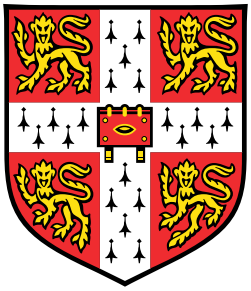
Statistical methods for the integrative analysis of single-cell multi-omics data



Ricard Argelaguet

European Bioinformatics Institute
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Robinson College

April 2020

Chapter 1

MOFA+: an improved framework for the comprehensive integration of structured single-cell data

In Chapter 2 we developed Multi-Omics Factor Analysis (MOFA), a statistical framework for the unsupervised integration of multi-modal data.
MOFA addresses key challenges in data integration, including overfitting, noise reduction, handling of missing values and improved interpretation of the model output. Hower, when applied to increasingly-large (single-cell) data sets, the inference scheme implemented in MOFA is still limited in scalability. In addition, the increased experimental throughput has facilitated the simultaneous study of multiple experimental conditions, even in a combinatorial fashion[**Replogle2020**].
MOFA makes strong assumptions about the dependencies across samples and it hence has no principled way of modelling data sets where the samples are structured into multiple groups, where groups can correspond to batches, donors or independent studies. By pooling and contrasting information across experimental conditions, it would be possible to obtain more comprehensive insights into the complexity underlying biological systems.

In this new Chapter we improve the model formulation in MOFA with the aim of performing integrative analysis of large-scale datasets that are *structured* into multiple data modalities and/or multiple groups.

1.1 Theoretical fundations

1.1.1 Exponential family distributions

Exponential family distributions are a parametric class of probability distributions that have characteristic mathematical properties which make them amenable for probabilistic modelling.
The majority of commonly used probability distributions belong to the exponential family, including the normal or Gaussian, Gamma, Poisson, Bernoulli, Exponential, etc. Exponential family

distributions can be represented in the following form:

$$p(\mathbf{x}|\boldsymbol{\theta}) = h(\mathbf{x}) \exp\{\eta(\boldsymbol{\theta})T(\mathbf{x}) - A(\boldsymbol{\theta})\} \tag{1.1}$$

where $\mathbf{x}$ is a multivariate random variable and $\boldsymbol{\theta}$ are the distribution's parameters. Each term has a common notation: T($\mathbf{x}$): sufficient statistics; $\eta(\boldsymbol{\theta})$: natural parameters; $h(\mathbf{x})$: base mesure; $A(\eta)$: the log-partition function (or the normaliser).

The exponential family form for the probability distributions frequently used in this thesis are the following:

Univariate normal distribution:

$$\begin{aligned}
&\eta(\mu, \sigma) = [\frac{\mu}{\sigma^2}; -\frac{1}{2\sigma^2}] \\
&h(x) = \frac{1}{\sqrt{2\pi}} \\
&T(x) = [x; x^2] \\
&A(\mu, \sigma) = \frac{\mu^2}{2\sigma^2} + \log \|\sigma\|
\end{aligned}$$

Multivariate normal distribution:

$$\begin{aligned}
&\eta(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = [\Sigma^{-1}\mu; -0.5\Sigma^{-1}] \\
&T(x) = [x; xx^T] \\
&h(x) = (2\pi)^{-\frac{k}{2}} \\
&A(\theta) = -0.25\eta_1^T\eta_2 - 1\eta_1 - 0.5\log(\| - 2\eta_2\|)
\end{aligned}$$

Gamma distribution:

$$\begin{aligned}
&\eta = [\alpha - 1; -\beta] \\
&T(x) = [\log x; x] \\
&h(x) = 1 \\
&A(\theta) = \log(\Gamma(\eta_1 + 1)) - (\eta_1 + 1)\log(-\eta_2)
\end{aligned}$$

Beta distribution:

$$\begin{aligned}
&\eta = [\alpha; \beta] \\
&T(x) = [\log x; \log(1 - x)] \\
&h(x) = \frac{1}{x(1 - x)} \\
&A(\theta) = \log(\Gamma(\eta_1)) + \log(\Gamma(\eta_2)) - \log(\Gamma(\eta_1 + \eta_2))
\end{aligned}$$

In the context of Bayesian inference, the main property that make exponential family distributions indispensable is that they have conjugate priors (i.e. a combination of likelihood and prior distributions which ensure a closed-form posterior distribution which is of the same form as the prior). As we have discussed in Chapter 2, this property is crucial for enabling efficient statistical inference, otherwise posterior distributions must be computed using expensive and approximate numerical methods.

1.1.2 Gradient ascent

Gradient ascent is a first-order optimization algorithm for finding the maximum of a function [**Bishop2006**, **Murphy**]. Formally, for a differentiable function $F(x)$, the iterative scheme of gradient ascent is:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \rho^{(t)} \nabla F(\mathbf{x}^{(t)}) \tag{1.2}$$

In short, it works by taking steps proportional to the gradient ∇F evaluated at each iteration t. Importantly, the step size $\rho^{(t)}$ is typically adjusted at each iteration t such that it satisfies the Robbins-Monro conditions: $\sum_t \rho^{(t)} = \infty$ and $\sum_t (\rho^{(t)})^2 < \infty$. Then F is guaranteed to converge to the global maximum [**Robbins-Monro1951**] if the objective function is convex. If F is not convex, the algorithm is sensible to the initialisation $\mathbf{x}^{t=0}$ and can converge to local maxima instead of the global maximum.

1.1.2.1 Stochastic gradient ascent

Gradient ascent becomes prohibitively slow with large datasets, mainly because of the computational cost involved in the iterative calculation of gradients [**Spall2003**].
A simple strategy to speed up gradient ascent is to replace the actual gradient ∇F by an estimate $\hat{\nabla} F$ using a randomly selected subset of the data (minibatch). The iterative scheme is then defined in the same way as in standard gradient ascent:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \rho^{(t)} \hat{\nabla} F(\mathbf{x}^{(t)}) \tag{1.3}$$

1.1.2.2 Natural gradient ascent

Gradient ascent becomes problematic when applied to probabilistic models. To give the intuition, consider a probabilistic model with a hidden variable x and corresponding parameters θ, with a general objective function $\mathcal{L}(\theta)$. From the definition of a derivative:

$$\nabla \mathcal{L}(\theta) = \lim_{||h|| \to 0} \frac{\mathcal{L}(\theta + h) - \mathcal{L}(\theta)}{||h||}$$

where h represents an infinitesimally small positive step in the space of θ.
To find the direction of steepest ascent, one would need to search over all possible directions d in an

infinitely small distance h, and select the $\hat{d}$ that gives the largest gradient:

$$\nabla\mathcal{L}(\theta) = \lim_{h\to 0}\frac{1}{h}\underset{d\,s.t.\|d\|=h}{\arg\max}\ \mathcal{L}(\theta+d) - \mathcal{L}(\theta)$$

Importantly, this operation requires a distance metric to quantify what a *small* distance h means. In standard gradient ascent, this is measured using an Euclidean norm, and the direction of steepest ascent is hence dependent on the Euclidean geometry of the θ space. Why is this problematic when working with probability distributions? Because it does not consider the uncertainity that underlies probability distributions. A small step from $\theta^{(t)}$ to $\theta^{(t+1)}$ does not guarantee an equivalently small change from $\mathcal{L}(\theta^{(t)})$ to $\mathcal{L}(\theta^{(t+1)})$.
To illustrate this, consider the following example of four random variables

$$\begin{aligned}\psi_1 &\sim \mathcal{N}(0\,|\,5) \qquad \psi_3 \sim \mathcal{N}(0\,|\,1)\\ \psi_2 &\sim \mathcal{N}(10\,|\,5) \qquad \psi_4 \sim \mathcal{N}(10\,|\,1)\end{aligned} \tag{1.4}$$

Using the Euclidean metric, the distance between ψ_1 and ψ_2 is the same as the distance between ψ_3 and ψ_4. However, the distance in distribution space (measured for example by the KL divergence) is much larger between ψ_1 and ψ_2 than between ψ_3 and ψ_4:

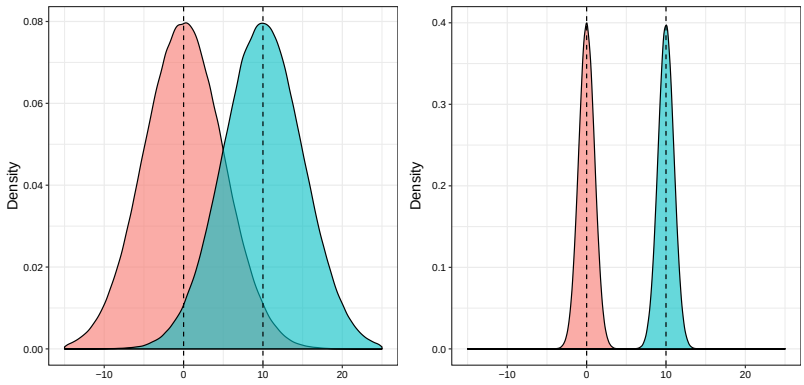


Figure 1.1: Illustration of the problem of using Euclidean distances to measure distances between parameters of distributions.
In both plots, the red and blue distributions are separated by the same Euclidean distance of 100. Yet, the distance in probability space between the two distributions is higher in the right.

This basic simulation suggests that replacing the Euclidean distance by the KL divergence as a distance metric may be more appropriate in the context of probabilistic modelling:

$$\nabla_{KL}\mathcal{L}(\theta) = \lim_{h\to 0}\frac{1}{h}\underset{d\,s.t.KL[p_\theta||p_{\theta+d}]=h}{\arg\max}\ \mathcal{L}(\theta+d) - \mathcal{L}(\theta)$$

The direction of steepest ascent measured by the KL divergence is called the natural gradient [**Amari1998**, **Martens2014**].

To find the optimal $\hat{d}_{KL}$, one needs to solve the following optimisation problem:

$$\arg\min_{d} \mathcal{L}(\theta + d) \qquad \text{subject to} \quad KL[p_\theta || p_{\theta+d}] < c$$

where c is an arbitrary constant. Previous works have shown that this can be solved by introducing Lagrange multipliers and Taylor expansions (see **[Amari1998**, **Kristiadi2019]**). The solution corresponds to the standard (Euclidean) gradient pre-multiplied by the inverse of the Fisher Information Matrix of $q(x|\theta)$:

$$\hat{d}_{KL} \propto \mathbf{F}^{-1}(\theta)\nabla_\theta \mathcal{L}(\theta) \tag{1.5}$$

where $\mathbf{F}(\theta)$ is defined as

$$\mathbf{F}(\theta) = \mathbb{E}_{q(x|\theta)}[(\nabla_\theta \log q(x|\theta))(\nabla_\theta \log q(x|\theta))^T]$$

In conclusion, while the standard gradient points to the direction of steepest ascent in Euclidean space, the natural gradient points to the direction of steepest ascent in a space where distances are defined by the KL divergence **[Kristiadi2019**, **Amari1998**, **Hoffman2012]**.

