Supplementary Note

1 Bayesian Sparse Tensor Decomposition Model

1.1 Notation

Capital letters denote matrices; suppose Y is a matrix with dimensions $I \times J$. We reference the (i, j)th element of the matrix Y by y_{ij}. The ith row of Y is a row vector of length J denoted by $\mathbf{y}_{i\cdot}$ and the jth column of Y is a column vector of length I denoted by $\mathbf{y}_{\cdot j}$ or $\mathbf{y}_j$.

Tensors (by which we mean a 3-dimensional array) are represented by curly letters; for example, $\mathcal{Y} \in \mathbb{R}^{I \times J \times K}$ is a tensor with dimensions I by J by K. An element of the tensor can be referenced using three indices, e.g. y_{ijk} for $i \in \{1, \ldots, I\}$, $j \in \{1, \ldots, J\}$ and $k \in \{1, \ldots, K\}$. We would also like to represent various subsets of the data in the tensor. Let us consider an example; suppose that $\mathcal{Y} \in \mathcal{R}^{N \times L \times T}$ contains gene expression data for N individuals, L genes and T tissues. A 2-dimensional slice of the tensor can be obtained by specifying one index, for example, the data for tissue t is a matrix given by $Y_{\cdot\cdot t} \in \mathcal{R}^{N \times L}$; similarly, all the data for an individual n is given by the matrix $Y_{n\cdot\cdot} \in \mathcal{R}^{L \times T}$. The tensor analog of a matrix row or column is the vector obtained when two indices are specified, for example, $\mathbf{y}_{\cdot lt} \in \mathcal{R}^N$ represents the data (for all individuals) for gene l in tissue t.

In the following sections, we use c and k as an index over components, n as an index over individuals, l as an index over genes and t as an index over tissues.

1.2 Model description

Let $\mathcal{Y} \in \mathbb{R}^{N \times L \times T}$ be a tensor containing expression data for N individuals at L genes in T tissues. For now we will assume that there is no missing data. (In section 1.10 we describe extensions to the model to deal with missing tissue samples and randomly missing elements in the tensor.) We also assume that the data for each gene (in each tissue) has been mean centred and variance normalised (i.e. $\mathbf{y}_{\cdot lt}$ has zero mean and unit variance).

We aim to find an alternative representation of the data in terms of C latent components. Specifically, the data is modelled as a linear combination of C components and additive noise,

$$y_{nlt} = \sum_{c=1}^{C} a_{nc} b_{tc} x_{cl} + \epsilon_{nlt}, \tag{1}$$

where $X \in \mathbb{R}^{C \times L}$ is a gene loadings (or scores) matrix; each row of X defines the relative contribution of each measured gene in the component. A is an N by C matrix of individual scores which describes the individual specific mixing weights for each component. Similarly, the tissue scores matrix $B \in \mathbb{R}^{T \times C}$ contains tissue specific mixing weights. Finally, $\mathcal{E}$ is an N by L by T tensor of noise. This model is known in the literature as the PARAFAC (parallel factors) decomposition or CANDECOMP (canonical decomposition) (Carroll and Chang, 1970; Harshman and Lundy, 1994).

We fit the model in a Bayesian framework; the next section defines priors for the model, including a shrinkage prior on X.

1.3 Priors

Sparsity prior on the gene loadings matrix

We expect biological processes to only involve a relatively small subset of the total number of genes in the genome and therefore want to identify components with sparse loadings vectors. In order to encourage sparsity in the model, we use a spike and slab prior on the gene loadings matrix (Lucas et al., 2006; Mitchell and Beauchamp, 1988). The spike and slab distribution consists of a mixture of a point mass at zero (the "spike") and a Gaussian (the "slab"). Genes involved in the component will have gene loadings modelled by the Gaussian distribution, while genes with zero effect should be captured by the delta function.

The prior on x_{cl} is given by

$$P(x_{cl}|p_{cl}, \beta_c) = p_{cl}\mathcal{N}(x_{cl}|0, \beta_c^{-1}) + (1 - p_{cl})\delta_0(x_{cl}), \tag{2}$$

where p_{cl} is a mixing weight and β_c is the precision of the Gaussian (Lucas et al., 2006). This prior is a more general alternative to the original spike and slab distribution from Mitchell and Beauchamp (1988) where a single mixing parameter is specified for each component i.e. $p_{cl} = p_c$. As in Lucas et al. (2006) we found that this more general spike and slab allowed us to model sparser signals in the data and resulted in lower false positive rates.

