

MOTL: enhancing multi-omics matrix factorization with transfer learning

David P. Hirst^{1*}, Morgane Térézol¹, Laura Cantini², Paul Villoutreix¹, Matthieu Vignes³ and Anaïs Baudot^{1,4,5*}

¹ Aix Marseille Univ, INSERM, MMG, Centuri, Marseille, France

² Institut Pasteur, Université Paris Cité, CNRS UMR 3738, Machine Learning for Integrative Genomics Group, Paris F-75015, France

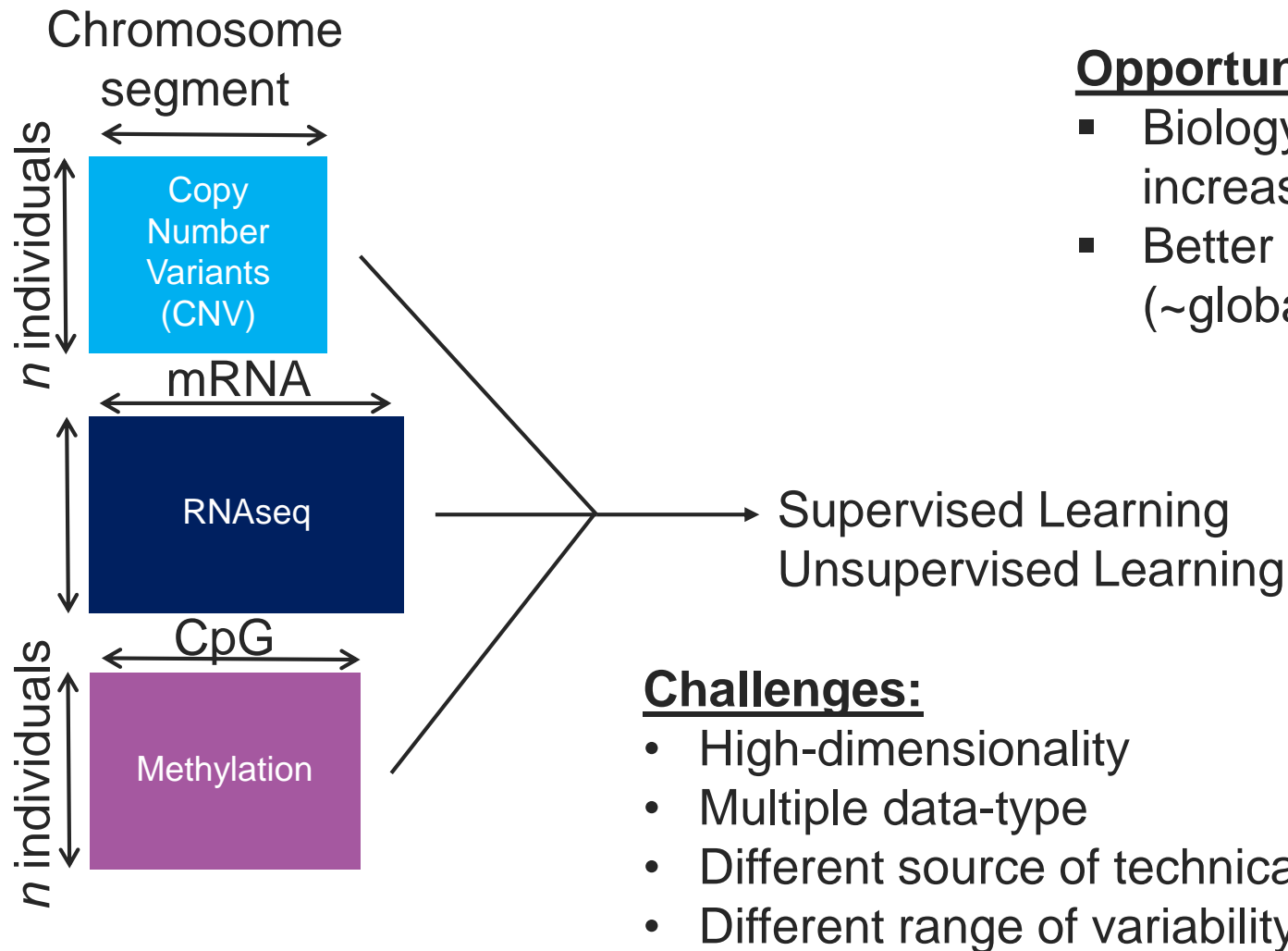
³ School of Mathematical and Computational Sciences, College of Science, Massey University, Palmerston North, New Zealand

⁴ CNRS, Marseille, France

⁵ Barcelona Supercomputing Center, Barcelona, Spain

Arnaud GLOAGUEN

Context : Multi-omics data analysis



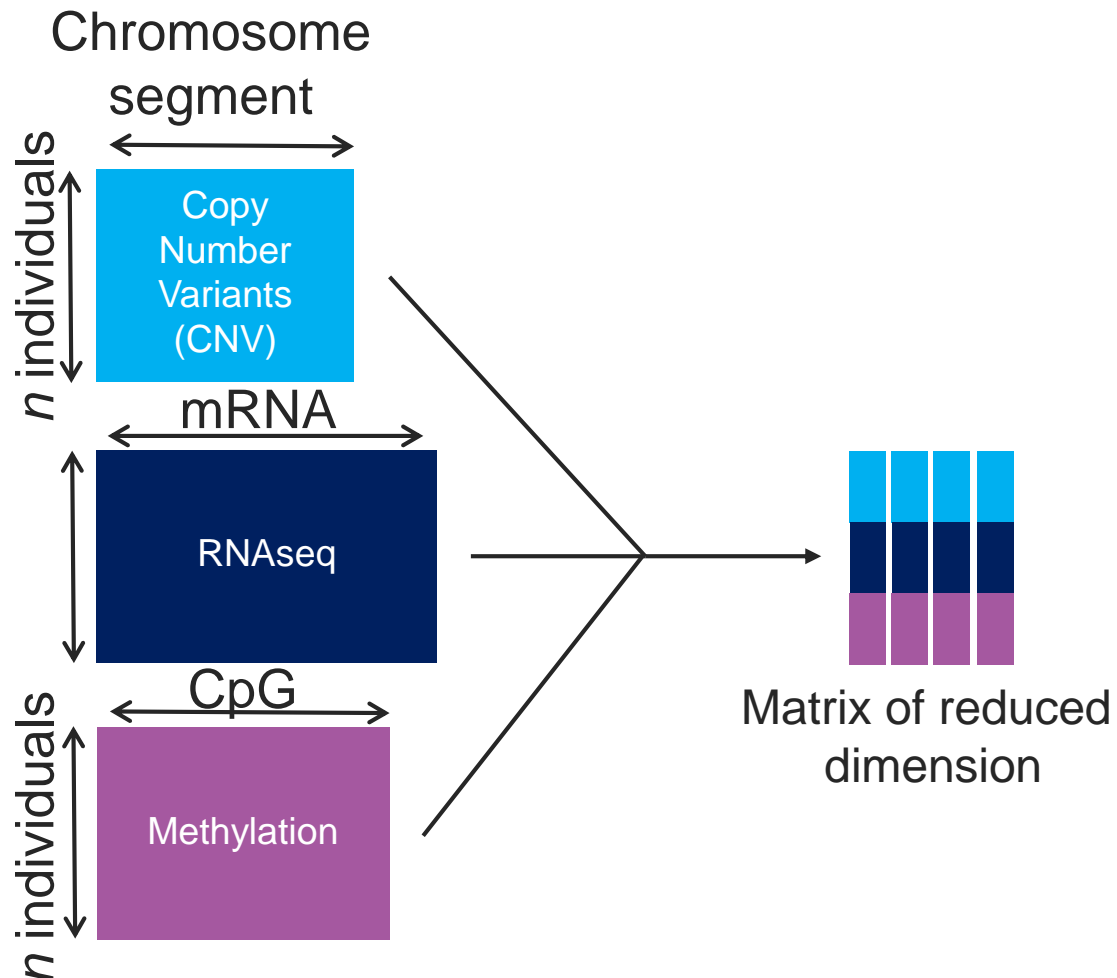
Opportunities:

- Biology and medicine revolutionized by the increased availability of multi-omics datasets
- Better understanding of a biological system (~global view / links / noise reduction)

Challenges:

- High-dimensionality
- Multiple data-type
- Different source of technical noise
- Different range of variability

Context : Multi-omics data analysis methods



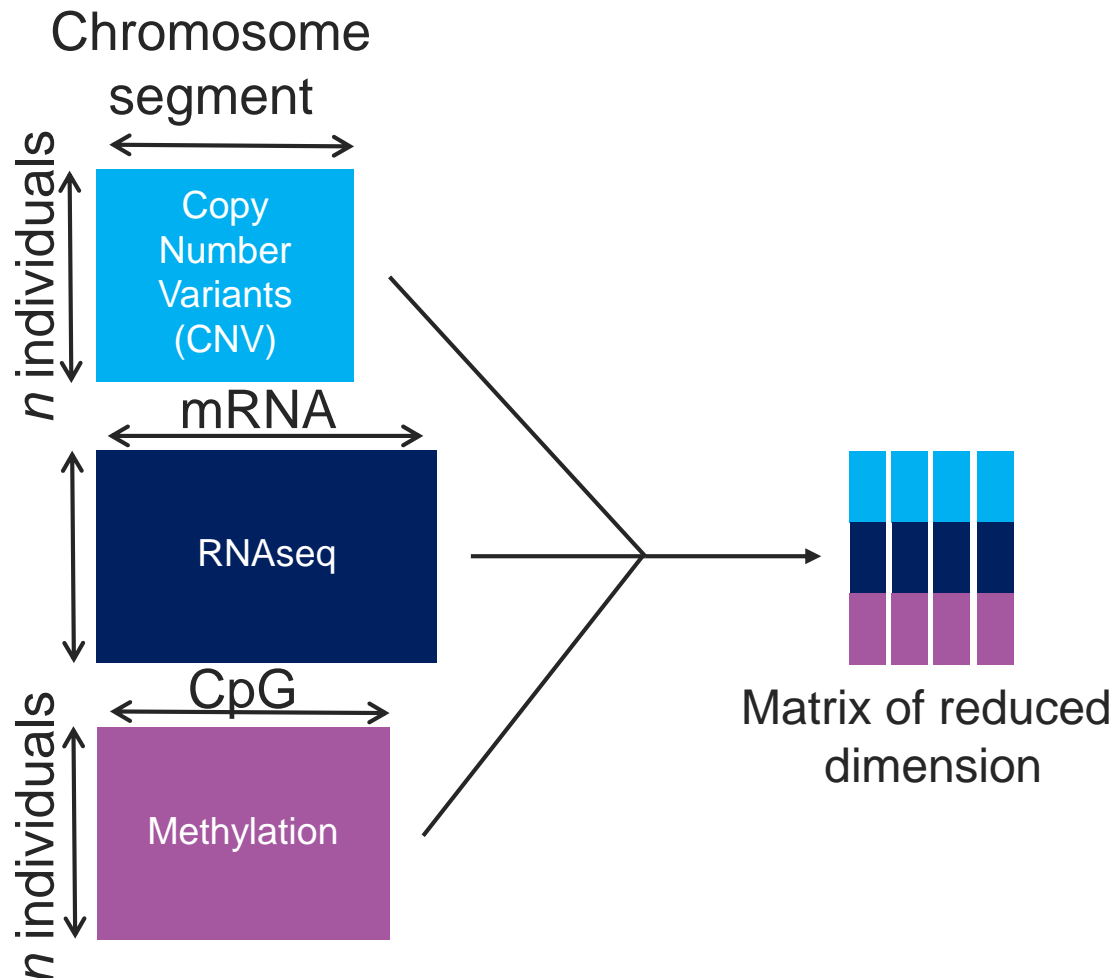
Famillies of methos:

- Bayesian
- Network
- Deep Learning
- ... jDR

Avanatges of jDR:

- Computationally efficient
- Interpretable
- Flexible (clustering / biomarkers / predict outcome)
- Effective for large data-sets

Context : Multi-omics data analysis methods



Families of methods:

- Bayesian
- Network
- Deep Learning
- ... jDR

Advantages of jDR:

- Computationally efficient
- Interpretable
- Flexible (clustering / biomarkers / predict outcome)
- Effective for large data-sets

What can we do when by essence there are too few observations ?

Transfer Learning



Transfer Learning

Source Task

Transfer Learning

Source Task



Transfer Learning

Source Task



Hours and hours of training

Transfer Learning

Source Task



Hours and hours of training

Target Task

Transfer Learning

Source Task



Hours and hours of training

Target Task



Transfer Learning

Source Task



Hours and hours of training

Positive Transfer



Target Task



Transfer Learning

Source Task



Hours and hours of training

Positive Transfer



Target Task



Level significantly better for a beginner with the same number of training hours as the other (small number of hours).

Transfer Learning

Source Task



Hours and hours of training



Hundreds and Hundreds of samples

Positive Transfer



Target Task



Level significantly better for a beginner with the same number of training hours as the other (small number of hours).

Transfer Learning

Source Task



Hours and hours of training



Hundreds and Hundreds of samples

Positive Transfer



Target Task



Level significantly better for a beginner with the same number of training hours as the other (small number of hours).



For a small number of patients on the Target domain, prediction are significantly better than training solely from the Target Domain

Transfer Learning

Source Task



Positive Transfer



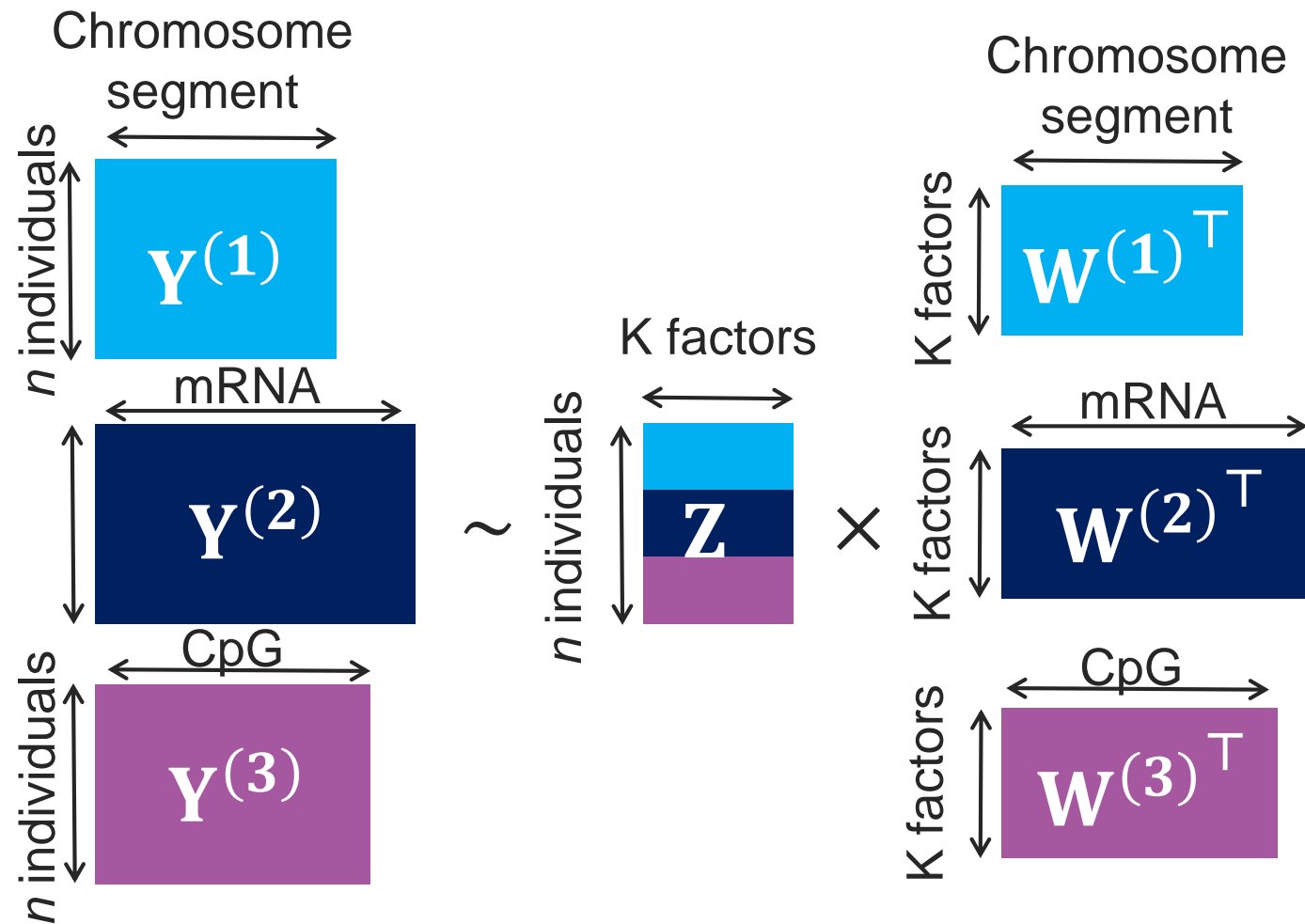
Target Task

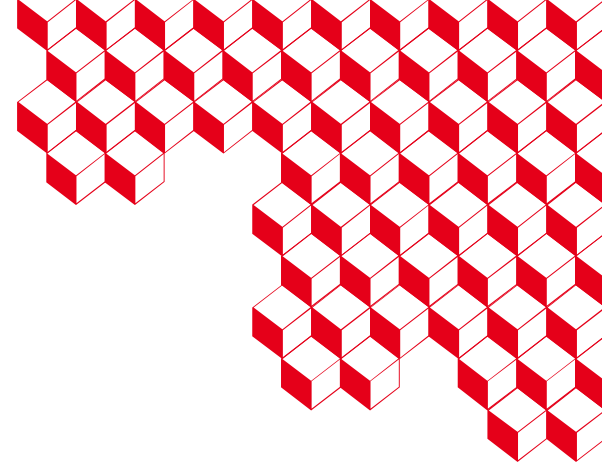


Interest (positivity) of the transfer is linked to:

- ❖ The « Proximity » between tasks.
- ❖ The number of samples in the Source/Target domain.

Multi-Omics Factor Analysis (MOFA)





Results:

MOTL: a new Transfer Learning framework for multi-omics matrix factorization.

MOTL Model



MOTL Model

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

MOTL Model

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Where :

MOTL Model

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Where :

❖ $\mathbf{t}_i^{(m)} \in \mathbb{R}^{p_m}$ corresponds to the observation of individual i on the Target data-set for omics m .

MOTL Model

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Where :

- ❖ $\mathbf{t}_i^{(m)} \in \mathbb{R}^{p_m}$ corresponds to the observation of individual i on the Target data-set for omics m .
- ❖ $\mathbf{W}^{(m)} \in \mathbb{R}^{p_m \times K}$ is the set of weight vectors learned by MOFA for omics m .

MOTL Model

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Where :

- ❖ $\mathbf{t}_i^{(m)} \in \mathbb{R}^{p_m}$ corresponds to the observation of individual i on the Target data-set for omics m .
- ❖ $\mathbf{W}^{(m)} \in \mathbb{R}^{p_m \times K}$ is the set of weight vectors learned by MOFA for omics m .
→ As opposed to MOFA, it is considered as fixed.

MOTL Model

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Where :

- ❖ $\mathbf{t}_i^{(m)} \in \mathbb{R}^{p_m}$ corresponds to the observation of individual i on the Target data-set for omics m .
- ❖ $\mathbf{W}^{(m)} \in \mathbb{R}^{p_m \times K}$ is the set of weight vectors learned by MOFA for omics m .
→ As opposed to MOFA, it is considered as fixed.
- ❖ $\mathbf{z}_i \in \mathbb{R}^K$ is a the common latent space that will be learnt by MOTL.

MOTL Model

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Where :

- ❖ $\mathbf{t}_i^{(m)} \in \mathbb{R}^{p_m}$ corresponds to the observation of individual i on the Target data-set for omics m .
- ❖ $\mathbf{W}^{(m)} \in \mathbb{R}^{p_m \times K}$ is the set of weight vectors learned by MOFA for omics m .
→ As opposed to MOFA, it is considered as fixed.
- ❖ $\mathbf{z}_i \in \mathbb{R}^K$ is a the common latent space that will be learnt by MOTL.
→ It is a random variable with the priori assumption that (same as MOFA):

$$\forall i, k \in \{1, \dots, N\} \times \{1, \dots, K\} \quad z_k^{(i)} \sim \mathcal{N}(0, 1)$$

MOTL Model

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Where :

❖ $\mathbf{t}_i^{(m)} \in \mathbb{R}^{p_m}$ corresponds to the observation of individual i on the Target data-set for omics m .

❖ $\mathbf{W}^{(m)} \in \mathbb{R}^{p_m \times K}$ is the set of weight vectors learned by MOFA for omics m .
→ As opposed to MOFA, it is considered as fixed.

❖ $\mathbf{z}_i \in \mathbb{R}^K$ is a the common latent space that will be learnt by MOTL.
→ It is a random variable with the priori assumption that (same as MOFA):

$$\forall i, k \in \{1, \dots, N\} \times \{1, \dots, K\} \quad z_k^{(i)} \sim \mathcal{N}(0, 1)$$

❖ $\boldsymbol{\epsilon}^{(m)} \in \mathbb{R}^{p_m}$ is a noise vector such that (same as MOFA):

$$\forall j, m \in \{1, \dots, p_m\} \times \{1, \dots, M\} \quad \epsilon_j^{(m)} \sim \mathcal{N}\left(0, 1/\tau_j^{(m)}\right)$$

MOTL Model

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Where :

- ❖ $\mathbf{t}_i^{(m)} \in \mathbb{R}^{p_m}$ corresponds to the observation of individual i on the Target data-set for omics m .
- ❖ $\mathbf{W}^{(m)} \in \mathbb{R}^{p_m \times K}$ is the set of weight vectors learned by MOFA for omics m .
→ As opposed to MOFA, it is considered as fixed.
- ❖ $\mathbf{z}_i \in \mathbb{R}^K$ is a the common latent space that will be learnt by MOTL.
→ It is a random variable with the priori assumption that (same as MOFA):

$$\forall i, k \in \{1, \dots, N\} \times \{1, \dots, K\} \quad z_k^{(i)} \sim \mathcal{N}(0, 1)$$

- ❖ $\boldsymbol{\epsilon}^{(m)} \in \mathbb{R}^{p_m}$ is a noise vector such that (same as MOFA):

$$\forall j, m \in \{1, \dots, p_m\} \times \{1, \dots, M\} \quad \epsilon_j^{(m)} \sim \mathcal{N}\left(0, 1/\tau_j^{(m)}\right)$$

for MOFA, $\tau_j^{(m)}$ is considered as random. For MOTL we will considered it as fixed for now.

MOTL Model

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Where :

❖ $\mathbf{t}_i^{(m)} \in \mathbb{R}^{p_m}$ corresponds to the observation of individual i on the Target data-set for omics m .

❖ $\mathbf{W}^{(m)} \in \mathbb{R}^{p_m \times K}$ is the set of weight vectors learned by MOFA for omics m .
→ As opposed to MOFA, it is considered as fixed.

❖ $\mathbf{z}_i \in \mathbb{R}^K$ is a the common latent space that will be learnt by MOTL.
→ It is a random variable with the priori assumption that (same as MOFA):

$$\forall i, k \in \{1, \dots, N\} \times \{1, \dots, K\} \quad z_k^{(i)} \sim \mathcal{N}(0, 1)$$

❖ $\boldsymbol{\epsilon}^{(m)} \in \mathbb{R}^{p_m}$ is a noise vector such that (same as MOFA):

$$\forall j, m \in \{1, \dots, p_m\} \times \{1, \dots, M\} \quad \epsilon_j^{(m)} \sim \mathcal{N}\left(0, 1/\tau_j^{(m)}\right)$$

for MOFA, $\tau_j^{(m)}$ is considered as random. For MOTL we will considered it as fixed for now.

❖ $\mathbf{a}^{(m)} \in \mathbb{R}^{p_m}$ is an intercepts non present in MOFA. It is considered as fixed and learn only on the source set.

MOTL Model – Likelihood and prior

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

MOTL Model – Likelihood and prior

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Distribution of $t_{ij}^{(m)} | \mathbf{z}_i$?

MOTL Model – Likelihood and prior

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Distribution of $t_{ij}^{(m)} | \mathbf{z}_i$?

$$t_{ij}^{(m)} =$$

MOTL Model – Likelihood and prior

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Distribution of $t_{ij}^{(m)} | \mathbf{z}_i$?

$$t_{ij}^{(m)} = a_j^{(m)} + \sum_{k=1}^K w_{jk} z_k^{(i)} + \epsilon_j^{(m)}$$

MOTL Model – Likelihood and prior

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Distribution of $t_{ij}^{(m)} | \mathbf{z}_i$?

$$t_{ij}^{(m)} = a_j^{(m)} + \sum_{k=1}^K w_{jk} z_k^{(i)} + \epsilon_j^{(m)} = a_j^{(m)} + \mathbf{w}_j \cdot \mathbf{z}_i + \epsilon_j^{(m)}$$

MOTL Model – Likelihood and prior

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Distribution of $t_{ij}^{(m)} | \mathbf{z}_i$?

$$t_{ij}^{(m)} = a_j^{(m)} + \sum_{k=1}^K w_{jk} z_k^{(i)} + \epsilon_j^{(m)} = a_j^{(m)} + \mathbf{w}_j \cdot \mathbf{z}_i + \epsilon_j^{(m)} \quad \text{and} \quad \epsilon_j^{(m)} \sim \mathcal{N}\left(0, 1/\tau_j^{(m)}\right) \text{ hence:}$$

MOTL Model – Likelihood and prior

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Distribution of $t_{ij}^{(m)} | \mathbf{z}_i$?

$$t_{ij}^{(m)} = a_j^{(m)} + \sum_{k=1}^K w_{jk} z_k^{(i)} + \epsilon_j^{(m)} = a_j^{(m)} + \mathbf{w}_{j\cdot} \mathbf{z}_i + \epsilon_j^{(m)} \quad \text{and} \quad \epsilon_j^{(m)} \sim \mathcal{N}\left(0, 1/\tau_j^{(m)}\right) \text{ hence:}$$

$$t_{ij}^{(m)} | \mathbf{z}_i \sim \mathcal{N}\left(a_j^{(m)} + \mathbf{w}_{j\cdot} \mathbf{z}_i, 1/\tau_j^{(m)}\right)$$

MOTL Model – Likelihood and prior

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Distribution of $t_{ij}^{(m)} | \mathbf{z}_i$?

$$t_{ij}^{(m)} = a_j^{(m)} + \sum_{k=1}^K w_{jk} z_k^{(i)} + \epsilon_j^{(m)} = a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i + \epsilon_j^{(m)} \quad \text{and} \quad \epsilon_j^{(m)} \sim \mathcal{N}(0, 1/\tau_j^{(m)}) \quad \text{hence:}$$

$$t_{ij}^{(m)} | \mathbf{z}_i \sim \mathcal{N}(a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i, 1/\tau_j^{(m)})$$

We can thus write the likelihood $p(\mathbf{t}_1^{(1)}, \dots, \mathbf{t}_N^{(1)}, \dots, \mathbf{t}_1^{(M)}, \dots, \mathbf{t}_N^{(M)} | \mathbf{z}_1, \dots, \mathbf{z}_N) = p(\mathbf{T} | \mathbf{Z})$,

MOTL Model – Likelihood and prior

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Distribution of $t_{ij}^{(m)} | \mathbf{z}_i$?

$$t_{ij}^{(m)} = a_j^{(m)} + \sum_{k=1}^K w_{jk} z_k^{(i)} + \epsilon_j^{(m)} = a_j^{(m)} + \mathbf{w}_{j\cdot} \mathbf{z}_i + \epsilon_j^{(m)} \quad \text{and} \quad \epsilon_j^{(m)} \sim \mathcal{N}(0, 1/\tau_j^{(m)}) \text{ hence:}$$

$$t_{ij}^{(m)} | \mathbf{z}_i \sim \mathcal{N}(a_j^{(m)} + \mathbf{w}_{j\cdot} \mathbf{z}_i, 1/\tau_j^{(m)})$$

We can thus write the likelihood $p(\mathbf{t}_1^{(1)}, \dots, \mathbf{t}_N^{(1)}, \dots, \mathbf{t}_1^{(M)}, \dots, \mathbf{t}_N^{(M)} | \mathbf{z}_1, \dots, \mathbf{z}_N) = p(\mathbf{T} | \mathbf{Z})$,

$$p(\mathbf{T} | \mathbf{Z}) =$$

MOTL Model – Likelihood and prior

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Distribution of $t_{ij}^{(m)} | \mathbf{z}_i$?

$$t_{ij}^{(m)} = a_j^{(m)} + \sum_{k=1}^K w_{jk} z_k^{(i)} + \epsilon_j^{(m)} = a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i + \epsilon_j^{(m)} \quad \text{and} \quad \epsilon_j^{(m)} \sim \mathcal{N}(0, 1/\tau_j^{(m)}) \quad \text{hence:}$$

$$t_{ij}^{(m)} | \mathbf{z}_i \sim \mathcal{N}(a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i, 1/\tau_j^{(m)})$$

We can thus write the likelihood $p(\mathbf{t}_1^{(1)}, \dots, \mathbf{t}_N^{(1)}, \dots, \mathbf{t}_1^{(M)}, \dots, \mathbf{t}_N^{(M)} | \mathbf{z}_1, \dots, \mathbf{z}_N) = p(\mathbf{T} | \mathbf{Z})$,

$$p(\mathbf{T} | \mathbf{Z}) = \prod_{m=1}^M \prod_{i=1}^N p(\mathbf{t}_i^{(m)} | \mathbf{z}_i)$$

MOTL Model – Likelihood and prior

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Distribution of $t_{ij}^{(m)} | \mathbf{z}_i$?

$$t_{ij}^{(m)} = a_j^{(m)} + \sum_{k=1}^K w_{jk} z_k^{(i)} + \epsilon_j^{(m)} = a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i + \epsilon_j^{(m)} \quad \text{and} \quad \epsilon_j^{(m)} \sim \mathcal{N}(0, 1/\tau_j^{(m)}) \quad \text{hence:}$$

$$t_{ij}^{(m)} | \mathbf{z}_i \sim \mathcal{N}(a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i, 1/\tau_j^{(m)})$$

We can thus write the likelihood $p(\mathbf{t}_1^{(1)}, \dots, \mathbf{t}_N^{(1)}, \dots, \mathbf{t}_1^{(M)}, \dots, \mathbf{t}_N^{(M)} | \mathbf{z}_1, \dots, \mathbf{z}_N) = p(\mathbf{T} | \mathbf{Z})$,

$$p(\mathbf{T} | \mathbf{Z}) = \prod_{m=1}^M \prod_{i=1}^N p(\mathbf{t}_i^{(m)} | \mathbf{z}_i) = \prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} p(t_{ij}^{(m)} | \mathbf{z}_i)$$

MOTL Model – Likelihood and prior

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Distribution of $t_{ij}^{(m)} | \mathbf{z}_i$?

$$t_{ij}^{(m)} = a_j^{(m)} + \sum_{k=1}^K w_{jk} z_k^{(i)} + \epsilon_j^{(m)} = a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i + \epsilon_j^{(m)} \quad \text{and} \quad \epsilon_j^{(m)} \sim \mathcal{N}(0, 1/\tau_j^{(m)}) \quad \text{hence:}$$

$$t_{ij}^{(m)} | \mathbf{z}_i \sim \mathcal{N}(a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i, 1/\tau_j^{(m)})$$

We can thus write the likelihood $p(\mathbf{t}_1^{(1)}, \dots, \mathbf{t}_N^{(1)}, \dots, \mathbf{t}_1^{(M)}, \dots, \mathbf{t}_N^{(M)} | \mathbf{z}_1, \dots, \mathbf{z}_N) = p(\mathbf{T} | \mathbf{Z})$,

$$p(\mathbf{T} | \mathbf{Z}) = \prod_{m=1}^M \prod_{i=1}^N p(\mathbf{t}_i^{(m)} | \mathbf{z}_i) = \prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} p(t_{ij}^{(m)} | \mathbf{z}_i) = \prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N}(a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i, 1/\tau_j^{(m)})$$

MOTL Model – Likelihood and prior

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Distribution of $t_{ij}^{(m)} | \mathbf{z}_i$?

$$t_{ij}^{(m)} = a_j^{(m)} + \sum_{k=1}^K w_{jk} z_k^{(i)} + \epsilon_j^{(m)} = a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i + \epsilon_j^{(m)} \quad \text{and} \quad \epsilon_j^{(m)} \sim \mathcal{N}(0, 1/\tau_j^{(m)}) \quad \text{hence:}$$

$$t_{ij}^{(m)} | \mathbf{z}_i \sim \mathcal{N}(a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i, 1/\tau_j^{(m)})$$

We can thus write the likelihood $p(\mathbf{t}_1^{(1)}, \dots, \mathbf{t}_N^{(1)}, \dots, \mathbf{t}_1^{(M)}, \dots, \mathbf{t}_N^{(M)} | \mathbf{z}_1, \dots, \mathbf{z}_N) = p(\mathbf{T} | \mathbf{Z})$,

$$p(\mathbf{T} | \mathbf{Z}) = \prod_{m=1}^M \prod_{i=1}^N p(\mathbf{t}_i^{(m)} | \mathbf{z}_i) = \prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} p(t_{ij}^{(m)} | \mathbf{z}_i) = \prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N}(a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i, 1/\tau_j^{(m)})$$

Furthermore

MOTL Model – Likelihood and prior

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Distribution of $t_{ij}^{(m)} | \mathbf{z}_i$?

$$t_{ij}^{(m)} = a_j^{(m)} + \sum_{k=1}^K w_{jk} z_k^{(i)} + \epsilon_j^{(m)} = a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i + \epsilon_j^{(m)} \quad \text{and} \quad \epsilon_j^{(m)} \sim \mathcal{N}(0, 1/\tau_j^{(m)}) \quad \text{hence:}$$

$$t_{ij}^{(m)} | \mathbf{z}_i \sim \mathcal{N}(a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i, 1/\tau_j^{(m)})$$

We can thus write the likelihood $p(\mathbf{t}_1^{(1)}, \dots, \mathbf{t}_N^{(1)}, \dots, \mathbf{t}_1^{(M)}, \dots, \mathbf{t}_N^{(M)} | \mathbf{z}_1, \dots, \mathbf{z}_N) = p(\mathbf{T} | \mathbf{Z})$,

$$p(\mathbf{T} | \mathbf{Z}) = \prod_{m=1}^M \prod_{i=1}^N p(\mathbf{t}_i^{(m)} | \mathbf{z}_i) = \prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} p(t_{ij}^{(m)} | \mathbf{z}_i) = \prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N}(a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i, 1/\tau_j^{(m)})$$

Furthermore

$$p(\mathbf{Z}) =$$

MOTL Model – Likelihood and prior

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Distribution of $t_{ij}^{(m)} | \mathbf{z}_i$?

$$t_{ij}^{(m)} = a_j^{(m)} + \sum_{k=1}^K w_{jk} z_k^{(i)} + \epsilon_j^{(m)} = a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i + \epsilon_j^{(m)} \quad \text{and} \quad \epsilon_j^{(m)} \sim \mathcal{N}(0, 1/\tau_j^{(m)}) \quad \text{hence:}$$

$$t_{ij}^{(m)} | \mathbf{z}_i \sim \mathcal{N}(a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i, 1/\tau_j^{(m)})$$

We can thus write the likelihood $p(\mathbf{t}_1^{(1)}, \dots, \mathbf{t}_N^{(1)}, \dots, \mathbf{t}_1^{(M)}, \dots, \mathbf{t}_N^{(M)} | \mathbf{z}_1, \dots, \mathbf{z}_N) = p(\mathbf{T} | \mathbf{Z})$,

$$p(\mathbf{T} | \mathbf{Z}) = \prod_{m=1}^M \prod_{i=1}^N p(\mathbf{t}_i^{(m)} | \mathbf{z}_i) = \prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} p(t_{ij}^{(m)} | \mathbf{z}_i) = \prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N}(a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i, 1/\tau_j^{(m)})$$

Furthermore

$$p(\mathbf{Z}) = \prod_{i=1}^N p(\mathbf{z}_i)$$

MOTL Model – Likelihood and prior

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Distribution of $t_{ij}^{(m)} | \mathbf{z}_i$?

$$t_{ij}^{(m)} = a_j^{(m)} + \sum_{k=1}^K w_{jk} z_k^{(i)} + \epsilon_j^{(m)} = a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i + \epsilon_j^{(m)} \quad \text{and} \quad \epsilon_j^{(m)} \sim \mathcal{N}(0, 1/\tau_j^{(m)}) \quad \text{hence:}$$

$$t_{ij}^{(m)} | \mathbf{z}_i \sim \mathcal{N}(a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i, 1/\tau_j^{(m)})$$

We can thus write the likelihood $p(\mathbf{t}_1^{(1)}, \dots, \mathbf{t}_N^{(1)}, \dots, \mathbf{t}_1^{(M)}, \dots, \mathbf{t}_N^{(M)} | \mathbf{z}_1, \dots, \mathbf{z}_N) = p(\mathbf{T} | \mathbf{Z})$,

$$p(\mathbf{T} | \mathbf{Z}) = \prod_{m=1}^M \prod_{i=1}^N p(\mathbf{t}_i^{(m)} | \mathbf{z}_i) = \prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} p(t_{ij}^{(m)} | \mathbf{z}_i) = \prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N}(a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i, 1/\tau_j^{(m)})$$

Furthermore

$$p(\mathbf{Z}) = \prod_{i=1}^N p(\mathbf{z}_i) = \prod_{i=1}^N \prod_{k=1}^K p(z_k^{(i)})$$

MOTL Model – Likelihood and prior

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

Distribution of $t_{ij}^{(m)} | \mathbf{z}_i$?

$$t_{ij}^{(m)} = a_j^{(m)} + \sum_{k=1}^K w_{jk} z_k^{(i)} + \epsilon_j^{(m)} = a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i + \epsilon_j^{(m)} \quad \text{and} \quad \epsilon_j^{(m)} \sim \mathcal{N}(0, 1/\tau_j^{(m)}) \quad \text{hence:}$$

$$t_{ij}^{(m)} | \mathbf{z}_i \sim \mathcal{N}(a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i, 1/\tau_j^{(m)})$$

We can thus write the likelihood $p(\mathbf{t}_1^{(1)}, \dots, \mathbf{t}_N^{(1)}, \dots, \mathbf{t}_1^{(M)}, \dots, \mathbf{t}_N^{(M)} | \mathbf{z}_1, \dots, \mathbf{z}_N) = p(\mathbf{T} | \mathbf{Z})$,

$$p(\mathbf{T} | \mathbf{Z}) = \prod_{m=1}^M \prod_{i=1}^N p(\mathbf{t}_i^{(m)} | \mathbf{z}_i) = \prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} p(t_{ij}^{(m)} | \mathbf{z}_i) = \prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N}(a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i, 1/\tau_j^{(m)})$$

Furthermore

$$p(\mathbf{Z}) = \prod_{i=1}^N p(\mathbf{z}_i) = \prod_{i=1}^N \prod_{k=1}^K p(z_k^{(i)}) = \prod_{i=1}^N \prod_{k=1}^K \mathcal{N}(0, 1)$$

MOTL Model – Joint probability

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

MOTL Model – Joint probability

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

We can thus write the joint probability $p(\mathbf{T}, \mathbf{Z})$:

MOTL Model – Joint probability

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

We can thus write the joint probability $p(\mathbf{T}, \mathbf{Z})$:

$$p(\mathbf{T}, \mathbf{Z})$$

MOTL Model – Joint probability

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

We can thus write the joint probability $p(\mathbf{T}, \mathbf{Z})$:

$$p(\mathbf{T}, \mathbf{Z}) = p(\mathbf{T} | \mathbf{Z}) p(\mathbf{Z})$$

MOTL Model – Joint probability

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

We can thus write the joint probability $p(\mathbf{T}, \mathbf{Z})$:

$$p(\mathbf{T}, \mathbf{Z}) = p(\mathbf{T} | \mathbf{Z}) p(\mathbf{Z}) = \prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N} \left(a_j^{(m)} + w_{j:z_i^{(i)}}, 1/\tau_j^{(m)} \right) \prod_{i=1}^N \prod_{k=1}^K \mathcal{N}(0, 1)$$

MOTL Model – Joint probability

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

We can thus write the joint probability $p(\mathbf{T}, \mathbf{Z})$:

$$p(\mathbf{T}, \mathbf{Z}) = p(\mathbf{T} | \mathbf{Z}) p(\mathbf{Z}) = \prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N} \left(a_j^{(m)} + w_{j:z_i^{(i)}}, 1/\tau_j^{(m)} \right) \prod_{i=1}^N \prod_{k=1}^K \mathcal{N}(0, 1)$$

So much products... and what now ???