1.1.3 Derivation of a stochastic variational inference algorithm

In this section I will show how to derive a stochastic variational inference algorithm for general Bayesian models. This work is inspired from **[Hoffman2012]** which we adapted and implemented in the MOFA+ model. A comprehensive mathematical derivation of the algorithm is not sought in this chapter. Instead, I will describe a modified and simplified derivation to gist the essential. For a complete mathematical derivation we refer the reader to **[Hoffman2012]**.
Also, this section builds upon three theoretical fundations that have been introduced before: Variational inference (**??**), exponential family distributions (Section 1.1.1) and (natural) gradient ascent (Section 1.1.2).

1.1.3.1 Model definition

Consider a probabilistic model with a set of unobserved random variables, observations and (non-random) parameters. We begin by classifying the variables of the model into four different categories:

- observations ($\mathbf{Y}$): N different vectors $\mathbf{y}_n$, each one containing the observed variables for the n-th sample.
- local (hidden) variables ($\mathbf{Z}$): N different vectors $\mathbf{z}_n$, each one containing K hidden variables associated with the n-th sample.
- global (hidden) variables ($\boldsymbol{\beta}$): one vector that contains B hidden variables not indexed by n.
- parameters (non-random) for the global variables ($\boldsymbol{\alpha}_\beta$).
- parameters (non-random) for the local variables ($\boldsymbol{\alpha}_z$).

This leads to the following factorisation of the joint distribution:

$$p(\mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha}_\beta, \boldsymbol{\alpha}_z) = p(\mathbf{Z}|\boldsymbol{\alpha}_z)p(\boldsymbol{\beta}|\boldsymbol{\alpha}_\beta) \prod_{n=1}^{N} p(\mathbf{y}_n|\mathbf{z}_n, \boldsymbol{\beta}) \tag{1.6}$$

and the corresponding graphical model representation:

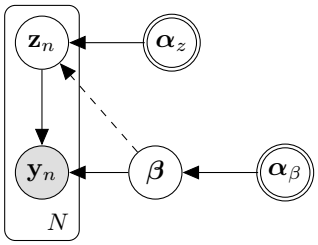


Figure 1.2: Graphical model for a general probabilistic model where unobserved variables are classified as global and local.
The dashed line indicates that the connection between global and local variables is optional and it is not used in the MOFA model.

Notice that the difference between local and global variables lies on the conditional dependency assumptions. The local variables for the n-th sample $\mathbf{z}_n$ are conditionally independent from any other observation $\mathbf{y}_j$ or local variable $\mathbf{z}_j$ (where $j \neq n$), given that the global variables β are observed:

$$p(\mathbf{y}_n, \mathbf{z}_n|\mathbf{y}_j, \mathbf{z}_{nj}, \boldsymbol{\beta}, \boldsymbol{\alpha}_\beta, \boldsymbol{\alpha}_{z_n}, \boldsymbol{\alpha}_{z_j}) = p(\mathbf{y}_n, \mathbf{z}_n|\boldsymbol{\beta}, \boldsymbol{\alpha}_\beta, \boldsymbol{\alpha}_{z_n})$$

To relate this formulation to the MOFA model, the local variables would contain the factors whereas the global variables would contain the feature weights.

For simplicity in the derivation, we will assume the existence of a single global variable β, a single parameter α_β for the global variables and a single parameter $\alpha_{z_{nk}}$ for each local variable.

The first assumption in the model is that the prior distributions of the local and global variables are members of the exponential family (see Equation (1.1))

$$\begin{aligned} p(\beta|\alpha_\beta) &= h(\beta)\exp\{\eta_g(\alpha_\beta)t(\beta) - a_g(\alpha_\beta)\} \\ p(z_{nk}|\alpha_z) &= h(z_{nk})\exp\{\eta_l(\alpha_z)t(z_{nk}) - a_l(\alpha_z)\} \end{aligned} \tag{1.7}$$

The second assumption is that the complete conditionals of the unobserved variables are also members of the exponential family:

$$\begin{aligned} p(\beta|\mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha}) &= h(\beta)\exp\{\eta_g(\mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha})^T t(\beta) - a_g(\eta_g(\mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha}))\} \\ p(\mathbf{z}_n|\mathbf{y}_{nj}, \mathbf{z}_{nj}, \beta) &= h(\mathbf{z}_n)\exp\{\eta_l(\mathbf{y}_{nj}, \mathbf{z}_{nj}, \beta)^T t(\mathbf{z}_n) - a_l(\eta_l(\mathbf{y}_{nj}, \mathbf{z}_{nj}, \beta))\} \end{aligned} \tag{1.8}$$

1.1.3.2 Setting up the inference problem

First, we set up the variational distributions for both the local variables and the global variables. Here we are going to assume that all unobserved variables are independent (mean-field assumption)

$$q(\mathbf{z}, \beta) = q(\beta|\lambda) \prod_{n=1}^{N} \prod_{k=1}^{K} p(z_{nk}|\phi_{nk})$$

and belong to the same exponential family as the corresponding prior distribution:

$$q(\beta|\lambda) = h(\beta) \exp\{\eta_g(\lambda) t(\beta) - a_g(\lambda)\} \tag{1.9}$$

$$q(z_{nk}|\phi_{nk}) = h(z_{nk}) \exp\{\eta_l(\phi_n) t(z_{nk}) - a_l(z_{nk})\} \tag{1.10}$$

where λ are the parameters governing the variational distribution for the global variables and ϕ_{nk} are the parameters governing the variational distribution for the k-th local variable and the n-th sample.

From the assumptions above, the ELBO (the objective function in variational inference, see **??**) factorises as:

$$\begin{aligned} \mathcal{L} &= \mathbb{E}_{q(\mathbf{Z},\beta)}[\log p(\mathbf{Y}, \mathbf{Z}, \beta)] - \mathbb{E}_{q(\mathbf{Z})}[\log q(\mathbf{Z})] - \mathbb{E}_{q(\beta)}[\log q(\beta)] \\ &= \sum_{n=1}^{N} \mathbb{E}_{q(\mathbf{z}_n,\beta)}[\log p(\mathbf{y}_n, \mathbf{z}_n, \beta)] - \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{E}_{q(z_{nk})}[\log q(z_{nk})] - \mathbb{E}_{q(\beta)}[\log q(\beta)] \end{aligned} \tag{1.11}$$

Notice that the objective decomposes into global terms (not involving N) and local terms (involving N). The local terms can be approximated using estimates of the gradient by subsampling the data set. Assumign a mini-batch of size S:

$$\hat{\mathcal{L}} = \frac{N}{S} \sum_{n=1}^{S} \mathbb{E}_{q(\mathbf{z}_n,\beta)}[\log p(\mathbf{y}_n, \mathbf{z}_n, \beta)] - \frac{N}{S} \sum_{s=1}^{S} \sum_{k=1}^{K} \mathbb{E}_{q(z_{nk})}[\log q(z_{nk})] - \mathbb{E}_{q(\beta)}[\log q(\beta)]$$

If the samples are independent then the expectation of this noisy gradient is equal to the true gradient, which is the main principle of stochastic optimisation.

Now that the optimisation problem is defined, the next step is to derive an iterative algorithm to find the values of the variational parameters that maximise the ELBO.

1.1.3.3 Calculating the gradient for the global parameters

To derive the updates for the global parameters we first write the ELBO in terms of λ:

$$\mathcal{L}(\lambda) = \mathbb{E}_{q(z,\beta)}[\log p(\beta|\mathbf{Y}, \mathbf{Z})] - \mathbb{E}_{q(\beta)}[\log q(\beta)] + \text{const.}$$

where the constant term captures all quantities that do not depend on β. Then, from the assumption that the complete conditionals and the variational distributions belong to the exponential family (Equations (1.8) to (1.9)):

$$\begin{aligned}\mathcal{L}(\lambda) &= \mathbb{E}_{q(z,\beta)}[\eta_g(\mathbf{Y},\mathbf{Z},\boldsymbol{\alpha})^T t(\beta)] - \mathbb{E}_{q(\beta)}[\lambda^T t(\beta) - a_g(\lambda)] + \text{const.} \\ &= \mathbb{E}_{q(z)}[\eta_g(\mathbf{Y},\mathbf{Z},\boldsymbol{\alpha})^T]\nabla a(\lambda) - \lambda^T \nabla a_g(\lambda) - a_g(\lambda) + \text{const.}\end{aligned}$$

where we have used the exponential family identity $\mathbb{E}_{q(\beta)}[t(\beta)] = \nabla a_g(\lambda)$.

Taking the gradient with respect to λ:

$$\nabla_\lambda \mathcal{L}(\lambda) = \nabla^2_\lambda a_g(\lambda)(\mathbb{E}_{q(z)}[\eta_g(\mathbf{Y},\mathbf{Z},\boldsymbol{\alpha})] - \lambda) \tag{1.12}$$

and setting it to zero leads to the solution:

$$\lambda = \mathbb{E}_{q(z)}[\eta_g(\mathbf{Y},\mathbf{Z},\boldsymbol{\alpha})] \tag{1.13}$$

1.1.3.4 Calculating the gradient for the local parameters

Turning to the local parameters, as a function of ϕ_{nk} the ELBO becomes:

$$\mathcal{L}(\phi_{nk}) = \mathbb{E}_{q(\beta,\mathbf{z}_{nj})}[\log p(\mathbf{z}_{nj}|\mathbf{y}_n,\mathbf{z}_{nj},\beta)] - \mathbb{E}_{q(z_{nk})}[\log q(z_{nk})] + \text{const.}$$

Again, from the assumption that the complete conditionals and the variational distributions belong to the exponential family (Equations (1.8) to (1.9)):

$$\begin{aligned}\mathcal{L}(\phi_{nk}) &= \mathbb{E}_{q(\beta,\mathbf{z}_{nj})}[\eta_l(\mathbf{y}_n,\mathbf{z}_{nj},\beta)^T t(\mathbf{z}_{nj})] - \mathbb{E}_{q(z_{nk})}[\phi_{nk} t(z_{nk}) - a_l(\phi_{nk})] + \text{const.} \\ &= \mathbb{E}_{q(\beta,\mathbf{z}_{nj})}[\eta_l(\mathbf{y}_n,\mathbf{z}_{nj},\beta)]^T \nabla a_l(\phi_{nk}) - \phi_{nk}\nabla a_l(\phi_{nk}) - a_l(\phi_{nk}) + \text{const.}\end{aligned}$$

Taking the gradient with respect to ϕ_{nk}:

$$\nabla_\phi \mathcal{L}(\phi_{nk}) = \nabla^2_\phi a_l(\phi_{nk})(\mathbb{E}_{q(\beta,\mathbf{z}_{nj})}[\eta_l(\mathbf{y}_n,\mathbf{z}_{nj},\beta)] - \phi_{nk}) \tag{1.14}$$

and setting it to zero leads to the following solution:

$$\phi_{nk} = \mathbb{E}_{q(\beta,\mathbf{z}_{nj})}[\eta_l(\mathbf{y}_n,\mathbf{z}_{nj},\beta)] \tag{1.15}$$

1.1.3.5 Coordinate ascent variational inference algorithm

Now that we have the gradients for both the local and the global parameters, we can define a gradient ascent algorithm to optimise the model:

However, as discussed in Section 1.1.2.2, the use of euclidean-based gradients ignores important information about the geometry of the distribution and is thus not optimal for the optimisation of

Algorithm 1 Coordinate ascent mean-field variational inference algorithm

Initialise the global parameters $\boldsymbol{\lambda}^{(t=0)}$ randomly
repeat
  for each local variational parameter ϕ_{nk} **do**
    $\phi_{nk}^{(t)} \leftarrow \mathbb{E}_{q(\boldsymbol{\beta},\mathbf{z}_{nj})}[\eta_l(\mathbf{y}_n, \mathbf{z}_{nj}, \boldsymbol{\beta})]$
  end for
  for each global variational parameter λ **do**
    $\lambda(t) = \mathbb{E}_{q(z)}[\eta_g(\mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha})]$
  end for
until Convergence

probabilistic models. Next, we will derive a similar coordinate ascent algorithm but using instead the natural gradient.

