

# Models for multimodal data integration



Britta Velten  
Heidelberg University  
<https://velten-group.org/>



CHARLES  
UNIVERSITY



SORBONNE  
UNIVERSITÉ



UNIVERSITÄT  
HEIDELBERG  
ZUKUNFT  
SEIT 1386



UNIVERSITY  
OF WARSAW



UNIVERSITÀ  
DEGLI STUDI  
DI MILANO



EUROPEAN  
UNIVERSITY  
ALLIANCE

# Organisational

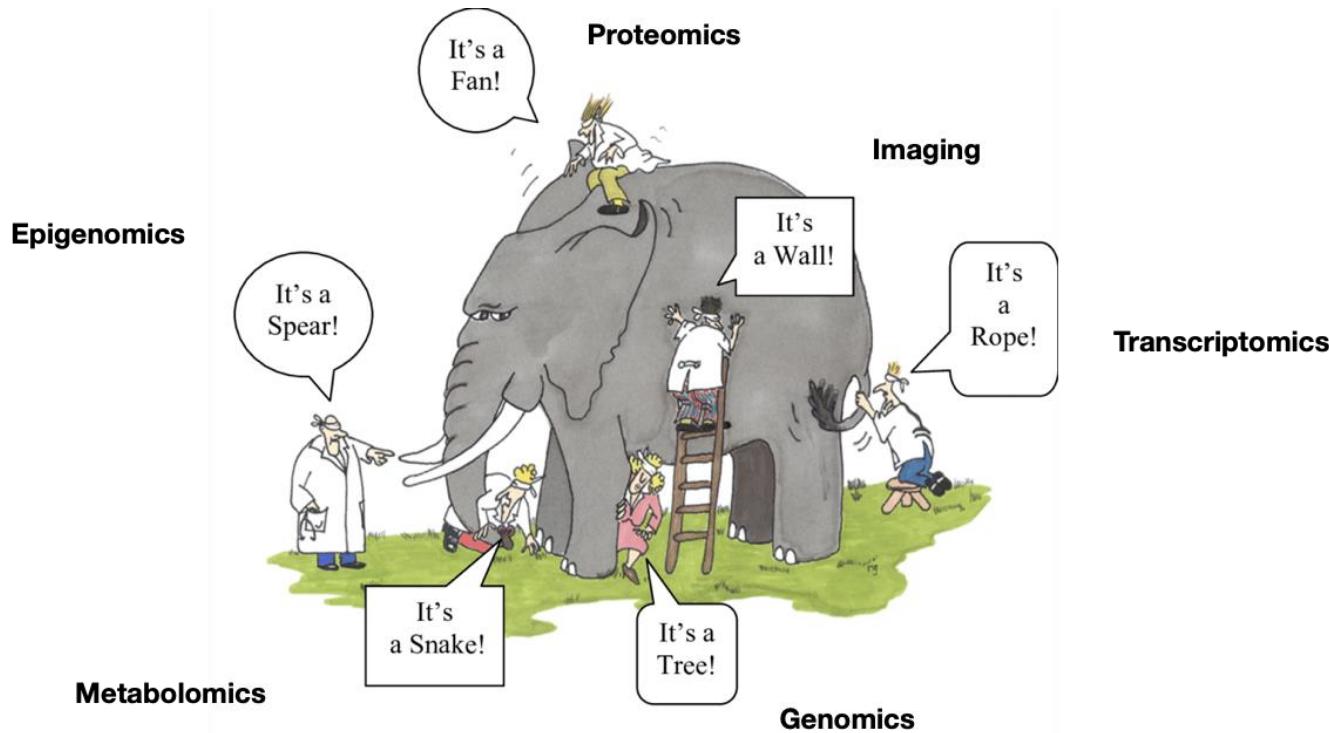
Tutorial: see link on DeepLife Homepage

*Contact person: Jana Braunger ([jana.braunger@cos.uni-heidelberg.de](mailto:jana.braunger@cos.uni-heidelberg.de))*

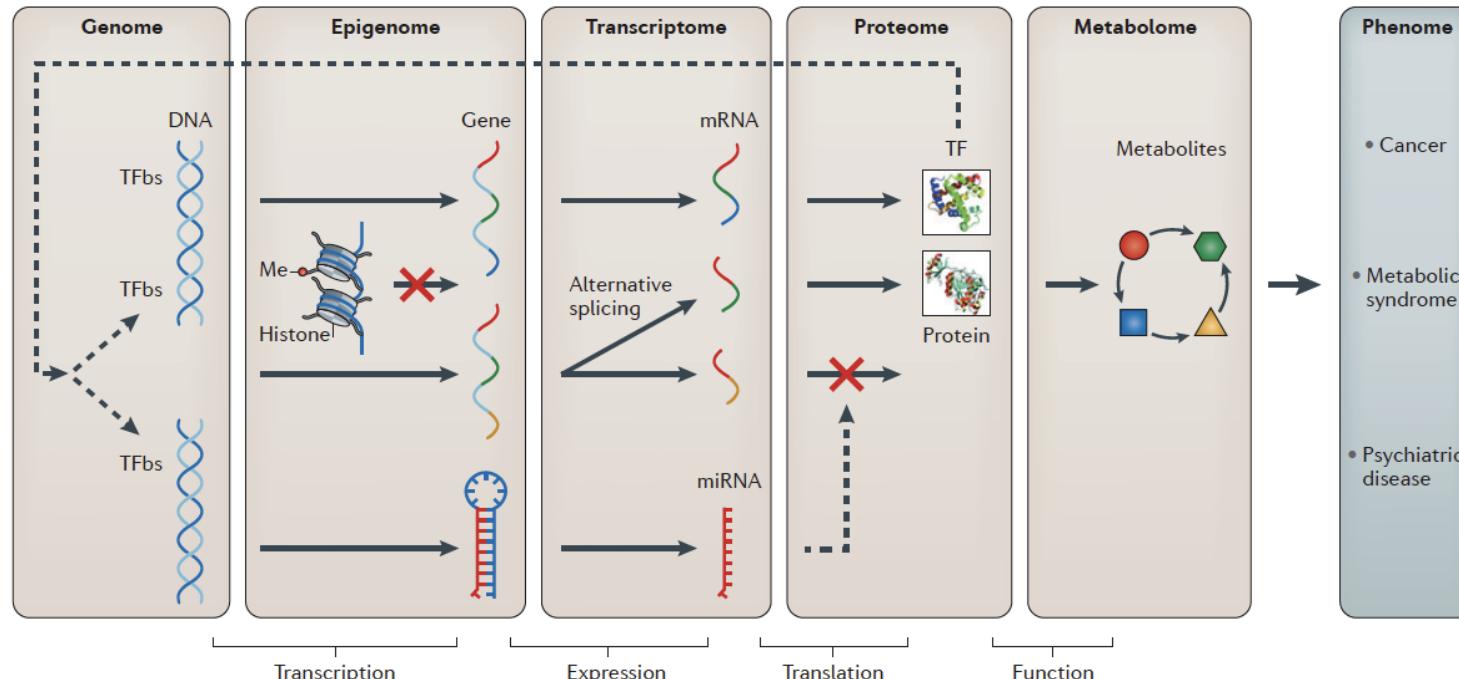
Debrief session: **March 3, 4.30 to 5pm**

# What are multi-modal data and why are they important?

# Limitations of single omics

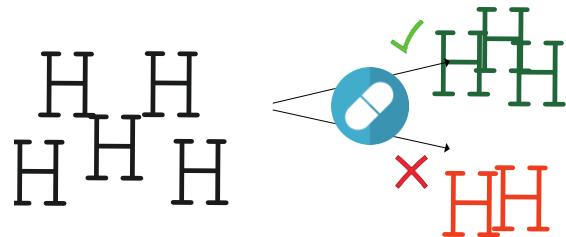


# Interactions between molecular layers

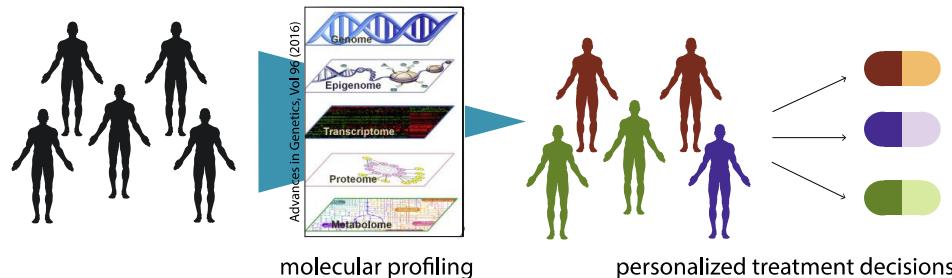


# Example 1: Precision Medicine

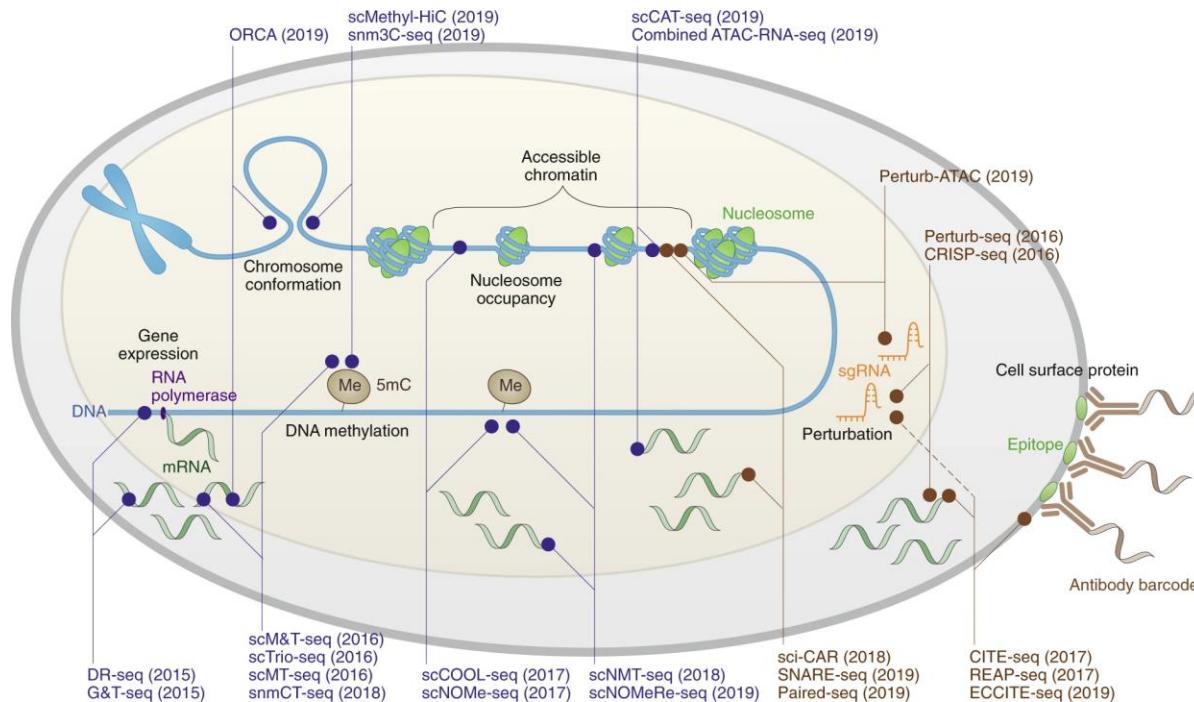
Challenge: Patient heterogeneity



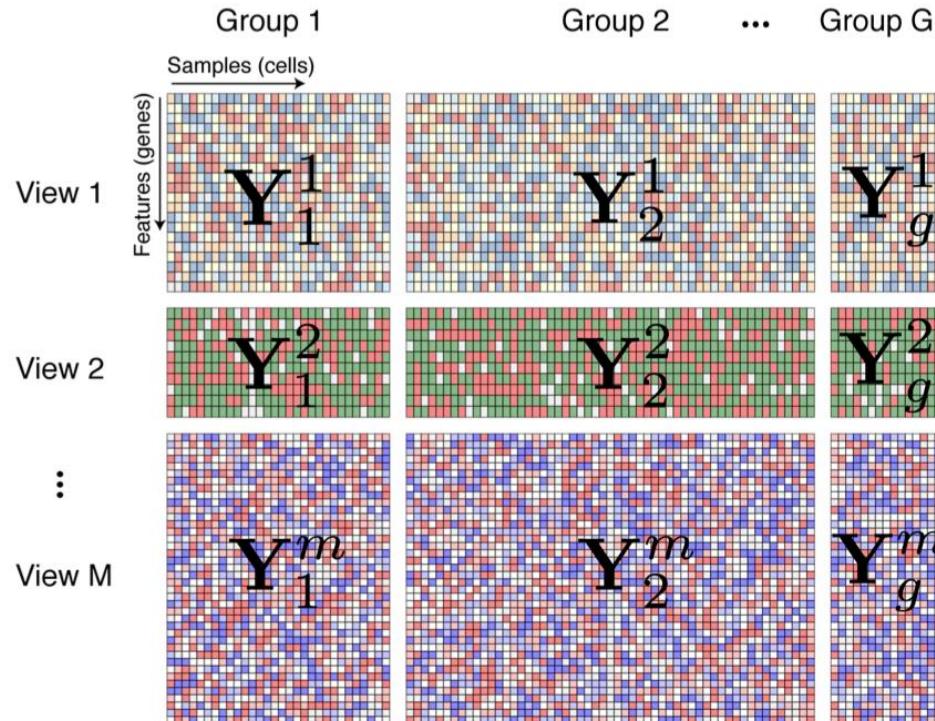
Potential solution: Stratification based on multi-omics profiles