Following convention, a gamma prior is placed on the precision parameters $\beta_c \sim \mathcal{G}(\beta_c|e, f)$ where e and f are hyper-parameters. The prior on p_{cl} is given by

$$p_{cl} \sim \rho_c\mathcal{B}(p_{cl}|g, h) + (1 - \rho_c)\delta_0(p_{cl}) \tag{3}$$

where ρ_c is component-level mixing parameter. p_{cl} encodes sparsity of the (c, l)th element in the loadings matrix. A spike and slab prior on p_{cl}, with a Beta distribution for the slab, reflects our belief that some elements in the loadings matrix should be zero, and others should have an non-zero value. If ρ_c takes a value close to 0, then the expression for p_{cl} will be dominated by the delta function at zero and the majority of the loadings vector will take values close to, or equal to, zero, resulting in a sparse component. We learn ρ_c alongside the other parameters. To complete the prior on x_{cl} we place a beta distribution on $\rho_c \sim \mathcal{B}(\rho_c|r, z)$ with hyperparameters r and z.

In order to make inference easier, we follow the approach used in Titsias and Lázaro-Gredilla (2011) and factorise the spike and slab distribution as $x_{cl} = w_{cl}s_{cl}$ where

$$w_{cl} \sim \mathcal{N}(w_{cl}|0, \beta_c^{-1}), \tag{4}$$

$$s_{cl} \sim \mathcal{B}\text{ernoulli}(s_{cl}|p_{cl}). \tag{5}$$

The random variable s_{cl} reflects the model's evidence that the (c, l)th element of X is non-zero. The magnitude of w_{cl} can be thought of as an effect size for the (c, l)th element.

We use the same trick to make inference on p_{cl} tractable; let $p_{cl} = \psi_{cl}\phi_{cl}$ where

$$\psi_{cl} \sim \mathcal{B}(\psi_{cl}|g, h), \tag{6}$$

$$\phi_{cl} \sim \mathcal{B}\text{ernoulli}(\phi_{cl}|\rho_c). \tag{7}$$

Prior on the individual and tissue scores matrices

We put a standard multivariate normal prior on the component vectors in both scores matrices.

$$\begin{aligned} P(\mathbf{a}_{\cdot c}) &= \mathcal{N}_N(\mathbf{a}_{\cdot c}|0, I_N), \\ P(\mathbf{b}_{\cdot c}) &= \mathcal{N}_T(\mathbf{b}_{\cdot c}|0, I_T). \end{aligned} \tag{8}$$

This corresponds to a prior belief that individuals and tissues are independent. Without loss of generality, we can fix the variance of these distributions to 1, because of the scaling indeterminacy of factor analysis models. A scaling can be incorporated into the precision parameters (β_c) in the gene loadings matrix.

Prior on noise precision

To complete the model specification, we use a Gaussian error term. The noise levels for each gene are modelled independently, where λ_{lt} is the noise precision for each gene and tissue combination,

$$\boldsymbol{\epsilon}_{\cdot lt} \sim \mathcal{N}_N(\boldsymbol{\epsilon}_{\cdot lt}|0, \lambda_{lt}^{-1} I_N). \tag{9}$$

The precision parameters are given a Gamma distribution with hyper-parameters u and v,

$$\lambda_{lt} \sim \mathcal{G}(\lambda_{lt}|u, v). \tag{10}$$

1.4 Full model

The full model can be written as

$$\begin{aligned} P(\mathcal{Y}|\theta) &= \prod_{lt} \mathcal{N}_N\Big(\mathbf{y}_{\cdot lt}|\sum_c \mathbf{a}_{\cdot c} b_{tc} w_{cl} s_{cl}, \lambda_{lt}^{-1} I_N\Big) \\ P(\mathbf{a}_{\cdot c}) &= \mathcal{N}_N(\mathbf{a}_{\cdot c}|0, I_N) \\ P(\mathbf{b}_{\cdot c}) &= \mathcal{N}_N(\mathbf{b}_{\cdot c}|0, I_T) \\ P(w_{cl}|\beta_c) &= \mathcal{N}\big(w_{cl}|0, \beta_c^{-1}\big) \\ P(s_{cl}|\psi_{cl}, \phi_{cl}) &= \mathcal{B}\text{ernoulli}\big(s_{cl}|\psi_{cl}, \phi_{cl}\big) \\ P(\beta_c) &= \mathcal{G}(\beta_c|e, f) \\ P(\psi_{cl}) &= \mathcal{B}\text{eta}(\psi_{cl}|g, h) \\ P(\phi_{cl}|\rho_c) &= \mathcal{B}\text{ernoulli}(\phi_{cl}|\rho_c) \\ P(\rho_c) &= \mathcal{B}\text{eta}(\rho_c|r, z) \\ P(\lambda_{lt}) &= \mathcal{G}(\lambda_{lt}|u, v) \end{aligned} \tag{11}$$

where θ denotes the set of all parameters.