1.1.3.6 Deriving the natural gradients for the global variational parameters

From Equation (1.12), the gradient of the ELBO with respect to the global parameters λ is:

$$\nabla_\lambda \mathcal{L}(\lambda) = \nabla^2_\lambda a_g(\lambda)(\mathbb{E}_{q(z)}[\eta_g(\mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha})] - \lambda)$$

Premultiplying by $\mathbf{F}(\beta)^{-1} = \nabla^2_\lambda a_g(\lambda)$ gives the natural gradient for the global parameters:

$$\hat{\nabla}_\lambda \mathcal{L}(\lambda) = \mathbb{E}_{q(z)}[\eta_g(\mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha})] - \lambda$$

1.1.3.7 Deriving the natural gradients for the local variational parameters

From equation X, the gradient of the ELBO with respect to the local parameters $\boldsymbol{\phi}$ is:

$$\nabla_\phi \mathcal{L}(\phi_{nk}) = \nabla^2_\phi a_l(\phi_{nk})(\mathbb{E}_{q(\beta,\mathbf{z}_{nj})}[\eta_l(\mathbf{y}_n, \mathbf{z}_{nj}, \beta)] - \phi_{nk})$$

Premultiplying by $\mathbf{F}(z_{nk})^{-1} = \nabla^2_\phi a_l(\phi_{nk})$ gives the natural gradient for the global parameters:

$$\hat{\nabla}_\phi \mathcal{L}(\phi_{nk}) = \mathbb{E}_{q(\beta,\mathbf{z}_{nj})}[\eta_l(\mathbf{y}_n, \mathbf{z}_{nj}, \beta)] - \phi_{nk}$$

Remarkably, the natural gradient for both the local and global variational parameters is simply the standard gradient subtracting the current value of the parameters. Hence, the Fisher Information matrix does *not* need to be explicitly computed at each iteration, which this leads to a considerable simplification of the problem.

1.1.4 Hyperparameters

- **Batch size**: controls the number of samples that are used to compute the gradients at each iteration. A trade off exists where high batch sizes lead to a more expensive computation of the gradient but yield a less noisy estimate.

- **Learning rate**: The learning rate $p(t)$ controls the step size in the direction of the natural gradient, with high learning rates leading to higher steps. In the natural gradient setting, the learning rate also controls how much memory from previous iterations is translated to the current updates. The particular case of a constant learning rate of 1 yields no memory from previous iterations (thus simplifies to standard gradient ascent). To ensure proper convergence, the learning rate has to be decayed during training. Several strategies exist[**Ranganath2013**], here we used the simple function $\rho(t) = \frac{\rho_0}{(1+\kappa t)^{3/4}}$, which introduces two extra hyperparameters: (1) The forgetting rate κ, which controls the decay of the learning rate, and ρ_0 which determines the initial learning rate.

1.2 Model description

In MOFA v2 we generalise the model to a disjoint set of M input views (i.e. groups of features) and G input groups (i.e. groups of samples).

The data is factorised according to the following model:

$$\mathbf{Y}_g^m = \mathbf{Z}_g \mathbf{W}^{mT} + \boldsymbol{\epsilon}_g^m \tag{1.16}$$

where $\mathbf{Z}_g \in \mathbb{R}^{N_g \times K}$ are a set of G matrices that contains the factor values for the g-th group and $\mathbf{W}^m \in \mathbb{R}^{D_m \times K}$ are a set of M matrices that define the feature weights for the m-th view. $\boldsymbol{\epsilon}_g^m \in \mathbb{R}^{D_m}$ captures the residuals, or the noise for each feature in ech group.

EXPLAIN INTUITION REGRESSED OUT GROUP EFFECT

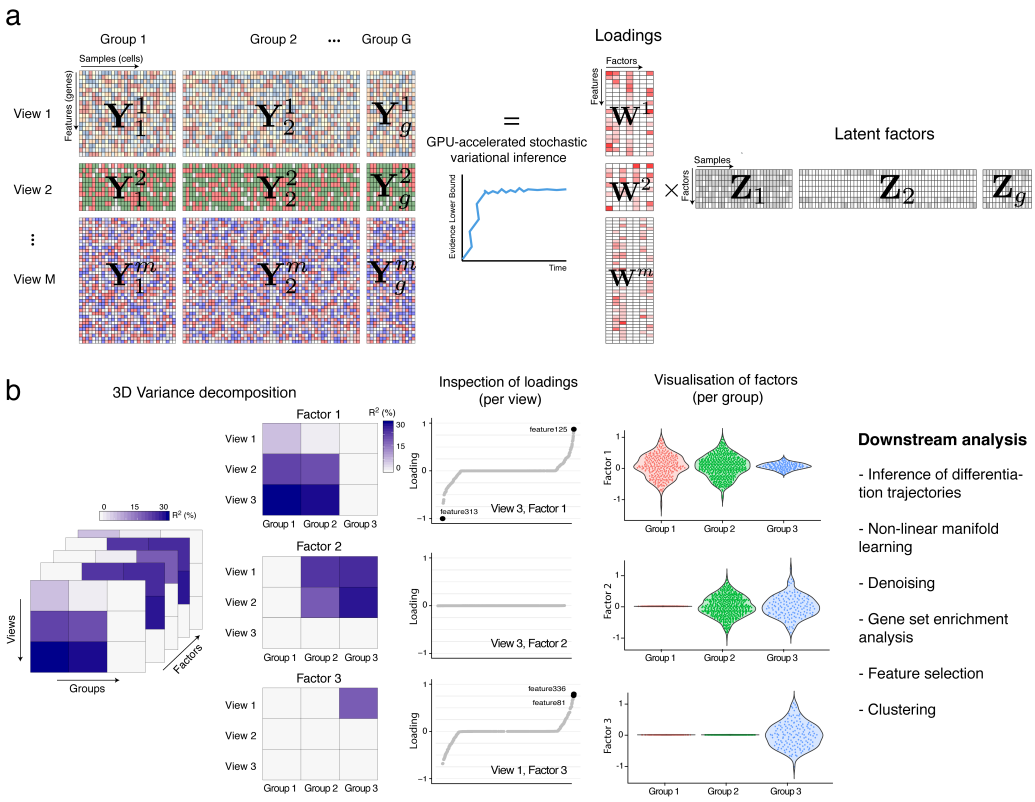


Figure 1.3:
Multi-Omics Factor Analysis v2 (MOFA+) provides an unsupervised framework for the integration of multi-group and multi-view single-cell data.
(a) Model overview: the input consists of multiple data sets structured into M views and G groups. Views consist of non-overlapping sets of features that can represent different assays. Analogously, groups consist of non-overlapping sets of samples that can represent different conditions or experiments. Missing values are allowed in the input data. MOFA+ exploits the dependencies between the features to learn a low-dimensional representation of the data (Z) defined by K latent factors that capture the global sources of molecular variability. For each factor, the weights (W) link the high-dimensional space with the low-dimensional manifold and provide a measure of feature importance. The sparsity-inducing priors on both the factors and the weights enable the model to disentangle variation that is unique to or shared across the different groups and views. Model inference is performed using GPU-accelerated stochastic variational inference.
(b) The trained MOFA+ model can be queried for a range of downstream analyses: 3D variance decomposition, quantifying the amount of variance explained by each factor in each group and view, inspection of feature weights, visualisation of factors and other applications such as clustering, inference of non-linear differentiation trajectories, denoising and feature selection.

1.2.1 Model priors and likelihood

1.2.1.1 Prior on the weights

This remains the same as in MOFA v1. We adopt a two-level sparsity prior with an Automatic Relevance Determination per factor and view, and a feature-wise spike-and-slab prior (reparametrised[**Titsias2011**]):

$$p(\hat{w}^m_{dk}, s^m_{dk}) = \mathcal{N}\left(\hat{w}^m_{dk} \,|\, 0, 1/\alpha^m_k\right) \mathrm{Ber}(s^m_{dk} \,|\, \theta^m_k) \tag{1.17}$$

with the corresponding conjugate priors for θ and α:

$$p(\theta_k^m) = \text{Beta}\left(\theta_k^m \,|\, a_0^\theta, b_0^\theta\right) \tag{1.18}$$

$$p(\alpha_k^m) = \mathcal{G}\left(\alpha_k^m \,|\, a_0^\alpha, b_0^\alpha\right) \tag{1.19}$$

The aim of the ARD prior is to disentangle the activity of factors to the different views, such that the weight vector $\mathbf{w}_{:,k}^m$ is shrunk to zero if the factor k does not explain any variation in view m. The aim of the spike-and-slab prior is to push individual weights to zero to yield a more interpretable solution.
For more details, we refer the reader to Chapter 2.

1.2.1.2 Prior on the factors

In MOFA v1 we adopted an isotropic Gaussian prior:

$$p(z_{nk}) = \mathcal{N}\left(z_{nk} \,|\, 0, 1\right) \tag{1.20}$$

which assumes *a priori* an unstructured latent space. This is the assumption that we want to break. Following the same logic as in the factor and view-wise ARD prior, the integration of multiple groups of samples requires introducing a *structured* prior that captures the existence of different groups, such that some factors are allowed to be active in different subsets of groups.

To formalise the intuition above we simply need to copy the double sparsity prior from the weights to the factors:

$$p(\hat{z}_{nk}^g, s_{nk}^g) = \mathcal{N}(\hat{z}_{nk}^g \,|\, 0, 1/\alpha_k^g)\,\text{Ber}(s_{nk}^g \,|\, \theta_k^g) \tag{1.21}$$

$$p(\theta_k^g) = \text{Beta}\left(\theta_k^g \,|\, a_0^\theta, b_0^\theta\right) \tag{1.22}$$

$$p(\alpha_k^g) = \mathcal{G}\left(\alpha_k^g \,|\, a_0^\alpha, b_0^\alpha\right), \tag{1.23}$$

where g is the index of the sample groups.
Notice that the spike-and-slab prior is introduced for completeness but is not necessarily required, and can be disabled by fixing $\mathbb{E}[\theta_k^g] = 1$.

