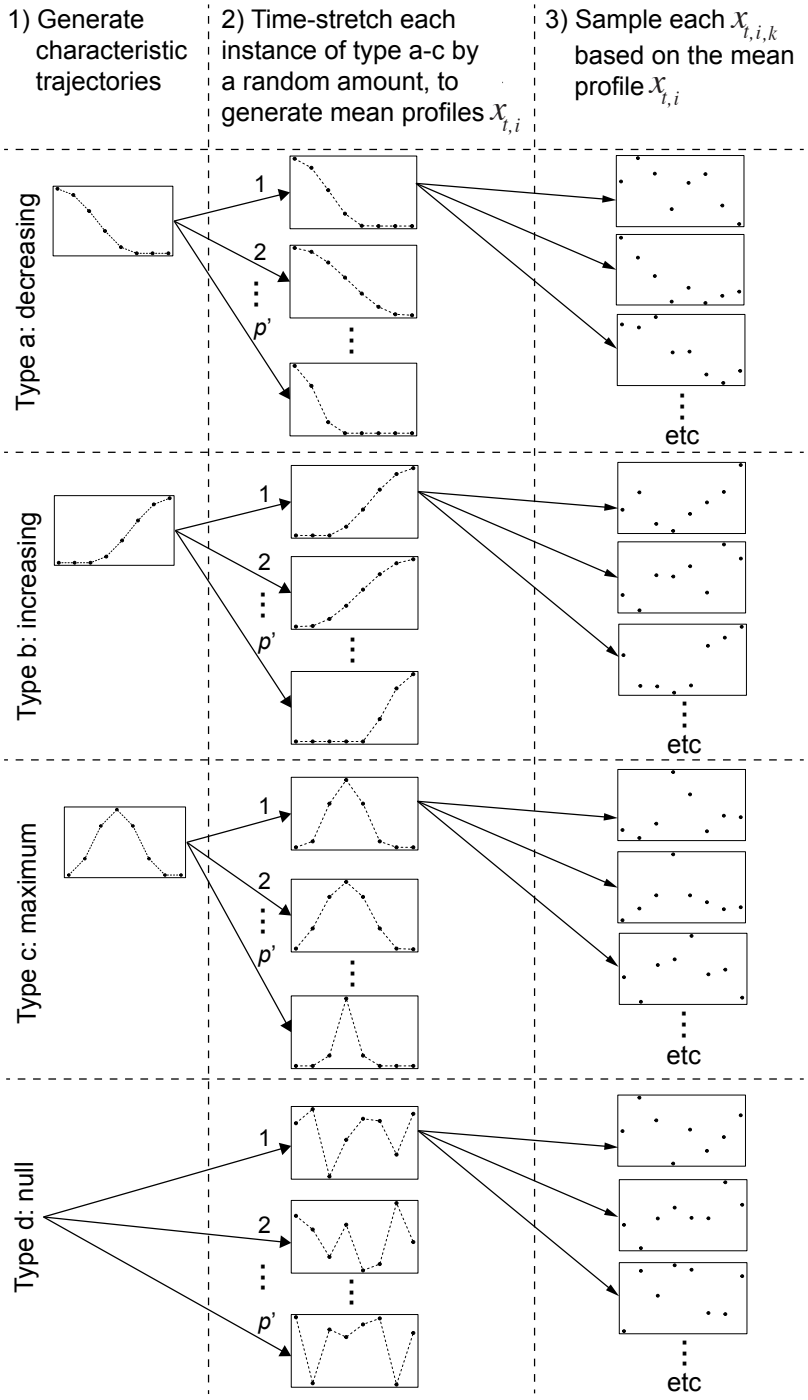

Fig 6: Overview of the procedure for generating the simulated data.

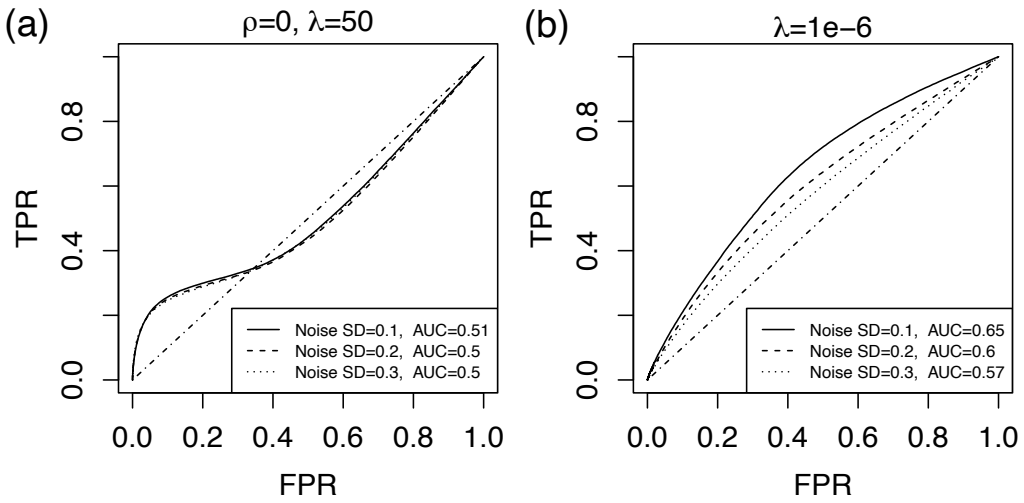

Fig 7: Accuracy of network inference by the model without either sparsity within, or across, time. (a) Model performance when sparsity across time is removed. (b) Model performance when sparsity within time is removed. Abbreviations: TP, true positives; FP, false positives.

model. To exclude sparsity across time, we enforce $\rho = 0$, and to exclude sparsity within time, we use $\lambda \to 0$: these results are shown in Figure 7. With $\rho = 0$, there is no correlation of the model parameters across time, and so we see the effect of inferring the networks separately for each time-point; i.e., sparsity across time is removed: in this case, AUC $= 0.5$ indicates that none of the intended structure in the data is being detected. Alternatively, as $\lambda \to 0$, the prior becomes flat or uninformative, and so in this case we see the effect of fitting the model without the sparsity within time. We again note that decoupling these types of sparsity is made possible by design with the model structure we propose, unlike alternatives such as Laplace-NEG (Shimamura et al., 2016). Then for the full model (which includes the priors to enforce both the sparsity within time and across time), we repeated the simulation for various values of sparsity parameter λ: Figure 8 shows the results (with hyperparameter $k = 20$, equivalent results with $k = 10$ and $k = 50$ are shown in Figures S10 and S11 in Supplement D). When we include the priors for both sparsity within and across time, we can achieve AUC of 0.9 or more, as long as the sparsity parameter λ is large enough. This result demonstrates that our priors are responsible for good detection of network edges with respect to the ground-truth in these simulated data. We also found that these results were not very sensitive to p', the number of covariates included in the simulated data. Figure S12 shows equivalent results to Figure 8, except with the number of covariates halved to $p' = 5$. In this case we found that the network inference is a bit more accurate, as would be expected with a smaller number of variables to predict; although the difference is minimal as long as the sparsity is great enough.

Dropouts, or missing values, are a well known source of technical noise in single-cell

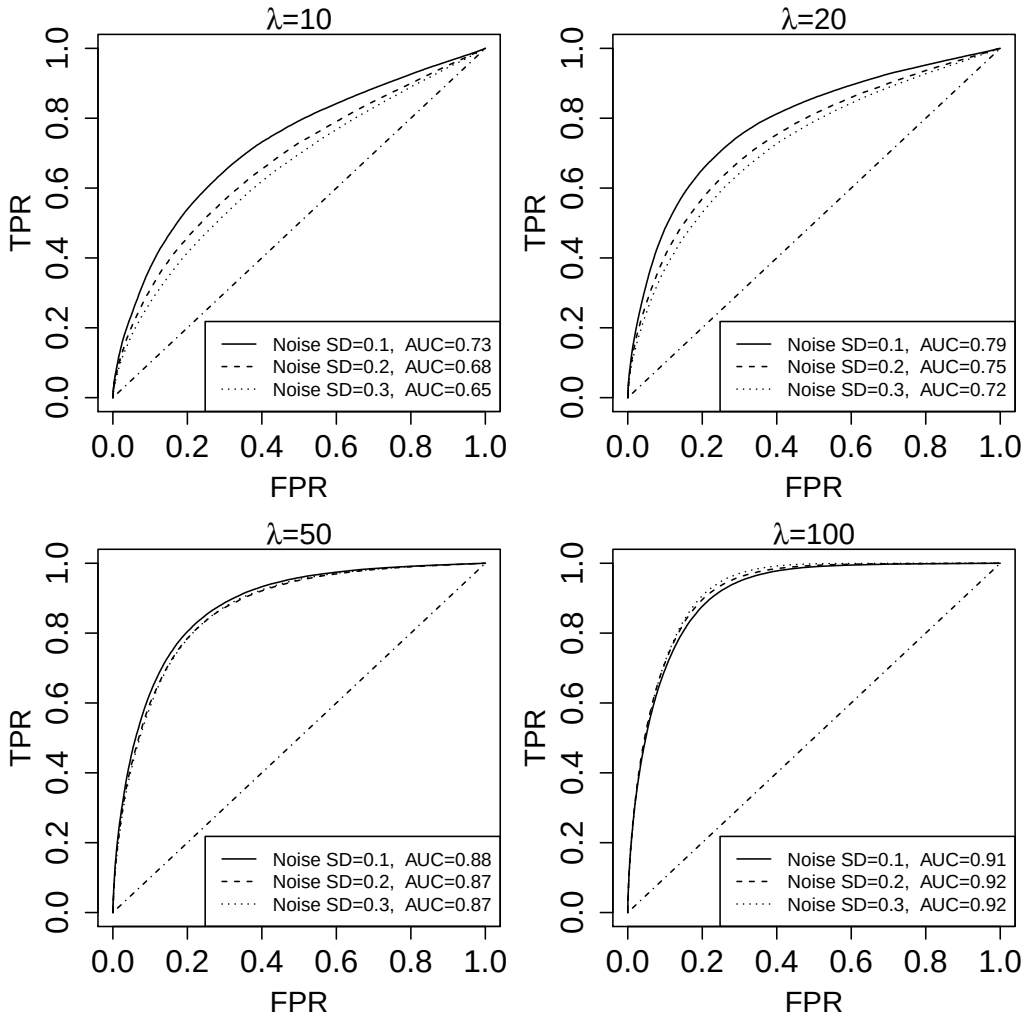

Fig 8: Accuracy of network inference by the model with both sparsity within and across time. Abbreviations: TP, true positives; FP, false positives.

transcriptome data. These missing values are replaced by zeros, leading to 'zero inflation'. A characteristic of this dropout effect is that data-values which are already small are more likely to drop out (i.e., get missed out), than values which are larger in magnitude. This is data missing not-at-random, an effect that can be challenging to model (Kharchenko, Silberstein and Scadden, 2014). Dropout rates (i.e., the proportion of data-values missing from the data-set) are often over 60% in typical single-cell transcriptome data-sets that we have seen, such as the one analysed in Section 4. To test the robustness of our method to dropouts, we used a well known and effective model of the dropout effect, published

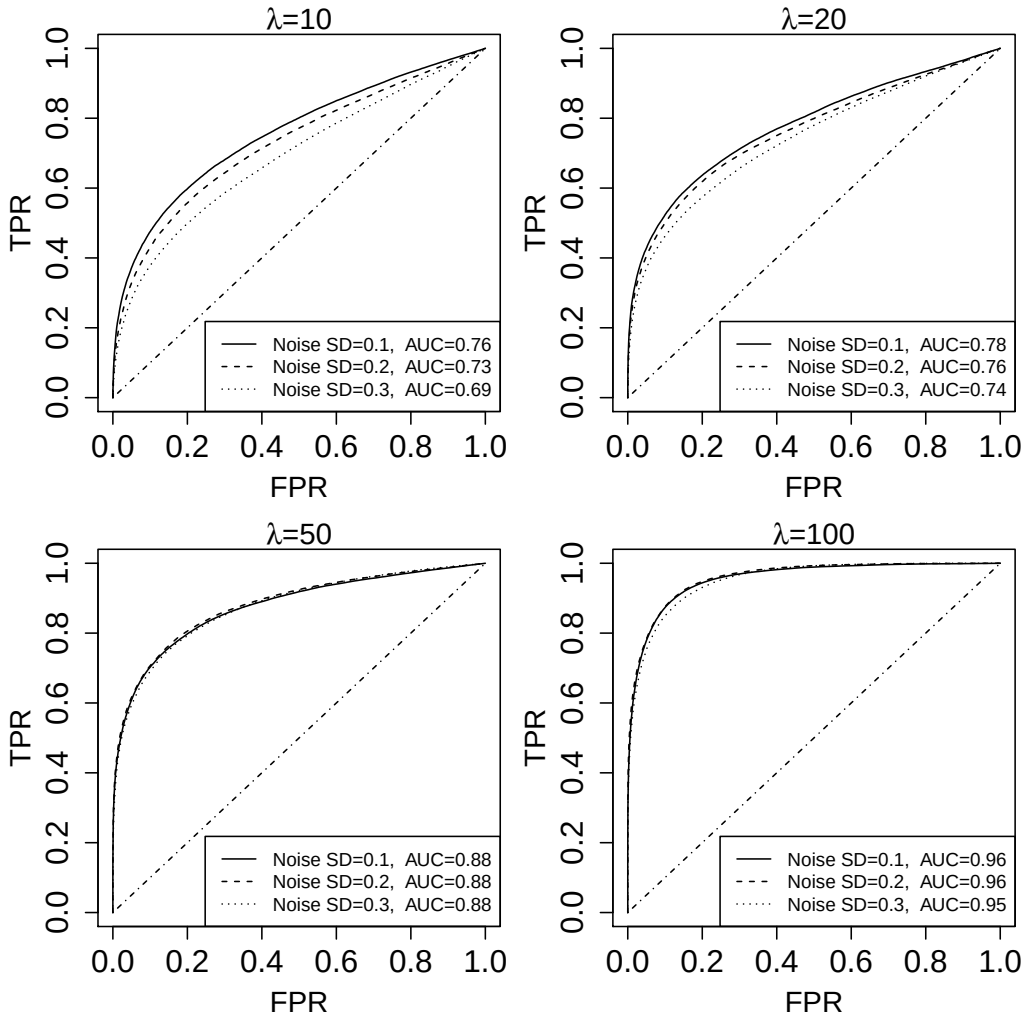

Fig 9: Accuracy of network inference in the simulation study with 60% dropouts (with $k = 20$). Abbreviations: TP, true positives; FP, false positives.

previously by Pierson and Yau (2015). This model of dropouts specifies the probability of an observed data-value dropping out as

$$p_{t,i,k} = \exp(-\omega \tilde{x}^2_{t,i,k}), \tag{10}$$

where the parameter ω controls the dropout rate (decreasing ω increases the number of dropouts), and $\tilde{x}_{t,i,k}$ is the data-value that would have been present without the dropout effect. This is essentially a hurdle model, with $h_{t,i,k} \sim \text{Bernouilli}(p_{t,i,k})$, so that the observed data $x_{t,i,k}$ (i.e., with dropouts included) is modelled as $x_{t,i,k} = h_{t,i,k} \cdot \tilde{x}_{t,i,k}$. We found that under this model, $\omega = 2$ leads to a dropout rate of around 66% in data-sets generated

according to the data-simulation procedure presented earlier in this section. We used this value of $\omega = 2$, and repeated our simulation study now with the addition of this dropout effect, carrying out out the same ROC-curve analysis as before. The results of this analysis (again with $\lambda = 50$ and $k = 20$) are shown in Figure 9: we found that our method is quite resilient to dropouts, with only a moderate decrease in performance compared to the results shown in Figure 8. The time-varying aspect of the model apparently helps to maintain performance when many dropouts are present, because when some values in the time-series are missing, sparsity across time encourages interpolation over the missing values. Interestingly, in very sparse cases, the dropout effect may even be helpful, possibly via a de-noising mechanism, as follows. Referring to the generative model of equation (10), it's clear that the dropouts mostly take place for small values of $x_{t,i,k}$. As these are much more likely to correspond to noise than larger values do, this leads to a strong de-noising effect. Finally, we note that to include a hurdle model or dropout effect in a model likelihood such as the one proposed in Section 2.3 would result in a much more computationally intensive model fitting procedure than the one we propose in our Algorithm 1 of Section 2.5.