1.5 Hyperparameters

We place uninformative priors on the noise precision, λ_{lt}, and the 'slab' precision β_c by setting $u = 10^{-6}$, $v = 10^6$, $e = 10^{-6}$ and $f = 10^6$. We put a flat (uniform) prior on the component sparsity parameters (ρ_c) by setting $r = z = 1$. To encourage sparsity in the gene loadings we use a prior on ψ_{cl} with $g = h = 0$.

1.6 Inference via variational Bayes

Inference is performed using an Bayesian technique called variational Bayes (VB) which allows us to evaluate an approximation to the posterior distribution. Suppose the approximate posterior distribution is given by $Q(\theta)$. VB aims to minimise the Kullbeck-Lieber (KL) divergence between $Q(\theta)$ and the true posterior $P(\theta|\mathcal{Y})$ given by

$$\mathrm{KL}(Q|P) = \int Q(\theta) \log \frac{Q(\theta)}{P(\theta|\mathcal{Y})} d\theta. \tag{12}$$

The KL divergence takes positive values, or a value of 0 if and only if $Q(\theta)$ is identical to $P(\theta|\mathcal{Y})$. We can write the marginal log-likelihood in terms of the KL divergence and a term called the negative free energy denoted by $F(Q)$,

$$\log P(\mathcal{Y}) = \underbrace{\int Q(\theta) \log \frac{P(\mathcal{Y}, \theta)}{Q(\theta)} d\theta}_{:=F(Q)} + \mathrm{KL}(Q|P). \tag{13}$$

Minimising the KL divergence is equivalent to maximising $F(Q)$. Also note that because the KL divergence can not take a negative value, $F(Q)$ is a lower bound to the log-marginal likelihood.

A common approach to optimising $F(Q)$ is the mean field VB algorithm where the approximate posterior is assumed to fully factorise. If the model priors are chosen to be conjugate, then (conditional) analytic solutions can be obtained for the variational parameters that increase $F(Q)$ and the algorithm consists of iteratively updating parameters until convergence. An alternative approach can be used if the priors are not conjugate; in this scenario, a fixed form for the posterior distribution is used and the parameters of this approximate distribution are estimated in order to maximise the negative free energy.

The majority of the parameters in our model have conjugate priors. However, in some cases, fully factorising over parameters may be too strong an assumption and also an unnecessary assumption. We retain dependence between w_{cl} and s_{cl}, rather than assuming they are independent (Titsias and Lázaro-Gredilla, 2011). Titsias and Lázaro-Gredilla (2011) show that this approach results in more robust and accurate estimates. All other parameters with conjugate priors are assumed to fully factorise in the approximate posterior distribution.

Unfortunately the parameters ψ_{cl}, ϕ_{cl} and ρ_c do not have conjugate priors and we are not able to use the results from mean field VB. Instead, we specify that their posterior distributions are point masses and optimise the free energy to find these point estimates.

The approximate posterior distribution $Q(\theta)$ for the model takes the following form

$$Q(\theta) = \prod_c Q(\mathbf{a}_{\cdot c}) \prod_{t,c} Q(b_{tc}) \prod_{c,l} Q(w_{cl}|s_{cl}) Q(s_{cl}) \prod_c Q(\beta_c) \prod_{c,l} \delta_{\psi^*_{cl}}(\psi_{cl}) \prod_{c,l} \delta_{\phi^*_{cl}}(\phi_{cl}) \prod_c \delta_{\rho^*_c}(\rho_c) \prod_{l,t} Q(\lambda_{lt}) \tag{14}$$

Our VB algorithm consists of iteratively updating each parameter given current estimates of the other parameters. All updates are guaranteed to increase (or at least not decrease) the negative free energy. All parameters are initialised randomly from their prior distribution other than parameters s_{cl}, ψ_{cl} and ϕ_{cl}, which are initialised to 0.5.

1.6.1 Variational Bayes updates

Parameters of the approximate posterior distributions are denoted using an asterix (*).