1.2.1.3 Prior on the noise

The variable $\boldsymbol{\epsilon}$ captures the residuals, or the noise, which is assumed to be normally distributed and heteroskedastic. In MOFA v2 we generalise the noise to have an estimate per individual feature and per group:

$$p(\epsilon_g^m) = \mathcal{N}\left(\epsilon_g^m \,|\, 0, /\tau_g^m \mathbf{I}_{Dm}\right) \tag{1.24}$$

$$p(\tau_g^m) = \prod_{d=1}^{D_m} \mathcal{G}\left(\tau_g^m \,|\, a_0^\tau, b_0^\tau\right) \tag{1.25}$$

This formulation is important to capture the (realistic) events where a specific feature may be highly variable in one group but non-variable in another group.
In addition, as in MOFA v1, non-gaussian noise models can also be defined, but unless otherwise stated, we will always assume Gaussian residuals.

1.2.1.4 Graphical model

In summary, the updated model formulation introduces asymmetric sparsity prior in both the weights and the factors, which enables the model to simultaneously integrate multiple views as well as multiple groups of samples:

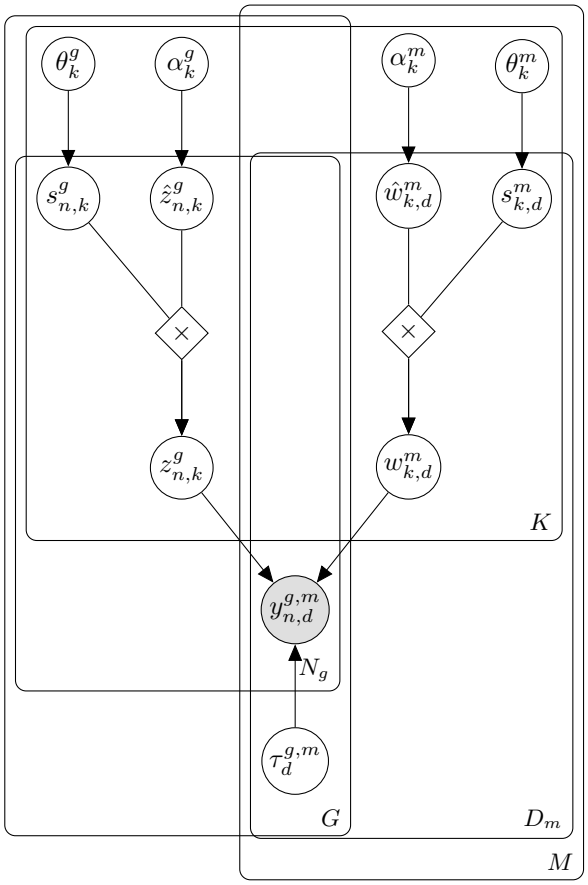


Figure 1.4: Graphical model for MOFA+$^{\text{oo}}$ .The white circles represent hidden variables that are inferred by the model, whereas the grey circles represent the observed variables. There are a total of five plates, each one representing a dimension of the model: M for the number of views, G for the number of groups, K for the number of factors, D_m for the number of features in view m and N_g for the number of samples in group g

1.2.2 Solving the rotational invariance problem

Conventional Factor Analysis is invariant to rotation in the latent space[**Zhao2009**]. To demonstrate this property, let us apply an arbitrary rotation to the loadings and the factors, specified by the

rotation matrix $\mathbf{R} \in \mathbb{R}^{K \times K}$:

$$\tilde{\mathbf{Z}} = \mathbf{Z}\mathbf{R}^{-1}$$
$$\tilde{\mathbf{W}} = \mathbf{R}\mathbf{W}$$

First, note that the model likelihood is unchanged by this rotation, irrespective of the prior distribution used.

$$p(\mathbf{Y}|\tilde{\mathbf{Z}}\tilde{\mathbf{W}}, \tau) = p(\mathbf{Y}|\mathbf{Z}\mathbf{R}^{-1}\mathbf{R}\mathbf{W}, \tau) = p(\mathbf{Y}|\mathbf{Z}\mathbf{W}, \tau)$$

However, the prior distributions of the factors and the loadings are only invariant to rotations when using isotropic Normal priors:

$$\ln p(\mathbf{W}) \propto \sum_{k=1}^{K}\sum_{d=1}^{D} w_{d,k}^2 = \mathrm{Tr}(\mathbf{W}^T\mathbf{W}) = \mathrm{Tr}(\mathbf{W}^T\mathbf{R}^{-1}\mathbf{R}\mathbf{W}) = \mathrm{Tr}(\tilde{\mathbf{W}}^T\tilde{\mathbf{W}})$$

where we have used the property $\mathbf{R}^T = \mathbf{R}^{-1}$ that applies to rotation matrices. The same derivation follows for the factors $\mathbf{Z}$.
In practice, this property renders conventional Factor Analysis unidentifiable, hence limiting its interpretation and applicability. Sparsity assumptions, however, partially address the rotational invariance problem [**Hore2015**].

It is important to remark that the factors are nonetheless invariant to permutations. This implies that under different initial conditions, the order of the factors is not necessarily the same in independent model fittings. To address this we manually sort factors *a posteriori* based on total variance explained.

1.3 Model validation

We validated the new features of MOFA+ using simulated data drawn from its generative model.

1.3.1 Stochastic variational inference

We simulated data with varying sample sizes, with the other dimensions fixed to $M = 3$ views, $G = 3$ groups, $D = 1000$ features (per view), and $K = 25$ factors.

We trained a set of models with (deterministic) variational inference (VI) and a set of models with stochastic variational inference (SVI). Overall, we observe that SVI yields Evidence Lower Bounds that matched those obtained from conventional inference across a range of batch sizes, learning rates and forgetting rates.
In terms of speed, GPU-accelerated SVI inference was up to ≈ 20x faster than VI, with speed differences becoming more pronounced with increasing number of cells. For completeness, we also compared the the convergence time estimates for SVI when using CPU versus GPU. We observe that for large sample sizes there is a speed improvement even when using CPUs, although these advantages become more prominent when using GPUs.

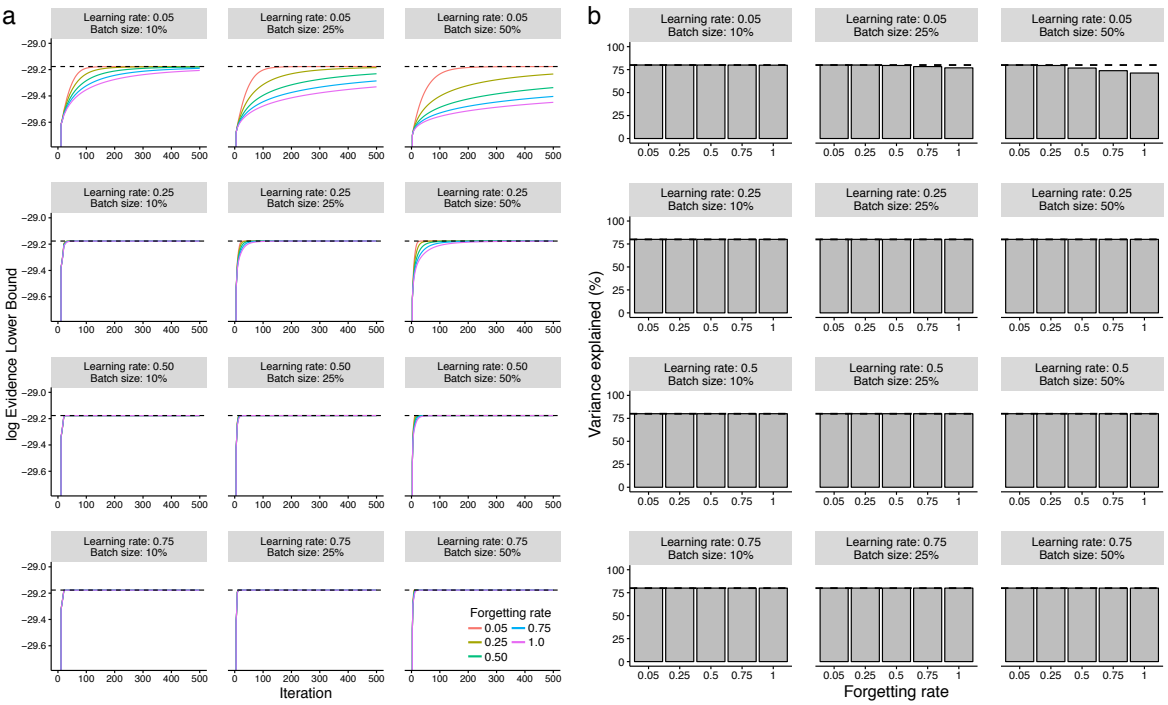


Figure 1.5: Validation of stochastic variational inference using simulated data.
(a) Line plots display the iteration number of the inference (x-axis) and the log- Evidence Lower Bound (ELBO) on the y-axis. Panels correspond to different values of batch sizes (10%, 25%, 50% of the data) and initial learning rates (0.05, 0.25, 0.5, 0.75). Colors correspond to different forgetting rates (0.05, 0.25, 0.5, 0.75, 1.0). The dashed horizontal line indicates the ELBO achieved using standard VI.
(b) Bar plots display the forgetting rate (x-axis) and the total variance explained (%) in the y-axis. Panels correspond to different values of batch sizes (10%, 25%, 50% of the data) and initial learning rates (0.05, 0.25, 0.5, 0.75). The dashed line indicates the variance explained achieved using standard VI.

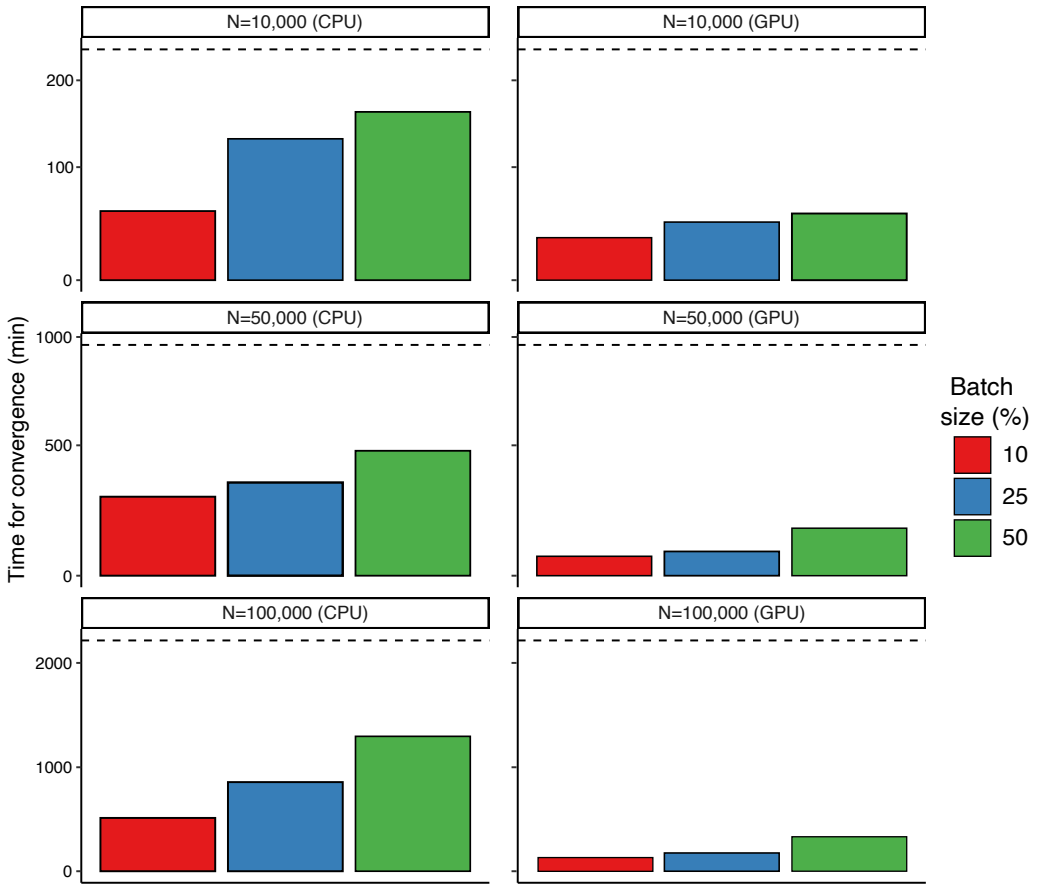


Figure 1.6: Evaluation of convergence speed for stochastic variational inference using simulated data.
Bar plots show the time elapsed for training MOFA+ models with stochastic variational inference (SVI). Colors represent different batch sizes (10%, 25% or 50%). The dashed line indicates the training time for standard VI.
VI models were trained using a single E5-2680v3 CPU. SVI models were trained either using a single E5-2680v3 CPU (first column) or using an Nvidia GTX 1080Ti GPU (second column).

