Methods for: Multi-Omics factor analysis disentangles heterogeneity in blood cancer

Ricard Argelaguet, Britta Velten, Damien Arnol, Sascha Dietrich, Thorsten Zenz, John C. Marioni, Florian Buettner, Wolfgang Huber, Oliver Stegle

Contents

1 Multi-Omics Factor Analysis model **2**
1.1 Model definition . . . 2
1.2 Inference . . . 3
1.3 Modelling non-Gaussian data . . . 3
1.4 Downstream analyses . . . 4
1.5 Comparison to existing group factor analysis models . . . 5
1.6 Implementation and practical considerations for training . . . 5

2 Model validation using simulations **6**
2.1 Recovery of the true number of factors . . . 6
2.2 Non-Gaussian data . . . 7
2.3 Identifying co-variation patterns across data sets . . . 7

3 Detailed methods on CLL analysis **7**
3.1 Data processing . . . 7
3.2 Robustness . . . 7
3.3 Inspection of loadings . . . 7
3.4 Gene set enrichment analysis . . . 8
3.5 Imputation . . . 8
3.6 Survival Analysis . . . 8

Appendix **9**
Introduction to variational Bayes inference . . . 9
Update equations for the Gaussian model . . . 10
Evidence Lower bound calculation for the Gaussian model . . . 12
Updates for the non-Gaussian model . . . 13

1 Multi-Omics Factor Analysis model

MultiOmics Factor Analysis (MOFA) is a statistical model aimed at disentangling sources of variation in multi-omics data. Here, we define the statistical model (section 1.1) and introduce a scalable inference procedure (section 1.2). Next, we describe extensions to handle non-Gaussian data (section 1.3) as well as the pipeline used for automatic annotation of factors (section 1.4). Finally, we provide a comparative overview of MOFA and previous work on group factor analysis models (section 1.5) and conclude with practical considerations for training (section 1.6)

1.1 Model definition

Starting from data consisting of M matrices $\mathbf{Y}^m = [y^m_{nd}] \in \mathbb{R}^{N \times D_m}$ the matrices are jointly factorized as

$$\mathbf{Y}^m = \mathbf{Z}\mathbf{W}^{mT} + \boldsymbol{\epsilon}^m, \tag{1}$$

where $\mathbf{Z} = [z_{nk}] \in \mathbb{R}^{N \times K}$ is a single matrix that contains the low-dimensional latent variables, $\mathbf{W}^m = [w^m_{dk}] \in \mathbb{R}^{D_m \times K}$ are loading matrices that relate the high-dimensional space to the low dimensional representation, and $\boldsymbol{\epsilon}^m = [\epsilon^m_d] \in \mathbb{R}^{D_m}$ denotes residual noise. We start by assuming Gaussian residuals $\boldsymbol{\epsilon}^m$, similar to standard (group) factor analysis models, while allowing for heteroscedasticity across features (for extensions to non-Gaussian settings see section 1.3).

$$p(\epsilon^m_d) = \mathcal{N}\left(\epsilon^m_d \,|\, 0, 1/\tau^m_d\right). \tag{2}$$

This results in the following normal likelihood

$$p(y^m_{nd}) = \mathcal{N}\left(y^m_{nd} \,|\, \mathbf{w}^m_{d,:}\mathbf{z}_{n,:}, 1/\tau^m_d\right), \tag{3}$$

where $\mathbf{w}^m_{d,:}$ denotes the d-th row of the loading matrix $\mathbf{W}^m$ and $\mathbf{z}_{n,:}$ the n-th row of the latent factor matrix $\mathbf{Z}$. For a fully probabilistic treatment we place prior distributions on the weights $\mathbf{W}^m$, the latent variables $\mathbf{Z}$ as well as on the precision of the noise $\boldsymbol{\tau}^m$. We use a standard Gaussian prior on the latent variables and a conjugate Gamma prior for the precision:

$$p(z_{n,k}) = \mathcal{N}\left(z_{n,k} \,|\, 0, 1\right), \tag{4}$$

$$p(\tau^m_d) = \mathcal{G}\left(\tau^m_d \,|\, a^\tau_0, b^\tau_0\right), \tag{5}$$

with $a^\tau_0, b^\tau_0 = 1e^{-14}$ to obtain uninformative priors.
A key determinant of the model's properties is the regularization used on the weights $\mathbf{W}^m$. MOFA encodes two levels of sparsity: a view- and factor-wise sparsity and a feature-wise sparsity. The aim of the factor- and view-wise sparsity is to identify which factors are active in which view, such that the weight vector $\mathbf{w}^m_{:,k}$ is shrunk to zero if the factor k is neglible for variation in view m. In addition, the feature-wise sparsity puts zero weights on individual features that do not drive variation. This further improves interpretability, as typically only a small number of features remains "active", i. e., has non-zero weight. We achive both levels of sparsity by placing appropriate priors on the weight matrices.

Specifically, we combine an Automatic Relevance Determination (ARD) prior [15] for the view- and factor-wise sparsity with a spike-and-slab prior [16] for the feature-wise sparsity, similar to [12]. However, as the spike-and-slab prior

$$p(w) = (1-\theta)\mathbb{1}_0(w) + \theta\mathcal{N}(w \,|\, 0, 1/\alpha) \tag{6}$$

contains a Dirac delta function, which makes the inference troublesome, here we use a re-parametrization of the weights w as a product of a Gaussian random variable $\hat{w}$ and a Bernoulli random variable s that is more amenable to variational inference [19, 5] resulting in the following prior:

$$p(\hat{w}^m_{d,k}, s^m_{d,k}) = \mathcal{N}(\hat{w}^m_{d,k} \,|\, 0, 1/\alpha^m_k)\,\mathrm{Ber}(s^m_{d,k} \,|\, \theta^m_k) \tag{7}$$

In this formulation α^m_k controls the strength of factor k in view m and θ^m_k controls the degree of contribution from the spike term, determining the overall feature-wise sparsity levels of factor k in view m. In order to automatically learn these parameters we use the following conjugated priors

$$p(\theta^m_k) = \mathrm{Beta}\left(\theta^m_k \,|\, a^\theta_0, b^\theta_0\right) \tag{8}$$

$$p(\alpha^m_k) = \mathcal{G}\left(\alpha^m_k \,|\, a^\alpha_0, b^\alpha_0\right), \tag{9}$$

with hyper-parameters $a^\theta_0, b^\theta_0 = 1$ and $a^\alpha_0, b^\alpha_0 = 1e^{-14}$ to get uninformative priors.
In practice, the ARD prior yields a matrix $\boldsymbol{\alpha} \in \mathbb{R}^{M \times K}$ that defines four different types of factors:

- Factors that do not explain variation in any data set (inactive factors): all values in the corresponding columns of $\boldsymbol{\alpha}$ are large.
- Factors that explain variation in all data sets (fully shared factors): all values in the corresponding columns of $\boldsymbol{\alpha}$ are small.
- Factors that explain variation in a single data set (unique factors): all values in the corresponding columns of $\boldsymbol{\alpha}$ are very large, except one.
- Factors that explain variation in a subset of data sets (partially shared factors): some values in the corresponding columns of $\boldsymbol{\alpha}$ are very large whereas others are small.