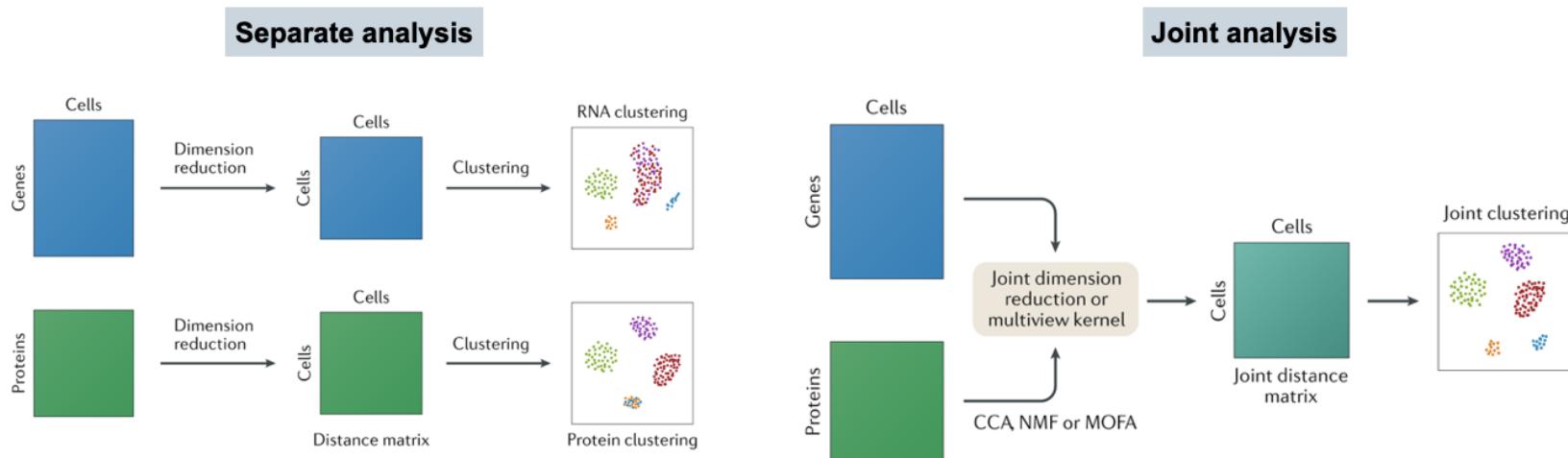
# Example 2: Single cell biology



# Multi-modal data

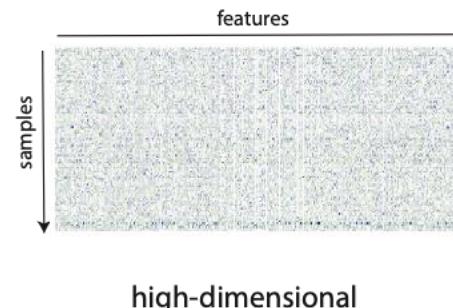
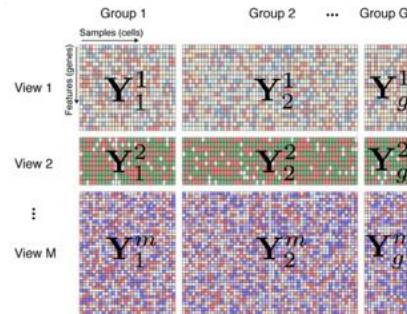
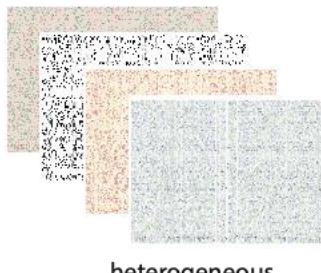


# Why a joint analysis?



# Challenges

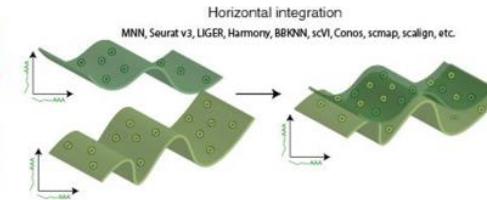
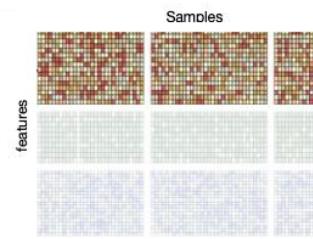
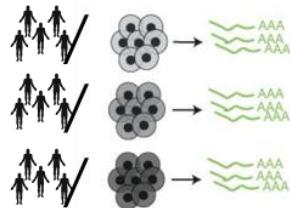
- **heterogeneous data:** distinct statistical properties and inherent structure
- complex **correlation structures** and **hidden confounders**
- **high-dimensional** data requires appropriate **regularization** strategies
- algorithms need to be **scalable** to large data sets
- large amounts (and different patterns) of **missing values**



# How to integrate multi-modal data?

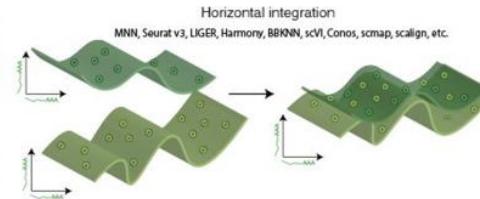
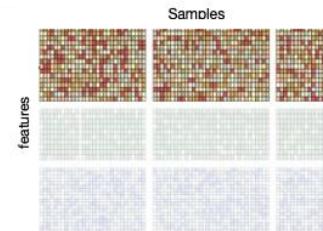
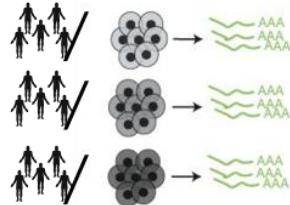
# Step 1: Choose an anchor for integration

Horizontal integration  
(features as anchors)

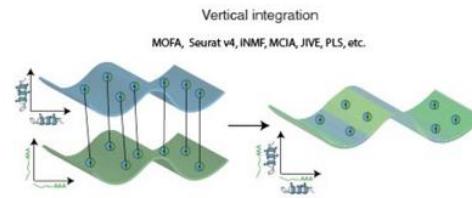
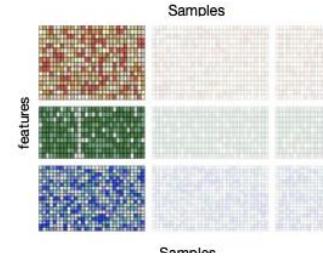
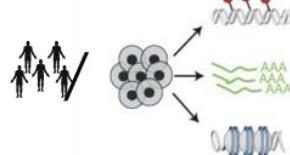


# Step 1: Choose an anchor for integration

Horizontal integration  
(features as anchors)

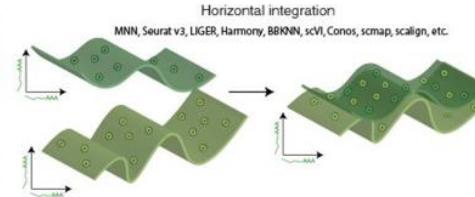
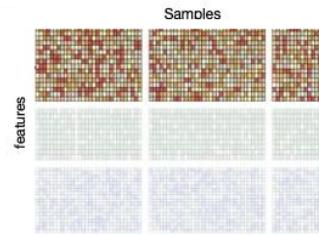
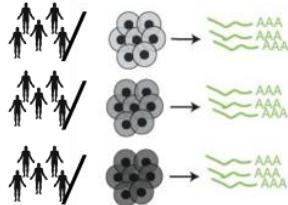


Vertical integration  
(samples as anchors)

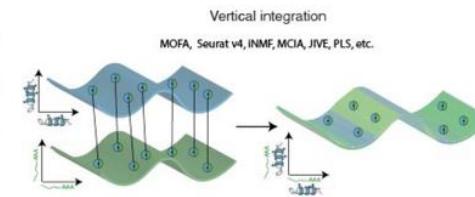
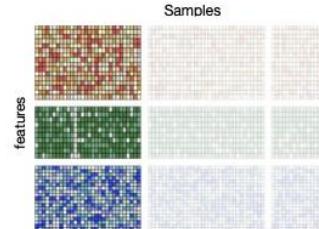
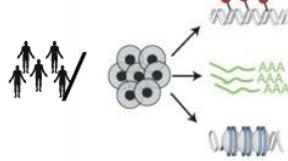


# Step 1: Choose an anchor for integration

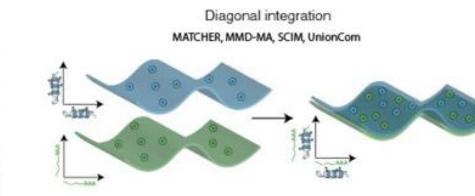
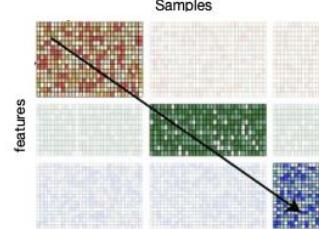
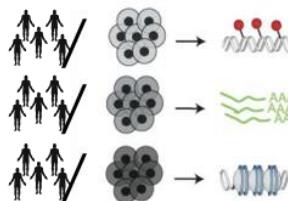
Horizontal integration  
(features as anchors)



Vertical integration  
(samples as anchors)

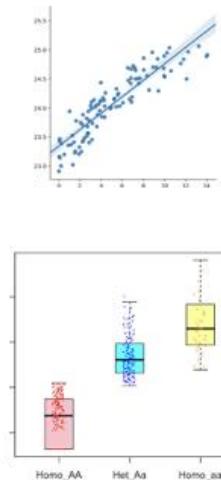


Diagonal integration  
(no anchors)

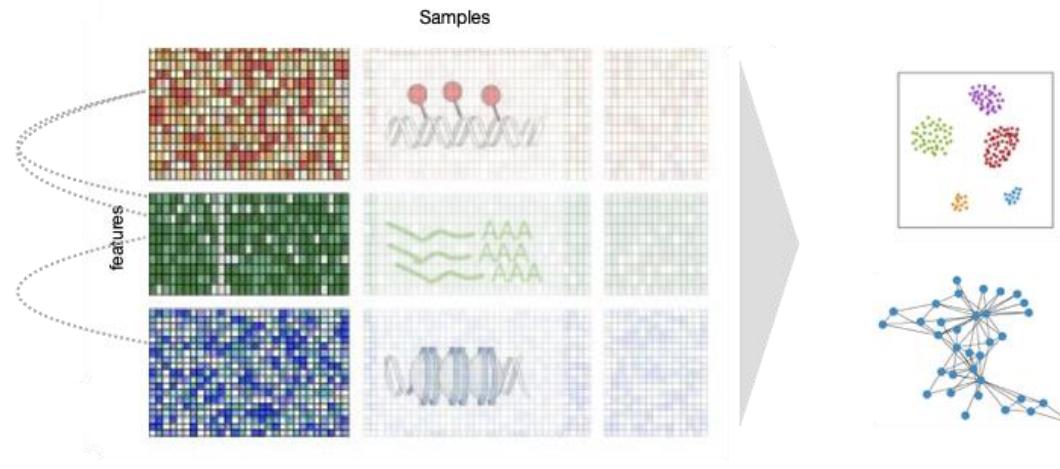


# Step 2: What type of associations?

Local analysis



Global analysis

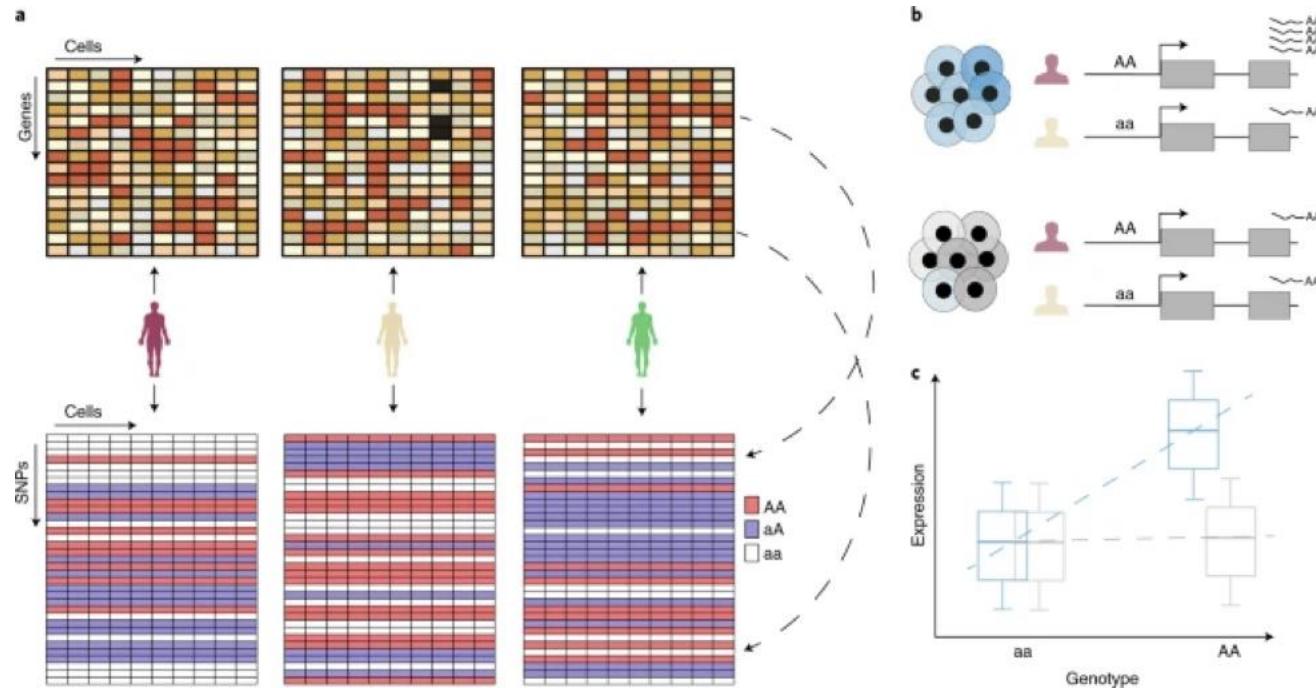


# Local analysis

- **Aim:** Find associations between individual features across molecular layers
  - **eQTL studies:** SNP S changes the expression of gene A
  - **Pharmacogenomics:** drug response depends on expression of genes A, B and C
- **Methods:** supervised models, e.g.
  - regression or classification models
  - association tests

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\alpha} + \boldsymbol{\varepsilon}$$

# Example: single cell eQTL analysis



# Considerations for local analysis

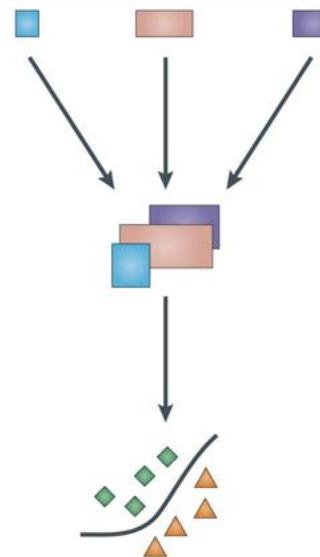
- Linear or non-linear relationship  $Y = f(X)$
- Univariate or multivariate  $X$ , need for regularisation?
- Noise model?
- Confounding factors, error covariance?