1.3.2 Multi-group structure

Finally, we evaluated whether the double view and group-wise sparsity prior enables the detection of factors with simultaneous differential activity between groups and views.
We simulated data with the following parameters: $M = 2$ modalities, $G = 2$ groups, $D = 1000$ features, $N = 1000$ samples and $K = 10$ factors. Differential factor activities are incorporated in the simulation process by turning some factors off in random sets of modalities and groups (Figure 1.7, see ground truth). The task is to recover the true factor activity structure given a random initialisation.
We fit three models: Bayesian Factor Analysis (no sparsity priors), MOFA v1 (only view-wise sparsity prior) and MOFA+ (view-wise and group-wise sparsity prior). Indeed, we observe that when having factors that explain differing amounts of variance across groups and across views, MOFA+ was able to more accurately reconstruct the true factor activity patterns:

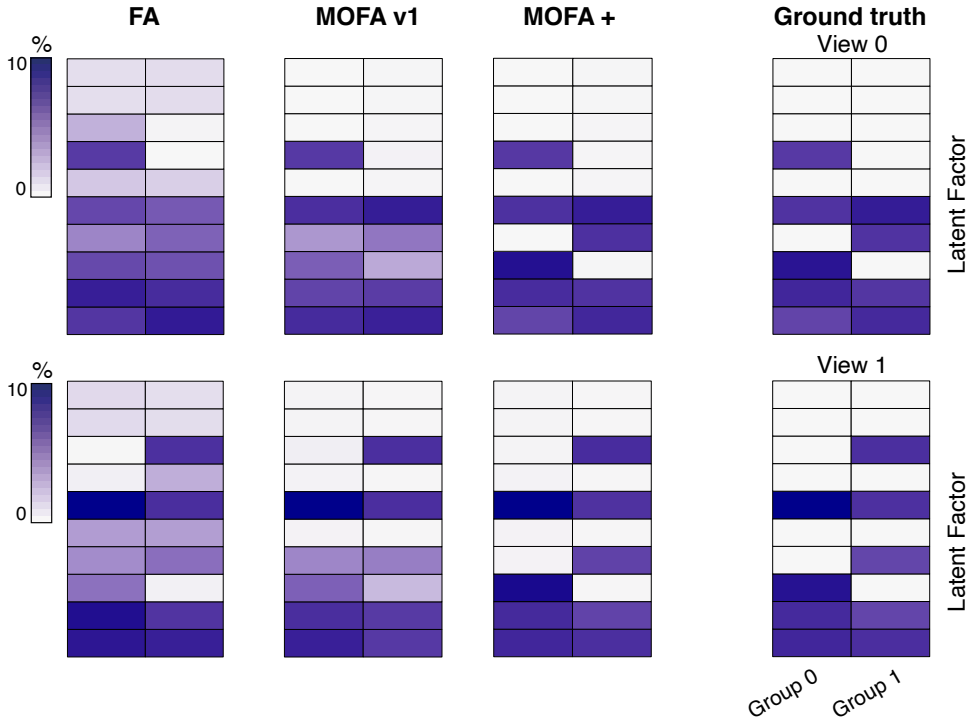


Figure 1.7: Validation of group-wise ARD prior in the factors using simulated data. Representative example of the resulting variance explained patterns. The first row of heatmaps correspond to modality 0 and the second row to modality 1. In each heatmap, the first column corresponds to group 0 and the second column to group 1. Rows correspond to the inferred factors. The colour scale displays the percentage of variance explained by a given factor in a given modality and group. The heatmaps displayed in columns one to three show the solutions yielded by different models (Bayesian Factor Analysis; MOFA; MOFA+). The ground truth is shown in the right panel.

1.4 Applications

1.4.1 Integration of a heterogeneous time-course single-cell RNA-seq dataset

To demonstrate the novel multi-group integration framework, we considered a time course scRNA-seq dataset comprising 16,152 cells that were isolated from a total of 8 mouse embryos from developmental stages E6.5, E7.0 and E7.25 (two biological replicates per stage), encompassing post-implantation and early gastrulation[**Pijuan-Sala2019**]. This data set, which has been introduced in Chapter 3, consists on a single view (RNA expression) but with a clear group structure where cells belongs to different biological replicates at different time points. Different embryos are expected to contain similar subpopulations of cells but also some differences due to developmental progression. As a proof of principle, we used MOFA+ to disentangle stage-specific transcriptional signatures from signatures that are shared across all stages.

MOFA+ identified 7 Factors that explain at least 1% of variance (across all groups). Notably, this latent representation captures between 35% and 55% of the total transcriptional heterogeneity per embryo:

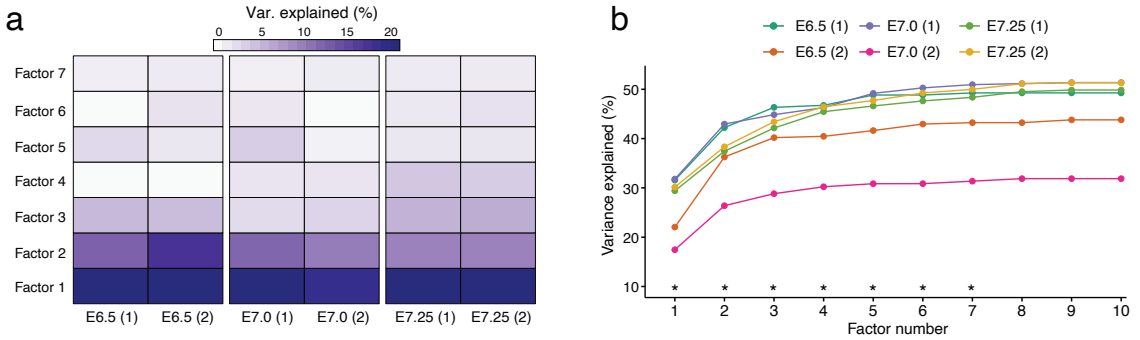


Figure 1.8: MOFA+ variance explained estimates in the gastrulation scRNA-seq atlas. (a) Heatmap displays the variance explained (%) for each factor (rows) in each group (pool of mouse embryos at a specific developmental stage, columns). The bar plots show the variance explained per group with all factors.
(b) Cumulative variance explained (per group, y-axis) versus factor number (x-axis). Asterisks indicate the factors that are selected for downstream analysis (minimum of 1% variance explained).

1.4.1.1 Characterisation of individual factors

Some factors recover the existence of post-implantation developmental cell types, including extra-embryonic (ExE) tissue (Factor 1 and Factor 2), and the emergence of mesoderm cells from the primitive streak (Factor 4). Consistently, the top weights for these factors are enriched for lineage-specific gene expression markers, including *Ttr* and *Apoa1* for ExE endoderm Figure 1.9; *Rhox5* and *Bex3* for ExE ectoderm (not shown); *Mesp1* and *Phlda2* for nascent mesoderm Figure 1.10. Other factors captured technical variation due to metabolic stress that affects all batches in a similar fashion (Factor 3, Figure 1.11).
The characterisation of other factors is described in[**Argelaguet2020**] and is not reproduced here for simplicity.

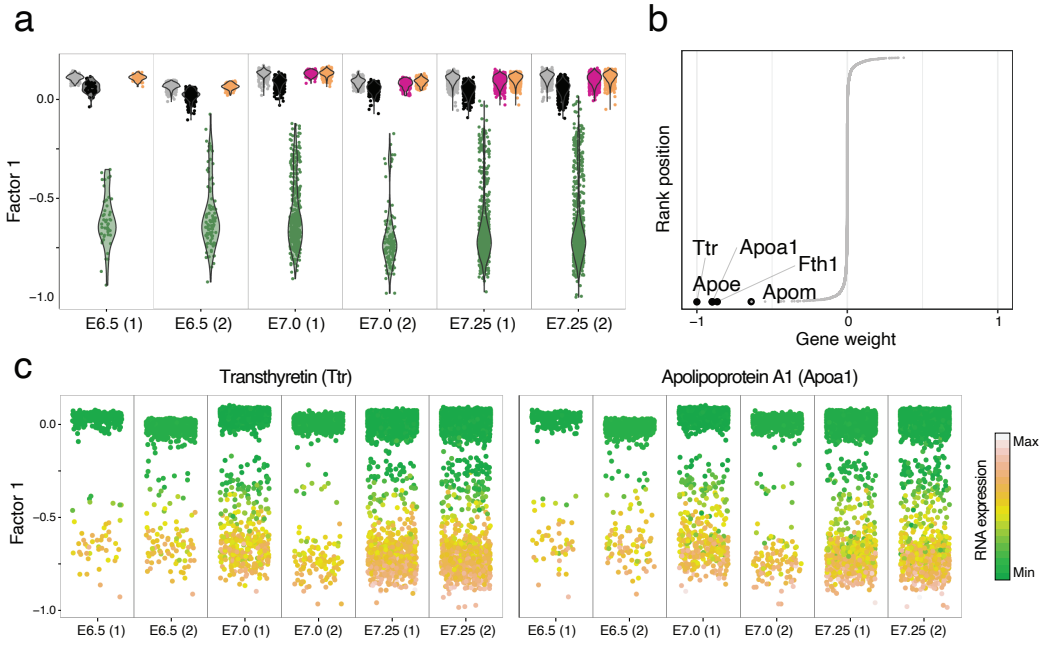


Figure 1.9: Characterisation of Factor 1 as extra-embryonic (ExE) endoderm formation.
(a) Beeswarm plot of Factor values for each group. Cells are grouped and coloured by cell type.
(b) Plot of gene weights. Highlighted are the top five genes with largest weight (in absolute values)
(c) Beeswarm plot of Factor values for each group. Cells are coloured by the expression of the two genes with largest weight (in absolute values).

