Methods for: Multi-omics factor analysis disentangles patient heterogeneity in chronic lymphocytic leukaemia

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

September 10, 2017

Contents

1 Multi-Omics Factor Analysis model

1.1 Model description

The Multi-Omics Factor Analysis (MOFA) model builds upon the Group Factor Analysis framework [3, 7, 11, 12], which aims at explaining dependencies between groups of variables (views) instead of dependencies between individual variables, as standard factor analysis does [1]. In particular, we extended the model proposed in [11] with three key features to make it applicable to a wide range of multi-omics data sets: (a) Two level of sparsities combined with fast variational inference, (b) explicit modelling of non-gaussian data types, and (c) handling of missing values.
This section describes the basic mathematical details of the Gaussian model.

The input data consists of M matrices $\mathbf{Y}^m \in \mathbb{R}^{N \times D_m}$ which are decomposed as follows:

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

where $\mathbf{Z} \in \mathbb{R}^{N \times K}$ are the low-dimensional latent variables, $\mathbf{W}^m \in \mathbb{R}^{D_m \times K}$ are the loadings that relate the high-dimensional space and low dimensional representations, and $\boldsymbol{\epsilon}^m \in \mathbb{R}^{N \times D_m}$ is the noise term. Conventionally, we define the noise to be normally distributed with zero mean and diagonal covariance matrix, which corresponds to heteroskedastic observations:

$$p(\epsilon_d^m) = \mathcal{N}\left(\epsilon_d^m \,|\, 0, 1/\tau_d^m\right),$$

resulting in the following normal likelihood:

$$p(y_{nd}^m) = \mathcal{N}\left(y_{nd}^m \,|\, \mathbf{w}_{d,:}^m \mathbf{z}_{n,:}, 1/\tau_d^m\right)$$

We fit this model in a Bayesian framework and place prior distributions on the weights $\mathbf{W}^m$, the latent variables $\mathbf{Z}$ as well as on the precision of the noise $\mathbf{T}^m$. Conventionally, we use a standard Gaussian prior on the latent variables:

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

and a Gamma distribution for the precision of the noise:

$$p(\tau_d^m) = \mathcal{G}\left(\tau_d^m \,|\, a_0^\tau, b_0^\tau\right)$$

The prior distribution on the weight matrices is the most important part of the model as it encodes the sparsity assumptions. Here we incorporate 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 $\mathbf{w}_{:,k}^m$ is shrunk to zero if the factor k is not driving any variation in view m. In contrast, the feature-wise sparsity enforces zero-weights in individual features that do not drive variation, which yields an interpretable solution with a small number of active features.

To encode both sparsity levels we use a combination of an Automatic Relevance Determination (ARD) prior for the view- and factor-wise sparsity and a spike-and-slab prior for the feature-wise sparsity, similar to [6]. This can be written as:

$$p(\hat{w}_{d,k}^m, s_{d,k}^m) = \mathcal{N}(\hat{w}_{d,k}^m | 0, \frac{1}{\alpha_k^m}) \mathrm{Ber}(s_{d,k}^m | \theta_k^m) \tag{1}$$

$$p(\theta_k^m) = \mathrm{Beta}\left(\theta_k^m \,|\, a_0^\theta, b_0^\theta\right) \tag{2}$$

$$p(\alpha_k^m) = \mathcal{G}\left(\alpha_k^m \,|\, a_0^\alpha, b_0^\alpha\right) \tag{3}$$

In this formulation α_k^m controls the strength of factor k in view m. Therefore, in practice, the ARD prior yields a matrix $\boldsymbol{\alpha} \in \mathbb{R}^{M \times K}$ that defines three different types of factors:

- Factors that explain variation in a single data set (unique factors): all elements in the column vector $\boldsymbol{\alpha}_{:,k}$ are very large except one.
- Factors that explain variation in a subset of data sets (partially shared factors): some elements in the column vector $\boldsymbol{\alpha}_{:,k}$ are very large whereas others are small.
- Factors that explain variation in all data sets (fully shared factors): all elements in the column vector $\boldsymbol{\alpha}_{:,k}$ are small.