# Global analysis

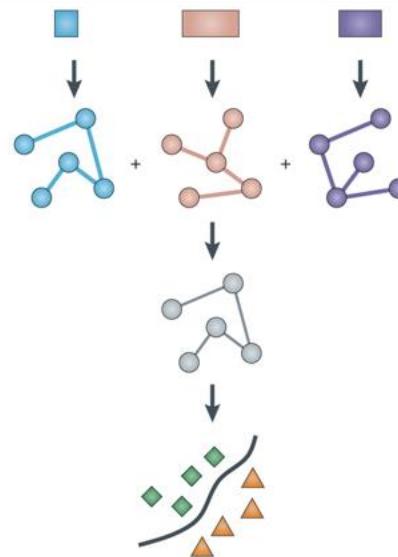
- **Aim:** Find major sources of variation and identify sample structures based on multiple molecular layers, e.g.
  - Single cell clustering
  - Patient stratification
- **Methods:** Mostly unsupervised models, e.g.
  - joint dimension reduction or clustering
  - analysis of covariance structures among features
  - graph-based models

# Global analysis

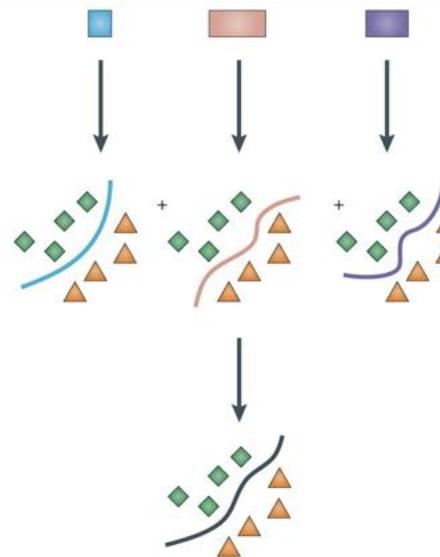
Concatenation-based



Graph-based

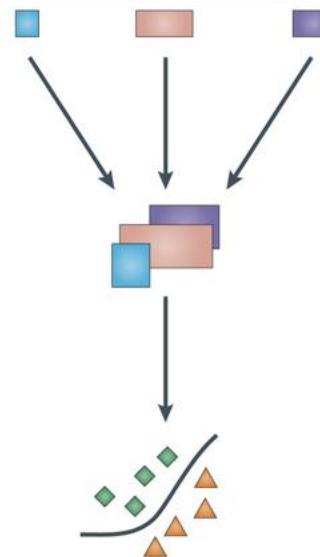


Joint dimension reduction

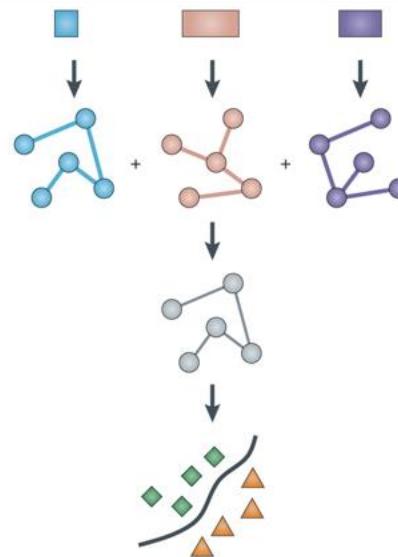


# Global analysis

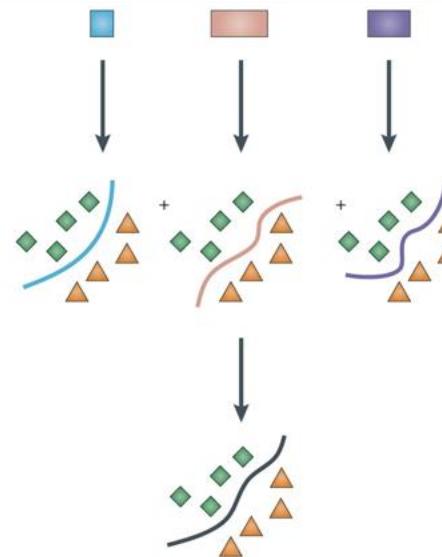
Concatenation-based



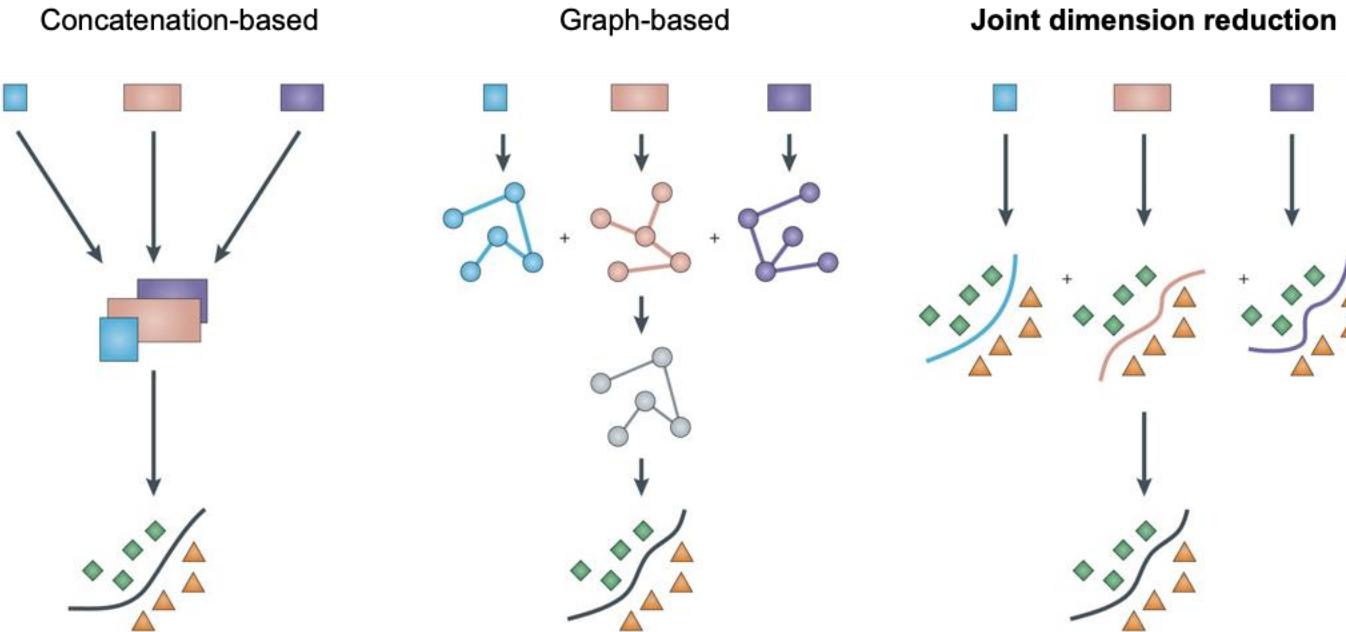
Graph-based



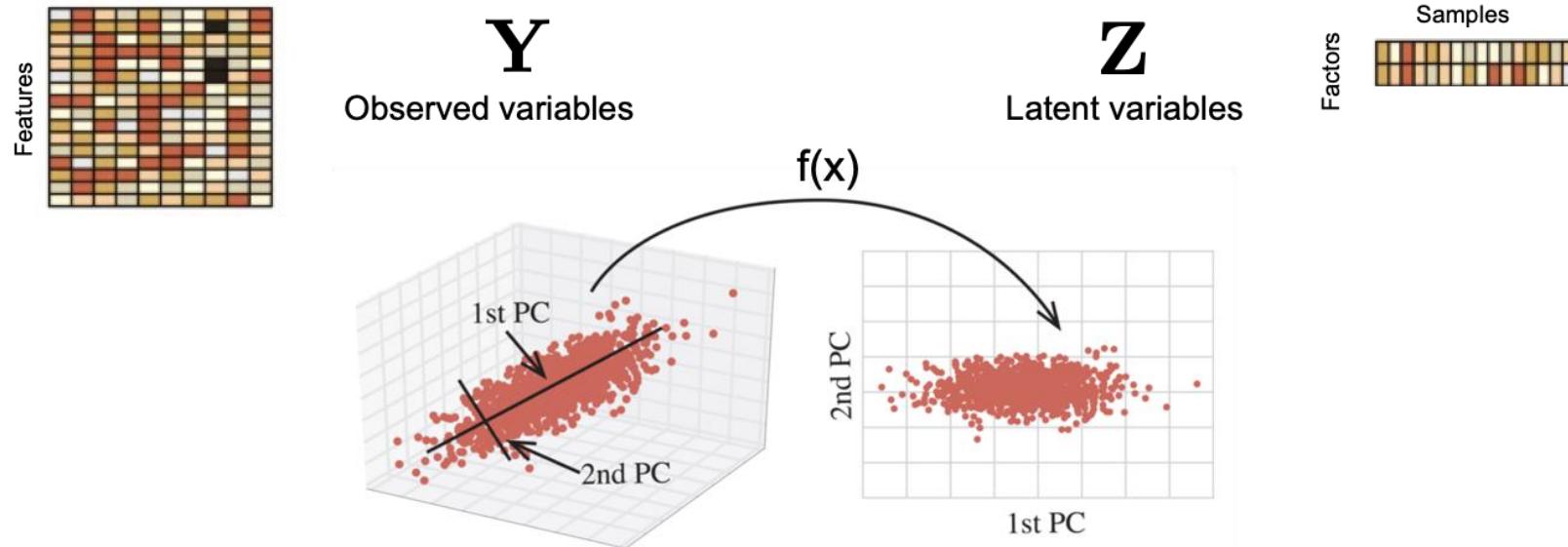
Joint dimension reduction



# Global analysis



# Latent variable models



**Examples:** AEs, VAEs, PCA, factor analysis, Gaussian-process LVMs, principal curves....

# Common Aim

Maximize variance / minimize reconstruction error in the low-dimensional space



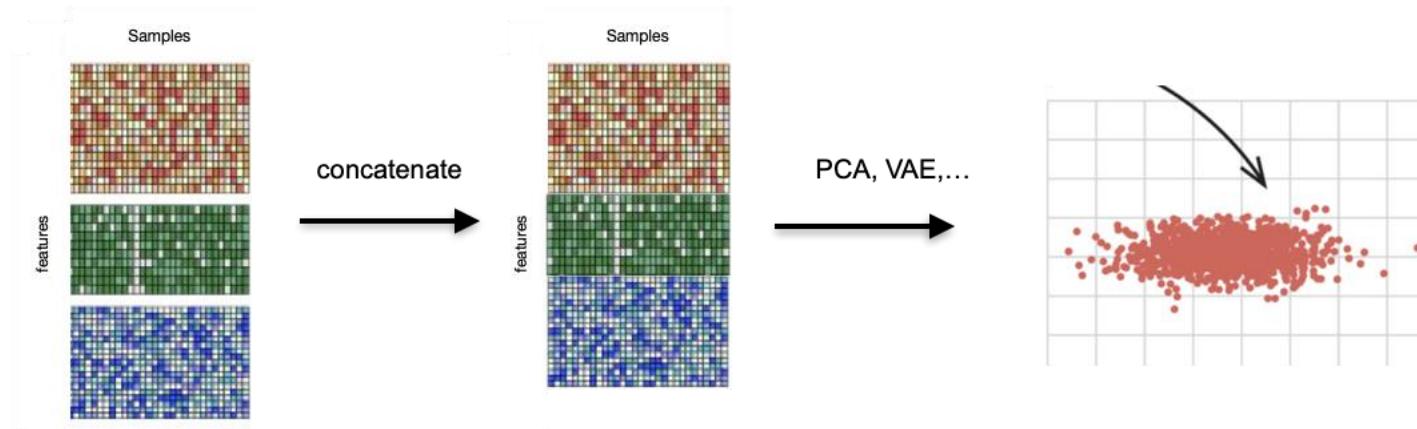
# Common Aim

Maximize variance / minimize reconstruction error in the low-dimensional space



# How to find joint embeddings from multiple data modalities?

# Naïve approach: Concatenation



## Problems:

- does not respect heterogeneity of data modalities (e.g. noise model, quality,...)
- No information on relationships between modalities (only between individual features)
- Data modalities with many features dominate the embedding

# Tailored approaches (Examples)

Canonical correlation analysis (CCA)

Co-inertia analysis (CoIA)

Integrative NMF

Bayesian group factor analysis & MOFA

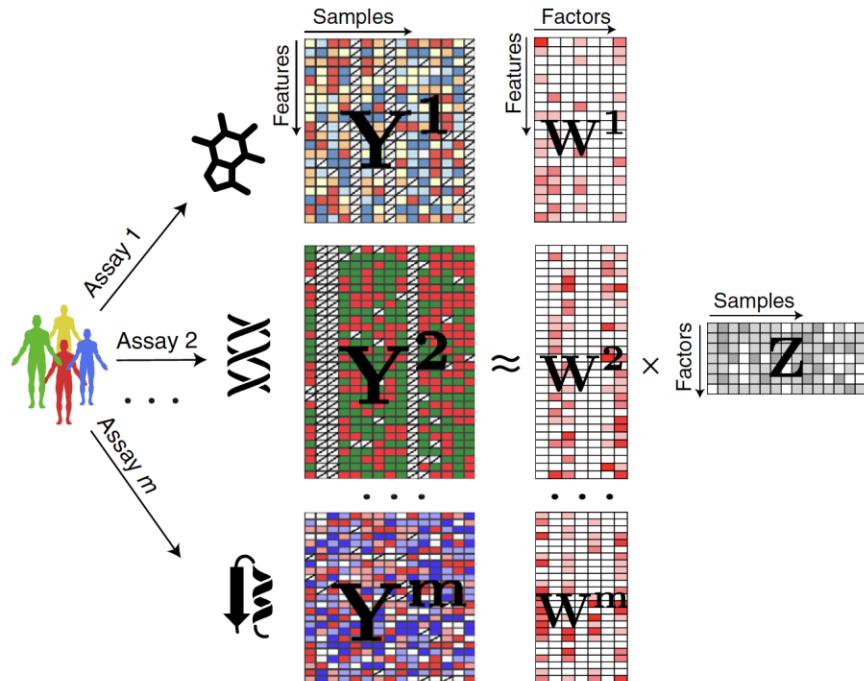
Consensus PCA

Partial least squares (PLS)

Multi-view variational auto-encoders

# MOFA: Probabilistic factor model for multi-modal data

- joint **low-dimensional representation** in terms of (hidden) factors representing the principal axes of variation
- Interpretable model using a **two-level sparsity priors**
  - *Which modalities are important for a factor?*
  - *Which features are important for a factor?*
- **Scalable inference** using (stochastic) variational Bayes