MOTL Model – Joint probability

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

We can thus write the joint probability $p(\mathbf{T}, \mathbf{Z})$:

$$p(\mathbf{T}, \mathbf{Z}) = p(\mathbf{T} | \mathbf{Z}) p(\mathbf{Z}) = \prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N} \left(a_j^{(m)} + w_{j:z_i^{(i)}}, 1/\tau_j^{(m)} \right) \prod_{i=1}^N \prod_{k=1}^K \mathcal{N}(0, 1)$$

So much products... and what now ???

Amongst the possibility to estimate $\hat{\mathbf{Z}}$, Maximum A Posteriori (MAP) consists in maximizing the a posteriori distribution, meaning

MOTL Model – Joint probability

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

We can thus write the joint probability $p(\mathbf{T}, \mathbf{Z})$:

$$p(\mathbf{T}, \mathbf{Z}) = p(\mathbf{T} | \mathbf{Z}) p(\mathbf{Z}) = \prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N} \left(a_j^{(m)} + w_{j:z_i^{(i)}}, 1/\tau_j^{(m)} \right) \prod_{i=1}^N \prod_{k=1}^K \mathcal{N}(0, 1)$$

So much products... and what now ???

Amongst the possibility to estimate $\hat{\mathbf{Z}}$, Maximum A Posteriori (MAP) consists in maximizing the a posteriori distribution, meaning

$$\hat{\mathbf{Z}} = \underset{\mathbf{Z}}{\operatorname{argmax}} p(\mathbf{Z} | \mathbf{T})$$

MOTL Model – Joint probability

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

We can thus write the joint probability $p(\mathbf{T}, \mathbf{Z})$:

$$p(\mathbf{T}, \mathbf{Z}) = p(\mathbf{T} | \mathbf{Z}) p(\mathbf{Z}) = \prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N} \left(a_j^{(m)} + w_{j:z_i}^{(i)}, 1/\tau_j^{(m)} \right) \prod_{i=1}^N \prod_{k=1}^K \mathcal{N}(0, 1)$$

So much products... and what now ???

Amongst the possibility to estimate $\hat{\mathbf{Z}}$, Maximum A Posteriori (MAP) consists in maximizing the a posteriori distribution, meaning

$$\hat{\mathbf{Z}} = \underset{\mathbf{Z}}{\operatorname{argmax}} p(\mathbf{Z} | \mathbf{T}) = \underset{\mathbf{Z}}{\operatorname{argmax}} p(\mathbf{T} | \mathbf{Z}) p(\mathbf{Z})$$

MOTL Model – Joint probability

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

We can thus write the joint probability $p(\mathbf{T}, \mathbf{Z})$:

$$p(\mathbf{T}, \mathbf{Z}) = p(\mathbf{T} | \mathbf{Z}) p(\mathbf{Z}) = \prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N} \left(a_j^{(m)} + w_{j:z_i}^{(i)}, 1/\tau_j^{(m)} \right) \prod_{i=1}^N \prod_{k=1}^K \mathcal{N}(0, 1)$$

So much products... and what now ???

Amongst the possibility to estimate $\hat{\mathbf{Z}}$, Maximum A Posteriori (MAP) consists in maximizing the a posteriori distribution, meaning

$$\hat{\mathbf{Z}} = \underset{\mathbf{Z}}{\operatorname{argmax}} p(\mathbf{Z} | \mathbf{T}) = \underset{\mathbf{Z}}{\operatorname{argmax}} p(\mathbf{T} | \mathbf{Z}) p(\mathbf{Z}) = \underset{\mathbf{Z}}{\operatorname{argmax}} p(\mathbf{T}, \mathbf{Z})$$

MOTL Model – Joint probability

$$\forall m, i \in \{1, \dots, M\} \times \{1, \dots, N\} \quad \mathbf{t}_i^{(m)} = \mathbf{a}^{(m)} + \mathbf{W}^{(m)} \mathbf{z}_i + \boldsymbol{\epsilon}^{(m)}$$

We can thus write the joint probability $p(\mathbf{T}, \mathbf{Z})$:

$$p(\mathbf{T}, \mathbf{Z}) = p(\mathbf{T} | \mathbf{Z}) p(\mathbf{Z}) = \prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N} \left(a_j^{(m)} + w_{j:z_i}^{(i)}, 1/\tau_j^{(m)} \right) \prod_{i=1}^N \prod_{k=1}^K \mathcal{N}(0, 1)$$

So much products... and what now ???

Amongst the possibility to estimate $\hat{\mathbf{Z}}$, Maximum A Posteriori (MAP) consists in maximizing the a posteriori distribution, meaning

$$\hat{\mathbf{Z}} = \underset{\mathbf{Z}}{\operatorname{argmax}} p(\mathbf{Z} | \mathbf{T}) = \underset{\mathbf{Z}}{\operatorname{argmax}} p(\mathbf{T} | \mathbf{Z}) p(\mathbf{Z}) = \underset{\mathbf{Z}}{\operatorname{argmax}} p(\mathbf{T}, \mathbf{Z})$$

Though it can be quite hard to maximize such posterior distribution.

Bayesian Variational Inference

The idea behind variational inference is to replace $p(\mathbf{Z}|\mathbf{T})$ but a much simpler distribution to maximize, $q(\mathbf{Z})$ called the variational distribution. This distribution can be estimated by minimizing the Kullback-Leibler divergence between the two distributions:

Bayesian Variational Inference

The idea behind variational inference is to replace $p(\mathbf{Z}|\mathbf{T})$ but a much simpler distribution to maximize, $q(\mathbf{Z})$ called the variational distribution. This distribution can be estimated by minimizing the Kullback-Leibler divergence between the two distributions:

$$KL(q||p) = \int q(\mathbf{Z}) \log \left(\frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{T})} \right) d\mathbf{Z}$$

Bayesian Variational Inference

The idea behind variational inference is to replace $p(\mathbf{Z}|\mathbf{T})$ but a much simpler distribution to maximize, $q(\mathbf{Z})$ called the variational distribution. This distribution can be estimated by minimizing the Kullback-Leibler divergence between the two distributions:

$$KL(q||p) = \int q(\mathbf{Z}) \log \left(\frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{T})} \right) d\mathbf{Z}$$

If all possible “ q ” are available, minimizing this divergence should lead to $p(\mathbf{Z}|\mathbf{T})$. However here, this idea is to restrict the available space of “ q ” so that it is easy to compute but still close to $p(\mathbf{Z}|\mathbf{T})$.

Bayesian Variational Inference

The idea behind variational inference is to replace $p(\mathbf{Z}|\mathbf{T})$ but a much simpler distribution to maximize, $q(\mathbf{Z})$ called the variational distribution. This distribution can be estimated by minimizing the Kullback-Leibler divergence between the two distributions:

$$KL(q||p) = \int q(\mathbf{Z}) \log \left(\frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{T})} \right) d\mathbf{Z}$$

If all possible “ q ” are available, minimizing this divergence should lead to $p(\mathbf{Z}|\mathbf{T})$. However here, this idea is to restrict the available space of “ q ” so that it is easy to compute but still close to $p(\mathbf{Z}|\mathbf{T})$.

Though, the KL divergence is already difficult to deal with. Indeed,

Bayesian Variational Inference

The idea behind variational inference is to replace $p(\mathbf{Z}|\mathbf{T})$ but a much simpler distribution to maximize, $q(\mathbf{Z})$ called the variational distribution. This distribution can be estimated by minimizing the Kullback-Leibler divergence between the two distributions:

$$KL(q||p) = \int q(\mathbf{Z}) \log \left(\frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{T})} \right) d\mathbf{Z}$$

If all possible “ q ” are available, minimizing this divergence should lead to $p(\mathbf{Z}|\mathbf{T})$. However here, this idea is to restrict the available space of “ q ” so that it is easy to compute but still close to $p(\mathbf{Z}|\mathbf{T})$.

Though, the KL divergence is already difficult to dealt with. Indeed,

$$KL(q||p) = \int q(\mathbf{Z}) \log \left(\frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{T})} \right) d\mathbf{Z}$$

Bayesian Variational Inference

The idea behind variational inference is to replace $p(\mathbf{Z}|\mathbf{T})$ but a much simpler distribution to maximize, $q(\mathbf{Z})$ called the variational distribution. This distribution can be estimated by minimizing the Kullback-Leibler divergence between the two distributions:

$$KL(q||p) = \int q(\mathbf{Z}) \log \left(\frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{T})} \right) d\mathbf{Z}$$

If all possible “ q ” are available, minimizing this divergence should lead to $p(\mathbf{Z}|\mathbf{T})$. However here, this idea is to restrict the available space of “ q ” so that it is easy to compute but still close to $p(\mathbf{Z}|\mathbf{T})$.

Though, the KL divergence is already difficult to dealt with. Indeed,

$$KL(q||p) = \int q(\mathbf{Z}) \log \left(\frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{T})} \right) d\mathbf{Z} = \int q(\mathbf{Z}) [\log(q(\mathbf{Z})) - \log(p(\mathbf{Z}|\mathbf{T}))] d\mathbf{Z}$$

Bayesian Variational Inference

The idea behind variational inference is to replace $p(\mathbf{Z}|\mathbf{T})$ but a much simpler distribution to maximize, $q(\mathbf{Z})$ called the variational distribution. This distribution can be estimated by minimizing the Kullback-Leibler divergence between the two distributions:

$$KL(q||p) = \int q(\mathbf{Z}) \log \left(\frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{T})} \right) d\mathbf{Z}$$

If all possible “ q ” are available, minimizing this divergence should lead to $p(\mathbf{Z}|\mathbf{T})$. However here, this idea is to restrict the available space of “ q ” so that it is easy to compute but still close to $p(\mathbf{Z}|\mathbf{T})$.

Though, the KL divergence is already difficult to dealt with. Indeed,

$$\begin{aligned} KL(q||p) &= \int q(\mathbf{Z}) \log \left(\frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{T})} \right) d\mathbf{Z} = \int q(\mathbf{Z}) [\log(q(\mathbf{Z})) - \log(p(\mathbf{Z}|\mathbf{T}))] d\mathbf{Z} \\ &= \int q(\mathbf{Z}) \log(q(\mathbf{Z})) d\mathbf{Z} - \int q(\mathbf{Z}) \log(p(\mathbf{Z}|\mathbf{T})) d\mathbf{Z} \end{aligned}$$

Bayesian Variational Inference

The idea behind variational inference is to replace $p(\mathbf{Z}|\mathbf{T})$ but a much simpler distribution to maximize, $q(\mathbf{Z})$ called the variational distribution. This distribution can be estimated by minimizing the Kullback-Leibler divergence between the two distributions:

$$KL(q||p) = \int q(\mathbf{Z}) \log \left(\frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{T})} \right) d\mathbf{Z}$$

If all possible “ q ” are available, minimizing this divergence should lead to $p(\mathbf{Z}|\mathbf{T})$. However here, this idea is to restrict the available space of “ q ” so that it is easy to compute but still close to $p(\mathbf{Z}|\mathbf{T})$.

Though, the KL divergence is already difficult to dealt with. Indeed,

$$\begin{aligned} KL(q||p) &= \int q(\mathbf{Z}) \log \left(\frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{T})} \right) d\mathbf{Z} = \int q(\mathbf{Z}) [\log(q(\mathbf{Z})) - \log(p(\mathbf{Z}|\mathbf{T}))] d\mathbf{Z} \\ &= \int q(\mathbf{Z}) \log(q(\mathbf{Z})) d\mathbf{Z} - \int q(\mathbf{Z}) \log(p(\mathbf{Z}|\mathbf{T})) d\mathbf{Z} = \mathbb{E}_q[\log(q(\mathbf{Z}))] - \mathbb{E}_q[\log(p(\mathbf{Z}|\mathbf{T}))] \end{aligned}$$

Bayesian Variational Inference

The idea behind variational inference is to replace $p(\mathbf{Z}|\mathbf{T})$ but a much simpler distribution to maximize, $q(\mathbf{Z})$ called the variational distribution. This distribution can be estimated by minimizing the Kullback-Leibler divergence between the two distributions:

$$KL(q||p) = \int q(\mathbf{Z}) \log \left(\frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{T})} \right) d\mathbf{Z}$$

If all possible “ q ” are available, minimizing this divergence should lead to $p(\mathbf{Z}|\mathbf{T})$. However here, this idea is to restrict the available space of “ q ” so that it is easy to compute but still close to $p(\mathbf{Z}|\mathbf{T})$.

Though, the KL divergence is already difficult to dealt with. Indeed,

$$\begin{aligned} KL(q||p) &= \int q(\mathbf{Z}) \log \left(\frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{T})} \right) d\mathbf{Z} = \int q(\mathbf{Z}) [\log(q(\mathbf{Z})) - \log(p(\mathbf{Z}|\mathbf{T}))] d\mathbf{Z} \\ &= \int q(\mathbf{Z}) \log(q(\mathbf{Z})) d\mathbf{Z} - \int q(\mathbf{Z}) \log(p(\mathbf{Z}|\mathbf{T})) d\mathbf{Z} = \mathbb{E}_q[\log(q(\mathbf{Z}))] - \mathbb{E}_q[\log(p(\mathbf{Z}|\mathbf{T}))] \\ &= \mathbb{E}_q[\log(q(\mathbf{Z}))] - \mathbb{E}_q \left[\log \left(\frac{p(\mathbf{Z}, \mathbf{T})}{p(\mathbf{T})} \right) \right] \end{aligned}$$

Bayesian Variational Inference

The idea behind variational inference is to replace $p(\mathbf{Z}|\mathbf{T})$ but a much simpler distribution to maximize, $q(\mathbf{Z})$ called the variational distribution. This distribution can be estimated by minimizing the Kullback-Leibler divergence between the two distributions:

$$KL(q||p) = \int q(\mathbf{Z}) \log \left(\frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{T})} \right) d\mathbf{Z}$$

If all possible “ q ” are available, minimizing this divergence should lead to $p(\mathbf{Z}|\mathbf{T})$. However here, this idea is to restrict the available space of “ q ” so that it is easy to compute but still close to $p(\mathbf{Z}|\mathbf{T})$.

Though, the KL divergence is already difficult to dealt with. Indeed,

$$\begin{aligned} KL(q||p) &= \int q(\mathbf{Z}) \log \left(\frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{T})} \right) d\mathbf{Z} = \int q(\mathbf{Z}) [\log(q(\mathbf{Z})) - \log(p(\mathbf{Z}|\mathbf{T}))] d\mathbf{Z} \\ &= \int q(\mathbf{Z}) \log(q(\mathbf{Z})) d\mathbf{Z} - \int q(\mathbf{Z}) \log(p(\mathbf{Z}|\mathbf{T})) d\mathbf{Z} = \mathbb{E}_q[\log(q(\mathbf{Z}))] - \mathbb{E}_q[\log(p(\mathbf{Z}|\mathbf{T}))] \\ &= \mathbb{E}_q[\log(q(\mathbf{Z}))] - \mathbb{E}_q \left[\log \left(\frac{p(\mathbf{Z}, \mathbf{T})}{p(\mathbf{T})} \right) \right] = \mathbb{E}_q[\log(p(\mathbf{T}))] - \mathbb{E}_q[\log(p(\mathbf{Z}, \mathbf{T})) - \log(q(\mathbf{Z}))] \end{aligned}$$

Bayesian Variational Inference

The idea behind variational inference is to replace $p(\mathbf{Z}|\mathbf{T})$ but a much simpler distribution to maximize, $q(\mathbf{Z})$ called the variational distribution. This distribution can be estimated by minimizing the Kullback-Leibler divergence between the two distributions:

$$KL(q||p) = \int q(\mathbf{Z}) \log \left(\frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{T})} \right) d\mathbf{Z}$$

If all possible “ q ” are available, minimizing this divergence should lead to $p(\mathbf{Z}|\mathbf{T})$. However here, this idea is to restrict the available space of “ q ” so that it is easy to compute but still close to $p(\mathbf{Z}|\mathbf{T})$.

Though, the KL divergence is already difficult to deal with. Indeed,

$$\begin{aligned} KL(q||p) &= \int q(\mathbf{Z}) \log \left(\frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{T})} \right) d\mathbf{Z} = \int q(\mathbf{Z}) [\log(q(\mathbf{Z})) - \log(p(\mathbf{Z}|\mathbf{T}))] d\mathbf{Z} \\ &= \int q(\mathbf{Z}) \log(q(\mathbf{Z})) d\mathbf{Z} - \int q(\mathbf{Z}) \log(p(\mathbf{Z}|\mathbf{T})) d\mathbf{Z} = \mathbb{E}_q[\log(q(\mathbf{Z}))] - \mathbb{E}_q[\log(p(\mathbf{Z}|\mathbf{T}))] \\ &= \mathbb{E}_q[\log(q(\mathbf{Z}))] - \mathbb{E}_q \left[\log \left(\frac{p(\mathbf{Z}, \mathbf{T})}{p(\mathbf{T})} \right) \right] = \mathbb{E}_q[\log(p(\mathbf{T}))] - \mathbb{E}_q[\log(p(\mathbf{Z}, \mathbf{T})) - \log(q(\mathbf{Z}))] \\ &= \log(p(\mathbf{T})) - \mathbb{E}_q[\log(p(\mathbf{Z}, \mathbf{T})) - \log(q(\mathbf{Z}))] \end{aligned}$$

Bayesian Variational Inference

The idea behind variational inference is to replace $p(\mathbf{Z}|\mathbf{T})$ but a much simpler distribution to maximize, $q(\mathbf{Z})$ called the variational distribution. This distribution can be estimated by minimizing the Kullback-Leibler divergence between the two distributions:

$$KL(q||p) = \int q(\mathbf{Z}) \log \left(\frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{T})} \right) d\mathbf{Z}$$

If all possible “ q ” are available, minimizing this divergence should lead to $p(\mathbf{Z}|\mathbf{T})$. However here, this idea is to restrict the available space of “ q ” so that it is easy to compute but still close to $p(\mathbf{Z}|\mathbf{T})$.

Though, the KL divergence is already difficult to deal with. Indeed,

$$\begin{aligned} KL(q||p) &= \int q(\mathbf{Z}) \log \left(\frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{T})} \right) d\mathbf{Z} = \int q(\mathbf{Z}) [\log(q(\mathbf{Z})) - \log(p(\mathbf{Z}|\mathbf{T}))] d\mathbf{Z} \\ &= \int q(\mathbf{Z}) \log(q(\mathbf{Z})) d\mathbf{Z} - \int q(\mathbf{Z}) \log(p(\mathbf{Z}|\mathbf{T})) d\mathbf{Z} = \mathbb{E}_q[\log(q(\mathbf{Z}))] - \mathbb{E}_q[\log(p(\mathbf{Z}|\mathbf{T}))] \\ &= \mathbb{E}_q[\log(q(\mathbf{Z}))] - \mathbb{E}_q \left[\log \left(\frac{p(\mathbf{Z}, \mathbf{T})}{p(\mathbf{T})} \right) \right] = \mathbb{E}_q[\log(p(\mathbf{T}))] - \mathbb{E}_q[\log(p(\mathbf{Z}, \mathbf{T})) - \log(q(\mathbf{Z}))] \\ &= \underbrace{\log(p(\mathbf{T}))}_{\text{Intractable}} - \mathbb{E}_q[\log(p(\mathbf{Z}, \mathbf{T})) - \log(q(\mathbf{Z}))] \end{aligned}$$

Bayesian Variational Inference

The idea behind variational inference is to replace $p(\mathbf{Z}|\mathbf{T})$ but a much simpler distribution to maximize, $q(\mathbf{Z})$ called the variational distribution. This distribution can be estimated by minimizing the Kullback-Leibler divergence between the two distributions:

$$KL(q||p) = \int q(\mathbf{Z}) \log \left(\frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{T})} \right) d\mathbf{Z}$$

If all possible “ q ” are available, minimizing this divergence should lead to $p(\mathbf{Z}|\mathbf{T})$. However here, this idea is to restrict the available space of “ q ” so that it is easy to compute but still close to $p(\mathbf{Z}|\mathbf{T})$.

Though, the KL divergence is already difficult to dealt with. Indeed,

$$\begin{aligned} KL(q||p) &= \int q(\mathbf{Z}) \log \left(\frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{T})} \right) d\mathbf{Z} = \int q(\mathbf{Z}) [\log(q(\mathbf{Z})) - \log(p(\mathbf{Z}|\mathbf{T}))] d\mathbf{Z} \\ &= \int q(\mathbf{Z}) \log(q(\mathbf{Z})) d\mathbf{Z} - \int q(\mathbf{Z}) \log(p(\mathbf{Z}|\mathbf{T})) d\mathbf{Z} = \mathbb{E}_q[\log(q(\mathbf{Z}))] - \mathbb{E}_q[\log(p(\mathbf{Z}|\mathbf{T}))] \\ &= \mathbb{E}_q[\log(q(\mathbf{Z}))] - \mathbb{E}_q \left[\log \left(\frac{p(\mathbf{Z}, \mathbf{T})}{p(\mathbf{T})} \right) \right] = \mathbb{E}_q[\log(p(\mathbf{T}))] - \mathbb{E}_q[\log(p(\mathbf{Z}, \mathbf{T})) - \log(q(\mathbf{Z}))] \\ &= \underbrace{\log(p(\mathbf{T}))}_{\text{Intractable}} - \underbrace{\mathbb{E}_q[\log(p(\mathbf{Z}, \mathbf{T})) - \log(q(\mathbf{Z}))]}_{\mathcal{L}(q)} \end{aligned}$$

Evidence Lower BOund (ELBO)



Evidence Lower BOund (ELBO)

$\mathcal{L}(q)$ is also called the Evidence LOver Bound, because :



Evidence Lower BOund (ELBO)

$\mathcal{L}(q)$ is also called the Evidence LOver Bound, because :

$$\mathcal{L}(q) = \log(p(\mathbf{T})) - KL(q||p)$$

Evidence Lower BOund (ELBO)

$\mathcal{L}(q)$ is also called the Evidence LOver Bound, because :

$$\mathcal{L}(q) = \log(p(\mathbf{T})) - KL(q||p) \leq \log(p(\mathbf{T}))$$

Evidence Lower BOund (ELBO)

$\mathcal{L}(q)$ is also called the Evidence LOver Bound, because :

$$\mathcal{L}(q) = \log(p(\mathbf{T})) - KL(q||p) \leq \log(p(\mathbf{T}))$$

Where $\log(p(\mathbf{T}))$ is also called the evidence. Maximizing $\mathcal{L}(q)$ with respect to “ q ” is similar to minimizing $KL(q||p)$ with respect to “ q ” as the evidence does not depends on “ q ”.

Evidence Lower BOund (ELBO)

$\mathcal{L}(q)$ is also called the Evidence LOver Bound, because :

$$\mathcal{L}(q) = \log(p(\mathbf{T})) - KL(q||p) \leq \log(p(\mathbf{T}))$$

Where $\log(p(\mathbf{T}))$ is also called the evidence. Maximizing $\mathcal{L}(q)$ with respect to “ q ” is similar to minimizing $KL(q||p)$ with respect to “ q ” as the evidence does not depends on “ q ”.

Thus, Bayesian Variational Inference proposes to solve the following optimization problem:

Evidence Lower BOund (ELBO)

$\mathcal{L}(q)$ is also called the Evidence LOver Bound, because :

$$\mathcal{L}(q) = \log(p(\mathbf{T})) - KL(q||p) \leq \log(p(\mathbf{T}))$$

Where $\log(p(\mathbf{T}))$ is also called the evidence. Maximizing $\mathcal{L}(q)$ with respect to “ q ” is similar to minimizing $KL(q||p)$ with respect to “ q ” as the evidence does not depends on “ q ”.

Thus, Bayesian Variational Inference proposes to solve the following optimization problem:

$$\hat{q} = \underset{q}{argmax} \mathcal{L}(q)$$

Evidence Lower BOund (ELBO)

$\mathcal{L}(q)$ is also called the Evidence LOver Bound, because :

$$\mathcal{L}(q) = \log(p(\mathbf{T})) - KL(q||p) \leq \log(p(\mathbf{T}))$$

Where $\log(p(\mathbf{T}))$ is also called the evidence. Maximizing $\mathcal{L}(q)$ with respect to “ q ” is similar to minimizing $KL(q||p)$ with respect to “ q ” as the evidence does not depends on “ q ”.

Thus, Bayesian Variational Inference proposes to solve the following optimization problem:

$$\hat{q} = \underset{q}{\operatorname{argmax}} \mathcal{L}(q) = \underset{q}{\operatorname{argmin}} KL(q||p)$$

Mean Field Approximation



Mean Field Approximation

In practice, such optimization can still be hard to solve and assume that the “ q ” distribution will take the following specific form:



Mean Field Approximation

In practice, such optimization can still be hard to solve and assume that the “ q ” distribution will take the following specific form:

$$q(\mathbf{Z}) =$$

Mean Field Approximation

In practice, such optimization can still be hard to solve and assume that the “ q ” distribution will take the following specific form:

$$q(\mathbf{Z}) = \prod_{i=1}^{P=Nb_{parameters}} q_j(\mathbf{Z}_j)$$

Mean Field Approximation

In practice, such optimization can still be hard to solve and assume that the “ q ” distribution will take the following specific form:

$$q(\mathbf{Z}) = \prod_{i=1}^{P=Nb_{parameters}} q_j(\mathbf{z}_j) = \prod_{i=1}^N \prod_{k=1}^K q_{ik}(z_i^{(k)})$$

Mean Field Approximation

In practice, such optimization can still be hard to solve and assume that the “ q ” distribution will take the following specific form:

$$q(\mathbf{Z}) = \prod_{i=1}^{P=Nb_{parameters}} q_j(\mathbf{z}_j) = \prod_{i=1}^N \prod_{k=1}^K q_{ik}(z_i^{(k)})$$

A distribution ascent algorithm can thus be derived. If we recall the expression for the ELBO:

Mean Field Approximation

In practice, such optimization can still be hard to solve and assume that the “ q ” distribution will take the following specific form:

$$q(\mathbf{Z}) = \prod_{i=1}^{P=Nb_{parameters}} q_j(\mathbf{z}_j) = \prod_{i=1}^N \prod_{k=1}^K q_{ik}(z_i^{(k)})$$

A distribution ascent algorithm can thus be derived. If we recall the expression for the ELBO:

$$\mathcal{L}(q) = \mathbb{E}_q[\log(p(\mathbf{Z}, \mathbf{T}))] - \mathbb{E}_q[\log(q(\mathbf{Z}))]$$

Mean Field Approximation

In practice, such optimization can still be hard to solve and assume that the “ q ” distribution will take the following specific form:

$$q(\mathbf{Z}) = \prod_{i=1}^{P=Nb_{parameters}} q_j(\mathbf{Z}_j) = \prod_{i=1}^N \prod_{k=1}^K q_{ik}(z_i^{(k)})$$

A distribution ascent algorithm can thus be derived. If we recall the expression for the ELBO:

$$\mathcal{L}(q) = \mathbb{E}_q[\log(p(\mathbf{Z}, \mathbf{T}))] - \mathbb{E}_q[\log(q(\mathbf{Z}))]$$

Let us start by express $\mathbb{E}_q[\log(q(\mathbf{Z}))]$ as a function of $q_i(\mathbf{Z}_i)$:

Mean Field Approximation

In practice, such optimization can still be hard to solve and assume that the “ q ” distribution will take the following specific form:

$$q(\mathbf{Z}) = \prod_{i=1}^{P=Nb_{parameters}} q_j(\mathbf{Z}_j) = \prod_{i=1}^N \prod_{k=1}^K q_{ik}(z_i^{(k)})$$

A distribution ascent algorithm can thus be derived. If we recall the expression for the ELBO:

$$\mathcal{L}(q) = \mathbb{E}_q[\log(p(\mathbf{Z}, \mathbf{T}))] - \mathbb{E}_q[\log(q(\mathbf{Z}))]$$

Let us start by express $\mathbb{E}_q[\log(q(\mathbf{Z}))]$ as a function of $q_i(\mathbf{Z}_i)$:

$$\mathbb{E}_q[\log(q(\mathbf{Z}))] = \int_{\mathbf{Z}} q(\mathbf{Z}) \log(q(\mathbf{Z})) d\mathbf{Z}$$

Mean Field Approximation

In practice, such optimization can still be hard to solve and assume that the “ q ” distribution will take the following specific form:

$$q(\mathbf{Z}) = \prod_{i=1}^{P=Nb_{parameters}} q_j(\mathbf{Z}_j) = \prod_{i=1}^N \prod_{k=1}^K q_{ik}(z_i^{(k)})$$

A distribution ascent algorithm can thus be derived. If we recall the expression for the ELBO:

$$\mathcal{L}(q) = \mathbb{E}_q[\log(p(\mathbf{Z}, \mathbf{T}))] - \mathbb{E}_q[\log(q(\mathbf{Z}))]$$

Let us start by express $\mathbb{E}_q[\log(q(\mathbf{Z}))]$ as a function of $q_i(\mathbf{Z}_i)$:

$$\mathbb{E}_q[\log(q(\mathbf{Z}))] = \int_{\mathbf{Z}} q(\mathbf{Z}) \log(q(\mathbf{Z})) d\mathbf{Z} = \int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{j=1}^P q_j(\mathbf{Z}_j) \right) \log \left(\prod_{j=1}^P q_j(\mathbf{Z}_j) \right) d\mathbf{Z}_1 \dots d\mathbf{Z}_P$$

Mean Field Approximation

In practice, such optimization can still be hard to solve and assume that the “ q ” distribution will take the following specific form:

$$q(\mathbf{Z}) = \prod_{i=1}^{P=Nb_{parameters}} q_j(\mathbf{Z}_j) = \prod_{i=1}^N \prod_{k=1}^K q_{ik}(z_i^{(k)})$$

A distribution ascent algorithm can thus be derived. If we recall the expression for the ELBO:

$$\mathcal{L}(q) = \mathbb{E}_q[\log(p(\mathbf{Z}, \mathbf{T}))] - \mathbb{E}_q[\log(q(\mathbf{Z}))]$$

Let us start by express $\mathbb{E}_q[\log(q(\mathbf{Z}))]$ as a function of $q_i(\mathbf{Z}_i)$:

$$\begin{aligned} \mathbb{E}_q[\log(q(\mathbf{Z}))] &= \int_{\mathbf{Z}} q(\mathbf{Z}) \log(q(\mathbf{Z})) d\mathbf{Z} = \int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{j=1}^P q_j(\mathbf{Z}_j) \right) \log \left(\prod_{j=1}^P q_j(\mathbf{Z}_j) \right) d\mathbf{Z}_1 \dots d\mathbf{Z}_P \\ &= \int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{j=1}^P q_j(\mathbf{Z}_j) \right) \sum_{j=1}^P \log(q_j(\mathbf{Z}_j)) d\mathbf{Z}_1 \dots d\mathbf{Z}_P \end{aligned}$$

Mean Field Approximation

In practice, such optimization can still be hard to solve and assume that the “ q ” distribution will take the following specific form:

$$q(\mathbf{Z}) = \prod_{i=1}^{P=Nb_{parameters}} q_j(\mathbf{Z}_j) = \prod_{i=1}^N \prod_{k=1}^K q_{ik}(z_i^{(k)})$$

A distribution ascent algorithm can thus be derived. If we recall the expression for the ELBO:

$$\mathcal{L}(q) = \mathbb{E}_q[\log(p(\mathbf{Z}, \mathbf{T}))] - \mathbb{E}_q[\log(q(\mathbf{Z}))]$$