Loadings matrix

$$\begin{aligned} Q(w_{cl}|s_{cl}) &= \mathcal{N}\Big(w_{cl}\Big|s_{cl}m^*_{cl}, \big(s_{cl}\sigma^*_{cl} + (1-s_{cl})\langle\beta_c\rangle\big)^{-1}\Big) \\ \sigma^*_{cl} &= \langle\beta_c\rangle + \sum_{nt}\langle\lambda_{lt}\rangle\langle a^2_{nc}\rangle\langle b^2_{tc}\rangle \\ m^*_{cl} &= \sigma^{*-1}_{cl}\big(\sum_{n,t}\langle\lambda_{lt}\rangle y_{nlt}\langle a_{nc}\rangle\langle b_{tc}\rangle - \sum_{n,t}\langle\lambda_{lt}\rangle\langle a_{nc}\rangle\langle b_{tc}\rangle\sum_{k\neq c}\langle w_{kl}s_{kl}\rangle\langle a_{nk}\rangle\langle b_{tk}\rangle\Big) \end{aligned} \tag{15}$$

$$\begin{aligned} Q(s_{cl}) &= \mathcal{B}\text{ernoulli}(s_{cl}|\gamma^*_{cl}) \\ \gamma^*_{cl} &= \frac{1}{1+e^{-u^*_{cl}}} \\ u^*_{cl} &= \log(\psi^*_{cl}\phi^*_{cl}) - \frac{1}{2}\log\sigma^*_{cl} + \frac{\sigma^*_{cl}}{2}{m^*_{cl}}^2 - \log(1-\psi^*_{cl}\phi^*_{cl}) + \frac{1}{2}\log\langle\beta_c\rangle \end{aligned} \tag{16}$$

Sparsity parameters

We derive point estimates for the parameters ψ_{cl}, ϕ_{cl} and ρ_c by directly optimising the negative free energy. The relevant terms of the negative free energy are given by $\tilde{F}$.

$$\begin{aligned} \tilde{F} &:= \sum_{c,l}\log P(s_{cl}|\psi_{cl},\phi_{cl}) + \sum_{c,l}\log P(\psi_{cl}) + \sum_{c,l}\log P(\phi_{cl}|\rho_c) + \sum_c \log P(\rho_c) \\ &= \sum_{c,l}\big(\langle s_{cl}\rangle\log(\psi_{cl}\phi_{cl}) + \langle 1-s_{cl}\rangle\log(1-\psi_{cl}\phi_{cl})\big) \\ &\quad + \sum_{c,l}\big((g-1)\log\psi_{cl} + (h-1)\log(1-\psi_{cl})\big) \\ &\quad + \sum_{c,l}\big(\phi_{cl}\log\rho_c + (1-\phi_{cl})\log(1-\rho_c)\big) \\ &\quad + \sum_c\big((r-1)\log\rho_c + (z-1)\log(1-\rho_c)\big) \end{aligned} \tag{17}$$

The equation $\frac{\delta F}{\delta\rho_c} = 0$ has a closed form solution so we can find ρ^*_c as follows,

$$\rho^*_c = \frac{\sum_l \phi^*_{cl} + r - 1}{L + r + z - 2} \tag{18}$$

Since we expect ψ_{cl} and ϕ_{cl} to be highly coupled, we use Newton's method to simultaneously find $(\psi^*_{cl}, \phi^*_{cl})$ to optimise $\tilde{F}$. The optimisation problem we need to solve is

$$(\psi^*_{cl}, \phi^*_{cl}) = \text{argmax}_{(\psi_{cl}, \phi_{cl})} \tilde{F} \tag{19}$$

The gradient and Hessian matrix of $\tilde{F}$ are given by

$$\mathbf{g} = \begin{pmatrix} \frac{\langle s_{cl} \rangle}{\psi_{cl}} - \frac{\langle 1 - s_{cl} \rangle \phi_{cl}}{1 - \psi_{cl}\phi_{cl}} + \frac{g-1}{\psi_{cl}} - \frac{h-1}{1-\psi_{cl}} \\ \frac{\langle s_{cl} \rangle}{\phi_{cl}} - \frac{\langle 1 - s_{cl} \rangle \psi_{cl}}{1 - \psi_{cl}\phi_{cl}} + \log \rho_c - \log(1 - \rho_c) \end{pmatrix} \tag{20}$$

$$H = \begin{pmatrix} -\frac{\langle s_{cl} \rangle}{\psi^2_{cl}} - \frac{\langle 1 - s_{cl} \rangle \phi^2_{cl}}{(1 - \psi_{cl}\phi_{cl})^2} - \frac{qr-1}{\psi^2_{cl}} - \frac{q(1-r)-1}{(1-\psi_{cl})^2} & -\frac{\langle 1 - s_{cl} \rangle}{(1 - \psi_{cl}\phi_{cl})^2} \\ -\frac{\langle 1 - s_{cl} \rangle}{(1 - \psi_{cl}\phi_{cl})^2} & -\frac{\langle s_{cl} \rangle}{\phi^2_{cl}} - \frac{\langle 1 - s_{cl} \rangle \psi^2_{cl}}{(1 - \psi_{cl}\phi_{cl})^2} \end{pmatrix} \tag{21}$$