Finally, the joint probability density function is given by

$$\begin{aligned} p(\mathbf{Y}, \hat{\mathbf{W}}, \mathbf{S}, \mathbf{Z}, \boldsymbol{\Theta}, \boldsymbol{\alpha}, \boldsymbol{\tau}) = & \prod_{m=1}^{M} \prod_{n=1}^{N} \prod_{d=1}^{D_m} \mathcal{N}\left(y_{nd}^m \mid \sum_{k=1}^{K} s_{dk}^m \hat{w}_{dk}^m z_{nk}, 1/\tau_d\right) \\ & \prod_{d=1}^{D_m} \prod_{k=1}^{K} \mathcal{N}\left(\hat{w}_{dk}^m \mid 0, 1/\alpha_k^m\right) \mathrm{Ber}(s_{d,k}^m | \theta_k^m) \\ & \prod_{n=1}^{N} \prod_{k=1}^{K} \mathcal{N}\left(z_{nk} \mid 0, 1\right) \\ & \prod_{m=1}^{M} \prod_{k=1}^{K} \mathrm{Beta}\left(\theta_k^m \mid a_0^\theta, b_0^\theta\right) \\ & \prod_{m=1}^{M} \prod_{k=1}^{K} \mathcal{G}\left(\alpha_k^m \mid a_0^\alpha, b_0^\alpha\right) \\ & \prod_{m=1}^{M} \prod_{d=1}^{D_m} \mathcal{G}\left(\tau_d^m \mid a_0^\tau, b_0^\tau\right). \end{aligned} \tag{10}$$

This completes the definition of the model, which is graphically illustrated in Figure S1.

1.2 Inference

To ensure scalable inference we use a variational approach with a mean-field approximation[4]. The core idea of variational Bayes is to approximate the true posterior distribution over all unobserved variables in the model using a variational distribution that has a factorized form. Here the assumed form of the variational distribution is the following:

$$\begin{aligned} q(\mathbf{Z}, \mathbf{S}, \hat{\mathbf{W}}, \boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\theta}) &= q(\mathbf{Z}) q(\boldsymbol{\alpha}) q(\boldsymbol{\theta}) q(\boldsymbol{\tau}) q(\mathbf{S}, \hat{\mathbf{W}}) \\ &= \prod_{n=1}^{N} \prod_{k=1}^{K} q(z_{n,k}) \prod_{m=1}^{M} \prod_{k=1}^{K} q(\alpha_k^m) q(\theta_k^m) \prod_{m=1}^{M} \prod_{d=1}^{D_m} q(\tau_d^m) \prod_{k=1}^{K} q(\hat{w}_{d,k}^m, s_{d,k}^m) \end{aligned}$$

This assumption allows the derivation of a simple iterative inference scheme, in which the variational distribution is optimised to get as close as possible to the true distribution by minimising their Kullback-Leibler divergence, or equivalently, a lower bound on the marginal likelihood, also called evidence lower bound (ELBO) [4]. This inference approach ensures that the model scales linearly with the number of factors, the number of features, the number of views and number of samples (Figure S5).
For details on the inference and the update equations see the Appendix.

1.3 Modelling non-Gaussian data

So far, we have decribed the model and inference scheme under the assumption of Gaussian data. In practice, other types of data are often encountered, e.g. binary data or count data. The Gaussian linear model described in section 1.1 can be adapted to these types of data using generalised linear models in Eq. (1) and the corresponding likelihood in Eq. (3).

To implement efficient variational inference in conjunction with a non-Gaussian likelihood we adapt prior work from [18] and [11] using variational lower bounds. The key idea is to approximate the non-Gaussian distribution of the observations by a lower bound of a Gaussian variational form and iteratively improve the fit by introducing variational parameters that are updated alongside the model parameters. The approximation can be obtained using a second-order Taylor expansion on the logarithm of the likelihood assuming its second derivative to be bounded by κ:

$$-\log p(\mathbf{Y}|\mathbf{X}) = \sum_{n=1}^{N}\sum_{d=1}^{D} f_{nd}(x_{nd}) \tag{11}$$

$$f_{nd}(x_{nd}) \leq \frac{\kappa_d}{2}(x_{nd}+\zeta_{nd})^2 + f'(\zeta_{nd})(x_{nd}-\zeta_{nd}) + f_{nd}(\zeta_{nd}), \tag{12}$$

where ζ_{nd} are the variational parameters to be optimised.
The Gaussian form allows to efficiently re-use the variational updates from the Gaussian case on modified pseudo-data, implicitly defined by the variational lower bounds.
Here we implement two examples of non-Gaussian likelihoods, a Bernoulli likelihood to model binary data and a Poisson likelihood to model count data. For details on the derivation and the update equations see the Appendix.

1.4 Downstream analyses

Interpretation of the factors is a critical part of latent variable models, and yet it remains a major challenge that impedes their broad applicability in the biological community. In MOFA we provide an R package called *MOFAtools* with a simple pipeline for the characterisation of the latent factors.
The first step, after a model has been trained, is to disentangle the variation explained by each factor in each view. To do so, we compute the fraction of total variance explained (R^2) by each factor k in view m

$$R^2_{m,k} = 1 - \frac{\sum_{n,d}(y^m_{nd} - (\mathbf{z}_{:,k}\mathbf{w}^m_{k,:})_{nd} - \mu^m_d)^2}{\sum_{n,d}(y^m_{nd} - \mu^m_d)^2} \tag{13}$$

as well as the fraction of variance explained per view taking into account all factors

$$R^2_{m} = 1 - \frac{\sum_{n,d}(y^m_{nd} - (\mathbf{z}_{:,k}\mathbf{w}^m_{k,:})_{nd} - \mu^m_d)^2}{\sum_{n,d}(y^m_{nd} - \mu^m_d)^2}. \tag{14}$$

Here, μ^m_d denotes the view- and feature-wise intercept term of the model estimated either by feature-wise mean or from the weights of a constant factor if included in the model, as described in section 1.6.3. Subsequently, each factor is characterised by three complementary analyses:

- **Inspection of the samples in factor space:** By visualizing the latent factors we obtain a low-dimensional representation which captures the main drivers of variation and can yield important insights into sample heterogeneity, similar to Principal Component Analysis. This can be done via beeswarm plots (for individual factors) or scatter plots (for factor pairs), which can be coloured or shaped by known covariates. Alternatively, one can directly evaluate the correlation of the latent factors to the known covariates. This is the most useful approach for the detection of technical sources of variation such as batch effects or sample quality.

- **Inspection of top weighted features in active views:** When the variation associated to a given factor is driven by a small number of features, the inspection of the features with highest loading can reveal which biological process underlies the variation captured by the latent factor.

- **Feature set enrichment analysis (FSEA) in active views:** Inspecting the loadings of single features can be challenging in many cases, especially if the effect size per feature is small. Thus, we provide tools for feature set enrichment analysis to combine the signal from functionally related sets of features (e. g., gene sets [8]) and derive a feature-set based factor annotation.