In contrast, the feature-wise sparsity models each individual weights as a mixture of a point distribution at zero (spike) and a normal distribution (slab), thereby allowing weights of non-important features to have high density at zero. However, the presence of a Dirac delta mass function makes the application of variational inference troublesome, so here we used a re-parameterization of the spike and slab prior as a product of a Gaussian random variable and a Bernoulli random variable that is more amenable to variational inference [10]. The degree of contribution from the spike term, or the probability of success in the Bernoulli distribution in the re-parametrised form, is controled by θ_k^m, which is also learnt by the model.

This completes the definition of the model, which is graphically illustrated in Fig S1. The joint probability density function is:

$$\begin{aligned} p(\mathbf{Y}, \hat{\mathbf{W}}, \mathbf{S}, \mathbf{Z}, \mathbf{\Theta}) = & \prod_{m=1}^{M} \prod_{n=1}^{N} \prod_{d=1}^{D_m} \mathcal{N}\left(y_{nd}^m \,|\, \sum_{k=1}^{K} s_{dk}^m \hat{w}_{dk}^m z_{nk}, 1/\tau_d\right) \\ & \prod_{d=1}^{D_m} \prod_{k=1}^{K} \mathcal{N}\left(\hat{w}_{dk}^m \,|\, 0, 1/\alpha_k^m\right) \mathrm{Ber}(s_{d,k}^m | \theta_k^m) \\ & \prod_{n=1}^{N} \prod_{k=1}^{K} \mathcal{N}\left(z_{nk} \,|\, 0, 1\right) \\ & \prod_{m=1}^{M} \prod_{k=1}^{K} \mathrm{Beta}\left(\theta_k^m \,|\, a_0^\theta, b_0^\theta\right) \end{aligned} \tag{4}$$

1.2 Model inference

To ensure scalable inference we use a variational approach with a mean-field approximation for all variables except the spike-and-slab weights [2], where we adapt prior work from [10] and introduce a paired mean field approximation $q(w_{d,k}^m, s_{d,k}^m)$. The core idea of variational Bayes is to approximate the true posterior distribution over all unobserved variables using a variational distribution that has a factorized form, which allows the derivation of a simple iterative inference scheme. The approximated variational distribution is then optimised to get as close as possible to the true distribution by minimising the Kullback-Leibler divergence, or equivaently, a lower bound on the marginal likelihood, also called evidence lower bound (ELBO) [2]. The assumed form of the variational distribution is the following:

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

This factorised approximation ensures that the model scales linearly with the number of factors, the number of features, the number of views and number of samples (Fig SX).
For details on the inference and the update equations see the Appendix.

1.3 Integration of non-gaussian views

To implement efficient variational inference in conjunction with a non-Gaussian likelihood we adapt prior work from [9] using local variational bounds. The key idea is to approximate non-Gaussian data by a normally-distributed pseudo-data based on a second-order Taylor expansion. This allows to efficiently re-use the variational updates from the Gaussian case. The approximation requires the introduction of variational parameters that are adjusted alongside the updates to iteratively improve the fit. The exact form of the Gaussian pseudo-data distribution depends on the likelihood and the lower bound of gaussian form chosen on it.
Here we implemented 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 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.

1.5 Learning the number of factors

The model automatically learns the dimensionality of the latent factor space by setting factors to zero, during training, if they do not explain significant variation in any view. This is achieved by the view and factor wise ARD prior (equation 1). In practice, a threshold is required to define a factor as active or inactive. Here we define a factor as inactive, and therefore dropped from the model during training, if it explained less than 3% of variation in all views.

1.6 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 [2]. In practice, we set the default threshold for convergence as a change in ELBO smaller than 0.1.

1.7 Centering and scaling of the data

MOFA does not require the data to be centered or scaled. The first property is achieved by incorporating a constant vector of ones in the latent factor matrix, and initialising the corresponding weight vector to the true means. This ensures that the rest of the factors capture variation independent of the mean.
The second property is achived by the factor and view sparsity, which ensures different scale of the weights for each view.

1.8 Model training and robustness

A drawback of the iterative variational Bayes algorithm is that it is not guaranteed to find the optimal solution [2]. Consequently, we follow common practice [4] and ran the method multiple times under different initialisations, and we subsequently select the model with the highest ELBO. Additionally, it is recommended to check that the factors are consistently found across multiple runs.

1.9 Downstream analysis functions: pipeline for factor annotation

To do...

1.10 Implementation

What do we describe here?

