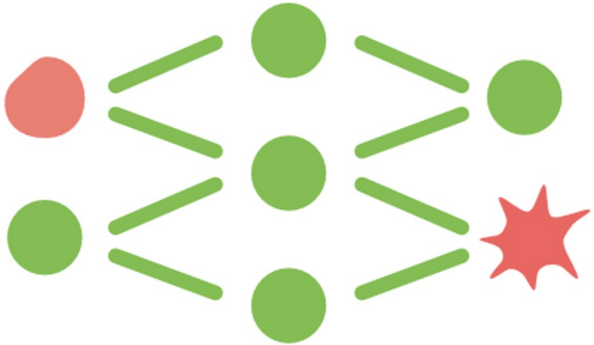
Deep Learning in Life Sciences



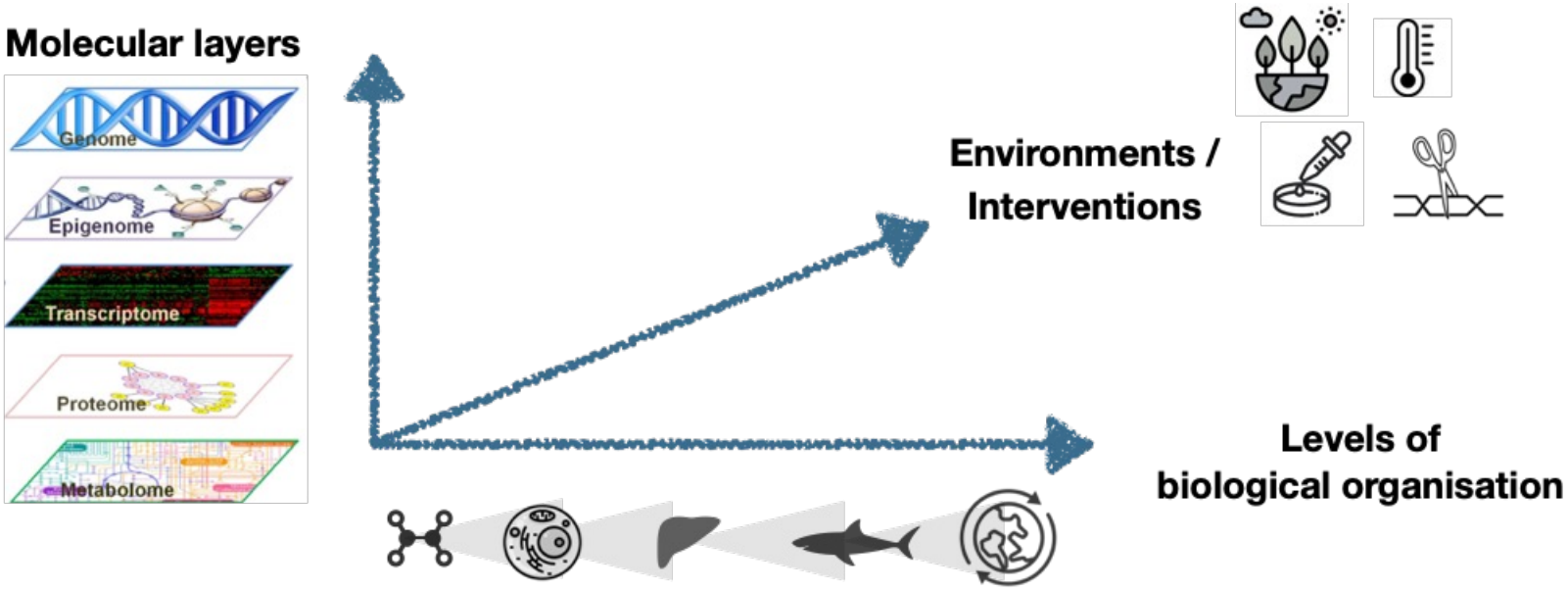
Models for multimodal data integration

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

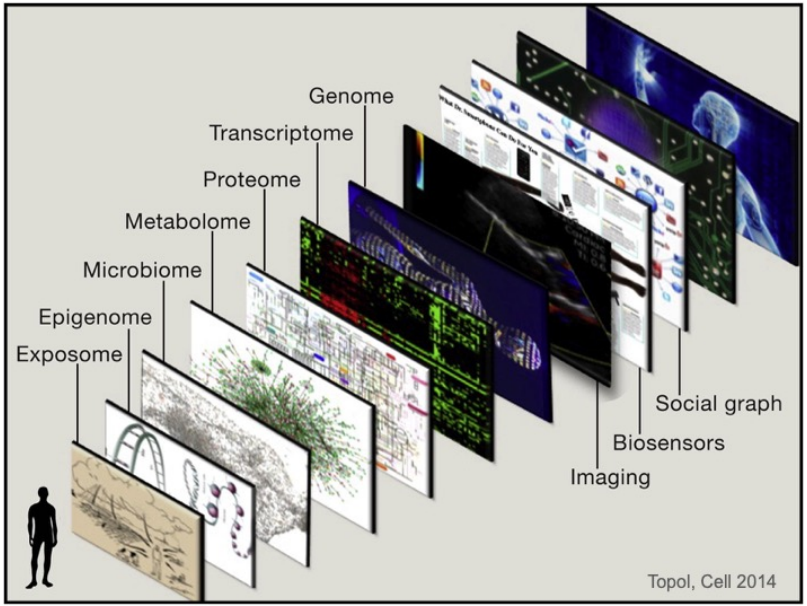
CHARLES UNIVERSITY
SORBONNE UNIVERSITÉ
UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386
UNIVERSITY OF WARSAW
UNIVERSITÀ DEGLI STUDI DI MILANO
4eu+ EUROPEAN UNIVERSITY ALLIANCE

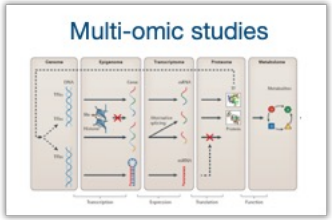
What are multi-modal data and why are they important?