4. Single-cell gene-expression data. In this section we present an example application of our proposed methodology, to single-cell gene-expression data. These data have been published previously by Nowakowski et al. (2017), and are publicly available from the NCBI database of genotypes and phenotypes (dbGaP), under accession number phs000989.v3 In this context, $x_{t,i,k}$ represents the log-expression of gene i, defined as log(transcript counts + 1), in sample k from time t. For the pseudo-time assignments for each cell, we use cell-type classifications provided with the data, together with an ordering for these cell-types according to the developmental lineage (for full details see Supplement C). We fitted the model to $n = 1557$ cell samples, and $p = 22988$ genes/nodes, reduced to $p = 212$ for each individual model fit by variable screening. For the fitting we used values of $\lambda = 20$ and $k = 1$: these values were chosen by grid-search stochastic EM (Figure S13 in Supplement D). Fitting the model as described, we obtained posterior distributions for each model parameter $b_{t,j}^{(i)}$, and we used the posterior medians as posterior summaries, $\hat{b}_{t,j}^{(i)}$. To fit each model, we ran the Gibbs' sampler proposed in Algorithm 1 for 1×10^4 samples (after 1×10^3 samples burn-in), which took 1.4 hours for each target-node on one core of a Macbook Pro laptop (mid 2015, 2.8 GHz, 16GB RAM).

The model was fitted initially to a panel of 25 genes, as target-nodes: these genes were chosen in an unbiased way by searching the biological sciences literature for genes that are important in this biological setting, and then analysing those that were present in this data-set after quality control. Estimated model parameters $\hat{b}_{t,j}^{(i)}$ for a selection of these genes are shown in Figure 10, and the full panel is shown in Figures S14 and S15.

We carried out Geweke (Geweke et al., 1991) and Heidel (Heidelberger and Welch, 1981) convergence tests, using the `R` package `CODA` (Plummer et al., 2006), for the sampler outputs for all the parameters $b_{t,j}^{(i)}$ shown in Figures 10, S14 and S15. These convergence test

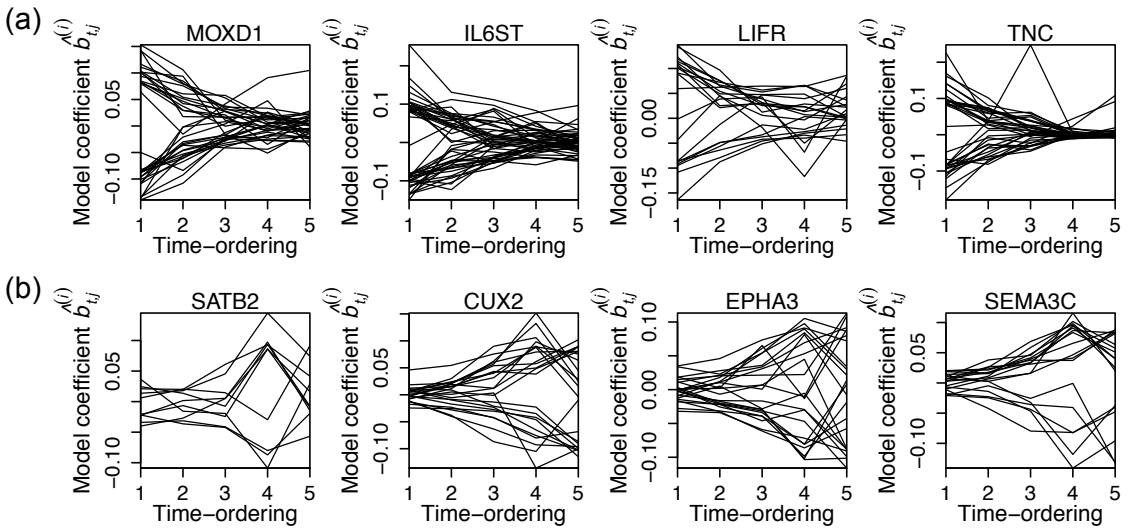

Fig 10: Inferred model parameters $\hat{b}^{(i)}_{t,j}$, for genes characteristic of: (a) stem-cells; (b) mature cells (neurons). Non-zero parameters $\hat{b}^{(i)}_{t,j}$ infer the local network structure around gene/node i. Parameters which are zero for every time-point are not plotted.

results appear in Figure 11. In convergence tests such as these, if an individual p-value is significantly small, it can be taken as evidence that the chain has not yet converged. Hence, the uniform distributions of p-values shown in Figure 11, in which these p-values are aggregated over all the test results, indicate that the MCMC sampler has converged for these target-nodes. Then, to give an indication of how 'stiff' or 'sloppy' these parameters are, we estimated the standard-deviations of these posterior distributions for this panel of genes: these are plotted against the corresponding posterior averages in Figure 12. These posterior standard deviations are typically much smaller in magnitude than the posterior averages, demonstrating that the posteriors are not 'sloppy', and indicating that the estimates from our model are reliable. We also wanted to make sure that our results are not driven by a few outlier cells. So we repeated the inference for this same panel of genes, but now using only a random sample of 50% of the cells originally used, i.e., $n = 779$. The results of this analysis are plotted in Figure S16, for the same genes as are shown in Figure 10. The results shown in these figures are clearly very similar, and therefore we conclude that our results here are not driven by outliers.

Figure 10a shows inferred model parameters $\hat{b}^{(i)}_{t,j}$, for a selection of nodes/genes which are characteristic of stem cells, and of neurons (i.e., mature cells), selected from the full panel of 25 genes. We expect stem cells to predominate at earlier times, and hence we expect to see decreasing time-series for genes which are characteristic of this type of cell. On the other hand, we expect mature cells such as neurons to predominate at later times, and

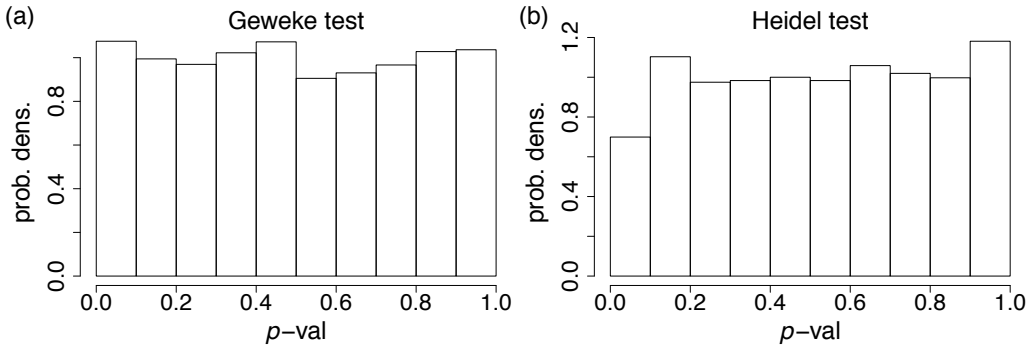

Fig 11: Convergence-test results, for the parameters $b_{t,j}^{(i)}$ which appear in Figures 10 and S14 and S15.

so we expect to see increasing time-series for genes characteristic of this type of cell. As would be expected for stem-cell genes, important model parameters $b_{t,j}^{(i)}$ tend to decrease in magnitude during the developmental trajectory as cells go from stem-cell to mature cell types (e.g., gene transcript MOXD1). Figure 10b then shows, as would be expected, that important model parameters $\hat{b}_{t,j}^{(i)}$ become non-zero (corresponding to network edges appearing) late in the developmental trajectory, when the cells become neurons and hence their characteristic gene regulatory program is activated (e.g., for SATB2). Equivalent results to Figure 10a-b for the full panel of 25 genes analysed then appear in Figures S14 and S15 respectively in Supplement D. In these figures, we also see similar results: for genes that tend to be active in stem-cells, model parameters $b_{t,j}^{(i)}$ tend to decrease in magnitude during the developmental trajectory as cells go from stem-cell to mature cell types (Figure S14), and *vice-versa* for genes which are important to mature cells such as neurons (Figure S15).

We wish to infer a network edge between nodes i and j if $|\hat{b}_{t,j}^{(i)}| > 0$. We estimate these $\hat{b}_{t,j}^{(i)}$ from the posterior medians, but because we find that many of these medians are close to, but not exactly zero, we set $\hat{b}_{t,j}^{(i)}$ to zero in such cases by thresholding. Therefore, we infer 'no edge' between nodes i and j when the posterior median is close to zero. Hence, if (and only if) $|\hat{b}_{t,j}^{(i)}| > \phi$, where ϕ is the threshold parameter, we would infer a network edge between nodes i and j at time t (for the model fit around node i). We note that the local model fitting (equation (1)) does not depend on this network estimation. Hence, this thresholding can take place independently of the computationally-intensive MCMC sampling. Thus, we leave ϕ as a tuning parameter, which can be varied by the user in real time to interpret results, equivalently to changing the resolution or granularity in a visualisation. We recommend the user does a full sweep through $\phi \in [0, \infty]$ to interpret the results. We also note that if $|\hat{b}_{t,i}^{(j)}| > \phi$ (for the independent model fit around node j rather than node i), we would independently infer an edge between nodes i and j at time

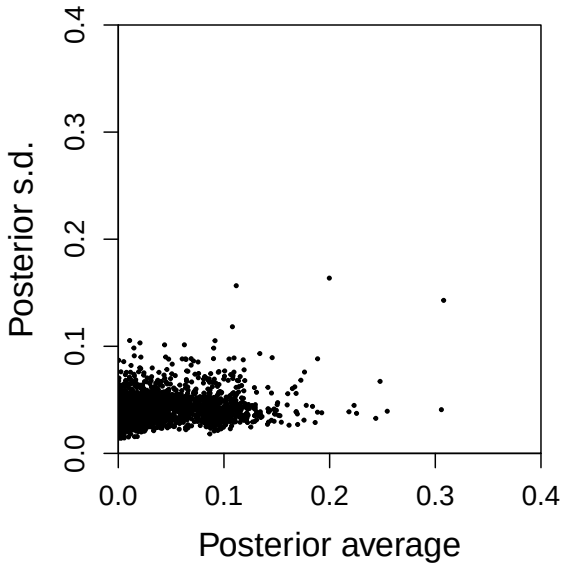

Fig 12: Estimates of the spread of the posterior distributions for the parameters $b_{t,j}^{(i)}$ which appear in Figures 10 and S14 and S15.

t. Thus, some inconsistency may arise, due to these independent model fits around nodes i and j. To deal with this, we use the 'min_symmetrisation' scheme of Kolar et al. (2010), inferring an edge between nodes i and j at time t, i.e., $\hat{A}_{i,j,t} \neq 0$, if and only if $|\hat{b}_{t,j}^{(i)}| > \phi$ and $|\hat{b}_{t,i}^{(j)}| > \phi$.

Plots of the inferred network structure around an example of a gene shown in Figure 10b, namely SATB2, are shown in Figure 13, after 'min_symmetrisation' (Section 2.2) with $\phi = 0.05$. In addition to the neuronal identity gene SATB2 (Alcamo et al., 2008), several of the genes shown in Figure 13 are already known to be important in neuronal development, including NEUROD1 which initiates the programme of neuronal development (Pataskar et al., 2016) and RUNX1T1 which regulates the differentiation of neurons from neural stem cells (Linqing et al., 2015), as well as the neuronal circuit-formation gene BCL11A (John et al., 2012). Intriguingly, this network structure also includes MIR133A1HG and LINC00478, which are (respectively) examples of micro-RNA (miRNA) and long non-coding RNA (lncRNA). Non-coding RNA transcripts such as these do not get translated into proteins, as would usually be the case for a transcript from a region of DNA which codes for a gene. Instead, non-coding RNA transcripts are known to play an important role in gene regulation (Cech and Steitz, 2014). However, we still only understand a small amount about their function, and gene regulation involving these sorts of non-coding RNA is an important research topic. We note that MIR133A1HG and LINC00478 are promising candidates for for further experimental investigation which have been identified using our proposed methodology.