1.5 Comparison to existing group factor analysis models

MOFA builds upon the Group Factor Analysis (GFA) framework [20, 13, 6, 21] (see Table 1). This class of models aims at explaining dependencies between groups of variables (views) instead of dependencies between individual variables, as standard factor analysis does [1]. Several implementations of this framework are available, with significant differences in the modelling strategy and inference framework. The strategy for disentangling sources of variation in multi-view data was introduced in [20] and [13], where a view and factor-wise (ARD) sparsity was proposed. Other works have focused on introducing new feature-wise sparsity assumptions on the weights, e.g. for tensor decomposition or bi-clustering applications [21], [10] and [6]. Also, to detect weak shared signals [17] extended GFA to a mixture model.
In MOFA, we extended the initial model proposed in [20] with three key features to make it applicable to a wide range of multi-omics data sets: (a) Combination of two-level sparsity with fast variational inference, (b) explicit modelling of non-Gaussian data types, and (c) implementation able of handling missing values.

Publication	**Inference**	**Group-wise sparsity**	**Feature-wise sparsity**	**Missing values**	**Likelihood**	**Noise model**
Virtanen et al, 2012	VB	ARD	None	No	Gaussian	Homo-scedastic
Klami et al, 2014	VB	ARD	None	No	Gaussian	Homo-scedastic
Bunte et al, 2016	Gibbs	ARD	Spike and Slab	No	Gaussian	Homo-scedastic
Hore et al, 2016	VB	None	Spike and Slab	Yes	Gaussian	Hetero-scedastic
Remes et al, 2016	VB	ARD	None	No	Gaussian	Homo-scedastic
Zhao et al, 2015	Gibbs	ARD	Three-parameter beta prior	No	Gaussian	Hetero-scedastic
MOFA	VB	ARD	Spike and Slab	Yes	Gaussian, Poisson, Bernoulli	Hetero-scedastic

Table 1: **Overview of GFA methods.** Abbreviations used: VB (variational Bayes inference), Gibbs (Gibbs sampling based inference), ARD(Automatic Relevance Determination)

1.6 Implementation and practical considerations for training

1.6.1 Monitoring convergence

In contrast to sampling methods, variational approximations have the appealing property that convergence is easily monitored by changes in the ELBO, which is required to increase monotonically [4]. In practice, we set a default threshold for convergence corresponding to a change in ELBO smaller than 0.1%.

1.6.2 Handling of missing values

The model naturally accounts for missing values, as non-observed data points do not intervene in the likelihood and can be ignored in the update equations. In practice, we use a binary mask $\mathcal{O}^m \in \mathbb{R}^{N \times D_m}$ for each view m, such that $\mathcal{O}_{n,d} = 1$ when feature d is observed for sample n, 0 otherwise.

1.6.3 Data pre-processing

MOFA does not require the data to be centered or scaled. The first property is achieved by incorporating a constant intercept term in the latent factor matrix that is not updated during training. The corresponding weight vectors are initialised to the true means, so that the rest of the factors capture variation independent of the mean. The second property is achieved by the factor- and view-wise sparsity, which allows different scales of the weights for each view.

Also, when using the Gaussian noise model, it is recommended to use methods for variance stabilisation (e.g. as implemented in [14] for RNAseq data) prior to model training in order to make the normality assumption of the model more appropriate.

1.6.4 Robustness

A drawback of the iterative variational Bayes algorithm is that it is not guaranteed to find the optimal solution [4]. In practice, for an exploratory analysis we suggest using a single fit, which in most of the cases yields an acceptable solution. For a more robust inference, we adopt common practice [10] and run MOFA multiple times (e.g. 10 trials) under different initialisations and we select the model with the highest ELBO. Also, for an extensive assessment of the robustness, we recommend checking the consistency of factors across multiple runs as well as under downsampling of the data. See section 3.2 for robustness analysis in the CLL data.

1.6.5 Determining the number of factors

The model automatically learns the dimensionality of the factor space by removing inactive factors during training if they do not explain significant variation in any view. This is achieved by the view- and factor-wise ARD prior (Eq. (7)). In practice, a threshold is required to define a factor as inactive, which depends on the aim and the data set. For example, to do an exploratory analysis of the main sources of variation only a few number of relatively strong factors is required, so one can set the threshold of activity to roughly 3% of variance explained in at least one view. In contrast, if doing predictions then even minor sources of variation can be important and the threshold can be reduced to less than 1% of variance explained.

1.6.6 Rotational invariance

An important consequence of the definition of MOFA (and most factor analysis models [1, 20]) is their unidentifiability due to rotational and scaling invariance, which means that the factors and corresponding loadings can only be identified up to an orthogonal rotation. This property needs to be kept in mind when comparing different models and values in the factor and weight space should be interpreted relative to each other.

1.6.7 Code availability

The MOFA model is implemented as a Python package and the downstream analysis functions are implemented as an R package, both accessible at: http://github.com/PMBio/MOFA.
Simulation and analysis code are accessible at http://github.com/PMBio/MOFA_CLL.

2 Model validation using simulations

2.1 Recovery of the true number of factors

To assess the technical capabilities of MOFA we validated the model using observations simulated from the generative model, where we varied the number of views, the number of features, the number of factors and the fraction of missing values. In particular, to constrain our simulations to realistic multi-omics scenarios, we ranged the number of views from 1 to 20, the number of features from 100 to 10,000, the number of factors from 5 to 60 and the percentage of missing values from 10 to 90%. The number of samples was fixed to 100.
All trials were started with a sufficiently high number of factors ($k = 100$), and inactive factors were dropped as described in section 1.6.5, with a threshold of 3% of variance explained in at least one view. To test the robustness under different random initialisations, ten models were trained for every configuration.

In most of the settings the model robustly recovered the correct number of factors (Figure S2). Exceptions occured when the dimensionality of the latent space was too large (more than 50 factors) (Figure S2a) or when an excessive amount of missing values (more than 80%) was present in the data (Figure S2d). Little variability is observed across different initialisations.

2.2 Non-Gaussian data

A key improvement of MOFA with respect to previous methods is the use of non-Gaussian likelihoods to properly model different data modalities. In particular, we implemented a Bernoulli likelihood to model binary data and a Poisson likelihood to model count data.
To assess the performance of both likelihood models, we simulated binary and count data using the generative model and we fit two sets of models for each data type: a group of models with a Gaussian likelihood and a group of models with a Bernoulli and Poisson likelihood, respectively.
Although both likelihoods are able to recover the true number of factors, the models with the non-Gaussian likelihoods clearly result in a better fit to the data, with distributions of the predicted values closer to the true shape of the non-Gaussian distributions (Figure S3 and Figure S4).

2.3 Identifying co-variation patterns across data sets

The main task of MOFA is to disentangle the different sources of variation in a multi-view data set. To evaluate its performance on this task, we simulated data from the generative model where the factors were clearly set to be active or inactive in a specific view, and we assessed whether MOFA recovers the true activity of the factors. Subsequently, we also compared the performance with iCluster, a commonly used method in multi-omics studies. In principle, the latent variable model underlying iCluster is focused on clustering of samples, but it can also be used to perform variance decomposition. However, its underlying assumptions are not suited for this task, because in a given view the same amount of regularization is used for each latent factor. Therefore, the model is unable to properly distinguish factors that drive variation in distinct subsets of views.
MOFA correctly infers the true activity pattern of the factors per view in all settings while iCluster infers incorrect sharedness of factors across views, especially with increasing dimensionality of the latent space (Figure S6).