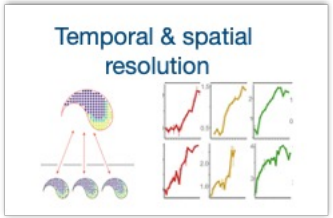
A holistic understanding of biological processes requires multi-dimensional approaches

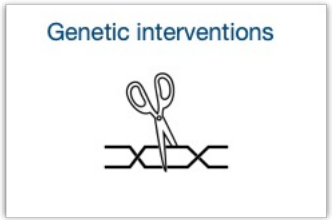


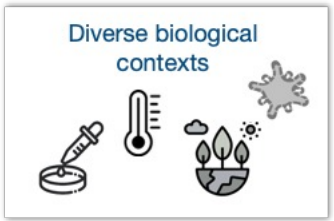
Technologies & data sources

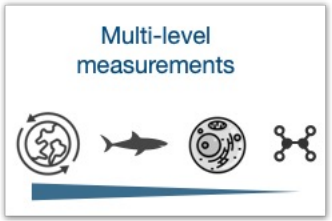






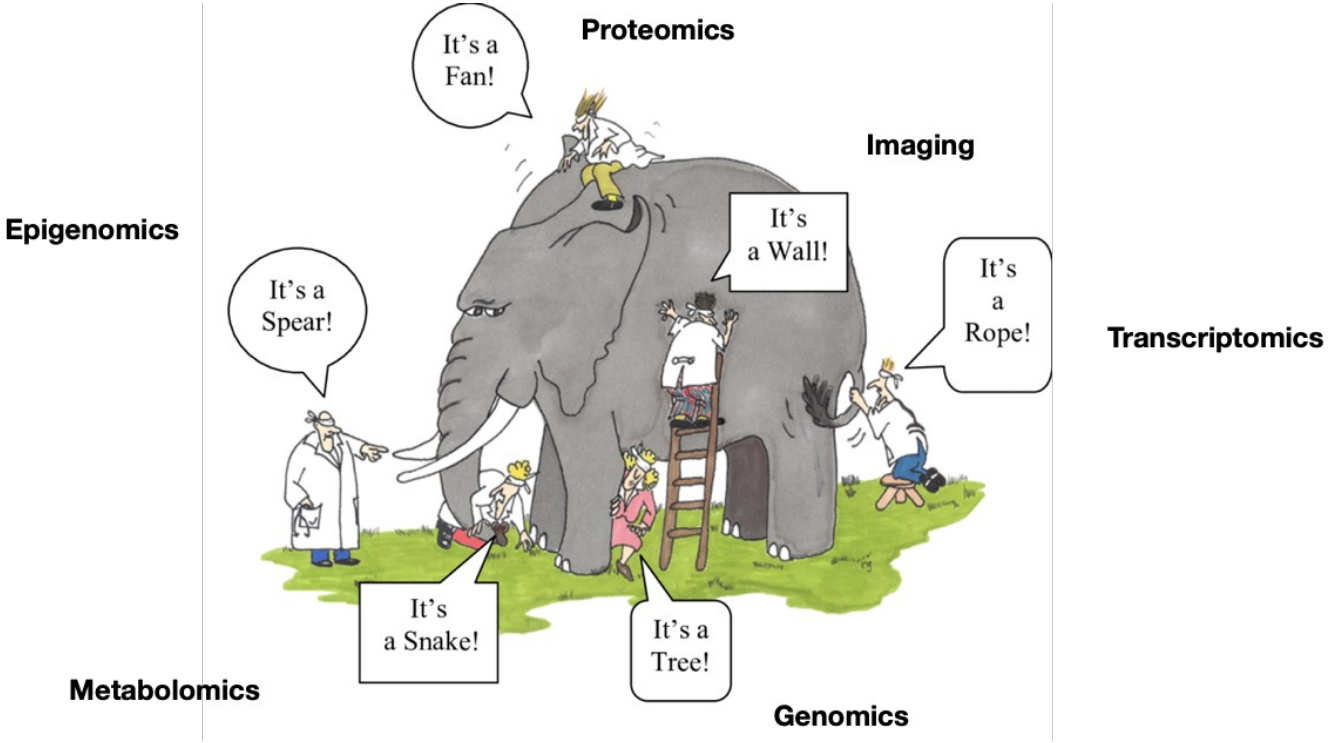






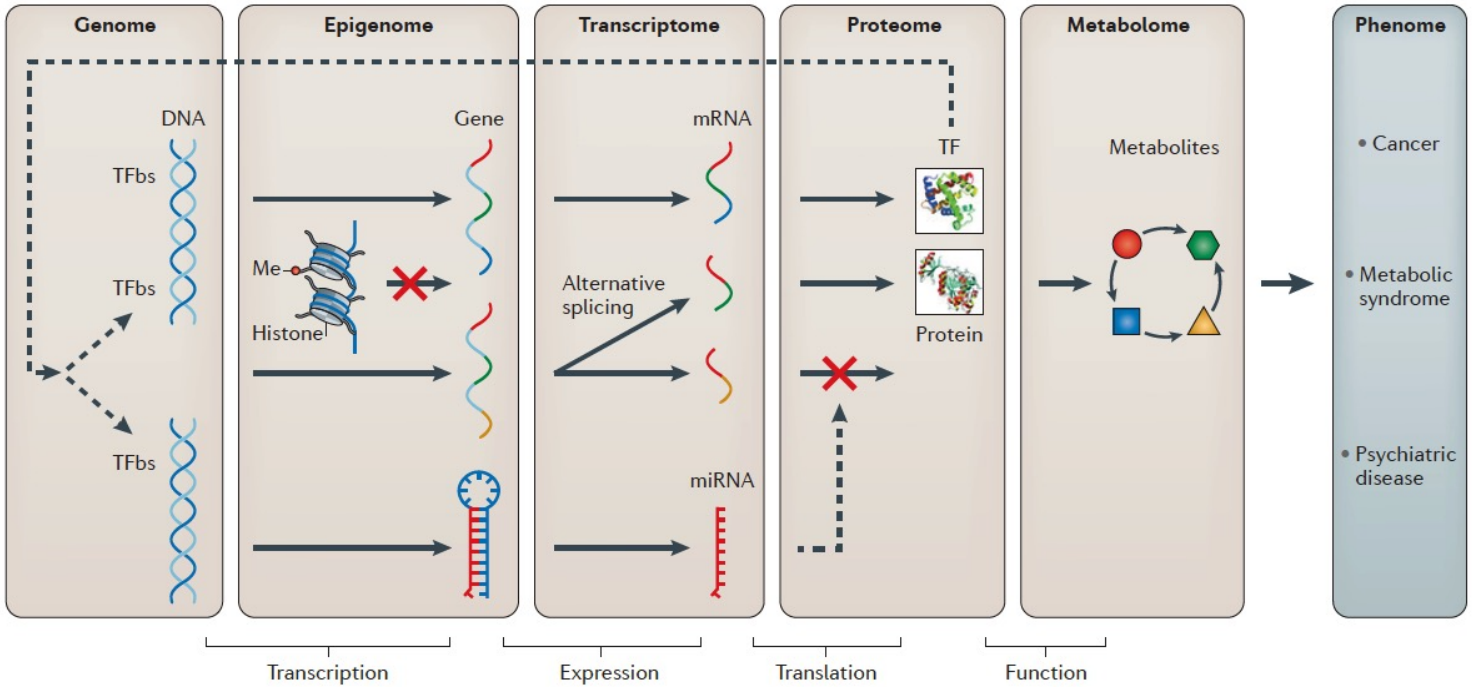
Challenge: *Interpretable data integration*