# MOFA - Software

<https://biofam.github.io/MOFA2/index.html>



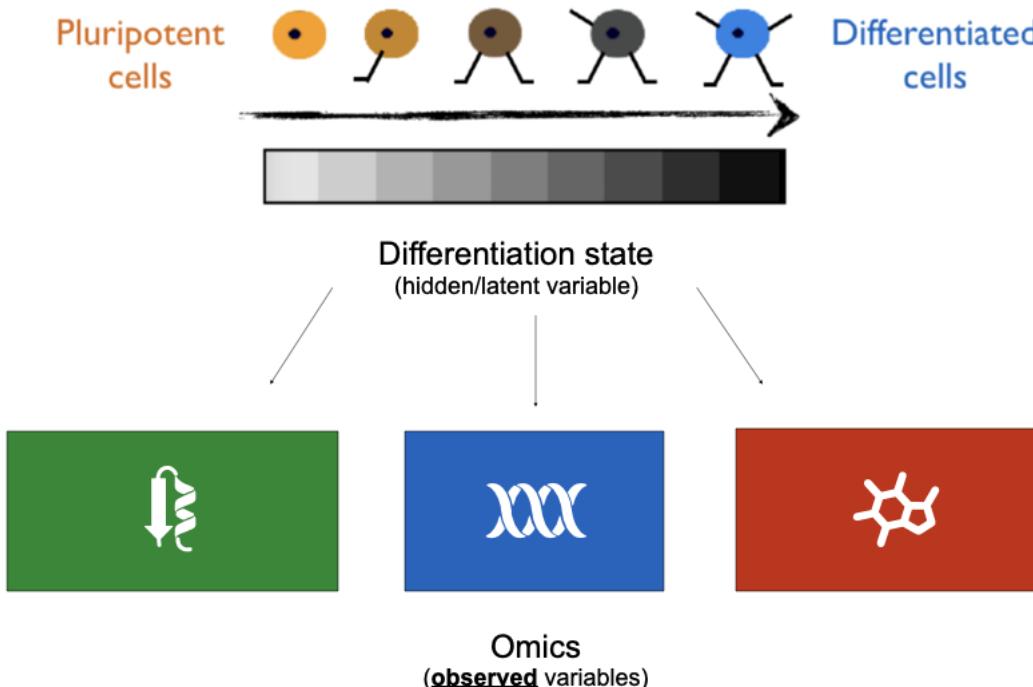
## Overview

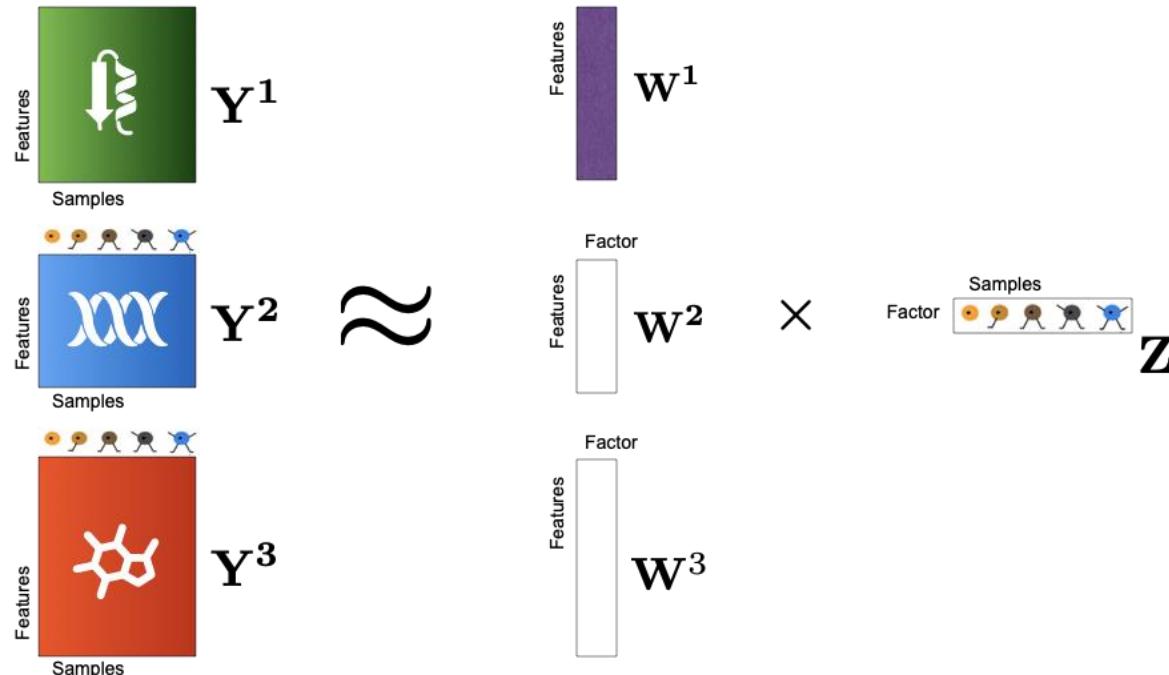
MOFA is a factor analysis model that provides a **general framework for the integration of multi-omic data sets** in an unsupervised fashion. Intuitively, MOFA can be viewed as a versatile and statistically rigorous generalization of principal component analysis to multi-omics data. Given several data matrices with measurements of multiple -omics data types on the same or on

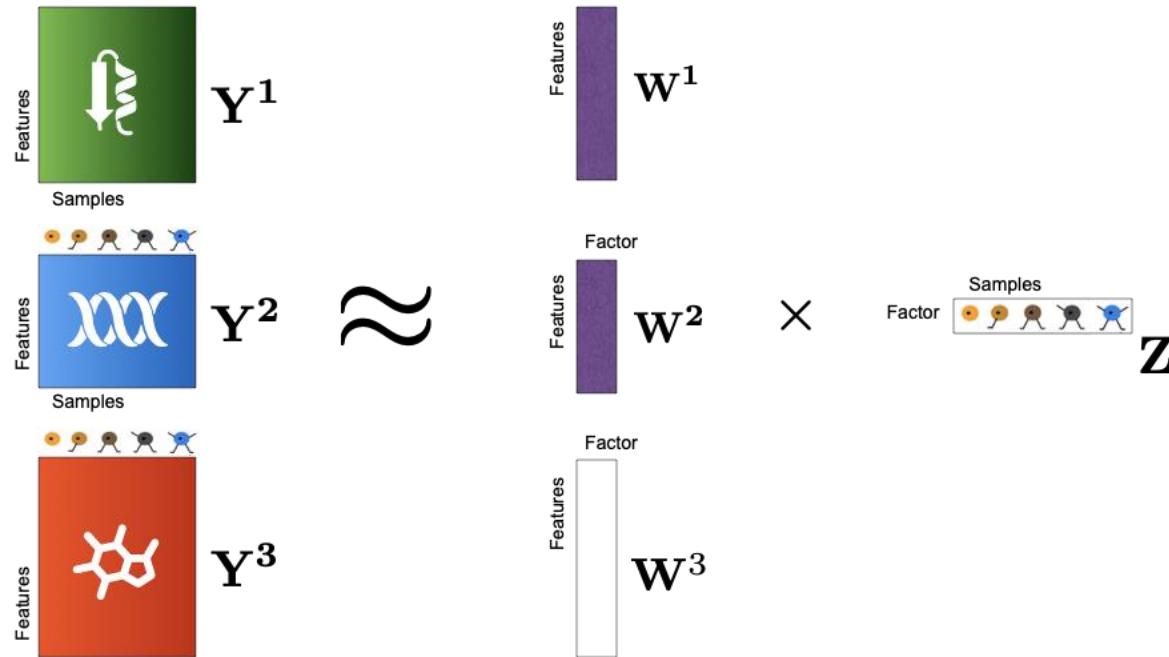
Python: mofapy2, muon  
R: MOFA2

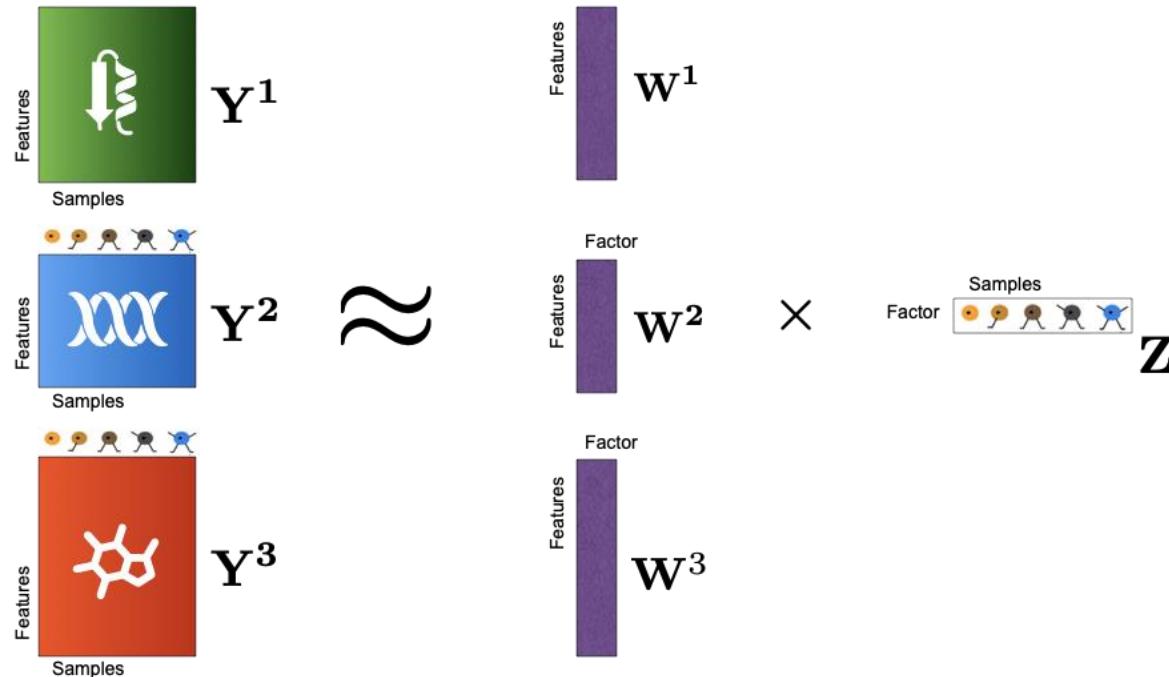
# What is the intuition behind MOFA?

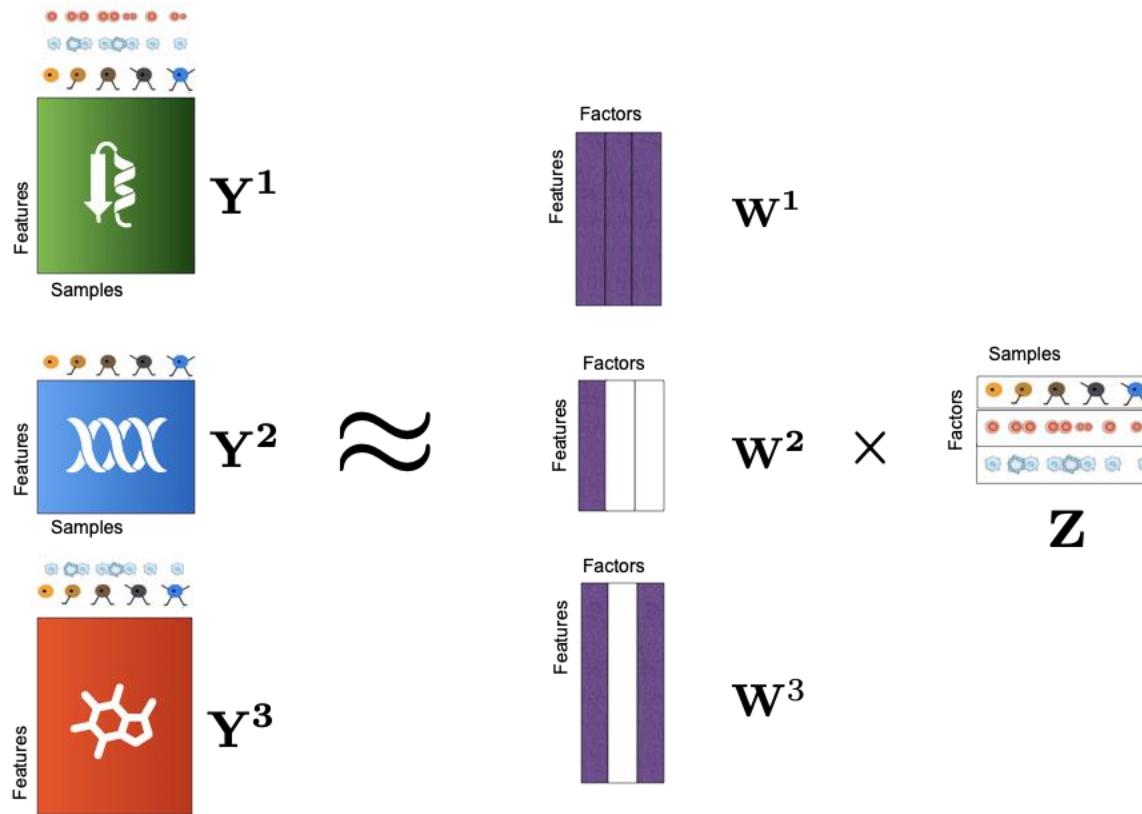
## Differentiation process





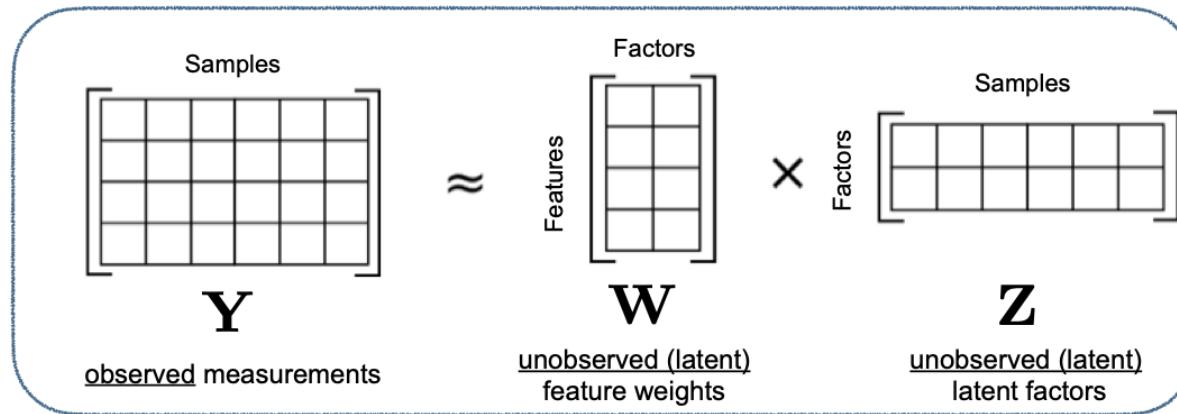






# What is the mathematical model behind MOFA?

# Probabilistic factor model



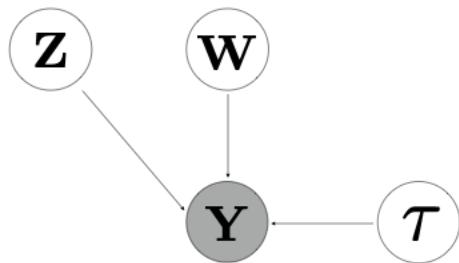
**Probabilistic** (Bayesian) models combine

- a *likelihood function* (statistical model for the observed data)
- a *prior* (distribution of unobserved components)

This can provide **regularisation** & incorporation of **prior information** (prior) and an explicit model of **uncertainties**