We update (ψ_{cl}, ϕ_{cl}) as follows,

$$\begin{pmatrix} \psi^{i+1}_{cl} \\ \phi^{i+1}_{cl} \end{pmatrix} = \begin{pmatrix} \psi^{i}_{cl} \\ \phi^{i}_{cl} \end{pmatrix} - \alpha H^{i-1} \mathbf{g}^i \tag{22}$$

where α is a step-size determined using a backtracking line search, i.e. we start with $\alpha = 1$ then reduce α until we satisfy $\tilde{F}^{i+1} > \tilde{F}^i$

Update for $\mathbf{a}_c$

$$\begin{aligned} Q(\mathbf{a}_{\cdot c}) &= \mathcal{N}_N(\mathbf{a}_{\cdot c} | \boldsymbol{\mu}^*_c, \Omega^{*-1}_c) \\ \Omega^*_c &= (1 + \sum_{l,t} \langle \lambda_{lt} \rangle \langle b^2_{tc} \rangle \langle w^2_{cl} s^2_{cl} \rangle) I_N \\ \boldsymbol{\mu}^*_c &= \Omega^{*-1}_c \Big(\sum_{l,t} \langle \lambda_{lt} \rangle \mathbf{y}_{\cdot lt} \langle b_{tc} \rangle \langle w_{cl} s_{cl} \rangle - \sum_{l,t} \langle \lambda_{lt} \rangle \langle b_{tc} \rangle \langle w_{cl} s_{cl} \rangle \sum_{k \neq c} \langle \mathbf{a}_{\cdot k} \rangle \langle b_{tk} \rangle \langle w_{kl} s_{kl} \rangle \Big) \end{aligned} \tag{23}$$

Update for b_{tc}

$$\begin{aligned} Q(b_{tc}) &= \mathcal{N}(b_{tc} | \nu^*_{tc}, \tau^{*-1}_{tc}) \\ \tau^*_{tc} &= 1 + \sum_{n,l} \langle \lambda_{lt} \rangle \langle a^2_{nc} \rangle \langle w^2_{cl} s^2_{cl} \rangle \\ \nu^*_{tc} &= \tau^{*-1}_{tc} \Big(\sum_{n,l} \langle \lambda_{lt} \rangle y_{nlt} \langle a_{nc} \rangle \langle x_{cl} \rangle - \sum_{n,l} \langle \lambda_{lt} \rangle \langle a_{nc} \rangle \langle w_{cl} s_{cl} \rangle \sum_{k \neq c} \langle a_{nk} \rangle \langle b_{tk} \rangle \langle w_{kl} s_{kl} \rangle \Big) \end{aligned} \tag{24}$$

Update for β_c

$$\begin{aligned} Q(\beta_c) &= \mathcal{G}(e^*_c, f^*_c) \\ e^*_c &= e + \frac{L}{2} \\ f^*_c &= \Big(\frac{1}{f} + \frac{1}{2} \sum_l \langle w^2_{cl} \rangle \Big)^{-1} \end{aligned} \tag{25}$$

Update for λ_{lt}

$$Q(\lambda_{lt}) = \mathcal{G}(\lambda_{lt}|u^*_{lt}, v^*_{lt}) \tag{26}$$

$$u^*_{lt} = u + \frac{NT}{2}$$

$$v^*_{lt} = \left(\frac{1}{v} + \frac{1}{2}\sum_n \left\langle \left(y_{nlt} - \sum_c a_{nc} b_{tc} w_{cl} s_{cl}\right)^2 \right\rangle\right)^{-1} \tag{27}$$

1.6.2 Negative free energy

The negative free energy is a lower bound of the model evidence (marginal likelihood). The updates given above are guaranteed to increase the free energy.