Let us start by express $\mathbb{E}_q[\log(q(\mathbf{Z}))]$ as a function of $q_i(\mathbf{Z}_i)$:

$$\begin{aligned} \mathbb{E}_q[\log(q(\mathbf{Z}))] &= \int_{\mathbf{Z}} q(\mathbf{Z}) \log(q(\mathbf{Z})) d\mathbf{Z} = \int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{j=1}^P q_j(\mathbf{Z}_j) \right) \log \left(\prod_{j=1}^P q_j(\mathbf{Z}_j) \right) d\mathbf{Z}_1 \dots d\mathbf{Z}_P \\ &= \int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{j=1}^P q_j(\mathbf{Z}_j) \right) \sum_{j=1}^P \log(q_j(\mathbf{Z}_j)) d\mathbf{Z}_1 \dots d\mathbf{Z}_P \\ &= \int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{j=1}^P q_j(\mathbf{Z}_j) \right) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_1 \dots d\mathbf{Z}_P + \int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{j=1}^P q_j(\mathbf{Z}_j) \right) \sum_{\substack{j=1 \\ j \neq i}}^P \log(q_j(\mathbf{Z}_j)) d\mathbf{Z}_1 \dots d\mathbf{Z}_P \end{aligned}$$

Mean Field Approximation

In practice, such optimization can still be hard to solve and assume that the “ q ” distribution will take the following specific form:

$$q(\mathbf{Z}) = \prod_{i=1}^{P=Nb_{parameters}} q_j(\mathbf{Z}_j) = \prod_{i=1}^N \prod_{k=1}^K q_{ik}(z_i^{(k)})$$

A distribution ascent algorithm can thus be derived. If we recall the expression for the ELBO:

$$\mathcal{L}(q) = \mathbb{E}_q[\log(p(\mathbf{Z}, \mathbf{T}))] - \mathbb{E}_q[\log(q(\mathbf{Z}))]$$

Let us start by express $\mathbb{E}_q[\log(q(\mathbf{Z}))]$ as a function of $q_i(\mathbf{Z}_i)$:

$$\begin{aligned} \mathbb{E}_q[\log(q(\mathbf{Z}))] &= \int_{\mathbf{Z}} q(\mathbf{Z}) \log(q(\mathbf{Z})) d\mathbf{Z} = \int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{j=1}^P q_j(\mathbf{Z}_j) \right) \log \left(\prod_{j=1}^P q_j(\mathbf{Z}_j) \right) d\mathbf{Z}_1 \dots d\mathbf{Z}_P \\ &= \int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{j=1}^P q_j(\mathbf{Z}_j) \right) \sum_{j=1}^P \log(q_j(\mathbf{Z}_j)) d\mathbf{Z}_1 \dots d\mathbf{Z}_P \\ &= \int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{j=1}^P q_j(\mathbf{Z}_j) \right) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_1 \dots d\mathbf{Z}_P + \underbrace{\int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{j=1}^P q_j(\mathbf{Z}_j) \right) \sum_{\substack{j=1 \\ j \neq i}}^P \log(q_j(\mathbf{Z}_j)) d\mathbf{Z}_1 \dots d\mathbf{Z}_P}_{\mathcal{C}} \end{aligned}$$

$$\mathcal{C} = \text{Constante}(q_1, q_2, \dots, q_{i-1}, q_{i+1}, \dots, q_p)$$

Mean Field Approximation



Mean Field Approximation

$$\mathbb{E}_q[\log(q(\mathbf{Z}))] =$$

Mean Field Approximation

$$\mathbb{E}_q[\log(q(\mathbf{Z}))] = \int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{i=1}^P q_i(\mathbf{Z}_i) \right) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_1 \dots d\mathbf{Z}_P + \mathcal{C}$$

Mean Field Approximation

$$\begin{aligned}\mathbb{E}_q[\log(q(\mathbf{Z}))] &= \int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{i=1}^P q_i(\mathbf{Z}_i) \right) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_1 \dots d\mathbf{Z}_P + \mathcal{C} \\ &= \int_{\mathbf{Z}_1} q_1(\mathbf{Z}_1) d\mathbf{Z}_1 \int_{\mathbf{Z}_2} q_2(\mathbf{Z}_2) d\mathbf{Z}_2 \dots \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_i \dots \int_{\mathbf{Z}_p} q_p(\mathbf{Z}_p) d\mathbf{Z}_p + \mathcal{C}\end{aligned}$$

Mean Field Approximation

$$\begin{aligned}\mathbb{E}_q[\log(q(\mathbf{Z}))] &= \int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{i=1}^P q_i(\mathbf{Z}_i) \right) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_1 \dots d\mathbf{Z}_P + \mathcal{C} \\ &= \underbrace{\int_{\mathbf{Z}_1} q_1(\mathbf{Z}_1) d\mathbf{Z}_1}_1 \underbrace{\int_{\mathbf{Z}_2} q_2(\mathbf{Z}_2) d\mathbf{Z}_2}_{1} \dots \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_i \dots \underbrace{\int_{\mathbf{Z}_p} q_p(\mathbf{Z}_p) d\mathbf{Z}_p}_1 + \mathcal{C}\end{aligned}$$

Mean Field Approximation

$$\begin{aligned}\mathbb{E}_q[\log(q(\mathbf{Z}))] &= \int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{i=1}^P q_i(\mathbf{Z}_i) \right) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_1 \dots d\mathbf{Z}_P + \mathcal{C} \\ &= \underbrace{\int_{\mathbf{Z}_1} q_1(\mathbf{Z}_1) d\mathbf{Z}_1}_1 \underbrace{\int_{\mathbf{Z}_2} q_2(\mathbf{Z}_2) d\mathbf{Z}_2 \dots \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_i \dots \int_{\mathbf{Z}_p} q_p(\mathbf{Z}_p) d\mathbf{Z}_p}_1 + \mathcal{C} \\ &= \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_i + \mathcal{C}\end{aligned}$$

Mean Field Approximation

$$\begin{aligned}\mathbb{E}_q[\log(q(\mathbf{Z}))] &= \int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{i=1}^P q_i(\mathbf{Z}_i) \right) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_1 \dots d\mathbf{Z}_P + \mathcal{C} \\ &= \underbrace{\int_{\mathbf{Z}_1} q_1(\mathbf{Z}_1) d\mathbf{Z}_1}_1 \underbrace{\int_{\mathbf{Z}_2} q_2(\mathbf{Z}_2) d\mathbf{Z}_2}_{1} \dots \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_i \dots \underbrace{\int_{\mathbf{Z}_p} q_p(\mathbf{Z}_p) d\mathbf{Z}_p}_1 + \mathcal{C} \\ &= \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_i + \mathcal{C}\end{aligned}$$

Now let us express $\mathbb{E}_q[\log(p(\mathbf{Z}, \mathbf{T}))]$ as a function of $q_i(\mathbf{Z}_i)$:

Mean Field Approximation

$$\begin{aligned}\mathbb{E}_q[\log(q(\mathbf{Z}))] &= \int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{i=1}^P q_i(\mathbf{Z}_i) \right) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_1 \dots d\mathbf{Z}_P + \mathcal{C} \\ &= \underbrace{\int_{\mathbf{Z}_1} q_1(\mathbf{Z}_1) d\mathbf{Z}_1}_1 \underbrace{\int_{\mathbf{Z}_2} q_2(\mathbf{Z}_2) d\mathbf{Z}_2}_{1} \dots \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_i \dots \underbrace{\int_{\mathbf{Z}_p} q_p(\mathbf{Z}_p) d\mathbf{Z}_p}_1 + \mathcal{C} \\ &= \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_i + \mathcal{C}\end{aligned}$$

Now let us express $\mathbb{E}_q[\log(p(\mathbf{Z}, \mathbf{T}))]$ as a function of $q_i(\mathbf{Z}_i)$:

$$\mathbb{E}_q[\log(p(\mathbf{Z}, \mathbf{T}))] = \int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{i=1}^P q_i(\mathbf{Z}_i) \right) \log(p(\mathbf{Z}, \mathbf{T})) d\mathbf{Z}_1 \dots d\mathbf{Z}_P$$

Mean Field Approximation

$$\begin{aligned}\mathbb{E}_q[\log(q(\mathbf{Z}))] &= \int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{i=1}^P q_i(\mathbf{Z}_i) \right) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_1 \dots d\mathbf{Z}_P + \mathcal{C} \\ &= \underbrace{\int_{\mathbf{Z}_1} q_1(\mathbf{Z}_1) d\mathbf{Z}_1}_1 \underbrace{\int_{\mathbf{Z}_2} q_2(\mathbf{Z}_2) d\mathbf{Z}_2 \dots \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_i \dots \int_{\mathbf{Z}_p} q_p(\mathbf{Z}_p) d\mathbf{Z}_p}_1 + \mathcal{C} \\ &= \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_i + \mathcal{C}\end{aligned}$$

Now let us express $\mathbb{E}_q[\log(p(\mathbf{Z}, \mathbf{T}))]$ as a function of $q_i(\mathbf{Z}_i)$:

$$\begin{aligned}\mathbb{E}_q[\log(p(\mathbf{Z}, \mathbf{T}))] &= \int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{i=1}^P q_i(\mathbf{Z}_i) \right) \log(p(\mathbf{Z}, \mathbf{T})) d\mathbf{Z}_1 \dots d\mathbf{Z}_P \\ &= \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \left[\int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{\substack{j=1 \\ j \neq i}}^P q_j(\mathbf{Z}_j) \right) \log(p(\mathbf{Z}, \mathbf{T})) d\mathbf{Z}_1 \dots d\mathbf{Z}_P \right] d\mathbf{Z}_i\end{aligned}$$

Mean Field Approximation

$$\begin{aligned}
 \mathbb{E}_q[\log(q(\mathbf{Z}))] &= \int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{i=1}^P q_i(\mathbf{Z}_i) \right) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_1 \dots d\mathbf{Z}_P + \mathcal{C} \\
 &= \underbrace{\int_{\mathbf{Z}_1} q_1(\mathbf{Z}_1) d\mathbf{Z}_1}_1 \underbrace{\int_{\mathbf{Z}_2} q_2(\mathbf{Z}_2) d\mathbf{Z}_2 \dots \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_i \dots \int_{\mathbf{Z}_p} q_p(\mathbf{Z}_p) d\mathbf{Z}_p}_1 + \mathcal{C} \\
 &= \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_i + \mathcal{C}
 \end{aligned}$$

Now let us express $\mathbb{E}_q[\log(p(\mathbf{Z}, \mathbf{T}))]$ as a function of $q_i(\mathbf{Z}_i)$:

$$\begin{aligned}
 \mathbb{E}_q[\log(p(\mathbf{Z}, \mathbf{T}))] &= \int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{i=1}^P q_i(\mathbf{Z}_i) \right) \log(p(\mathbf{Z}, \mathbf{T})) d\mathbf{Z}_1 \dots d\mathbf{Z}_P \\
 &= \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \underbrace{\left[\int_{\mathbf{Z}_1} \dots \int_{\mathbf{Z}_P} \left(\prod_{\substack{j=1 \\ j \neq i}}^P q_j(\mathbf{Z}_j) \right) \log(p(\mathbf{Z}, \mathbf{T})) d\mathbf{Z}_1 \dots d\mathbf{Z}_P \right]}_{\mathbb{E}_{j \neq i}[\log(p(\mathbf{Z}, \mathbf{T}))]} d\mathbf{Z}_i
 \end{aligned}$$

Mean Field Approximation



Mean Field Approximation

It is possible to introduce a distribution $\tilde{p}(\mathbf{Z}_i, \mathbf{T})$ such that:



Mean Field Approximation

It is possible to introduce a distribution $\tilde{p}(\mathbf{Z}_i, \mathbf{T})$ such that:

$$\log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) = \mathbb{E}_{j \neq i} [\log(p(\mathbf{Z}, \mathbf{T}))] + \mathcal{C}'(p)$$

Mean Field Approximation

It is possible to introduce a distribution $\tilde{p}(\mathbf{Z}_i, \mathbf{T})$ such that:

$$\log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) = \mathbb{E}_{j \neq i} [\log(p(\mathbf{Z}, \mathbf{T}))] + \mathcal{C}'(p)$$

Hence:

Mean Field Approximation

It is possible to introduce a distribution $\tilde{p}(\mathbf{Z}_i, \mathbf{T})$ such that:

$$\log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) = \mathbb{E}_{j \neq i} [\log(p(\mathbf{Z}, \mathbf{T}))] + \mathcal{C}'(p)$$

Hence:

$$\mathbb{E}_q [\log(p(\mathbf{Z}, \mathbf{T}))] = \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) d\mathbf{Z}_i - \mathcal{C}'(p)$$

Mean Field Approximation

It is possible to introduce a distribution $\tilde{p}(\mathbf{Z}_i, \mathbf{T})$ such that:

$$\log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) = \mathbb{E}_{j \neq i} [\log(p(\mathbf{Z}, \mathbf{T}))] + \mathcal{C}'(p)$$

Hence:

$$\mathbb{E}_q [\log(p(\mathbf{Z}, \mathbf{T}))] = \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) d\mathbf{Z}_i - \mathcal{C}'(p)$$

Hence, $\mathcal{L}(q)$ can be written as:

Mean Field Approximation

It is possible to introduce a distribution $\tilde{p}(\mathbf{Z}_i, \mathbf{T})$ such that:

$$\log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) = \mathbb{E}_{j \neq i} [\log(p(\mathbf{Z}, \mathbf{T}))] + \mathcal{C}'(p)$$

Hence:

$$\mathbb{E}_q [\log(p(\mathbf{Z}, \mathbf{T}))] = \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) d\mathbf{Z}_i - \mathcal{C}'(p)$$

Hence, $\mathcal{L}(q)$ can be written as:

$$\mathcal{L}(q) =$$

Mean Field Approximation

It is possible to introduce a distribution $\tilde{p}(\mathbf{Z}_i, \mathbf{T})$ such that:

$$\log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) = \mathbb{E}_{j \neq i} [\log(p(\mathbf{Z}, \mathbf{T}))] + \mathcal{C}'(p)$$

Hence:

$$\mathbb{E}_q [\log(p(\mathbf{Z}, \mathbf{T}))] = \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) d\mathbf{Z}_i - \mathcal{C}'(p)$$

Hence, $\mathcal{L}(q)$ can be written as:

$$\mathcal{L}(q) = \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) - \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_i + \mathcal{C}''$$

Mean Field Approximation

It is possible to introduce a distribution $\tilde{p}(\mathbf{Z}_i, \mathbf{T})$ such that:

$$\log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) = \mathbb{E}_{j \neq i} [\log(p(\mathbf{Z}, \mathbf{T}))] + \mathcal{C}'(p)$$

Hence:

$$\mathbb{E}_q [\log(p(\mathbf{Z}, \mathbf{T}))] = \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) d\mathbf{Z}_i - \mathcal{C}'(p)$$

Hence, $\mathcal{L}(q)$ can be written as:

$$\mathcal{L}(q) = \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) - \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_i + \mathcal{C}'' = -KL(q_i || \tilde{p}(\mathbf{Z}_i, \mathbf{T})) + \mathcal{C}''$$

Mean Field Approximation

It is possible to introduce a distribution $\tilde{p}(\mathbf{Z}_i, \mathbf{T})$ such that:

$$\log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) = \mathbb{E}_{j \neq i} [\log(p(\mathbf{Z}, \mathbf{T}))] + \mathcal{C}'(p)$$

Hence:

$$\mathbb{E}_q [\log(p(\mathbf{Z}, \mathbf{T}))] = \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) d\mathbf{Z}_i - \mathcal{C}'(p)$$

Hence, $\mathcal{L}(q)$ can be written as:

$$\mathcal{L}(q) = \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) - \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_i + \mathcal{C}'' = -KL(q_i || \tilde{p}(\mathbf{Z}_i, \mathbf{T})) + \mathcal{C}''$$

Hence, minimizing $\mathcal{L}(q)$ with respect to $q_i(\mathbf{Z}_i)$ leads to the following update:

Mean Field Approximation

It is possible to introduce a distribution $\tilde{p}(\mathbf{Z}_i, \mathbf{T})$ such that:

$$\log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) = \mathbb{E}_{j \neq i} [\log(p(\mathbf{Z}, \mathbf{T}))] + \mathcal{C}'(p)$$

Hence:

$$\mathbb{E}_q [\log(p(\mathbf{Z}, \mathbf{T}))] = \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) d\mathbf{Z}_i - \mathcal{C}'(p)$$

Hence, $\mathcal{L}(q)$ can be written as:

$$\mathcal{L}(q) = \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) - \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_i + \mathcal{C}'' = -KL(q_i || \tilde{p}(\mathbf{Z}_i, \mathbf{T})) + \mathcal{C}''$$

Hence, minimizing $\mathcal{L}(q)$ with respect to $q_i(\mathbf{Z}_i)$ leads to the following update:

$$\log(\hat{q}_i(\mathbf{Z}_i))$$

Mean Field Approximation

It is possible to introduce a distribution $\tilde{p}(\mathbf{Z}_i, \mathbf{T})$ such that:

$$\log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) = \mathbb{E}_{j \neq i} [\log(p(\mathbf{Z}, \mathbf{T}))] + \mathcal{C}'(p)$$

Hence:

$$\mathbb{E}_q [\log(p(\mathbf{Z}, \mathbf{T}))] = \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) d\mathbf{Z}_i - \mathcal{C}'(p)$$

Hence, $\mathcal{L}(q)$ can be written as:

$$\mathcal{L}(q) = \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) - \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_i + \mathcal{C}'' = -KL(q_i || \tilde{p}(\mathbf{Z}_i, \mathbf{T})) + \mathcal{C}''$$

Hence, minimizing $\mathcal{L}(q)$ with respect to $q_i(\mathbf{Z}_i)$ leads to the following update:

$$\log(\hat{q}_i(\mathbf{Z}_i)) = \log \left(\underset{q_i}{\operatorname{argmax}} \mathcal{L}(q) \right)$$

Mean Field Approximation

It is possible to introduce a distribution $\tilde{p}(\mathbf{Z}_i, \mathbf{T})$ such that:

$$\log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) = \mathbb{E}_{j \neq i} [\log(p(\mathbf{Z}, \mathbf{T}))] + \mathcal{C}'(p)$$

Hence:

$$\mathbb{E}_q [\log(p(\mathbf{Z}, \mathbf{T}))] = \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) d\mathbf{Z}_i - \mathcal{C}'(p)$$

Hence, $\mathcal{L}(q)$ can be written as:

$$\mathcal{L}(q) = \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) - \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_i + \mathcal{C}'' = -KL(q_i || \tilde{p}(\mathbf{Z}_i, \mathbf{T})) + \mathcal{C}''$$

Hence, minimizing $\mathcal{L}(q)$ with respect to $q_i(\mathbf{Z}_i)$ leads to the following update:

$$\log(\hat{q}_i(\mathbf{Z}_i)) = \log \left(\underset{q_i}{\operatorname{argmax}} \mathcal{L}(q) \right) = \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T}))$$

Mean Field Approximation

It is possible to introduce a distribution $\tilde{p}(\mathbf{Z}_i, \mathbf{T})$ such that:

$$\log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) = \mathbb{E}_{j \neq i} [\log(p(\mathbf{Z}, \mathbf{T}))] + \mathcal{C}'(p)$$

Hence:

$$\mathbb{E}_q [\log(p(\mathbf{Z}, \mathbf{T}))] = \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) d\mathbf{Z}_i - \mathcal{C}'(p)$$

Hence, $\mathcal{L}(q)$ can be written as:

$$\mathcal{L}(q) = \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) - \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_i + \mathcal{C}'' = -KL(q_i || \tilde{p}(\mathbf{Z}_i, \mathbf{T})) + \mathcal{C}''$$

Hence, minimizing $\mathcal{L}(q)$ with respect to $q_i(\mathbf{Z}_i)$ leads to the following update:

$$\log(\hat{q}_i(\mathbf{Z}_i)) = \log \left(\underset{q_i}{\operatorname{argmax}} \mathcal{L}(q) \right) = \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) = \mathbb{E}_{j \neq i} [\log(p(\mathbf{Z}, \mathbf{T}))] + \mathcal{C}'(p)$$

Mean Field Approximation

It is possible to introduce a distribution $\tilde{p}(\mathbf{Z}_i, \mathbf{T})$ such that:

$$\log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) = \mathbb{E}_{j \neq i} [\log(p(\mathbf{Z}, \mathbf{T}))] + \mathcal{C}'(p)$$

Hence:

$$\mathbb{E}_q [\log(p(\mathbf{Z}, \mathbf{T}))] = \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) d\mathbf{Z}_i - \mathcal{C}'(p)$$

Hence, $\mathcal{L}(q)$ can be written as:

$$\mathcal{L}(q) = \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) - \int_{\mathbf{Z}_i} q_i(\mathbf{Z}_i) \log(q_i(\mathbf{Z}_i)) d\mathbf{Z}_i + \mathcal{C}'' = -KL(q_i || \tilde{p}(\mathbf{Z}_i, \mathbf{T})) + \mathcal{C}''$$

Hence, minimizing $\mathcal{L}(q)$ with respect to $q_i(\mathbf{Z}_i)$ leads to the following update:

$$\log(\hat{q}_i(\mathbf{Z}_i)) = \log \left(\underset{q_i}{\operatorname{argmax}} \mathcal{L}(q) \right) = \log(\tilde{p}(\mathbf{Z}_i, \mathbf{T})) = \mathbb{E}_{j \neq i} [\log(p(\mathbf{Z}, \mathbf{T}))] + \mathcal{C}'(p)$$

We can then alternate for each factor until convergence. Convergence is guaranteed because bound is convex with respect to each of the factors $q_i(\mathbf{Z}_i)$ (Boyd and Vandenberghe, 2004)...

Back to MOTL



Back to MOTL

In the case of MOTL, what is $\mathbb{E}_{j \neq i} [\log(p(\mathbf{Z}, \mathbf{T}))]$?



Back to MOTL

In the case of MOTL, what is $\mathbb{E}_{j \neq i} [\log(p(\mathbf{Z}, \mathbf{T}))]$?

$$\log(p(\mathbf{Z}, \mathbf{T})) = \log \left(\prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N} \left(a_j^{(m)} + \mathbf{w}_{j:i} \mathbf{z}_i, 1/\tau_j^{(m)} \right) \prod_{i=1}^N \prod_{k=1}^K \mathcal{N}(0, 1) \right)$$

Back to MOTL

In the case of MOTL, what is $\mathbb{E}_{j \neq i} [\log(p(\mathbf{Z}, \mathbf{T}))]$?

$$\log(p(\mathbf{Z}, \mathbf{T})) = \log \left(\prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N} \left(a_j^{(m)} + \mathbf{w}_{j:i} \mathbf{z}_i, 1/\tau_j^{(m)} \right) \prod_{i=1}^N \prod_{k=1}^K \mathcal{N}(0, 1) \right)$$

We recall that if $X \sim \mathcal{N}(\mu, \sigma^2)$, then:

Back to MOTL

In the case of MOTL, what is $\mathbb{E}_{j \neq i} [\log(p(\mathbf{Z}, \mathbf{T}))]$?

$$\log(p(\mathbf{Z}, \mathbf{T})) = \log \left(\prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N} \left(a_j^{(m)} + \mathbf{w}_{j:\mathbf{z}_i}, 1/\tau_j^{(m)} \right) \prod_{i=1}^N \prod_{k=1}^K \mathcal{N}(0, 1) \right)$$

We recall that if $X \sim \mathcal{N}(\mu, \sigma^2)$, then:

$$\log(p(X = x)) = -\frac{1}{2} \log(2\pi) - \log(\sigma) - \frac{1}{2} \left(\frac{x - \mu}{\sigma} \right)^2$$

Hence

Back to MOTL

In the case of MOTL, what is $\mathbb{E}_{j \neq i}[\log(p(\mathbf{Z}, \mathbf{T}))]$?

$$\log(p(\mathbf{Z}, \mathbf{T})) = \log \left(\prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N} \left(a_j^{(m)} + \mathbf{w}_{j:\mathbf{z}_i}, 1/\tau_j^{(m)} \right) \prod_{i=1}^N \prod_{k=1}^K \mathcal{N}(0, 1) \right)$$

We recall that if $X \sim \mathcal{N}(\mu, \sigma^2)$, then:

$$\log(p(X = x)) = -\frac{1}{2} \log(2\pi) - \log(\sigma) - \frac{1}{2} \left(\frac{x - \mu}{\sigma} \right)^2$$

Hence

$$\log(p(\mathbf{Z}, \mathbf{T})) = \sum_{m=1}^M \sum_{i=1}^N \sum_{j=1}^{p_m} \left[\frac{1}{2} \log(\tau_j^{(m)}) - \frac{1}{2} \log(2\pi) - \frac{\tau_j^{(m)}}{2} \left(t_{ij}^{(m)} - a_j^{(m)} - \mathbf{w}_{j:\mathbf{z}_i} \right)^2 \right] + \sum_{i=1}^N \sum_{k=1}^K \left[-\frac{1}{2} \log(2\pi) - \frac{1}{2} z_k^{(i)2} \right]$$

Back to MOTL

In the case of MOTL, what is $\mathbb{E}_{j \neq i}[\log(p(\mathbf{Z}, \mathbf{T}))]$?

$$\log(p(\mathbf{Z}, \mathbf{T})) = \log \left(\prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N} \left(a_j^{(m)} + \mathbf{w}_{j:\mathbf{z}_i}, 1/\tau_j^{(m)} \right) \prod_{i=1}^N \prod_{k=1}^K \mathcal{N}(0, 1) \right)$$

We recall that if $X \sim \mathcal{N}(\mu, \sigma^2)$, then:

$$\log(p(X = x)) = -\frac{1}{2} \log(2\pi) - \log(\sigma) - \frac{1}{2} \left(\frac{x - \mu}{\sigma} \right)^2$$

Hence

$$\log(p(\mathbf{Z}, \mathbf{T})) = \sum_{m=1}^M \sum_{i=1}^N \sum_{j=1}^{p_m} \left[\frac{1}{2} \log(\tau_j^{(m)}) - \frac{1}{2} \log(2\pi) - \frac{\tau_j^{(m)}}{2} \left(t_{ij}^{(m)} - a_j^{(m)} - \mathbf{w}_{j:\mathbf{z}_i} \right)^2 \right] + \sum_{i=1}^N \sum_{k=1}^K \left[-\frac{1}{2} \log(2\pi) - \frac{1}{2} z_k^{(i)^2} \right]$$

$$\mathbb{E}_{\mathbf{z}_l^{(n)}}[\log(p(\mathbf{Z}, \mathbf{T}))]$$

Back to MOTL

In the case of MOTL, what is $\mathbb{E}_{j \neq i}[\log(p(\mathbf{Z}, \mathbf{T}))]$?

$$\log(p(\mathbf{Z}, \mathbf{T})) = \log \left(\prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N} \left(a_j^{(m)} + \mathbf{w}_{j:\mathbf{z}_i}, 1/\tau_j^{(m)} \right) \prod_{i=1}^N \prod_{k=1}^K \mathcal{N}(0, 1) \right)$$

We recall that if $X \sim \mathcal{N}(\mu, \sigma^2)$, then:

$$\log(p(X = x)) = -\frac{1}{2} \log(2\pi) - \log(\sigma) - \frac{1}{2} \left(\frac{x - \mu}{\sigma} \right)^2$$

Hence

$$\begin{aligned} \log(p(\mathbf{Z}, \mathbf{T})) &= \sum_{m=1}^M \sum_{i=1}^N \sum_{j=1}^{p_m} \left[\frac{1}{2} \log(\tau_j^{(m)}) - \frac{1}{2} \log(2\pi) - \frac{\tau_j^{(m)}}{2} \left(t_{ij}^{(m)} - a_j^{(m)} - \mathbf{w}_{j:\mathbf{z}_i} \right)^2 \right] + \sum_{i=1}^N \sum_{k=1}^K \left[-\frac{1}{2} \log(2\pi) - \frac{1}{2} z_k^{(i)^2} \right] \\ \mathbb{E}_{\neq z_l^{(n)}}[\log(p(\mathbf{Z}, \mathbf{T}))] &= \mathbb{E}_{\neq z_l^{(n)}} \left\{ \sum_{i=1}^N \sum_{j=1}^{p_m} \left[-\frac{\tau_j^{(m)}}{2} \left(t_{ij}^{(m)} - a_j^{(m)} - \sum_{\substack{k=1 \\ k \neq l}}^K w_{jk} z_k^{(n)} - w_{jl} z_l^{(n)} \right)^2 \right] - \frac{1}{2} z_l^{(n)^2} + \mathcal{C} \right\} \end{aligned}$$

Back to MOTL

In the case of MOTL, what is $\mathbb{E}_{j \neq i}[\log(p(\mathbf{Z}, \mathbf{T}))]$?

$$\log(p(\mathbf{Z}, \mathbf{T})) = \log \left(\prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N} \left(a_j^{(m)} + \mathbf{w}_{j:\mathbf{z}_i}, 1/\tau_j^{(m)} \right) \prod_{i=1}^N \prod_{k=1}^K \mathcal{N}(0, 1) \right)$$

We recall that if $X \sim \mathcal{N}(\mu, \sigma^2)$, then:

$$\log(p(X = x)) = -\frac{1}{2} \log(2\pi) - \log(\sigma) - \frac{1}{2} \left(\frac{x - \mu}{\sigma} \right)^2$$

Hence

$$\begin{aligned} \log(p(\mathbf{Z}, \mathbf{T})) &= \sum_{m=1}^M \sum_{i=1}^N \sum_{j=1}^{p_m} \left[\frac{1}{2} \log(\tau_j^{(m)}) - \frac{1}{2} \log(2\pi) - \frac{\tau_j^{(m)}}{2} \left(t_{ij}^{(m)} - a_j^{(m)} - \mathbf{w}_{j:\mathbf{z}_i} \right)^2 \right] + \sum_{i=1}^N \sum_{k=1}^K \left[-\frac{1}{2} \log(2\pi) - \frac{1}{2} z_k^{(i)2} \right] \\ \mathbb{E}_{\neq z_l^{(n)}}[\log(p(\mathbf{Z}, \mathbf{T}))] &= \mathbb{E}_{\neq z_l^{(n)}} \left\{ \sum_{i=1}^N \sum_{j=1}^{p_m} \left[-\frac{\tau_j^{(m)}}{2} \left(t_{ij}^{(m)} - a_j^{(m)} - \sum_{\substack{k=1 \\ k \neq l}}^K w_{jk} z_k^{(n)} - w_{jl} z_l^{(n)} \right)^2 \right] - \frac{1}{2} z_l^{(n)2} + \mathcal{C} \right\} \\ &= \mathbb{E}_{\neq z_l^{(n)}} \left\{ -\frac{1}{2} z_l^{(n)2} \left(1 + \sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl}^2 \right) + z_l^{(n)} \left(\sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl} \left(t_{ij}^{(m)} - a_j^{(m)} - \sum_{\substack{k=1 \\ k \neq l}}^K w_{jk} z_k^{(n)} \right) \right) + \mathcal{C} \right\} \end{aligned}$$

Back to MOTL

In the case of MOTL, what is $\mathbb{E}_{j \neq i}[\log(p(\mathbf{Z}, \mathbf{T}))]$?

$$\log(p(\mathbf{Z}, \mathbf{T})) = \log \left(\prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N} \left(a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i, 1/\tau_j^{(m)} \right) \prod_{i=1}^N \prod_{k=1}^K \mathcal{N}(0, 1) \right)$$

We recall that if $X \sim \mathcal{N}(\mu, \sigma^2)$, then:

$$\log(p(X = x)) = -\frac{1}{2} \log(2\pi) - \log(\sigma) - \frac{1}{2} \left(\frac{x - \mu}{\sigma} \right)^2$$

Hence

$$\begin{aligned} \log(p(\mathbf{Z}, \mathbf{T})) &= \sum_{m=1}^M \sum_{i=1}^N \sum_{j=1}^{p_m} \left[\frac{1}{2} \log(\tau_j^{(m)}) - \frac{1}{2} \log(2\pi) - \frac{\tau_j^{(m)}}{2} \left(t_{ij}^{(m)} - a_j^{(m)} - \mathbf{w}_{j:} \mathbf{z}_i \right)^2 \right] + \sum_{i=1}^N \sum_{k=1}^K \left[-\frac{1}{2} \log(2\pi) - \frac{1}{2} z_k^{(i)2} \right] \\ \mathbb{E}_{\neq z_l^{(n)}}[\log(p(\mathbf{Z}, \mathbf{T}))] &= \mathbb{E}_{\neq z_l^{(n)}} \left\{ \sum_{i=1}^N \sum_{j=1}^{p_m} \left[-\frac{\tau_j^{(m)}}{2} \left(t_{ij}^{(m)} - a_j^{(m)} - \sum_{\substack{k=1 \\ k \neq l}}^K w_{jk} z_k^{(n)} - w_{jl} z_l^{(n)} \right)^2 \right] - \frac{1}{2} z_l^{(n)2} + \mathcal{C} \right\} \\ &= \mathbb{E}_{\neq z_l^{(n)}} \left\{ -\frac{1}{2} z_l^{(n)2} \left(1 + \sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl}^2 \right) + z_l^{(n)} \left(\sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl} \left(t_{ij}^{(m)} - a_j^{(m)} - \sum_{\substack{k=1 \\ k \neq l}}^K w_{jk} z_k^{(n)} \right) \right) + \mathcal{C} \right\} \\ &= -\frac{1}{2} z_l^{(n)2} \left(1 + \sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl}^2 \right) + z_l^{(n)} \left(\sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl} \left(t_{ij}^{(m)} - a_j^{(m)} - \sum_{\substack{k=1 \\ k \neq l}}^K w_{jk} \mathbb{E}_{\neq z_l^{(n)}}[z_k^{(n)}] \right) \right) + \mathcal{C}' \end{aligned}$$

Back to MOTL

In the case of MOTL, what is $\mathbb{E}_{j \neq i}[\log(p(\mathbf{Z}, \mathbf{T}))]$?

$$\log(p(\mathbf{Z}, \mathbf{T})) = \log \left(\prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N}(a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i, 1/\tau_j^{(m)}) \prod_{i=1}^N \prod_{k=1}^K \mathcal{N}(0, 1) \right)$$

We recall that if $X \sim \mathcal{N}(\mu, \sigma^2)$, then:

$$\log(p(X = x)) = -\frac{1}{2} \log(2\pi) - \log(\sigma) - \frac{1}{2} \left(\frac{x - \mu}{\sigma} \right)^2$$

Hence

$$\begin{aligned} \log(p(\mathbf{Z}, \mathbf{T})) &= \sum_{m=1}^M \sum_{i=1}^N \sum_{j=1}^{p_m} \left[\frac{1}{2} \log(\tau_j^{(m)}) - \frac{1}{2} \log(2\pi) - \frac{\tau_j^{(m)}}{2} (t_{ij}^{(m)} - a_j^{(m)} - \mathbf{w}_{j:} \mathbf{z}_i)^2 \right] + \sum_{i=1}^N \sum_{k=1}^K \left[-\frac{1}{2} \log(2\pi) - \frac{1}{2} z_k^{(i)2} \right] \\ \mathbb{E}_{\neq z_l^{(n)}}[\log(p(\mathbf{Z}, \mathbf{T}))] &= \mathbb{E}_{\neq z_l^{(n)}} \left\{ \sum_{i=1}^N \sum_{j=1}^{p_m} \left[-\frac{\tau_j^{(m)}}{2} \left(t_{ij}^{(m)} - a_j^{(m)} - \sum_{\substack{k=1 \\ k \neq l}}^K w_{jk} z_k^{(n)} - w_{jl} z_l^{(n)} \right)^2 \right] - \frac{1}{2} z_l^{(n)2} + \mathcal{C} \right\} \\ &= \mathbb{E}_{\neq z_l^{(n)}} \left\{ -\frac{1}{2} z_l^{(n)2} \left(1 + \sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl}^2 \right) + z_l^{(n)} \left(\sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl} \left(t_{ij}^{(m)} - a_j^{(m)} - \sum_{\substack{k=1 \\ k \neq l}}^K w_{jk} z_k^{(n)} \right) \right) + \mathcal{C} \right\} \\ &= -\frac{1}{2} z_l^{(n)2} \left(1 + \sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl}^2 \right) + z_l^{(n)} \left(\sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl} \left(t_{ij}^{(m)} - a_j^{(m)} - \sum_{\substack{k=1 \\ k \neq l}}^K w_{jk} \mathbb{E}_{\neq z_l^{(n)}}[z_k^{(n)}] \right) \right) + \mathcal{C}' \end{aligned}$$