3 Detailed methods on CLL analysis

3.1 Data processing

The data was obtained from [7] where details on the data generation and processing can be found. For the training of MOFA we included 62 drug response measurements (excluding NSC 74859 and bortezomib due to bad quality) at five concentrations each ($d = 310$) with a threshold at 1.1 to remove outliers. Mutations were considered if present in at least 3 samples ($d = 69$). Low counts from RNAseq data were filtered out and the data was normalized using the *estimateSizeFactors* and *varianceStabilizingTransformation* function of DESeq2 [14]. For training we used the top $d = 5000$ most variable mRNAs after exclusion of genes from the Y chromosome. Methylation data was transformed to M-values and we extracted the top 1% most variable CpG sites excluding sex chromosomes ($d = 4248$). We included patients diagnosed with CLL and having data in at least two views into the MOFA model leading to a total of $n = 200$ samples.

3.2 Robustness

Applying MOFA to this data set we recovered up to ten latent factors explaining a minimum of 3% of variation in at least one view (Figure S7a,b). The inferred factors and weights are robust to the random initializations (Figure S7c,d) as well as to downsampling of the data (Figure S8). Also, the MOFA factors show near orthogonality, as opposed to the strongly correlated factors inferred by iCluster (Figure S9), demonstrating that MOFA captures independent sources of variation.
Model selection was performed as described in section 1.6.4 and a single model was selected for downstream analysis, which is highlighted in Figure S7b.

3.3 Inspection of loadings

The first step to characterise the latent factors is the direct inspection of loadings. Importantly, the scale of the weights inferred by the MOFA model are proportional to the scale of the corresponding observations. Therefore, the weights of views with different scale (i.e mRNA and drug response) cannot be directly compared. Only the values of weights from the same view are directly comparable. For this reason, for visualisation purposes and to facilitate interpretation all loadings are always displayed in a

relative scale from 0 to 1.

3.4 Gene set enrichment analysis

Gene set enrichment analysis is performed using a parametric t-test comparing the means of the foreground set (the weights of genes that belong to gene set $\mathcal{G}$) and the background set (the weights of genes that do not belong to gene set $\mathcal{G}$), similar to [9]. Resulting p-values are adjusted for multiple testing on each factor using Benjamini-Hochberg procedure to control the false discovery rate at a chosen level (here $\alpha = 0.01$) [3].

3.5 Imputation

Unobserved data points are directly imputed from the MOFA model equation

$$\mathbf{Y}^m = \mathbf{Z}\mathbf{W}^{mT} \tag{15}$$

To compare imputation performance, we trained MOFA on the subset of samples with all measurements ($n = 121$) and masked at random either single values or all measurements for given samples in the drug response. The number of factors used for the task was learned by MOFA as described in section 1.6.5, but different thresholds for dropping factors were used depending on the type of imputation. For imputing values missing at random, even minor sources of variation are required so we decreased the threshold of explained variance from 3% to 0.01%. In contrast, to impute full assays, it is important to have a few strong and shared factors that contain the covariation between the data modalities, so we increased the threshold of explained variance from 3% to 10%.

3.6 Survival Analysis

In order to assess the association of MOFA factors with clinical outcome we used time to next treatment as response variable in a Cox proportional hazard model including all samples for which this information was available, i.e. $n = 174$ of which 96 are uncensored cases. For univariate associations (as shown in Figure 5a) we scaled all predictors to ensure comparability of the hazard ratios and we rotated factors (see section 1.6.6) such that their Hazard ratio is greater or equal to 1.
To investigate the predictive power of different datasets, we used a multivariate Cox model and compared Harrell's C-index of predictions in a stratified 5-fold cross-validation scheme. As predictors we included the top 10 principal components on the data of each single view, a concatenated data set ('all') as well as the 10 MOFA factors. Missing values in a view were imputed by the feature-wise mean. In a second set of models we used the complete set of all features in a view and used a ridge penalty in the Cox model as implemented in the R package *glmnet* to get predictions based on each view as well as the concatenated data, which leads to similar prediction performance as the principal component approach (Figure S17). The Kaplan Meier plots were generated using an optimal cut point on each factor calculated based on the maximally selected rank statistics as implemented in the R package *survminer* with p-values based on a Log-Rank test between the resulting groups.

Appendix

This appendix contains detailed explanations of the model inference, and is divided in three sections. The first section gives a general introduction to variational inference. The second section describes the derivation of the updates for the variational Bayesian algorithm, including the computation of the ELBO, for the Gaussian model. Finally, the third section describes the derivation of the updates for the extension to non-Gaussian likelihoods.

Mathematical notation

- Matrices are denoted with bold capital letters: $\mathbf{W}$
- Vectors are denoted with bold non-capital letters. If the vector comes from a matrix, two indices separated by a comma will always be shown at the bottom: the first one corresponding to the row and the second one to the column. The symbol ':' denotes the entire row/column. For instance, $\mathbf{w}_{j,:}$ refers to the entire jth row from the $\mathbf{W}$ matrix.
- Scalars are denoted with non-bold non-capital letters. If the value comes from a matrix, two indices separated by a comma will always be shown at the bottom: the first one corresponding to the row and the second one to the column. For instance, $w_{j,k}$ refers to the value coming from the jth row and the kth column from the $\mathbf{W}$ matrix.
- $\mathbf{0}_k$ is a zero vector of length K.
- $\mathbf{I}_k$ is the identity matrix with rank K.
- $\mathbb{E}_q[x]$ denotes the expectation of x under the distribution q. Sometimes, when the expectations are taken with respec to the same distribution many times, to avoid cluttered notation we will use $\langle x \rangle$.
- $\mathcal{N}\left(x \,|\, \mu, \sigma\right)$: x follows a univariate normal distribution with mean μ and variance σ.
- $\mathcal{G}\left(x \,|\, a, b\right)$: x follows a gamma distribution with parameters a and b.
- $diag(\mathbf{x})$ is the diagonal operator that takes as input a vector and outputs a diagonal matrix with $\mathbf{x}$ in the diagonal.

Introduction to variational Bayes inference

Briefly, in variational inference the true intractable posterior distribution of the unobserved variables $p(\mathbf{X}|\mathbf{Y})$ is approximated by a simpler distribution of factorized form $q(\mathbf{X}) = \prod_i q(\mathbf{X}_i)$ that leads to an efficient inference scheme. Here, $\mathbf{X}$ denotes all the hidden variables (including parameters) and $\mathbf{Y}$ denotes all the observed variables.
Under this approximation, the true log marginal likelihood $\log p(\mathbf{Y})$ is lower bounded by:

$$\begin{aligned} \mathcal{L}(\mathbf{X}) &= \int q(\mathbf{X}) \Big(\log \frac{p(\mathbf{X}|\mathbf{Y})}{q(\mathbf{X})} + \log p(\mathbf{Y}) \Big) d\mathbf{X} \\ &= \log p(\mathbf{Y}) - \mathrm{KL}(q(\mathbf{X}) || p(\mathbf{X}|\mathbf{Y})) \\ &\leq \log p(\mathbf{Y}) \end{aligned} \tag{16}$$

$\mathcal{L}(\mathbf{X})$ is defined as the Evidence Lower Bound (ELBO), which is equal to the sum of the model evidence and the negative KL-divergence between the true posterior and the variational distribution. The key observation here is that increasing the ELBO is equivalent to decreasing the KL-divergence between the two distributions.
Variational learning involves optimising the functional $\mathcal{L}(\mathbf{X})$ with respect to the distribution $q(\mathbf{X})$. If we allow any possible choice of $q(\mathbf{X})$, then the maximum of the lower bound $\mathcal{L}(\mathbf{X})$ will occur when the KL-divergence vanishes, which occurs when $q(\mathbf{X})$ equals the true posterior distribution $p(\mathbf{X}|\mathbf{Y})$. Nevertheless, since the true posterior is intractable, this does not lead to any simplification of the problem. Instead, it is necessary to consider a restricted family of variational distributions that are tractable to compute and then seek the member of this family for which the KL divergence is minimised [2].