Limitations of single omics



Adapted from Daigneault, Methodological Innovations Online 2013

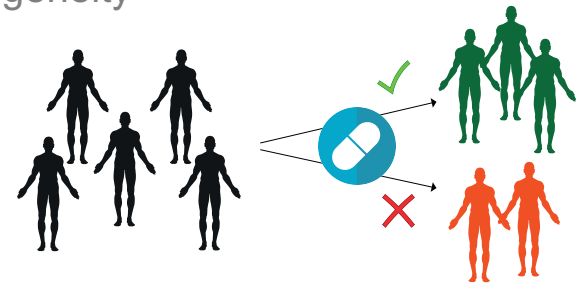
Interactions between molecular layers



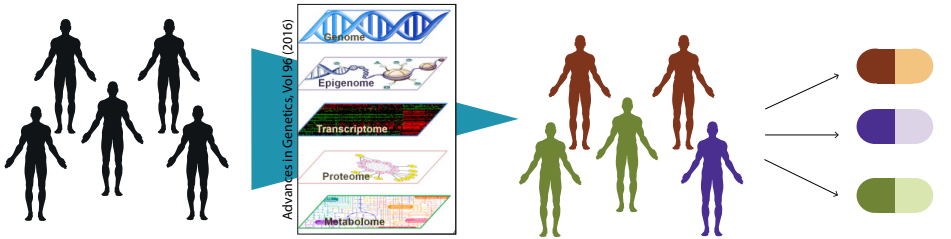
Ritchie et al, *Nat Rev Genet* 2015

Example 1: Precision Medicine

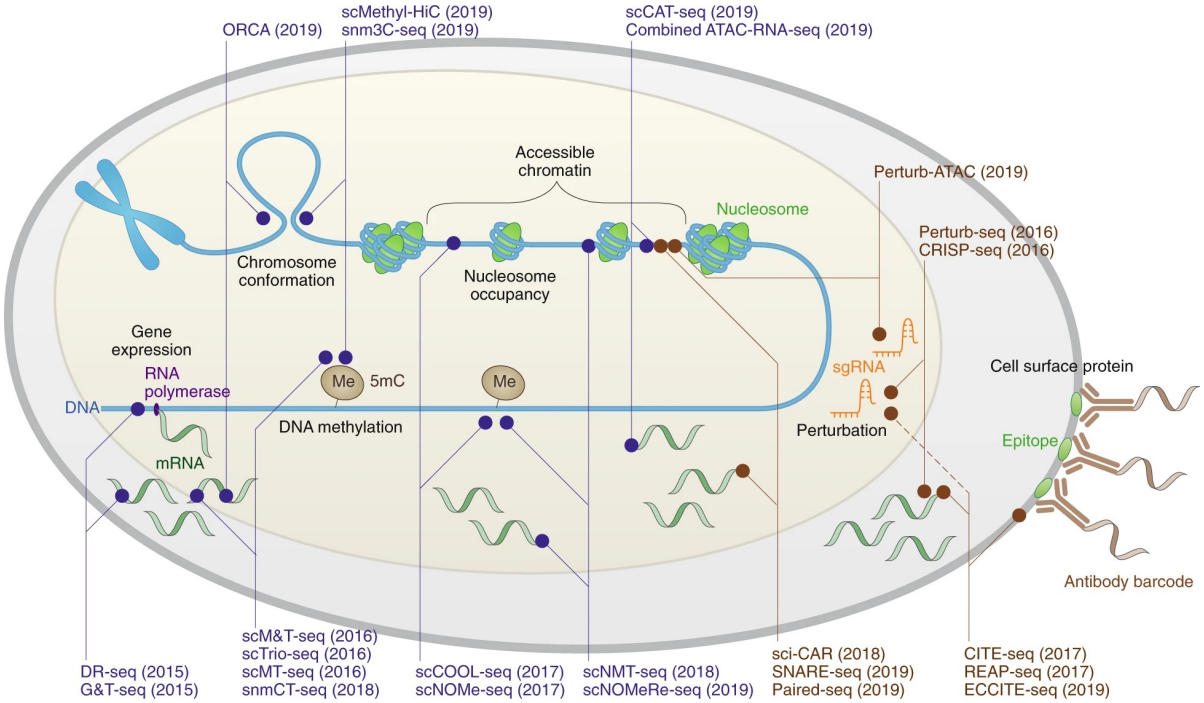
Challenge: Patient heterogeneity



Potential solution: Stratification based on multi-omics profiles

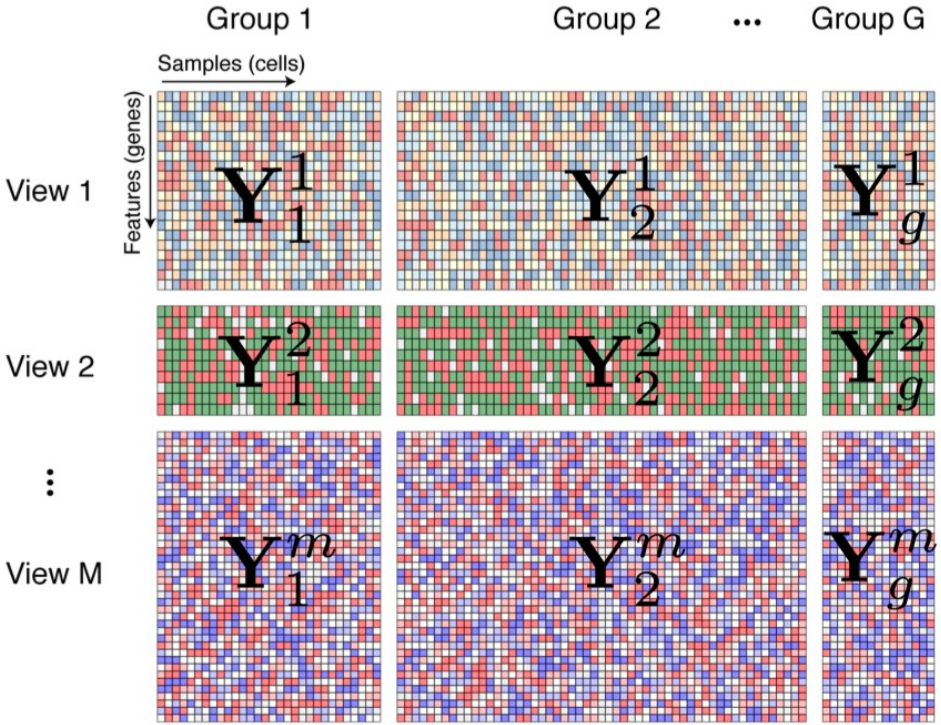


Example 2: Single cell biology



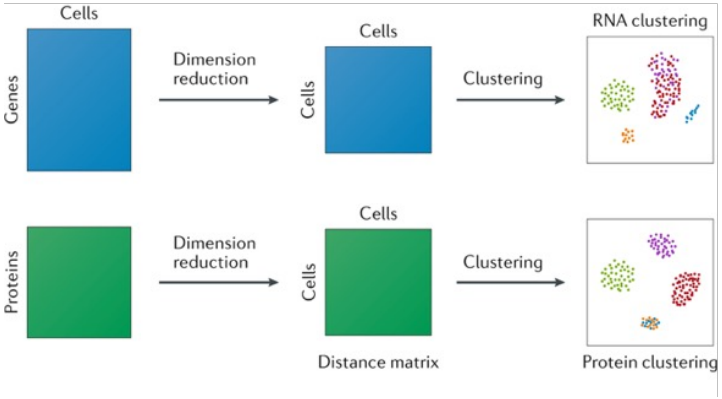
Zhu et al, Nat Meth 2020

Multi-modal data

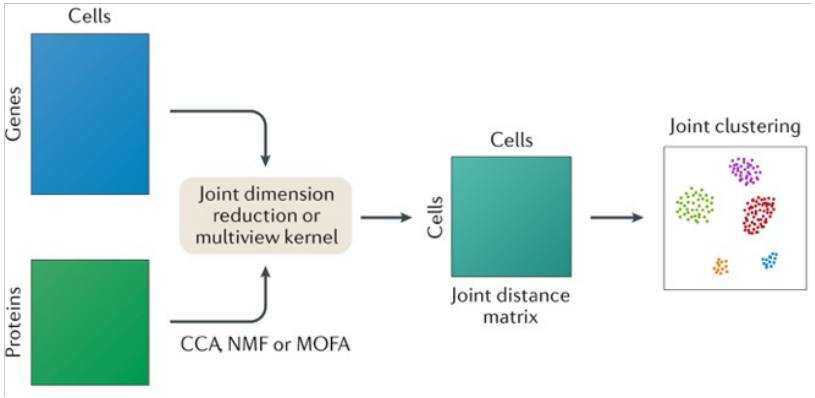