Back to MOTL

In the case of MOTL, what is $\mathbb{E}_{j \neq i}[\log(p(\mathbf{Z}, \mathbf{T}))]$?

$$\log(p(\mathbf{Z}, \mathbf{T})) = \log \left(\prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N} \left(a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i, 1/\tau_j^{(m)} \right) \prod_{i=1}^N \prod_{k=1}^K \mathcal{N}(0, 1) \right)$$

We recall that if $X \sim \mathcal{N}(\mu, \sigma^2)$, then:

$$\log(p(X = x)) = -\frac{1}{2} \log(2\pi) - \log(\sigma) - \frac{1}{2} \left(\frac{x - \mu}{\sigma} \right)^2$$

Hence

$$\begin{aligned} \log(p(\mathbf{Z}, \mathbf{T})) &= \sum_{m=1}^M \sum_{i=1}^N \sum_{j=1}^{p_m} \left[\frac{1}{2} \log(\tau_j^{(m)}) - \frac{1}{2} \log(2\pi) - \frac{\tau_j^{(m)}}{2} \left(t_{ij}^{(m)} - a_j^{(m)} - \mathbf{w}_{j:} \mathbf{z}_i \right)^2 \right] + \sum_{i=1}^N \sum_{k=1}^K \left[-\frac{1}{2} \log(2\pi) - \frac{1}{2} z_k^{(i)2} \right] \\ \mathbb{E}_{\neq z_l^{(n)}}[\log(p(\mathbf{Z}, \mathbf{T}))] &= \mathbb{E}_{\neq z_l^{(n)}} \left\{ \sum_{i=1}^N \sum_{j=1}^{p_m} \left[-\frac{\tau_j^{(m)}}{2} \left(t_{ij}^{(m)} - a_j^{(m)} - \sum_{\substack{k=1 \\ k \neq l}}^K w_{jk} z_k^{(n)} - w_{jl} z_l^{(n)} \right)^2 \right] - \frac{1}{2} z_l^{(n)2} + \mathcal{C} \right\} \\ &= \mathbb{E}_{\neq z_l^{(n)}} \left\{ -\frac{1}{2} z_l^{(n)2} \left(1 + \sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl}^2 \right) + z_l^{(n)} \left(\sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl} \left(t_{ij}^{(m)} - a_j^{(m)} - \sum_{\substack{k=1 \\ k \neq l}}^K w_{jk} z_k^{(n)} \right) \right) + \mathcal{C} \right\} \\ &= -\frac{1}{2} z_l^{(n)2} \left(1 + \sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl}^2 \right) + z_l^{(n)} \left(\sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl} \left(t_{ij}^{(m)} - a_j^{(m)} - \sum_{\substack{k=1 \\ k \neq l}}^K w_{jk} \mathbb{E}_{\neq z_l^{(n)}}[z_k^{(n)}] \right) \right) + \mathcal{C}' \end{aligned}$$

$1/\sigma_{nl}^2$

Back to MOTL

In the case of MOTL, what is $\mathbb{E}_{j \neq i}[\log(p(\mathbf{Z}, \mathbf{T}))]$?

$$\log(p(\mathbf{Z}, \mathbf{T})) = \log \left(\prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N} \left(a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i, 1/\tau_j^{(m)} \right) \prod_{i=1}^N \prod_{k=1}^K \mathcal{N}(0, 1) \right)$$

We recall that if $X \sim \mathcal{N}(\mu, \sigma^2)$, then:

$$\log(p(X = x)) = -\frac{1}{2} \log(2\pi) - \log(\sigma) - \frac{1}{2} \left(\frac{x - \mu}{\sigma} \right)^2$$

Hence

$$\begin{aligned} \log(p(\mathbf{Z}, \mathbf{T})) &= \sum_{m=1}^M \sum_{i=1}^N \sum_{j=1}^{p_m} \left[\frac{1}{2} \log(\tau_j^{(m)}) - \frac{1}{2} \log(2\pi) - \frac{\tau_j^{(m)}}{2} \left(t_{ij}^{(m)} - a_j^{(m)} - \mathbf{w}_{j:} \mathbf{z}_i \right)^2 \right] + \sum_{i=1}^N \sum_{k=1}^K \left[-\frac{1}{2} \log(2\pi) - \frac{1}{2} z_k^{(i)2} \right] \\ \mathbb{E}_{\neq z_l^{(n)}}[\log(p(\mathbf{Z}, \mathbf{T}))] &= \mathbb{E}_{\neq z_l^{(n)}} \left\{ \sum_{i=1}^N \sum_{j=1}^{p_m} \left[-\frac{\tau_j^{(m)}}{2} \left(t_{ij}^{(m)} - a_j^{(m)} - \sum_{\substack{k=1 \\ k \neq l}}^K w_{jk} z_k^{(n)} - w_{jl} z_l^{(n)} \right)^2 \right] - \frac{1}{2} z_l^{(n)2} + \mathcal{C} \right\} \\ &= \mathbb{E}_{\neq z_l^{(n)}} \left\{ -\frac{1}{2} z_l^{(n)2} \left(1 + \sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl}^2 \right) + z_l^{(n)} \left(\sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl} \left(t_{ij}^{(m)} - a_j^{(m)} - \sum_{\substack{k=1 \\ k \neq l}}^K w_{jk} z_k^{(n)} \right) \right) + \mathcal{C} \right\} \\ &= -\frac{1}{2} z_l^{(n)2} \left(1 + \sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl}^2 \right) + z_l^{(n)} \left(\sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl} \left(t_{ij}^{(m)} - a_j^{(m)} - \sum_{\substack{k=1 \\ k \neq l}}^K w_{jk} \mathbb{E}_{\neq z_l^{(n)}}[z_k^{(n)}] \right) \right) + \mathcal{C}' \end{aligned}$$

$1/\sigma_{nl}^2$

Back to MOTL

In the case of MOTL, what is $\mathbb{E}_{j \neq i}[\log(p(\mathbf{Z}, \mathbf{T}))]$?

$$\log(p(\mathbf{Z}, \mathbf{T})) = \log \left(\prod_{m=1}^M \prod_{i=1}^N \prod_{j=1}^{p_m} \mathcal{N} \left(a_j^{(m)} + \mathbf{w}_{j:} \mathbf{z}_i, 1/\tau_j^{(m)} \right) \prod_{i=1}^N \prod_{k=1}^K \mathcal{N}(0, 1) \right)$$

We recall that if $X \sim \mathcal{N}(\mu, \sigma^2)$, then:

$$\log(p(X = x)) = -\frac{1}{2} \log(2\pi) - \log(\sigma) - \frac{1}{2} \left(\frac{x - \mu}{\sigma} \right)^2$$

Hence

$$\begin{aligned} \log(p(\mathbf{Z}, \mathbf{T})) &= \sum_{m=1}^M \sum_{i=1}^N \sum_{j=1}^{p_m} \left[\frac{1}{2} \log(\tau_j^{(m)}) - \frac{1}{2} \log(2\pi) - \frac{\tau_j^{(m)}}{2} \left(t_{ij}^{(m)} - a_j^{(m)} - \mathbf{w}_{j:} \mathbf{z}_i \right)^2 \right] + \sum_{i=1}^N \sum_{k=1}^K \left[-\frac{1}{2} \log(2\pi) - \frac{1}{2} z_k^{(i)2} \right] \\ \mathbb{E}_{\neq z_l^{(n)}}[\log(p(\mathbf{Z}, \mathbf{T}))] &= \mathbb{E}_{\neq z_l^{(n)}} \left\{ \sum_{i=1}^N \sum_{j=1}^{p_m} \left[-\frac{\tau_j^{(m)}}{2} \left(t_{ij}^{(m)} - a_j^{(m)} - \sum_{\substack{k=1 \\ k \neq l}}^K w_{jk} z_k^{(n)} - w_{jl} z_l^{(n)} \right)^2 \right] - \frac{1}{2} z_l^{(n)2} + \mathcal{C} \right\} \\ &= \mathbb{E}_{\neq z_l^{(n)}} \left\{ -\frac{1}{2} z_l^{(n)2} \left(1 + \sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl}^2 \right) + z_l^{(n)} \left(\sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl} \left(t_{ij}^{(m)} - a_j^{(m)} - \sum_{\substack{k=1 \\ k \neq l}}^K w_{jk} z_k^{(n)} \right) \right) + \mathcal{C} \right\} \\ &= -\frac{1}{2} z_l^{(n)2} \left(1 + \sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl}^2 \right) + z_l^{(n)} \left(\sum_{i=1}^N \sum_{j=1}^{p_m} \tau_j^{(m)} w_{jl} \left(t_{ij}^{(m)} - a_j^{(m)} - \sum_{\substack{k=1 \\ k \neq l}}^K w_{jk} \mathbb{E}_{\neq z_l^{(n)}}[z_k^{(n)}] \right) \right) + \mathcal{C}' \end{aligned}$$

$\frac{1}{\sigma_{nl}^2}$
 μ_{nl}/σ_{nl}^2



Results:

Evaluation protocol using simulated multi-omics data.

Multi-omics data simulated with ground truth factors



Multi-omics data simulated with ground truth factors

Ground truth factors are generated such that:



Multi-omics data simulated with ground truth factors

Ground truth factors are generated such that:

- 30 instances of a multi-omics dataset, \mathbf{Y} split into a target dataset \mathbf{T} and a learning dataset \mathbf{L} .

Multi-omics data simulated with ground truth factors

Ground truth factors are generated such that:

- 30 instances of a multi-omics dataset, \mathbf{Y} split into a target dataset \mathbf{T} and a learning dataset \mathbf{L} .
- Considering $\mathbf{W}^{(m)} = \left[\mathbf{w}_1^{(m)\top}, \dots, \mathbf{w}_{p_m}^{(m)\top} \right]^\top \in \mathbb{R}^{D_m \times K}$; $K \in \{20; 30\}$, $D_m = 2000$,

Multi-omics data simulated with ground truth factors

Ground truth factors are generated such that:

- 30 instances of a multi-omics dataset, \mathbf{Y} split into a target dataset \mathbf{T} and a learning dataset \mathbf{L} .
- Considering $\mathbf{W}^{(m)} = \left[\mathbf{w}_1^{(m)\top}, \dots, \mathbf{w}_{p_m}^{(m)\top} \right]^\top \in \mathbb{R}^{D_m \times K}$; $K \in \{20; 30\}$, $D_m = 2000$,
 - $y_{id}^{(1)} \sim \mathcal{P} \left(\log(1 + e^{\mathbf{w}_d^{(1)\top} \mathbf{z}_i}) \right)$

Multi-omics data simulated with ground truth factors

Ground truth factors are generated such that:

- 30 instances of a multi-omics dataset, \mathbf{Y} split into a target dataset \mathbf{T} and a learning dataset \mathbf{L} .
- Considering $\mathbf{W}^{(m)} = [\mathbf{w}_1^{(m)T}, \dots, \mathbf{w}_{p_m}^{(m)T}]^T \in \mathbb{R}^{D_m \times K}$; $K \in \{20; 30\}$, $D_m = 2000$,
 - $y_{id}^{(1)} \sim \mathcal{P}(\log(1 + e^{\mathbf{w}_d^{(1)T} \mathbf{z}_i}))$
 - $y_{id}^{(2)} \sim \mathcal{N}(\mathbf{w}_d^{(2)T} \mathbf{z}_i, \sigma_d)$, where $\sigma_d \sim \mathcal{U}(0.25, 0.75)$

Multi-omics data simulated with ground truth factors

Ground truth factors are generated such that:

- 30 instances of a multi-omics dataset, \mathbf{Y} split into a target dataset \mathbf{T} and a learning dataset \mathbf{L} .
- Considering $\mathbf{W}^{(m)} = \left[\mathbf{w}_1^{(m)T}, \dots, \mathbf{w}_{p_m}^{(m)T} \right]^T \in \mathbb{R}^{D_m \times K}$; $K \in \{20; 30\}$, $D_m = 2000$,
 - $y_{id}^{(1)} \sim \mathcal{P} \left(\log(1 + e^{\mathbf{w}_d^{(1)T} \mathbf{z}_i}) \right)$
 - $y_{id}^{(2)} \sim \mathcal{N} \left(\mathbf{w}_d^{(2)T} \mathbf{z}_i, \sigma_d \right)$, where $\sigma_d \sim \mathcal{U}(0.25, 0.75)$
 - $y_{id}^{(3)} \sim \mathcal{B} \left(\frac{1}{1 + e^{-\mathbf{w}_d^{(3)T} \mathbf{z}_i}} \right)$

Multi-omics data simulated with ground truth factors

Ground truth factors are generated such that:

- 30 instances of a multi-omics dataset, \mathbf{Y} split into a target dataset \mathbf{T} and a learning dataset \mathbf{L} .
- Considering $\mathbf{W}^{(m)} = \left[\mathbf{w}_1^{(m)T}, \dots, \mathbf{w}_{p_m}^{(m)T} \right]^T \in \mathbb{R}^{D_m \times K}$; $K \in \{20; 30\}$, $D_m = 2000$,
 - $y_{id}^{(1)} \sim \mathcal{P} \left(\log(1 + e^{\mathbf{w}_d^{(1)T} \mathbf{z}_i}) \right)$
 - $y_{id}^{(2)} \sim \mathcal{N} \left(\mathbf{w}_d^{(2)T} \mathbf{z}_i, \sigma_d \right)$, where $\sigma_d \sim \mathcal{U}(0.25, 0.75)$
 - $y_{id}^{(3)} \sim \mathcal{B} \left(\frac{1}{1 + e^{-\mathbf{w}_d^{(3)T} \mathbf{z}_i}} \right)$
- With:

Multi-omics data simulated with ground truth factors

Ground truth factors are generated such that:

- 30 instances of a multi-omics dataset, \mathbf{Y} split into a target dataset \mathbf{T} and a learning dataset \mathbf{L} .
- Considering $\mathbf{W}^{(m)} = [\mathbf{w}_1^{(m)T}, \dots, \mathbf{w}_{p_m}^{(m)T}]^T \in \mathbb{R}^{D_m \times K}$; $K \in \{20; 30\}$, $D_m = 2000$,
 - $y_{id}^{(1)} \sim \mathcal{P}(\log(1 + e^{\mathbf{w}_d^{(1)T} \mathbf{z}_i}))$
 - $y_{id}^{(2)} \sim \mathcal{N}(\mathbf{w}_d^{(2)T} \mathbf{z}_i, \sigma_d)$, where $\sigma_d \sim \mathcal{U}(0.25, 0.75)$
 - $y_{id}^{(3)} \sim \mathcal{B}\left(\frac{1}{1 + e^{-\mathbf{w}_d^{(3)T} \mathbf{z}_i}}\right)$
- With:
 - $w_{kd}^{(m)} = \hat{w}_{kd}^{(m)} \times s_{kd}^{(m)}$, where

Multi-omics data simulated with ground truth factors

Ground truth factors are generated such that:

- 30 instances of a multi-omics dataset, \mathbf{Y} split into a target dataset \mathbf{T} and a learning dataset \mathbf{L} .

- Considering $\mathbf{W}^{(m)} = [\mathbf{w}_1^{(m)T}, \dots, \mathbf{w}_{p_m}^{(m)T}]^T \in \mathbb{R}^{D_m \times K}$; $K \in \{20; 30\}$, $D_m = 2000$,

- $y_{id}^{(1)} \sim \mathcal{P} \left(\log(1 + e^{\mathbf{w}_d^{(1)T} \mathbf{z}_i}) \right)$
- $y_{id}^{(2)} \sim \mathcal{N} \left(\mathbf{w}_d^{(2)T} \mathbf{z}_i, \sigma_d \right)$, where $\sigma_d \sim \mathcal{U}(0.25, 0.75)$
- $y_{id}^{(3)} \sim \mathcal{B} \left(\frac{1}{1 + e^{-\mathbf{w}_d^{(3)T} \mathbf{z}_i}} \right)$

- With:
 - $w_{kd}^{(m)} = \hat{w}_{kd}^{(m)} \times s_{kd}^{(m)}$, where $\left\{ \begin{array}{l} \hat{w}_{kd}^{(m)} \sim \mathcal{N} \left(\mu^{(m)}, \sigma_k^{(m)} \right) \text{ with} \end{array} \right.$

Multi-omics data simulated with ground truth factors

Ground truth factors are generated such that:

- 30 instances of a multi-omics dataset, \mathbf{Y} split into a target dataset \mathbf{T} and a learning dataset \mathbf{L} .
- Considering $\mathbf{W}^{(m)} = [\mathbf{w}_1^{(m)T}, \dots, \mathbf{w}_{p_m}^{(m)T}]^T \in \mathbb{R}^{D_m \times K}$; $K \in \{20; 30\}$, $D_m = 2000$,
 - $y_{id}^{(1)} \sim \mathcal{P}(\log(1 + e^{\mathbf{w}_d^{(1)T} \mathbf{z}_i}))$
 - $y_{id}^{(2)} \sim \mathcal{N}(\mathbf{w}_d^{(2)T} \mathbf{z}_i, \sigma_d)$, where $\sigma_d \sim \mathcal{U}(0.25, 0.75)$
 - $y_{id}^{(3)} \sim \mathcal{B}\left(\frac{1}{1 + e^{-\mathbf{w}_d^{(3)T} \mathbf{z}_i}}\right)$
- With:
 - $w_{kd}^{(m)} = \hat{w}_{kd}^{(m)} \times s_{kd}^{(m)}$, where $\left\{ \begin{array}{l} \hat{w}_{kd}^{(m)} \sim \mathcal{N}(\mu^{(m)}, \sigma_k^{(m)}) \text{ with } \mu^{(1)} = 5 \text{ and } \mu^{(2)} = \mu^{(3)} = 0 \end{array} \right.$

Multi-omics data simulated with ground truth factors

Ground truth factors are generated such that:

- 30 instances of a multi-omics dataset, \mathbf{Y} split into a target dataset \mathbf{T} and a learning dataset \mathbf{L} .
- Considering $\mathbf{W}^{(m)} = [\mathbf{w}_1^{(m)T}, \dots, \mathbf{w}_{p_m}^{(m)T}]^T \in \mathbb{R}^{D_m \times K}$; $K \in \{20; 30\}$, $D_m = 2000$,
 - $y_{id}^{(1)} \sim \mathcal{P}(\log(1 + e^{\mathbf{w}_d^{(1)T} \mathbf{z}_i}))$
 - $y_{id}^{(2)} \sim \mathcal{N}(\mathbf{w}_d^{(2)T} \mathbf{z}_i, \sigma_d)$, where $\sigma_d \sim \mathcal{U}(0.25, 0.75)$
 - $y_{id}^{(3)} \sim \mathcal{B}\left(\frac{1}{1 + e^{-\mathbf{w}_d^{(3)T} \mathbf{z}_i}}\right)$
- With:
 - $w_{kd}^{(m)} = \hat{w}_{kd}^{(m)} \times s_{kd}^{(m)}$, where $\left\{ \begin{array}{l} \hat{w}_{kd}^{(m)} \sim \mathcal{N}(\mu^{(m)}, \sigma_k^{(m)}) \text{ with } \begin{cases} \mu^{(1)} = 5 \text{ and } \mu^{(2)} = \mu^{(3)} = 0 \\ \sigma_k^{(1)}, \sigma_k^{(2)} \sim \mathcal{U}(0.5, 1.5) \text{ and } \sigma_k^{(3)} \sim \mathcal{U}(0.1, 0.2) \end{cases} \end{array} \right.$

Multi-omics data simulated with ground truth factors

Ground truth factors are generated such that:

- 30 instances of a multi-omics dataset, \mathbf{Y} split into a target dataset \mathbf{T} and a learning dataset \mathbf{L} .
- Considering $\mathbf{W}^{(m)} = [\mathbf{w}_1^{(m)T}, \dots, \mathbf{w}_{p_m}^{(m)T}]^T \in \mathbb{R}^{D_m \times K}$; $K \in \{20; 30\}$, $D_m = 2000$,
 - $y_{id}^{(1)} \sim \mathcal{P}(\log(1 + e^{\mathbf{w}_d^{(1)T} \mathbf{z}_i}))$
 - $y_{id}^{(2)} \sim \mathcal{N}(\mathbf{w}_d^{(2)T} \mathbf{z}_i, \sigma_d)$, where $\sigma_d \sim \mathcal{U}(0.25, 0.75)$
 - $y_{id}^{(3)} \sim \mathcal{B}\left(\frac{1}{1 + e^{-\mathbf{w}_d^{(3)T} \mathbf{z}_i}}\right)$
- With:
 - $w_{kd}^{(m)} = \hat{w}_{kd}^{(m)} \times s_{kd}^{(m)}$, where
$$\begin{cases} \hat{w}_{kd}^{(m)} \sim \mathcal{N}(\mu^{(m)}, \sigma_k^{(m)}) & \text{with } \begin{cases} \mu^{(1)} = 5 \text{ and } \mu^{(2)} = \mu^{(3)} = 0 \\ \sigma_k^{(1)}, \sigma_k^{(2)} \sim \mathcal{U}(0.5, 1.5) \text{ and } \sigma_k^{(3)} \sim \mathcal{U}(0.1, 0.2) \end{cases} \\ s_{kd}^{(m)} \sim \mathcal{B}(\theta_k^{(m)}) & \text{with } \theta_k^{(m)} \sim \mathcal{U}(0.15, 0.25) \end{cases}$$

Multi-omics data simulated with ground truth factors

Ground truth factors are generated such that:

- 30 instances of a multi-omics dataset, \mathbf{Y} split into a target dataset \mathbf{T} and a learning dataset \mathbf{L} .
- Considering $\mathbf{W}^{(m)} = [\mathbf{w}_1^{(m)T}, \dots, \mathbf{w}_{p_m}^{(m)T}]^T \in \mathbb{R}^{D_m \times K}$; $K \in \{20; 30\}$, $D_m = 2000$,
 - $y_{id}^{(1)} \sim \mathcal{P}(\log(1 + e^{\mathbf{w}_d^{(1)T} \mathbf{z}_i}))$
 - $y_{id}^{(2)} \sim \mathcal{N}(\mathbf{w}_d^{(2)T} \mathbf{z}_i, \sigma_d)$, where $\sigma_d \sim \mathcal{U}(0.25, 0.75)$
 - $y_{id}^{(3)} \sim \mathcal{B}\left(\frac{1}{1 + e^{-\mathbf{w}_d^{(3)T} \mathbf{z}_i}}\right)$
- With:
 - $w_{kd}^{(m)} = \hat{w}_{kd}^{(m)} \times s_{kd}^{(m)}$, where
$$\begin{cases} \hat{w}_{kd}^{(m)} \sim \mathcal{N}(\mu^{(m)}, \sigma_k^{(m)}) & \text{with } \begin{cases} \mu^{(1)} = 5 \text{ and } \mu^{(2)} = \mu^{(3)} = 0 \\ \sigma_k^{(1)}, \sigma_k^{(2)} \sim \mathcal{U}(0.5, 1.5) \text{ and } \sigma_k^{(3)} \sim \mathcal{U}(0.1, 0.2) \end{cases} \\ s_{kd}^{(m)} \sim \mathcal{B}(\theta_k^{(m)}) & \text{with } \theta_k^{(m)} \sim \mathcal{U}(0.15, 0.25) \end{cases}$$
 - $z_k^{(i)} \sim \mathcal{N}(\mu_{g(i)k}, \sigma_z)$, where

Multi-omics data simulated with ground truth factors

Ground truth factors are generated such that:

- 30 instances of a multi-omics dataset, \mathbf{Y} split into a target dataset \mathbf{T} and a learning dataset \mathbf{L} .
- Considering $\mathbf{W}^{(m)} = [\mathbf{w}_1^{(m)T}, \dots, \mathbf{w}_{p_m}^{(m)T}]^T \in \mathbb{R}^{D_m \times K}$; $K \in \{20; 30\}$, $D_m = 2000$,
 - $y_{id}^{(1)} \sim \mathcal{P}(\log(1 + e^{\mathbf{w}_d^{(1)T} \mathbf{z}_i}))$
 - $y_{id}^{(2)} \sim \mathcal{N}(\mathbf{w}_d^{(2)T} \mathbf{z}_i, \sigma_d)$, where $\sigma_d \sim \mathcal{U}(0.25, 0.75)$
 - $y_{id}^{(3)} \sim \mathcal{B}\left(\frac{1}{1 + e^{-\mathbf{w}_d^{(3)T} \mathbf{z}_i}}\right)$
- With:
 - $w_{kd}^{(m)} = \hat{w}_{kd}^{(m)} \times s_{kd}^{(m)}$, where
$$\begin{cases} \hat{w}_{kd}^{(m)} \sim \mathcal{N}(\mu^{(m)}, \sigma_k^{(m)}) \text{ with } \begin{cases} \mu^{(1)} = 5 \text{ and } \mu^{(2)} = \mu^{(3)} = 0 \\ \sigma_k^{(1)}, \sigma_k^{(2)} \sim \mathcal{U}(0.5, 1.5) \text{ and } \sigma_k^{(3)} \sim \mathcal{U}(0.1, 0.2) \end{cases} \\ s_{kd}^{(m)} \sim \mathcal{B}(\theta_k^{(m)}) \quad \text{with } \theta_k^{(m)} \sim \mathcal{U}(0.15, 0.25) \end{cases}$$
 - $z_k^{(i)} \sim \mathcal{N}(\mu_{g(i)k}, \sigma_z)$, where
$$\begin{cases} P(\mu_{g(i)k} = 3) = P(\mu_{g(i)k} = 7) = 1/8 \text{ and } P(\mu_{g(i)k} = 5) = 3/4 \end{cases}$$

Multi-omics data simulated with ground truth factors

Ground truth factors are generated such that:

- 30 instances of a multi-omics dataset, \mathbf{Y} split into a target dataset \mathbf{T} and a learning dataset \mathbf{L} .
- Considering $\mathbf{W}^{(m)} = [\mathbf{w}_1^{(m)T}, \dots, \mathbf{w}_{p_m}^{(m)T}]^T \in \mathbb{R}^{D_m \times K}$; $K \in \{20; 30\}$, $D_m = 2000$,
 - $y_{id}^{(1)} \sim \mathcal{P}(\log(1 + e^{\mathbf{w}_d^{(1)T} \mathbf{z}_i}))$
 - $y_{id}^{(2)} \sim \mathcal{N}(\mathbf{w}_d^{(2)T} \mathbf{z}_i, \sigma_d)$, where $\sigma_d \sim \mathcal{U}(0.25, 0.75)$
 - $y_{id}^{(3)} \sim \mathcal{B}\left(\frac{1}{1 + e^{-\mathbf{w}_d^{(3)T} \mathbf{z}_i}}\right)$
- With:
 - $w_{kd}^{(m)} = \hat{w}_{kd}^{(m)} \times s_{kd}^{(m)}$, where
$$\begin{cases} \hat{w}_{kd}^{(m)} \sim \mathcal{N}(\mu^{(m)}, \sigma_k^{(m)}) \text{ with } \begin{cases} \mu^{(1)} = 5 \text{ and } \mu^{(2)} = \mu^{(3)} = 0 \\ \sigma_k^{(1)}, \sigma_k^{(2)} \sim \mathcal{U}(0.5, 1.5) \text{ and } \sigma_k^{(3)} \sim \mathcal{U}(0.1, 0.2) \end{cases} \\ s_{kd}^{(m)} \sim \mathcal{B}(\theta_k^{(m)}) \quad \text{with } \theta_k^{(m)} \sim \mathcal{U}(0.15, 0.25) \end{cases}$$
 - $z_k^{(i)} \sim \mathcal{N}(\mu_{g(i)k}, \sigma_z)$, where
$$\begin{cases} P(\mu_{g(i)k} = 3) = P(\mu_{g(i)k} = 7) = 1/8 \text{ and } P(\mu_{g(i)k} = 5) = 3/4 \\ \sigma_z \in \{0.5; 1\} \end{cases}$$

Multi-omics data simulated with ground truth factors

Ground truth factors are generated such that:

- 30 instances of a multi-omics dataset, \mathbf{Y} split into a target dataset \mathbf{T} and a learning dataset \mathbf{L} .
- Considering $\mathbf{W}^{(m)} = [\mathbf{w}_1^{(m)T}, \dots, \mathbf{w}_{p_m}^{(m)T}]^T \in \mathbb{R}^{D_m \times K}$; $K \in \{20; 30\}$, $D_m = 2000$,
 - $y_{id}^{(1)} \sim \mathcal{P}(\log(1 + e^{\mathbf{w}_d^{(1)T} \mathbf{z}_i}))$
 - $y_{id}^{(2)} \sim \mathcal{N}(\mathbf{w}_d^{(2)T} \mathbf{z}_i, \sigma_d)$, where $\sigma_d \sim \mathcal{U}(0.25, 0.75)$
 - $y_{id}^{(3)} \sim \mathcal{B}\left(\frac{1}{1 + e^{-\mathbf{w}_d^{(3)T} \mathbf{z}_i}}\right)$
- With:
 - $w_{kd}^{(m)} = \hat{w}_{kd}^{(m)} \times s_{kd}^{(m)}$, where
$$\begin{cases} \hat{w}_{kd}^{(m)} \sim \mathcal{N}(\mu^{(m)}, \sigma_k^{(m)}) \text{ with } \begin{cases} \mu^{(1)} = 5 \text{ and } \mu^{(2)} = \mu^{(3)} = 0 \\ \sigma_k^{(1)}, \sigma_k^{(2)} \sim \mathcal{U}(0.5, 1.5) \text{ and } \sigma_k^{(3)} \sim \mathcal{U}(0.1, 0.2) \end{cases} \\ s_{kd}^{(m)} \sim \mathcal{B}(\theta_k^{(m)}) \quad \text{with } \theta_k^{(m)} \sim \mathcal{U}(0.15, 0.25) \end{cases}$$
 - $z_k^{(i)} \sim \mathcal{N}(\mu_{g(i)k}, \sigma_z)$, where
$$\begin{cases} P(\mu_{g(i)k} = 3) = P(\mu_{g(i)k} = 7) = 1/8 \text{ and } P(\mu_{g(i)k} = 5) = 3/4 \\ \sigma_z \in \{0.5; 1\} \end{cases}$$

Multi-omics data simulated with ground truth factors

Ground truth factors are generated such that:

- 30 instances of a multi-omics dataset, \mathbf{Y} split into a target dataset \mathbf{T} and a learning dataset \mathbf{L} .
 - Considering $\mathbf{W}^{(m)} = [\mathbf{w}_1^{(m)T}, \dots, \mathbf{w}_{p_m}^{(m)T}]^T \in \mathbb{R}^{D_m \times K}$; $K \in \{20; 30\}$, $D_m = 2000$,
 - $y_{id}^{(1)} \sim \mathcal{P}(\log(1 + e^{\mathbf{w}_d^{(1)T} \mathbf{z}_i}))$
 - $y_{id}^{(2)} \sim \mathcal{N}(\mathbf{w}_d^{(2)T} \mathbf{z}_i, \sigma_d)$, where $\sigma_d \sim \mathcal{U}(0.25, 0.75)$
 - $y_{id}^{(3)} \sim \mathcal{B}\left(\frac{1}{1 + e^{-\mathbf{w}_d^{(3)T} \mathbf{z}_i}}\right)$
 - With:
 - $w_{kd}^{(m)} = \hat{w}_{kd}^{(m)} \times s_{kd}^{(m)}$, where

$$\begin{cases} \hat{w}_{kd}^{(m)} \sim \mathcal{N}(\mu^{(m)}, \sigma_k^{(m)}) \text{ with } \begin{cases} \mu^{(1)} = 5 \text{ and } \mu^{(2)} = \mu^{(3)} = 0 \\ \sigma_k^{(1)}, \sigma_k^{(2)} \sim \mathcal{U}(0.5, 1.5) \text{ and } \sigma_k^{(3)} \sim \mathcal{U}(0.1, 0.2) \end{cases} \\ s_{kd}^{(m)} \sim \mathcal{B}(\theta_k^{(m)}) \quad \text{with } \theta_k^{(m)} \sim \mathcal{U}(0.15, 0.25) \end{cases}$$
 - $z_k^{(i)} \sim \mathcal{N}(\mu_{g(i)k}, \sigma_z)$, where

$$\begin{cases} P(\mu_{g(i)k} = 3) = P(\mu_{g(i)k} = 7) = 1/8 \text{ and } P(\mu_{g(i)k} = 5) = 3/4 \\ \sigma_z \in \{0.5; 1\} \end{cases}$$
- $N_{\text{target}} = 10$ with 2 groups of 5 samples, $N_{\text{learning}} \sim \begin{cases} 400 \text{ with 20 groups of size drawn in } \{10, 20, 30\} \end{cases}$

Multi-omics data simulated with ground truth factors

Ground truth factors are generated such that:

- 30 instances of a multi-omics dataset, \mathbf{Y} split into a target dataset \mathbf{T} and a learning dataset \mathbf{L} .
 - Considering $\mathbf{W}^{(m)} = [\mathbf{w}_1^{(m)T}, \dots, \mathbf{w}_{p_m}^{(m)T}]^T \in \mathbb{R}^{D_m \times K}$; $K \in \{20; 30\}$, $D_m = 2000$,
 - $y_{id}^{(1)} \sim \mathcal{P}(\log(1 + e^{\mathbf{w}_d^{(1)T} \mathbf{z}_i}))$
 - $y_{id}^{(2)} \sim \mathcal{N}(\mathbf{w}_d^{(2)T} \mathbf{z}_i, \sigma_d)$, where $\sigma_d \sim \mathcal{U}(0.25, 0.75)$
 - $y_{id}^{(3)} \sim \mathcal{B}\left(\frac{1}{1 + e^{-\mathbf{w}_d^{(3)T} \mathbf{z}_i}}\right)$
 - With:
 - $w_{kd}^{(m)} = \hat{w}_{kd}^{(m)} \times s_{kd}^{(m)}$, where

$$\begin{cases} \hat{w}_{kd}^{(m)} \sim \mathcal{N}(\mu^{(m)}, \sigma_k^{(m)}) \text{ with } \begin{cases} \mu^{(1)} = 5 \text{ and } \mu^{(2)} = \mu^{(3)} = 0 \\ \sigma_k^{(1)}, \sigma_k^{(2)} \sim \mathcal{U}(0.5, 1.5) \text{ and } \sigma_k^{(3)} \sim \mathcal{U}(0.1, 0.2) \end{cases} \\ s_{kd}^{(m)} \sim \mathcal{B}(\theta_k^{(m)}) \quad \text{with } \theta_k^{(m)} \sim \mathcal{U}(0.15, 0.25) \end{cases}$$
 - $z_k^{(i)} \sim \mathcal{N}(\mu_{g(i)k}, \sigma_z)$, where

$$\begin{cases} P(\mu_{g(i)k} = 3) = P(\mu_{g(i)k} = 7) = 1/8 \text{ and } P(\mu_{g(i)k} = 5) = 3/4 \\ \sigma_z \in \{0.5; 1\} \end{cases}$$
- $N_{\text{target}} = 10$ with 2 groups of 5 samples, $N_{\text{learning}} \sim \begin{cases} 400 \text{ with 20 groups of size drawn in } \{10, 20, 30\} \\ 1000 \text{ with 40 groups of size drawn in } \{10, 25, 40\} \end{cases}$

Pre-processing & Factorization



Pre-processing & Factorization

Pre-processing



Pre-processing & Factorization

Pre-processing

- MOFA: Remove null variance features from **L** and **T**.



Pre-processing & Factorization

Pre-processing

- MOFA: Remove null variance features from **L** and **T**.
- MOTL: Remove features from **T** that had been removed from corresponding **L**.