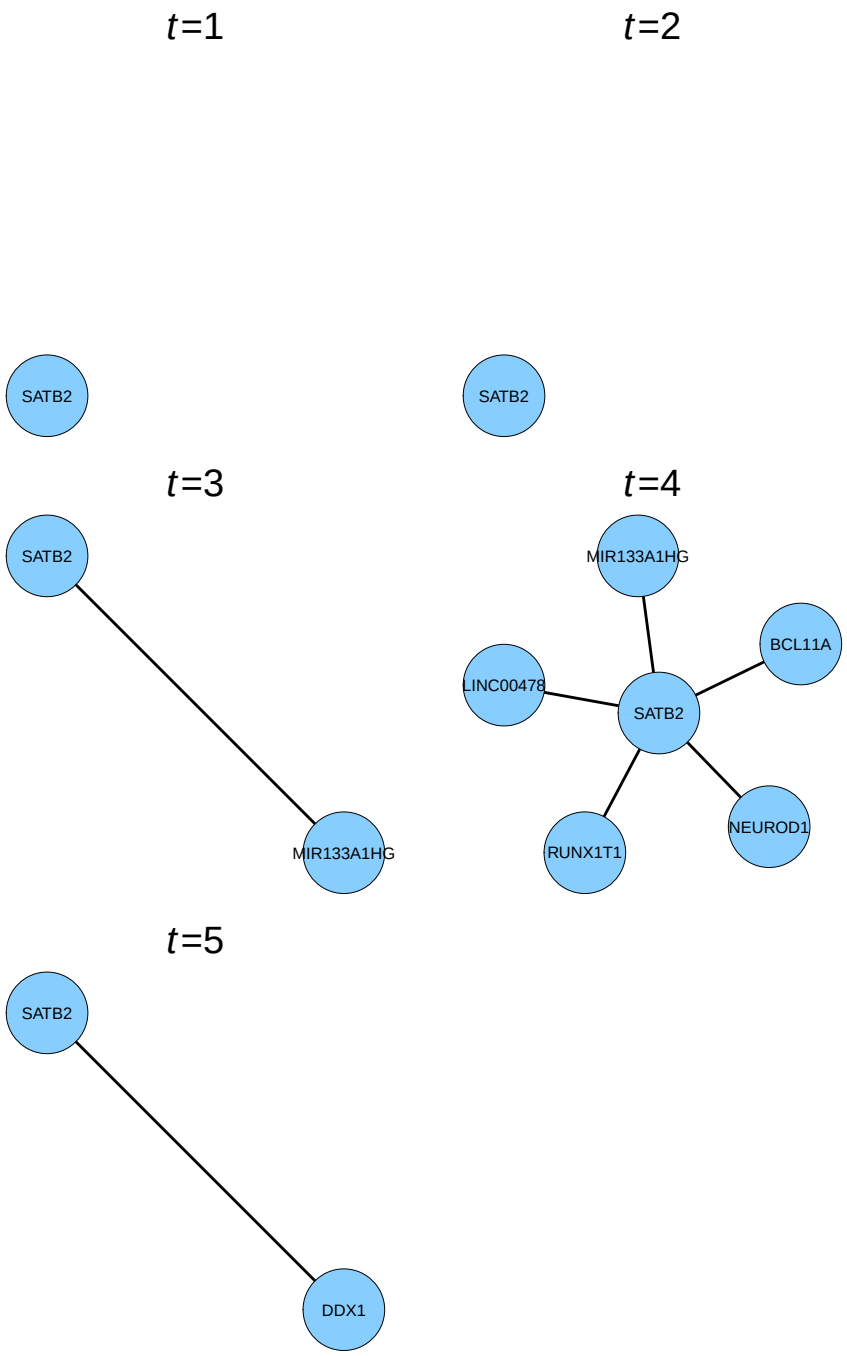

Fig 13: Time-varying network structure inferred around the gene SATB2. This gene is characteristic of certain types of neuron, and hence we would expect network structure to appear at later times, when the cell type-specific gene regulatory program becomes activated.

Our method	SCENIC
BCL11A	ARPP21
DDX1	CHL1
LINC00478	KIAA1598
MIR133A1HG	MEF2C
NEUROD1	NFIA
RUNX1T1	RUNX1T1

TABLE 1

Comparison of nodes inferred in static network structure by our method, and the SCENIC method.

Next, we compared our method with alternative network inference method for single-cell transcriptome data. Methods for inferring time-varying network structure in data of this type include alternatives designed for many fewer nodes than our method can handle, such as the work of Matsumoto et al. (2017), which is designed to infer structure in networks with fewer than 100 nodes. However, there is also a static network inference method available for single-cell transcriptome data called 'SCENIC' (Aibar et al., 2017), that can be used to infer structure in large networks of 20000 or more nodes, and that is therefore also appropriate for the data-set analysed here. For comparison with our proposed methodology, we ran the SCENIC method on the same single-cell gene-expression data-set already analysed. Equivalently to our proposed method, SCENIC returns fitted model parameters that indicate the strength of the network connection between a pair of nodes, or genes: maintaining equivalent notation, we label these SCENIC model parameters $b_j^{\prime(i)}$. Thus, by again choosing a threshold ϕ', it is possible to infer network structure by inferring edges between the pair of nodes i and j if the corresponding fitted model parameter $|\hat{b}_j^{\prime(i)}| > \phi'$. We choose ϕ' so as to maintain the same number of connections to each target in the network inferred by the SCENIC method, as compared with our method. Table 1 shows the genes inferred in the network structure around the neuronal identity gene SATB2 (summarised from Figure 13), together with the genes equivalently found from the SCENIC method (setting ϕ' to maintain the same number of connections).

Of the genes shown in Table 1, just as with those found by our method, those found by the SCENIC method are mostly already known to be involved in neural development, as follows. ARPP21 is involved with branching of dendrites (Rehfeld et al., 2018), CHL1 and KIAA1598 are thought to be involved in neuronal migration and axon formation (Alsanie et al., 2017; Toriyama et al., 2006), and MEF2C and NFIA are known to be important for neural stem and progenitor cell differentiation (Li et al., 2008; Piper et al., 2010). However, we note that the SCENIC method is not able to infer time-varying network structure, as our method can: to make the comparison shown Table 1, the time-varying aspect of the network structure inferred by our method had to be 'flattened out'.

It is challenging to visualise in a meaningful way the entire structure of a large network, such as the full genome-wide network inferred here, if it is inferred for all 22989 nodes. This challenge becomes even larger when the dimension of time is added. After fitting the

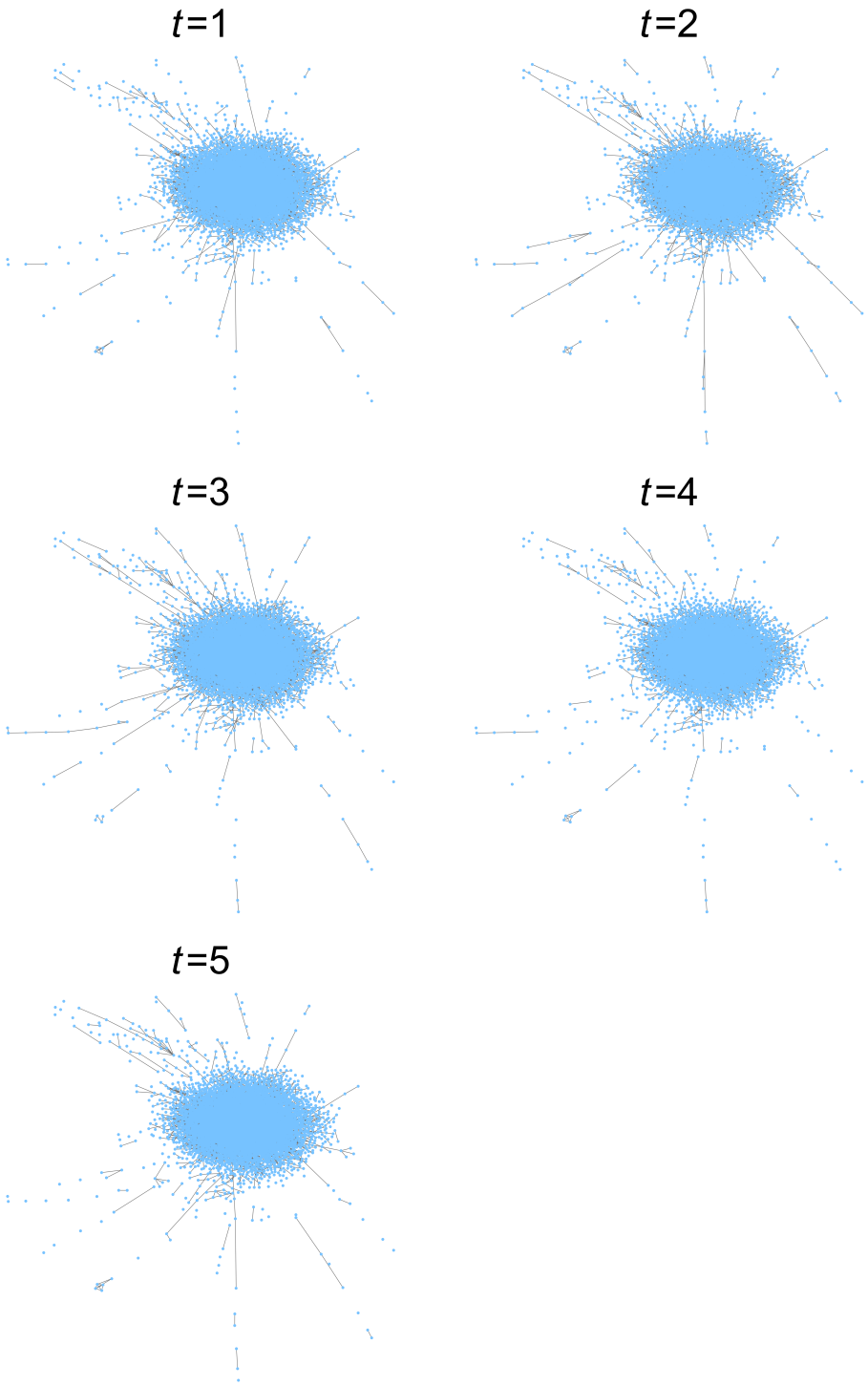

Fig 14: Time-varying network structure of the fully-connected component (11133 nodes) of the inferred genomic network.

model to all 22989 target-nodes on a high-performance computing cluster, we inferred the structure of this network, and found the fully connected component (11133 nodes), which is shown in Figure 14. As a minimum, it can be seen from this figure that the network structure changes gradually rather than suddenly with time, as we would expect from our proposed methodology.

5. Discussion. In this paper, we have proposed a new model to infer time-varying network structure. This model makes use of a novel prior structure we introduce here, which extends the Bayesian lasso to the time-varying case. The novel structure of this prior allows for effective modelling of time-varying network structure even in situations where there are very few time-points, as is typical in cell-biological (i.e., 'omics) data. We also found that the model fitting and inference procedure we have proposed works well even in with large networks of over 20000 nodes, which compares very well with alternatives (see for example the work by Matsumoto et al. (2017)).

We used simulated data to assess the ability of the proposed model to accurately infer time-varying network structure, and we showed that the model is effective in inferring time-varying genomic network structure from single-cell gene-expression data. However, we note that genomic network structure which is inferred from only gene-expression data (as we do here) is not guaranteed to correspond to true gene regulatory patterns. To strengthen any belief that the inferred genomic network structure corresponds to true gene regulatory patterns rather than simply gene co-expression patterns, evidence from, for example, chromatin binding and epigenomic data could also be incorporated into the model (Novershtern, Regev and Friedman, 2011). We intend to incorporate such data as the next stage of the development of this model. Specifically, we will do this by allowing the sparsity parameter λ to vary for each pair of nodes i and j, depending on any prior evidence of a physical interaction between the protein-product of gene j with the DNA or surrounding chromatin of gene i.

Another characteristic of the single-cell transcriptome data analysed here is that the data are zero-inflated. This is a case of data missing-not-at-random, because the dropout events which lead to the extra zeros in the data are more likely to occur when the true transcriptome level is low (Kharchenko, Silberstein and Scadden, 2014). As part of the next stage of the development of this model, we intend to account for dropouts as other authors have done (van Dijk et al., 2017), for example by explicitly including the dropout events in the model likelihood (Pierson and Yau, 2015). We also note that existing time-inference methods for data such as those presented here are algorithmic, rather than model-based. Hence it is not easy to obtain uncertainties on the inferred times when using these methods. Thus, we would like to develop a model-based time-inference method that will provide such uncertainties, and then feed these uncertainties directly into the time-varying network model we have proposed. We also note that in other contexts, it could complicate the analysis if there is uneven time-sampling. For example, if we expect highly deterministic behaviour with little noise, but have data with time-sampling at known but uneven time-

points, the method might need to be adapted. Specifically, in that context we would expect to see larger changes in parameters over larger time-intervals: this structure is not explicitly captured by our model, in its current form.

Understanding interactions between genes and their transcriptional regulators is a fundamental question in genomics, and network models are a natural way to represent and analyse groups of interactions between genes and their regulators. Biomedical science in the high-throughput genomic age has been developing ever more innovative ways to collect increasingly vast quantities of data. However, the statistical techniques to represent, analyse and interpret such data still lag behind the means to generate them. In particular, there is currently a lack of good computational statistical methodology to represent and analyse changes in gene-regulatory interactions as cells are specified and change state - an issue we address with the time-varying network model that we propose here. The computational-statistical tools that we are developing allow novel characterisation of genomic interactions in important settings, adding to knowledge of fundamental biological principles, and motivating further investigation by targeted experiments.

Acknowledgements. We are grateful to Aaron Diaz, Tom Nowakowski, Alex Pollen, and Aparna Bhaduri, for helpful discussions, insightful comments, and useful advice throughout this project, and for providing early access to the data. The work of the first author was supported by the MRC grant MR/P014070/1. The work of the second and third author was partially supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1.

SUPPLEMENTARY MATERIAL

Supplement: Supplementary Information
(Appears below). Derivations; Details of data pre-processing; Supplementary Figures S1-S14.

Supplement: Software
(Online respository). An R package containing an efficient implementation of the model proposed in this paper can be installed in R by typing: `install.packages("devtools")` and then: `devtools::install_github("tombartlett/SBDN")`
This package contains an R function which calls a C++ implementation of the Gibbs sampler described in Algorithm 1.

References.

Aibar, S., González-Blas, C. B., Moerman, T., Imrichova, H., Hulselmans, G., Rambow, F., Marine, J.-C., Geurts, P., Aerts, J., van den Oord, J. et al. (2017). SCENIC: single-cell regulatory network inference and clustering. *Nature methods* **14** 1083.

Alcamo, E. A., Chirivella, L., Dautzenberg, M., Dobreva, G., Fariñas, I., Grosschedl, R. and McConnell, S. K. (2008). Satb2 regulates callosal projection neuron identity in the developing cerebral cortex. *Neuron* **57** 364–377.

Alexander, R. P., Kim, P. M., Emonet, T. and Gerstein, M. B. (2009). Understanding modularity in molecular networks requires dynamics. *Science signaling* **2** pe44.

Alsanie, W., Penna, V., Schachner, M., Thompson, L. and Parish, C. (2017). Homophilic binding of the neural cell adhesion molecule CHL1 regulates development of ventral midbrain dopaminergic pathways. *Scientific reports* **7** 9368.

Andrews, D. F. and Mallows, C. L. (1974). Scale mixtures of normal distributions. *Journal of the Royal Statistical Society. Series B (Methodological)* **36** 99–102.

Carvalho, C. M., Polson, N. G. and Scott, J. G. (2010). The horseshoe estimator for sparse signals. *Biometrika* **97** 465–480.

Castillo, I., Schmidt-Hieber, J., Van der Vaart, A. et al. (2015). Bayesian linear regression with sparse priors. *The Annals of Statistics* **43** 1986–2018.

Cech, T. R. and Steitz, J. A. (2014). The noncoding RNA revolution—trashing old rules to forge new ones. *Cell* **157** 77–94.

Crane, H. et al. (2016). Dynamic random networks and their graph limits. *The Annals of Applied Probability* **26** 691–721.

Durante, D., Dunson, D. B. et al. (2016). Locally adaptive dynamic networks. *The Annals of Applied Statistics* **10** 2203–2232.

Fan, J., Feng, Y. and Wu, Y. (2009). Network exploration via the adaptive LASSO and SCAD penalties. *The annals of applied statistics* **3** 521.

Friedman, J., Hastie, T. and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **9** 432–441.