Why a joint analysis?

Separate analysis



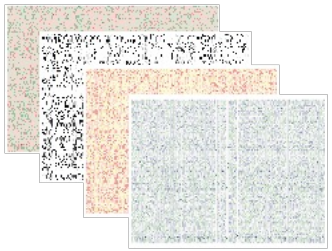
Joint analysis



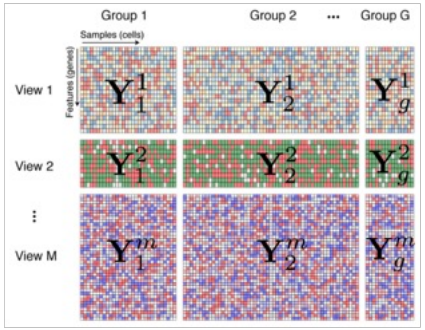
Adapted from Stuart & Satija, Nat Rev Gen 2019

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



heterogeneous

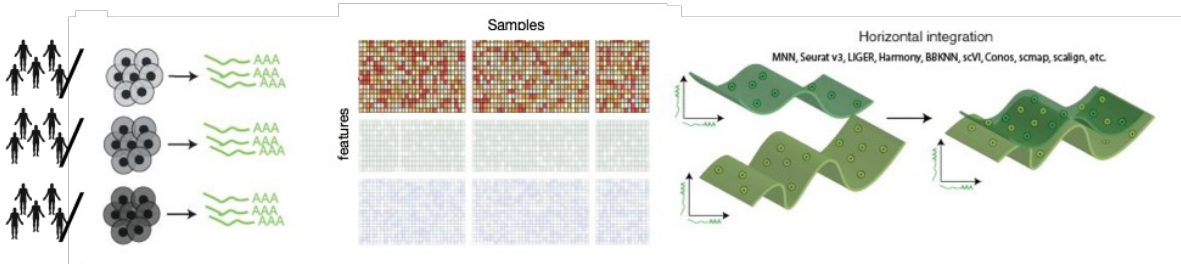


high-dimensional

How to integrate multi-modal data?

Step 1: Choose an anchor for integration

Horizontal integration
(features as anchors)

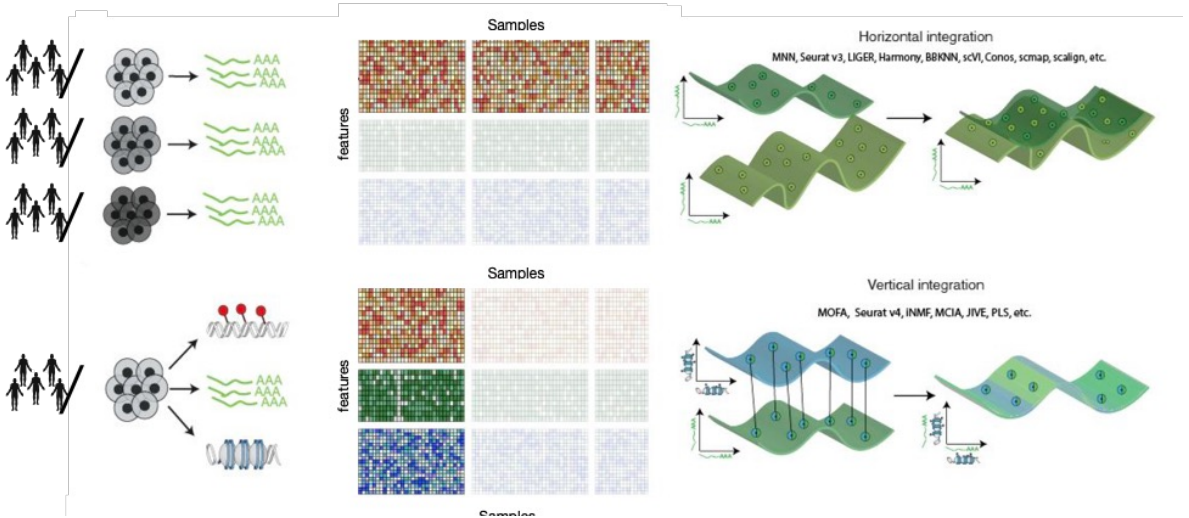


Adapted from Argelaguet*, Cuomo* et al. Nature Biotech 2021

Step 1: Choose an anchor for integration

Horizontal integration
(features as anchors)