Pre-processing & Factorization

Pre-processing

- MOFA: Remove null variance features from **L** and **T**.
- MOTL: Remove features from **T** that had been removed from corresponding **L**.

Factorization



Pre-processing & Factorization

Pre-processing

- MOFA: Remove null variance features from **L** and **T**.
- MOTL: Remove features from **T** that had been removed from corresponding **L**.

Factorization

- MOFA to factorize **L** and **T**

Pre-processing & Factorization

Pre-processing

- MOFA: Remove null variance features from **L** and **T**.
- MOTL: Remove features from **T** that had been removed from corresponding **L**.

Factorization

- MOFA to factorize **L** and **T**
 - $\hat{K} = \min(N, K_{diff})$ où K_{diff} is the number of ground truth factors differentially active during simulation.

Pre-processing & Factorization

Pre-processing

- MOFA: Remove null variance features from **L** and **T**.
- MOTL: Remove features from **T** that had been removed from corresponding **L**.

Factorization

- MOFA to factorize **L** and **T**
 - $\hat{K} = \min(N, K_{diff})$ où K_{diff} is the number of ground truth factors differentially active during simulation.
 - Same likelihoods as the ones for simulation

Pre-processing & Factorization

Pre-processing

- MOFA: Remove null variance features from **L** and **T**.
- MOTL: Remove features from **T** that had been removed from corresponding **L**.

Factorization

- MOFA to factorize **L** and **T**
 - $\hat{K} = \min(N, K_{diff})$ où K_{diff} is the number of ground truth factors differentially active during simulation.
 - Same likelihoods as the ones for simulation
 - 10.000 iterations to ensure convergence

Pre-processing & Factorization

Pre-processing

- MOFA: Remove null variance features from **L** and **T**.
- MOTL: Remove features from **T** that had been removed from corresponding **L**.

Factorization

- MOFA to factorize **L** and **T**
 - $\hat{K} = \min(N, K_{diff})$ où K_{diff} is the number of ground truth factors differentially active during simulation.
 - Same likelihoods as the ones for simulation
 - 10.000 iterations to ensure convergence
 - Other parameters to default.

Pre-processing & Factorization

Pre-processing

- MOFA: Remove null variance features from **L** and **T**.
- MOTL: Remove features from **T** that had been removed from corresponding **L**.

Factorization

- MOFA to factorize **L** and **T**
 - $\hat{K} = \min(N, K_{diff})$ où K_{diff} is the number of ground truth factors differentially active during simulation.
 - Same likelihoods as the ones for simulation
 - 10.000 iterations to ensure convergence
 - Other parameters to default.
- MOTL to factorize **T**

Pre-processing & Factorization

Pre-processing

- MOFA: Remove null variance features from **L** and **T**.
- MOTL: Remove features from **T** that had been removed from corresponding **L**.

Factorization

- MOFA to factorize **L** and **T**
 - $\hat{K} = \min(N, K_{diff})$ où K_{diff} is the number of ground truth factors differentially active during simulation.
 - Same likelihoods as the ones for simulation
 - 10.000 iterations to ensure convergence
 - Other parameters to default.
- MOTL to factorize **T**
 - Get weight and precision values from MOFA models. Precision values are held fixe for Gaussian and Poisson data. For Bernouilli, average of precision parameters across samples to initialize and then updates **following pseudo-Gaussian strategy**. **Intercepts**

Pre-processing & Factorization

Pre-processing

- MOFA: Remove null variance features from **L** and **T**.
- MOTL: Remove features from **T** that had been removed from corresponding **L**.

Factorization

- MOFA to factorize **L** and **T**
 - $\hat{K} = \min(N, K_{diff})$ où K_{diff} is the number of ground truth factors differentially active during simulation.
 - Same likelihoods as the ones for simulation
 - 10.000 iterations to ensure convergence
 - Other parameters to default.
- MOTL to factorize **T**
 - Get weight and precision values from MOFA models. Precision values are held fixe for Gaussian and Poisson data. For Bernouilli, average of precision parameters across samples to initialize and then updates **following pseudo-Gaussian strategy. Intercepts**
 - The algorithm was stopped when the absolute change in ELBO was under 0.0005% (default MOFA) for two consecutive checks over a maximum of 10.000 iterations.

Pre-processing & Factorization

Pre-processing

- MOFA: Remove null variance features from **L** and **T**.
- MOTL: Remove features from **T** that had been removed from corresponding **L**.

Factorization

- MOFA to factorize **L** and **T**
 - $\hat{K} = \min(N, K_{diff})$ où K_{diff} is the number of ground truth factors differentially active during simulation.
 - Same likelihoods as the ones for simulation
 - 10.000 iterations to ensure convergence
 - Other parameters to default.
- MOTL to factorize **T**
 - Get weight and precision values from MOFA models. Precision values are held fixe for Gaussian and Poisson data. For Bernouilli, average of precision parameters across samples to initialize and then updates **following pseudo-Gaussian strategy. Intercepts**
 - The algorithm was stopped when the absolute change in ELBO was under 0.0005% (default MOFA) for two consecutive checks over a maximum of 10.000 iterations.

Post-processing, for each inferred of ground truth $\mathbf{W}^{(m)} \in \mathbb{R}^{p_m \times K}$:

Pre-processing & Factorization

Pre-processing

- MOFA: Remove null variance features from **L** and **T**.
- MOTL: Remove features from **T** that had been removed from corresponding **L**.

Factorization

- MOFA to factorize **L** and **T**
 - $\hat{K} = \min(N, K_{diff})$ où K_{diff} is the number of ground truth factors differentially active during simulation.
 - Same likelihoods as the ones for simulation
 - 10.000 iterations to ensure convergence
 - Other parameters to default.
- MOTL to factorize **T**
 - Get weight and precision values from MOFA models. Precision values are held fixe for Gaussian and Poisson data. For Bernoulli, average of precision parameters across samples to initialize and then updates **following pseudo-Gaussian strategy. Intercepts**
 - The algorithm was stopped when the absolute change in ELBO was under 0.0005% (default MOFA) for two consecutive checks over a maximum of 10.000 iterations.

Post-processing, for each inferred of ground truth $\mathbf{W}^{(m)} \in \mathbb{R}^{p_m \times K}$:

- Each $\mathbf{w}_{d:}^{(m)}$ is scaled by its Frobenius norm.

Pre-processing & Factorization

Pre-processing

- MOFA: Remove null variance features from **L** and **T**.
- MOTL: Remove features from **T** that had been removed from corresponding **L**.

Factorization

- MOFA to factorize **L** and **T**
 - $\hat{K} = \min(N, K_{diff})$ où K_{diff} is the number of ground truth factors differentially active during simulation.
 - Same likelihoods as the ones for simulation
 - 10.000 iterations to ensure convergence
 - Other parameters to default.
- MOTL to factorize **T**
 - Get weight and precision values from MOFA models. Precision values are held fixe for Gaussian and Poisson data. For Bernoulli, average of precision parameters across samples to initialize and then updates **following pseudo-Gaussian strategy. Intercepts**
 - The algorithm was stopped when the absolute change in ELBO was under 0.0005% (default MOFA) for two consecutive checks over a maximum of 10.000 iterations.

Post-processing, for each inferred of ground truth $\mathbf{W}^{(m)} \in \mathbb{R}^{p_m \times K}$:

- Each $\mathbf{w}_{d:}^{(m)}$ is scaled by its Frobenius norm.
- Each $\mathbf{w}_{:k}^{(m)}$ is centered.

Pre-processing & Factorization

Pre-processing

- MOFA: Remove null variance features from **L** and **T**.
- MOTL: Remove features from **T** that had been removed from corresponding **L**.

Factorization

- MOFA to factorize **L** and **T**
 - $\hat{K} = \min(N, K_{diff})$ où K_{diff} is the number of ground truth factors differentially active during simulation.
 - Same likelihoods as the ones for simulation
 - 10.000 iterations to ensure convergence
 - Other parameters to default.
- MOTL to factorize **T**
 - Get weight and precision values from MOFA models. Precision values are held fixe for Gaussian and Poisson data. For Bernoulli, average of precision parameters across samples to initialize and then updates **following pseudo-Gaussian strategy. Intercepts**
 - The algorithm was stopped when the absolute change in ELBO was under 0.0005% (default MOFA) for two consecutive checks over a maximum of 10.000 iterations.

Post-processing, for each inferred of ground truth $\mathbf{W}^{(m)} \in \mathbb{R}^{p_m \times K}$:

- Each $\mathbf{w}_{d:}^{(m)}$ is scaled by its Frobenius norm.
- Each $\mathbf{w}_{:k}^{(m)}$ is centered.
- $\mathbf{w}_{:k}^{(m)}$ are concatenated across views.

Metrics – F1-score



Metrics – F1-score

F1 score for differentially active factors on the Target dataset

Metrics – F1-score

F1 score for differentially active factors on the Target dataset

- A factor $\hat{\mathbf{z}}_k \in \mathbb{R}^{N_{target}}$ estimated on \mathbf{T} with weights $\hat{\mathbf{w}}_k = \left[\hat{\mathbf{w}}_k^{(1)\top}, \hat{\mathbf{w}}_k^{(2)\top}, \hat{\mathbf{w}}_k^{(3)\top} \right]^\top \in \mathbb{R}^{3D_m}$ is considered as differentially active if:

Metrics – F1-score

F1 score for differentially active factors on the Target dataset

- A factor $\hat{\mathbf{z}}_k \in \mathbb{R}^{N_{target}}$ estimated on \mathbf{T} with weights $\hat{\mathbf{w}}_k = \left[\hat{\mathbf{w}}_k^{(1)\top}, \hat{\mathbf{w}}_k^{(2)\top}, \hat{\mathbf{w}}_k^{(3)\top} \right]^\top \in \mathbb{R}^{3D_m}$ is considered as differentially active if:
 - $p_{val} \leq 0.05$ from Wilcoxon rank sum test (2 groups) or Kruskal-Wallis test (more than 2 groups) BH-adjusted.

Metrics – F1-score

F1 score for differentially active factors on the Target dataset

- A factor $\hat{\mathbf{z}}_k \in \mathbb{R}^{N_{target}}$ estimated on \mathbf{T} with weights $\hat{\mathbf{w}}_k = \left[\hat{\mathbf{w}}_k^{(1)\top}, \hat{\mathbf{w}}_k^{(2)\top}, \hat{\mathbf{w}}_k^{(3)\top} \right]^\top \in \mathbb{R}^{3D_m}$ is considered as differentially active if:
 - $p_{val} \leq 0.05$ from Wilcoxon rank sum test (2 groups) or Kruskal-Wallis test (more than 2 groups) BH-adjusted.
 - If significant, $BestHit(\hat{\mathbf{z}}_k) = \underset{l}{\operatorname{argmax}} cor(\hat{\mathbf{w}}_k, \mathbf{w}_l)$, with \mathbf{w}_l a ground truth weight vector of \mathbf{T} .

Metrics – F1-score

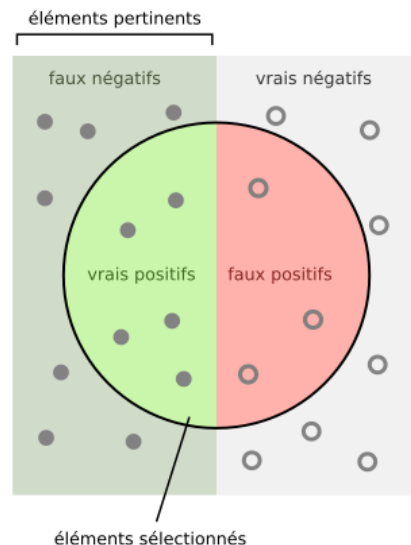
F1 score for differentially active factors on the Target dataset

- A factor $\hat{\mathbf{z}}_k \in \mathbb{R}^{N_{target}}$ estimated on \mathbf{T} with weights $\hat{\mathbf{w}}_k = \left[\hat{\mathbf{w}}_k^{(1)\top}, \hat{\mathbf{w}}_k^{(2)\top}, \hat{\mathbf{w}}_k^{(3)\top} \right]^\top \in \mathbb{R}^{3D_m}$ is considered as differentially active if:
 - $p_{val} \leq 0.05$ from Wilcoxon rank sum test (2 groups) or Kruskal-Wallis test (more than 2 groups) BH-adjusted.
 - If significant, $BestHit(\hat{\mathbf{z}}_k) = \underset{l}{\operatorname{argmax}} cor(\hat{\mathbf{w}}_k, \mathbf{w}_l)$, with \mathbf{w}_l a ground truth weight vector of \mathbf{T} .
 - If $BestHit(\hat{\mathbf{z}}_k)$ is a ground truth differentially active factor on \mathbf{T} , then this ground truth factor is considered as predicted differentially active on \mathbf{T} .

Metrics – F1-score

F1 score for differentially active factors on the Target dataset

- A factor $\hat{\mathbf{z}}_k \in \mathbb{R}^{N_{target}}$ estimated on \mathbf{T} with weights $\hat{\mathbf{w}}_k = \left[\hat{\mathbf{w}}_k^{(1)\top}, \hat{\mathbf{w}}_k^{(2)\top}, \hat{\mathbf{w}}_k^{(3)\top} \right]^\top \in \mathbb{R}^{3D_m}$ is considered as differentially active if:
 - $p_{val} \leq 0.05$ from Wilcoxon rank sum test (2 groups) or Kruskal-Wallis test (more than 2 groups) BH-adjusted.
 - If significant, $BestHit(\hat{\mathbf{z}}_k) = \underset{l}{\operatorname{argmax}} cor(\hat{\mathbf{w}}_k, \mathbf{w}_l)$, with \mathbf{w}_l a ground truth weight vector of \mathbf{T} .
 - If $BestHit(\hat{\mathbf{z}}_k)$ is a ground truth differentially active factor on \mathbf{T} , then this ground truth factor is considered as predicted differentially active on \mathbf{T} .



Combien de candidats sélectionnés sont pertinents ?

$$\text{Précision} = \frac{\text{vrais positifs}}{\text{vrais positifs} + \text{faux positifs}}$$

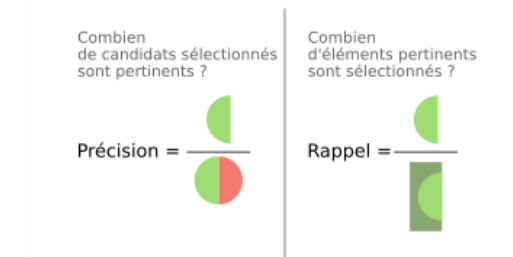
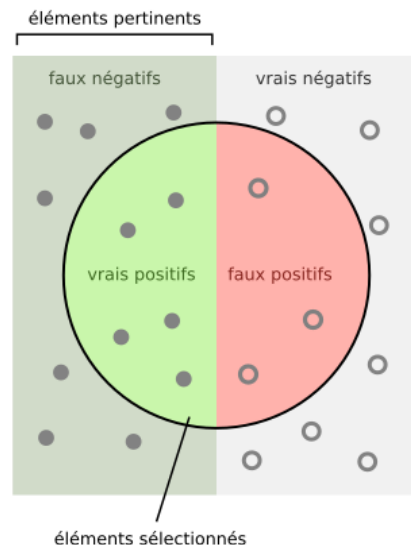
Combien d'éléments pertinents sont sélectionnés ?

$$\text{Rappel} = \frac{\text{vrais positifs}}{\text{vrais positifs} + \text{faux négatifs}}$$

Metrics – F1-score

F1 score for differentially active factors on the Target dataset

- A factor $\hat{\mathbf{z}}_k \in \mathbb{R}^{N_{target}}$ estimated on \mathbf{T} with weights $\hat{\mathbf{w}}_k = \left[\hat{\mathbf{w}}_k^{(1)\top}, \hat{\mathbf{w}}_k^{(2)\top}, \hat{\mathbf{w}}_k^{(3)\top} \right]^\top \in \mathbb{R}^{3D_m}$ is considered as differentially active if:
 - $p_{val} \leq 0.05$ from Wilcoxon rank sum test (2 groups) or Kruskal-Wallis test (more than 2 groups) BH-adjusted.
 - If significant, $BestHit(\hat{\mathbf{z}}_k) = \underset{l}{\operatorname{argmax}} cor(\hat{\mathbf{w}}_k, \mathbf{w}_l)$, with \mathbf{w}_l a ground truth weight vector of \mathbf{T} .
 - If $BestHit(\hat{\mathbf{z}}_k)$ is a ground truth differentially active factor on \mathbf{T} , then this ground truth factor is considered as predicted differentially active on \mathbf{T} .



F1 score \rightarrow Harmonic mean between precision and recall.

Metrics – F-measure



Metrics – F-measure

If $\{\mathbf{v}_1, \dots, \mathbf{v}_{K_v}\}$ is a set of inferred vectors and $\{\mathbf{x}_1, \dots, \mathbf{x}_{K_x}\}$ a set of ground truth vectors, then:

Metrics – F-measure

If $\{\mathbf{v}_1, \dots, \mathbf{v}_{K_v}\}$ is a set of inferred vectors and $\{\mathbf{x}_1, \dots, \mathbf{x}_{K_x}\}$ a set of ground truth vectors, then:

$$FM = \frac{2}{\frac{1}{Relevance} + \frac{1}{Recovery}}$$

Metrics – F-measure

If $\{\mathbf{v}_1, \dots, \mathbf{v}_{K_v}\}$ is a set of inferred vectors and $\{\mathbf{x}_1, \dots, \mathbf{x}_{K_x}\}$ a set of ground truth vectors, then:

$$FM = \frac{2}{\frac{1}{Relevance} + \frac{1}{Recovery}}$$

Where:

Metrics – F-measure

If $\{\mathbf{v}_1, \dots, \mathbf{v}_{K_v}\}$ is a set of inferred vectors and $\{\mathbf{x}_1, \dots, \mathbf{x}_{K_x}\}$ a set of ground truth vectors, then:

$$FM = \frac{2}{\frac{1}{Relevance} + \frac{1}{Recovery}}$$

Where:

$$Relevance = \frac{1}{K_v} \sum_{k_v=1}^{K_v} cor(\mathbf{v}_{k_v}, BestHit(\mathbf{v}_{k_v}))$$

Metrics – F-measure

If $\{\mathbf{v}_1, \dots, \mathbf{v}_{K_v}\}$ is a set of inferred vectors and $\{\mathbf{x}_1, \dots, \mathbf{x}_{K_x}\}$ a set of ground truth vectors, then:

$$FM = \frac{2}{\frac{1}{Relevance} + \frac{1}{Recovery}}$$

Where:

$$Relevance = \frac{1}{K_v} \sum_{k_v=1}^{K_v} cor(\mathbf{v}_{k_v}, BestHit(\mathbf{v}_{k_v}))$$

and:

Metrics – F-measure

If $\{\mathbf{v}_1, \dots, \mathbf{v}_{K_v}\}$ is a set of inferred vectors and $\{\mathbf{x}_1, \dots, \mathbf{x}_{K_x}\}$ a set of ground truth vectors, then:

$$FM = \frac{2}{\frac{1}{Relevance} + \frac{1}{Recovery}}$$

Where:

$$Relevance = \frac{1}{K_v} \sum_{k_v=1}^{K_v} cor(\mathbf{v}_{k_v}, BestHit(\mathbf{v}_{k_v}))$$

and:

$$Recovery = \frac{1}{K_x} \sum_{k_x=1}^{K_x} cor(\mathbf{x}_{k_x}, BestHit(\mathbf{x}_{k_x}))$$

Statistical Testing of differences between factorization models



Statistical Testing of differences between factorization models

A single regression is fit to model the F1 scores for simulated data

Statistical Testing of differences between factorization models

A single regression is fit to model the F1 scores for simulated data

$$y_i = \beta_0 + d_i \beta_d + f_i \beta_f + \epsilon_i.$$

Statistical Testing of differences between factorization models

A single regression is fit to model the F1 scores for simulated data

$$y_i = \beta_0 + d_i \beta_d + f_i \beta_f + \epsilon_i.$$

With:

Statistical Testing of differences between factorization models

A single regression is fit to model the F1 scores for simulated data

$$y_i = \beta_0 + \mathbf{d}_i \boldsymbol{\beta}_d + \mathbf{f}_i \boldsymbol{\beta}_f + \epsilon_i.$$

With:

$\mathbf{d}_i = (d_{i1}, \dots, d_{iT})$ indicates the simulation configuration

Statistical Testing of differences between factorization models

A single regression is fit to model the F1 scores for simulated data

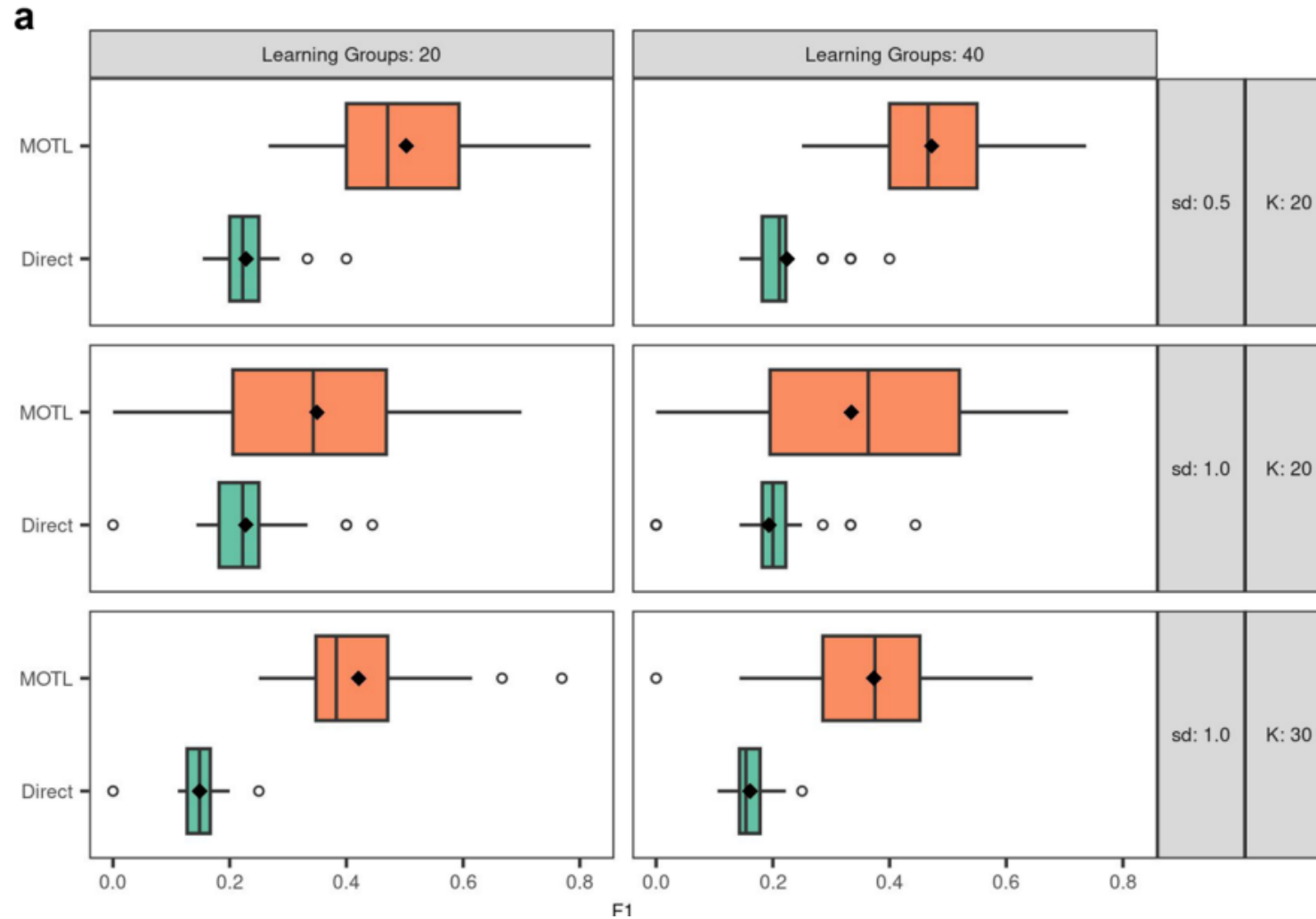
$$y_i = \beta_0 + \mathbf{d}_i \boldsymbol{\beta}_d + \mathbf{f}_i \boldsymbol{\beta}_f + \epsilon_i.$$

With:

$\mathbf{d}_i = (d_{i1}, \dots, d_{iT})$ indicates the simulation configuration

$\mathbf{f}_i = (f_{i1}, \dots, f_{iM})$ indicates the factorization method.

Results-Simulations



Results – Robustness evaluation



Results – Robustness evaluation

For each view and each proportion $p \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$



Results – Robustness evaluation

For each view and each proportion $p \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$

- Randomly select a $p \times D_m$ features.

Results – Robustness evaluation

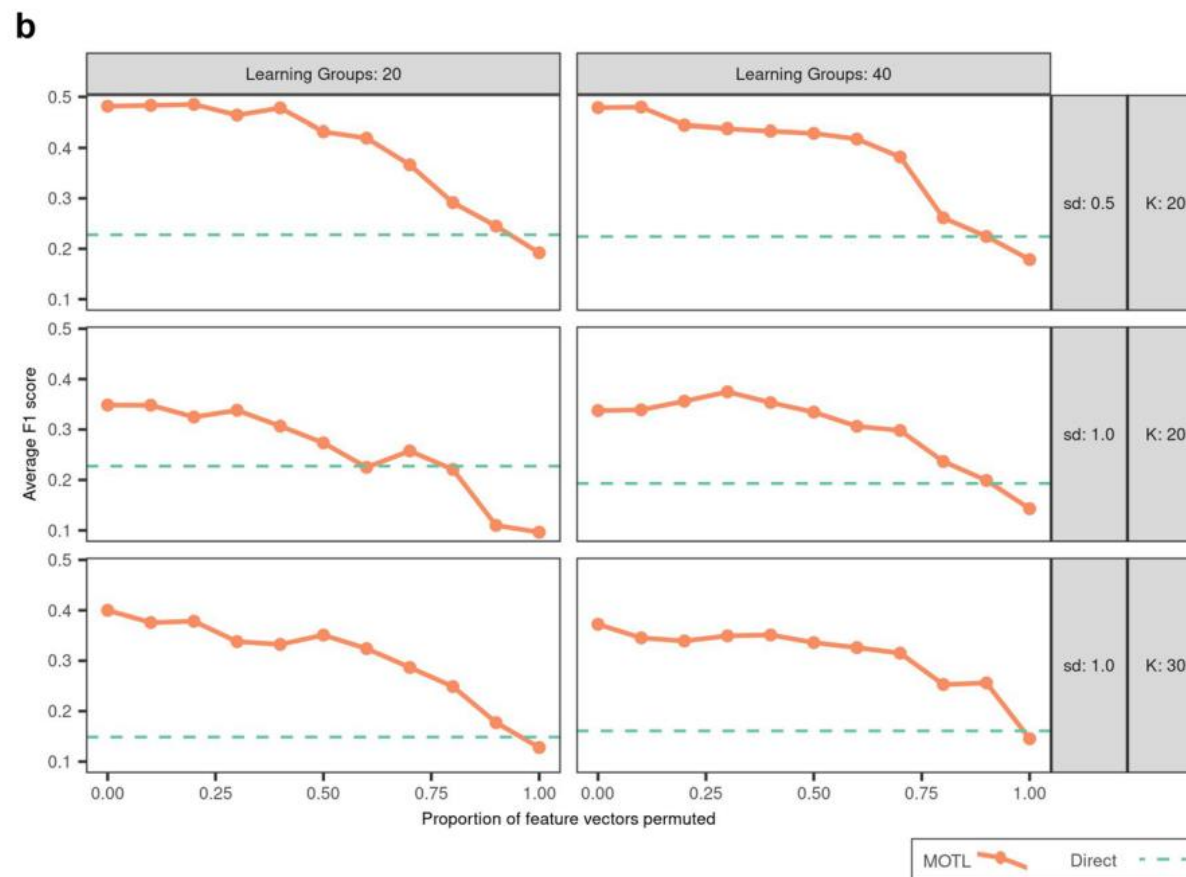
For each view and each proportion $p \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$

- Randomly select a $p \times D_m$ features.
- For each of these features j , randomly permute the elements of $\mathbf{w}_{j:}^{(m)} \in \mathbb{R}^K$ (where $\mathbf{w}_{j:}^{(m)}$ is inferred on \mathbf{L}).

Results – Robustness evaluation

For each view and each proportion $p \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$

- Randomly select a $p \times D_m$ features.
- For each of these features j , randomly permute the elements of $\mathbf{w}_{j:}^{(m)} \in \mathbb{R}^K$ (where $\mathbf{w}_{j:}^{(m)}$ is inferred on \mathbf{L}).



Results – Additional methods 1/2



Results – Additional methods 1/2

Additional factorization of \mathbf{T} , after removing features with null variance, with:

Results – Additional methods 1/2

Additional factorization of \mathbf{T} , after removing features with null variance, with:

IntNMF

Results – Additional methods 1/2

Additional factorization of \mathbf{T} , after removing features with null variance, with:

IntNMF

$$Q = \min_{W, H} \sum_{i=1}^m \theta^i \|X^i - WH^i\|_2.$$

Results – Additional methods 1/2

Additional factorization of \mathbf{T} , after removing features with null variance, with:

IntNMF

$$Q = \min_{W, H} \sum_{i=1}^m \theta^i \|X^i - WH^i\|_2.$$

$$\left(\theta^i = \frac{\text{Max} \{ \text{mean}\{\|X^i\|_2\}, i=1, \dots, m \}}{\text{mean}\{\|X^i\|_2\}}, i = 1, \dots, m \right)$$

Results – Additional methods 1/2

Additional factorization of \mathbf{T} , after removing features with null variance, with:

IntNMF

$$Q = \min_{W, H} \sum_{i=1}^m \theta^i \|X^i - WH^i\|_2.$$

$$\left(\theta^i = \frac{\text{Max} \{ \text{mean}\{\|X^i\|_2\}, i=1, \dots, m \}}{\text{mean}\{\|X^i\|_2\}}, i = 1, \dots, m \right)$$

Subtract the minimum value for each feature vector from all the values in that vector.

Use of nmf.mnnals from InterSim

Results – Additional methods 1/2

Additional factorization of \mathbf{T} , after removing features with null variance, with:

IntNMF

$$Q = \min_{W, H} \sum_{i=1}^m \theta^i \|X^i - WH^i\|_2.$$

$$\left(\theta^i = \frac{\text{Max} \{ \text{mean}\{\|X^i\|_2\}, i=1, \dots, m \}}{\text{mean}\{\|X^i\|_2\}}, i = 1, \dots, m \right)$$

Subtract the minimum value for each feature vector from all the values in that vector.

Use of nmf.mnnals from InterSim

moCluster → modified CPCA

Results – Additional methods 1/2

Additional factorization of \mathbf{T} , after removing features with null variance, with:

IntNMF

$$Q = \min_{W,H} \sum_{i=1}^m \theta^i \|\mathbf{X}^i - \mathbf{W}\mathbf{H}^i\|_2.$$

$$\left(\theta^i = \frac{\text{Max} \{ \text{mean}\{\|\mathbf{X}^i\|_2\}, i=1,\dots,m \}}{\text{mean}\{\|\mathbf{X}^i\|_2\}}, i = 1, \dots, m \right)$$

Subtract the minimum value for each feature vector from all the values in that vector.

Use of nmf.mnnals from InterSim

moCluster → modified CPCA

$$\underset{\mathbf{w}_1, \dots, \mathbf{w}_B, \mathbf{y}_{B+1}}{\text{Maximize}} \sum_{b=1}^B (\text{cov}(\mathbf{X}_b \mathbf{w}_b, \mathbf{y}_{B+1}))^m$$

$$\text{s.t. } \mathbf{w}_b^t \mathbf{M}_b \mathbf{w}_b = 1, b = 1, \dots, B, \text{ and } \text{var}(\mathbf{y}_{B+1}) = 1$$

Results – Additional methods 1/2

Additional factorization of \mathbf{T} , after removing features with null variance, with:

IntNMF

$$Q = \min_{W, H} \sum_{i=1}^m \theta^i \|X^i - WH^i\|_2.$$

$$\left(\theta^i = \frac{\text{Max} \{ \text{mean}\{\|X^i\|_2\}, i=1, \dots, m \}}{\text{mean}\{\|X^i\|_2\}}, i = 1, \dots, m \right)$$

Subtract the minimum value for each feature vector from all the values in that vector.

Use of nmf.mnnals from InterSim

moCluster → modified CPCA

$$\text{Maximize}_{\mathbf{w}_1, \dots, \mathbf{w}_B, \mathbf{y}_{B+1}} \sum_{b=1}^B (\text{cov}(\mathbf{X}_b \mathbf{w}_b, \mathbf{y}_{B+1}))^m$$

$$\text{s.t. } \mathbf{w}_b^t \mathbf{M}_b \mathbf{w}_b = 1, b = 1, \dots, B, \text{ and } \text{var}(\mathbf{y}_{B+1}) = 1$$

With $\mathbf{M}_b = \mathbf{I}$ and $m = 1$.

Soft thresholding on \mathbf{w}_b

Results – Additional methods 1/2

Additional factorization of \mathbf{T} , after removing features with null variance, with:

IntNMF

$$Q = \min_{W,H} \sum_{i=1}^m \theta^i \|\mathbf{X}^i - \mathbf{W}\mathbf{H}^i\|_2.$$

$$\left(\theta^i = \frac{\text{Max} \{ \text{mean}\{\|\mathbf{X}^i\|_2\}, i=1,\dots,m \}}{\text{mean}\{\|\mathbf{X}^i\|_2\}}, i = 1, \dots, m \right)$$

Subtract the minimum value for each feature vector from all the values in that vector.

Use of nmf.mnnals from InterSim

moCluster → modified CPCA

$$\text{Maximize}_{\mathbf{w}_1, \dots, \mathbf{w}_B, \mathbf{y}_{B+1}} \sum_{b=1}^B (\text{cov}(\mathbf{X}_b \mathbf{w}_b, \mathbf{y}_{B+1}))^m$$

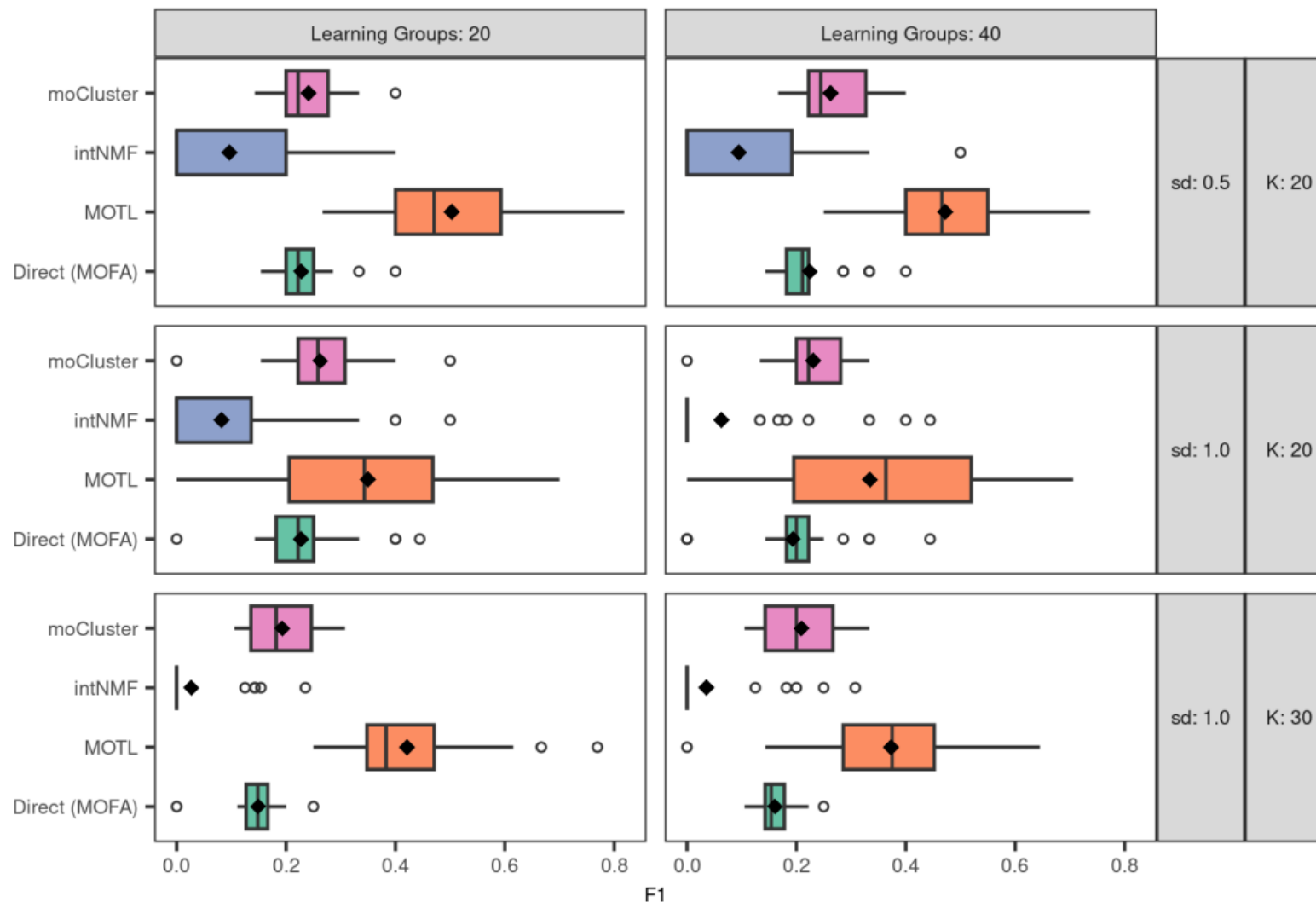
$$\text{s.t. } \mathbf{w}_b^t \mathbf{M}_b \mathbf{w}_b = 1, b = 1, \dots, B, \text{ and } \text{var}(\mathbf{y}_{B+1}) = 1$$

With $\mathbf{M}_b = \mathbf{I}$ and $m = 1$.