Geweke, J. et al. (1991). *Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments* **196**. Federal Reserve Bank of Minneapolis, Research Department Minneapolis, MN.

Heidelberger, P. and Welch, P. D. (1981). A spectral method for confidence interval generation and run length control in simulations. *Communications of the ACM* **24** 233–245.

John, A., Brylka, H., Wiegreffe, C., Simon, R., Liu, P., Jüttner, R., Crenshaw, E. B., Luyten, F. P., Jenkins, N. A., Copeland, N. G. et al. (2012). Bcl11a is required for neuronal morphogenesis and sensory circuit formation in dorsal spinal cord development. *Development* **139** 1831–1841.

Kalaitzis, A., Lafferty, J., Lawrence, N. and Zhou, S. (2013). The bigraphical lasso. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)* 1229–1237.

Kharchenko, P. V., Silberstein, L. and Scadden, D. T. (2014). Bayesian approach to single-cell differential expression analysis. *Nature methods* **11** 740–742.

Kolar, M., Song, L., Ahmed, A. and Xing, E. P. (2010). Estimating time-varying networks. *The Annals of Applied Statistics* **4** 94–123.

Kyung, M., Gill, J., Ghosh, M. and Casella, G. (2010). Penalized regression, standard errors, and Bayesian lassos. *Bayesian Analysis* **5** 369–411.

Lauritzen, S. L. (1996). *Graphical models* **17**. Clarendon Press.

Lebre, S., Becq, J., Devaux, F., Stumpf, M. P. and Lelandais, G. (2010). Statistical inference of the time-varying structure of gene-regulation networks. *BMC systems biology* **4** 130.

Li, H., Radford, J. C., Ragusa, M. J., Shea, K. L., McKercher, S. R., Zaremba, J. D., Soussou, W., Nie, Z., Kang, Y.-J., Nakanishi, N. et al. (2008). Transcription factor MEF2C influences neural stem/progenitor cell differentiation and maturation in vivo. *Proceedings of the National Academy of Sciences* **105** 9397–9402.

Linqing, Z., Guohua, J., Haoming, L., Xuelei, T., Jianbing, Q. and Meiling, T. (2015). RUNX1T1 regulates the neuronal differentiation of radial glial cells from the rat hippocampus. *Stem cells translational medicine* **4** 110–116.

Matias, C. and Miele, V. (2016). Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **79** 1119-1141.

Matsumoto, H., Kiryu, H., Furusawa, C., Ko, M. S., Ko, S. B., Gouda, N., Hayashi, T. and Nikaido, I. (2017). SCODE: an efficient regulatory network inference algorithm from single-cell RNA-Seq during differentiation. *Bioinformatics* **33** 2314–2321.

Mayer, S., Chen, J., Velmeshev, D., Mayer, A., Eze, U. C., Bhaduri, A., Cunha, C. E., Jung, D.,

Arjun, A., Li, E. et al. (2019). Multimodal Single-Cell Analysis Reveals Physiological Maturation in the Developing Human Neocortex. *Neuron*.

Novershtern, N., Regev, A. and Friedman, N. (2011). Physical Module Networks: an integrative approach for reconstructing transcription regulation. *Bioinformatics* **27** i177–i185.

Nowakowski, T. J., Bhaduri, A., Pollen, A. A., Alvarado, B., Mostajo-Radji, M. A., Di Lullo, E., Haeussler, M., Sandoval-Espinosa, C., Liu, S. J., Velmeshev, D., Ounadjela, J. R., Shuga, J., Wang, X., Lim, D. A., West, J. A., Leyrat, A. A., Kent, W. J. and Kriegstein, A. R. (2017). Spatiotemporal gene expression trajectories reveal developmental hierarchies of the human cortex. *Science* **358** 1318–1323.

Palla, K., Caron, F. and Teh, Y. W. (2016). Bayesian nonparametrics for Sparse Dynamic Networks. *arXiv preprint arXiv:1607.01624*.

Park, T. and Casella, G. (2008). The bayesian lasso. *Journal of the American Statistical Association* **103** 681–686.

Pataskar, A., Jung, J., Smialowski, P., Noack, F., Calegari, F., Straub, T. and Tiwari, V. K. (2016). NeuroD1 reprograms chromatin and transcription factor landscapes to induce the neuronal program. *The EMBO journal* **35** 24–45.

Pensky, M. (2016). Dynamic network models and graphon estimation. *arXiv preprint arXiv:1607.00673*.

Pierson, E. and Yau, C. (2015). ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome biology* **16** 1–10.

Piper, M., Barry, G., Hawkins, J., Mason, S., Lindwall, C., Little, E., Sarkar, A., Smith, A. G., Moldrich, R. X., Boyle, G. M. et al. (2010). NFIA controls telencephalic progenitor cell differentiation through repression of the Notch effector Hes1. *Journal of Neuroscience* **30** 9127–9139.

Plummer, M., Best, N., Cowles, K. and Vines, K. (2006). CODA: Convergence Diagnosis and Output Analysis for MCMC. *R News* **6** 7–11.

Qiu, P., Simonds, E. F., Bendall, S. C., Gibbs Jr, K. D., Bruggner, R. V., Linderman, M. D., Sachs, K., Nolan, G. P. and Plevritis, S. K. (2011). Extracting a cellular hierarchy from high-dimensional cytometry data with SPADE. *Nature biotechnology* **29** 886–891.

Rehfeld, F., Maticzka, D., Grosser, S., Knauff, P., Eravci, M., Vida, I., Backofen, R. and Wulczyn, F. G. (2018). The RNA-binding protein ARPP21 controls dendritic branching by functionally opposing the miRNA it hosts. *Nature communications* **9** 1235.

Rosengren, S. and Trapman, P. (2016). A Dynamic Erdos Renyi Graph Model. *arXiv preprint arXiv:1604.05127*.

Sarkar, P. and Chakrabarti, D. (2014). Nonparametric link prediction in large scale dynamic networks. *Electronic Journal of Statistics* **8** 2022–2065.

Schaefer, A., Margulies, D. S., Lohmann, G., Gorgolewski, K. J., Smallwood, J., Kiebel, S. J. and Villringer, A. (2014). Dynamic network participation of functional connectivity hubs assessed by resting-state fMRI. *Frontiers in human neuroscience* **8** 195.

Sekara, V., Stopczynski, A. and Lehmann, S. (2016). Fundamental structures of dynamic social networks. *Proceedings of the national academy of sciences* **113** 9977–9982.

Shimamura, K., Ueki, M., Kawano, S. and Konishi, S. (2016). Bayesian generalized fused lasso modeling via NEG distribution. *arXiv preprint arXiv:1602.04910*.

Suvà, M. L., Rheinbay, E., Gillespie, S. M., Patel, A. P., Wakimoto, H., Rabkin, S. D., Riggi, N., Chi, A. S., Cahill, D. P., Nahed, B. V. et al. (2014). Reconstructing and reprogramming the tumor-propagating potential of glioblastoma stem-like cells. *Cell* **157** 580–594.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67** 91–108.

Tibshirani, R. J., Taylor, J. E., Candes, E. J. and Hastie, T. (2011). *The solution path of the generalized lasso*. Stanford University.

Toriyama, M., Shimada, T., Kim, K. B., Mitsuba, M., Nomura, E., Katsuta, K., Sakumura, Y., Roepstorff, P. and Inagaki, N. (2006). Shootin1: A protein involved in the organization of an asym-

metric signal for neuronal polarization. *The Journal of cell biology* **175** 147–157.

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. (2014). The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature biotechnology* **32** 381–386.

van der Pas, S., Salomond, J.-B., Schmidt-Hieber, J. et al. (2016). Conditions for posterior contraction in the sparse normal means problem. *Electronic journal of statistics* **10** 976–1000.

van Dijk, D., Nainys, J., Sharma, R., Kathail, P., Carr, A. J., Moon, K. R., Mazutis, L., Wolf, G., Krishnaswamy, S. and Pe'er, D. (2017). MAGIC: A diffusion-based imputation method reveals gene-gene interactions in single-cell RNA-sequencing data. *BioRxiv* 111591.

Xu, K. S. and Hero III, A. O. (2013). Dynamic stochastic blockmodels: Statistical models for time-evolving networks. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction* 201–210. Springer.

Zhang, S., Zhao, J. and Zhang, X.-S. (2012). Common community structure in time-varying networks. *Physical Review E* **85** 056110.

Supplement

Supplement A: derivation of equation (7)

Because $\theta^{(i)}_{t+1,j} \perp \theta^{(i)}_{t-1,j}, \theta^{(i)}_{t-2,j}, ... | \theta^{(i)}_{t,j}$ (equation (5)), the partial correlation of $\theta^{(i)}_{t+m,j}$ with $\theta^{(i)}_{t+l,j}$ will be zero for all $|m-l| > 1$. Hence, all entries of the precision matrix $[\mathbf{\Sigma}^{(i)}_j]^{-1}$ will be zero except the diagonal and the elements immediately adjacent to it (i.e., the sub- and super-diagonals). Therefore,

$[\mathbf{\Sigma}^{(i)}_j]^{-1}\mathbf{\Sigma}^{(i)}_j = \mathbb{I} \implies$

$$\begin{bmatrix} 1 & \rho^{(i)}_j & (\rho^{(i)}_j)^2 & \cdots & (\rho^{(i)}_j)^T \\ \rho^{(i)}_j & 1 & \rho^{(i)}_j & \cdots & (\rho^{(i)}_j)^{T-1} \\ (\rho^{(i)}_j)^2 & \rho^{(i)}_j & 1 & \cdots & (\rho^{(i)}_j)^{T-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (\rho^{(i)}_j)^T & (\rho^{(i)}_j)^{T-1} & (\rho^{(i)}_j)^{T-2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} \eta_{1,1} & \eta_{1,2} & 0 & \cdots & 0 \\ \eta_{2,1} & \eta_{2,2} & \eta_{2,3} & \cdots & 0 \\ 0 & \eta_{3,2} & \eta_{3,3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \eta_{T,T} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

and hence,

$$\eta_{1,1} + \rho^{(i)}_j \eta_{2,1} = 1,$$

$$\rho^{(i)}_j \eta_{1,1} + \eta_{2,1} = 0,$$

$$\implies \quad \eta_{1,1} = \frac{1}{1 - [\rho^{(i)}_j]^2}$$

$$\text{and } \eta_{2,1} = \frac{-\rho^{(i)}_j}{1 - [\rho^{(i)}_j]^2} = \eta_{1,2}, \tag{16}$$

and by symmetry (or equivalent argument), also

$$\eta_{1,1} = \frac{1}{1 - [\rho^{(i)}_j]^2}$$

$$\text{and } \eta_{T,T-1} = \frac{-\rho^{(i)}_j}{1 - [\rho^{(i)}_j]^2} = \eta_{T-1,T}.$$

Then,

$$\eta_{1,2} + \rho^{(i)}_j \eta_{2,2} + [\rho^{(i)}_j]^2 \eta_{3,2} = 0 \implies \rho^{(i)}_j \eta_{2,2} + [\rho^{(i)}_j]^2 \eta_{3,2} = \frac{\rho^{(i)}_j}{1 - [\rho^{(i)}_j]^2}$$

$$\text{and} \quad [\rho^{(i)}_j]^2 \eta_{1,2} + \rho^{(i)}_j \eta_{2,2} + \eta_{3,2} = 0 \implies \rho^{(i)}_j \eta_{2,2} + \eta_{3,2} = \frac{[\rho^{(i)}_j]^3}{1 - [\rho^{(i)}_j]^2},$$

and so subtracting the second of these equations from the first leads to

$$\left([\rho_j^{(i)}]^2 - 1\right)\eta_{3,2} = \frac{\rho_j^{(i)}\left(1 - [\rho_j^{(i)}]^2\right)}{1 - [\rho_j^{(i)}]^2}$$

and so

$$\eta_{3,2} = \frac{-\rho_j^{(i)}}{1 - [\rho_j^{(i)}]^2} = \eta_{2,3}, \tag{17}$$

$$\text{and therefore also} \quad \rho_j^{(i)}\eta_{2,2} - \frac{\rho_j^{(i)}}{1 - [\rho_j^{(i)}]^2} = \frac{[\rho_j^{(i)}]^3}{1 - [\rho_j^{(i)}]^2}$$

$$\text{and hence} \quad \eta_{2,2} = \frac{1 + [\rho_j^{(i)}]^2}{1 - [\rho_j^{(i)}]^2}.$$

Because the sub- and super-diagonal terms found in equation (17) and (16) are the same, the derivations for the other terms $\eta_{t,t+1} = \eta_{t+1,t}$ and $\eta_{t,t}$, $t = 3, ..., T-1$ will be identical and therefore we have

$$\left([\mathbf{\Sigma}_j^{(i)}]^{-1}\right)_{t,t'} = \begin{cases} 1/(1 - [\rho_j^{(i)}]^2), & \text{if } t' = t = 1 \text{ or } t' = t = T \\ (1 + [\rho_j^{(i)}]^2)/(1 - [\rho_j^{(i)}]^2), & \text{if } t' = t > 1 \text{ and } t' = t < T \\ -\rho_j^{(i)}/(1 - [\rho_j^{(i)}]^2), & \text{if } t' = t + 1 \text{ or } t' = t - 1 \\ 0, & \text{otherwise.} \end{cases}$$

