Comp790-166: Computational Biology

Lecture 20

April 4, 2023

Today

- Finish graph alignment clean-up with Refina
- Multiomics data integration with MOFA

Quick Announcements

- Homework 2 is now available online and due **Friday April 21.** → `https://github.com/natalies-teaching/CompBio2023/tree/main/Homework2_2023`
- Final project presentations will be on the last two days of class (April 24 and April 26). Note that you can just give us an update. The writeup will be due on our Final Exam Day (May 1).

Review Questions

1. How was the structural information about the graph encoded in REGAL?
2. What was one of the key innovations , which made REGAL more efficient?

Calculating Node Similarity Within and Between Graphs

Define distance between nodes u and v in terms of structure ($\mathbf{d}$), or attributes ($\mathbf{f}$) as,

$$\text{sim}(u, v) = \exp\left[-\gamma_s \cdot \|\mathbf{d}_u - \mathbf{d}_v\|_2^2 - \gamma_a \cdot \text{dist}\left(\mathbf{f}_u, \mathbf{f}_v\right)\right]$$

Each $\mathbf{d}_u$ is defined as,

$$\mathbf{d}_u = \sum_{k=1}^{K} \delta^{k-1} \mathbf{d}_u^k$$

Here, δ is a discount factor for greater hop distances.

Approximation for the Embedding Matrix, $\mathbf{Y}$

Theorem: Given graphs, $\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2(\mathcal{V}_2, \mathcal{E}_2)$ with an $n \times n$ joint combined structural and attribute-based similarity matrix $\mathbf{S} \approx \mathbf{Y}\mathbf{Z}^T$, its node embedding matrix $\mathbf{Y}$ can be approximated as,

$$\tilde{Y} = CU\Sigma^{-1/2}. \tag{1}$$

Here, $\mathbf{C}$ is the $n \times p$ matrix of similarities between the n nodes and the p chosen landmarks and $\mathbf{W}^\dagger = \mathbf{U}\Sigma\mathbf{V}^T$ is the full rank SVD of the pseudoinverse of the small $p \times p$ landmark similarity matrix matrix, $\mathbf{W}^\dagger$.

Matched Neighborhood Consistency (MNC)

- **Neighbors of node i in G_1** $\rightarrow \mathcal{N}_{G_1}(i) = \{j \in \mathcal{V}_1 : (i,j) \in \mathcal{E}_1\}$
- **Mapped Neighborhood:** The set of nodes onto which π maps i's neighbors.
  - $\tilde{\mathcal{N}}^{\pi}_{G_2}(i) = \{j \in \mathcal{V}_2 : \exists k \in \mathcal{N}_{G_1}(i) \text{ s.t. } \pi(k) = j\}$

Then, given a node i in G_1 and a node j in G_2, define the neighborhood consistency as,

$$\text{MNC}(i,j) = \frac{\left|\tilde{\mathcal{N}}^{\pi}_{G_2}(i) \cap \mathcal{N}_{G_2}(j)\right|}{\left|\tilde{\mathcal{N}}^{\pi}_{G_2}(i) \cup \mathcal{N}_{G_2}(j)\right|}$$

Rewriting the MNC

Recall the MNC (matched neighborhood consistency) between the graphs. This can we rewritten in matrix form, such that $\text{MNC}(i,j) = \mathbf{S}_{ij}^{\text{MNC}}$ as,

$$\mathbf{S}^{MNC} = \mathbf{A}_1\mathbf{M}\mathbf{A}_2 \oslash (\mathbf{A}_1\mathbf{M}\mathbf{1}^{n_2} \otimes \mathbf{1}^{n_2} + \mathbf{1}^{n_1} \otimes \mathbf{A}_2\mathbf{1}^{n_2} - \mathbf{A}_1\mathbf{M}\mathbf{A}_2)$$

Here,

- $\mathbf{M}$ is a binary alignment matrix
- $\oslash$ is element-wise division
- $\otimes$ is outerproduct

* aka writing Jaccard similarity in matrix format

Computing a Refined Alignment

Compute refined alignments, $\mathbf{M}'$ by multiplicative updating each node's alignment score (in $\mathbf{M}$) with its matched neighborhood consistency as,

$$\mathbf{M}' = \mathbf{M} \circ \mathbf{S}^{\mathrm{MNC}}$$

Here, $\circ$ is Hadamard product.

- The proposed refinement score should iteratively increase alignment scores for nodes that have high MNC.

Modifying Updates with Some Important Intuition

- **Higher degree nodes are easier to align**. The part of MNC we care about is counting nodes' matched neighbors. This simplifies the update rule to,

$$\mathbf{M}' = \mathbf{M} \circ \mathbf{A}_1\mathbf{M}\mathbf{A}_2$$

Now the MNC update is simply,

$$\mathbf{A}_1\mathbf{M}\mathbf{A}_2$$

.

Intuition, Continued

- **Do Not Rely Too Much on Initial $\mathbf{M}_0$.** Add a small ϵ at each iteration to correct for false negatives.
- **Normalization:** To encourage performance and to keep **M** from exploding, row normalize **M**, followed by column normalization at every iteration.

Summary

Algorithm 1 RefiNA $(\mathbf{A}_1, \mathbf{A}_2, \mathbf{M}_0, K, \epsilon)$

Input: adjacency matrices $\mathbf{A}_1, \mathbf{A}_2$, initial alignment matrix $\mathbf{M}_0$, number of iterations K, token match score ϵ
for $k = 1 \rightarrow K$ **do** ▷ Refinement iterations
    $\mathbf{M}_k = \mathbf{M}_{k-1} \circ \mathbf{A}_1\mathbf{M}_{k-1}\mathbf{A}_2$ ▷ MNC update
    $\mathbf{M}_k = \mathbf{M}_k + \epsilon$ ▷ Add token match scores
    $\mathbf{M}_k = \text{Normalize}(\mathbf{M}_k)$ ▷ By row then column
end for
return $\mathbf{M}_K$

Figure: from Algorithm 1

Question

Which order neighborhood should be considered? It seems that the authors only consider immediate neighbors. Do you think there is benefit to considering higher order neighborhoods?

Making RefiNA Sparse

- To scale RefiNA to large graphs, the authors focus on updating a small number of alignment scores for each node.
- Intuitively, forgo updating scores of likely unaligned node pairs.
- The solution is to only update some entries of $\mathbf{M}$.