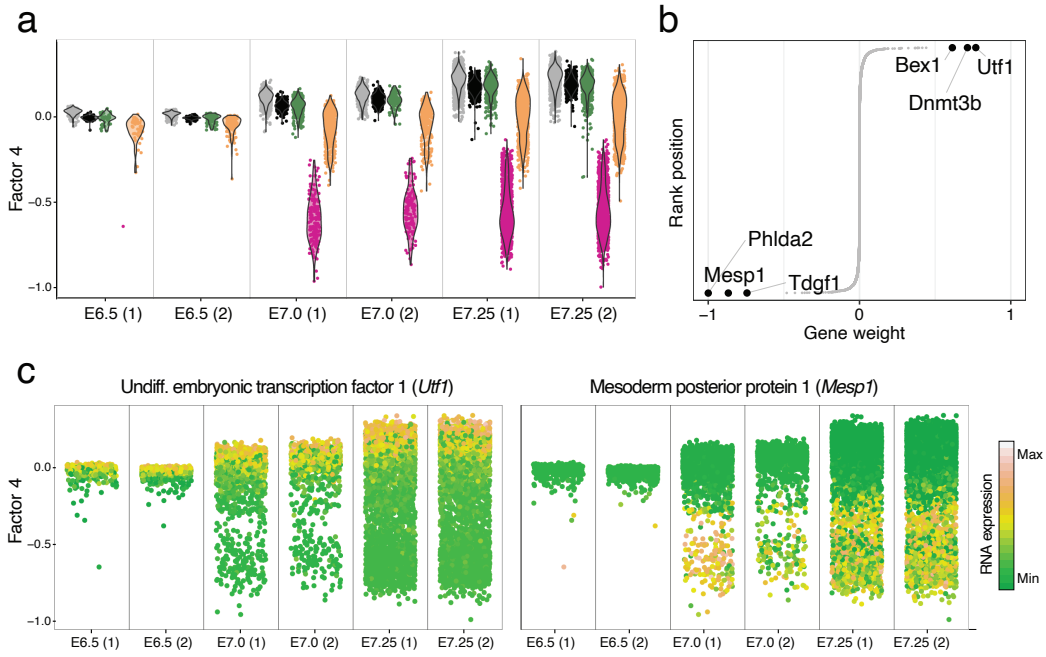


Figure 1.10:
Characterisation of Factor 4 as mesoderm commitment.
(a) Beeswarm plot of Factor values for each group. Cells are grouped and coloured by cell type.
(b) Plot of gene weights. Highlighted are the top five genes with largest weight (in absolute values)
(c) Beeswarm plot of Factor values for each group. Cells are coloured by the expression of the two genes with largest weight (in absolute values).

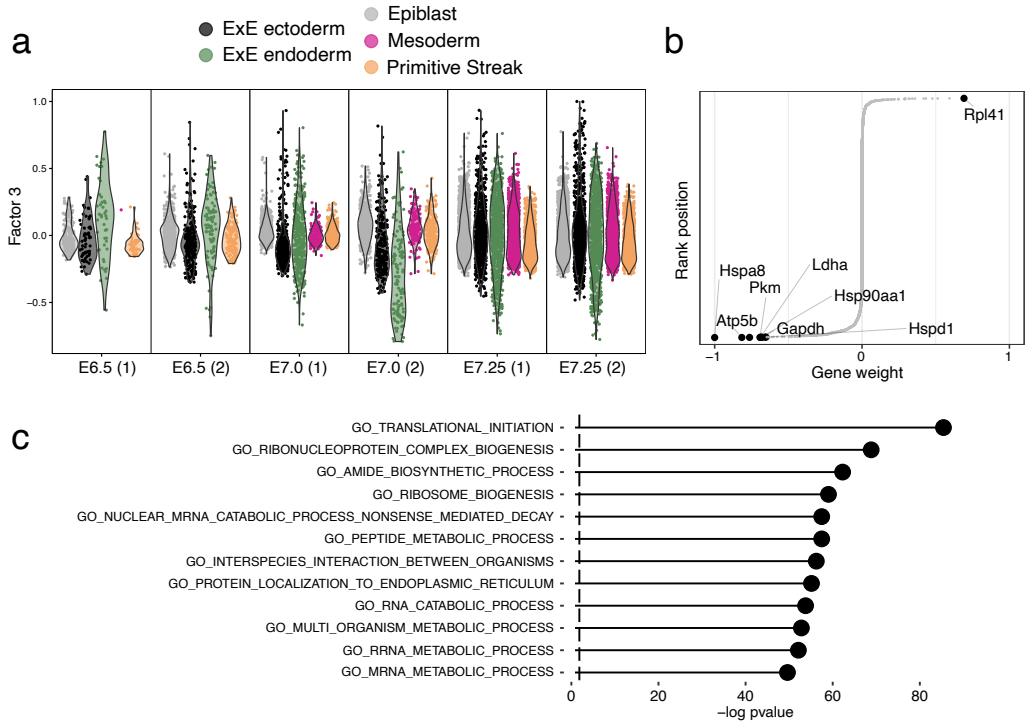


Figure 1.11:
Characterisation of Factor 3 as cell-to-cell differences in metabolic activity.
(a) Beeswarm plot of Factor values for each group. Cells are grouped and coloured by cell type.
(b) Plot of gene weights. Highlighted are the top seven genes with largest weight (in absolute values)
(c) Gene set enrichment analysis applied to the gene weights using the Reactome gene sets[**Fabregat2015**]. Significance is assessed via a parametric. Resulting p-values were adjusted for multiple testing using the Benjamini-Hochberg procedure.

Interestingly, Factors display differeent signatures of activity (variance explained) across developmental stages. For example, the variance explained by Factor 1 remains constant across developmental progression (Figure 1.8), indicating that commitment to ExE endoderm fate occurs early in the embryo and the proportion of this cell type remains relatively constant. In contrast, the activity of Factor 4 increases with developmental progression, consistent with a higher proportion of cells committing to mesoderm after ingression through the primitive streak.

In conclusion, this application shows how MOFA+ can identify biologically relevant structure in *structured* scRNA-seq datasets.

1.4.2 Identification of context-dependent methylation signatures associated with cellular diversity in the mammalian cortex

As a second use case, we considered how MOFA+ can be used to investigate cellular heterogeneity in epigenetic signatures between populations of neurons. This application illustrates how a multi-group and multi-view structure can be defined from seemingly uni-modal data.

We considered a data set of of 3,069 cells isolated from the frontal cortex of young adult mouse, where DNA methylation was profiled using single-cell bisulfite sequencing[**Luo2018**].
Some background to motivate our experimental design: in mammalian genomes, DNA methylation predominantly occurs at CpG dinucleotides (mCG), with more than 75% of CpG sites being methylated in differentiated cell types. By contrast non-CpG methylation (mCH) has been historically dismissed as methodological artifact of incomplete bisulfite conversion, until recent works have confirmed their existence in restricted cell types. Yet, evidence for a potential functional role remains controversial[**He2015**].

Here we used MOFA+ to dissect the cellular heterogeneity associated with mCH and mCG in the mouse frontal cortex. As input data we quantified mCH and mCG levels at gene bodies, promoters and putative enhancer elements. Each combination of genomic and sequence context was defined as a separate view.
As described in Chapters 1 and 3, methylation levels were calculated per cell and genomic feature using a binomial model where the number of successes correspond to the number of reads that support methylation (or accessibility) and the number of trials the total number of reads.
Finally, to explore the influence of the neuron's location we grouped cells according to their cortical layer: Deep, Middle or Superficial (**??**). Notably, the resulting data set is extremely sparse, which hampers the use of conventional dimensionality reduction techniques.The probabilistic framework underlying MOFA+ naturally enables the handling of missing values by ignoring the corresponding terms in the likelihood function.

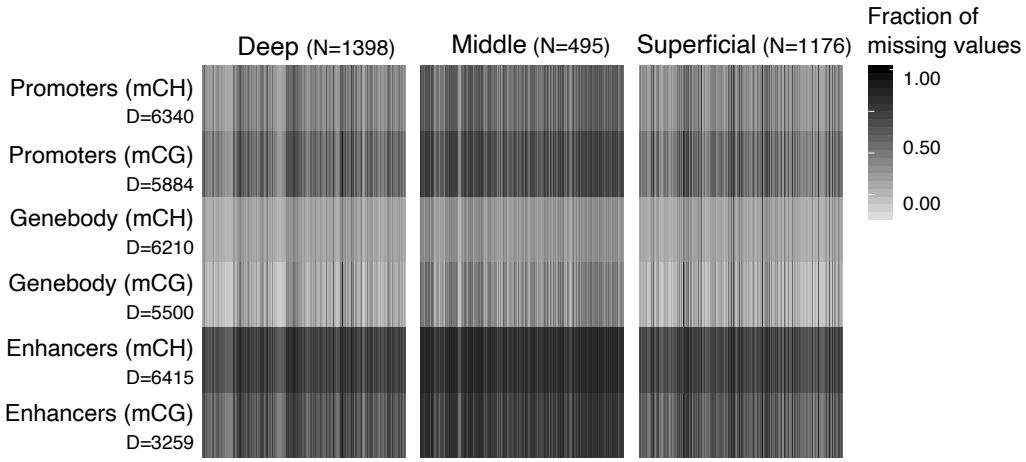


Figure 1.12

MOFA+ identifies 10 factors with a minimum variance explained of 1% in at least one data modality.

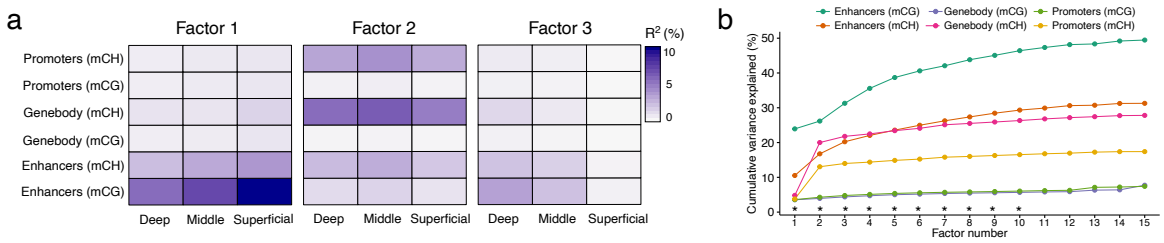


Figure 1.13: MOFA+ variance explained estimates in the frontal cortex DNA methylation data set.
(a) Percentage of variance explained for each factor across the different groups (cortical layer, x-axis) and views (genomic context, y-axis). For simplicity, only the first three factors are shown.
(b) Cumulative variance explained (per group, y-axis) versus factor number (x-axis). Asterisks indicate the factors that are selected for downstream analysis (minimum of 1% variance explained in at least one data modality).

Factor 1, the major source of variation, is linked to the existence of inhibitory and excitatory neurons, the two major classes of neurons (Figure 1.15). This factor shows significant mCG activity across all cortical layers, mostly driven by coordinated changes in enhancer elements, but to some extent also gene bodies.