Supplement B: derivations of the steps in Algorithm 1

Starting with equation (9),

$$P(\mathbf{x}_{:,i}, \mathbf{B}^{(i)}, \boldsymbol{\rho}^{(i)}, \mathbf{s}^{(i)}, a_i, \tau_i | \mathbf{X}_{:,\setminus i}, \lambda, k) = \left\{\prod_{t=1}^{T}\prod_{k=1}^{n_t}\sqrt{\frac{\tau_i}{2\pi}}e^{-\tau_i\left(x_{t,i,k} - \mathbf{b}_{t,:}^{(i)}\cdot\mathbf{x}_{t,\setminus i,k}^{\top} - a_i\right)^2/2}\right\}$$
$$\frac{1}{\sqrt{2\pi}}e^{-\left\{\tau_i + a_i^2/2\right\}}\prod_{j=1}^{p-1}\left\{\frac{k}{e^k - 1}e^{k\rho_j^{(i)}}\frac{\lambda^2}{2}e^{-\lambda^2/(2\nu_j^{(i)})}\frac{[\nu_j^{(i)}]^{-3/2}}{(2\pi)^{T/2}|\mathbf{\Sigma}_j^{(i)}|^{1/2}}e^{-\mathbf{b}_{:,j}^{(i)\top}[\mathbf{\Sigma}_j^{(i)}]^{-1}\mathbf{b}_{:,j}^{(i)}\nu_j^{(i)}/2}\right\},$$

we can write down the following expressions for conditional posteriors, for a Gibbs sampler:

$$P(\mathbf{b}_{:,j}^{(i)} | \mathbf{x}_{:,i}, \mathbf{X}_{:,\setminus i}, ...) \propto \left\{\prod_{t=1}^{T}\prod_{k=1}^{n_t} e^{-\tau_i\left(x_{t,i,k} - \mathbf{b}_{t,:}^{(i)}\cdot\mathbf{x}_{t,\setminus i,k}^{\top} - a_i\right)^2/2}\right\} e^{-\mathbf{b}_{:,j}^{(i)\top}\nu_j^{(i)}[\mathbf{\Sigma}_j^{(i)}]^{-1}\mathbf{b}_{:,j}^{(i)}/2} = g_{\mathbf{b}_j}(\mathbf{b}_{:,j}^{(i)}), \tag{18}$$

$$P(a_i | \mathbf{x}_{:,i}, \mathbf{X}_{:,\setminus i}, ...) \propto \left\{\prod_{t=1}^{T}\prod_{k=1}^{n_t} e^{-\tau_i\left(x_{t,i,k} - \mathbf{b}_{t,:}^{(i)}\cdot\mathbf{x}_{t,\setminus i,k}^{\top} - a_i\right)^2/2}\right\} e^{-a_i^2/2}, \tag{19}$$

$$P(\tau_i | \mathbf{x}_{:,i}, \mathbf{X}_{:,\setminus i}, ...) \propto \left\{\prod_{t=1}^{T}\prod_{k=1}^{n_t} \sqrt{\tau_i} e^{-\tau_i\left(x_{t,i,k} - \mathbf{b}_{t,:}^{(i)}\cdot\mathbf{x}_{t,\setminus i,k}^{\top} - a_i\right)^2/2}\right\} e^{-\tau_i}, \tag{20}$$

$$P(\nu_j^{(i)}|\mathbf{X}_{:,i},\mathbf{X}_{:,\setminus i},...) \propto\ e^{-\lambda^2/(2\nu_j^{(i)})}[\nu_j^{(i)}]^{-3/2}e^{-\mathbf{b}_{:,j}^{(i)\top}[\mathbf{\Sigma}_j^{(i)}]^{-1}\mathbf{b}_{:,j}^{(i)}\nu_j^{(i)}/2}, \tag{21}$$

and

$$P(\rho_j^{(i)}|\mathbf{X}_{:,i},\mathbf{X}_{:,\setminus i},...) \propto\ e^{k\rho_j^{(i)}}\frac{\nu_j^{(i)}}{|\mathbf{\Sigma}_j^{(i)}|^{1/2}}e^{-\mathbf{b}_{:,j}^{(i)\top}\nu_j^{(i)}[\mathbf{\Sigma}_j^{(i)}]^{-1}\mathbf{b}_{:,j}^{(i)}/2} = g_{\rho_j}(\rho_j^{(i)}). \tag{13}$$

Equation (18) can be written as:

$$\begin{aligned} g_{\mathbf{b}_j}(\mathbf{b}_{:,j}^{(i)}) &= \left\{\prod_{t=1}^{T}\prod_{k=1}^{n_t} e^{-\tau_i\left(x_{t,i,k}-\mathbf{b}_{t,:}^{(i)}\cdot\mathbf{x}_{t,\setminus i,k}^{\top}-a_i\right)^2/2}\right\} e^{-\mathbf{b}_{:,j}^{(i)\top}\nu_j^{(i)}[\mathbf{\Sigma}_j^{(i)}]^{-1}\mathbf{b}_{:,j}^{(i)}/2} \\ &= \left\{\prod_{t=1}^{T}\prod_{k=1}^{n_t} e^{-\tau_i\left(b_{t,j}^{(i)}x_{t,j,k}-x_{t,i,k}+\mathbf{b}_{t,\setminus j}^{(i)}\left(\mathbf{x}_{t,\setminus i,k}\right)_{\setminus j}^{\top}+a_i\right)^2/2}\right\} e^{-\mathbf{b}_{:,j}^{(i)\top}\nu_j^{(i)}[\mathbf{\Sigma}_j^{(i)}]^{-1}\mathbf{b}_{:,j}^{(i)}/2}, \end{aligned} \tag{22}$$

and equation (22) is recognised as the product of several Normal density functions. It is well known that the product of Normal density functions (of the same variable) is another Normal density function (e.g., a Normal likelihood with a Normal prior gives a Normal posterior). Specifically, if we combine n univariate Normal density functions with means $\mu_1^2,\mu_2^2,...,\mu_n^2$ and variances $\sigma_1^2,\sigma_2^2,...,\sigma_n^2$ then we get a univariate Normal with mean and variance specified according to:

$$\frac{1}{\sigma^2_{\text{combined}}} = \sum_{i=1}^{n}\frac{1}{\sigma_i^2} \tag{23}$$

and

$$\frac{\mu_{\text{combined}}}{\sigma^2_{\text{combined}}} = \sum_{i=1}^{n}\frac{\mu_i}{\sigma_i^2}, \tag{24}$$

and more generally if we multiply n multivariate Normal density functions with mean vectors $\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,...,\boldsymbol{\mu}_n$ and covariance matrices $\mathbf{\Sigma}_1,\mathbf{\Sigma}_2,...\mathbf{\Sigma}_n$, then we get a Normal density function with mean vector and covariance matrix given by

$$\mathbf{\Sigma}^{-1}_{\text{combined}} = \sum_{i=1}^{n}\mathbf{\Sigma}_i^{-1} \tag{25}$$

and

$$\mathbf{\Sigma}^{-1}_{\text{combined}}\boldsymbol{\mu}_{\text{combined}} = \sum_{i=1}^{n}\mathbf{\Sigma}_i^{-1}\boldsymbol{\mu}_i. \tag{26}$$

The inner-most product in equation (22) can be written as

$$\prod_{t=1}^{T}\prod_{k=1}^{n_t} e^{-\tau_i\left(b_{t,j}^{(i)}x_{t,j,k}-x_{t,i,k}+\mathbf{b}_{t,\setminus j}^{(i)}\left(\mathbf{x}_{t,\setminus i,k}\right)_{\setminus j}^{\top}+a_i\right)^2/2} = \prod_{t=1}^{T}\prod_{k=1}^{n_t} e^{-\tau_i x_{t,j,k}^2\left(b_{t,j}^{(i)}-\left\{x_{t,i,k}-\mathbf{b}_{t,\setminus j}^{(i)}\left(\mathbf{x}_{t,\setminus i,k}\right)_{\setminus j}^{\top}-a_i\right\}/x_{t,j,k}\right)^2/2},$$

and so and using the logic of equations (23) and (24) to combine Normal distributions of $b_{t,j}^{(i)}$,

$$\begin{aligned} &\prod_{t=1}^{T}\prod_{k=1}^{n_t} e^{-\tau_i x_{t,j,k}^2\left(b_{t,j}^{(i)}-\left\{x_{t,i,k}-\mathbf{b}_{t,\setminus j}^{(i)}\left(\mathbf{x}_{t,\setminus i,k}\right)_{\setminus j}^{\top}-a_i\right\}/x_{t,j,k}\right)^2/2} \\ &\qquad\propto\ e^{-\tau_i\left\{\sum_{t=1}^{T}\sum_{k=1}^{n_t}x_{t,j,k}^2\right\}\left(b_{t,j}^{(i)}-\sum_{t=1}^{T}\sum_{k=1}^{n_t}x_{t,j,k}\left\{x_{t,i,k}-\mathbf{b}_{t,\setminus j}^{(i)}\left(\mathbf{x}_{t,\setminus i,k}\right)_{\setminus j}^{\top}-a_i\right\}/\sum_{t=1}^{T}\sum_{k=1}^{n_t}x_{t,j,k}^2\right)^2/2}, \end{aligned}$$

where 'proportional to' is with respect to finding an un-normalised distribution for $b_{t,j}^{(i)}$. Hence (also referring back to equation (22)),

$$\prod_{t=1}^{T}\prod_{k=1}^{n_t} e^{-\tau_i\left(x_{t,i,k}-\mathbf{b}_{t,:}^{(i)}\cdot\mathbf{x}_{t,\setminus i,k}^{\top}-a_i\right)^2/2}$$
$$\propto \prod_{t=1}^{T} e^{-\tau_i\left\{\sum_{t=1}^{T}\sum_{k=1}^{n_t}x_{t,j,k}^2\right\}\left(b_{t,j}^{(i)}-\sum_{t=1}^{T}\sum_{k=1}^{n_t}x_{t,j,k}\left\{x_{t,i,k}-\mathbf{b}_{t,\setminus j}^{(i)}\left(\mathbf{x}_{t,\setminus i,k}\right)_{\setminus j}^{\top}-a_i\right\}/\sum_{t=1}^{T}\sum_{k=1}^{n_t}x_{t,j,k}^2\right)^2/2}$$
$$\propto\ e^{-(\mathbf{b}_{:,j}^{(i)}-\mathbf{m}_j^{(i)})^{\top}[\mathbf{V}_j^{(i)}]^{-1}(\mathbf{b}_{:,j}^{(i)}-\mathbf{m}_j^{(i)})/2}$$

(because the product of independent univariate Normal density function of different variables is proportional to a multivariate Normal density function), where the t^{th} element of $\mathbf{m}_j^{(i)}$ is

$$m_{t,j}^{(i)}=\sum_{t=1}^{T}\sum_{k=1}^{n_t}x_{t,j,k}\left\{x_{t,i,k}-\mathbf{b}_{t,\setminus j}^{(i)}\left(\mathbf{x}_{t,\setminus i,k}\right)_{\setminus j}^{\top}-a_i\right\}/\sum_{t=1}^{T}\sum_{k=1}^{n_t}x_{t,j,k}^2,$$

where $\mathbf{b}_{t,\setminus j}^{(i)}$ and $\left(\mathbf{x}_{t,\setminus i,k}\right)_{\setminus j}$ represent $\mathbf{b}_{t,:}^{(i)}$ and $\mathbf{x}_{t,\setminus i,k}$ without the j^{th} elements, respectively, and $\mathbf{V}_j^{(i)}$ is a diagonal matrix, with the t^{th} diagonal element equal to $1/\left\{\tau_i\sum_{t=1}^{T}\sum_{k=1}^{n_t}x_{t,j,k}^2\right\}$. Hence, using the logic of equations (25) and (26), and referring also to equation (18):

$$P(\mathbf{b}_{:,j}^{(i)}|\mathbf{x}_{:,i},\mathbf{X}_{:,\setminus i},...)\propto g_{\mathbf{b}_j}(\mathbf{b}_{:,j}^{(i)})=\left\{\prod_{t=1}^{T}\prod_{k=1}^{n_t}e^{-\tau_i\left(x_{t,i,k}-\mathbf{b}_{t,:}^{(i)}\cdot\mathbf{x}_{t,\setminus i,k}^{\top}-a_i\right)^2/2}\right\}e^{-\mathbf{b}_{:,j}^{(i)\top}\nu_j^{(i)}[\mathbf{\Sigma}_j^{(i)}]^{-1}\mathbf{b}_{:,j}^{(i)}/2}$$

$$\propto\ e^{-(\mathbf{b}_{:,j}^{(i)}-\widetilde{\mathbf{m}}_j^{(i)})^{\top}[\widetilde{\mathbf{\Sigma}}_j]^{-1}(\mathbf{b}_{:,j}^{(i)}-\widetilde{\mathbf{m}}_j^{(i)})/2}\propto f_{\mathcal{N}}(\mathbf{b}_{:,j}^{(i)}|\widetilde{\mathbf{m}}_j^{(i)},\widetilde{\mathbf{\Sigma}}_j)=\widetilde{g}_{\mathbf{b}_j}(\mathbf{b}_{:,j}^{(i)}),\quad(14)$$

where $[\widetilde{\mathbf{\Sigma}}_j]^{-1}=\nu_j^{(i)}[\mathbf{\Sigma}_j^{(i)}]^{-1}+[\mathbf{V}_j^{(i)}]^{-1}$, and $\widetilde{\mathbf{m}}_j^{(i)}=\widetilde{\mathbf{\Sigma}}_j[\mathbf{V}_j^{(i)}]^{-1}\mathbf{m}_j^{(i)}$, and $f_{\mathcal{N}}(\cdot|\boldsymbol{\mu},\mathbf{\Sigma})$ is the multivariate Normal density.

Referring again to equations (23) and (24), equation (19) can be re-written as

$$P(a_i|\mathbf{x}_{:,i},\mathbf{X}_{:,\setminus i},...)\propto\ e^{-a_i^2/2}\left\{\prod_{t=1}^{T}\prod_{k=1}^{n_t}e^{-\tau_i\left(a_i-\{x_{t,i,k}-\mathbf{b}_{t,:}^{(i)}\cdot\mathbf{x}_{t,\setminus i,k}^{\top}\}\right)^2/2}\right\}$$

$$\propto f_{\mathcal{N}}(a_i|\mu_a,\sigma_a)=g_a(a),\quad(10)$$

where $\sigma_a^{-2}=1+n\tau_i$ and $\mu_a=\sigma_a^2\tau_i\sum_{t=1}^{T}\sum_{k=1}^{n_t}\{x_{t,i,k}-\mathbf{b}_{t,:}^{(i)}\cdot\mathbf{x}_{t,\setminus i,k}^{\top}\}$.

Equation (20) can be written as:

$$P(\tau_i|\mathbf{x}_{:,i},\mathbf{X}_{:,\setminus i},...)\propto\ \tau_i^{n/2}e^{-\tau_i\left\{1+\sum_{t=1}^{T}\sum_{k=1}^{n_t}\left(x_{t,i,k}-\mathbf{b}_{t,:}^{(i)}\cdot\mathbf{x}_{t,\setminus i,k}^{\top}-a_i\right)^2/2\right\}}$$

$$\propto\ f_{\gamma}\left(\tau_i|k_\tau,\theta_\tau\right)=g_\tau(\tau_i),\quad(11)$$

where f_γ is the density of the gamma distribution with $k_\tau=1+\frac{n}{2}$ and

$$\theta_\tau=1/\{1+\sum\nolimits_{t=1}^{T}\sum\nolimits_{k=1}^{n_t}(x_{t,i,k}-\mathbf{b}_{t,:}^{(i)}\cdot\mathbf{x}_{t,\setminus i,k}^{\top}-a_i)^2/2\}.$$

Recalling equation (21),

$$P(\nu_j^{(i)}|\mathbf{x}_{:,i}, \mathbf{X}_{:,\setminus i}, ...) \propto e^{-\lambda^2/(2\nu_j^{(i)})}[\nu_j^{(i)}]^{-3/2} e^{-\mathbf{b}_{:,j}^{(i)\top}[\mathbf{\Sigma}_j^{(i)}]^{-1}\mathbf{b}_{:,j}^{(i)}\nu_j^{(i)}/2},$$

and also recalling the inverse Normal density

$$\begin{aligned} f_{IG}(x) =& \sqrt{\frac{\lambda'}{2\pi}} x^{-3/2} e^{-\lambda'(x-\mu')^2/(2[\mu']^2 x)}, \quad x > 0, \\ \propto& \; x^{-3/2} e^{-\lambda' x/(2[\mu']^2)} e^{-\lambda'/(2x)}, \end{aligned} \tag{27}$$

we can write

$$P(\nu_j^{(i)}|\mathbf{x}_{:,i}, \mathbf{X}_{:,\setminus i}, ...) \propto f_{IG}(\nu_j^{(i)}|\mu_\nu, \lambda_\nu) = g_{\nu_j}(\nu_j^{(i)}), \tag{12}$$

where f_{IG} is the density of the inverse Normal distribution (equation (27)), with parameters $\lambda_\nu = \lambda^2$, and $\mu_\nu = \lambda \Big/ \sqrt{\mathbf{b}_{:,j}^{(i)\top}[\mathbf{\Sigma}_j^{(i)}]^{-1}\mathbf{b}_{:,j}^{(i)}}$.