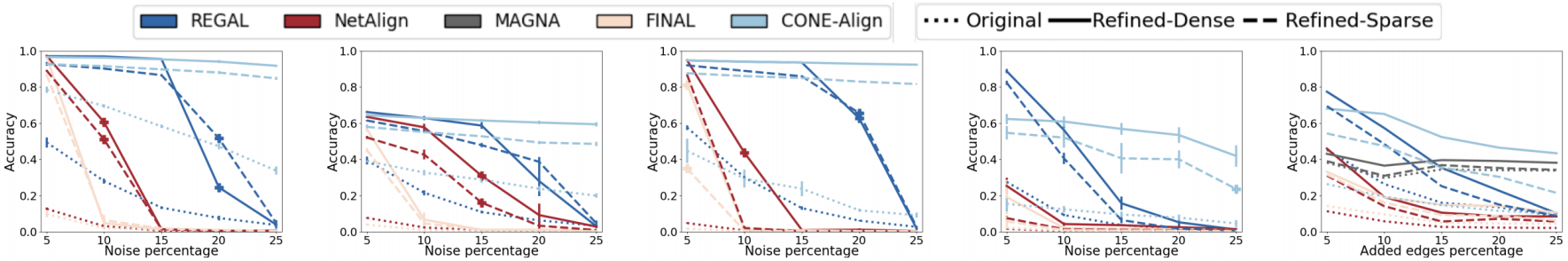
Specifically the updates to $\mathbf{M}$ are modified as,

$$\mathbf{M}_{k|\mathbf{U}_k} = \mathbf{M}_{k-1|\mathbf{U}_k} \circ \mathbf{U}_k$$

- $\mathbf{U}_k = \text{top} -\alpha\left(\mathbf{A}_1\mathbf{M}\mathbf{A}_2\right)$ is only the top α entries per row.
- $\mathbf{M}_{k|\mathbf{U}_k}$ is only the non-zero elements in $\mathbf{U}_k$

Noise Experiments

Create a permuted copy of $\mathbf{A}$, $\tilde{\mathbf{A}} = \mathbf{PAA}^T$. Then remove edges with some probability, p_s.



(a) Arenas (b) Hamsterster (c) PPI-H (d) Facebook (e) PPI-Y

Figure: from Fig. 2. Refinement with the sparse or dense formulation helps networks with different structures.

Run-time

Notice REGAL is faring pretty well (even with the dense formulation...)

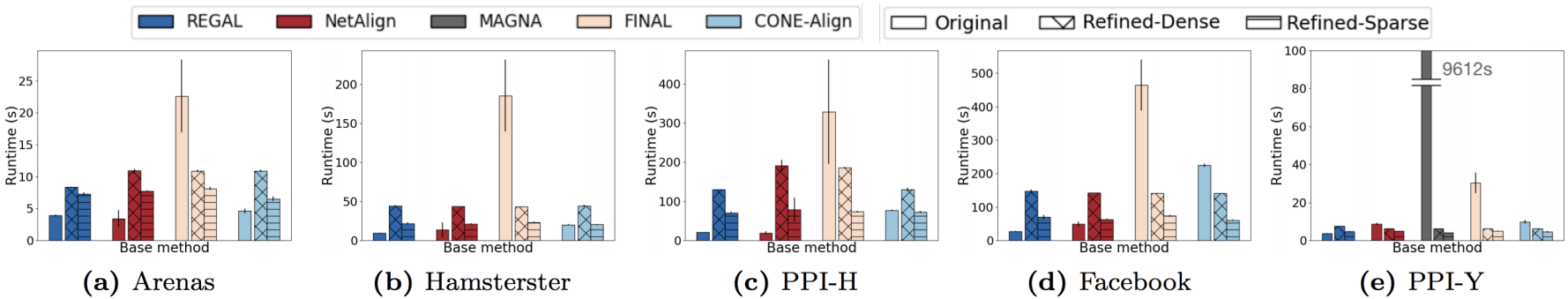


Figure: from Fig. 4.

Performance Based on a Binned Degree Distribution

Nodes were binned by degree into $\{low, medium, high\}$. The MNC of these nodes were further visualized based on accuracy.

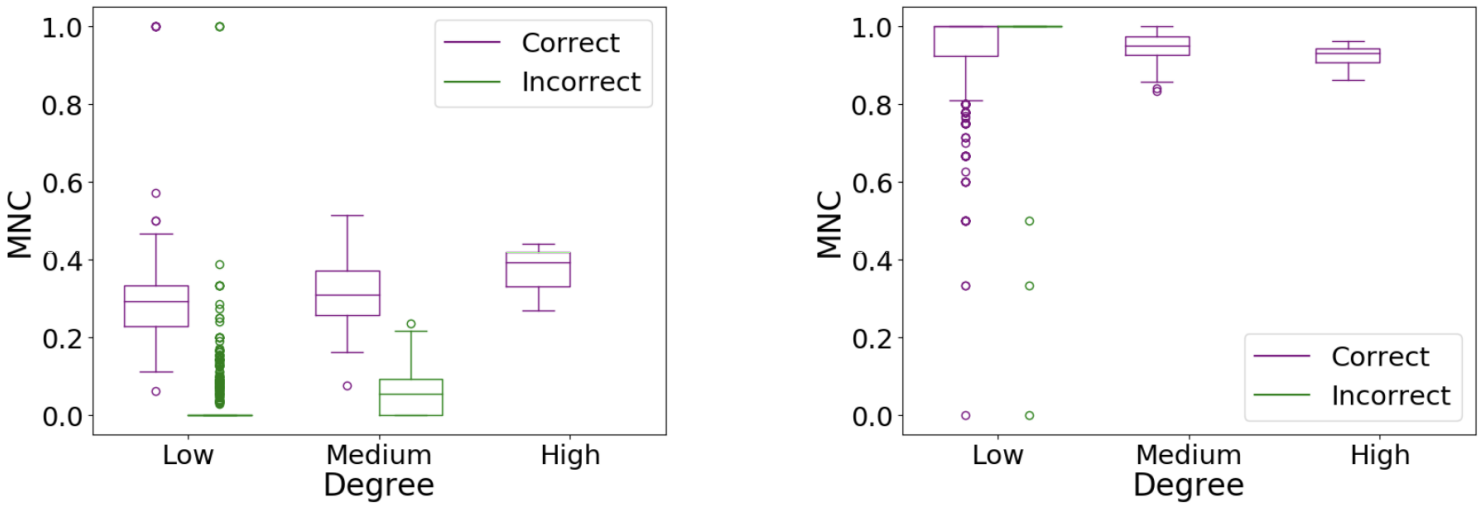


Figure: from Fig. 5

RefiNA Conclusion

- RefiNA is a posthoc method to clean up a network alignment.
- You can start with an alignment generated with any algorithm (though REGAL looks good!)
- You have the ability to make the updates more sparse by zeroing out entries in $\mathbf{M}$