$$\begin{aligned}
F(Q) = &-\frac{NLT}{2}\log 2\pi + \frac{N}{2}\sum_{l,t}\langle \log \lambda_{lt}\rangle - \frac{1}{2}\sum_{n,l,t}\langle\lambda_{lt}\rangle\big\langle (y_{nlt} - \Sigma_c a_{nc} b_{tc} w_{cl} s_{cl})^2\big\rangle \\
&- \frac{1}{2}\sum_c \langle \mathbf{a}_{\cdot c}^\top \mathbf{a}_{\cdot c}\rangle - \frac{1}{2}\sum_c \log|\Omega^*_c| + \frac{NC}{2} \\
&- \frac{1}{2}\sum_{t,c}\langle b^2_{tc}\rangle - \frac{1}{2}\sum_{t,c}\log|\nu_{tc}| + \frac{TC}{2} \\
&+ \frac{L}{2}\sum_c \big\langle \log \beta_c\big\rangle + \frac{CL}{2} - \frac{1}{2}\sum_{c,l}\langle\beta_c\rangle\big\langle w^2_{cl}\big\rangle \\
&- \frac{1}{2}\sum_{c,l}\gamma^*_{cl}\log\sigma^*_{cl} + \frac{1}{2}\sum_{c,l}(1-\gamma^*_{cl})\log\langle\beta_c\rangle \\
&\sum_c \Bigg(-\log\Gamma(e) - e\log f + (e-1)(\psi(e^*_c) + \log\hat{f}_c) - \frac{e^*_c f^*_c}{f} \\
&+ e^*_c + \log f^*_c + \log\Gamma(e^*_c) - (e^*_c - 1)\psi(e^*_c)\Bigg) \\
&+ \sum_{c,l}\Bigg(\langle s_{cl}\rangle\langle\log\psi_{cl}\phi_{cl}\rangle + (1-\langle s_{cl}\rangle)\langle\log\big(1-\psi_{cl}\phi_{cl}\big)\rangle \\
&- \langle s_{cl}\rangle\log\langle s_{cl}\rangle - (1-\langle s_{cl}\rangle)\log(1-\langle s_{cl}\rangle)\Bigg) \\
&+ \sum_{c,l}\Big((g-1)\log\psi^*_{cl} + (h-1)\log(1-\psi^*_{cl})\Big) \\
&+ \sum_{cl}\Big(\phi^*_{cl}\log\rho^*_c + (1-\phi^*_{cl})\log(1-\rho^*_c)\Big) \\
&+ \sum_{cl}\Big((r-1)\log\phi^*_{cl} + (z-1)\log(1-\phi^*_{cl})\Big) \\
&+ \sum_{lt}\Bigg(-\log\Gamma(u) - u\log v + (u-1)(\psi(u^*_{lt}) + \log v^*_{lt}) - \frac{u^*_{lt}v^*_{lt}}{v} \\
&+ u^*_{lt} + \log v^*_{lt} + \log\Gamma(u^*_{lt}) - (u^*_{lt}-1)\psi(u^*_{lt})\Bigg)
\end{aligned} \tag{28}$$

1.7 Identifiability

The components estimated by our model are not completely identifiable. The sign of the gene loadings, individual scores and tissue scores is not fully determined by the model, so that swapping the sign of any two of these parts of the model will produce an equivalent model

fit. Scaling of the components is constrained to some extent by the fixed unit variances used in the priors on the individual and tissue scores. The use of sparsity in the gene loadings goes some way to ensure that components are not rotationally invariant, but dense factors will clearly suffer from this problem, especially if they are active in only one tissue.

1.8 Implementation and complexity

We implement the model in `C++` using a matrix library called Eigen (`http://eigen.tuxfamily.org`). The complexity of the algorithm is $\mathcal{O}(NLTC^2)$ but parallelisation of matrix multiplications (via Eigen) and parallel updates of elements in the gene loadings matrix (via openmp) speed up the code considerably.

1.9 Convergence

Based on experience of running this method on simulated and real data, we run the method for 3,000 iterations. We check for convergence by tracking the change in $\langle S \rangle$. After 3,000 iterations, the average number of elements in $\langle S \rangle$ that cross the threshold 0.5 drops to less than 1 per iteration.

1.10 Handling missing data

In this section we describe two extensions to the model which allow for missing data. We consider two scenarios in which missing data might arise, missing tissue samples and randomly missing data points.

1.10.1 Missing tissue samples

Missing tissue samples arise when only a subset of the tissues are collected for a particular individual, or if data for a whole tissue sample is removed due to experimental errors. Missing samples correspond to missing vectors within the data tensor; for example, if data for individual n in tissue t is missing, then $\mathbf{y}_{n \cdot t}$ will be missing. We can reformulate our model to ignoring these missing samples.

Let $\mathcal{J}$ be a binary indicator matrix of dimensions N by T, where $\mathcal{J}_{nt} = 1$ if data for individual n in tissue t exists and $\mathcal{J}_{nt} = 0$ otherwise. Based on the data that does exist, the likelihood is given by,

$$P(\mathcal{Y}|\theta) = \prod_{n,l,t} \mathcal{N}\Big(y_{nlt}|\Sigma_c a_{nc} b_{tc} w_{cl} s_{cl}, \lambda_{lt}^{-1}\Big)^{\mathcal{J}_{nt}} \tag{29}$$

Using this likelihood and the priors defined in section 1.3, we can derive updates for the model parameters in a similar way as above. The resulting updates are identical to those given in section 1.6.1 except that the indicator matrix $\mathcal{J}$ needs to be added into any expression with a sum over n or t.