# Likelihood model



Observed variable



Unobserved variable

$$\mathbf{Y} = \mathbf{Z}\mathbf{W}^T + \mathcal{N}(\mathbf{0}, \tau^{-1})$$

Dimensions

**Y** Observation matrix  $(N,D)$

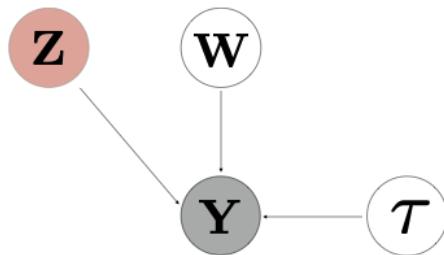
**Z** Latent variable matrix  $(N,K)$

**W** Weight matrix  $(D,K)$

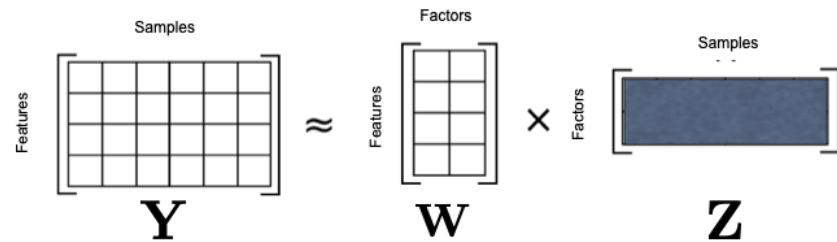
**T** Noise vector  $(D,)$

*N* is the number of samples  
*D* is the number of features  
*K* is the number of factors

# Prior for factors

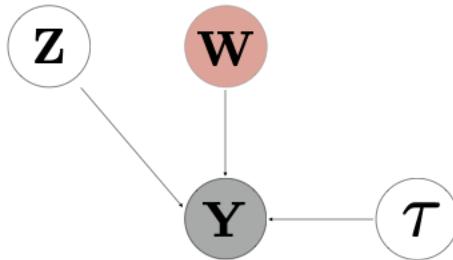


$$p(\mathbf{Z}) = \prod_{n=1}^N \prod_{k=1}^K \mathcal{N}(0, 1)$$



*N* is the number of samples  
*D* is the number of features  
*K* is the number of factors

# Prior for weights



*N* is the number of samples  
*D* is the number of features  
*K* is the number of factors

Automatic Relevance Determination (ARD)

$$p(\mathbf{W}|\alpha) = \prod_{k=1}^K \mathcal{N}(\mathbf{w}_{:,k}|0, \alpha_k^{-1} \mathbf{I})$$

Features Samples  $\approx$  Factors  $\times$  Samples

Y	W	Z
---	---	---

# Automatic relevance determination prior

$$p(w|\alpha) = \mathcal{N}(w|0, \frac{1}{\alpha})$$

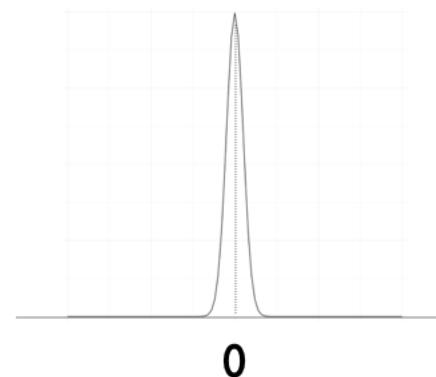
If precision  $\alpha$  is small:

*Broad distribution*

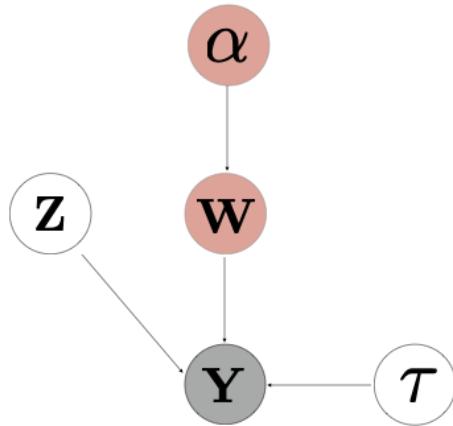


If precision  $\alpha$  is large:

*Distribution peaked at 0*



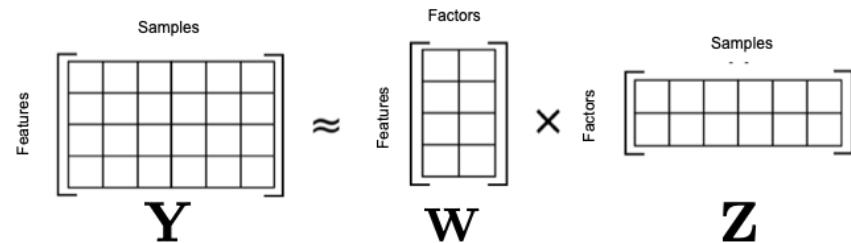
# Automatic relevance determination prior



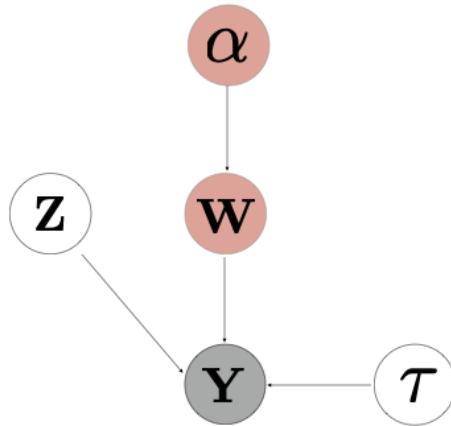
*N* is the number of samples  
*D* is the number of features  
*K* is the number of factors

$$p(\mathbf{W}|\alpha) = \prod_{k=1}^K \mathcal{N}(\mathbf{w}_{:,k}|0, \alpha_k^{-1} \mathbf{I})$$

$$p(\alpha_k) = \mathcal{G}(a_0^\alpha, b_0^\alpha) \quad \text{Uninformative gamma prior}$$



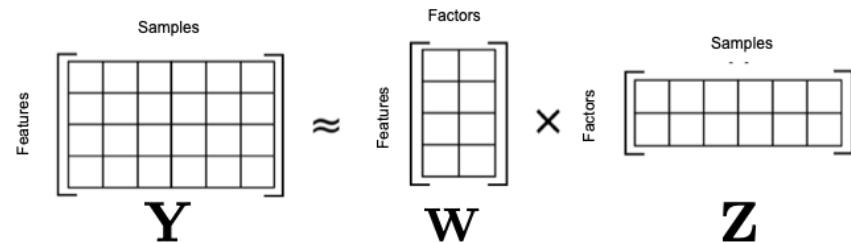
# Automatic relevance determination prior



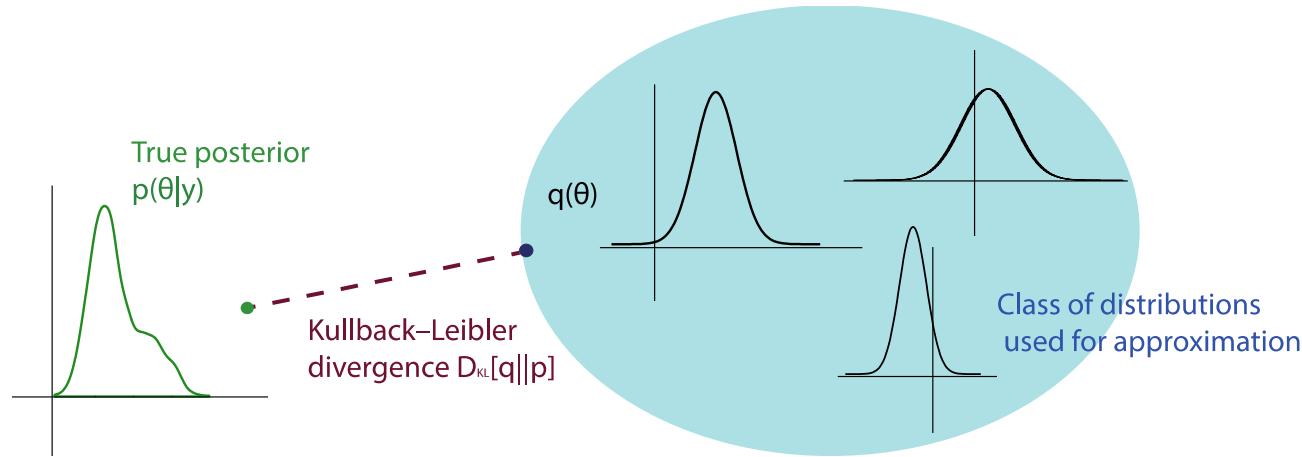
*N* is the number of samples  
*D* is the number of features  
*K* is the number of factors

$$p(\mathbf{W}|\alpha) = \prod_{k=1}^K \mathcal{N}(\mathbf{w}_{:,k}|0, \alpha_k^{-1} \mathbf{I})$$

$$p(\alpha_k) = \mathcal{G}(a_0^\alpha, b_0^\alpha) \quad \text{Uninformative gamma prior}$$

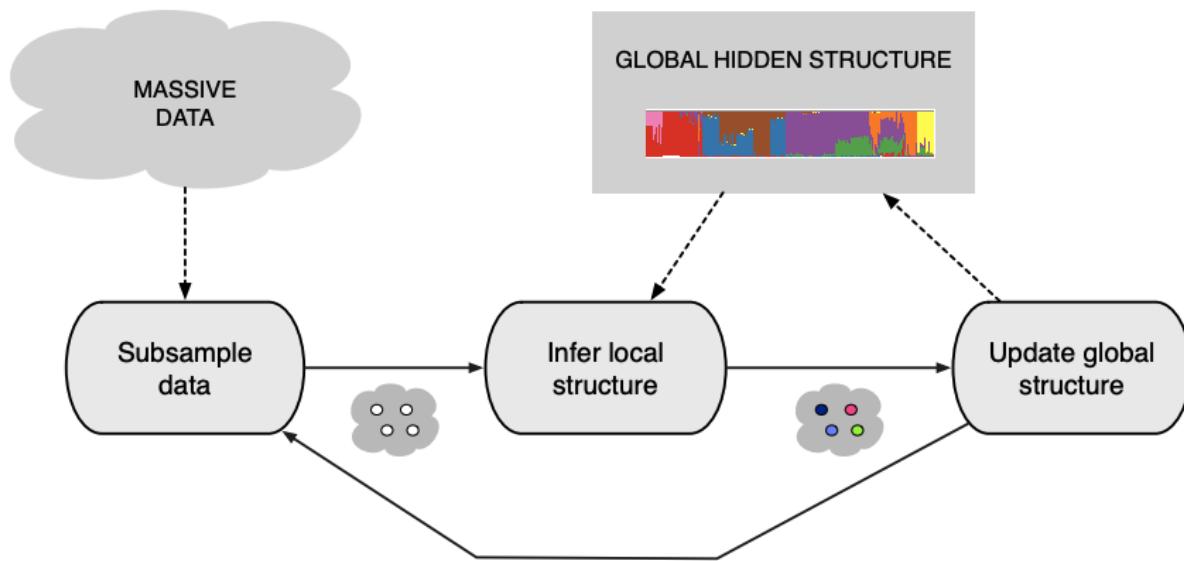


# Finding the posterior $p(Z, W, \alpha|Y)$

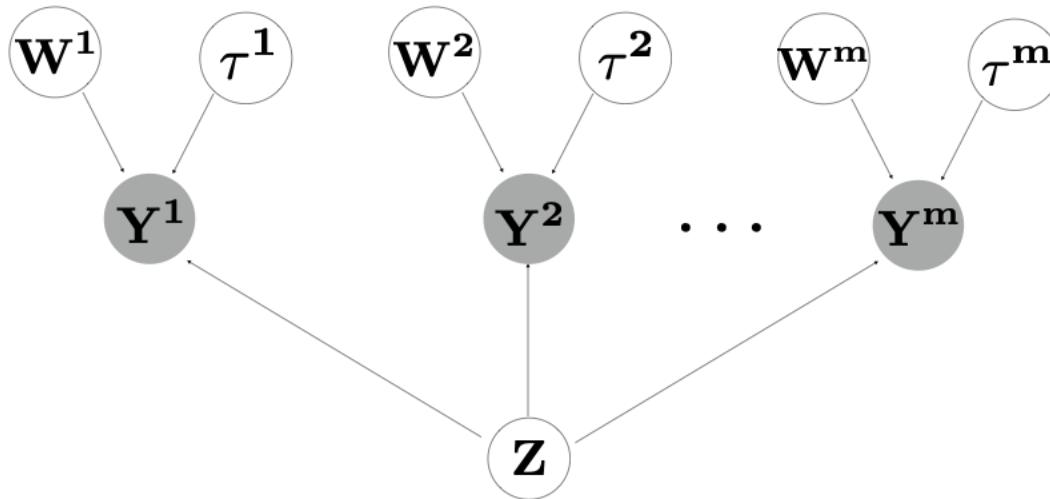


**Variational inference** provides an approximation of the posterior from a restricted class of distributions.

# Stochastic variational inference



# Multi-view probabilistic factor analysis



$$\mathbf{Y}^m = \mathbf{Z}\mathbf{W}^m + \mathcal{N}(0, \tau_m)$$

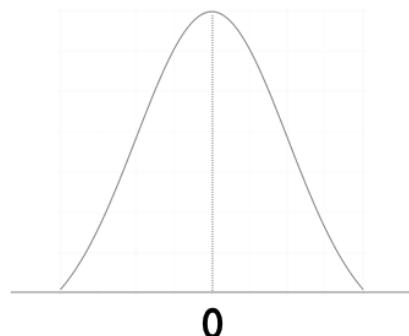
M is the number of “views” (omics)

# Determining activity of factors per view

$$p(w_k^{(m)} | \alpha_k^{(m)}) = N\left(w_k^{(m)} | 0, \frac{1}{\alpha_k^{(m)}}\right)$$

If precision  $\alpha_k^{(m)}$  is small:

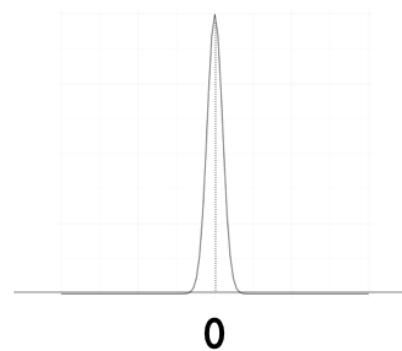
*Broad distribution*



*Factor k is active in view m*

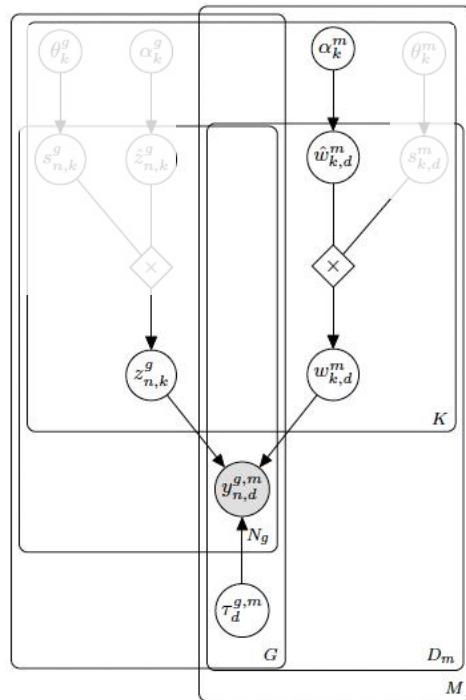
If precision  $\alpha_k^{(m)}$  is large:

*Distribution peaked at 0*



*Factor k is not active in view m*

# Plate notation of graphical model



# pyro



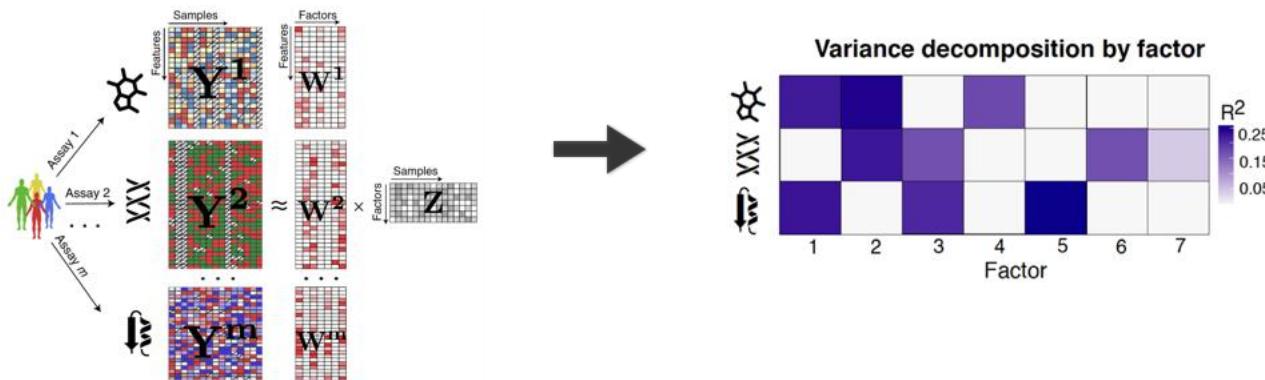
```
from pyro.nn import PyroSample

class BayesianRegression(PyroModule):
    def __init__(self, in_features, out_features):
        super().__init__()
        self.linear = PyroModule[nn.Linear](in_features, out_features)
        self.linear.weight = PyroSample(dist.Normal(0., 1.).expand([out_features, in_features]))
        self.linear.bias = PyroSample(dist.Normal(0., 10.).expand([out_features])).to_event()

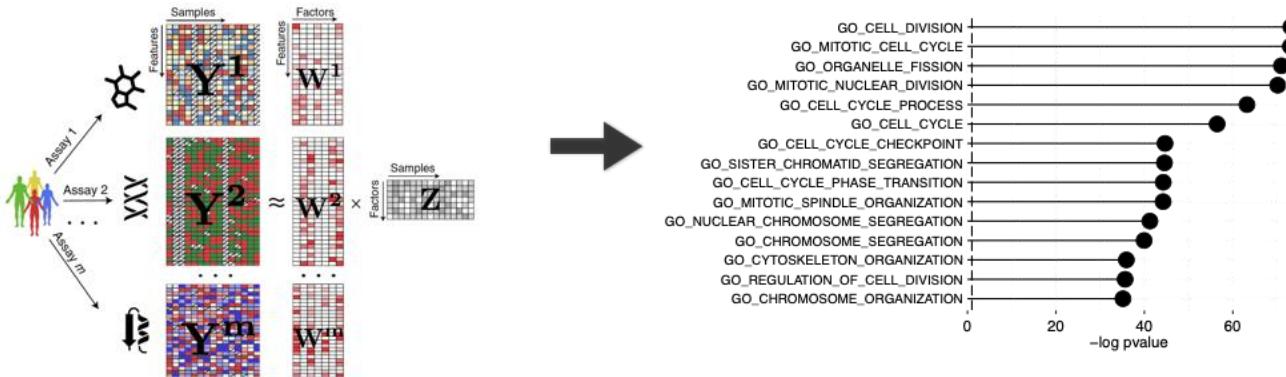
    def forward(self, x, y=None):
        sigma = pyro.sample("sigma", dist.Uniform(0., 10.))
        mean = self.linear(x).squeeze(-1)
        with pyro.plate("data", x.shape[0]):
            obs = pyro.sample("obs", dist.Normal(mean, sigma), obs=y)
        return mean
```

# What can be done with such models?

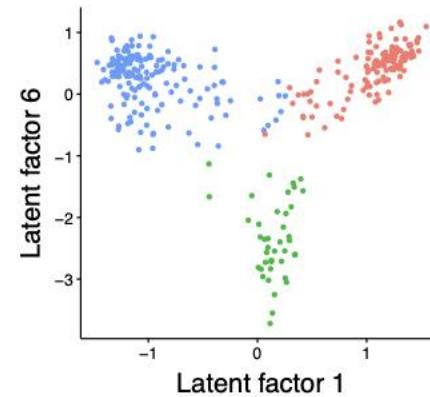
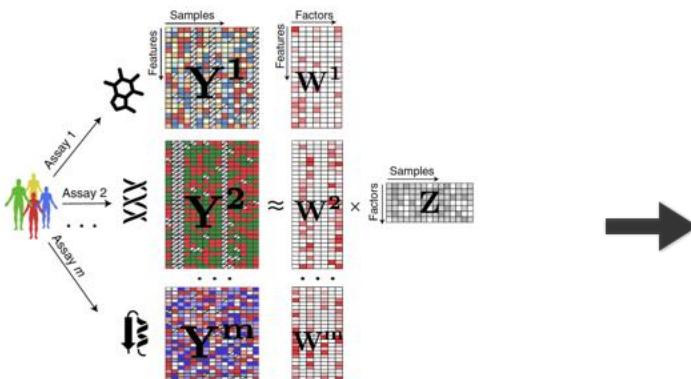
# Variance decomposition



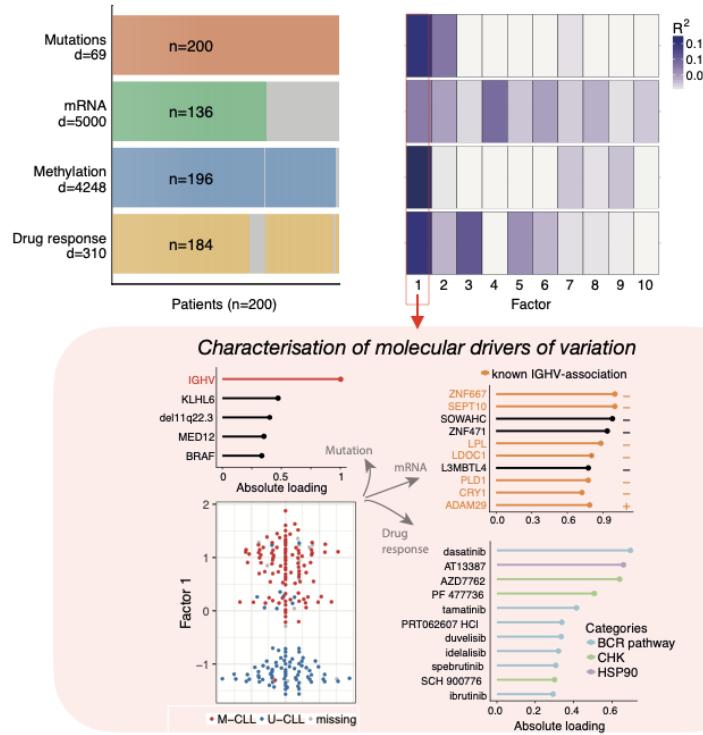
# Inspection of weights



# Visualisation of samples



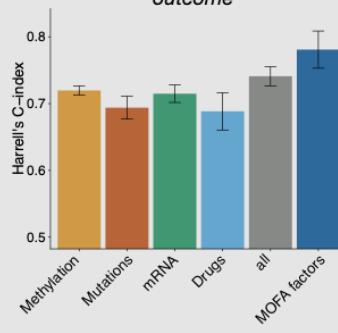
# Example: Precision medicine



200 chronic lymphocytic leukaemia samples (incompletely) characterised by

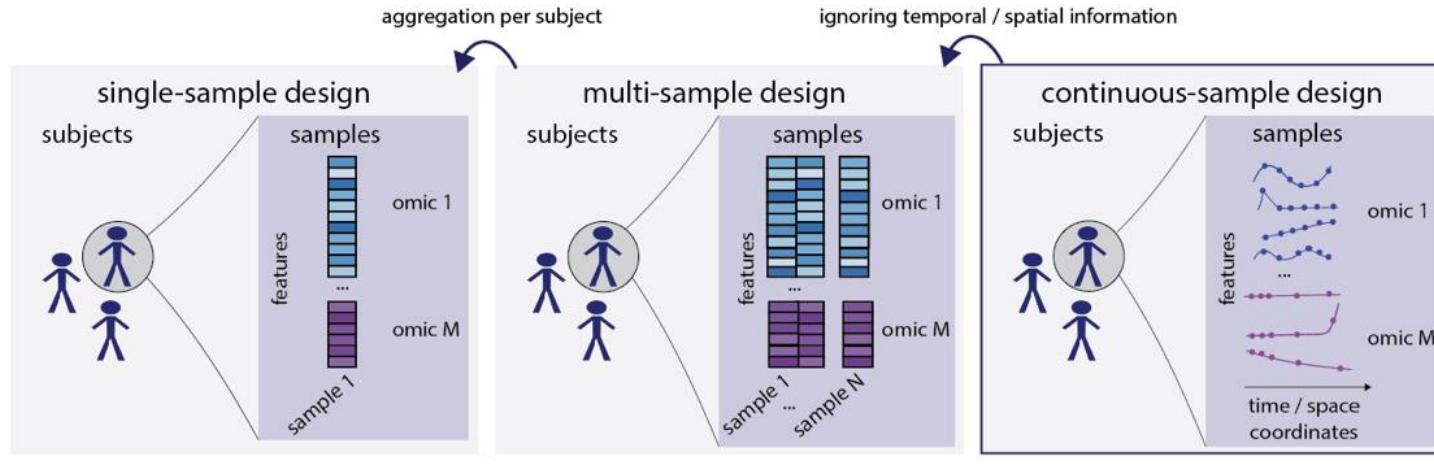
- genomic sequencing
- RNA-seq
- methylation arrays
- ex-vivo drug response assays

*Improved prediction of clinical outcome*



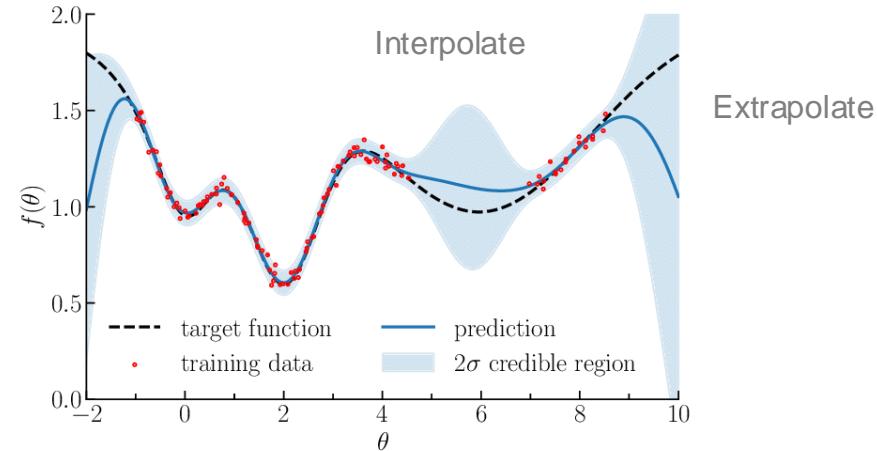
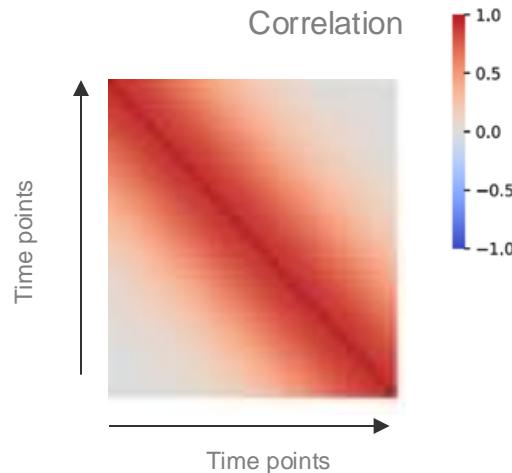
## Multi-modal data with time and space resolution

# Temporal or spatial resolution

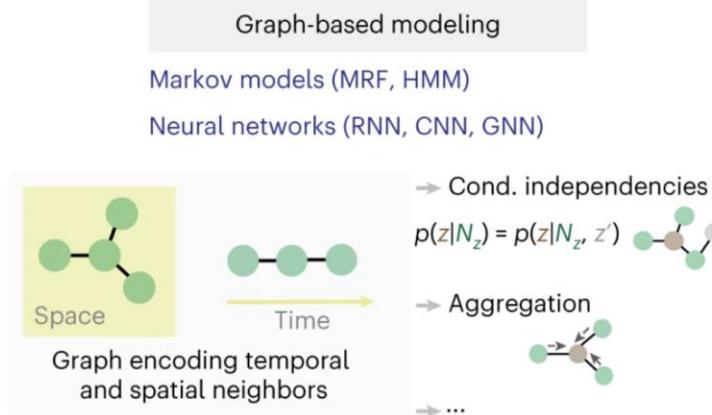
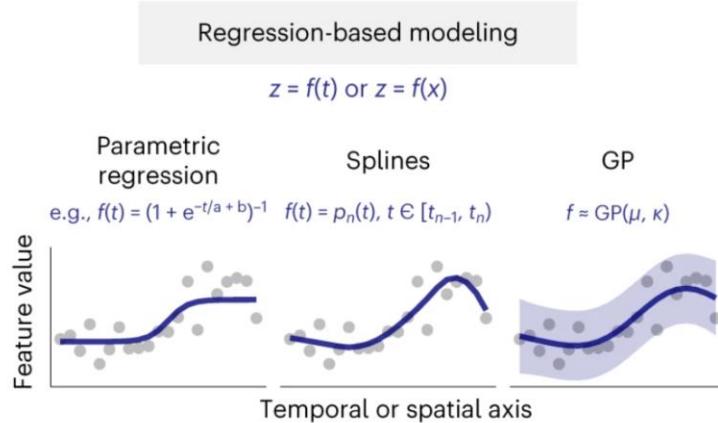


# What changes?

- Samples are non-i.i.d: nearby time points or spatial positions can be correlated
- Assumptions of smoothness along time or space can help to
  - mitigate noise
  - interpolate or extrapolate to unmeasured time points or positions



# Avenues for accounting for time & space



# Gaussian processes

## Definition

A Gaussian process is a statistical distribution  $f_t$ ,  $t \in T$ , for which any finite linear combination of samples has a joint Gaussian distribution.