Another Approach for Multiomics Data Integration

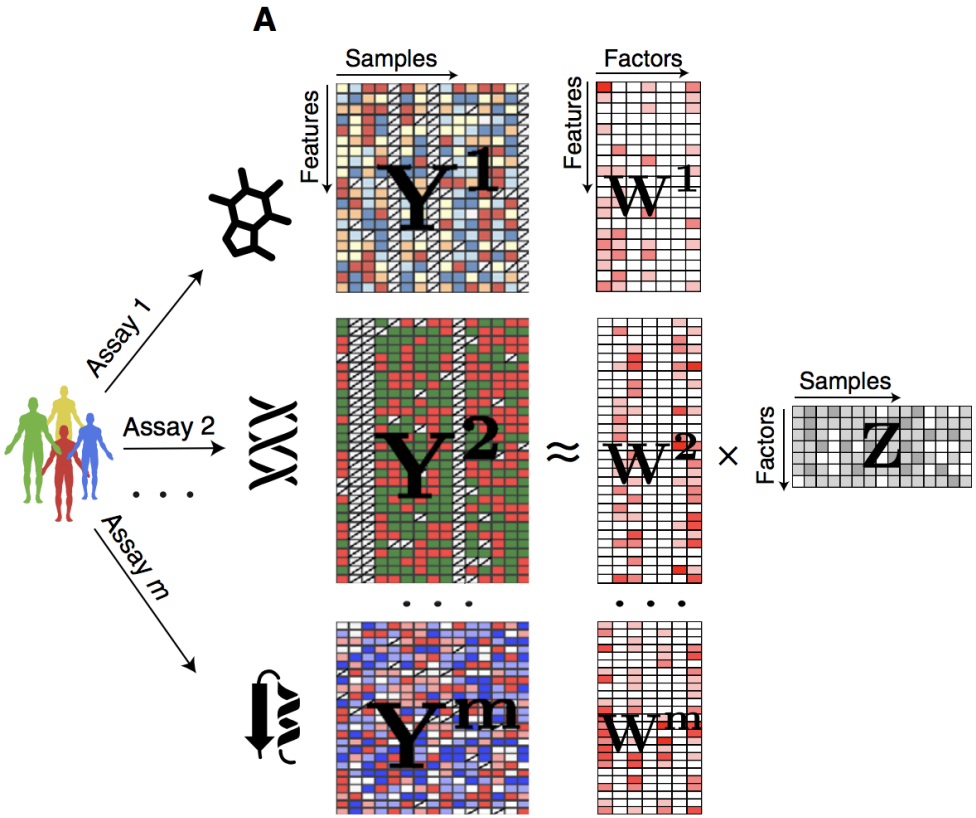


Figure: from Argelaguet *et al.* Molecular Systems Biology. 2018.

Complaints by MOFA Authors

The MOFA authors sought to develop a method that facilitates the following.

- **Interpretability:** The goal is to reconstruct the underlying ‘factors‘ that drive variation across samples in each modality.
- **Biological vs Technical Variation:** Should effectively untangle biological vs technical variation
- **Decomposing Variability:** Identify particular features and particular ‘factors‘ driving significant variance within and between modalities.

Notation and Goals

- Start with M data matrices, $\mathbf{Y}^1, \mathbf{Y}^2, \ldots, \mathbf{Y}^m$ on a set of N samples
- A Particular modality's matrix, $\mathbf{Y}_m$ has dimensions, $N \times D_m$

The goal is to write the input data matrix, $\mathbf{Y}^m$ as a product of a common factor matrix, $\mathbf{Z}$ (factors $\times$ samples) and a 'wight' matrix (e.g. one that relates features to factor space) as,

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

Unpacking...

Assuming there are k factors and given

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

,

- $\mathbf{Z} \in \mathbb{R}^{N \times K}$ relate the original samples to the factors
- $\mathbf{W}^m \in \mathbb{R}^{D_m \times K}$ relates the original features to the factors

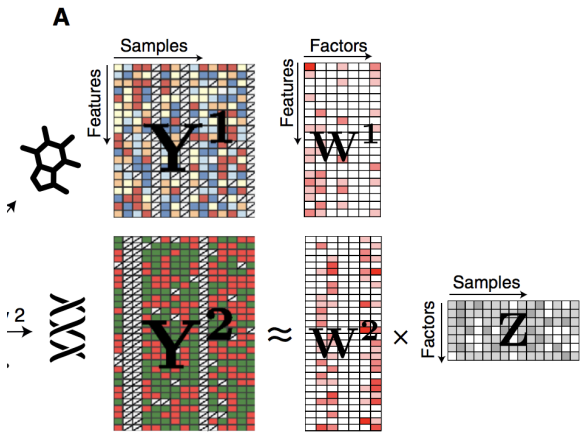


Figure: from Argelaguet *et al.* Molecular Systems Biology. 2018.

Overview of the Model

This is latent variable model and feature dependencies are attempted to be explained in terms of k latent classes (or 'factors').

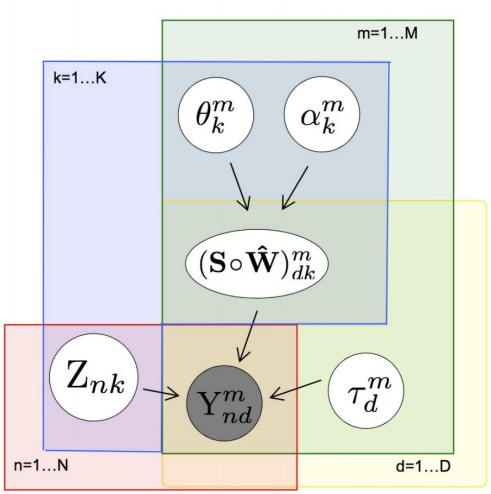


Figure: from Argelaguet *et al.* Molecular Systems Biology. 2018. As always, gray nodes are observed variables and white notes are the unobserved variables inferred by the model.