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 simulation 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 dimensionality of the latent space from 5 to 60 and the percentage of missing values from 10 to 90%.
All trials were started with a sufficiently high number of factors ($k = 100$). To test the robustness under different initialisations, ten models were trained for every simulation scenario.

In most of the settings the model robustly recovers the correct number of factors. Exceptions occur when the dimensionality of the latent space is too large (more than 50 factors) (Fig S2a) or when an excessive

amount of missing values (more than 80%) is present in the data (Fig S2d). Little variability is observed across different initialisations.

2.2 Non-Gaussian likelihoods

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 or 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 (Fig S4 and Fig S5).

2.3 Disentangling sources of variation

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 were the factors were clearly set to be active or inactive in a specific view, and we assessed whether MOFA model recover 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.
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 (Fig. S7).

3 Detailed methods on CLL analysis

3.1 Data processing

The data was obtained from [**cllpaper**]. Details on the data generation and processing can be found there. For the training of MOFA we included 62 drug response measurements (excluding NSC 74859 and bortezomib due to bad quality) at five concentrations each ($p = 310$) with a threshold at 1.1 to remove outliers. Mutations were considered if present in at least 3 samples ($p = 69$). Low counts from RNAseq data were filtered out and the data was normalized using the *estimateSizeFactors* and *varianceStabilizingTransformation* function of the DESeq2 [8]. For training we used the top $p = 5000$ most variable mRNAs after exclusion of genes from the Y chromosome. Methylation data was transformed to M-values and the top 1% CpG sites excluding sex chromosomes ($p = 4248$) were included. We included patients diagnosed with CLL and having data in at least two views into the MOFA model leading to $n = 200$ 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 S8a,b). The inferred factors and weights are robust to the random initializations across 25 runs (Figure S8c,d). Also, the MOFA factors show near orthogonality, as opposed to the strongly correlated factors inferred by iCluster (Figure S9).
A single model was selected for down-stream analysis based on the highest evidence lower bound, which is highlighted in FigX.

3.3 Inspection of loadings

The first step on characterisation of factors is the direct inspection of loadings. Importantly, the scale of the weights infered 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 can be directly compared. 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 i) and the background set (the weights of genes that do not belong to gene set i). p-values are adjusted for multiple testing using FDR correction.

3.5 Downsampling analysis

To finish...

3.6 Imputation

Unobserved data points are directly imputed from the MOFA model equation

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

In case of missing data points for samples on latent factors those are points are dropped from the product. For non-gaussian views imputations are based on the corresponding link functions of the generalised linear model, i.e.

$$\mathbf{Y}^m = \frac{\exp(\mathbf{Z}\mathbf{W}^{mT})}{1 + \exp(\mathbf{Z}\mathbf{W}^{mT})}$$

$$\mathbf{Y}^m = \exp(\mathbf{Z}\mathbf{W}^{mT})$$

for Bernoulli and Poisson views, respectively, and are rounded to integer if imputations in the range of the data are wanted.
To compare imputation performance we trained MOFA on the data of samples available in all measurement ($n = 121$) and masked either single values or a full sample in the drug response view at random. The number of factors used for the task was learned by MOFA, where depending on the amount of full missing cases the focus on inferring shared factors was increased. For values missing at random factors were dropped when explaining less than 0.001 of variance in all views, for full cases missing we used 0.1 as a threshold.

3.7 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 patients, for which this information was available $n = 174$, 96 uncensored cases). For univariate associations (as shown in Figure 5b) we scaled all predictors to ensure comparability of the hazard ratios and oriented factors such that their Hazard ratio is greater or equal to 1 due to rotational invariance of the factors.
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 as well as a concetenated 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. 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*.

4 Appendix

4.1 Update equations of the gaussian model

The optimal distribution $\hat{q}_i$ for each variable $\mathbf{x}_i$, is the following:

$$\log \hat{q}_i(\mathbf{x}_i) = \mathbb{E}_{i \neq j}[\log p(\mathbf{Y}, \mathbf{X})] + \text{const.} \tag{5}$$

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 $\hat{q}_i(\mathbf{z}_i)$:

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

Latent variables

Term from the likelihood $p(\mathbf{Y}|\hat{\mathbf{W}}, \mathbf{Z}, \boldsymbol{\tau}, \mathbf{S})$:

$$\sum_{m=1}^{M} \sum_{d=1}^{D=m} \langle \tau_d^m \rangle \langle s_{dk}^m \hat{w}_{dk}^m \rangle y_{nd}^m z_{nk} - \frac{1}{2} \sum_{m=1}^{M} \sum_{d=1}^{D_m} \langle \tau_d^m \rangle \langle (s_{dk}^m \hat{w}_{dk}^m)^2 \rangle z_{nk}^2$$
$$- \frac{1}{2} \sum_{m=1}^{M} \frac{1}{2} \sum_{d=1}^{D} \langle \tau_d^m \rangle \sum_{j \neq k} (\langle s_{dj}^m \hat{w}_{dj}^m \rangle \langle z_{nj} \rangle) \langle \hat{w}_{dk}^m s_{dk}^m \rangle \langle z_{nk} \rangle + \text{const.}$$

Term from the prior $p(z_{nk})$:

$$-\frac{1}{2} z_{nk}^2 + \text{const.}$$

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

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

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

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

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

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)

$$\log\hat{q}_i(\mathbf{x}_i)=\mathbb{E}_{i\neq j}[\log p(\mathbf{Y},\mathbf{X})]+\text{const.}\tag{6}$$

Term from the prior $\log p(\alpha_k^m)$:

$$(a_0^\alpha-1)\log(\alpha_k^m)-b_0^\alpha\alpha_k^m+\text{const.}$$

Term from the prior $\log p(\mathbf{w}_{:k}^m)=\sum_{d=1}^{D_m}\log p(s_{dk}^m,\hat{w}_{dk}^m)$:

$$\frac{D_m}{2}\log(\alpha_k^m)-\frac{\alpha_k^m}{2}\sum_{d=1}^{D}\langle\hat{w}_{dk}^2\rangle+\sum_{d=1}^{D_m}\{\langle s_{dk}^m\rangle\log\theta_0+(1-\langle s_{dk}^m\rangle)\log(1-\theta_0)\}+\text{const.}$$

Writing everything together:

$$\left(a_0^\alpha+\frac{D_m}{2}-1\right)\log\alpha_k^m-\left(b_0^\alpha+\frac{1}{2}\sum_{d=1}^{D_m}\langle(\hat{w}_{d,k}^m)^2\rangle\right)\alpha_k^m+\text{const.}$$

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)

Term from the prior $p(\tau_d^m)$:

$$(a_0^\tau-1)\log\tau_d^m-b_0^\tau\tau_d^m+\text{const.}$$

Term from the likelihood $\mathcal{N}\left(y_{n,d}^m\,|\,\sum_{k=1}^K\hat{w}_{dk}^m s_{dk}^m z_{nk},\tau_d^m\right)$:

$$\frac{N}{2}\log\tau_d^m-\frac{\tau_d^m}{2}\sum_{n=1}^N\langle(y_{nd}^m-\sum_k^K w_{dk}^m s_{dk}^m z_{nk})^2\rangle+\text{const.}$$

Rewriting everything together:

$$\left(a_0^\tau-1+\frac{N}{2}\right)\log\tau_d^m-\left(b_0+\frac{1}{2}\sum_{n=1}^N\langle(y_{nd}^m-\sum_k^K\hat{w}_{dk}^m s_{dk}^m z_{nk})^2\rangle\right)\tau_d^m$$

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 θ

Unless a given factor is specifically annotated in a given view, the sparisty parameter θ_k^m of the Spike and Slab prior on $w_{k,d}^m, \forall d$ is given a Beta prior: $P(\theta_k^m) = \text{Beta}(a_0, b_0)$. The postrerior $q(\theta_k^m)$ is Beta distributed and the update of its parameters a_k^m and b_k^m are given below:

$$a_k^m = \sum_d \langle S_{k,d}^m\rangle + a_0$$

$$b_k^m = b_0 - \sum_d \langle S_{k,d}^m\rangle + D_m$$

Lower bound

Likelihood term

Vector form:

$$-\sum_{m=1}^{M}\frac{ND_m}{2}\log(2\pi) + \frac{N}{2}\sum_{m=1}^{M}\sum_{d=1}^{D_m}\log(\tau_d^m) - \frac{1}{2}\sum_{m=1}^{M}\sum_{d=1}^{D_m}\left(\mathbf{y}_d^m - \sum_{k=1}^{K}\langle s_{dk}^m\hat{w}_{dk}^m\rangle\langle\mathbf{z}_k\rangle\right)^T\left(\tau_d^m\mathbf{I}\right)\left(\mathbf{y}_d^m - \sum_{k=1}^{K}\langle s_{dk}^m\hat{w}_{dk}^m\rangle\langle\mathbf{z}_k\rangle\right)$$