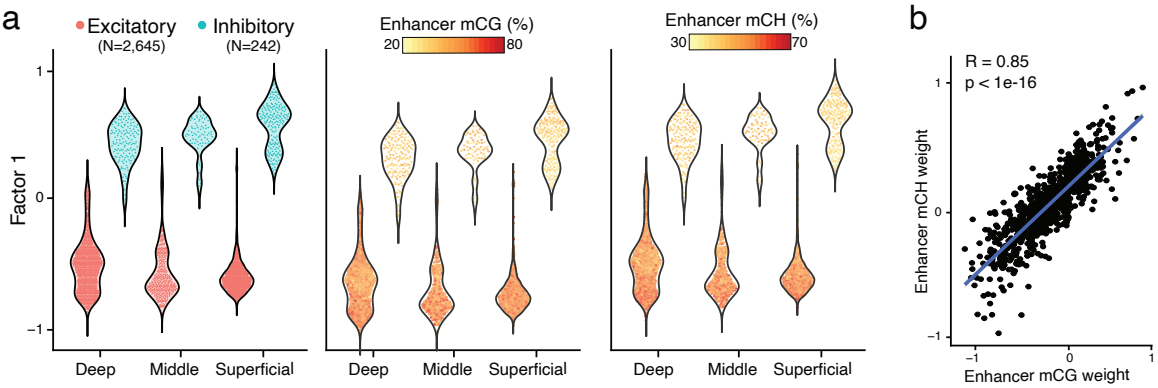


Figure 1.14: Characterisation of Factor 1 as DNA methylation signatures distinguishing inhibitory versus excitatory cell types[Luo2016].
(a) Beeswarm plots of Factor values per group (cortical layer). In the left plot, cells are coloured by neuron class. In the middle and right plots the cells are coloured by average mCG and mCH levels (%), respectively, of the top 100 enhancers with the largest weights.
(b) Correlation of enhancer mCG weights (x-axis) and mCH weights (y-axis)

Factor 2 captures genome-wide differences in global mCH levels (R=0.99, not shown), most likely to be a technical source of variation.

Factor 3 captures heterogeneity linked to the increased cellular diversity along cortical depth, with the Deep layer displaying significantly more diversity of excitatory cell types than the Superficial layer (**??**).

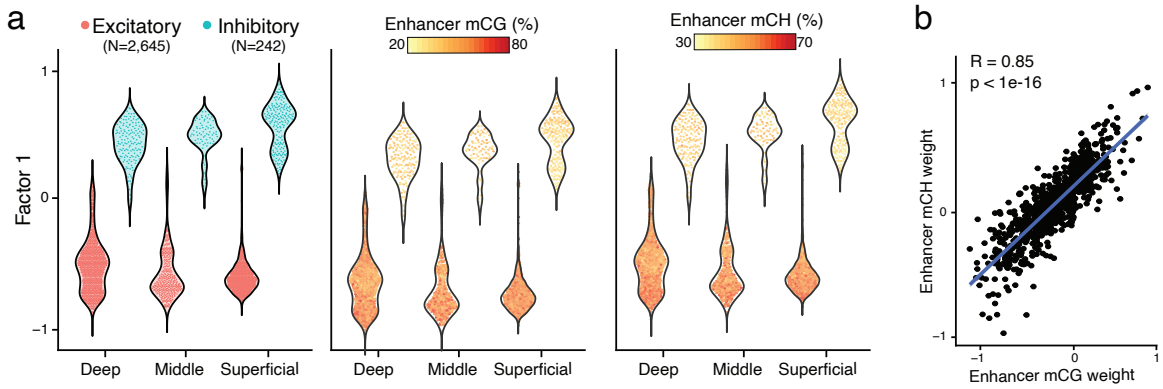


Figure 1.15: Characterisation of Factor 3 as increased cellular diversity along cortical depth.
(a) Beeswarm plots of Factor values per group (cortical layer). In the left plot, cells are coloured by neuron class. In the middle and right plots the cells are coloured by average mCG and mCH levels (%), respectively, of the top 100 enhancers with the largest weights.
(b) Correlation of enhancer mCG weights (x-axis) and mCH weights (y-axis).

The (linear) MOFA factors can be combined by further non-linear dimensionality reduction algorithms such as UMAP or t-SNE. In this case, we show that the t-SNE projections reveals the existence of multiple subpopulations of both excitatory and inhibitory cell types. Notably, the MOFA+ factors are significantly better at identifying these subpopulations than the conventional approach of using Principal Component Analysis with imputed measurements:

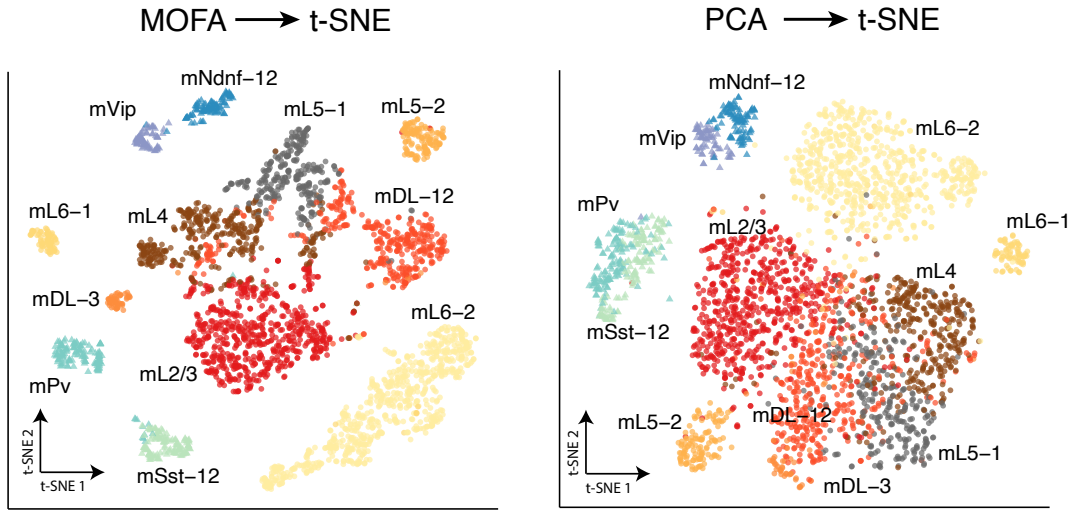


Figure 1.16: Comparison of MOFA+ Factors and Principal Components as input to t-SNE.
The scatterplots display t-SNE projections when using as input MOFA+ factors (left) or principal components (right). Each dot represents a cell, coloured by cell type[**Luo2016**]. To ensure a fair comparison we used the same number of PCs and MOFA+ Factors ($K = 15$). Feature-wise imputation of missing values is applied for the PCA.

Interestingly, in addition to the dominant mCG signal, MOFA+ connects Factor 1 and Factor 3 to variation in mCH, which suggest a role of mCH in cellular diversity. We hypothesise that this could be supported if the genomic regions that show mCH signatures are different than the ones marked

the conventional mCG signatures. To investigate this, we correlated the mCH and mCG feature loadings for each factor and genomic context (Figure 3e and Figure S14). In all cases we observe a strong positive dependency, indicating that mCH and mCG signatures are spatially correlated and target similar loci. Taken together, these results supports the hypothesis that mCH and mCG tag the same genomic loci and are associated with the same sources of variation, suggesting that the presence of mCH may be the result of non-specific *de novo* methylation as a by-product of the establishment of mCG.

1.4.3 Identification of molecular signatures of lineage commitment during mammalian embryogenesis

As a last application, we considered a substantially more complex dataset with multiple sample groups and data modalities. The dataset consists of a multi-omic atlas of mouse gastrulation where scNMT-seq was used to simultaneously profile RNA expression, DNA methylation and chromatin accessibility in 1,828 cells at multiple stages of development[**Argelaguet2019**]. This is the data set that I introduced in Chapter 3.

In this dataset MOFA+ can be used to deline the coordinated variation between the transcriptome and the epigenome and detect at which stage(s) of development it occurs.
The main difference with respect to the MOFA analysis presented in Chapter 3 is that MOFA+ enables a multi-stage characterisation of the data set's variation. In FIGURE XXX we only considered stage E7.5, whereas here we can employ the multi-group functionality to perform a simultaneous analysis across multiple stages.

1.4.3.1 Data processing

As input to the model we quantified DNA methylation and chromatin accessibility values over two sets of regulatory elements: gene promoters and enhancer elements (distal H3K27ac sites). RNA expression was quantified over protein-coding genes. More details on the feature quantification and data processing are described in Chapter 3.

As in the MOFA analysis presented in Chapter 3, here we defined separate views for the RNA expression and for each combination of genomic context and epigenetic readout. Cells were grouped according to their developmental stage (E5.5, E6.5 and E7.5), reflecting the underlying experimental design[**Argelaguet2019**] (Figure 1.17). As discussed in Chapter 3, the CpG methylation (endogenous DNA methylation) and GpC methylation (proxy for chromatin accessibility) result in very sparse readouts that are challenging to analyse with standard statistical approaches.

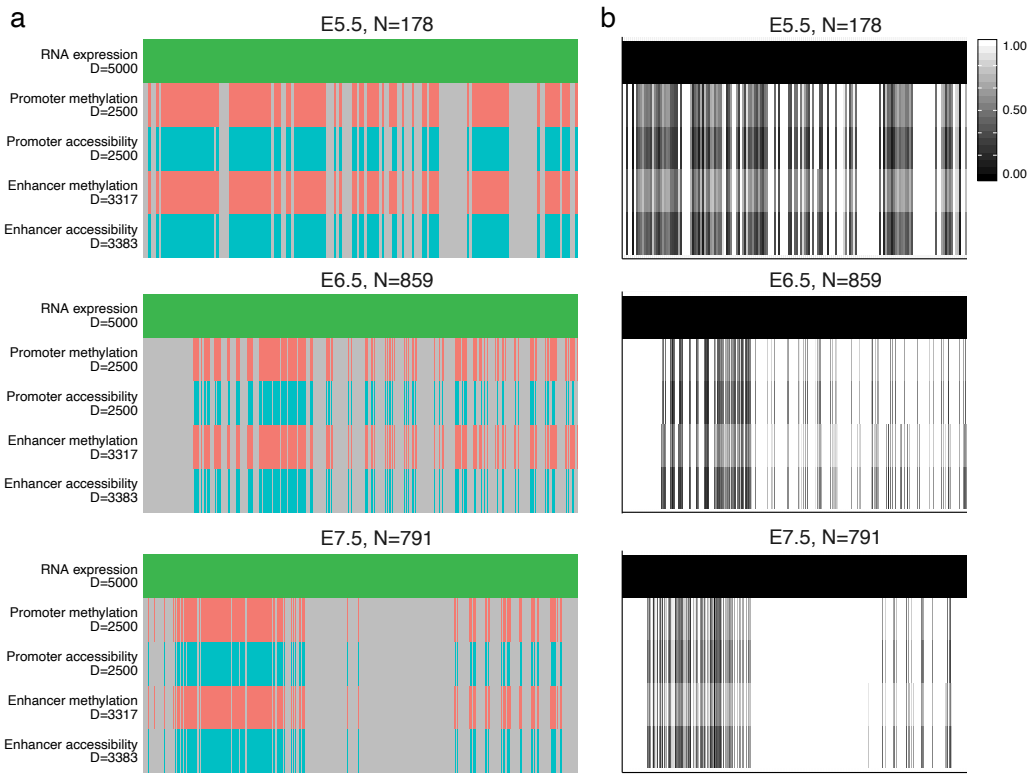


Figure 1.17: Overview of the scNMT-seq mouse gastrulation data set used as input for MOFA+.
(a) Structure of the input data in terms of modalities (x-axis) versus samples (y-axis). Each panel corresponds to a different group (embryonic stage). Grey bars represent missing modalities.
(b) Structure of the missing values in the data. For each cell and modality, the colour displays the fraction of missing values.