Starts Simple Enough....

The the observed value of feature d in sample n from modality m is modeled as,

$$y_{nd}^m \sim \mathcal{N}(y_{nd}^m \mid \mathbf{z}_{n,:}\mathbf{w}_{d,:}^{mT}, 1/\gamma_d^m)$$

- $\mathbf{w}_{d,:}^m$ gives the d-th row of $\mathbf{W}^m$
- $\mathbf{z}_{n,:}$ is the n-th row of the latent factor matrix, $\mathbf{Z}$.

Regularization in $\mathbf{W}^m$

In computing each $\mathbf{W}^m$ there two desired kinds of sparsity :

- View and factor-wise sparsity
- Feature-wise sparsity

The intuition is that $\mathbf{w}^m_{:,k}$ is shrunk to 0 if factor k does not drive any variation in view m.

In practice...

Sparsity is enforced for each $\mathbf{W}$ through appropriate priors. Specifically, they model $\mathbf{W}$ as a product of a Gaussian random variable, $\hat{w}$, and a Bernoulli random variable, s.

$$\mathbf{W} = \mathbf{S}\hat{\mathbf{W}}$$

- $s_{dk}^m \sim \text{Bernoulli}(s_{d,k}^m \mid \theta_k^m)$
- $\hat{W}_{dk}^m \sim \mathcal{N}(0, 1/\alpha_k)$

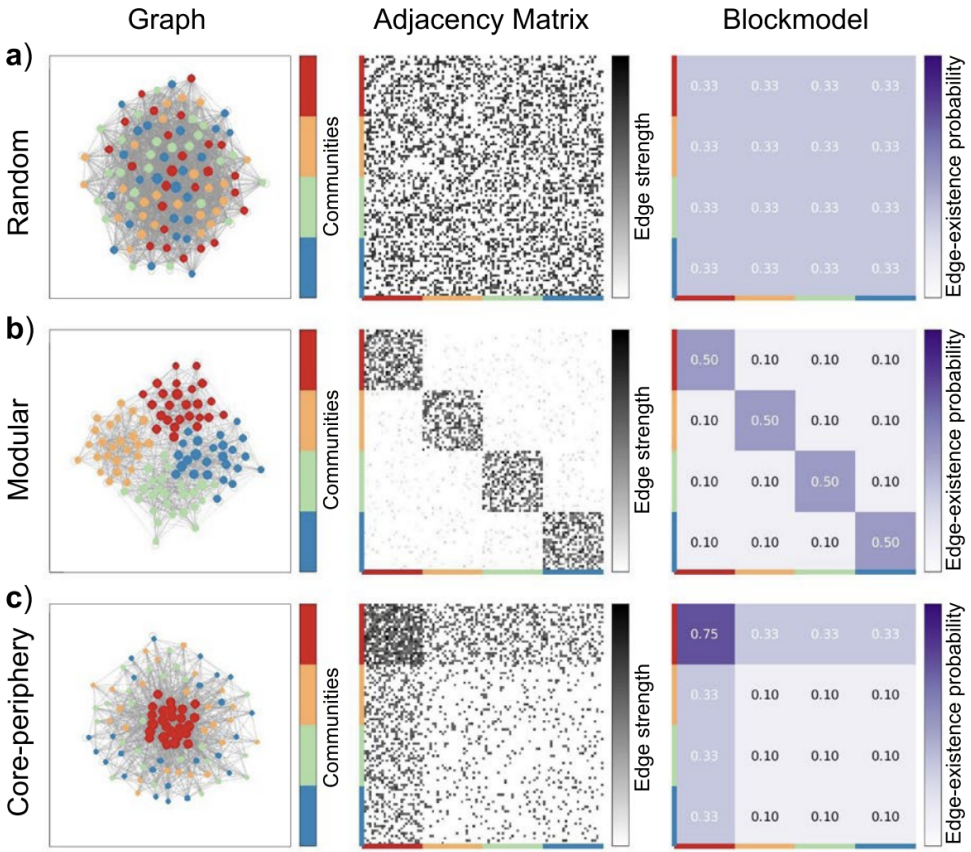
Thinking More About Sparsity

Let's think about what happens with bernoulli probabilities:

- A value of θ_k^m close to 0 implies that most of the weights of factor k in view m tend towards 0.
- A θ_k^m close to 1 alternatively implies that most of the weights are non-zero (e.g. a non-sparse factor)

Further, there are priors for $\alpha_k^m \sim$ Beta and $\sigma_k^m \sim \Gamma$

Living in the world of latent variables... remember stochastic block model?



Latent variable: node-to-community assignments

$$\log \mathcal{L}(\mathcal{X}, \mathcal{Z}) = \sum_i \sum_q Z_{iq} \log \alpha_q + \frac{1}{2} \sum_{i \neq j} \sum_{q,\ell} Z_{iq} Z_{j\ell} \log b\left(X_{ij}; \pi_{q\ell}\right)$$

- At some point in updating parameters via EM, we need to worry about the posterior or $P(\mathcal{Z} \mid \mathcal{X})$.

Dealing with Latent Variables

More help with factorized approximations...

- Our situation is the case of having an intractable posterior distribution of unobserved variables $p(\mathbf{X} \mid \mathbf{Y})$.
- Here, $\mathbf{X}$ is denoting our un-observed variables, and $\mathbf{Y}$ is denoting the observed variables

We can deal with $p(\mathbf{X} \mid \mathbf{Y})$ by approximating it with a factorized form,

$$q(\mathbf{X}) = \prod_i q(\mathbf{X}_i)$$

.

This can make inference more efficient.