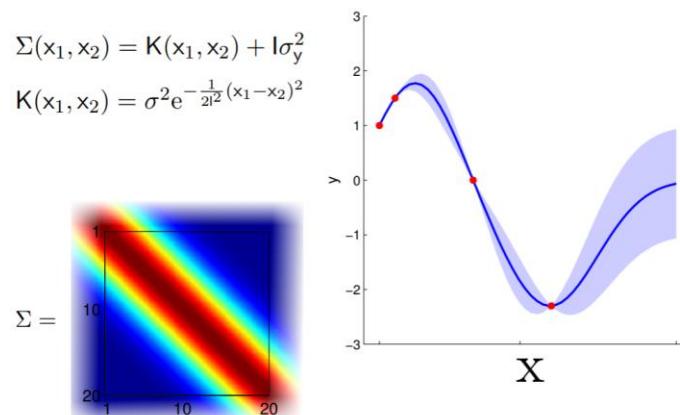
A Gaussian process with mean function  $\mu$  and covariance function  $k$  is denoted by

$$f \sim GP(\mu, k)$$

$$\Sigma(x_1, x_2) = K(x_1, x_2) + I\sigma_y^2$$

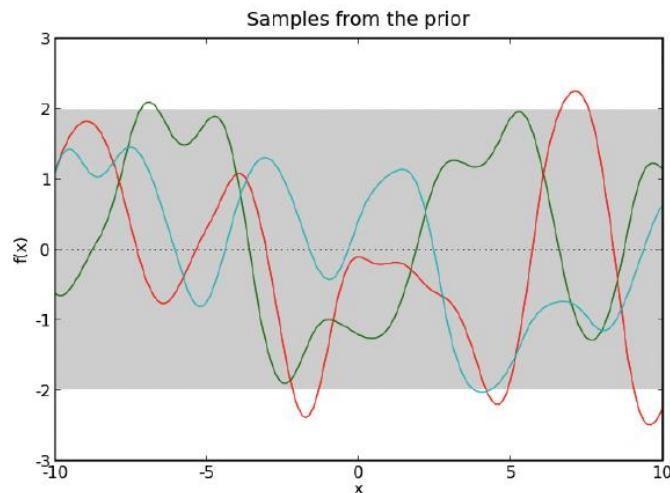
$$K(x_1, x_2) = \sigma^2 e^{-\frac{1}{2l^2}(x_1 - x_2)^2}$$

→ Probabilistic non-parametric method for learning functional relationships

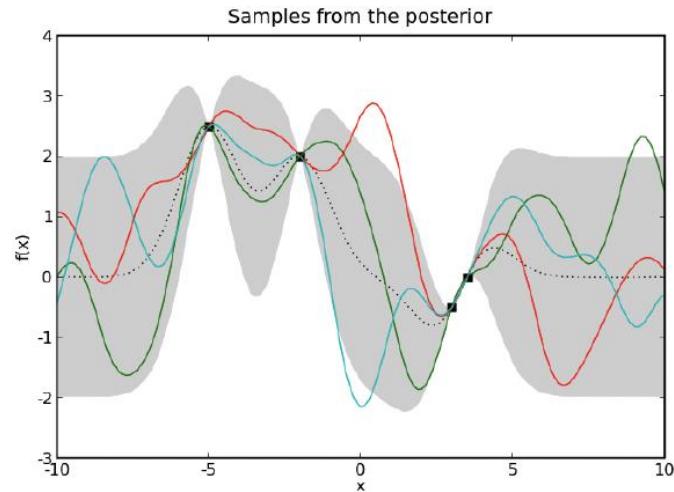


# Gaussian processes

Before observing any data:

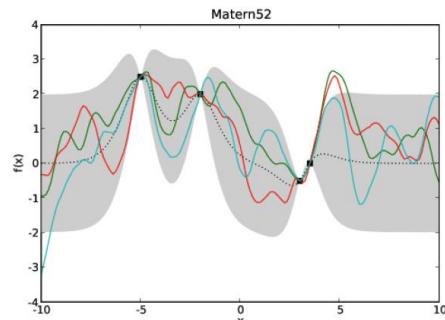
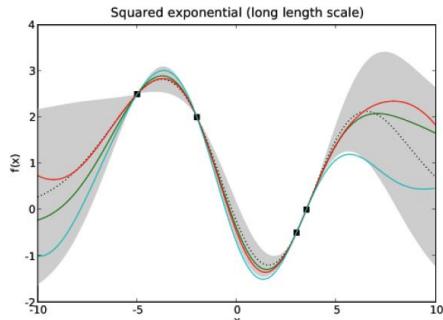
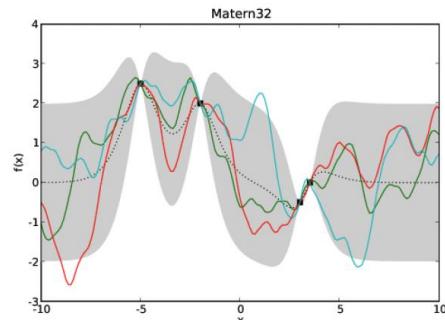
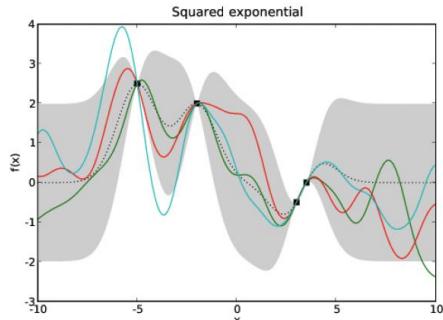


After measurements:



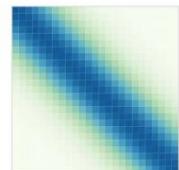
# Gaussian processes

Choice of kernel / covariance function determines smoothness



RBF KERNEL

$$\sigma^2 \exp\left(-\frac{\|t-t'\|^2}{2l^2}\right)$$



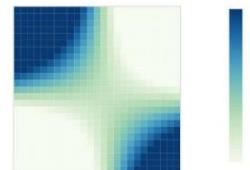
PERIODIC

$$\sigma^2 \exp\left(-\frac{2 \sin^2(\pi|t-t'|/p)}{l^2}\right)$$

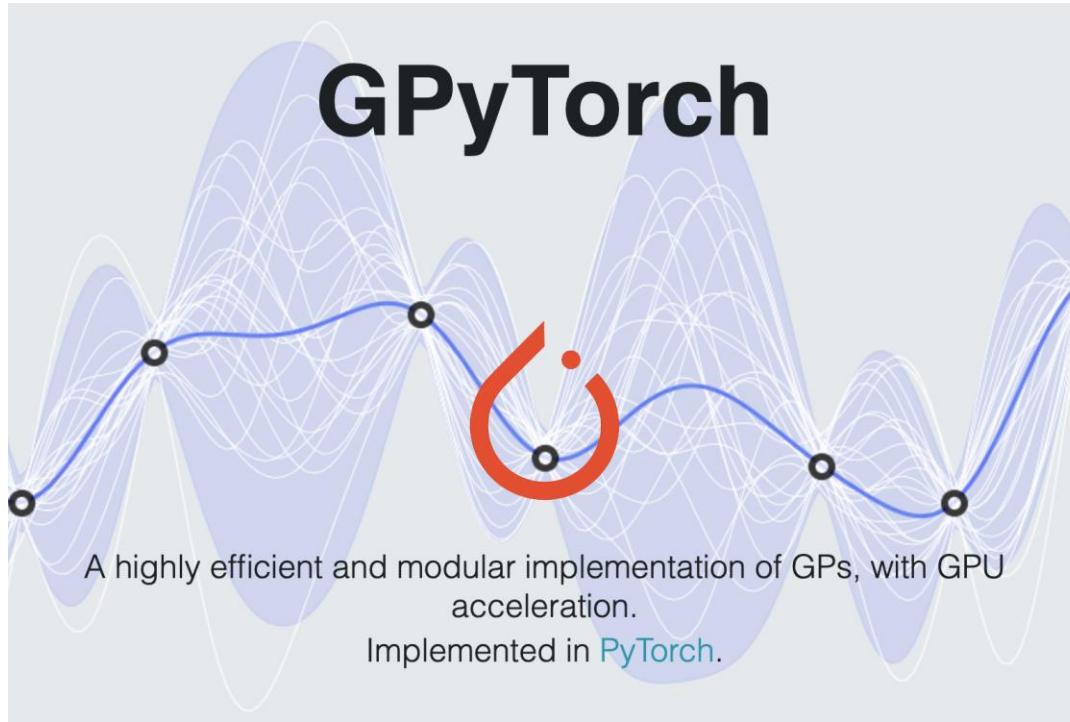


LINEAR

$$\sigma_b^2 + \sigma^2(t-c)(t'-c)$$



# gpytorch



# Example: Regression in gpytorch

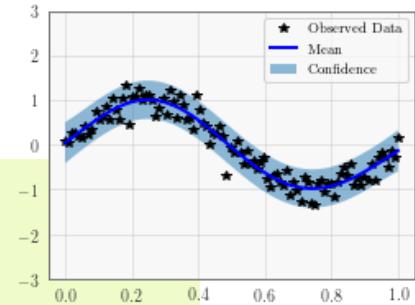
```

class ExactGPModel(gpytorch.models.ExactGP):
    def __init__(self, train_x, train_y, likelihood):
        super(ExactGPModel, self).__init__(train_x, train_y, likelihood)
        self.mean_module = gpytorch.means.ConstantMean()
        self.covar_module = gpytorch.kernels.ScaleKernel(gpytorch.kernels.RBFKernel())

    def forward(self, x):
        mean_x = self.mean_module(x)
        covar_x = self.covar_module(x)
        return gpytorch.distributions.MultivariateNormal(mean_x, covar_x)

# initialize likelihood and model
likelihood = gpytorch.likelihoods.GaussianLikelihood()
model = ExactGPModel(train_x, train_y, likelihood)

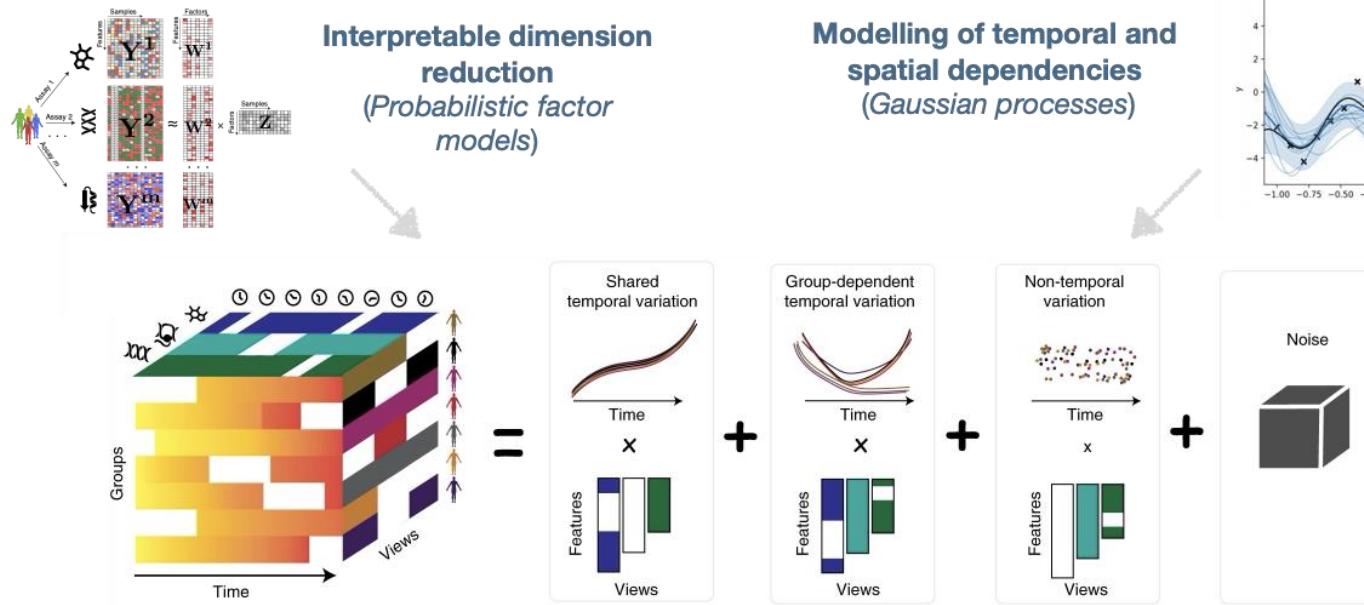
```



→ pytorch training loop

# How to use Gaussian processes for dimension reduction and data integration?

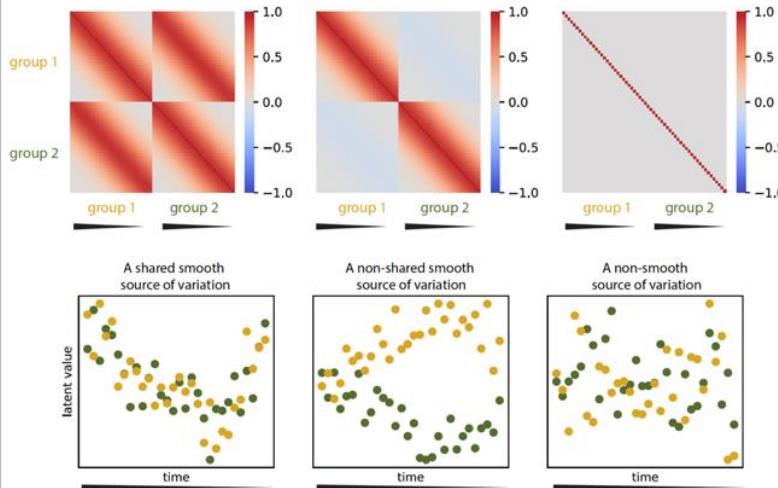
# GPFA for multi-modal data: MEFISTO



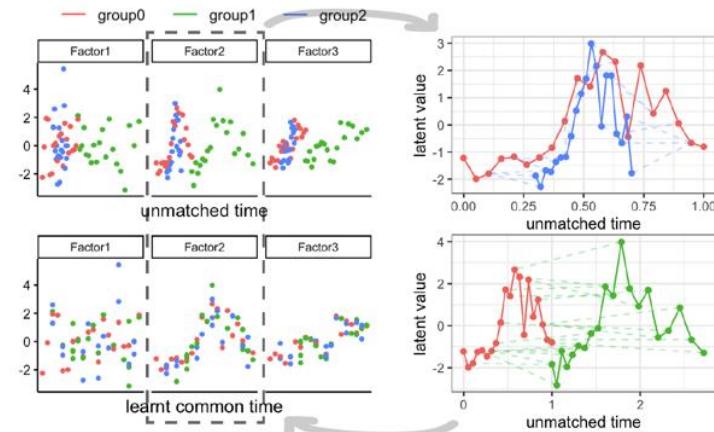
**Integrated alignment** of time points across individuals enables a meaningful comparisons of temporal patterns.