1.4.3.2 Model overview

In this data set MOFA+ identifies 8 factors with a minimum variance explained of 1% in the RNA expression (Figure 1.18). Interestingly, this plot indicates important differences in the heterogeneity of the considered data modalities. The MOFA+ Factors explain little amounts of variance in chromatin accessibility, both for promoters (≈15%) and enhancers (≈18%), mostly driven by Factors 1 and 2. In contrast, the model explains larger amounts of variation in DNA methylation (≈23% for promoters and (≈59%) for enhancers). However, as in chromatin accessibility, this variation is mostly driven by the first two Factors. Finally, for RNA expression there is a steady increase in the variance explained, suggesting that the sources of variation captured beyond Factor 2 are largely driven by RNA expression alone.

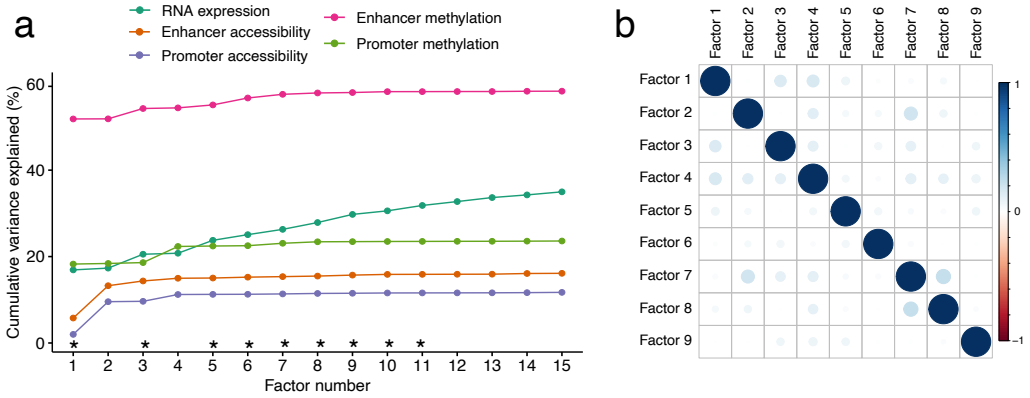


Figure 1.18: Unsupervised characterisation of MOFA+ factors from the scNMT-seq gastrulation data set.
(a) Cumulative variance explained (per view, y-axis) versus factor number (x-axis). Asterisks indicate the factors that are selected for downstream analysis (minimum of 1% variance explained in the RNA expression). Note that the variance estimates shown here are the sum across all groups. (b) Pearson correlation coefficients between selected factors. In MOFA+ there are no orthogonality constraints, but the factors are expected to be largely uncorrelated.

1.4.3.3 Characterisation of Factors 1 and 2

The first factor captured the formation of ExE endoderm, a cell type that is present across all stages (Figure 4a), in agreement with our previous results using the independently generated transcriptomic atlas of mouse gastrulation (Figure 2). MOFA+ links Factor 1 to changes across all molecular layers. Notably, the distribution of weights for DNA methylation are skewed towards negative values (at both enhancers and promoters), indicating that ExE endoderm cells are characterised by a state of global demethylation, consistent with previous studies 44.

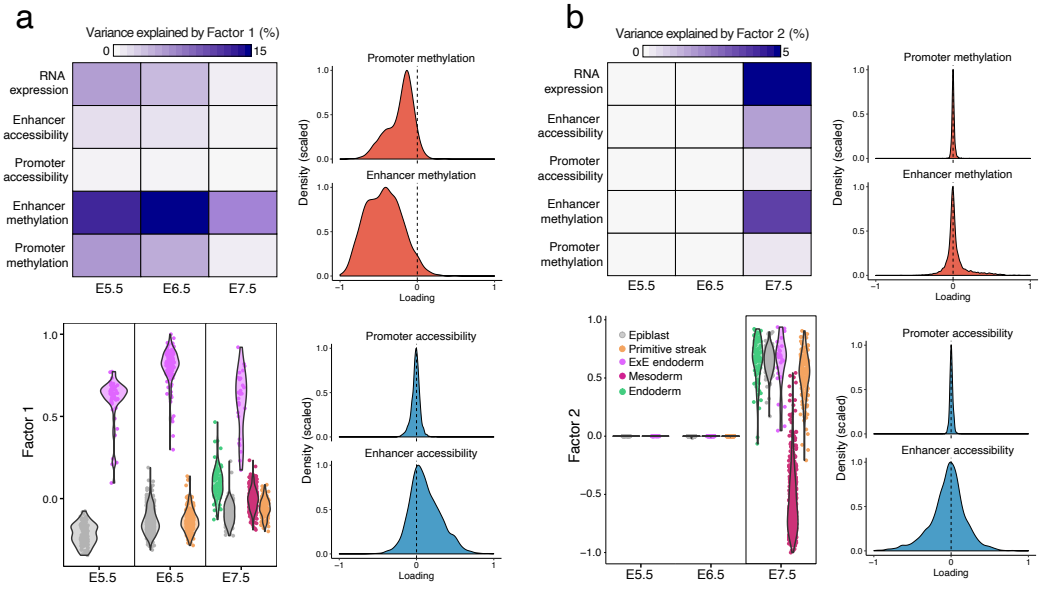


Figure 1.19

1.4.3.4 Characterisation of other Factors

As discussed above, the rest of the MOFA+ Factors explain significantly less variance than Factors 1 and 2, and they are mostly driven by the RNA expression (Figure 1.18). Their etiolgoy can be identified by the inspection of gene weights and by gene set enrichment analysis. For simplicity, I will only show Factor 6, which captures cell-cycle variation that is consistenly found across all three embryonic stages.

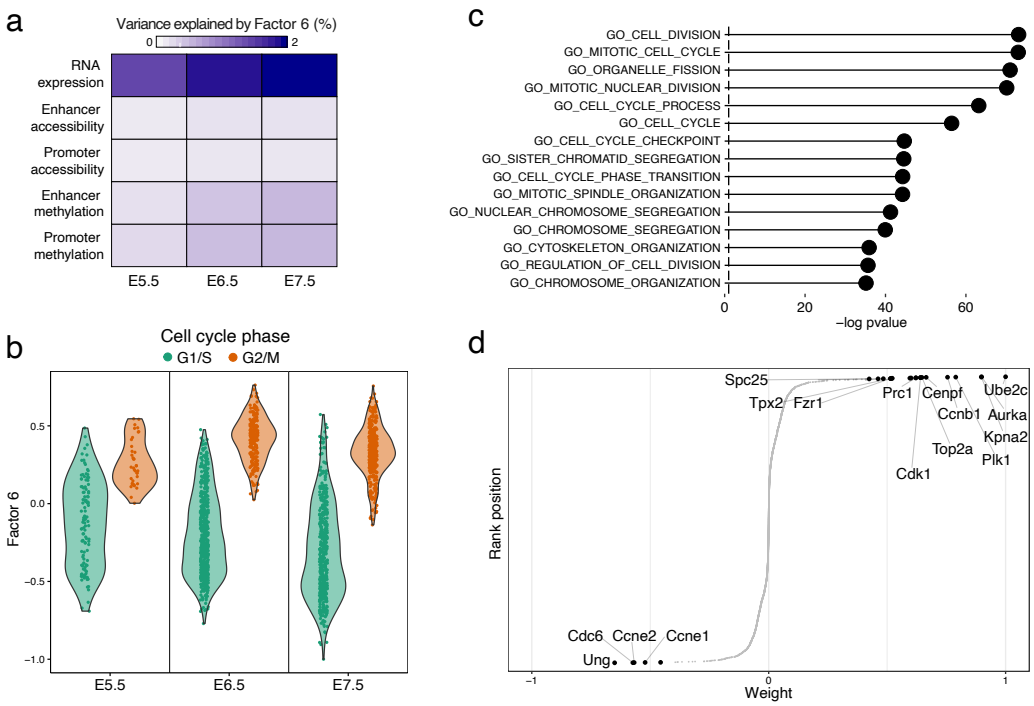


Figure 1.20: Characterisation of Factor 6 as cell cycle variation.
(a) Variance explained by Factor 6 in each group (embryonic stage, columns) and data modality (rows).
(b) Distribution of Factor 6 values per group (embryonic stage, x-axis), with cells coloured by the inferred cell cycle state using *cyclone*.
(c) Gene set enrichment analysis applied to the Factor 6 weights.
(d) Cumulative distribution of RNA weights for Factor 6. The top genes with the highest (absolute) weight are labeled.

1.4.3.5 Conclusion

1.5 A note on the implementation

The inference framework in MOFA+ is implemented in Python, whereas the downstream analysis and visualisations are implemented in R. GPU acceleration is implemented using CuPy[**Okuta2017**], an open-source matrix library accelerated with NVIDIA CUDA.
To facilitate adoption of the method, we deploy MOFA+ as open-source software (https://github.com/bioFAM/MOF with multiple tutorials and a web-based analysis workbench, hopefully enabling a user-friendly characterisation of structured single-cell data.

1.6 Limitations and opern perspectives

In this Chapter we proposed a generalisation of the MOFA model for the principled analysis of large-scale *structured* data sets. This solves some of the limitations of the MOFA model presented in Chapter 2, but a significant amount of challenges remain unsolved and could be addressed in future research:

- **Linearity**: this is arguably the major limitation of MOFA. Although it is critical for obtaining interpretable feature weights, this results in a significant loss of explanatory power. Deep generative models have proven successful in modelling complex observations. Their principle is the use of non-linear maps via neural networks to encode the parameters of probability distributions. Among this class of methods, variational autoencoders provide a rigurous and scalable non-linear generalisation of factor models. [**Ainsworth2018**].

- **Improving the stochastic inference scheme**: a common extension of stochastic gradient descent is the addition of a *momentum* term, which has been widely adopted in the training of artificial neural networks [**Zeiler2012**, **Ning1999**]. The idea is to take account of past updates when calculating the present step, using for example a moving average calculation. This has been shown to improve the stability of gradients vectors, thus leading to a faster convergence.

- **Modelling dependencies between groups**: often groups are not independent and have some type of structure among themselves. A clear example are time course experiments. Explicit modelling of these dependencies, when known, could help on model inference and interpretation.

- **Modelling continuous dependencies between samples and/or features**: in the MOFA framework the views and the groups correspond to discrete and non-overlapping sets. An interesting improvement would be to model continuous dependencies using Gaussian Process priors [**XX**]. A clear application for this is spatial transcriptomics data, where one could build a (spatial) covariance matrix using cell-to-cell distances which can then be imposed in the prior distribution of the latent factors (recall that in MOFA the prior distribution for the factors assumes independence between samples). This would improve the detection of sources of variation with a spatial component.s