Mean-field approximation

The most common type of variational Bayes, known as mean-field approach, assumes that the variational distribution factorises over M disjoint groups of variables:

$$q(\mathbf{X}) = \prod_{i=1}^{M} q(\mathbf{x}_i)$$

Evidently, this family of distributions do not usually contain the true posterior because the unobserved variables have dependencies, but this is a key assumption to obtain an analytical inference scheme [2]. It follows that the optimal distribution $\hat{q}_i$ that maximises the lower bound $\mathcal{L}(\mathbf{X})$, for each variable $\mathbf{x}_i$, can be calculated as follows:

$$\log \hat{q}_i(\mathbf{x}_i) = \mathbb{E}_{i \neq j}[\log p(\mathbf{Y}, \mathbf{X})] + \text{const} \tag{17}$$

where $\mathbb{E}_{i \neq j}$ denotes an expectation with respect to the q distributions over all variables $\mathbf{x}_j$ except for $\mathbf{x}_i$. The additive constant is set by normalising the distribution $\hat{q}_i(\mathbf{z}_i)$:

$$\hat{q}_i(\mathbf{x}_i) = \frac{\exp(\mathbb{E}_{i \neq j}[\log p(\mathbf{Y}, \mathbf{X})])}{\int \exp(\mathbb{E}_{i \neq j}[\log p(\mathbf{Y}, \mathbf{X})]) d\mathbf{X}}$$

This is the general expression which yields the set of variational distributions that maximise the lower bound of the log marginal likelihood, subject to the factorisation constraint. Or equivalently, the set of distributions that minimise the KL divergence between the $q(\mathbf{X})$ distribution and the true posterior $p(\mathbf{X})$.

Variational Bayes expectation maximization algorithm

Note that in Equation (17), for a given variable $\mathbf{x}_i$, the expectation on the right-hand side is taken with respect to the other variables' variational distribution $q_j(\mathbf{x}_j)$ for $j \neq i$. Therefore, there are circular dependencies between the different equations and there is no analytical solution for the parameters of the variational distribution. This naturally suggests an iterative algorithm similar to the Expectation Maximisation (EM) algorithm. In each step we update the moments and parameters of the variational distribution of the latent variables $q_j(\mathbf{x}_j)$ using the current estimates of the variational distributions of the parameters $q_{-j}(\mathbf{x}_{-j})$ [2]. The algorithm is stopped when the change in the ELBO is small enough.

Update equations for Gaussian data

Latent variables

Variational distribution:

$$q(\mathbf{Z}) = \prod_{k=1}^{K} \prod_{n=1}^{N} q(z_{nk}) = \prod_{k=1}^{K} \prod_{n=1}^{N} \mathcal{N}\left(z_{nk} \,|\, \mu_{z_{nk}}, \sigma_{z_{nk}}\right)$$

where

$$\sigma^2_{z_{nk}} = \Big(\sum_{m=1}^{M} \sum_{d=1}^{D_m} \tau_d^m \langle (s_{dk}^m \hat{w}_{dk}^m)^2 \rangle + 1 \Big)^{-1}$$

$$\mu_{z_{nk}} = \sigma^2_{z_{nk}} \sum_{m=1}^{M} \sum_{d=1}^{D_m} \langle \tau_d^m \rangle \langle s_{dk}^m \hat{w}_{dk}^m \rangle \Big(y_{nd}^m - \sum_{j \neq k} \langle s_{dj}^m \hat{w}_{dj}^m \rangle \langle z_{nj} \rangle \Big)$$

Spike and Slab weights

Variational distribution:

$$q(\hat{\mathbf{W}}, \mathbf{S}) = \prod_{m=1}^{M} \prod_{d=1}^{D_m} \prod_{k=1}^{K} q(\hat{w}_{dk}^m, s_{dk}^m) = \prod_{m=1}^{M} \prod_{d=1}^{D_m} \prod_{k=1}^{K} q(\hat{w}_{dk}^m | s_{dk}^m) q(s_{dk}^m)$$

Update for $q(s_{dk}^m)$:

$$\gamma_{dk^m} = q(s_{dk} = 1) = \frac{1}{1 + \exp(-\lambda_{dk^m})},$$

where

$$\lambda_{dk}^m = \langle \log \frac{\theta}{1-\theta} \rangle + 0.5 \log \frac{\langle \alpha_k^m \rangle}{\langle \tau_d^m \rangle} - 0.5 \log \Big(\sum_{n=1}^{N} \langle z_{nk}^2 \rangle + \frac{\langle \alpha_k^m \rangle}{\langle \tau_d^m \rangle} \Big)$$
$$+ \frac{\langle \tau_d^m \rangle}{2} \frac{\Big(\sum_{n=1}^{N} y_{nd}^m \langle z_{nk} \rangle - \sum_{j \neq k} \langle s_{dj}^m \hat{w}_{dj}^m \rangle \sum_{n=1}^{N} \langle z_{nk} \rangle \langle z_{nj} \rangle \Big)^2}{\sum_{n=1}^{N} \langle z_{nk}^2 \rangle + \frac{\langle \alpha_k^m \rangle}{\langle \tau_d^m \rangle}}$$

Update for $q(\hat{w}_{dk}^m)$:

$$q(\hat{w}_{dk}^m|s_{dk}^m = 0) = \mathcal{N}\left(\hat{w}_{dk}^m \,|\, 0, 1/\alpha_k^m\right),$$
$$q(\hat{w}_{dk}^m|s_{dk}^m = 1) = \mathcal{N}\left(\hat{w}_{dk}^m \,|\, \mu_{w_{dk}^m}, \sigma^2_{w_{dk}^m}\right),$$

where

$$\mu_{w_{dk}^m} = \frac{\sum_{n=1}^N y_{nd}^m \langle z_{nk}\rangle - \sum_{j\neq k}\langle s_{dj}^m \hat{w}_{dj}^m\rangle \sum_{n=1}^N \langle z_{nk}\rangle\langle z_{nj}\rangle}{\sum_{n=1}^N \langle z_{nk}^2\rangle + \frac{\langle \alpha_k^m\rangle}{\langle \tau_d^m\rangle}}$$
$$\sigma_{w_{dk}^m} = \frac{\langle \tau_d^m\rangle^{-1}}{\sum_{n=1}^N \langle z_{nk}^2\rangle + \frac{\langle \alpha_k^m\rangle}{\langle \tau_d^m\rangle}}$$