1.10.2 Missing data points

We will now briefly describe how to deal with randomly missing elements in the data tensor that may have arisen due to experimental errors. In this scenario, we treat the missing data points as parameters in the model and learn their posterior distribution.

We create a partition of the data such that $\mathcal{Y} = \mathcal{Y}^o \cup \mathcal{Y}^m$ where $\mathcal{Y}^o$ denotes the set of observed data and $\mathcal{Y}^m$ denotes the set of missing data. Let $\mathcal{S}^m$ be the set of triplets $\{n, l, t\}$ for which data is missing.

The prior for the missing data is,

$$P(\mathcal{Y}^m|\theta) = \prod_{\{n,l,t\}\in\mathcal{S}^m} \mathcal{N}\Big(y^m_{nlt}|\Sigma_c a_{nc} b_{tc} w_{cl} s_{cl}, \lambda_{lt}^{-1}\Big), \tag{30}$$

and assuming that the posterior factorises fully, the posterior is given by

$$Q(\mathcal{Y}^m) = \prod_{\{n,l,t\}\in S^m} \mathcal{N}(y^m_{nlt}|\Sigma_c \langle a_{nc}\rangle \langle b_{tc}\rangle \langle w_{cl} s_{cl}\rangle, \langle \lambda_{lt}^{-1}\rangle). \tag{31}$$

With this extension, the updates for the other model parameter are similar to those given in section 1.6.1, altered to reflect the uncertainty in the estimates of the missing data points, if $\{n, l, t\} \in S^m$, we need to replace y_{nlt} by $\langle y_{nlt}\rangle$ and y^2_{nlt} by $\langle y^2_{nlt}\rangle = \langle y_{nlt}\rangle^2 + \langle \lambda_{lt}^{-1}\rangle$.

1.11 Allowing for related individuals

As it stands, our model ignores any relatedness between samples. However genetic studies often contain closely related individuals by design, or distantly related individuals by chance when recruitment occurs within a small geographical area. A kinship matrix $K \in \mathbf{R}^{N\times N}$ can be used to summarise this genetic relatedness between individuals where an element of K, k_{ij}, is a measure of the relatedness between individual i and individual j. Data from related individuals are likely to be correlated due to shared genetic material and explicitly modelling these correlations may lead to better results.

Our model identifies both genetic structure (e.g. a *trans* network or the genetic basis of ageing) and non-genetic structure (e.g. environment signals or batch effects) in the data. We can accommodate these different types of components by using the following prior on the individual scores matrix A, using the kinship matrix to inform the model about the relatedness between individuals,

$$A \sim \prod_c \mathcal{N}_N(\mathbf{a}_{\cdot c}|0, \alpha_c K + (1-\alpha_c) I_N), \tag{32}$$

where the covariance of each scores vector is a mixture of the kinship matrix and the identity matrix with a different mixing parameter α_c for each component. If α_c is close to 1 then the covariance matrix in the prior is approximately the kinship matrix K, which imposes a structure so that related individuals have more similar scores, resulting in a 'genetic' component. On the other hand, if α_c is close to 0 then the prior has no genetic basis and we recover the i.i.d Gaussian prior already described for A. The mixing parameter α_c is given an uninformative Beta prior,

$$\alpha_c \sim \mathcal{B}\text{eta}(\alpha_c|1, 1). \tag{33}$$

Implementing this model involves a change to the update for A and the addition of an update for α_c but the remaining parameter updates remain the same. We assume that $\mathbf{a}_{\cdot c}$ and α_c are independent in the approximate posterior distribution and that the posterior distribution of α_c is a delta function at α_c^*.

The update for A becomes,

$$
\begin{aligned}
Q(\mathbf{a}_{\cdot c}) &= \mathcal{N}_N(\mathbf{a}_{\cdot c}|\boldsymbol{\mu}_c^*, \Omega_c^{*-1}) \\
\Omega_c^* &= \Big(\alpha_c^* K + (1-\alpha_c^*) I_N\Big)^{-1} + \big(\sum_{l,t} \langle\lambda_{lt}\rangle\langle b_{tc}^2\rangle\langle w_{cl}^2 s_{cl}^2\rangle\big) I_N \\
\boldsymbol{\mu}_c^* &= \Omega_c^{*-1}\Big(\sum_{l,t}\langle\lambda_{lt}\rangle \mathbf{y}_{\cdot lt}\langle b_{tc}\rangle\langle w_{cl}s_{cl}\rangle - \sum_{l,t}\langle\lambda_{lt}\rangle\langle b_{tc}\rangle\langle w_{cl}s_{cl}\rangle \sum_{k\neq c}\langle\mathbf{a}_{\cdot k}\rangle\langle b_{tk}\rangle\langle w_{kl}s_{kl}\rangle\Big) \quad (34)
\end{aligned}
$$