Vertical integration
(samples as anchors)



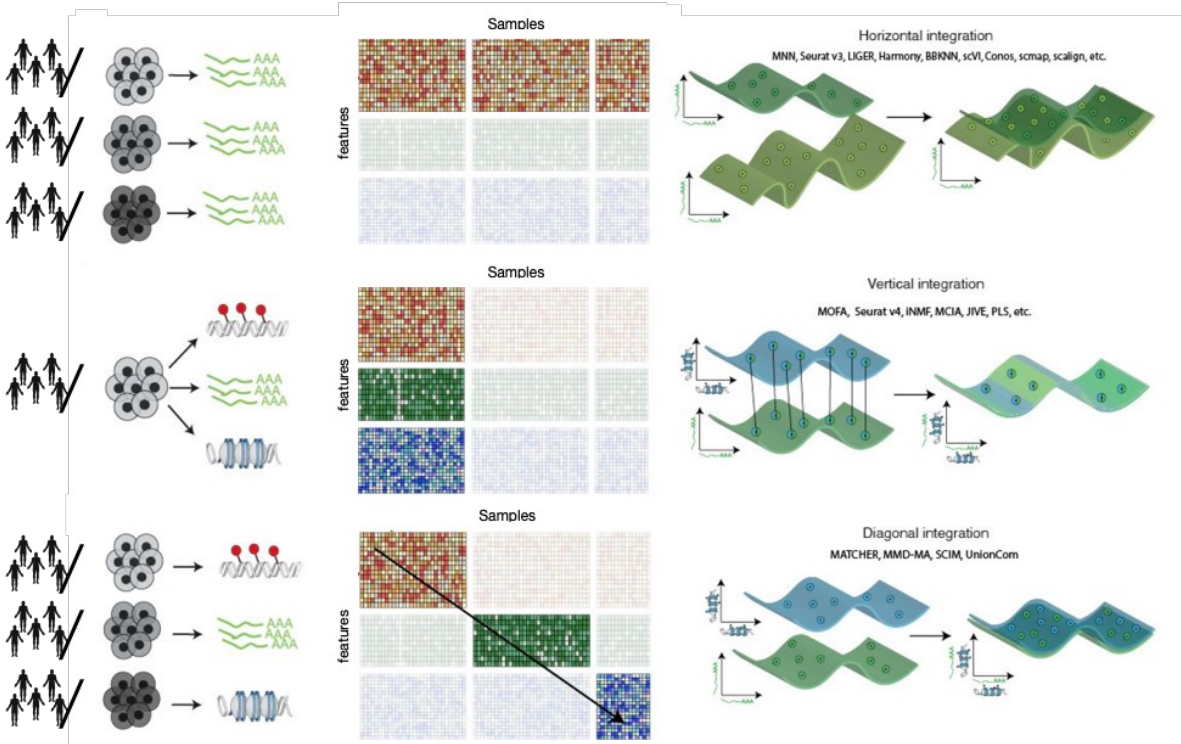
Adapted from Argelaguet*, Cuomo* et al. Nature Biotech 2021

Step 1: Choose an anchor for integration

Horizontal integration
(features as anchors)

Vertical integration
(samples as anchors)

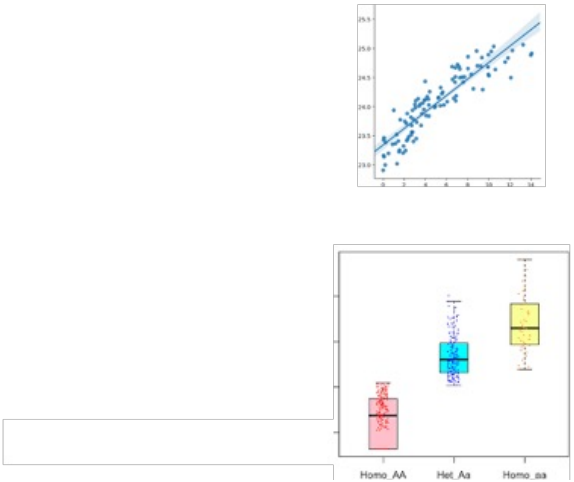
Diagonal integration
(no anchors)