Soft thresholding on \mathbf{w}_b

$\hat{K} = \min(N, K_{diff})$ où K_{diff} is the number of ground truth factors differentially active during simulation.

Results – Additional methods 2/2





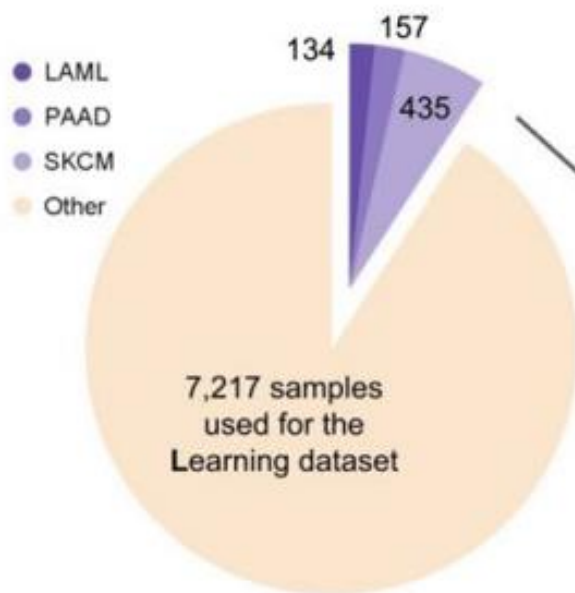
Results:

Evaluation protocol using TCGA multi-omics data.

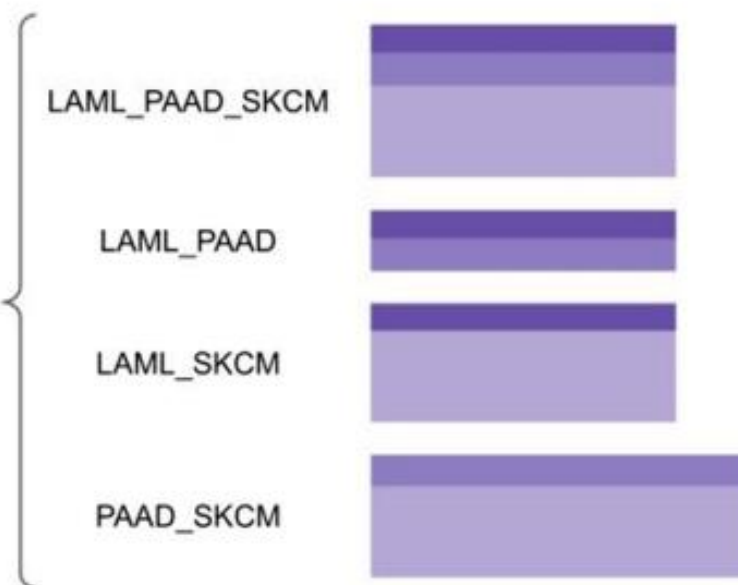
TCGA data - Pancancer

a

TCGA samples with full multi-omics profiles

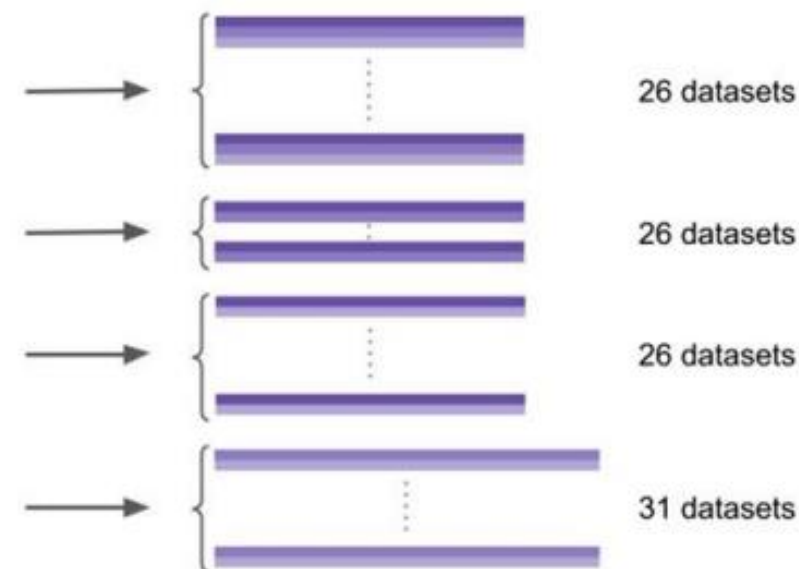


Reference datasets



5 samples per cancer type

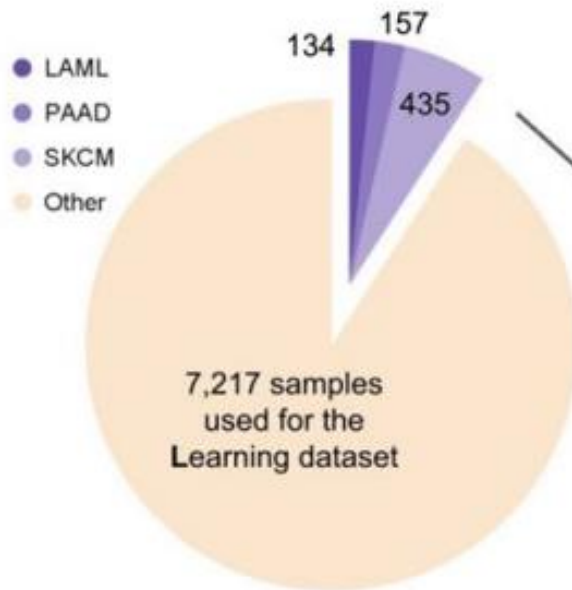
Target datasets



TCGA data - Pancancer

a

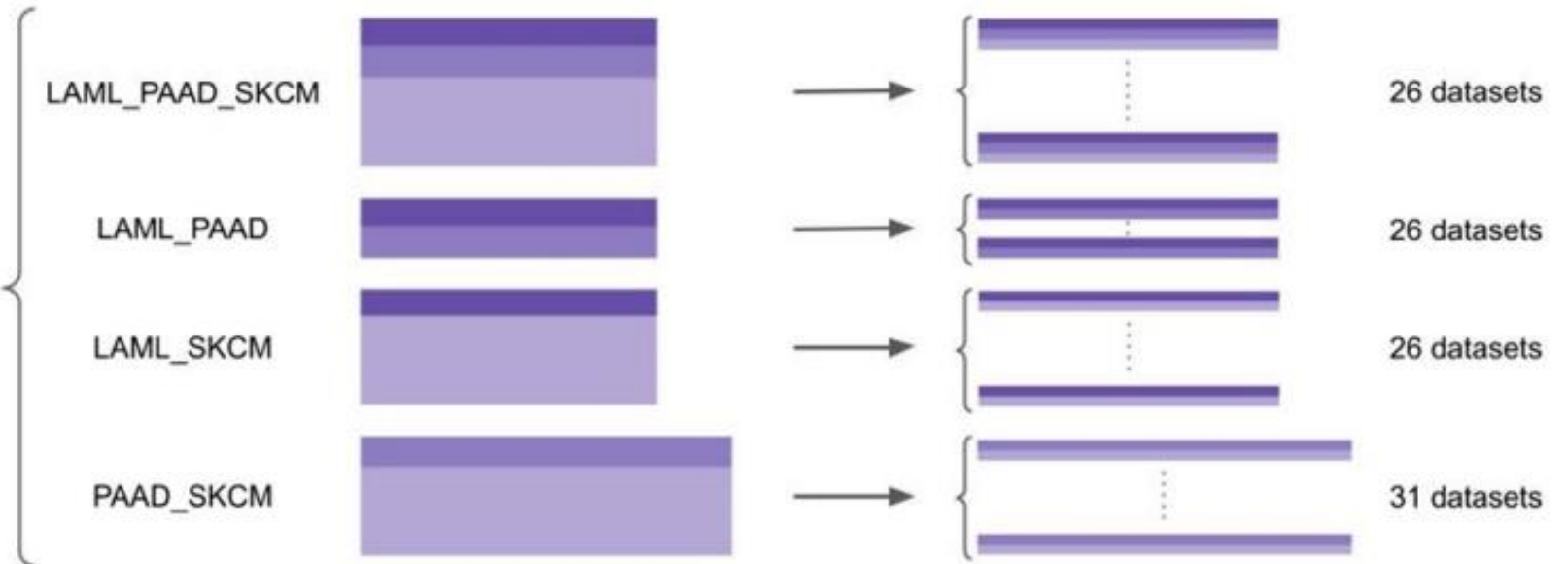
TCGA samples with full multi-omics profiles



Reference datasets

5 samples per cancer type

Target datasets

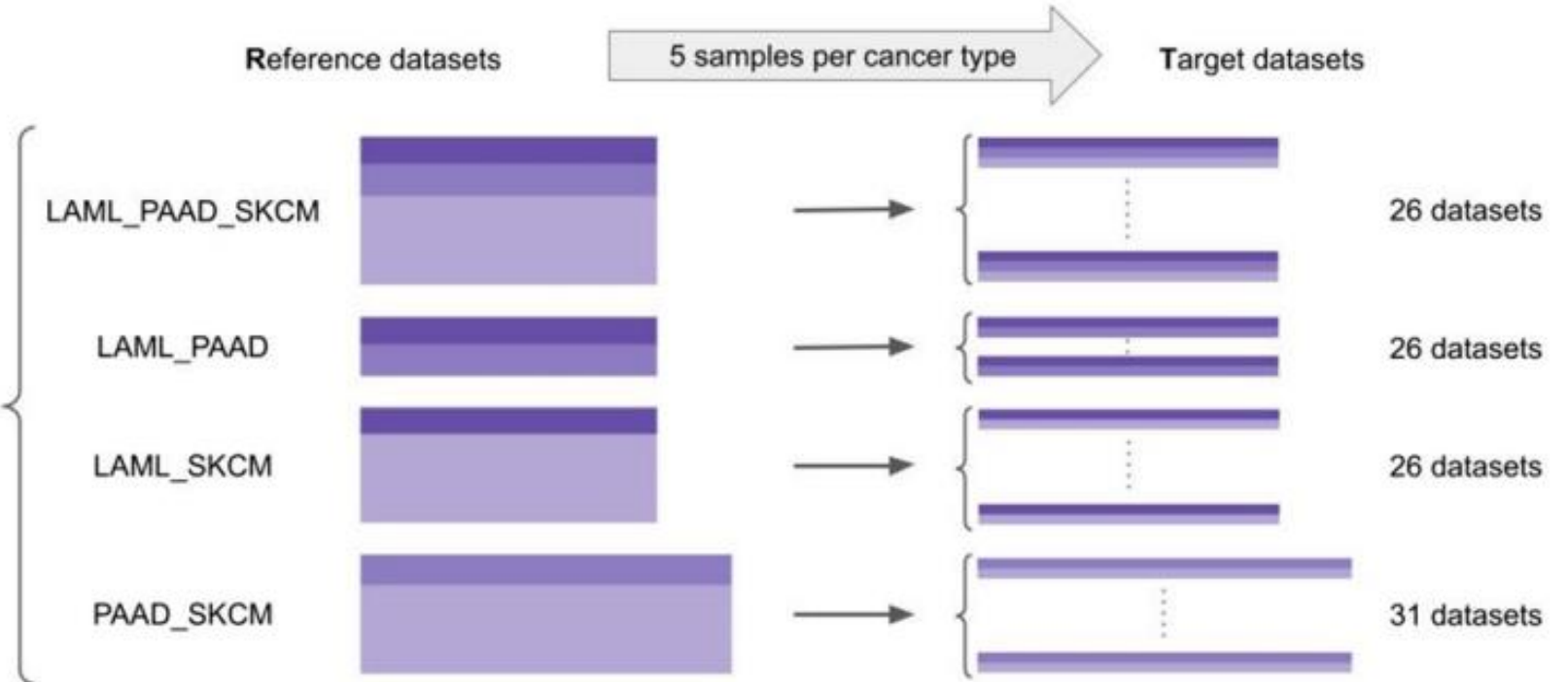
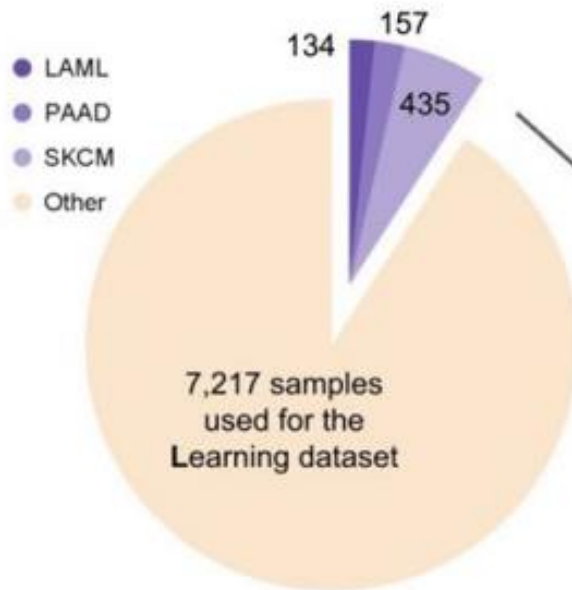


- 32 types of cancer, 29 in Learning, between 2 and 3 in Target.

TCGA data - Pancancer

a

TCGA samples with full multi-omics profiles

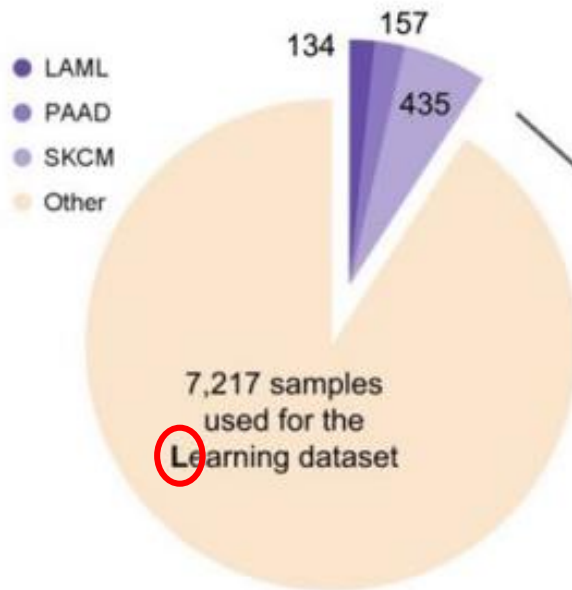


- 32 types of cancer, 29 in Learning, between 2 and 3 in Target.
- 4 omics data: mRNA, miRNA, DNAm, SNV.

TCGA data - Pancancer

a

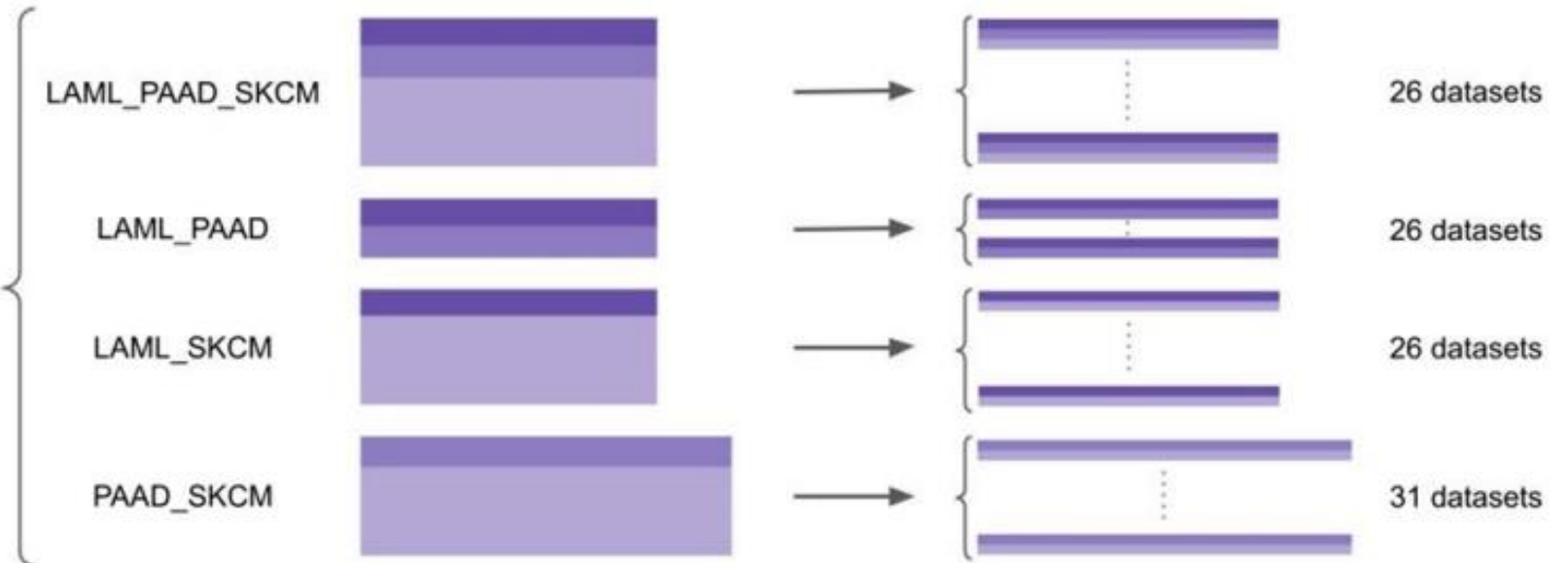
TCGA samples with full multi-omics profiles



Reference datasets

5 samples per cancer type

Target datasets

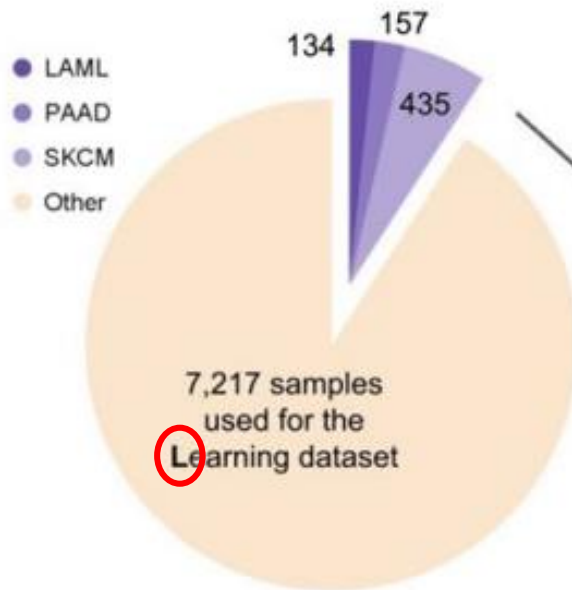


- 32 types of cancer, 29 in Learning, between 2 and 3 in Target.
- 4 omics data: mRNA, miRNA, DNAm, SNV.

TCGA data - Pancancer

a

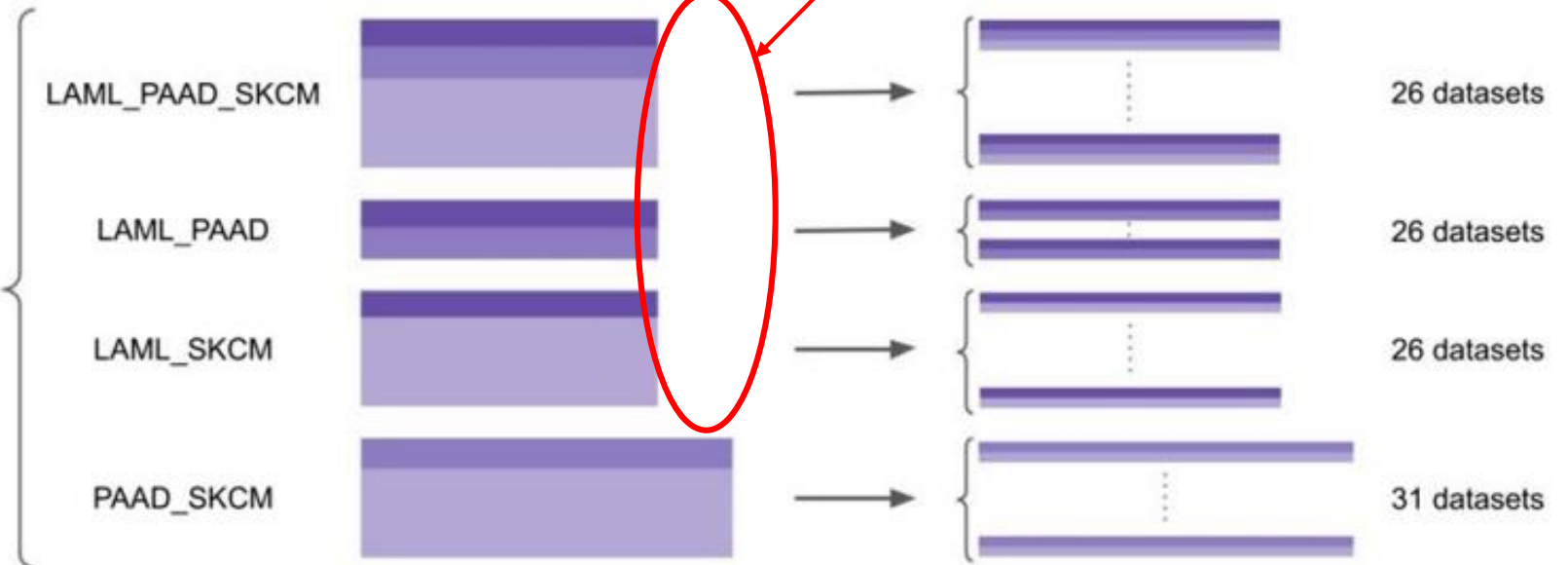
TCGA samples with full multi-omics profiles



Reference datasets

5 samples per cancer type

Target datasets



SNV not included for LAML (too sparse)

- 32 types of cancer, 29 in Learning, between 2 and 3 in Target.
- 4 omics data: mRNA, miRNA, DNAm, SNV.

TCGA data – MOFA factorizations - mRNA



TCGA data – MOFA factorizations - mRNA

Downloading



TCGA data – MOFA factorizations - mRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « Gene Expression Quantification »

TCGA data – MOFA factorizations - mRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « Gene Expression Quantification »

Filtering (logical?)

TCGA data – MOFA factorizations - mRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « Gene Expression Quantification »

Filtering (logical?)

- Remove genes if:

TCGA data – MOFA factorizations - mRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « Gene Expression Quantification »

Filtering (logical?)

- Remove genes if:
 - Null count for more than 90% of samples

TCGA data – MOFA factorizations - mRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « Gene Expression Quantification »

Filtering (logical?)

- Remove genes if:
 - Null count for more than 90% of samples
 - Null variance

TCGA data – MOFA factorizations - mRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « Gene Expression Quantification »

Filtering (logical?)

- Remove genes if:
 - Null count for more than 90% of samples
 - Null variance
 - More than 20% of NA values

TCGA data – MOFA factorizations - mRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « Gene Expression Quantification »

Filtering (logical?)

- Remove genes if:
 - Null count for more than 90% of samples
 - Null variance
 - More than 20% of NA values
 - Genes located on Y chromosome

TCGA data – MOFA factorizations - mRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « Gene Expression Quantification »

Filtering (logical?)

- Remove genes if:
 - Null count for more than 90% of samples
 - Null variance
 - More than 20% of NA values
 - Genes located on Y chromosome

Normalization

TCGA data – MOFA factorizations - mRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « Gene Expression Quantification »

Filtering (logical?)

- Remove genes if:
 - Null count for more than 90% of samples
 - Null variance
 - More than 20% of NA values
 - Genes located on Y chromosome

Normalization

- Normalization with DESeq2, size factor estimation per sample (then divide each sample by it):

TCGA data – MOFA factorizations - mRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « Gene Expression Quantification »

Filtering (logical?)

- Remove genes if:
 - Null count for more than 90% of samples
 - Null variance
 - More than 20% of NA values
 - Genes located on Y chromosome

Normalization

- Normalization with DESeq2, size factor estimation per sample (then divide each sample by it):

$$\hat{s}_j = \underset{i}{\text{median}} \frac{k_{ij}}{\left(\prod_{v=1}^m k_{iv} \right)^{1/m}}.$$

TCGA data – MOFA factorizations - mRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « Gene Expression Quantification »

Filtering (logical?)

- Remove genes if:
 - Null count for more than 90% of samples
 - Null variance
 - More than 20% of NA values
 - Genes located on Y chromosome

Normalization

- Normalization with DESeq2, size factor estimation per sample (then divide each sample by it):

$$\hat{s}_j = \operatorname{median}_i \frac{k_{ij}}{\left(\prod_{v=1}^m k_{iv} \right)^{1/m}}.$$

Transformation

TCGA data – MOFA factorizations - mRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « Gene Expression Quantification »

Filtering (logical?)

- Remove genes if:
 - Null count for more than 90% of samples
 - Null variance
 - More than 20% of NA values
 - Genes located on Y chromosome

Normalization

- Normalization with DESeq2, size factor estimation per sample (then divide each sample by it):

$$\hat{s}_j = \underset{i}{\text{median}} \frac{k_{ij}}{\left(\prod_{v=1}^m k_{iv} \right)^{1/m}}.$$

Transformation

- $\text{Log}_2(x+1)$

TCGA data – MOFA factorizations - mRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « Gene Expression Quantification »

Filtering (logical?)

- Remove genes if:
 - Null count for more than 90% of samples
 - Null variance
 - More than 20% of NA values
 - Genes located on Y chromosome

Normalization

- Normalization with DESeq2, size factor estimation per sample (then divide each sample by it):

$$\hat{s}_j = \operatorname{median}_i \frac{k_{ij}}{\left(\prod_{v=1}^m k_{iv} \right)^{1/m}}.$$

Transformation

- $\log_2(x+1)$

Filtering

TCGA data – MOFA factorizations - mRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « Gene Expression Quantification »

Filtering (logical?)

- Remove genes if:
 - Null count for more than 90% of samples
 - Null variance
 - More than 20% of NA values
 - Genes located on Y chromosome

Normalization

- Normalization with DESeq2, size factor estimation per sample (then divide each sample by it):

$$\hat{s}_j = \operatorname{median}_i \frac{k_{ij}}{\left(\prod_{v=1}^m k_{iv} \right)^{1/m}}.$$

Transformation

- $\log_2(x+1)$

Filtering

- Keep top 5000 variance features

TCGA data - MOFA factorizations - miRNA



TCGA data - MOFA factorizations - miRNA

Downloading



TCGA data - MOFA factorizations - miRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « miRNA Expression Quantification »

TCGA data - MOFA factorizations - miRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « miRNA Expression Quantification »

Filtering (logical?)

TCGA data - MOFA factorizations - miRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « miRNA Expression Quantification »

Filtering (logical?)

- Remove genes if:

TCGA data - MOFA factorizations - miRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « miRNA Expression Quantification »

Filtering (logical?)

- Remove genes if:
 - Null count for more than 90% of samples

TCGA data - MOFA factorizations - miRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « miRNA Expression Quantification »

Filtering (logical?)

- Remove genes if:
 - Null count for more than 90% of samples
 - Null variance

TCGA data - MOFA factorizations - miRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « miRNA Expression Quantification »

Filtering (logical?)

- Remove genes if:
 - Null count for more than 90% of samples
 - Null variance
 - More than 20% of NA values

TCGA data - MOFA factorizations - miRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « miRNA Expression Quantification »

Filtering (logical?)

- Remove genes if:
 - Null count for more than 90% of samples
 - Null variance
 - More than 20% of NA values

Normalization

TCGA data - MOFA factorizations - miRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « miRNA Expression Quantification »

Filtering (logical?)

- Remove genes if:
 - Null count for more than 90% of samples
 - Null variance
 - More than 20% of NA values

Normalization

- Normalization with DESeq2, size factor estimation per sample (then divide each sample by it):

TCGA data - MOFA factorizations - miRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « miRNA Expression Quantification »

Filtering (logical?)

- Remove genes if:
 - Null count for more than 90% of samples
 - Null variance
 - More than 20% of NA values

Normalization

- Normalization with DESeq2, size factor estimation per sample (then divide each sample by it):

$$\hat{s}_j = \operatorname{median}_i \frac{k_{ij}}{\left(\prod_{v=1}^m k_{iv} \right)^{1/m}}.$$

TCGA data - MOFA factorizations - miRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « miRNA Expression Quantification »

Filtering (logical?)

- Remove genes if:
 - Null count for more than 90% of samples
 - Null variance
 - More than 20% of NA values

Normalization

- Normalization with DESeq2, size factor estimation per sample (then divide each sample by it):

$$\hat{s}_j = \operatorname{median}_i \frac{k_{ij}}{\left(\prod_{v=1}^m k_{iv} \right)^{1/m}}.$$

Transformation

TCGA data - MOFA factorizations - miRNA

Downloading

- TCGABiolinks, « Transcriptome Profiling », « miRNA Expression Quantification »

Filtering (logical?)