Reminder About Latent Variables Here

Notably, $\mathbf{Z}$ and $(\mathbf{S} \circ \hat{\mathbf{W}})$

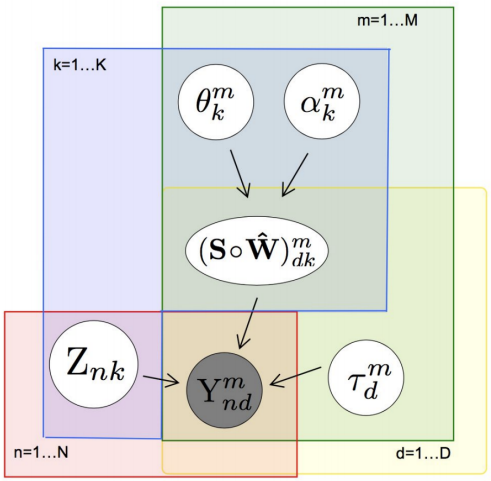


Figure: from Argelaguet *et al.* Molecular Systems Biology. 2018. As always, gray nodes are observed variables and white notes are the unobserved variables inferred by the model.

For MOFA : Mean Field Approximation

The assumption under the mean-field approach is that $q(\mathbf{X})$ factorized over M disjoint groups of variables as,

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

.

In the case of MOFA,

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

Update via EM - $q(\mathbf{Z})$

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

where

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

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

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

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

Moving into Results: Summary

After having inferred $\mathbf{W}^m s$ and $\mathbf{Z}$, several things can be done, such as, imputation, annotation of factors, etc.

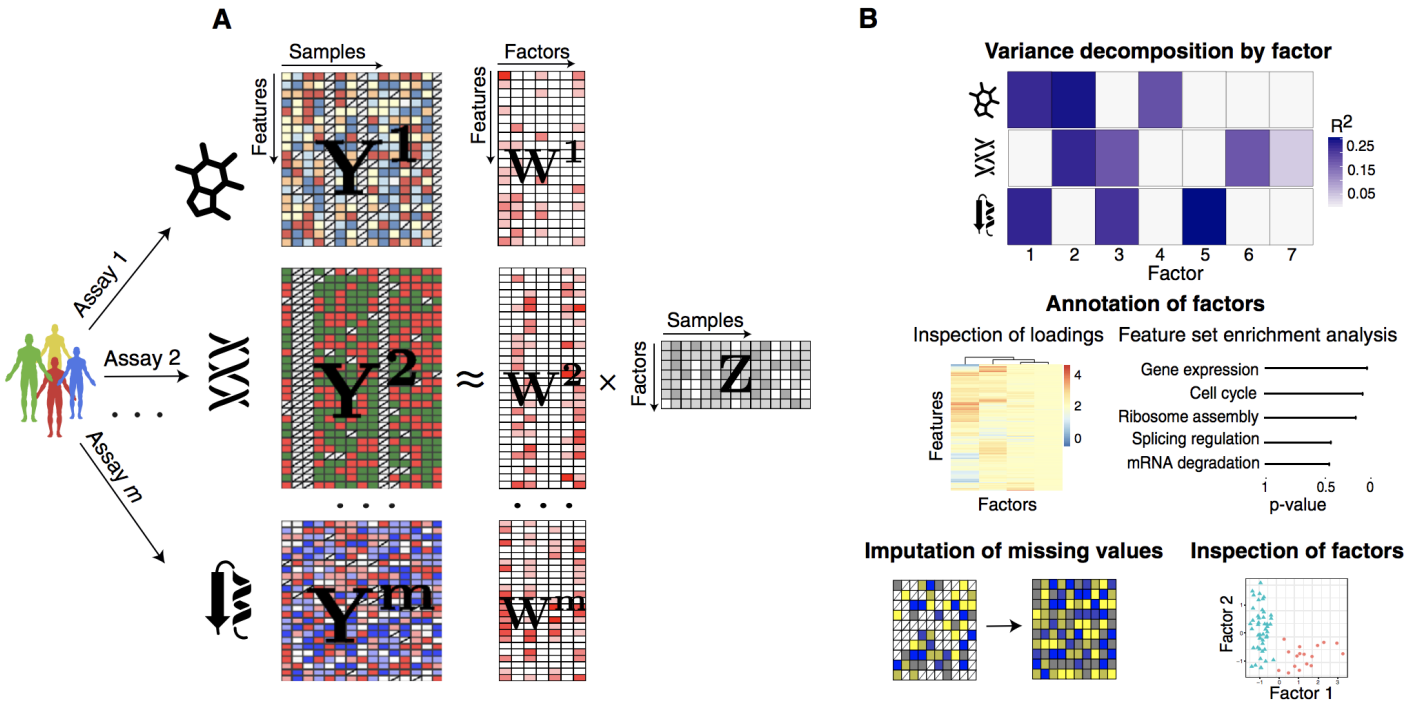


Figure: from Argelaguet *et al.* Molecular Systems Biology. 2018.

Getting Intuition About Variance Explained by Factors

- **Variance Explained by Factor k in View m :** Compute the fraction of variance explained by factor k in view m as,

$$R^2_{m,k} = 1 - \left(\sum_{n,d} y^m_{nd} - z_{nk} w^m_{kd} - \mu^m_d \right)^2 \Big/ \left(\sum_{n,d} y^m_{nd} - \mu^m_d \right)^2$$

Here, μ_d is the feature-wise mean.

Variance Explained per Modality

- **Variance Explained In Each Modality Over All Features** : Compute the fraction of variance explained per view as,

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

Here, μ_d^m is the feature-wise mean.

Results in Dataset 1 : CLL dataset

Find data here
`https://rdrr.io/github/bioFAM/MOFAdata/man/CLL_data.html`

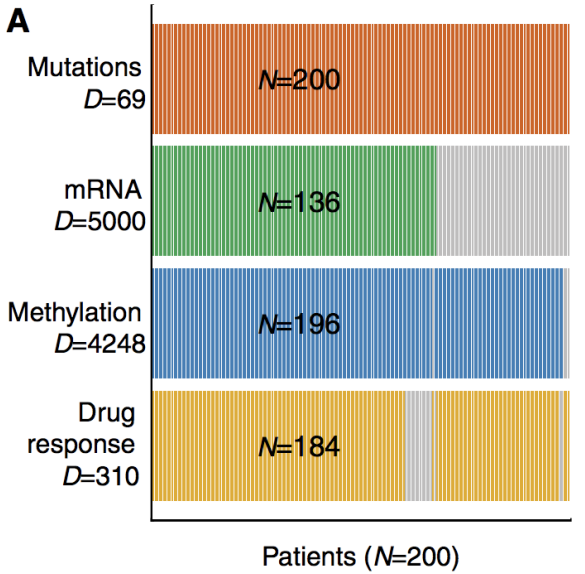


Figure: from Argelaguet *et al.* Molecular Systems Biology. 2018. Modalities and present/missing features for each patient.