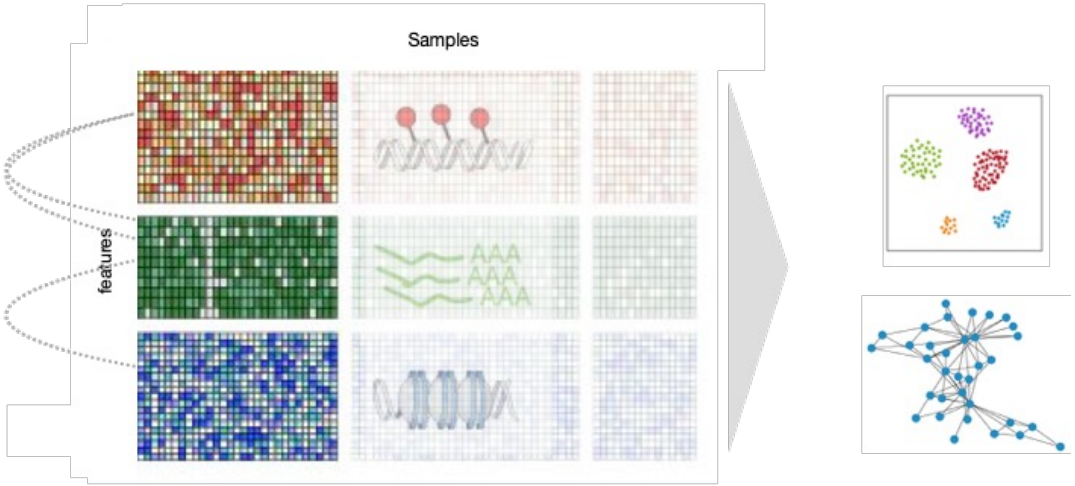
Adapted from Argelaguet*, Cuomo* et al. Nature Biotech 2021

Step 2: What type of associations?