# GPFA for multi-modal data: MEFISTO

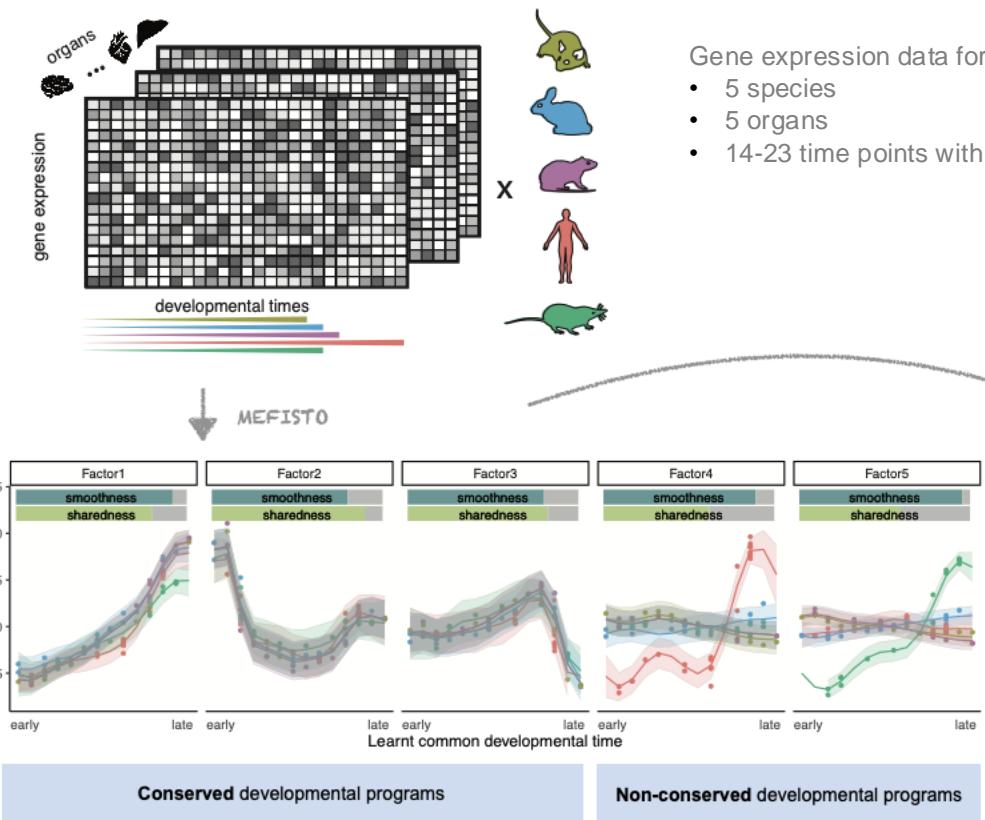
MEFISTO models **dependencies between time points or locations and sample groups** (e.g. individuals)



MEFISTO **aligns** temporal patterns across groups with unclear time correspondences

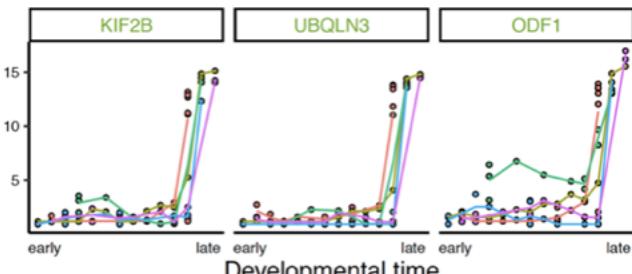


# Example: Evo-devo data

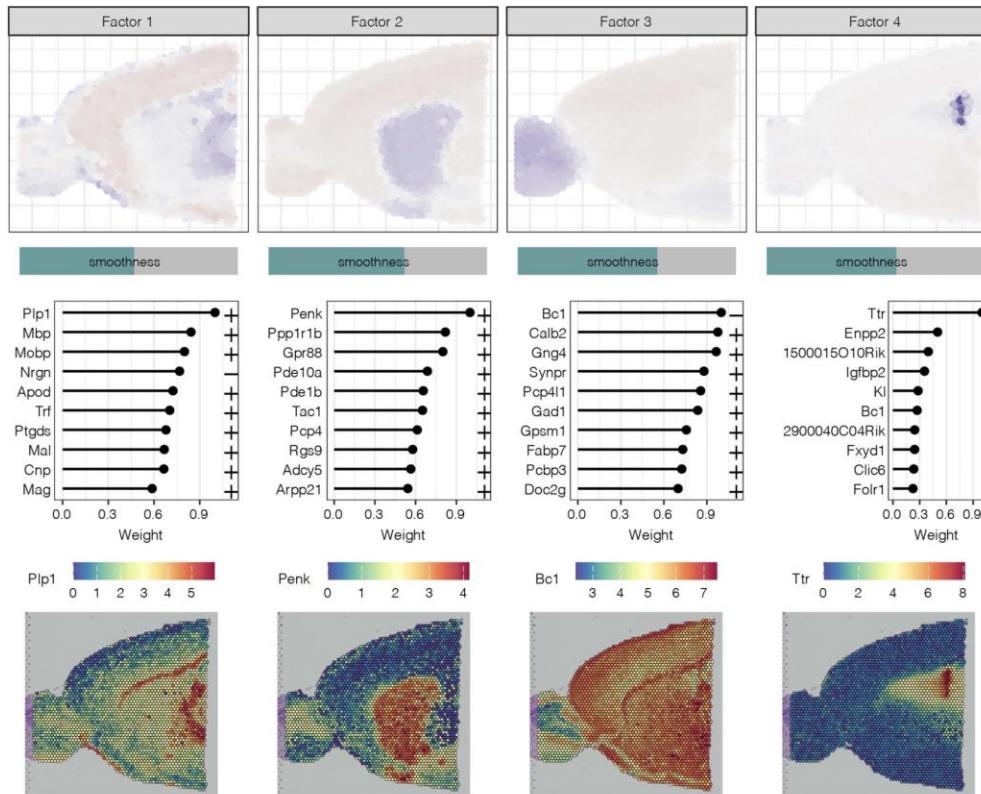


Analysis of top weights  
in MEFISTO

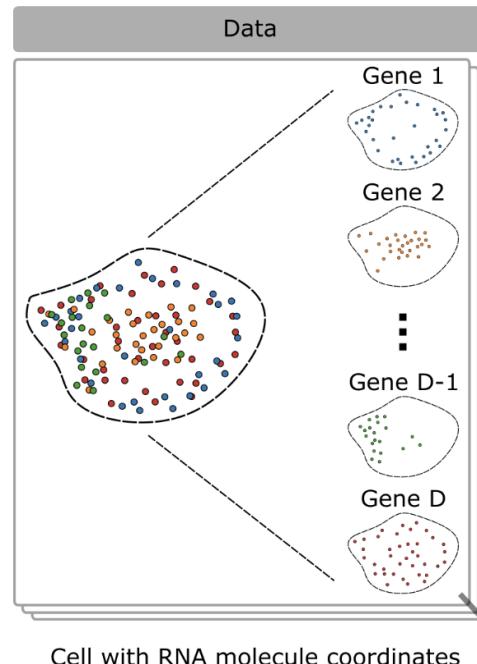
Testis



# Example: Spatial data



# Different likelihood models

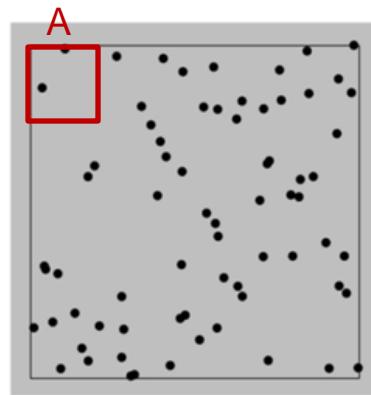


# Poisson point processes

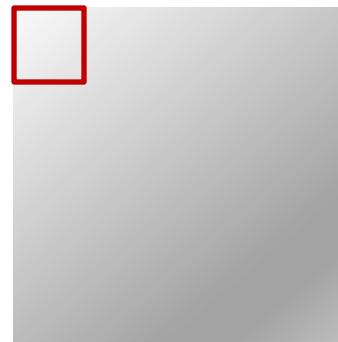
Spatial binning &  
Poisson likelihood



Poisson point process

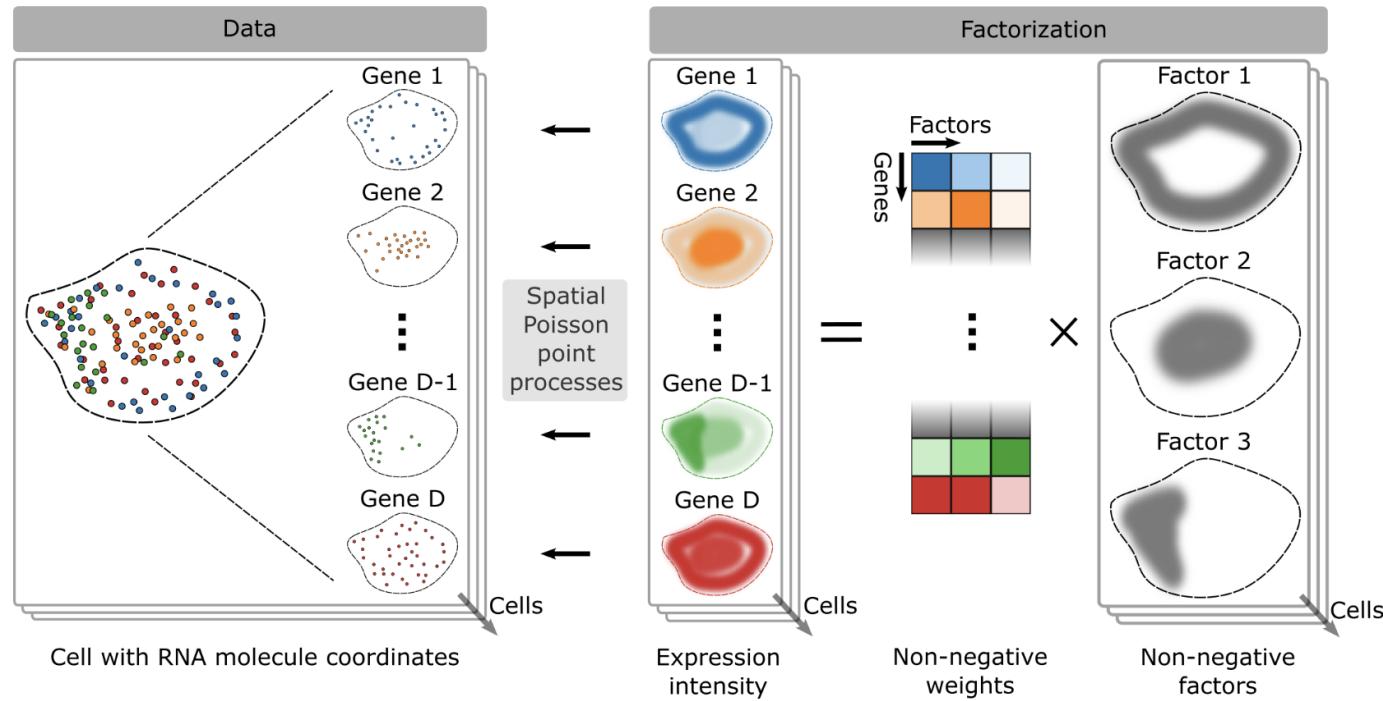


Intensity function  $\lambda$

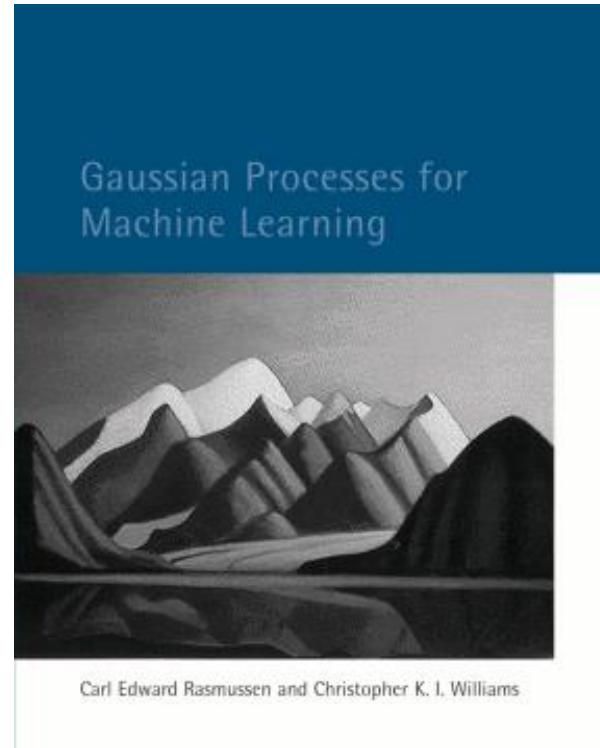
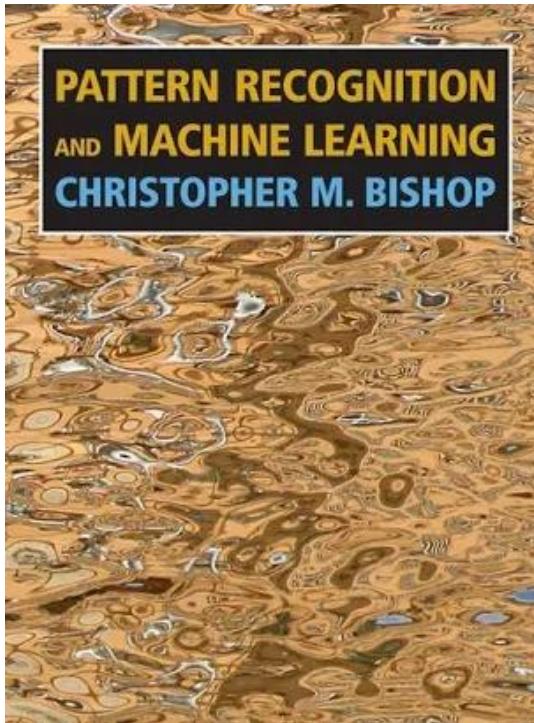


$$P\{N(A) = n\} = \frac{\lambda(|A|)^n}{n!} e^{-\lambda|A|}$$

# PP-GPFA for FISH data: FISHFactor



# Further reading



Carl Edward Rasmussen and Christopher K. I. Williams

# Questions?