Visualization of Samples

Samples can be projected into two dimensions based on the first two factors inferred in the matrix, **Z**

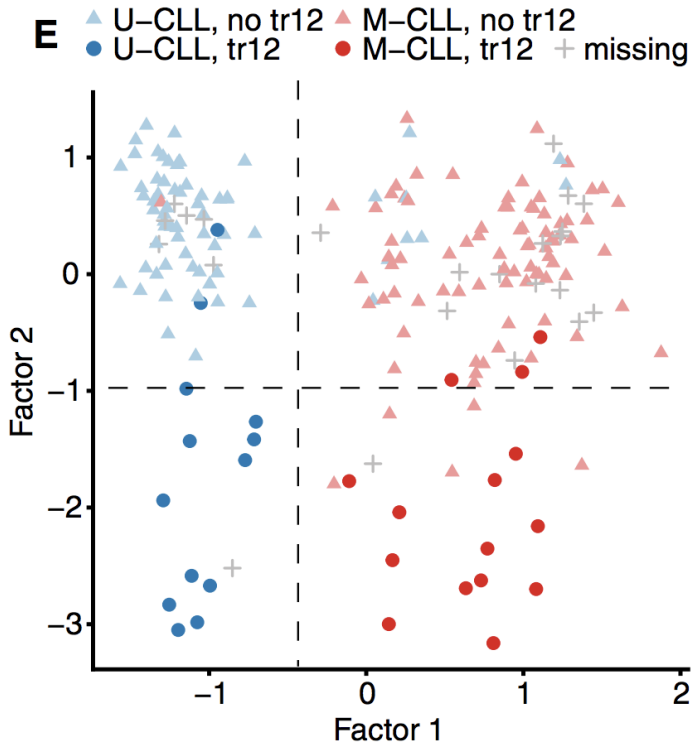


Figure: colors denote something related to the status of the tumor. Shape indicates chromosome 12 trisomy status.

Jointly Predicting Clinical Outcome in Contrast to a Single Type of Data

Experiment :

- Predict time to treatment for $N = 174$ patients using multivariate Cox regression trained using the 10 factors from MOFA
- Compare this performance on the prediction accuracy on individual modalities.
    - In this case, reduce each individual modalities to the top 10 PCs.

Results Predicting Time to Treatment

Note that the Y-axis is simply a statistic reflecting goodness of fit between true and predicted times to treatment.

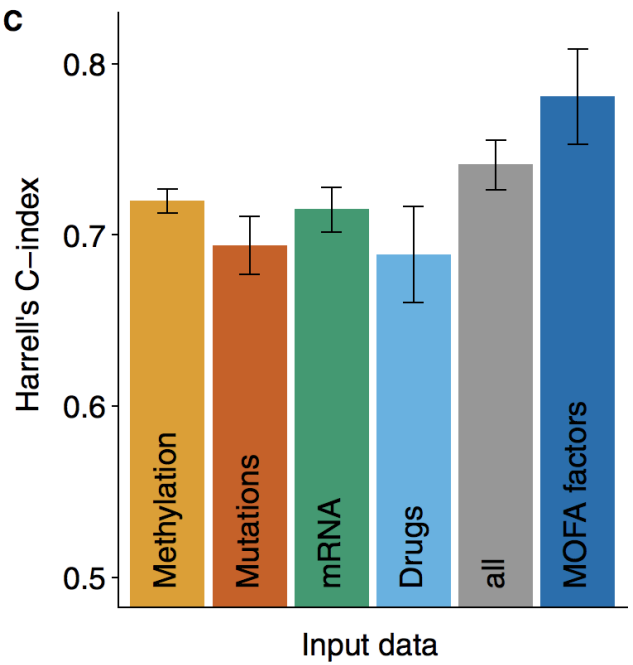


Figure: from Fig. 4 of Argelaguet *et al.* Molecular Systems Biology. 2018.

Instead of using PC Representations for Each Modality

Instead of using PC representations for each modality, the authors also compared to prediction with all features from each individual modality. Again, predicting time to treatment.

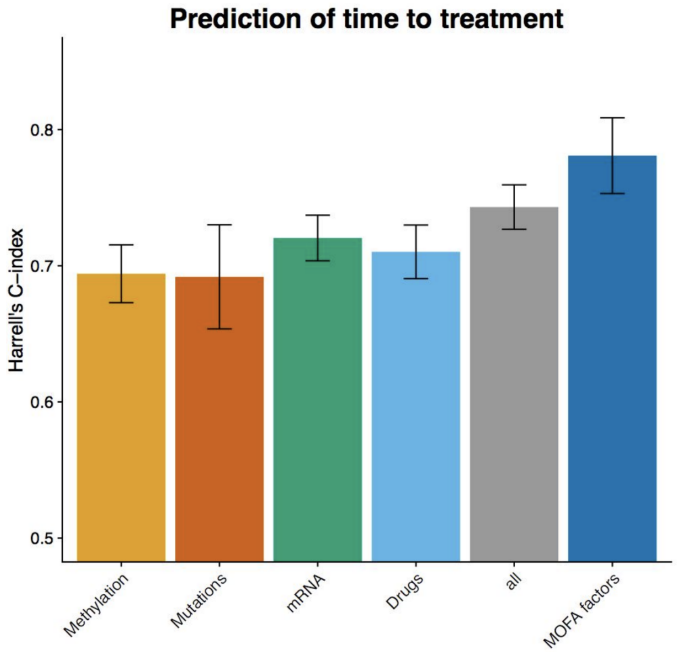


Figure: from Supp. Figure 20. Using all features still does not beat performance based on MOFA factors.

Certain Combinations of Modalities Can Turn out to be More Effective

Applying MOFA with individual modalities held out reveals sometimes certain combinations of modalities is more effective than just throwing everything in all together!