Taken together this means that we can update $q(\hat{w}_{dk}^m, s_{dk}^m)$ using:

$$q(\hat{w}_{dk}^m|s_{dk}^m) \times q(s_{dk}^m) = \mathcal{N}\left(\hat{w}_{dk}^m \,|\, s_{dk}^m \mu_{w_{dk}^m}, s_{dk}^m \sigma^2_{w_{dk}^m} + (1 - s_{dk}^m)/\alpha_k^m\right) \quad \times \quad (\lambda_{dk}^m)^{s_{dk}^m}(1-\lambda_{dk}^m)^{1-s_{dk}}$$

ARD precision (alpha)

Variational distribution:

$$q(\boldsymbol{\alpha}) = \prod_{m=1}^M \prod_{k=1}^K \mathcal{G}(\alpha_k^m|\hat{a}_{mk}^\alpha, \hat{b}_{mk}^\alpha)$$

where

$$\hat{a}_{mk}^\alpha = a_0^\alpha + \frac{D_m}{2}$$
$$\hat{b}_{mk}^\alpha = b_0^\alpha + \frac{\sum_{d=1}^{D_m}\langle (\hat{w}_{d,k}^m)^2\rangle}{2}$$

Noise precision (tau)

Variational distribution:

$$q(\boldsymbol{\tau}) = \prod_{m=1}^M \prod_{d=1}^{D_m} q(\tau_d^m) = \prod_{m=1}^M \prod_{d=1}^{D_m} \mathcal{G}(\tau_d^m|\hat{a}_{md}^\tau, \hat{b}_{md}^\tau)$$

where

$$\hat{a}_{md}^\tau = a_0^\tau + \frac{N}{2}$$
$$\hat{b}_{md}^\tau = b_0^\tau + \frac{1}{2}\sum_{n=1}^N \langle (y_{nd}^m - \sum_k^K \hat{w}_{dk}^m s_{dk}^m z_{n,k})^2\rangle$$

Spike and Slab sparsity parameter (theta)

Variational distribution:

$$q(\theta) = \prod_{m=1}^M \prod_{k=1}^K \mathrm{Beta}(\theta_k^m|\hat{a}_{mk}^\theta, \hat{b}_{mk}^\theta),$$

where

$$\hat{a}_{mk}^\theta = \sum_{d=1}^{D_m}\langle s_{dk}^m\rangle + a_0^\theta$$
$$\hat{b}_{mk}^\theta = b_0^\theta - \sum_{d=1}^{D_m}\langle s_{dk}^m\rangle + D_m$$

Evidence Lower bound

In order to monitor training and assess convergence we calculate the ELBO alongside with the other updates. The ELBO can be decomposed into a likelihood term and terms for each model variable $\mathbf{X}_i$:

$$\begin{aligned}\mathcal{L}(\mathbf{X}) &= \int q(\mathbf{X})\Big(\log \frac{p(\mathbf{X},\mathbf{Y})}{q(\mathbf{X})}\Big)d\mathbf{X} \\ &= \mathbb{E}_q \log p(\mathbf{Y}|\mathbf{X})\rangle + \sum_i \left(\mathbb{E}_q \log p(\mathbf{X}_i) - \mathbb{E}_q \log q(\mathbf{X}_i)\right),\end{aligned}$$

where the expectation is under the variational distribution of the current step. Each of the terms from the last term is computed as follows:

Likelihood term

If using the gaussian likelihood:

$$-\sum_{m=1}^{M}\frac{ND_m}{2}\log(2\pi) + \frac{N}{2}\sum_{m=1}^{M}\sum_{d=1}^{D_m}\log(\langle\tau_d^m\rangle) - \sum_{m=1}^{M}\sum_{d=1}^{D_m}\frac{\langle\tau_d^m\rangle}{2}\sum_{n=1}^{N}\left(y_{nd}^m - \sum_{k=1}^{K}\langle s_{dk}^m \hat{w}_{dk}^m\rangle\langle z_{nk}\rangle\right)^2$$

When not using the gaussian model, this expression is replaced by the corresponding likelihood.

W and S terms

$$\begin{aligned}\mathbb{E}_q[\log p(\hat{\mathbf{W}},\mathbf{S})] = &-\sum_{m=1}^{M}\frac{KD_m}{2}\log(2\pi) + \sum_{m=1}^{M}\frac{D_m}{2}\sum_{k=1}^{K}\log(\alpha_k^m) - \sum_{m=1}^{M}\frac{\alpha_k^m}{2}\sum_{d=1}^{D_m}\sum_{k=1}^{K}\langle(\hat{w}_{dk}^m)^2\rangle \\ &+ \langle\log(\theta)\rangle\sum_{m=1}^{M}\sum_{d=1}^{D_m}\sum_{k=1}^{K}\langle s_{dk}^m\rangle + \langle\log(1-\theta)\rangle\sum_{m=1}^{M}\sum_{d=1}^{D_m}\sum_{k=1}^{K}(1-\langle s_{dk}^m\rangle)\end{aligned}$$

$$\begin{aligned}\mathbb{E}_q[\log q(\hat{\mathbf{W}},\mathbf{S})] = &-\sum_{m=1}^{M}\frac{KD_m}{2}\log(2\pi) + \frac{1}{2}\sum_{m=1}^{M}\sum_{d=1}^{D_m}\sum_{k=1}^{K}\log(\langle s_{dk}^m\rangle\sigma_{w_{dk}^m}^2 + (1-\langle s_{dk}^m\rangle)/\alpha_k^m) \\ &+ \sum_{m=1}^{M}\sum_{d=1}^{D_m}\sum_{k=1}^{K}(1-\langle s_{dk}^m\rangle)\log(1-\langle s_{dk}^m\rangle) - \langle s_{dk}^m\rangle\log\langle s_{dk}^m\rangle\end{aligned}$$

Z term

$$\begin{aligned}\mathbb{E}_q[\log p(\mathbf{Z})] &= -\frac{NK}{2}\log(2\pi) - \frac{1}{2}\sum_{n=1}^{N}\langle z_{nk}^2\rangle \\ \mathbb{E}_q[\log q(\mathbf{Z})] &= -\frac{NK}{2}(1+\log(2\pi)) - \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K}\log(\sigma_{z_{nk}}^2)\end{aligned}$$

alpha term

$$\begin{aligned}\mathbb{E}_q[\log p(\boldsymbol{\alpha})] &= \sum_{m=1}^{M}\sum_{k=1}^{K}\Big(a_0^\alpha\log b_0^\alpha + (a_0^\alpha - 1)\langle\log\alpha_k\rangle - b_0^\alpha\langle\alpha_k\rangle - \log\Gamma(a_0^\alpha)\Big) \\ \mathbb{E}_q[\log q(\boldsymbol{\alpha})] &= \sum_{m=1}^{M}\sum_{k=1}^{K}\Big(\hat{a}_k^\alpha\log\hat{b}_k^\alpha + (\hat{a}_k^\alpha - 1)\langle\log\alpha_k\rangle - \hat{b}_k^\alpha\langle\alpha_k\rangle - \log\Gamma(\hat{a}_k^\alpha)\Big)\end{aligned}$$