Local analysis



Global analysis



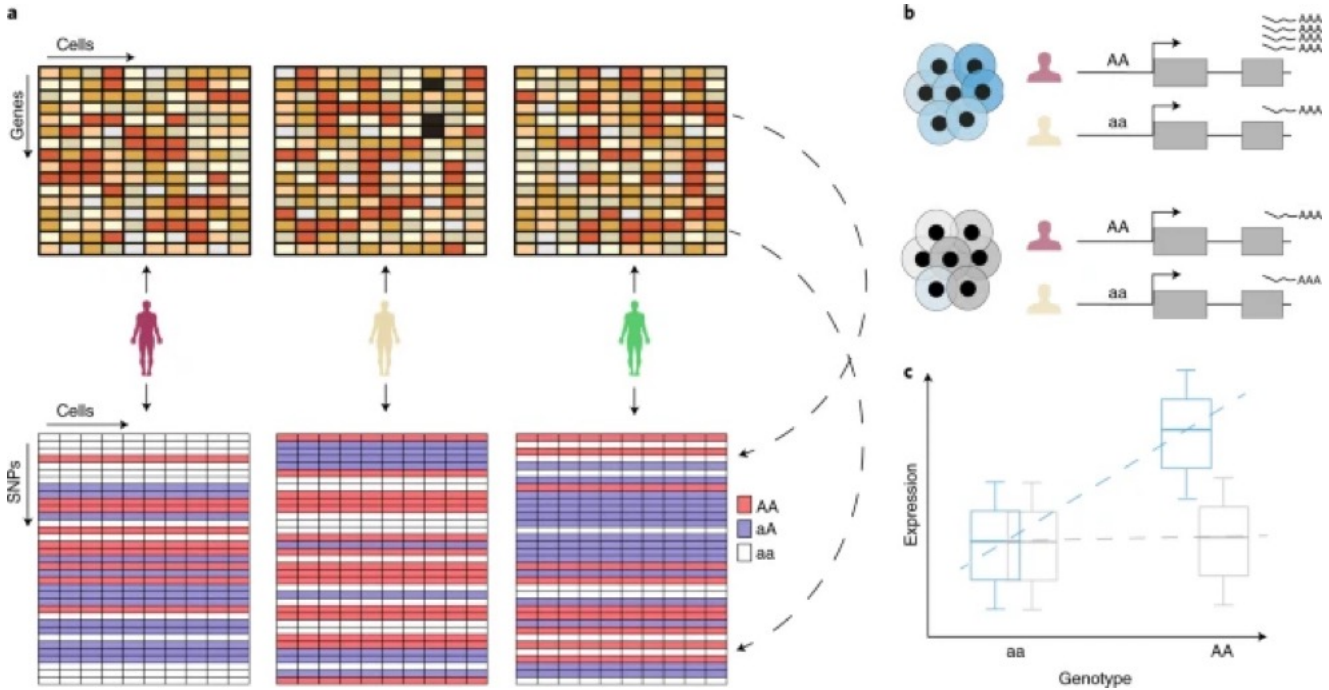
Local analysis

- **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



Adapted from Argelaguet*, Cuomo* et al. Nature Biotech 2021

Considerations for local analysis

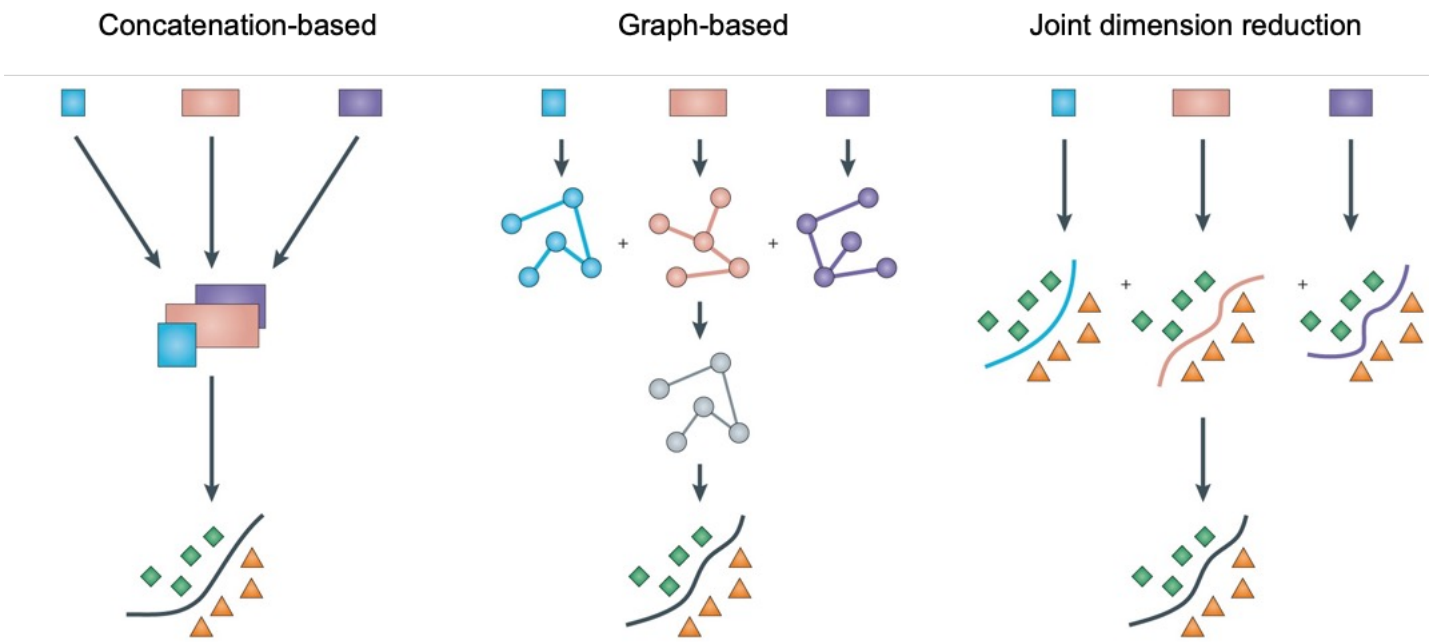
- Linear or non-linear relationship Y = f(X)
- Univariate or multivariate X, need for regularisation?
- Noise model?
- Confounding factors, error covariance?