Supplement C: data pre-processing and time-inference

The data used in this study were published previously [Nowakowski et al., 2017], and are publicly available from the NCBI database of genotypes and phenotypes (dbGaP), under accession number phs000989.v3. The downloaded data were normalised to give transcript read counts per million reads (CPM), hereafter referred to simply as 'read counts'. For quality control, cells with non-zero read counts for fewer than 1000 transcripts were removed, and transcripts with non-zero read counts for fewer than 30 cells were removed. All subsequent analyses were carried out on the log(read counts + 1) for the 22989 transcripts and 4691 cells which passed quality control.

We also obtained classifications for the cells from the lab that generated the data. We visualised these classifications as follows. First, we carried out a sparse singular value decomposition: we projected the data for the 4691 cells into a reduced dimensional space corresponding to the top 42 left singular vectors. The top 42 left singular vectors were used, because the top 42 singular values were deemed to be significant, under comparison with randomised versions of the same data. Then, we used t-SNE (t-distributed stochastic neighbour embedding) [Maaten and Hinton, 2008] to further reduce the dimension of the data to two dimensions. The cells are plotted in this two dimensional space in Figures S1 and S2. The cells are clearly partitioned in this visualisation according to the classifications provided by the lab which generated the data.

As cells transition from stem-like cells (called radial glia in Figures S1 and S2) to mature cell types such as neurons, they pass through various intermediate cell types, such as intermediate progenitor cells (IPCs). Cells with similar phenotypes (i.e., physical characteristics) are expected to have similar gene-expression profiles. Therefore, cells of similar types are expected to be close together in the lower dimensional projection of Figures S1 and S2. Hence, as cells transition from stem cells to mature cells, we can expect them to pass through adjacent regions in the lower dimensional projection in Figures S1 and S2, as part of their 'developmental trajectory'. Progression along this developmental trajectory can be quantified in terms of 'pseudo developmental-time' [Nowakowski et al., 2017]. We define 5 points in pseudo developmental-time, corresponding to: $t = 1$, radial glia; $t = 2$, dividing radial glia; $t = 3$, IPCs (intermediate progenitor cells), $t = 4$, newborn neurons, $t = 5$, upper layer PFC (pre-frontal cortex) neurons.

We use these 5 inferred pseudo developmental-time points as the times of the samples to feed into the proposed time-varying network model, with 1557 corresponding cell samples. To fit the model locally around each node whilst allowing all other 22988 other nodes to be potential predictors would lead to an unnecessarily high computational cost. Instead, we identify the 'important' set of genes

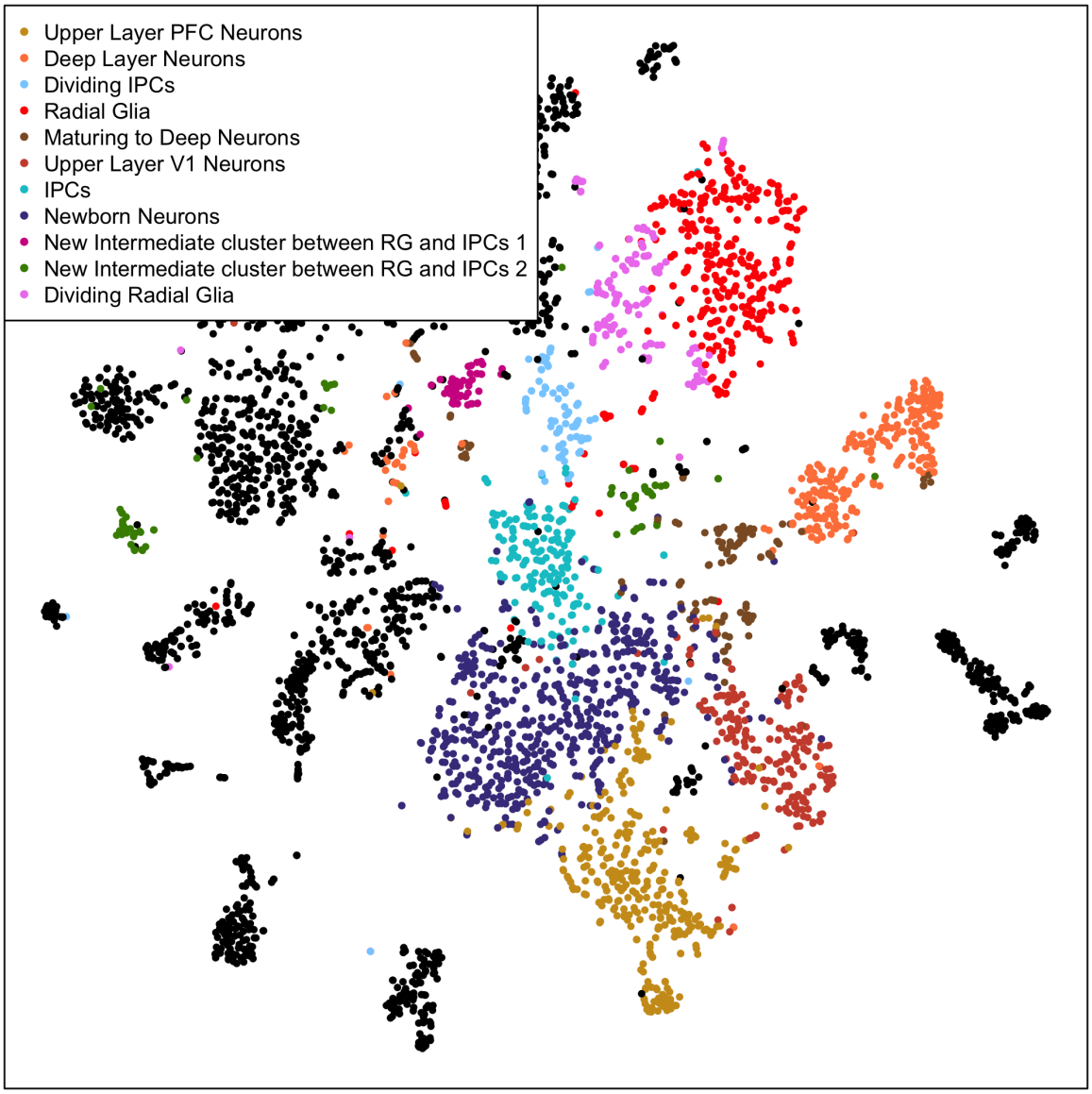

Figure S1: Low-dimensional projection of the data, with previously-obtained classifications.

with a lower computational burden, as follows. We adapt the variable screening method of Wang and Leng [2015], by finding the mean of their high-dimensional ordinary least-squares projection (HOLP) across each of the time-points. Then, for each gene we rank the 22988 other genes according to this mean HOLP, and select the $n/\log(n) = 212$ top genes according to this ranking. These 212 genes are then used as the set of possible predictors which we fit the model to. Hence, the local network structure around each node/gene is inferred from this choice of 212 other nodes/genes.

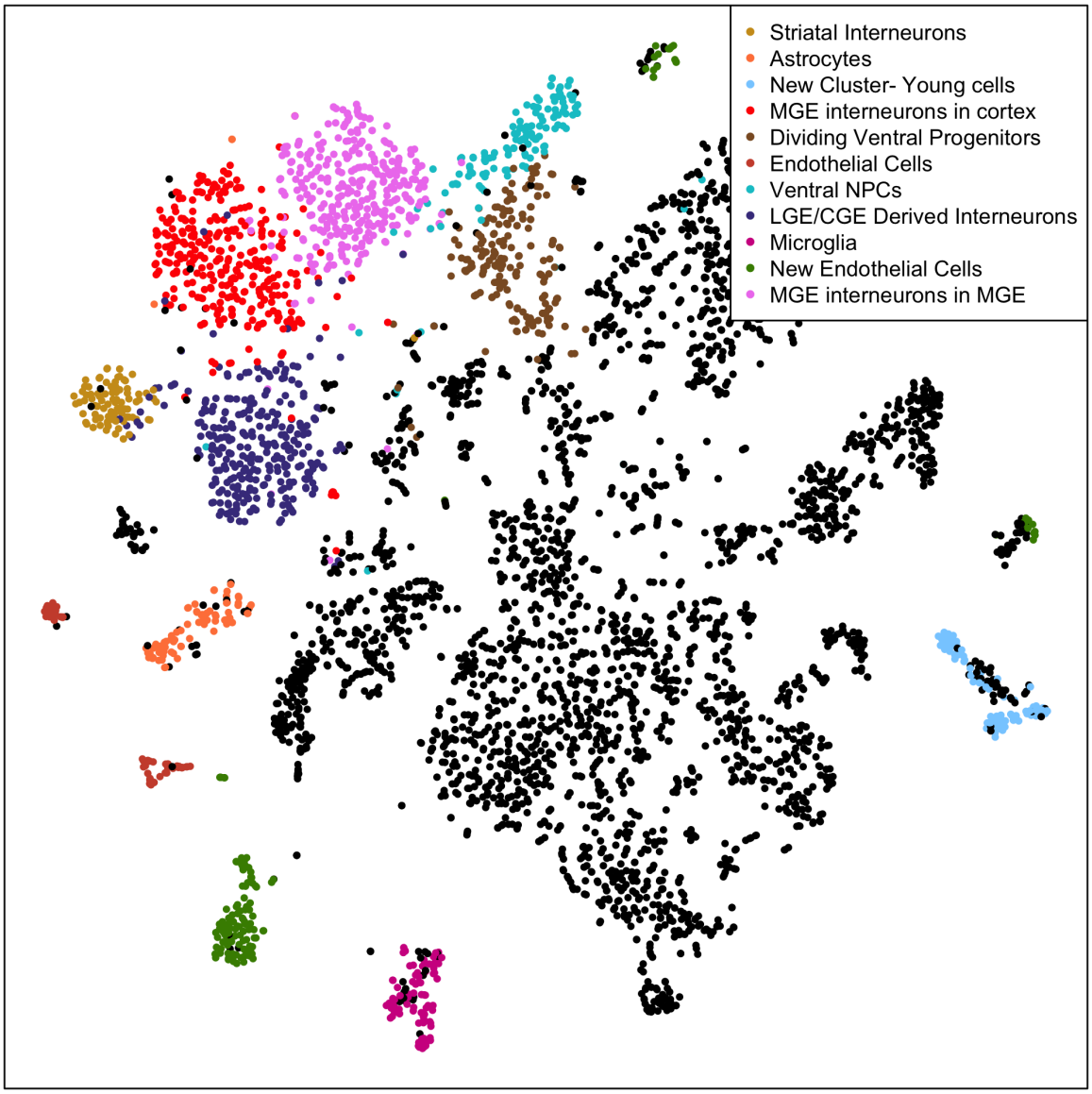

Figure S2: Low-dimensional projection of the data, with previously-obtained classifications.

Supplement D: supplementary figures

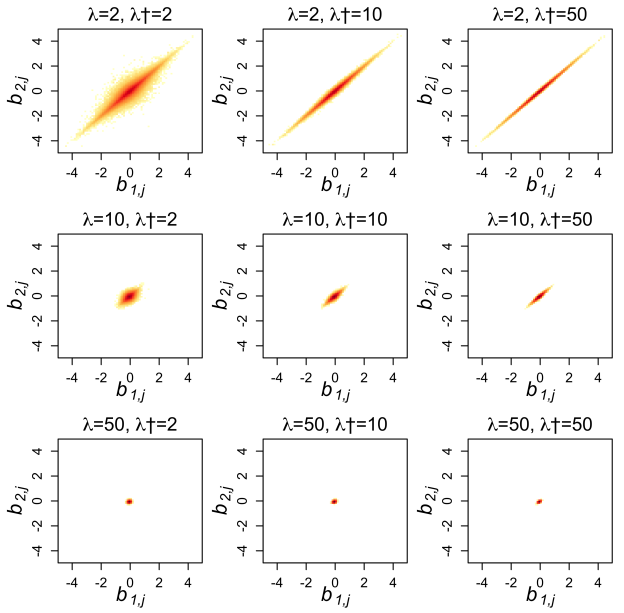

Figure S3: Heatmaps of the bivariate log-densities of prior samples for $\mathbf{b}_{:,j}^{(i)} = [b_{1,j}^{(i)}, b_{2,j}^{(i)}]^\top$, using the Laplace-NEG prior of Shimamura et al. [2016], for various values of λ and $\lambda^\dagger$, with $\gamma = 0.2$.

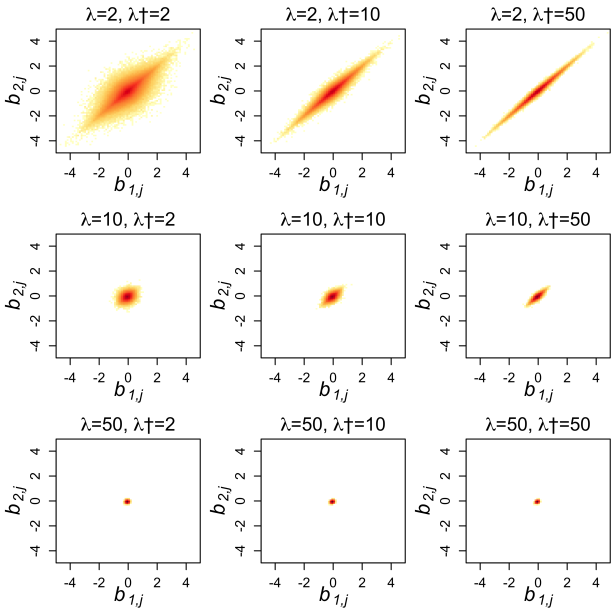

Figure S4: Heatmaps of the bivariate log-densities of prior samples for $\mathbf{b}_{:,j}^{(i)} = [b_{1,j}^{(i)}, b_{2,j}^{(i)}]^\top$, using the Laplace-NEG prior of Shimamura et al. [2016], for various values of λ and $\lambda^\dagger$, with $\gamma = 0.5$.