tau term

$$\mathbb{E}_q[\log p(\tau)] = \sum_{m=1}^{M} D_m a_0^\tau \log b_0^\tau + \sum_{m=1}^{M}\sum_{d=1}^{Dm}(a_0^\tau - 1)\langle \log \tau_d^m \rangle - \sum_{m=1}^{M}\sum_{d=1}^{Dm} b_0^\tau \langle \tau_d^m \rangle - \sum_{m=1}^{M} D_m \Gamma(a_0^\tau)$$

$$\mathbb{E}_q[\log q(\boldsymbol{\tau})] = \sum_{m=1}^{M}\sum_{d=1}^{D_m}\Big(\hat{a}_{dm}^\tau \log \hat{b}_{dm}^\tau + (\hat{a}_{dm}^\tau - 1)\langle \log \tau_d^m \rangle - \hat{b}_{dm}^\tau \langle \tau_d^m \rangle - \log \Gamma(\hat{a}_{dm}^\tau)\Big)$$

theta term

$$\mathbb{E}_q\left[\log p(\theta)\right] = \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{d=1}^{D_m}\left((a_0 - 1) \times \langle \log(\pi_{d,k}^m)\rangle + (b_0 - 1)\langle \log(1 - \pi_{d,k}^m)\rangle - \log(\mathrm{B}(a_0, b_0))\right)$$

$$\mathbb{E}_q\left[\log q(\theta)\right] = \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{d=1}^{D_m}\left((a_{k,d}^m - 1) \times \langle \log(\pi_{d,k}^m)\rangle + (b_{k,d}^m - 1)\langle \log(1 - \pi_{d,k}^m)\rangle - \log(\mathrm{B}(a_{k,d}^m, b_{k,d}^m))\right)$$

Updates for non-Gaussian data

To implement efficient variational inference in conjunction with a non-Gaussian likelihood we adapt prior work from [18] using local variational bounds. The key idea is to dynamically approximate non-Gaussian data by Gaussian pseudo-data based on a second-order Taylor expansion. To make the approximation justifiable we need to introduce variational parameters that are adjusted alongside the updates to improve the fit.

Denoting the parameters in the MOFA model as $\mathbf{X} = (\mathbf{Z}, \mathbf{W}, \boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\theta})$, recall that the variational framework approximates the posterior $p(\mathbf{X}|\mathbf{Y})$ with a distribution $q(\mathbf{X})$, which is indirectly optimised by optimising a lower bound $\mathcal{F}$ of the log model evidence. The resulting optimization problem can be re-written from Equation (16) as

$$\min_{q(\mathbf{X})} \mathcal{F} = \min_{q(\mathbf{X})} -\mathcal{L}(\mathbf{X}) = \min_{q(\mathbf{X})} \mathbb{E}_q\big[-\log p(\mathbf{Y}|\mathbf{X})\big] + \mathrm{KL}[q(\mathbf{X})||p(\mathbf{X})].$$

Expanding the MOFA model to non-Gaussian likelihoods we now assume a general likelihood of the form $p(\mathbf{Y}|\mathbf{X}) = p(\mathbf{Y}|\mathbf{C})$ with $\mathbf{C} = \mathbf{Z}\mathbf{W}^T$, that can write as

$$-\log p(\mathbf{Y}|\mathbf{X}) = \sum_{n=1}^{N}\sum_{d=1}^{D} f_{nd}(c_{nd})$$

with $f_{nd}(c_{nd}) = -\log p(y_{nd}|c_{nd})$. We dropped the view index m to keep notation uncluttered.

Extending [18] to our heteroscedastic noise model, we require $f_{nd}(c_{nd})$ to be twice differentiable and bounded by κ_d, such that $f''_{nd}(c_{nd}) \leq \kappa_d \forall n, d$. This holds true in many important models like for example the Bernoulli and Poisson case. Under this assumption a lower bound on the log likelihood can be constructed using Taylor expansion,

$$f_{nd}(c_{nd}) \leq \frac{\kappa_d}{2}(c_{nd} - \zeta_{nd})^2 + f'(\zeta_{nd})(c_{nd} - \zeta_{nd}) + f_{nd}(\zeta_{nd}) := q_{nd}(c_{nd}, \zeta_{nd}),$$

where $\boldsymbol{\zeta} = \zeta_{nd}$ are additional variational parameters that determine the location of the Taylor expansion and have to be optimised to make the lower bound as tight as possible. Plugging the bounds into above optimization problem, we obtain:

$$\min_{q(\mathbf{X}),\boldsymbol{\zeta}} \quad \sum_{d=1}^{D}\sum_{n=1}^{N} \mathbb{E}_q[q_{nd}(c_{nd}|\zeta_{nd})] + \mathrm{KL}[q(\mathbf{X})||p(\mathbf{X})]$$

The algorithm propsed in [18] then alternates between updates of $\boldsymbol{\zeta}$ and q($\boldsymbol{\Theta}$). The update for $\boldsymbol{\zeta}$ is given by

$$\zeta \leftarrow \mathbb{E}[\mathbf{W}]\mathbb{E}[\mathbf{Z}]^T$$

where the expctations are taken with respect to the corresponding q distributions.

On the other hand, the updates for $q(\mathbf{X})$ can be shown to be identical to the variational Bayesian updates with a conjugated Gaussian likelihood when replacing the observed data $\mathbf{Y}$ by a pseudo-data $\hat{\mathbf{Y}}$ and the

precisions τ_{nd} (which were treated as random variables) by the constant terms κ_d introduced above. The pseudodata is given by

$$\hat{y}_{nd} = \zeta_{nd} - f'(\zeta_{nd})/\kappa_d.$$

Depending on the log likelihoods $f(\cdot)$ different κ_d are used resulting in different pseudo-data updates. Two special cases implemented in MOFA are the Poisson and Bernoulli likelihood describe in the following.

Bernoulli likelihood for binary data

When the observations are binary, $y \in \{0, 1\}$, they can be modelled using a Bernoulli likelihood:

$$\mathbf{Y}|\mathbf{Z}, \mathbf{W} \sim \text{Ber}(\sigma(\mathbf{Z}\mathbf{W}^T)),$$

where $\sigma(a) = (1 + e^{-a})^{-1}$ is the logistic link function and $\mathbf{Z}$ and $\mathbf{W}$ are the latent factors and weights in our model, respectively.
In order to make the variational inference efficient and explicit as in the Gaussian case, we aim to approximate the Bernoulli data by a Gaussian pseudo-data as proposed in [18] and described above which allows to recycle all the updates from the model with Gaussian views. While [18] assumes a homoscedastic approximation with a spherical Gaussian, we adopt an approach following [11], which allows for heteroscedaticity and provides a tighter bound on the Bernoulli likelihood.
Denoting $c_{nd} = (\mathbf{Z}\mathbf{W}^T)_{nd}$ the Jaakkola upper bound [11] on the negative log-likelihood is given by