- Remove genes if:
 - Null count for more than 90% of samples
 - Null variance
 - More than 20% of NA values

Normalization

- Normalization with DESeq2, size factor estimation per sample (then divide each sample by it):

$$\hat{s}_j = \operatorname{median}_i \frac{k_{ij}}{\left(\prod_{v=1}^m k_{iv} \right)^{1/m}}.$$

Transformation

- $\log_2(x+1)$

TCGA data - MOFA factorizations - DNAm



TCGA data - MOFA factorizations - DNAm

Downloading



TCGA data - MOFA factorizations - DNAm

Downloading

- TCGABiolinks, « DNA Methylation », « Methylation Beta Value », «Illumina Human Methylation 450 »

TCGA data - MOFA factorizations - DNAm

Downloading

- TCGABiolinks, « DNA Methylation », « Methylation Beta Value », «Illumina Human Methylation 450 »

Transformation

TCGA data - MOFA factorizations - DNAm

Downloading

- TCGABiolinks, « DNA Methylation », « Methylation Beta Value », «Illumina Human Methylation 450 »

Transformation

- From beta values to M-values ($\text{logit2 } \log_2(p/(1 - p))$)

TCGA data - MOFA factorizations - DNAm

Downloading

- TCGABiolinks, « DNA Methylation », « Methylation Beta Value », «Illumina Human Methylation 450 »

Transformation

- From beta values to M-values ($\text{logit2 } \log_2(p/(1 - p))$)

Filtering

TCGA data - MOFA factorizations - DNAm

Downloading

- TCGABiolinks, « DNA Methylation », « Methylation Beta Value », «Illumina Human Methylation 450 »

Transformation

- From beta values to M-values ($\text{logit2 } \log_2(p/(1 - p))$)

Filtering

- Remove CpGs if:

TCGA data - MOFA factorizations - DNAm

Downloading

- TCGABiolinks, « DNA Methylation », « Methylation Beta Value », «Illumina Human Methylation 450 »

Transformation

- From beta values to M-values ($\text{logit2 } \log_2(p/(1 - p))$)

Filtering

- Remove CpGs if:
 - Null variance

TCGA data - MOFA factorizations - DNAm

Downloading

- TCGABiolinks, « DNA Methylation », « Methylation Beta Value », «Illumina Human Methylation 450 »

Transformation

- From beta values to M-values ($\text{logit2 } \log_2(p/(1 - p))$)

Filtering

- Remove CpGs if:
 - Null variance
 - More than 20% of NA values

TCGA data - MOFA factorizations - DNAm

Downloading

- TCGABiolinks, « DNA Methylation », « Methylation Beta Value », «Illumina Human Methylation 450 »

Transformation

- From beta values to M-values ($\text{logit2 } \log_2(p/(1 - p))$)

Filtering

- Remove CpGs if:
 - Null variance
 - More than 20% of NA values
 - Located on X or Y chromosome

TCGA data - MOFA factorizations - DNAm

Downloading

- TCGABiolinks, « DNA Methylation », « Methylation Beta Value », «Illumina Human Methylation 450 »

Transformation

- From beta values to M-values ($\text{logit2 } \log_2(p/(1 - p))$)

Filtering

- Remove CpGs if:
 - Null variance
 - More than 20% of NA values
 - Located on X or Y chromosome
 - Keep top 5000 variance features

TCGA data - MOFA factorizations - SNV



TCGA data - MOFA factorizations - SNV

Downloading



TCGA data - MOFA factorizations - SNV

Downloading

- TCGABiolinks, « Simple Nucleotide Variation », « open », «Masked Somatic Mutation » (WXS)

TCGA data - MOFA factorizations - SNV

Downloading

- TCGABiolinks, « Simple Nucleotide Variation », « open », «Masked Somatic Mutation » (WXS)

Filtering

TCGA data - MOFA factorizations - SNV

Downloading

- TCGABiolinks, « Simple Nucleotide Variation », « open », «Masked Somatic Mutation » (WXS)

Filtering

- Keep SNV if:

TCGA data - MOFA factorizations - SNV

Downloading

- TCGABiolinks, « Simple Nucleotide Variation », « open », «Masked Somatic Mutation » (WXS)

Filtering

- Keep SNV if:
 - "Frame_Shift_Del", "Frame_Shift_Ins", "In_Frame_Del", "In_Frame_Ins", "Missense_Mutation", "Nonsense_Mutation", "Nonstop_Mutation", "Splice_Site", "Translation_Start_Site"

TCGA data - MOFA factorizations - SNV

Downloading

- TCGABiolinks, « Simple Nucleotide Variation », « open », «Masked Somatic Mutation » (WXS)

Filtering

- Keep SNV if:
 - "Frame_Shift_Del", "Frame_Shift_Ins", "In_Frame_Del", "In_Frame_Ins", "Missense_Mutation", "Nonsense_Mutation", "Nonstop_Mutation", "Splice_Site", "Translation_Start_Site"
 - Tumor_Sample_Barcode

TCGA data - MOFA factorizations - SNV

Downloading

- TCGABiolinks, « Simple Nucleotide Variation », « open », «Masked Somatic Mutation » (WXS)

Filtering

- Keep SNV if:
 - "Frame_Shift_Del", "Frame_Shift_Ins", "In_Frame_Del", "In_Frame_Ins", "Missense_Mutation", "Nonsense_Mutation", "Nonstop_Mutation", "Splice_Site", "Translation_Start_Site"
 - Tumor_Sample_Barcode
- Remove SNV if:

TCGA data - MOFA factorizations - SNV

Downloading

- TCGABiolinks, « Simple Nucleotide Variation », « open », «Masked Somatic Mutation » (WXS)

Filtering

- Keep SNV if:
 - "Frame_Shift_Del", "Frame_Shift_Ins", "In_Frame_Del", "In_Frame_Ins", "Missense_Mutation", "Nonsense_Mutation", "Nonstop_Mutation", "Splice_Site", "Translation_Start_Site"
 - Tumor_Sample_Barcode
- Remove SNV if:
 - Null variance

TCGA data - MOFA factorizations - SNV

Downloading

- TCGABiolinks, « Simple Nucleotide Variation », « open », «Masked Somatic Mutation » (WXS)

Filtering

- Keep SNV if:
 - "Frame_Shift_Del", "Frame_Shift_Ins", "In_Frame_Del", "In_Frame_Ins", "Missense_Mutation", "Nonsense_Mutation", "Nonstop_Mutation", "Splice_Site", "Translation_Start_Site"
 - Tumor_Sample_Barcode
- Remove SNV if:
 - Null variance
 - MAF ≤ 0.01

TCGA data - MOFA factorizations - SNV

Downloading

- TCGABiolinks, « Simple Nucleotide Variation », « open », «Masked Somatic Mutation » (WXS)

Filtering

- Keep SNV if:
 - "Frame_Shift_Del", "Frame_Shift_Ins", "In_Frame_Del", "In_Frame_Ins", "Missense_Mutation", "Nonsense_Mutation", "Nonstop_Mutation", "Splice_Site", "Translation_Start_Site"
 - Tumor_Sample_Barcode
- Remove SNV if:
 - Null variance
 - MAF ≤ 0.01
- Keep top 5000 variance features

TCGA data - MOFA factorizations - SNV

Downloading

- TCGABiolinks, « Simple Nucleotide Variation », « open », «Masked Somatic Mutation » (WXS)

Filtering

- Keep SNV if:
 - "Frame_Shift_Del", "Frame_Shift_Ins", "In_Frame_Del", "In_Frame_Ins", "Missense_Mutation", "Nonsense_Mutation", "Nonstop_Mutation", "Splice_Site", "Translation_Start_Site"
 - Tumor_Sample_Barcode
- Remove SNV if:
 - Null variance
 - MAF ≤ 0.01
- Keep top 5000 variance features

```
#version gdc-1.0.0
#annotation.spec gdc-2.0.0-aliquot-merged-masked
#contigs chr1,chr2,chr3,chr4,chr5,chr6,chr7,chr8,chr9,chr10,chr11,chr12,chr13,chr14,chr15,chr16,chr17,chr18,chr19,chr20,chr21,chr22,chrX,chrY,chrM
#sort.order BarcodesAndCoordinate
#filedate 20221011
#normal.aliquot d4cf16c9-d317-4e83-b201-e3339052b017
#tumor.aliquot 38ca3db4-273b-4798-a732-4dbdc6dc2717
```

Hugo_Symbol	Entrez_Gene_Id	Center	NCBI_Build	Chromosome	Start_Position	End_Position	Strand	Variant_Classification	Variant_Type	Reference_Allele	Tumor_Seq_Allele1	Tumor_Seq_Allele2
CTNNB1	1499	BI	GRCh38	chr3	41224645	41224645	+	Missense_Mutation	SNP	T	T	C
MSL2	55167	BI	GRCh38	chr3	136151712	136151712	+	Missense_Mutation	SNP	T	T	C
NOTCH4	4855	BI	GRCh38	chr6	32202326	32202326	+	Missense_Mutation	SNP	G	G	A
GABRR2	2570	BI	GRCh38	chr6	89257808	89257808	+	Silent	SNP	G	G	A
FOXO4L4	349334	BI	GRCh38	chr9	65737178	65737178	+	Silent	SNP	C	C	T
SEC16A	9919	BI	GRCh38	chr9	136475054	136475054	+	Silent	SNP	C	C	T
MPP7	143098	BI	GRCh38	chr10	28089743	28089743	+	Missense_Mutation	SNP	C	C	T

Other specifications for MOFA factorizations



Other specifications for MOFA factorizations

- « We did not perform any batch effect correction on L datasets in order to preserve biological signal [53]. We checked each R for batch effects with visualizations of UMAP coordinates [54]. We used the R package uwot (v.0.1.14) to derive UMAP coordinates from MOFA factorizations, and we did not observe the need to correct R datasets for batch effects. »

Other specifications for MOFA factorizations

- « We did not perform any batch effect correction on L datasets in order to preserve biological signal [53]. We checked each R for batch effects with visualizations of UMAP coordinates [54]. We used the R package uwot (v.0.1.14) to derive UMAP coordinates from MOFA factorizations, and we did not observe the need to correct R datasets for batch effects. »
- mRNA, miRNA, and DNA → Gaussian
SNV → Bernoulli

Other specifications for MOFA factorizations

- « We did not perform any batch effect correction on L datasets in order to preserve biological signal [53]. We checked each R for batch effects with visualizations of UMAP coordinates [54]. We used the R package uwot (v.0.1.14) to derive UMAP coordinates from MOFA factorizations, and we did not observe the need to correct R datasets for batch effects. »
- mRNA, miRNA, and DNA → Gaussian
SNV → Bernoulli
- **L** → Start at 100 factors (is dropped if fraction of variance explained lower than 0.001 for all views)
« We set the threshold so low in order to retain factors that explained little of the variance in L, yet could be potentially relevant for transfer learning »

$$R^2_{m,k} = 1 - \left(\sum_{n,d} y_{nd}^m - z_{nk} w_{kd}^m - \mu_d^m \right)^2 / \left(\sum_{n,d} y_{nd}^m - \mu_d^m \right)^2$$

- R** → Start at 100 factors (is dropped if fraction of variance explained lower than 0.01 for all views)
T → Start at N_{target} factors (is dropped if fraction of variance explained lower than 0.01 for all views)

Other specifications for MOFA factorizations

- « We did not perform any batch effect correction on L datasets in order to preserve biological signal [53]. We checked each R for batch effects with visualizations of UMAP coordinates [54]. We used the R package uwot (v.0.1.14) to derive UMAP coordinates from MOFA factorizations, and we did not observe the need to correct R datasets for batch effects. »
- mRNA, miRNA, and DNA → Gaussian
SNV → Bernoulli
- **L** → Start at 100 factors (is dropped if fraction of variance explained lower than 0.001 for all views)
« We set the threshold so low in order to retain factors that explained little of the variance in L, yet could be potentially relevant for transfer learning »

$$R_{m,k}^2 = 1 - \left(\sum_{n,d} y_{nd}^m - z_{nk} w_{kd}^m - \mu_d^m \right)^2 / \left(\sum_{n,d} y_{nd}^m - \mu_d^m \right)^2$$

- R** → Start at 100 factors (is dropped if fraction of variance explained lower than 0.01 for all views)
- T** → Start at N_{target} factors (is dropped if fraction of variance explained lower than 0.01 for all views)
- 10,000 iterations to ensure convergence

MOTL factorizations



MOTL factorizations

Pre-processing of each T



MOTL factorizations

Pre-processing of each T

- Remove null variance features



MOTL factorizations

Pre-processing of each T

- Remove null variance features
- Remove features that were removed in corresponding **L** dataset;

MOTL factorizations

Pre-processing of each T

- Remove null variance features
- Remove features that were removed in corresponding **L** dataset;
- Same normalizations/transformation/filtering (geometric means from **L**, $\log_2(1+x)$, beta \rightarrow M values, SNV classification).

MOTL factorizations

Pre-processing of each T

- Remove null variance features
- Remove features that were removed in corresponding **L** dataset;
- Same normalizations/transformation/filtering (geometric means from **L**, $\log_2(1+x)$, beta \rightarrow M values, SNV classification).

Parameters

MOTL factorizations

Pre-processing of each T

- Remove null variance features
- Remove features that were removed in corresponding **L** dataset;
- Same normalizations/transformation/filtering (geometric means from **L**, $\log_2(1+x)$, beta \rightarrow M values, SNV classification).

Parameters

- Get weight and precision values from MOFA models. Precision values are held fixe for Gaussian and Poisson data. For Bernoulli, average of precision parameters across samples to initialize and then updates **following pseudo-Gaussian strategy**. **Intercepts**

MOTL factorizations

Pre-processing of each T

- Remove null variance features
- Remove features that were removed in corresponding **L** dataset;
- Same normalizations/transformation/filtering (geometric means from **L**, $\log_2(1+x)$, beta \rightarrow M values, SNV classification).

Parameters

- Get weight and precision values from MOFA models. Precision values are held fixe for Gaussian and Poisson data. For Bernoulli, average of precision parameters across samples to initialize and then updates **following pseudo-Gaussian strategy. Intercepts**
- The algorithm was stopped when the absolute change in ELBO was under 0.0005% (default MOFA) for every five iterations (to ensure that the algorithm had stopped dropping factors before converging) over a maximum of 10.000 iterations.

Evaluation method



Evaluation method

Ground Truth factors



Evaluation method

Ground Truth factors

- MOFA on each R



Evaluation method

Ground Truth factors

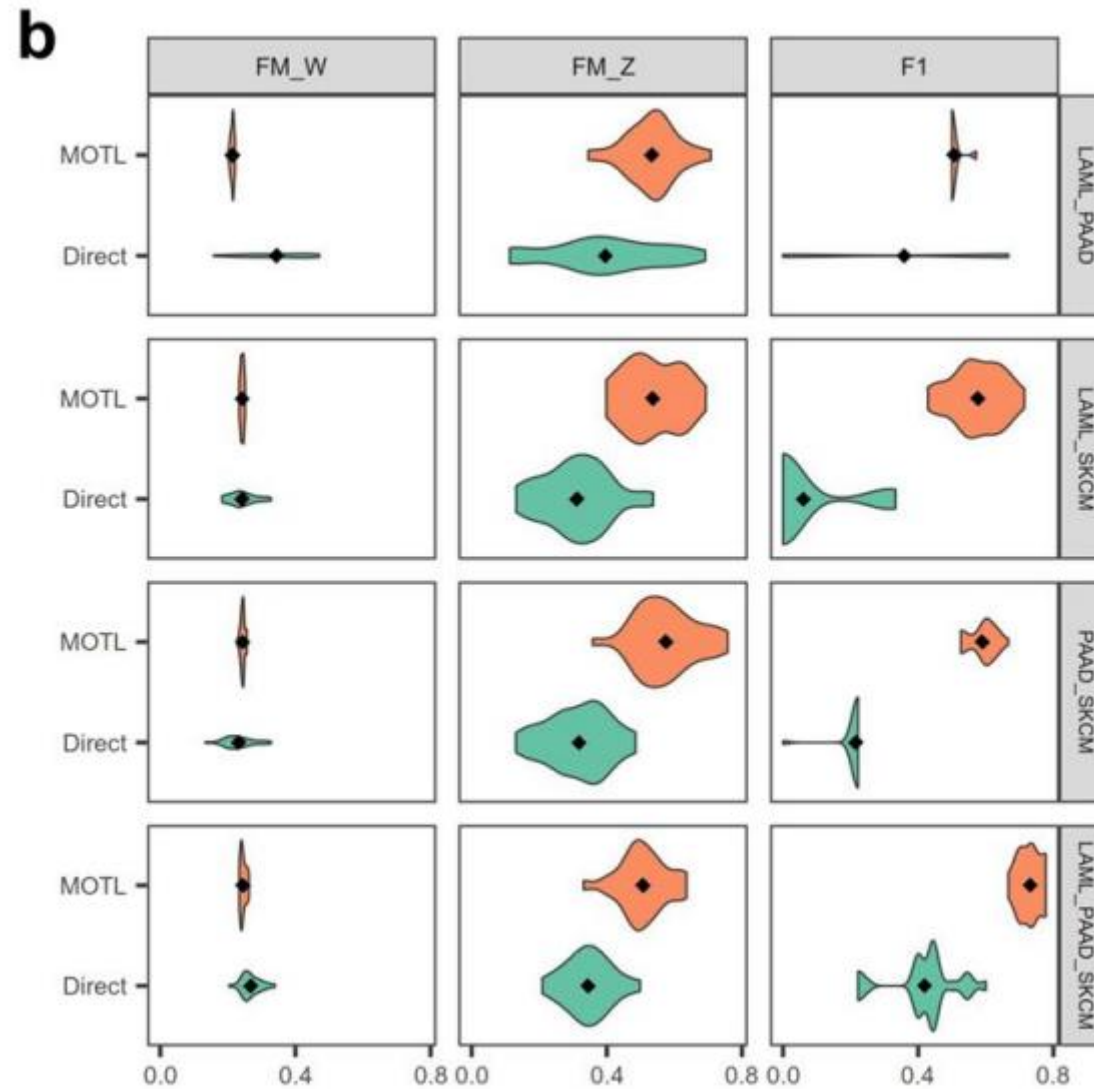
- MOFA on each **R**
- Wilcoxon or Kruskal-Wallis test to determine if a factor is differentially active or not.

Evaluation method

Ground Truth factors

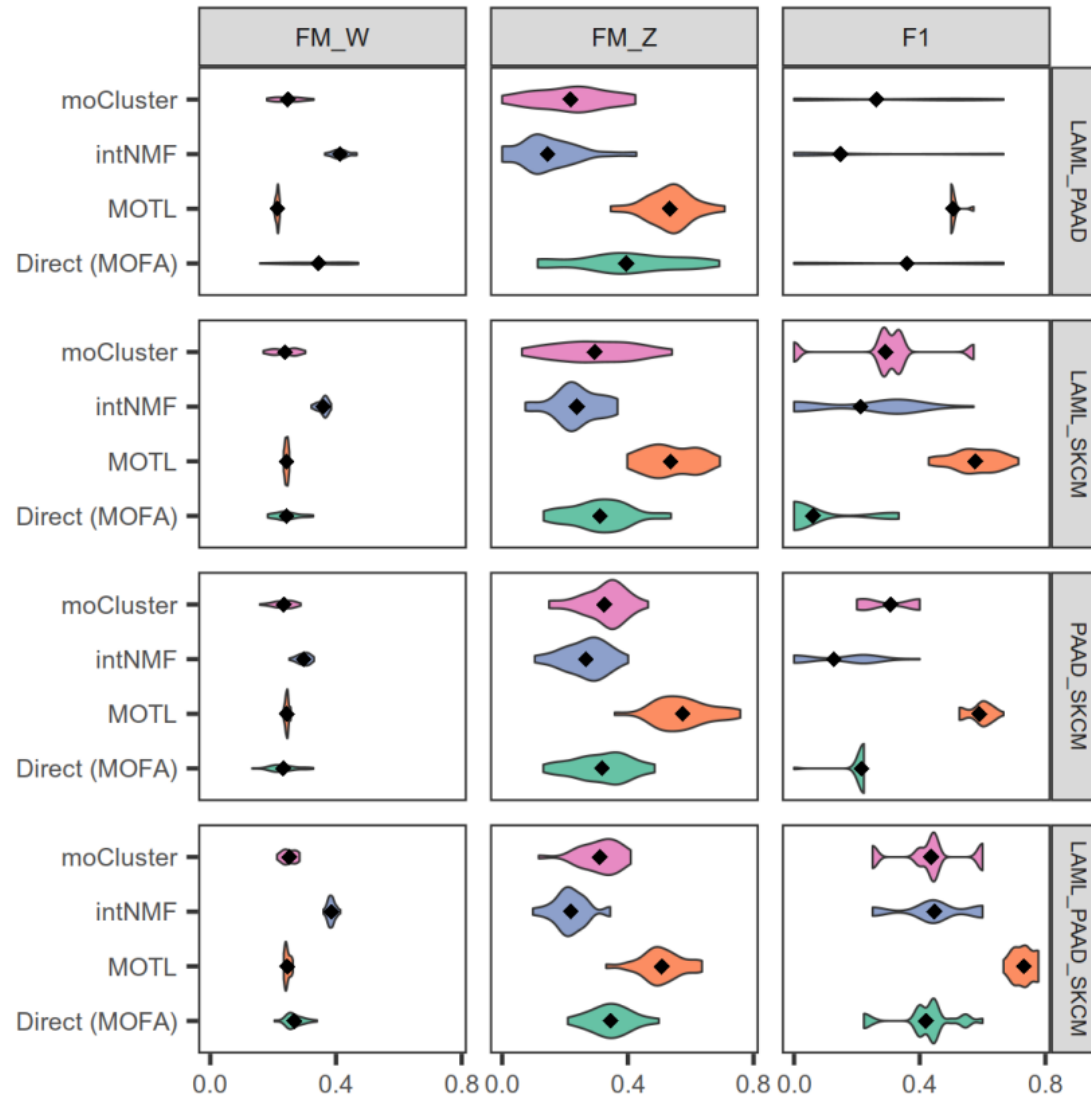
- MOFA on each **R**
- Wilcoxon or Kruskal-Wallis test to determine if a factor is differentially active or not.
- BH adjusted p-values.

Results

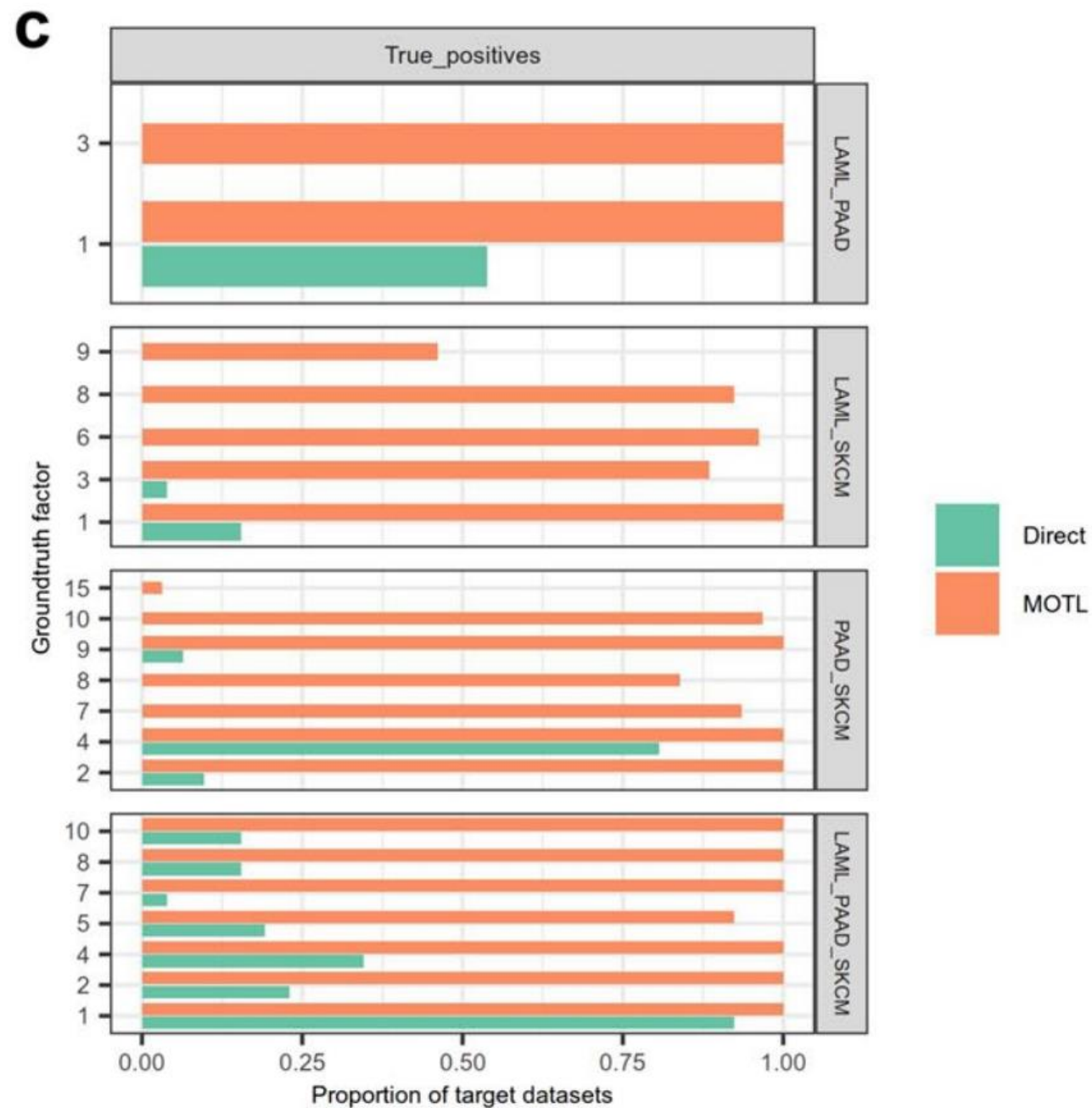


Results – Additionnal Methods

Same pre-processing as for MOFA factorization on each \mathbf{T} , plus off-set for each variable for intNMF.



Results





Results:

Application of MOTL to Glioblastoma

Glioblastoma



Glioblastoma

- Glioblastoma is a rare, heterogeneous, and aggressive cancer type

Glioblastoma

- Glioblastoma is a rare, heterogeneous, and aggressive cancer type
- Need to propose novel therapeutic options however scarcity of data (rare cancer) + invasive biopsies.

Glioblastoma

- Glioblastoma is a rare, heterogeneous, and aggressive cancer type
- Need to propose novel therapeutic options however scarcity of data (rare cancer) + invasive biopsies.
- Past study on patient-derived Glioblastoma Stem Cell (pd-GBSC):

Glioblastoma

- Glioblastoma is a rare, heterogeneous, and aggressive cancer type
- Need to propose novel therapeutic options however scarcity of data (rare cancer) + invasive biopsies.
- Past study on patient-derived Glioblastoma Stem Cell (pd-GBSC):
 - mRNA/DNA

Glioblastoma

- Glioblastoma is a rare, heterogeneous, and aggressive cancer type
- Need to propose novel therapeutic options however scarcity of data (rare cancer) + invasive biopsies.
- Past study on patient-derived Glioblastoma Stem Cell (pd-GBSC):
 - mRNA/DNAseq
 - 4 normal brain samples and 9 patient-derived Glioblastoma Stem Cell cultures.

Glioblastoma

- Glioblastoma is a rare, heterogeneous, and aggressive cancer type
- Need to propose novel therapeutic options however scarcity of data (rare cancer) + invasive biopsies.
- Past study on patient-derived Glioblastoma Stem Cell (pd-GBSC):
 - mRNA/DNA
 - 4 normal brain samples and 9 patient-derived Glioblastoma Stem Cell cultures.
 - 3 subtypes from transcriptome-based signatures from another study): Classical (CL), proneural (PN) and mesenchymal (MS) characterizing the Tumor Micro-Environment of glioma cells.

Glioblastoma

- Glioblastoma is a rare, heterogeneous, and aggressive cancer type
- Need to propose novel therapeutic options however scarcity of data (rare cancer) + invasive biopsies.
- Past study on patient-derived Glioblastoma Stem Cell (pd-GBSC):
 - mRNA/DNA
 - 4 normal brain samples and 9 patient-derived Glioblastoma Stem Cell cultures.
 - 3 subtypes from transcriptome-based signatures from another study): Classical (CL), proneural (PN) and mesenchymal (MS) characterizing the Tumor Micro-Environment of glioma cells.
- Target data-sets:

Glioblastoma

- Glioblastoma is a rare, heterogeneous, and aggressive cancer type
- Need to propose novel therapeutic options however scarcity of data (rare cancer) + invasive biopsies.
- Past study on patient-derived Glioblastoma Stem Cell (pd-GBSC):
 - mRNA/DNA
 - 4 normal brain samples and 9 patient-derived Glioblastoma Stem Cell cultures.
 - 3 subtypes from transcriptome-based signatures from another study): Classical (CL), proneural (PN) and mesenchymal (MS) characterizing the Tumor Micro-Environment of glioma cells.
- Target data-sets:
 - 4 pd-GBSC data-sets: Normal + 1 subtype or Normal + all subtypes.

Glioblastoma

- Glioblastoma is a rare, heterogeneous, and aggressive cancer type
- Need to propose novel therapeutic options however scarcity of data (rare cancer) + invasive biopsies.
- Past study on patient-derived Glioblastoma Stem Cell (pd-GBSC):
 - mRNA/DNA
 - 4 normal brain samples and 9 patient-derived Glioblastoma Stem Cell cultures.
 - 3 subtypes from transcriptome-based signatures from another study): Classical (CL), proneural (PN) and mesenchymal (MS) characterizing the Tumor Micro-Environment of glioma cells.
- Target data-sets:
 - 4 pd-GBSC data-sets: Normal + 1 subtype or Normal + all subtypes.
- Learning dataset:

Glioblastoma

- Glioblastoma is a rare, heterogeneous, and aggressive cancer type
- Need to propose novel therapeutic options however scarcity of data (rare cancer) + invasive biopsies.
- Past study on patient-derived Glioblastoma Stem Cell (pd-GBSC):
 - mRNA/DNA_m
 - 4 normal brain samples and 9 patient-derived Glioblastoma Stem Cell cultures.
 - 3 subtypes from transcriptome-based signatures from another study): Classical (CL), proneural (PN) and mesenchymal (MS) characterizing the Tumor Micro-Environment of glioma cells.
- Target data-sets:
 - 4 pd-GBSC data-sets: Normal + 1 subtype or Normal + all subtypes.
- Learning dataset:
 - All 32 cancers (no GBM) from mRNA/miRNA/SNV/DNA_m

MOFA factorization on Target data-sets



MOFA factorization on Target data-sets

Previous Pre-processing



MOFA factorization on Target data-sets

Previous Pre-processing

- mRNA: pre-processed data from the original study.

MOFA factorization on Target data-sets

Previous Pre-processing

- mRNA: pre-processed data from the original study.
 - Low-abundance genes with less than 10 counts (per sample) or less than 200 counts (all samples) were discarded for downstream analyses.

MOFA factorization on Target data-sets

Previous Pre-processing

- mRNA: pre-processed data from the original study.
 - Low-abundance genes with less than 10 counts (per sample) or less than 200 counts (all samples) were discarded for downstream analyses.
 - the normalized gene expression matrix (VST approach)

MOFA factorization on Target data-sets

Previous Pre-processing

- mRNA: pre-processed data from the original study.
 - Low-abundance genes with less than 10 counts (per sample) or less than 200 counts (all samples) were discarded for downstream analyses.
 - the normalized gene expression matrix (VST approach)

Pre-processing

MOFA factorization on Target data-sets

Previous Pre-processing

- mRNA: pre-processed data from the original study.
 - Low-abundance genes with less than 10 counts (per sample) or less than 200 counts (all samples) were discarded for downstream analyses.
 - the normalized gene expression matrix (VST approach)

Pre-processing

- mRNA:

MOFA factorization on Target data-sets

Previous Pre-processing

- mRNA: pre-processed data from the original study.
 - Low-abundance genes with less than 10 counts (per sample) or less than 200 counts (all samples) were discarded for downstream analyses.
 - the normalized gene expression matrix (VST approach)

Pre-processing

- mRNA:
 - removing genes that map to the Y chromosome

MOFA factorization on Target data-sets

Previous Pre-processing

- mRNA: pre-processed data from the original study.
 - Low-abundance genes with less than 10 counts (per sample) or less than 200 counts (all samples) were discarded for downstream analyses.
 - the normalized gene expression matrix (VST approach)

Pre-processing

- mRNA:
 - removing genes that map to the Y chromosome
 - had a count of zero in $\geq 90\%$ of samples

MOFA factorization on Target data-sets

Previous Pre-processing

- mRNA: pre-processed data from the original study.
 - Low-abundance genes with less than 10 counts (per sample) or less than 200 counts (all samples) were discarded for downstream analyses.
 - the normalized gene expression matrix (VST approach)

Pre-processing

- mRNA:
 - removing genes that map to the Y chromosome
 - had a count of zero in $\geq 90\%$ of samples
 - had zero variance across samples

MOFA factorization on Target data-sets

Previous Pre-processing

- mRNA: pre-processed data from the original study.
 - Low-abundance genes with less than 10 counts (per sample) or less than 200 counts (all samples) were discarded for downstream analyses.
 - the normalized gene expression matrix (VST approach)

Pre-processing

- mRNA:
 - removing genes that map to the Y chromosome
 - had a count of zero in $\geq 90\%$ of samples
 - had zero variance across samples
- DNAm:

MOFA factorization on Target data-sets

Previous Pre-processing

- mRNA: pre-processed data from the original study.
 - Low-abundance genes with less than 10 counts (per sample) or less than 200 counts (all samples) were discarded for downstream analyses.
 - the normalized gene expression matrix (VST approach)

Pre-processing

- mRNA:
 - removing genes that map to the Y chromosome
 - had a count of zero in $\geq 90\%$ of samples
 - had zero variance across samples
- DNAm:
 - β -values to M-values

MOFA factorization on Target data-sets

Previous Pre-processing

- mRNA: pre-processed data from the original study.
 - Low-abundance genes with less than 10 counts (per sample) or less than 200 counts (all samples) were discarded for downstream analyses.
 - the normalized gene expression matrix (VST approach)

Pre-processing

- mRNA:
 - removing genes that map to the Y chromosome
 - had a count of zero in $\geq 90\%$ of samples
 - had zero variance across samples
- DNAm:
 - β -values to M-values
 - removing CpG sites that had missing values in $\geq 20\%$ of samples

MOFA factorization on Target data-sets

Previous Pre-processing

- mRNA: pre-processed data from the original study.
 - Low-abundance genes with less than 10 counts (per sample) or less than 200 counts (all samples) were discarded for downstream analyses.
 - the normalized gene expression matrix (VST approach)

Pre-processing

- mRNA:
 - removing genes that map to the Y chromosome
 - had a count of zero in $\geq 90\%$ of samples
 - had zero variance across samples
- DNAm:
 - β -values to M-values
 - removing CpG sites that had missing values in $\geq 20\%$ of samples
 - had zero variance across samples

MOFA factorization on Target data-sets

Previous Pre-processing

- mRNA: pre-processed data from the original study.
 - Low-abundance genes with less than 10 counts (per sample) or less than 200 counts (all samples) were discarded for downstream analyses.
 - the normalized gene expression matrix (VST approach)

Pre-processing

- mRNA:
 - removing genes that map to the Y chromosome
 - had a count of zero in $\geq 90\%$ of samples
 - had zero variance across samples
- DNAm:
 - β -values to M-values
 - removing CpG sites that had missing values in $\geq 20\%$ of samples
 - had zero variance across samples
- Filtered both omics to include only the 5000 most variable features.

MOFA factorization on Target data-sets

Previous Pre-processing

- mRNA: pre-processed data from the original study.
 - Low-abundance genes with less than 10 counts (per sample) or less than 200 counts (all samples) were discarded for downstream analyses.
 - the normalized gene expression matrix (VST approach)

Pre-processing

- mRNA:
 - removing genes that map to the Y chromosome
 - had a count of zero in $\geq 90\%$ of samples
 - had zero variance across samples
- DNAm:
 - β -values to M-values
 - removing CpG sites that had missing values in $\geq 20\%$ of samples
 - had zero variance across samples
- Filtered both omics to include only the 5000 most variable features.

Parameters

MOFA factorization on Target data-sets

Previous Pre-processing

- mRNA: pre-processed data from the original study.
 - Low-abundance genes with less than 10 counts (per sample) or less than 200 counts (all samples) were discarded for downstream analyses.
 - the normalized gene expression matrix (VST approach)

Pre-processing

- mRNA:
 - removing genes that map to the Y chromosome
 - had a count of zero in $\geq 90\%$ of samples
 - had zero variance across samples
- DNAm:
 - β -values to M-values
 - removing CpG sites that had missing values in $\geq 20\%$ of samples
 - had zero variance across samples
- Filtered both omics to include only the 5000 most variable features.

Parameters

- Gaussian as the observed data likelihood for the mRNA and the DNA

MOFA factorization on Target data-sets

Previous Pre-processing

- mRNA: pre-processed data from the original study.
 - Low-abundance genes with less than 10 counts (per sample) or less than 200 counts (all samples) were discarded for downstream analyses.
 - the normalized gene expression matrix (VST approach)

Pre-processing

- mRNA:
 - removing genes that map to the Y chromosome
 - had a count of zero in $\geq 90\%$ of samples
 - had zero variance across samples
- DNAm:
 - β -values to M-values
 - removing CpG sites that had missing values in $\geq 20\%$ of samples
 - had zero variance across samples
- Filtered both omics to include only the 5000 most variable features.

Parameters

- Gaussian as the observed data likelihood for the mRNA and the DNA
- started with the maximum allowable number of factors and dropped factors based on a threshold of 0.01

MOTL factorization on Target data-sets



MOTL factorization on Target data-sets

Pre-processing



MOTL factorization on Target data-sets

Pre-processing

- mRNA:



MOTL factorization on Target data-sets

Pre-processing

- mRNA:
 - Null variance and removed from \mathbf{L} during pre-processing

MOTL factorization on Target data-sets

Pre-processing

- mRNA:
 - Null variance and removed from \mathbf{L} during pre-processing
 - DESeq2 to normalize mRNA counts with the geometric means from \mathbf{L} ,

MOTL factorization on Target data-sets

Pre-processing

- mRNA:
 - Null variance and removed from **L** during pre-processing
 - DESeq2 to normalize mRNA counts with the geometric means from **L**,
 - $\log_2(x + 1)$ transformed the normalized counts.