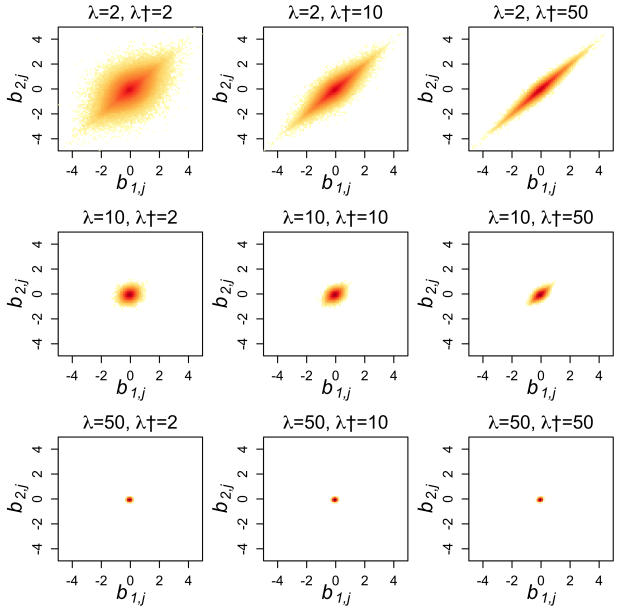

Figure S5: Heatmaps of the bivariate log-densities of prior samples for $\mathbf{b}_{:,j}^{(i)} = [b_{1,j}^{(i)}, b_{2,j}^{(i)}]^\top$, using the Laplace-NEG prior of Shimamura et al. [2016], for various values of λ and $\lambda^\dagger$, with $\gamma = 1$.

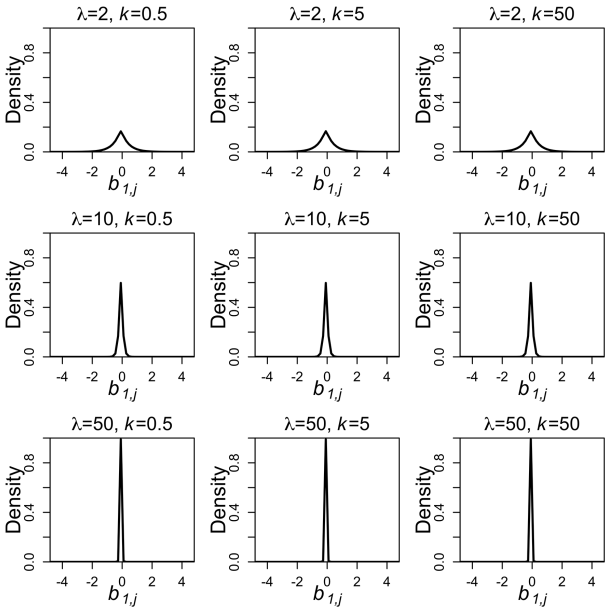

Figure S6: Marginal densities of prior samples for $b_{1,j}$, from the proposed novel decoupled-sparsity prior, for various values of λ and k.

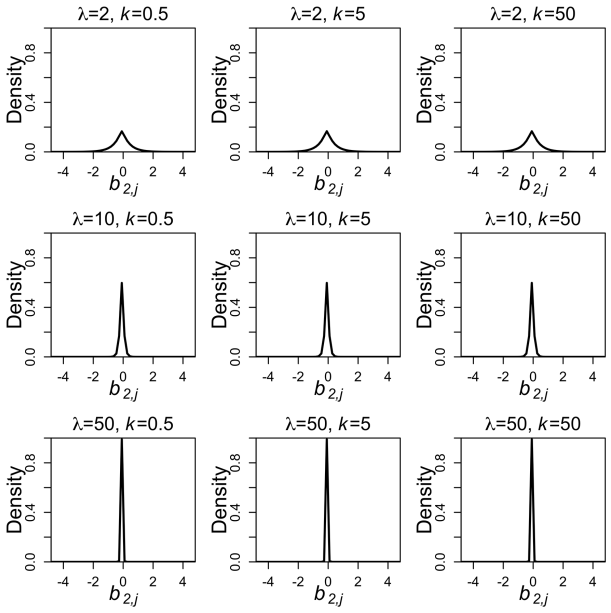

Figure S7: Marginal densities of prior samples for $b_{2,j}$, from the proposed novel decoupled-sparsity prior, for various values of λ and k.

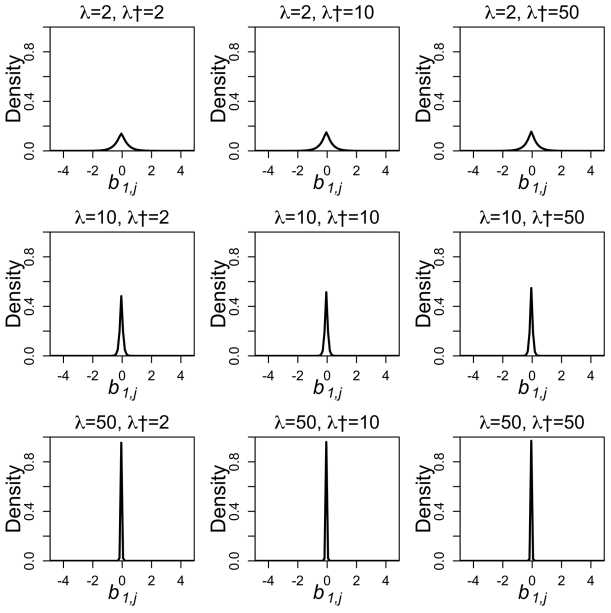

Figure S8: Marginal densities of prior samples for $b_{1,j}$, using the Laplace-NEG prior of Shimamura et al. [2016], for various values of λ and $\lambda^{\dagger}$, with $\gamma = 0.5$.

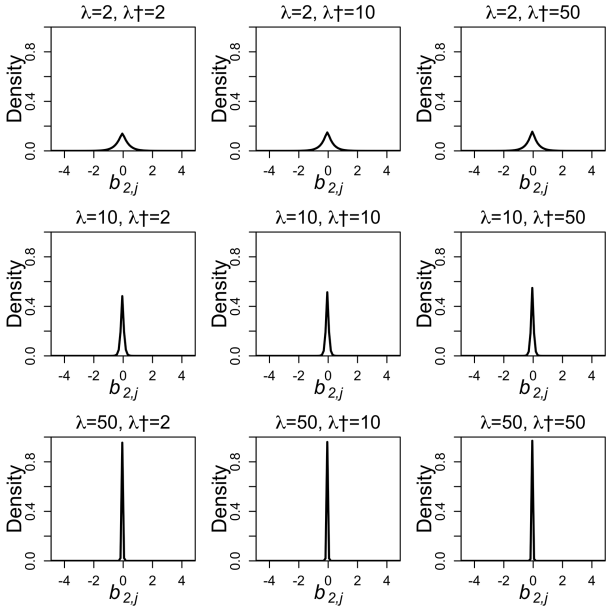

Figure S9: Marginal densities of prior samples for $b_{2,j}$, using the Laplace-NEG prior of Shimamura et al. [2016], for various values of λ and $\lambda^{\dagger}$, with $\gamma = 0.5$.

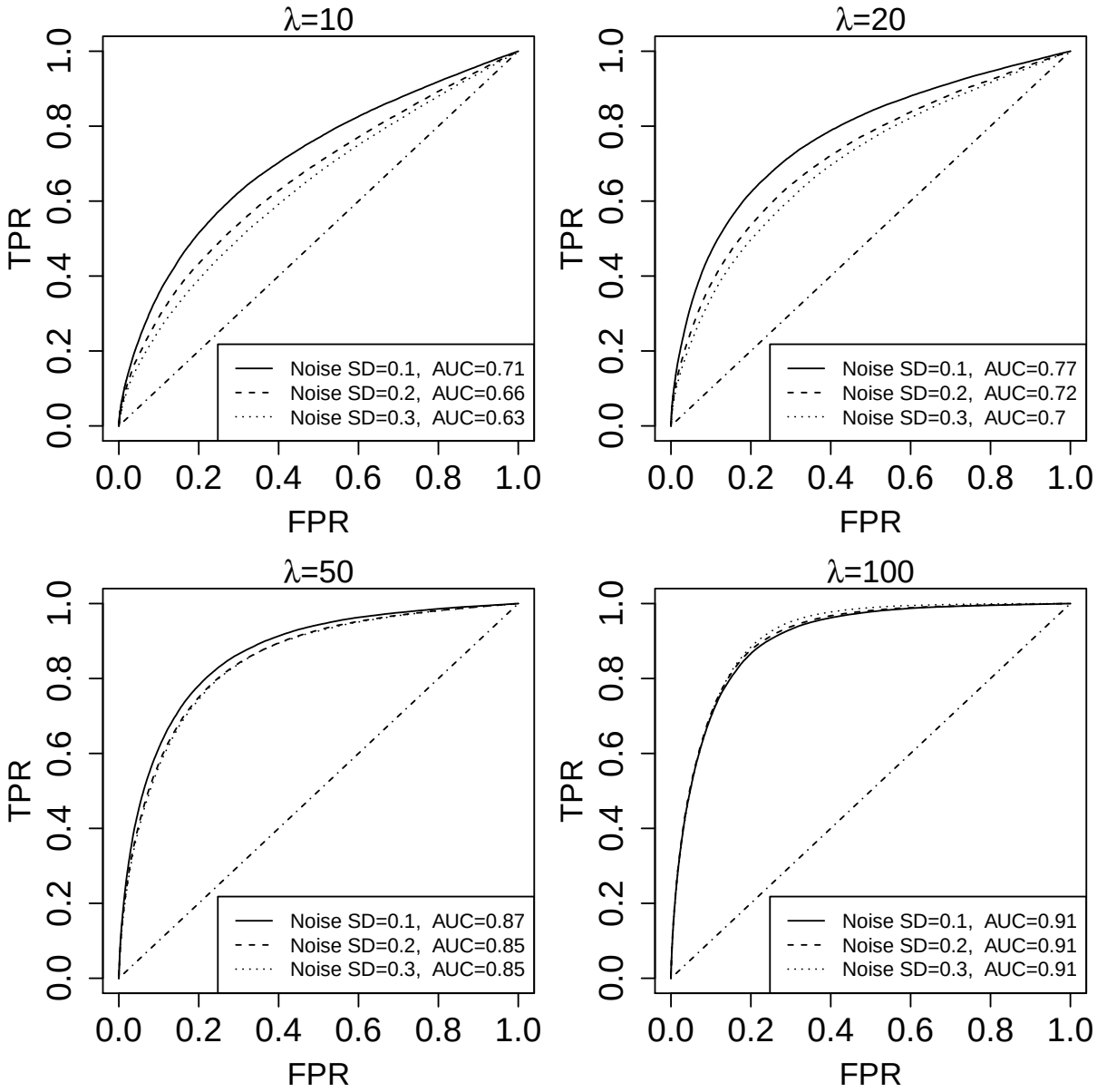

Figure S10: Accuracy of network inference, in the simulation study, with $k = 10$. Abbreviations: TP, true positives; FP, false positives.

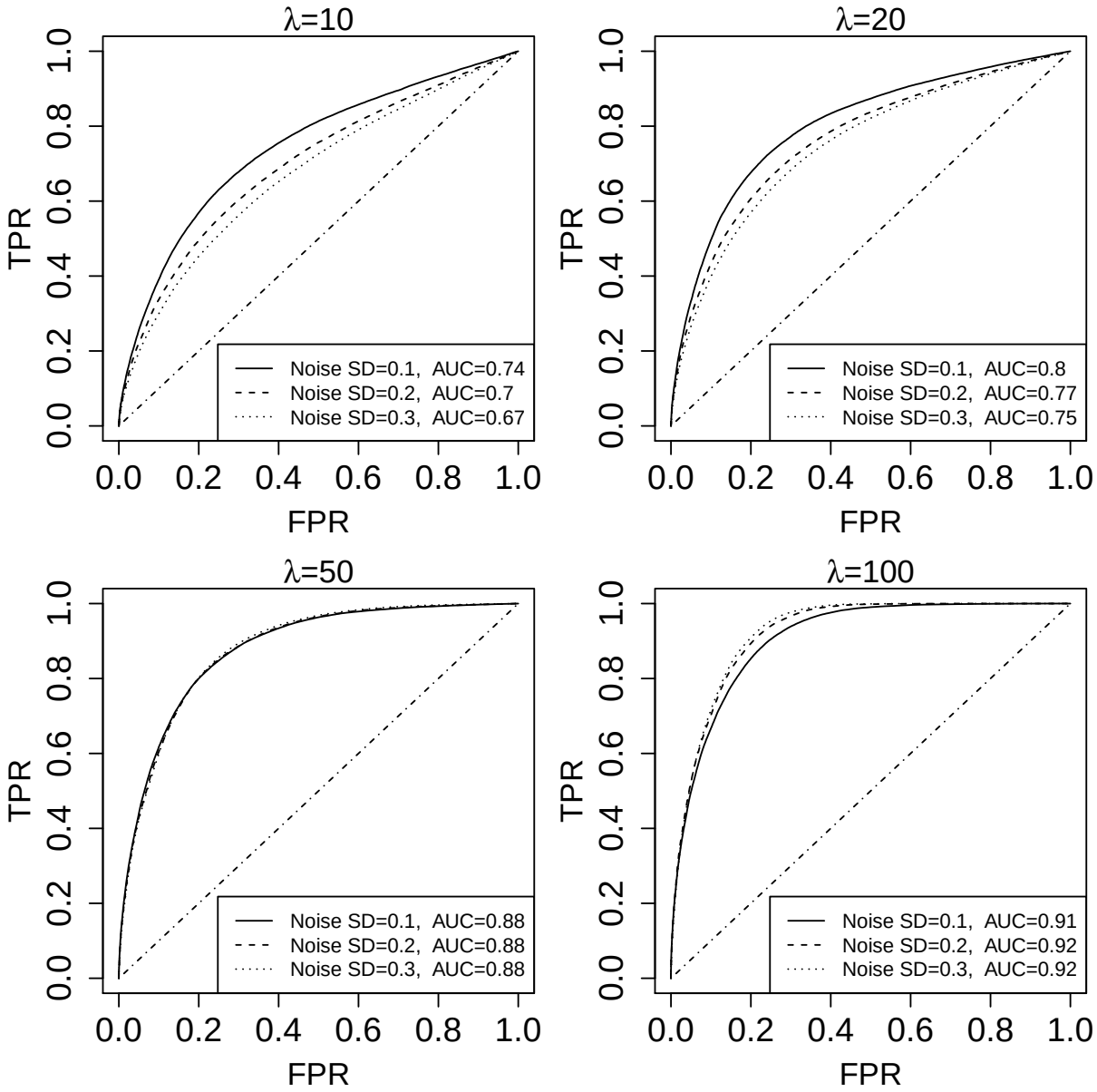

Figure S11: Accuracy of network inference, in the simulation study, with $k = 50$. Abbreviations: TP, true positives; FP, false positives.