$$\begin{aligned} -\log\left(p(y_{nd}|c_{nd})\right) &= -\log\left(\sigma\left((2y_{nd}-1)c_{nd}\right)\right) \\ &\leq -\log(\zeta_{nd}) - \frac{(2y_{nd}-1)c_{nd} - \zeta_{nd})}{2} + \lambda(\zeta_{nd})\left(c_{nd}^2 - \zeta_{nd}^2\right) \\ &=: b_J(\zeta_{nd}, c_{nd}, y_{nd}) \end{aligned}$$

with λ given by $\lambda(\zeta) = \frac{1}{4\zeta}\tanh\left(\frac{\zeta}{2}\right)$.
This can easily be derived from a first-order Taylor expansion on the function $f(x) = -\log(e^{\frac{x}{2}} + e^{-\frac{x}{2}}) = \frac{x}{2} - \log(\sigma(x))$ in x^2 and by the convexity of f in x^2 this bound is global as discussed in [11].
In order to make use of this tighter bound but still be able to re-use the variational updates from the Gaussian case we re-formulate the bound as a Gaussian likelihood on pseudo-data $\hat{\mathbf{Y}}$.
As above we can plug in the bound on the negative log-likelihood in the variational optimization problem to obtain

$$\min_{q(\mathbf{X}),\boldsymbol{\zeta}} \quad \sum_{d=1}^{D}\sum_{n=1}^{N} \mathbb{E}_q b_J(\zeta_{nd}, c_{nd}, y_{nd}) + \text{KL}[q(\mathbf{X})||p(\mathbf{X})].$$

This is minimized iteratively in the variational parameter ζ_{nd} and the variational distribution of Z,W:
Minimizing in the variational parameter ζ this leads to the updates given by

$$\zeta_{nd}^2 = \mathbb{E}[c_{nd}^2]$$

as described in [11], [4].
For the variational distribution $q(\mathbf{Z}, \mathbf{W})$ we observe that the Jaakkola bound can be re-written as

$$b_J(\zeta_{nd}, c_{nd}, y_{nd}) = -\log\left(\varphi\left(\hat{y}_{nd}; c_{nd}, \frac{1}{2\lambda(\zeta_{nd})}\right)\right) + \gamma(\zeta_{nd}),$$

where $\varphi(\cdot; \mu, \sigma^2)$ denotes the density function of a normal distribution with mean μ and variance σ^2 and γ is a term only depending on $\boldsymbol{\zeta}$. This allows us to re-use the updates for $\mathbf{Z}$ and $\mathbf{W}$ from a setting with Gaussian likelihood by considering the Gaussian pseudo-data

$$\hat{y}_{nd} = \frac{2y_{nd} - 1}{4\lambda(\zeta_{nd})}$$

updating the data precision as $\tau_{nd} = 2\lambda(\zeta_{nd})$ using updates generalized for sample- and feature-wise precision parameters on the data.

Poisson likelihood for count data

When observations are a natural numbers, such as count data $y \in \mathbb{N} = \{0, 1, \cdots\}$, they can be modelled using a Poisson likelihood:

$$p(y|c) = \lambda(c)^y e^{-\lambda(c)}$$

where $\lambda(c) > 0$ is the rate function and has to be convex and log-concave in order to ensure that the likelihood is log-concave.
As done in [18], here we choose the following rate function: $\lambda(c) = \log(1 + e^c)$.
Then an upper bound of the second derivative of the log-likelihood is given by

$$f''_{nd}(c_{nd}) \leq \kappa_d = 1/4 + 0.17 * \max(\mathbf{y}_{:,d})$$

The pseudodata updates are given by

$$\hat{y}_{nd} = \zeta_{nd} - \frac{\mathrm{S}(\zeta_{nd})(1 - y_{nd}/\lambda(\zeta_{nd}))}{\kappa_d}$$

References

[1] A. T. Basilevsky. *Statistical factor analysis and related methods: theory and applications*. Vol. 418. John Wiley & Sons, 2009.

[2] J. Beal. "Variational algorithms for approximate bayesian inference". University College London, 2003.

[3] Y. Benjamini and Y. Hochberg. "Controlling the false discovery rate: a practical and powerful approach to multiple testing". In: *Journal of the royal statistical society. Series B (Methodological)* (1995), pp. 289–300.

[4] C. M. Bishop. "Pattern recognition". In: *Machine Learning* 128 (2006), pp. 1–58.

[5] F. Buettner et al. "f-scLVM: scalable and versatile factor analysis for single-cell RNA-seq". In: *Genome Biology* (in press).

[6] K. Bunte et al. "Sparse group factor analysis for biclustering of multiple data sources". In: *Bioinformatics* 32.16 (2016), pp. 2457–2463.

[7] S. Dietrich, M. Oles, L. Sellner, et al. "Drug Perturbation Based Stratification of Blood Cancer". In: *Journal of Clinical Investigation* (in press).

[8] A. Fabregat et al. "The reactome pathway knowledgebase". In: *Nucleic acids research* 44.D1 (2015), pp. D481–D487.

[9] H. R. Frost, Z. Li, and J. H. Moore. "Principal component gene set enrichment (PCGSE)". In: *BioData mining* 8.1 (2015), p. 25.

[10] V. Hore et al. "Tensor decomposition for multiple-tissue gene expression experiments". In: *Nature Genetics* 48.9 (2016), pp. 1094–1100.

[11] T. S. Jaakkola and M. I. Jordan. "Bayesian parameter estimation via variational methods". In: *Statistics and Computing* 10.1 (2000), pp. 25–37.

[12] S. A. Khan et al. "Identification of structural features in chemicals associated with cancer drug response: a systematic data-driven analysis". In: *Bioinformatics* 30.17 (2014), pp. i497–i504.

[13] A. Klami et al. "Group factor analysis". In: *IEEE transactions on neural networks and learning systems* 26.9 (2015), pp. 2136–2147.

[14] M. I. Love, W. Huber, and S. Anders. "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2". In: *Genome biology* 15.12 (2014), p. 550.

[15] D. J. MacKay. "Bayesian methods for backpropagation networks". In: *Models of neural networks III*. Springer, 1996, pp. 211–254.

[16] T. J. Mitchell and J. J. Beauchamp. "Bayesian variable selection in linear regression". In: *Journal of the American Statistical Association* 83.404 (1988), pp. 1023–1032.

[17] S. Remes, T. Mononen, and S. Kaski. "Classification of weak multi-view signals by sharing factors in a mixture of Bayesian group factor analyzers". In: *arXiv preprint arXiv:1512.05610* (2015).

[18] M. Seeger and G. Bouchard. "Fast variational Bayesian inference for non-conjugate matrix factorization models". In: *Artificial Intelligence and Statistics*. 2012, pp. 1012–1018.

[19] M. K. Titsias and M. Lázaro-Gredilla. "Spike and slab variational inference for multi-task and multiple kernel learning". In: *Advances in neural information processing systems*. 2011, pp. 2339–2347.

[20] S. Virtanen et al. "Bayesian group factor analysis". In: *Artificial Intelligence and Statistics*. 2012, pp. 1269–1277.

[21] S. Zhao et al. "Bayesian group factor analysis with structured sparsity". In: *Journal of Machine Learning Research* 17.196 (2016), pp. 1–47.