An efficient implementation of this can be obtained by using the eigendecomposition of K to avoid inverting an N by N matrix in the calculation for Ω_c^*. In fact, we can avoid calculating Ω_c^* altogether as the expression for $\boldsymbol{\mu}_c$ only requires Ω_c^{*-1}. Using the eigendecomposition of $K = QDQ^t$ where Q is an orthonormal matrix of eigenvectors and D is a diagonal matrix with eigenvalues on the diagonal. We can now write,

$$
(\Omega_c^{*-1})_{nm} = Q\Big(((1-\alpha_c^*)I_N + \alpha_c^* D)^{-1} + \sum_{l,t}\langle\lambda_{lt}\rangle\langle b_{tc}^2\rangle\langle w_{cl}^2 s_{cl}^2\rangle I_N\Big)^{-1} Q^t. \quad (35)
$$

Using the above expression, the complexity in N scales quadratically. We note that this expression will no longer apply when there are missing samples in the data. The combination of the genetic prior and missing samples makes the complexity cubic in N which is why we do not use this approach when analysing the TwinsUK data set.

We evaluate the point estimates α_c^* using gradient ascent,

$$
\alpha_c^* \leftarrow \alpha_c^* + \Delta \sum_n (-1 + D_{nn})\Big(-\frac{1}{1-\alpha_c^* + \alpha_c^* D_{nn}} + \frac{\big(Q(\mu_c^*\mu_c^{*t} + \Omega_c^{*-1})\big)_{nn}}{(1-\alpha_c^* + \alpha_c^* D_{nn})^2}\Big), \quad (36)
$$

where $\Delta = 0.0001$ is the step size.

1.12 Linked matrix/tensor decomposition

The 3D tensor decomposition method that we have described above is actually a special case of a more general model we have implemented for linked tensor decomposition (see Supplementary Figure 38). Consider a study consisting of D types of omics data for a set of N individuals. Let each data set d be represented by the tensor, $\mathcal{Y}^{(d)} \in \mathbb{R}^{N\times L_d \times T_d}$ where L_d is the number of variables measured for data type d and T_d is the number of contexts (or conditions) in which these variables were measured. If data for type d is collected in only a single context then $T_d = 1$. Importantly, all tensors are linked by their shared first dimension (N).

The data is modelled as follows (Groves et al., 2011),

$$
y_{nlt}^{(d)} = \sum_c a_{nc} b_{tc}^{(d)} x_{cl}^{(d)} + \epsilon_{nlt}^{(d)} \qquad \text{for } d \in \{1, \cdots, D\} \quad (37)
$$

where $A \in \mathbb{R}^{N\times C}$ is the individual scores matrix (shared across all data types), $B^{(d)} \in \mathbb{R}^{T_d \times C}$ is a context specific scores matrix for data type d and $X^{(d)} \in \mathbb{R}^{C\times L_d}$ is a loadings matrix for data type d. A noise tensor for each data type is given by $\mathcal{E}^{(d)} \in \mathbb{R}^{N\times L_d \times T_d}$. Each data tensor is decomposed using equation (1), with the constraint that a single individual scores matrix is common across all data types. In practice, if $T_d = 1$ for a data type d, then $B^{(d)}$ has dimensions 1 by C and is fixed to a vector of ones during inference.

Again, spike and slab priors are used for the loadings matrices to encourage sparsity. Updates for the loadings and context scores matrices for a data type d are effectively identical to the (single) tensor decomposition already considered. Importantly, updates for $X^{(d)}$ and $B^{(d)}$ do not depend on $X^{(d')}$ and $B^{(d')}$ for any $d' \neq d$. The update for A is dependent on all other current parameter estimates. One way to think about this update is that it averages over the estimates for A that one would get if performing separate decompositions for each data type. (In reality this is not quite the case because the prior also needs to be considered.)

It is important to note that equation (37) can model a variety of different types of underlying structure in the data. Components can be shrunk to zero for a particular data type allowing for the model to capture signals that exist in an arbitrary subset of the data. For example, in Supplementary Figure 38, the yellow component is active in only data types 2 and 3.

This linked tensor decomposition is a generalisation of several models. In particular, the single 3D tensor decomposition we focus on in this paper is recovered if $D = 1$. If $T_1 = D = 1$, then the model collapses to sparse factor analysis. Group factor analysis is recovered if $T_d = 1$ for all d.