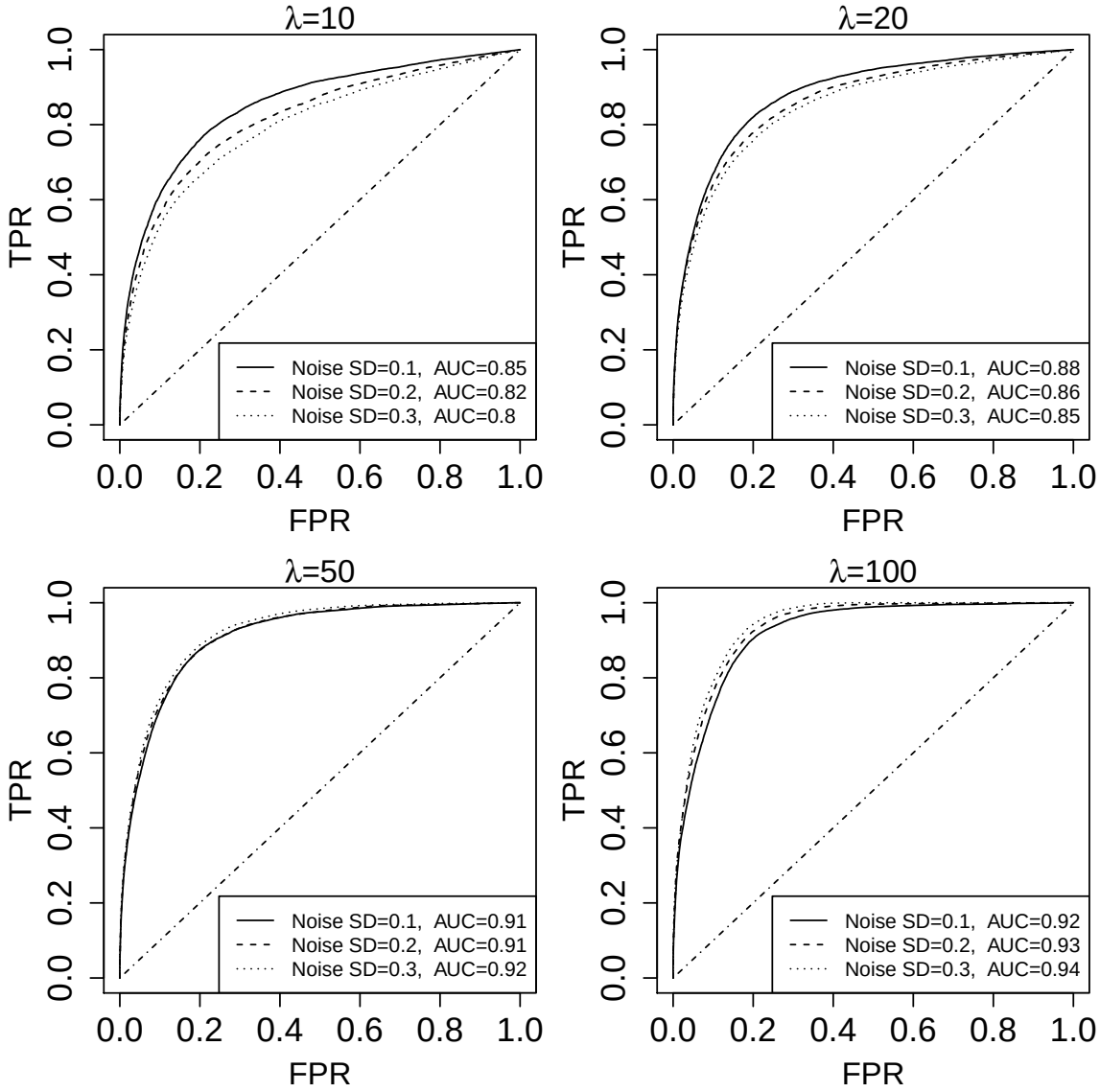

Figure S12: Accuracy of network inference, in the simulation study, with $p' = 5$. Abbreviations: TP, true positives; FP, false positives.

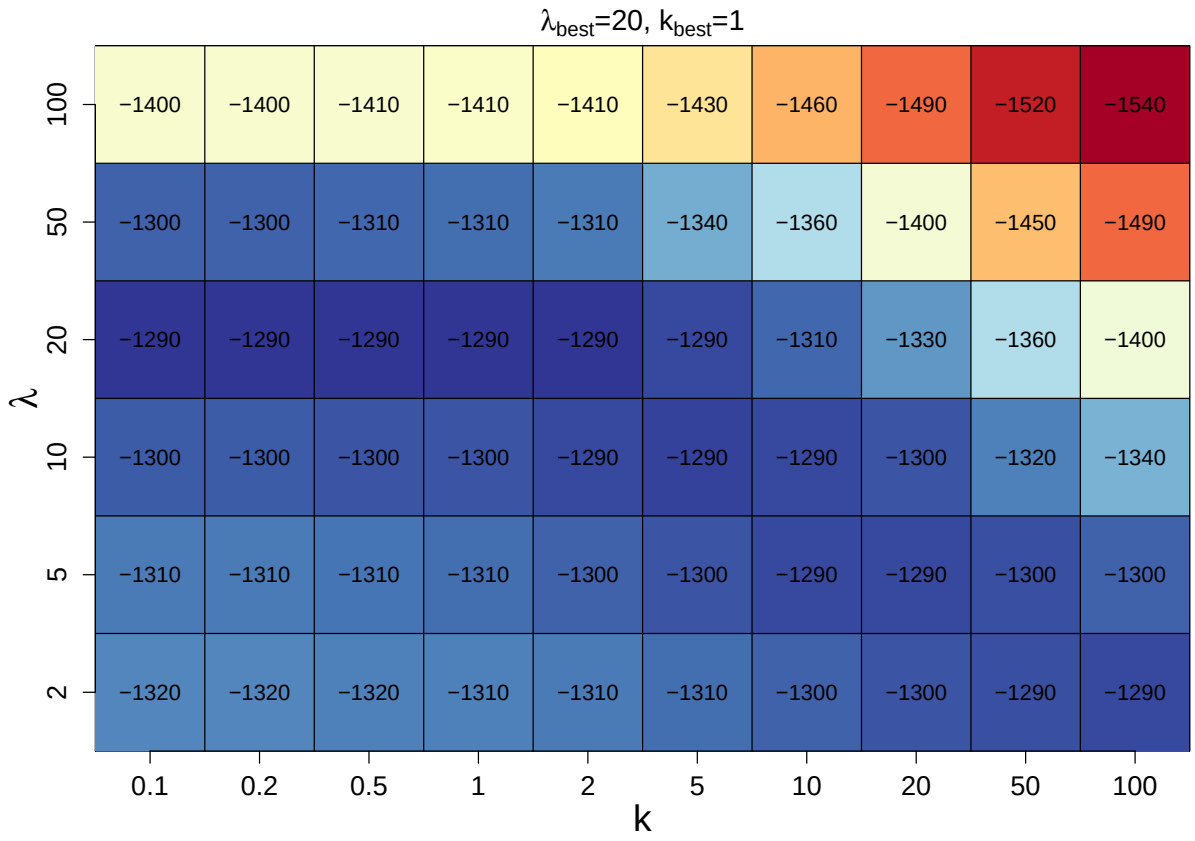

Figure S13: Model log-likelihood values for various values of λ and k, for grid-search stochastic expectation-maximization (EM) over all model fits, for the single-cell transcriptome data.

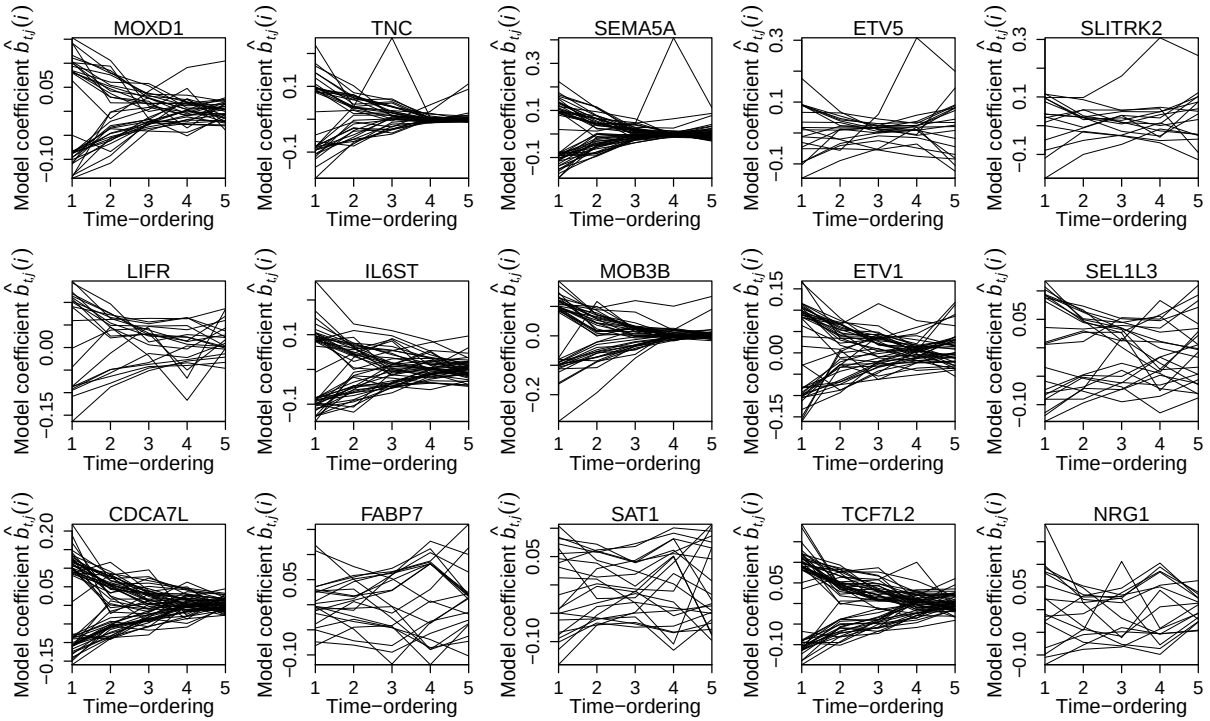

Figure S14: Inferred model parameters $\hat{b}^{(i)}_{t,j}$, for genes characteristic of stem-cells. Non-zero parameters $\hat{b}^{(i)}_{t,j}$ infer the local network structure around gene/node i. Parameters which are zero for every time-point are not plotted. This is as Figure 11a, with an expanded set of genes.

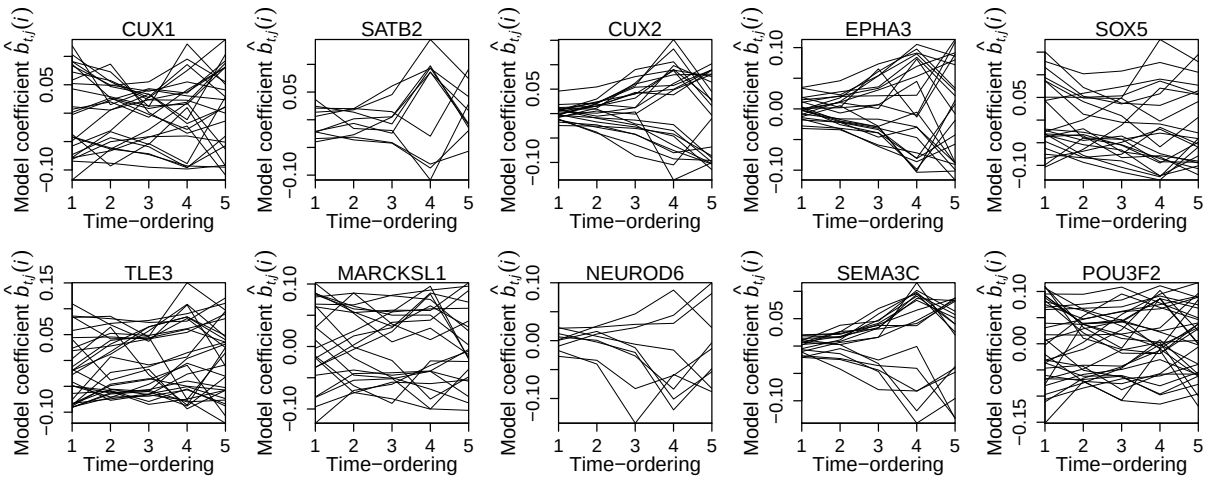

Figure S15: Inferred model parameters $\hat{b}^{(i)}_{t,j}$, for genes characteristic of neurons. Non-zero parameters $\hat{b}^{(i)}_{t,j}$ infer the local network structure around gene/node i. Parameters which are zero for every time-point are not plotted. This is as Figure 11b, with an expanded set of genes.

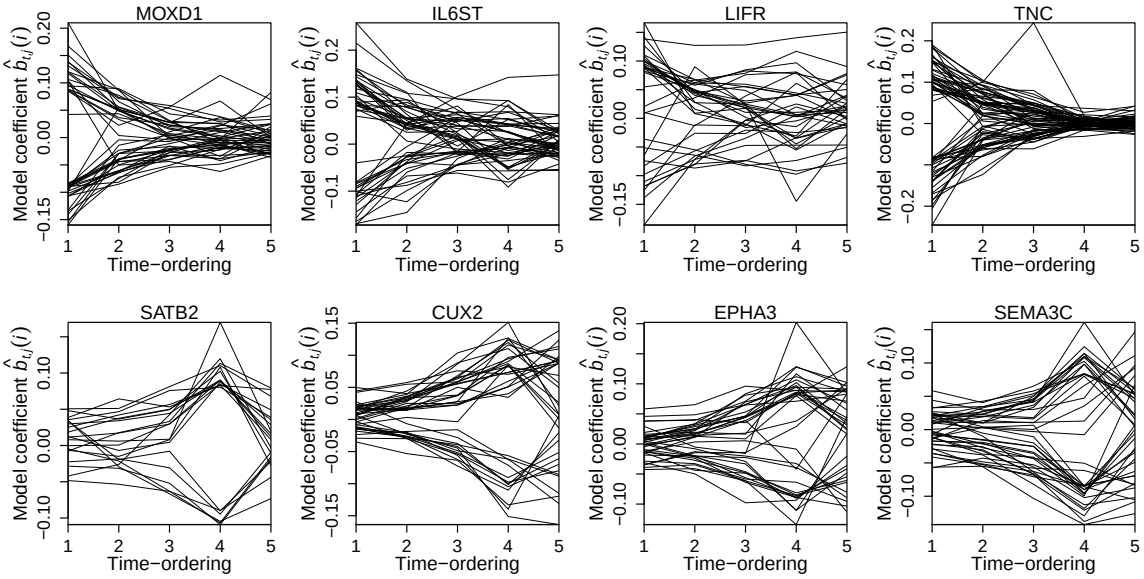

Figure S16: Inferred model parameters $\hat{b}_{t,j}^{(i)}$ for the same genes shown in Figure 11 after removing 50% of the cell-samples at random before model-fitting, for genes characteristic of: (a) stem-cells; (b) mature cells (neurons). Non-zero parameters $\hat{b}_{t,j}^{(i)}$ infer the local network structure around gene/node i. Parameters which are zero for every time-point are not plotted.