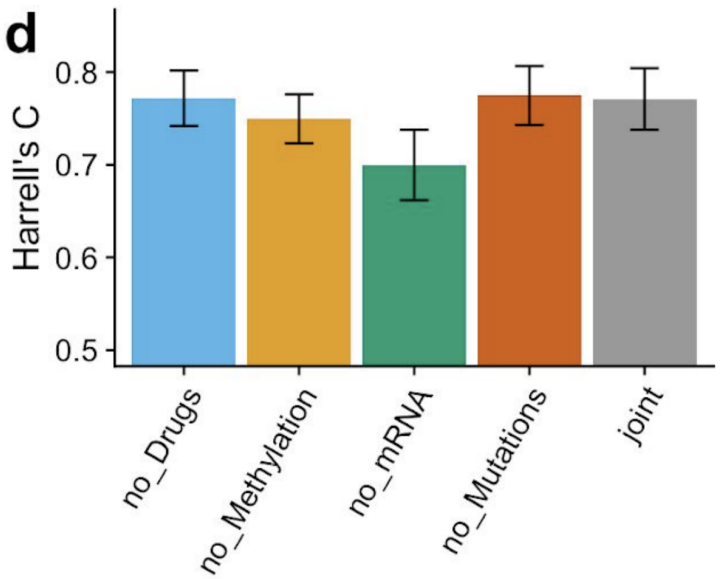


Figure: from Supplementary Figure 8.

Survival Analysis According to Particular Factors

Survival probability vs time-to-treatment for the individual MOFA factors

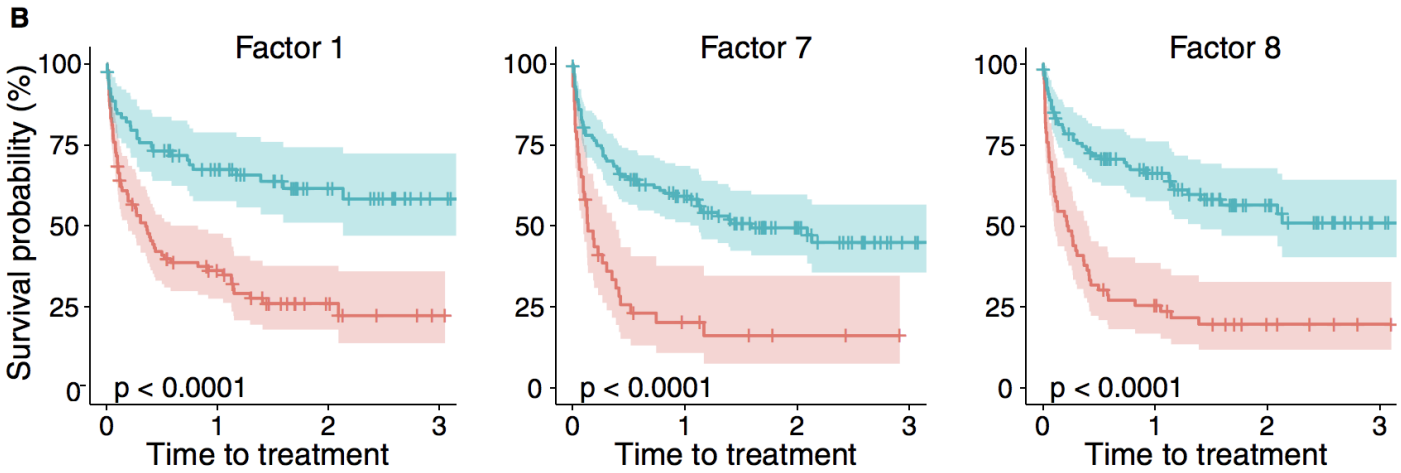


Figure: from Fig. 4. We can see distinct patterns based on survival probability against predicted time to next treatment.

Dataset 2 : Single-Cell Data for a Differentiation Trajectory

Data from a single-cell multiomics dataset. This is applied to 87 mouse embryonic stem calls, with 16 that were cultured in ‘2i1‘ media, giving them a naive pluripotent state. The remaining cells were serum grown and hence committed to a differentiation trajectory.

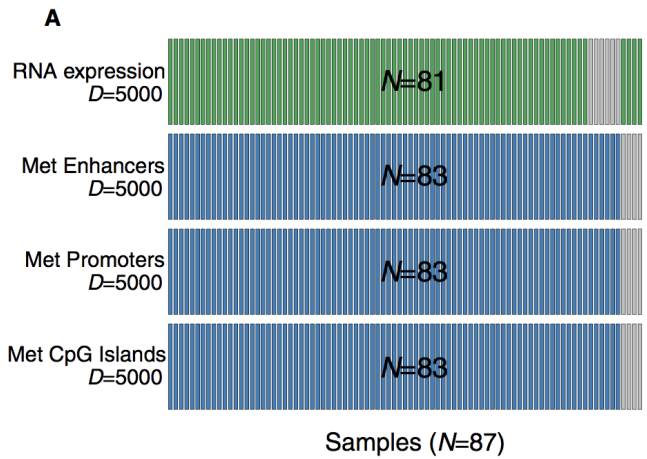
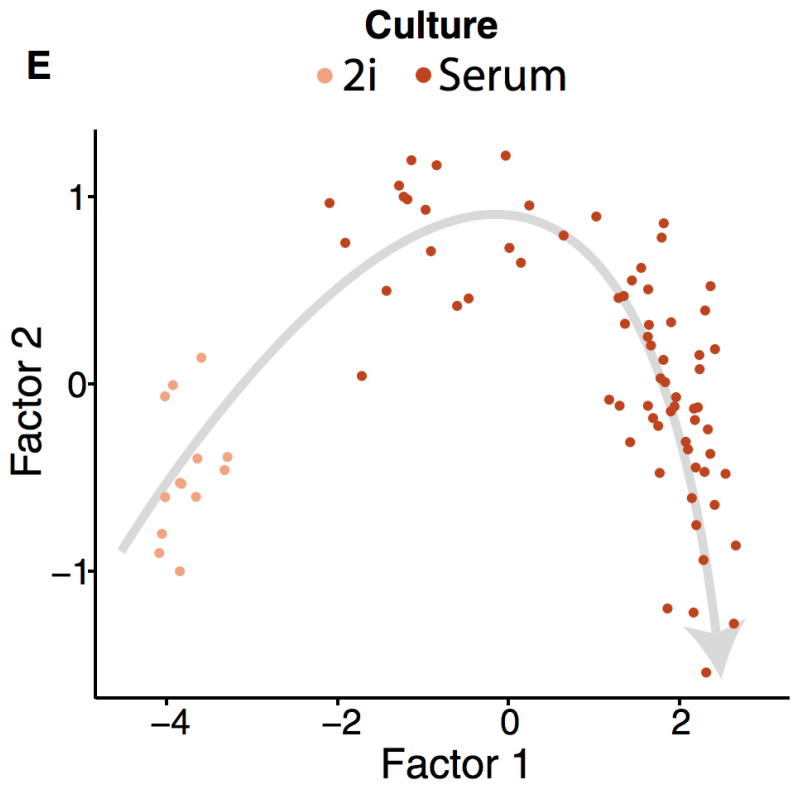