Scalar form:

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

Extending terms and rearranging:

W and S terms

$p(\hat{\mathbf{W}}, \mathbf{S})$:

$$-\sum_{m=1}^{M}\frac{KD_m}{2}\log(2\pi) + \sum_{m=1}^{M}\frac{D_m}{2}\sum_{k=1}^{K}\log(\alpha_k^m) - \sum_{m=1}^{M}\frac{\alpha_k^m}{2}\sum_{d=1}^{D_m}\sum_{k=1}^{K}\langle(\hat{w}_{dk}^m)^2\rangle$$
$$+\langle\log(\theta)\rangle\sum_{m=1}^{M}\sum_{d=1}^{D_m}\sum_{k=1}^{K}\langle s_{dk}^m\rangle + \langle\log(1-\theta)\rangle\sum_{m=1}^{M}\sum_{d=1}^{D_m}\sum_{k=1}^{K}(1-\langle s_{dk}^m\rangle)$$

$q(\hat{\mathbf{W}}, \mathbf{S})$:

$$-\sum_{m=1}^{M}\frac{KD_m}{2}\log(2\pi) + \frac{1}{2}\sum_{m=1}^{M}\sum_{d=1}^{D_m}\sum_{k=1}^{K}\log(\langle s_{dk}^m\rangle\sigma_{w_{dk}^m}^2 + (1-\langle s_{dk}^m\rangle)/\alpha_k^m)$$
$$+\sum_{m=1}^{M}\sum_{d=1}^{D_m}\sum_{k=1}^{K}(1-\langle s_{dk}^m\rangle)\log(1-\langle s_{dk}^m\rangle) - \langle s_{dk}^m\rangle\log\langle s_{dk}^m\rangle$$

Z term

$$\mathbb{E}[\log P(\mathbf{Z})] = -\frac{NK}{2}\log(2\pi) - \frac{1}{2}\sum_{n=1}^{N}\langle z_{nk}^2\rangle$$

$$\mathbb{E}[\log q(\mathbf{Z})] = -\frac{NK}{2}(1+\log(2\pi)) - \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K}\log(\sigma_{z_{nk}}^2)$$

alpha term

$$\mathbb{E}[\log p(\boldsymbol{\alpha})] = \sum_{m=1}^{M}\sum_{k=1}^{K}\Big(a_0^\alpha \log b_0^\alpha + (a_0^\alpha - 1)\langle\log\alpha_k\rangle - b_0^\alpha\langle\alpha_k\rangle - \log\Gamma(a_0^\alpha)\Big)$$

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

tau

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

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

Theta

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

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

4.2 Extensions to non-gaussian likelihoods

To implement efficient variational inference in conjunction with a non-Gaussian likelihood we adapt prior work from [9] using local variational bounds. The key idea is to 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 $\boldsymbol{\Theta} = (\mathbf{Z}, \mathbf{W}, \boldsymbol{\alpha}, \mathbf{T}, \boldsymbol{\theta})$, the variational framework approximates the posterior $p(\boldsymbol{\Theta}|\mathbf{Y})$ with a distribution $q(\boldsymbol{\Theta})$, which is indirectly optimised by optimising a lower bound $\mathcal{F}$ of the log model evidence. The resulting optimization problem is given by

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

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

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

with $f_{nd} = -\log p(y_{nd}|x_{nd})$. We dropped the view index m to keep notation uncluttered.
Extending [9] to our heteroscedastic noise model, we require $f_{nd}(x_{nd})$ to be twice differentiable and bounded by κ_d, such that $f''_{nd}(x_{nd}) \leq \kappa_d \forall n, d$. 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,

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

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:

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

The algorithm propsed in [9] 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 $q(\boldsymbol{\Theta})$ 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 $f(\Delta)$ different κ_d are used resulting in different pseudo-data updates. Two special cases implemented in MOFA are the Poisson and the Bernoulli likelihood:

Bernoulli likelihood for binary data

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

$$p(y|x) = \frac{e^{yx}}{1 + e^x}$$

The second derivative of the log likelihood is bounded by:

$$f''(x) = \sigma(x)\sigma(-x) \leq 1/4 := \kappa$$

where σ is the sigmoid function $f(x) = 1/(1 + e^{-x})$.
The pseudodata updates are given by

$$\hat{y}_{nd} = \zeta_{nd} - 4 * (\sigma(\zeta_{nd}) - y_{nd})$$

A tighter bound for binary data

While the above approach of efficient variational inference for non-gaussian likelihoods works well in many settings, it is possible to further improve the approximation when using heteroscedatic gaussian pseudo-data instead of a spherical gaussian. MOFA uses this to implement a tighter bound for binary data, which frequently occurs as a view in multi-omic data sets (e.g. somatic mutations or SNPs). As before we model binary data Y as

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

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 [9] and described above which allows to recycle all the updates from the model with Gaussian views. While [9] assumes a homoscedastic approximation with a spherical Gaussian, we adopt an approach following [5], which allows for heteroscedaticity and provides a tighter bound on the Bernoulli likelihood.
Denoting $x_{ij} = (\mathbf{Z}\mathbf{W}^T)_{ij}$ the Jaakkola upper bound [5] on the negative log-likelihood is given by

$$\begin{aligned} -\log\left(p(y_{ij}|x_{ij})\right) &= -\log\left(\sigma\left((2y_{ij}-1)x_{ij}\right)\right) \\ &\leq -\log(\zeta_{ij}) - \frac{(2y_{ij}-1)x_{ij} - \zeta_{ij})}{2} + \lambda(\zeta_{ij})\left(x_{ij}^2 - \zeta_{ij}^2\right) \\ &=: b_J(\zeta_{ij}, x_{ij}, y_{ij}) \end{aligned} \tag{7}$$

with λ given by $\lambda(\zeta) = \frac{1}{4\zeta}\tanh\left(\frac{\zeta}{2}\right)$.
This can be derived easily 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 [5].
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 $\tilde{Y}$.
As above we can plug in the bound on the negative log-likelihood in the variational optimization problem to obtain

$$\min_{q(\mathbf{\Theta}),\boldsymbol{\zeta}} \sum_{d=1}^{D}\sum_{n=1}^{N} \mathbb{E}_q b_J(\zeta_{ij}, x_{ij}, y_{ij}) + \mathrm{KL}[q(\mathbf{\Theta})||p(\mathbf{\Theta})].$$

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

$$\zeta_{ij}^2 = \mathbb{E}[x_{ij}^2]$$

as described in [5], [2].
For the variational distribution $q(\mathbf{Z}, \mathbf{W})$ we observe that the Jaakkola bound can be re-written as

$$b_J(\zeta_{ij}, x_{ij}, y_{ij}) = -\log\left(\varphi\left(\tilde{y}_{ij}; x_{ij}, \frac{1}{2\lambda(\zeta_{ij})}\right)\right) + c(\zeta_{ij}),$$

where $\varphi(x; \mu, \sigma^2)$ denotes the density function of a normal distribution with mean μ and variance σ^2 and c 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

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

updating the data precision as $\tau_{ij} = 2\lambda(\zeta_{ij})$ using updates that allow for sample- and feature-wise precision parameters on the data as described in the Appendix.

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:

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

where $\lambda(x) > 0$ 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 [9], here we choose the following rate function: $\lambda(x) = \log(1 + e^x)$.
Then an upper bound of the second derivative of the log-likelihood is given by

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

The pseudodata updates are given by

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

Bibliography

Basilevsky, Alexander T (2009). *Statistical factor analysis and related methods: theory and applications*. Vol. 418. John Wiley & Sons.

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

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

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

Jaakkola, Tommi S and Michael I Jordan (2000). "Bayesian parameter estimation via variational methods". In: *Statistics and Computing* 10.1, pp. 25–37.

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

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

Love, Michael I, Wolfgang Huber, and Simon Anders (2014). "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2". In: *Genome biology* 15.12, p. 550.

Seeger, Matthias and Guillaume Bouchard (2012). "Fast variational Bayesian inference for non-conjugate matrix factorization models". In: *Artificial Intelligence and Statistics*, pp. 1012–1018.

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

Virtanen, Seppo et al. (2012). "Bayesian group factor analysis". In: *Artificial Intelligence and Statistics*, pp. 1269–1277.

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