MOTL factorization on Target data-sets

Pre-processing

- mRNA:
 - Null variance and removed from **L** during pre-processing
 - DESeq2 to normalize mRNA counts with the geometric means from **L**,
 - $\log_2(x + 1)$ transformed the normalized counts.
- DNAm

MOTL factorization on Target data-sets

Pre-processing

- mRNA:
 - Null variance and removed from **L** during pre-processing
 - DESeq2 to normalize mRNA counts with the geometric means from **L**,
 - $\log_2(x + 1)$ transformed the normalized counts.
- DNAm
 - β -values to M-values

MOTL factorization on Target data-sets

Pre-processing

- mRNA:
 - Null variance and removed from **L** during pre-processing
 - DESeq2 to normalize mRNA counts with the geometric means from **L**,
 - $\log_2(x + 1)$ transformed the normalized counts.
- DNAm
 - β -values to M-values

Parameters

MOTL factorization on Target data-sets

Pre-processing

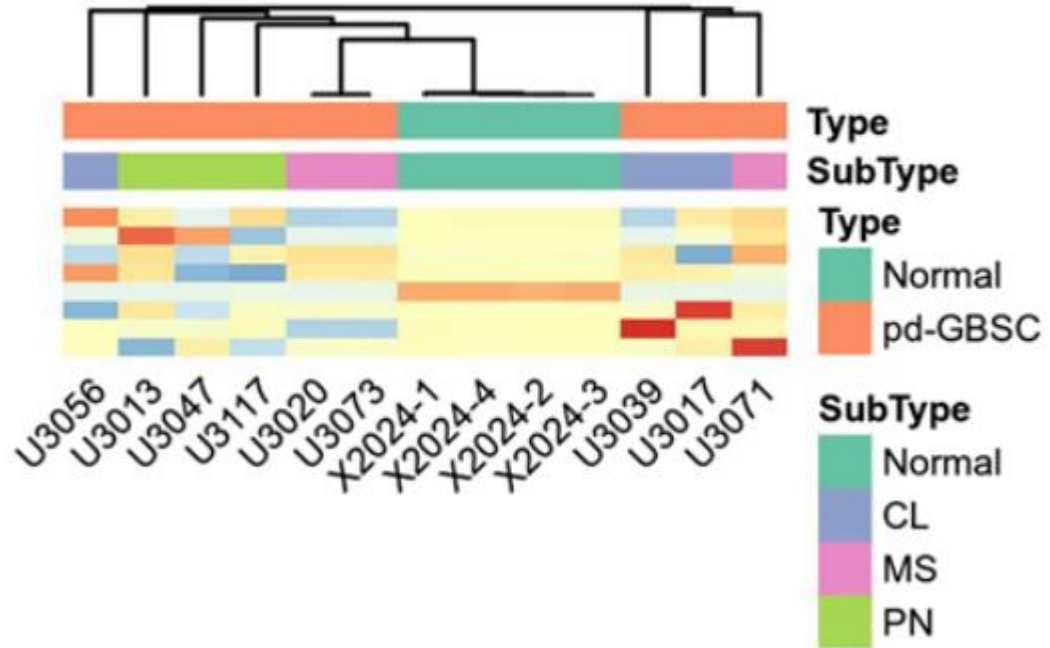
- mRNA:
 - Null variance and removed from **L** during pre-processing
 - DESeq2 to normalize mRNA counts with the geometric means from **L**,
 - $\log_2(x + 1)$ transformed the normalized counts.
- DNAm
 - β -values to M-values

Parameters

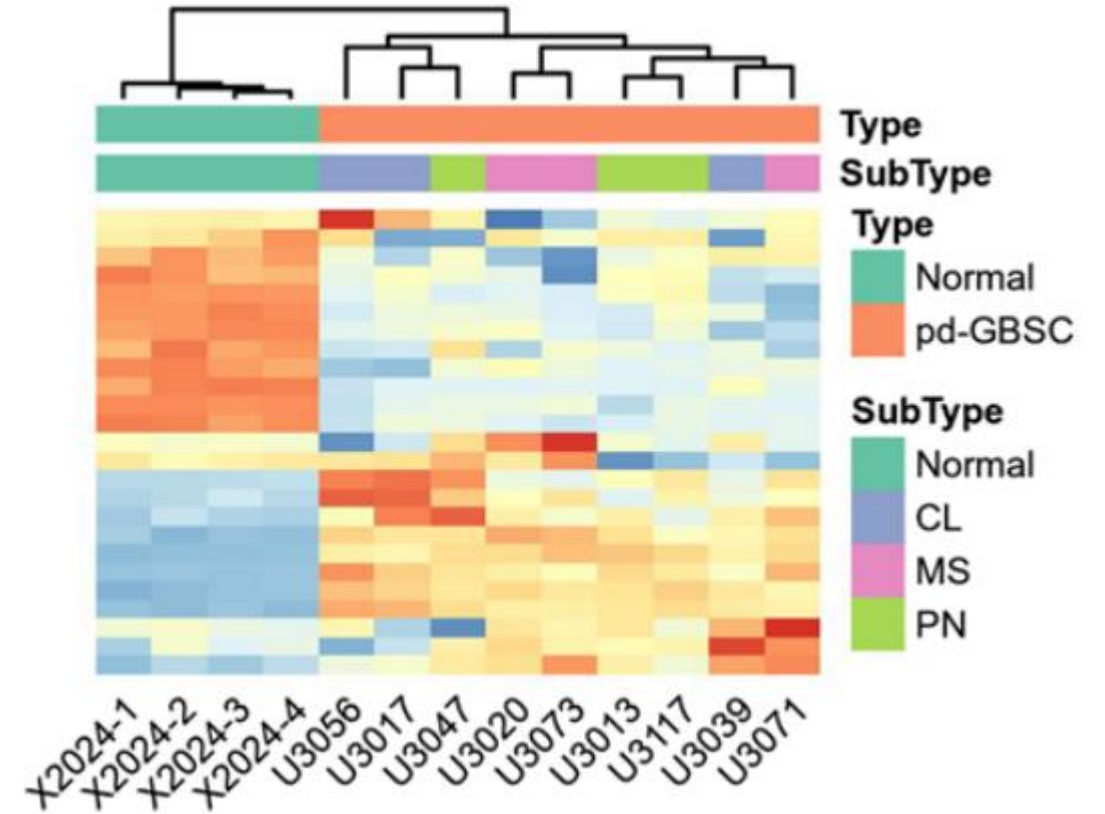
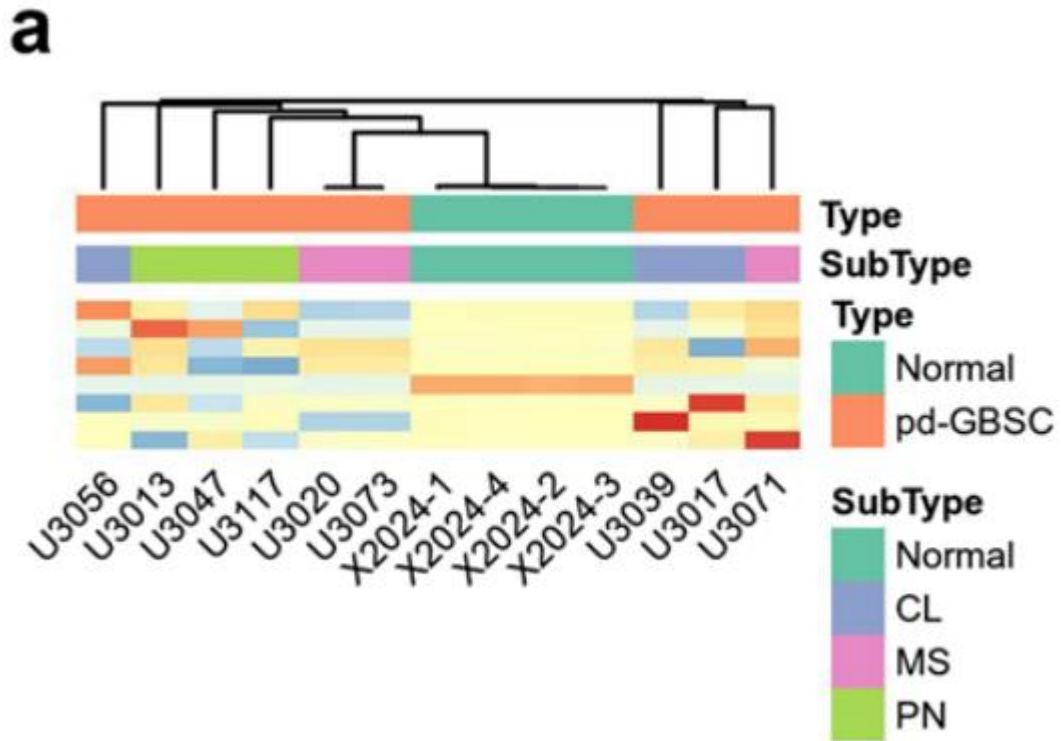
- Same as for TCGA study.

Results

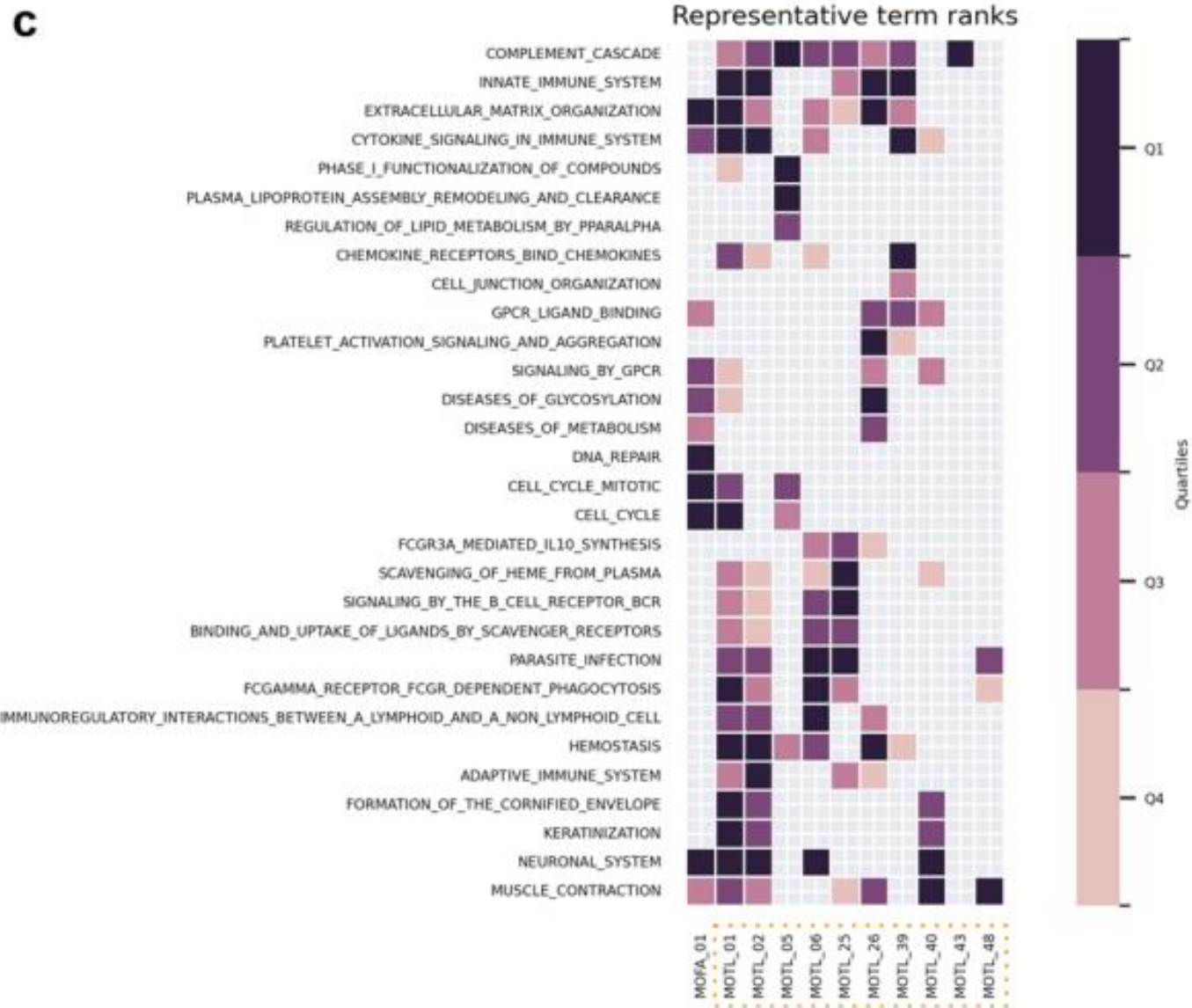
a



Results



Results - GSEA





Discussion

Discussion



Discussion

- extend the evaluations by using a larger range of sizes for target datasets, in order to identify a crossover point at which transfer learning no longer enhances matrix factorization

Discussion

- extend the evaluations by using a larger range of sizes for target datasets, in order to identify a crossover point at which transfer learning no longer enhances matrix factorization
- measure of similarity, between a given learning and target dataset, which would predict the effectiveness of using a transfer learning approach to apply matrix factorization to the target dataset.

Discussion

- extend the evaluations by using a larger range of sizes for target datasets, in order to identify a crossover point at which transfer learning no longer enhances matrix factorization
- measure of similarity, between a given learning and target dataset, which would predict the effectiveness of using a transfer learning approach to apply matrix factorization to the target dataset.
 - optimal transport

Discussion

- extend the evaluations by using a larger range of sizes for target datasets, in order to identify a crossover point at which transfer learning no longer enhances matrix factorization
- measure of similarity, between a given learning and target dataset, which would predict the effectiveness of using a transfer learning approach to apply matrix factorization to the target dataset.
 - optimal transport
 - maximum mean discrepancy

Discussion

- extend the evaluations by using a larger range of sizes for target datasets, in order to identify a crossover point at which transfer learning no longer enhances matrix factorization
- measure of similarity, between a given learning and target dataset, which would predict the effectiveness of using a transfer learning approach to apply matrix factorization to the target dataset.
 - optimal transport
 - maximum mean discrepancy
 - Alternatively, a relevant ontology

Discussion

- extend the evaluations by using a larger range of sizes for target datasets, in order to identify a crossover point at which transfer learning no longer enhances matrix factorization
- measure of similarity, between a given learning and target dataset, which would predict the effectiveness of using a transfer learning approach to apply matrix factorization to the target dataset.
 - optimal transport
 - maximum mean discrepancy
 - Alternatively, a relevant ontology
- how heterogeneous (i.e., representing a large diversity of tissues, diseases, experimental conditions...) a learning dataset needs to be, in order to yield factors which can be relevant for a given target dataset.

Discussion

- extend the evaluations by using a larger range of sizes for target datasets, in order to identify a crossover point at which transfer learning no longer enhances matrix factorization
- measure of similarity, between a given learning and target dataset, which would predict the effectiveness of using a transfer learning approach to apply matrix factorization to the target dataset.
 - optimal transport
 - maximum mean discrepancy
 - Alternatively, a relevant ontology
- how heterogeneous (i.e., representing a large diversity of tissues, diseases, experimental conditions...) a learning dataset needs to be, in order to yield factors which can be relevant for a given target dataset.
- application of MOTL to target datasets with non-cancer rare disease samples, using factors inferred from the TCGA learning dataset.

Discussion

- extend the evaluations by using a larger range of sizes for target datasets, in order to identify a crossover point at which transfer learning no longer enhances matrix factorization
- measure of similarity, between a given learning and target dataset, which would predict the effectiveness of using a transfer learning approach to apply matrix factorization to the target dataset.
 - optimal transport
 - maximum mean discrepancy
 - Alternatively, a relevant ontology
- how heterogeneous (i.e., representing a large diversity of tissues, diseases, experimental conditions...) a learning dataset needs to be, in order to yield factors which can be relevant for a given target dataset.
- application of MOTL to target datasets with non-cancer rare disease samples, using factors inferred from the TCGA learning dataset.
- For example, instead of fixing the feature weight values, they could be treated as random variables by MOTL, with priors informed by the factorization of the learning dataset.

Discussion

- extend the evaluations by using a larger range of sizes for target datasets, in order to identify a crossover point at which transfer learning no longer enhances matrix factorization
- measure of similarity, between a given learning and target dataset, which would predict the effectiveness of using a transfer learning approach to apply matrix factorization to the target dataset.
 - optimal transport
 - maximum mean discrepancy
 - Alternatively, a relevant ontology
- how heterogeneous (i.e., representing a large diversity of tissues, diseases, experimental conditions...) a learning dataset needs to be, in order to yield factors which can be relevant for a given target dataset.
- application of MOTL to target datasets with non-cancer rare disease samples, using factors inferred from the TCGA learning dataset.
- For example, instead of fixing the feature weight values, they could be treated as random variables by MOTL, with priors informed by the factorization of the learning dataset.
- TL for different multi-omics matrix factorization methods

Discussion

- extend the evaluations by using a larger range of sizes for target datasets, in order to identify a crossover point at which transfer learning no longer enhances matrix factorization
- measure of similarity, between a given learning and target dataset, which would predict the effectiveness of using a transfer learning approach to apply matrix factorization to the target dataset.
 - optimal transport
 - maximum mean discrepancy
 - Alternatively, a relevant ontology
- how heterogeneous (i.e., representing a large diversity of tissues, diseases, experimental conditions...) a learning dataset needs to be, in order to yield factors which can be relevant for a given target dataset.
- application of MOTL to target datasets with non-cancer rare disease samples, using factors inferred from the TCGA learning dataset.
- For example, instead of fixing the feature weight values, they could be treated as random variables by MOTL, with priors informed by the factorization of the learning dataset.
- TL for different multi-omics matrix factorization methods
- Extend projectR for multiomics (comparable accross jDR).

Discussion

- extend the evaluations by using a larger range of sizes for target datasets, in order to identify a crossover point at which transfer learning no longer enhances matrix factorization
- measure of similarity, between a given learning and target dataset, which would predict the effectiveness of using a transfer learning approach to apply matrix factorization to the target dataset.
 - optimal transport
 - maximum mean discrepancy
 - Alternatively, a relevant ontology
- how heterogeneous (i.e., representing a large diversity of tissues, diseases, experimental conditions...) a learning dataset needs to be, in order to yield factors which can be relevant for a given target dataset.
- application of MOTL to target datasets with non-cancer rare disease samples, using factors inferred from the TCGA learning dataset.
- For example, instead of fixing the feature weight values, they could be treated as random variables by MOTL, with priors informed by the factorization of the learning dataset.
- TL for different multi-omics matrix factorization methods
- Extend projectR for multiomics (comparable accross jDR).
- A consequence is that some features which are highly variable in the target dataset may not contribute to the MOTL factorization. Therefore a future extension could be to add flexibility into the MOTL workflow, so that all features that are highly variable in the target dataset contribute to the factorization, even if they were not retained for the factorization of the learning dataset.

Discussion

- extend the evaluations by using a larger range of sizes for target datasets, in order to identify a crossover point at which transfer learning no longer enhances matrix factorization
- measure of similarity, between a given learning and target dataset, which would predict the effectiveness of using a transfer learning approach to apply matrix factorization to the target dataset.
 - optimal transport
 - maximum mean discrepancy
 - Alternatively, a relevant ontology
- how heterogeneous (i.e., representing a large diversity of tissues, diseases, experimental conditions...) a learning dataset needs to be, in order to yield factors which can be relevant for a given target dataset.
- application of MOTL to target datasets with non-cancer rare disease samples, using factors inferred from the TCGA learning dataset.
- For example, instead of fixing the feature weight values, they could be treated as random variables by MOTL, with priors informed by the factorization of the learning dataset.
- TL for different multi-omics matrix factorization methods
- Extend projectR for multiomics (comparable accross jDR).
- A consequence is that some features which are highly variable in the target dataset may not contribute to the MOTL factorization. Therefore a future extension could be to add flexibility into the MOTL workflow, so that all features that are highly variable in the target dataset contribute to the factorization, even if they were not retained for the factorization of the learning dataset.
- did not identify benchmarks comparing linear methods based on MF with deep learning methods in the context of bulk multiomics data integration, in particular for small target datasets.

Discussion



Discussion

Drawbacks:



Discussion

Drawbacks:

Is it really a Transfer Learning method ?

Discussion

Drawbacks:

Is it really a Transfer Learning method ?

Focus on differentially active.

Discussion

Drawbacks:

Is it really a Transfer Learning method ?

Focus on differentially active.

Glioblastoma

Discussion

Drawbacks:

Is it really a Transfer Learning method ?

Focus on differentially active.

Glioblastoma

- Why using pre-process for MOFA

Discussion

Drawbacks:

Is it really a Transfer Learning method ?

Focus on differentially active.

Glioblastoma

- Why using pre-process for MOFA
- Discussion on omic asymmetry

Discussion

Drawbacks:

Is it really a Transfer Learning method ?

Focus on differentially active.

Glioblastoma

- Why using pre-process for MOFA
- Discussion on omic asymmetry

Advantages:

Discussion

Drawbacks:

Is it really a Transfer Learning method ?

Focus on differentially active.

Glioblastoma

- Why using pre-process for MOFA
- Discussion on omic asymmetry

Advantages:

First proposition for TL in the field of jDR

Discussion

Drawbacks:

Is it really a Transfer Learning method ?

Focus on differentially active.

Glioblastoma

- Why using pre-process for MOFA
- Discussion on omic asymmetry

Advantages:

First proposition for TL in the field of jDR

Experiments are clearly oriented towards TL

Discussion

Drawbacks:

Is it really a Transfer Learning method ?

Focus on differentially active.

Glioblastoma

- Why using pre-process for MOFA
- Discussion on omic asymmetry

Advantages:

First proposition for TL in the field of jDR

Experiments are clearly oriented towards TL

Prediction method for MOFA

Discussion

Drawbacks:

Is it really a Transfer Learning method ?

Focus on differentially active.

Glioblastoma

- Why using pre-process for MOFA
- Discussion on omic asymmetry

Advantages:

First proposition for TL in the field of jDR

Experiments are clearly oriented towards TL

Prediction method for MOFA

Glioblastoma application

Discussion

Drawbacks:

Is it really a Transfer Learning method ?

Focus on differentially active.

Glioblastoma

- Why using pre-process for MOFA
- Discussion on omic asymmetry

Advantages:

First proposition for TL in the field of jDR

Experiments are clearly oriented towards TL

Prediction method for MOFA

Glioblastoma application

...TL at CNRGH



Thank you for your attention.

Questions ?



ChAMP's representation: Kruskal-Wallis test

ChAMP's representation: Kruskal-Wallis test



The Kruskal-Wallis test is a generalization of the Wilcoxon-Mann-Whitney test that works for two samples.

ChAMP's representation: Kruskal-Wallis test



The Kruskal-Wallis test is a generalization of the Wilcoxon-Mann-Whitney test that works for two samples. They are both **non-parametric**.

ChAMP's representation: Kruskal-Wallis test

The Kruskal-Wallis test is a generalization of the Wilcoxon-Man-Withney test that works for two samples. They are both **non-parametric**.

The Wilcoxon-Man-Withney proposes to test the association between a continuous (ex: age) and a discrete variable (ex: sex).

ChAMP's representation: Kruskal-Wallis test

The Kruskal-Wallis test is a generalization of the Wilcoxon-Man-Withney test that works for two samples. They are both **non-parametric**.

The Wilcoxon-Man-Withney proposes to test the association between a continuous (ex: age) and a discrete variable (ex: sex).

Let us consider two samples (x_1, \dots, x_n) and (y_1, \dots, y_m) . They both represent the same continuous variable but are separated by the value of the discrete one.

The Kruskal-Wallis test is a generalization of the Wilcoxon-Man-Whitney test that works for two samples. They are both **non-parametric**.

The Wilcoxon-Mann-Whitney proposes to test the association between a continuous (ex: age) and a discrete variable (ex: sex).

Let us consider two samples (x_1, \dots, x_n) and (y_1, \dots, y_m) . They both represent the same continuous variable but are separated by the value of the discrete one.

$$\{$$

ChAMP's representation: Kruskal-Wallis test

The Kruskal-Wallis test is a generalization of the Wilcoxon-Man-Withney test that works for two samples. They are both **non-parametric**.

The Wilcoxon-Man-Withney proposes to test the association between a continuous (ex: age) and a discrete variable (ex: sex).

Let us consider two samples (x_1, \dots, x_n) and (y_1, \dots, y_m) . They both represent the same continuous variable but are separated by the value of the discrete one.

$$\left\{ \begin{array}{l} H_0: (x_1, \dots, x_n) \text{ and } (y_1, \dots, y_m) \text{ comes from the same distribution.} \end{array} \right.$$

ChAMP's representation: Kruskal-Wallis test

The Kruskal-Wallis test is a generalization of the Wilcoxon-Man-Withney test that works for two samples. They are both **non-parametric**.

The Wilcoxon-Man-Withney proposes to test the association between a continuous (ex: age) and a discrete variable (ex: sex).

Let us consider two samples (x_1, \dots, x_n) and (y_1, \dots, y_m) . They both represent the same continuous variable but are separated by the value of the discrete one.

$$\begin{cases} H_0: (x_1, \dots, x_n) \text{ and } (y_1, \dots, y_m) \text{ comes from the same distribution.} \\ H_1: \text{They do not.} \end{cases}$$

ChAMP's representation: Kruskal-Wallis test

The Kruskal-Wallis test is a generalization of the Wilcoxon-Man-Withney test that works for two samples. They are both **non-parametric**.

The Wilcoxon-Man-Withney proposes to test the association between a continuous (ex: age) and a discrete variable (ex: sex).

Let us consider two samples (x_1, \dots, x_n) and (y_1, \dots, y_m) . They both represent the same continuous variable but are separated by the value of the discrete one.

$$\begin{cases} H_0: (x_1, \dots, x_n) \text{ and } (y_1, \dots, y_m) \text{ comes from the same distribution.} \\ H_1: \text{They do not.} \end{cases}$$

The proposed statistic is: $U = \min(U_1, U_2) = \min\left(nm + \frac{n(n+1)}{2} - R_1, nm + \frac{m(m+1)}{2} - R_2\right)$,

where R_1 (resp. R_2) are the sum of the rank of the first (resp. second) sample when all samples are mixed and sorted.

ChAMP's representation: Kruskal-Wallis test

The Kruskal-Wallis test is a generalization of the Wilcoxon-Man-Withney test that works for two samples. They are both **non-parametric**.

The Wilcoxon-Man-Withney proposes to test the association between a continuous (ex: age) and a discrete variable (ex: sex).

Let us consider two samples (x_1, \dots, x_n) and (y_1, \dots, y_m) . They both represent the same continuous variable but are separated by the value of the discrete one.

$$\begin{cases} H_0: (x_1, \dots, x_n) \text{ and } (y_1, \dots, y_m) \text{ comes from the same distribution.} \\ H_1: \text{They do not.} \end{cases}$$

The proposed statistic is: $U = \min(U_1, U_2) = \min\left(nm + \frac{n(n+1)}{2} - R_1, nm + \frac{m(m+1)}{2} - R_2\right)$,

where R_1 (resp. R_2) are the sum of the rank of the first (resp. second) sample when all samples are mixed and sorted.

If n and m are high enough, it is possible to show that U follows a Gaussian distribution centered in $\frac{nm+1}{2}$.

ChAMP's representation: Kruskal-Wallis test

The Kruskal-Wallis test is a generalization of the Wilcoxon-Man-Withney test that works for two samples. They are both **non-parametric**.

The Wilcoxon-Man-Withney proposes to test the association between a continuous (ex: age) and a discrete variable (ex: sex).

Let us consider two samples (x_1, \dots, x_n) and (y_1, \dots, y_m) . They both represent the same continuous variable but are separated by the value of the discrete one.

$$\begin{cases} H_0: (x_1, \dots, x_n) \text{ and } (y_1, \dots, y_m) \text{ comes from the same distribution.} \\ H_1: \text{They do not.} \end{cases}$$

The proposed statistic is: $U = \min(U_1, U_2) = \min\left(nm + \frac{n(n+1)}{2} - R_1, nm + \frac{m(m+1)}{2} - R_2\right)$,

where R_1 (resp. R_2) are the sum of the rank of the first (resp. second) sample when all samples are mixed and sorted.

If n and m are high enough, it is possible to show that U follows a Gaussian distribution centered in $\frac{nm+1}{2}$.

Example: "Perfect" association; the n first elements are in the first sample and the m next are in the second one. Then:

ChAMP's representation: Kruskal-Wallis test

The Kruskal-Wallis test is a generalization of the Wilcoxon-Man-Withney test that works for two samples. They are both **non-parametric**.

The Wilcoxon-Man-Withney proposes to test the association between a continuous (ex: age) and a discrete variable (ex: sex).

Let us consider two samples (x_1, \dots, x_n) and (y_1, \dots, y_m) . They both represent the same continuous variable but are separated by the value of the discrete one.

$$\begin{cases} H_0: (x_1, \dots, x_n) \text{ and } (y_1, \dots, y_m) \text{ comes from the same distribution.} \\ H_1: \text{They do not.} \end{cases}$$

The proposed statistic is: $U = \min(U_1, U_2) = \min\left(nm + \frac{n(n+1)}{2} - R_1, nm + \frac{m(m+1)}{2} - R_2\right)$,

where R_1 (resp. R_2) are the sum of the rank of the first (resp. second) sample when all samples are mixed and sorted.

If n and m are high enough, it is possible to show that U follows a Gaussian distribution centered in $\frac{nm+1}{2}$.

Example: "Perfect" association; the n first elements are in the first sample and the m next are in the second one.

Then:

$$R_1 = 1 + 2 + \dots + n = \frac{n(n+1)}{2}$$

ChAMP's representation: Kruskal-Wallis test

The Kruskal-Wallis test is a generalization of the Wilcoxon-Man-Withney test that works for two samples. They are both **non-parametric**.

The Wilcoxon-Man-Withney proposes to test the association between a continuous (ex: age) and a discrete variable (ex: sex).

Let us consider two samples (x_1, \dots, x_n) and (y_1, \dots, y_m) . They both represent the same continuous variable but are separated by the value of the discrete one.

$$\begin{cases} H_0: (x_1, \dots, x_n) \text{ and } (y_1, \dots, y_m) \text{ comes from the same distribution.} \\ H_1: \text{They do not.} \end{cases}$$

The proposed statistic is: $U = \min(U_1, U_2) = \min\left(nm + \frac{n(n+1)}{2} - R_1, nm + \frac{m(m+1)}{2} - R_2\right)$,

where R_1 (resp. R_2) are the sum of the rank of the first (resp. second) sample when all samples are mixed and sorted.

If n and m are high enough, it is possible to show that U follows a Gaussian distribution centered in $\frac{nm+1}{2}$.

Example: "Perfect" association; the n first elements are in the first sample and the m next are in the second one.

Then:

$$R_1 = 1 + 2 + \dots + n = \frac{n(n+1)}{2}$$

$$\Rightarrow U_1 = nm + \frac{n(n+1)}{2} - R_1 = nm$$

ChAMP's representation: Kruskal-Wallis test

The Kruskal-Wallis test is a generalization of the Wilcoxon-Man-Withney test that works for two samples. They are both **non-parametric**.

The Wilcoxon-Man-Withney proposes to test the association between a continuous (ex: age) and a discrete variable (ex: sex).

Let us consider two samples (x_1, \dots, x_n) and (y_1, \dots, y_m) . They both represent the same continuous variable but are separated by the value of the discrete one.

$$\begin{cases} H_0: (x_1, \dots, x_n) \text{ and } (y_1, \dots, y_m) \text{ comes from the same distribution.} \\ H_1: \text{They do not.} \end{cases}$$

The proposed statistic is: $U = \min(U_1, U_2) = \min\left(nm + \frac{n(n+1)}{2} - R_1, nm + \frac{m(m+1)}{2} - R_2\right)$,

where R_1 (resp. R_2) are the sum of the rank of the first (resp. second) sample when all samples are mixed and sorted.

If n and m are high enough, it is possible to show that U follows a Gaussian distribution centered in $\frac{nm+1}{2}$.

Example: "Perfect" association; the n first elements are in the first sample and the m next are in the second one.

Then:

$$R_1 = 1 + 2 + \dots + n = \frac{n(n+1)}{2}$$

$$R_2 = (n+1) + (n+2) + \dots + (n+m) = \frac{m((n+1) + (n+m))}{2}$$

$$\Rightarrow U_1 = nm + \frac{n(n+1)}{2} - R_1 = nm$$

ChAMP's representation: Kruskal-Wallis test

The Kruskal-Wallis test is a generalization of the Wilcoxon-Man-Withney test that works for two samples. They are both **non-parametric**.

The Wilcoxon-Man-Withney proposes to test the association between a continuous (ex: age) and a discrete variable (ex: sex).

Let us consider two samples (x_1, \dots, x_n) and (y_1, \dots, y_m) . They both represent the same continuous variable but are separated by the value of the discrete one.

$$\begin{cases} H_0: (x_1, \dots, x_n) \text{ and } (y_1, \dots, y_m) \text{ comes from the same distribution.} \\ H_1: \text{They do not.} \end{cases}$$

The proposed statistic is: $U = \min(U_1, U_2) = \min\left(nm + \frac{n(n+1)}{2} - R_1, nm + \frac{m(m+1)}{2} - R_2\right)$,

where R_1 (resp. R_2) are the sum of the rank of the first (resp. second) sample when all samples are mixed and sorted.

If n and m are high enough, it is possible to show that U follows a Gaussian distribution centered in $\frac{nm+1}{2}$.

Example: "Perfect" association; the n first elements are in the first sample and the m next are in the second one.

Then:

$$R_1 = 1 + 2 + \dots + n = \frac{n(n+1)}{2}$$

$$\Rightarrow U_1 = nm + \frac{n(n+1)}{2} - R_1 = nm$$

$$R_2 = (n+1) + (n+2) + \dots + (n+m) = \frac{m((n+1) + (n+m))}{2}$$

$$\Rightarrow U_2 = nm + \frac{m(m+1)}{2} - R_2 = 0$$

ChAMP's representation: Kruskal-Wallis test

The Kruskal-Wallis test is a generalization of the Wilcoxon-Man-Withney test that works for two samples. They are both **non-parametric**.

The Wilcoxon-Man-Withney proposes to test the association between a continuous (ex: age) and a discrete variable (ex: sex).

Let us consider two samples (x_1, \dots, x_n) and (y_1, \dots, y_m) . They both represent the same continuous variable but are separated by the value of the discrete one.

$$\begin{cases} H_0: (x_1, \dots, x_n) \text{ and } (y_1, \dots, y_m) \text{ comes from the same distribution.} \\ H_1: \text{They do not.} \end{cases}$$

The proposed statistic is: $U = \min(U_1, U_2) = \min\left(nm + \frac{n(n+1)}{2} - R_1, nm + \frac{m(m+1)}{2} - R_2\right)$,

where R_1 (resp. R_2) are the sum of the rank of the first (resp. second) sample when all samples are mixed and sorted.

If n and m are high enough, it is possible to show that U follows a Gaussian distribution centered in $\frac{nm+1}{2}$.

Example: "Perfect" association; the n first elements are in the first sample and the m next are in the second one.

Then:

$$R_1 = 1 + 2 + \dots + n = \frac{n(n+1)}{2}$$

$$R_2 = (n+1) + (n+2) + \dots + (n+m) = \frac{m((n+1) + (n+m))}{2}$$

$$\Rightarrow U_1 = nm + \frac{n(n+1)}{2} - R_1 = nm$$

$$\Rightarrow U_2 = nm + \frac{m(m+1)}{2} - R_2 = 0$$

ChAMP's representation: Kruskal-Wallis test

The Kruskal-Wallis test is a generalization of the Wilcoxon-Man-Withney test that works for two samples. They are both **non-parametric**.

The Wilcoxon-Man-Withney proposes to test the association between a continuous (ex: age) and a discrete variable (ex: sex).

Let us consider two samples (x_1, \dots, x_n) and (y_1, \dots, y_m) . They both represent the same continuous variable but are separated by the value of the discrete one.

$$\begin{cases} H_0: (x_1, \dots, x_n) \text{ and } (y_1, \dots, y_m) \text{ comes from the same distribution.} \\ H_1: \text{They do not.} \end{cases}$$

The proposed statistic is: $U = \min(U_1, U_2) = \min\left(nm + \frac{n(n+1)}{2} - R_1, nm + \frac{m(m+1)}{2} - R_2\right)$,

where R_1 (resp. R_2) are the sum of the rank of the first (resp. second) sample when all samples are mixed and sorted.

If n and m are high enough, it is possible to show that U follows a Gaussian distribution centered in $\frac{nm+1}{2}$.

Example: "Perfect" association; the n first elements are in the first sample and the m next are in the second one.

Then:

$$R_1 = 1 + 2 + \dots + n = \frac{n(n+1)}{2}$$

$$R_2 = (n+1) + (n+2) + \dots + (n+m) = \frac{m((n+1) + (n+m))}{2}$$

$$\left. \begin{aligned} \Rightarrow U_1 &= nm + \frac{n(n+1)}{2} - R_1 = nm \\ \Rightarrow U_2 &= nm + \frac{m(m+1)}{2} - R_2 = 0 \end{aligned} \right\} \Rightarrow U = 0$$

ChAMP's representation: Kruskal-Wallis test

The Kruskal-Wallis test is a generalization of the Wilcoxon-Man-Withney test that works for two samples. They are both **non-parametric**.

The Wilcoxon-Man-Withney proposes to test the association between a continuous (ex: age) and a discrete variable (ex: sex).

Let us consider two samples (x_1, \dots, x_n) and (y_1, \dots, y_m) . They both represent the same continuous variable but are separated by the value of the discrete one.

$$\begin{cases} H_0: (x_1, \dots, x_n) \text{ and } (y_1, \dots, y_m) \text{ comes from the same distribution.} \\ H_1: \text{They do not.} \end{cases}$$

The proposed statistic is: $U = \min(U_1, U_2) = \min\left(nm + \frac{n(n+1)}{2} - R_1, nm + \frac{m(m+1)}{2} - R_2\right)$,

where R_1 (resp. R_2) are the sum of the rank of the first (resp. second) sample when all samples are mixed and sorted.

If n and m are high enough, it is possible to show that U follows a Gaussian distribution centered in $\frac{nm+1}{2}$.

Example: "Perfect" association; the n first elements are in the first sample and the m next are in the second one.

Then:

$$R_1 = 1 + 2 + \dots + n = \frac{n(n+1)}{2}$$

$$R_2 = (n+1) + (n+2) + \dots + (n+m) = \frac{m((n+1) + (n+m))}{2}$$

$$\left. \begin{aligned} \Rightarrow U_1 &= nm + \frac{n(n+1)}{2} - R_1 = nm \\ \Rightarrow U_2 &= nm + \frac{m(m+1)}{2} - R_2 = 0 \end{aligned} \right\} \Rightarrow U = 0$$

The test is likely to be rejected.