Figure: from Fig. 5. Distribution of samples profiled in each modality.

Visualizing Cells Based on **Z**

Using **Z** to plot cells based on the first two factors, the cells separate according to how they were cultured (which also corresponds to differentiation status!)



Inferred Factors on Genes and How They Relate to Pluripotency and Differentiation

Using the $\mathbf{W}^m s$ for the mRNA data, the authors compared the ranks of genes involved in pluripotency and differentiation, respectively to the loadings.

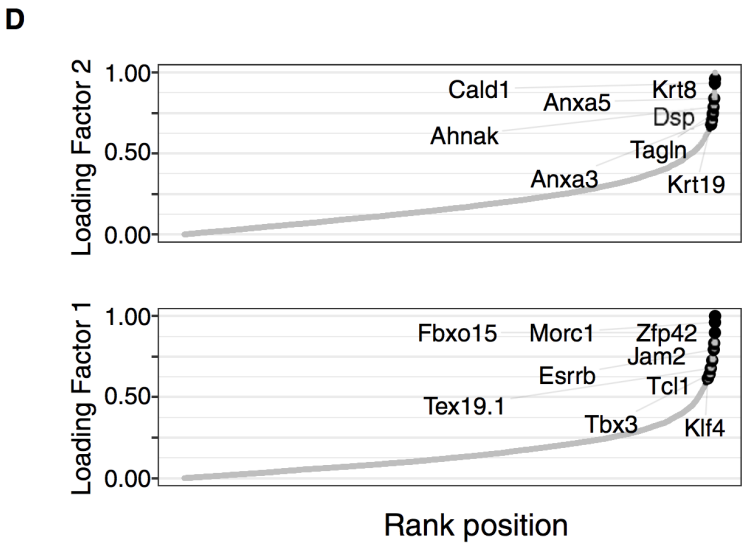


Figure: from Fig. 5. Pluripotency genes (top) vs differentiation related genes (bottom)

An Extension, MOFA+

A trivial extension....almost the same, except now we learn a $\mathbf{Z}$ per group.

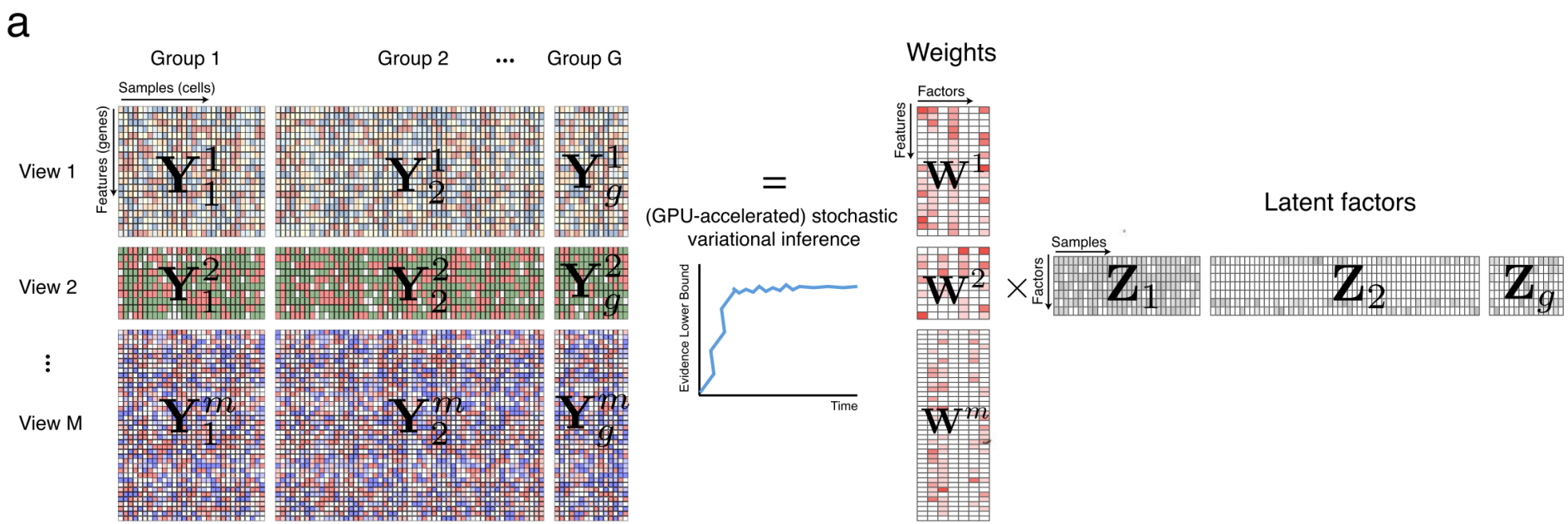


Figure: from Argalaguet *et al.* Genome Biology. 2020

So what do we think? Does anyone have any ideas for when this would be useful for your application?

Defining Y_{gm}

Similar to what we have seen with the regular MOFA,

$$Y_{gm} = Z_g W_m^T + \epsilon_{gm}$$

- Y_{gm} is the matrix of observations for the mth modality and gth group
- W_m is the weight matrix for the mth modality
- Z_g is the factor matrix for the gth group
- ϵ_{gm} is the residual noise for the mth modality in the gth group

Summary

- MOFA for decomposing a sample x feature matrix to (feature x factor)(factor x sample)
- MOFA+ for doing this calculation per group