Global analysis

- **Aim**: Find major sources of variation and identify sample structures based on multiple molecular layers, e.g.
  - Single cell clustering
  - Patient stratification

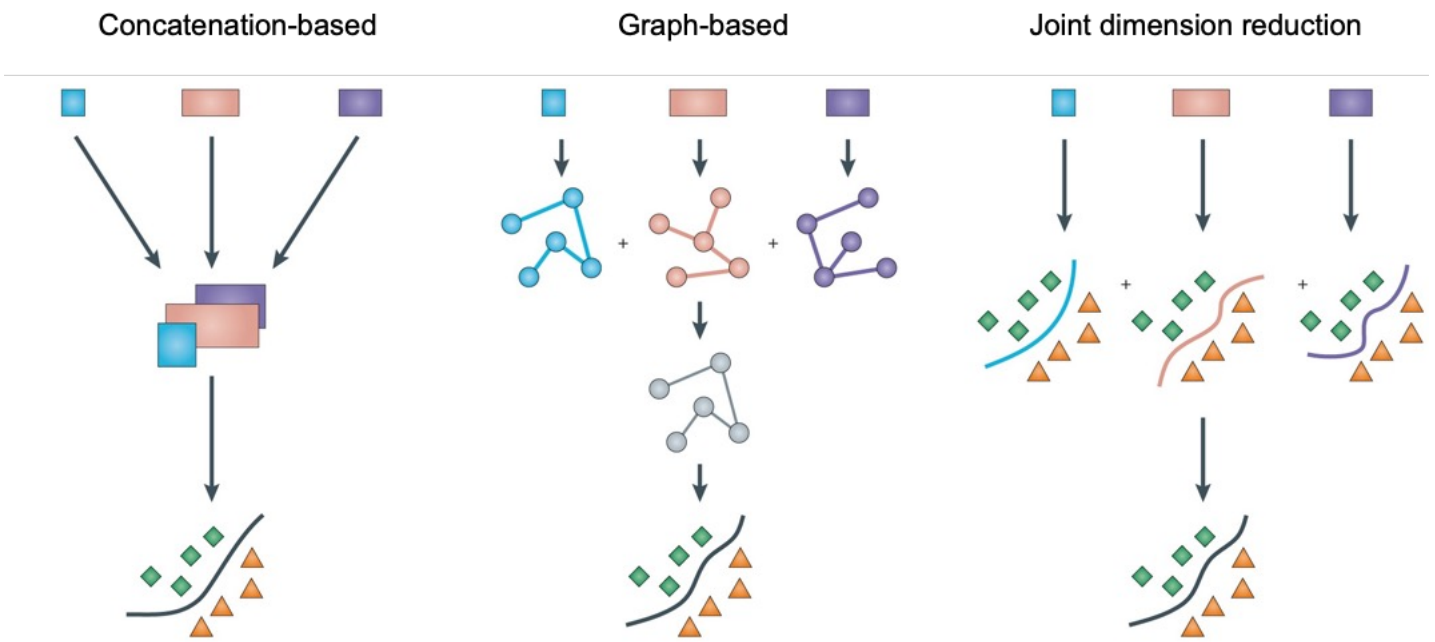
- **Methods**: Mostly unsupervised models, e.g.
  - joint dimension reduction or clustering
  - analysis of covariance structures among features
  - graph-based models

Global analysis



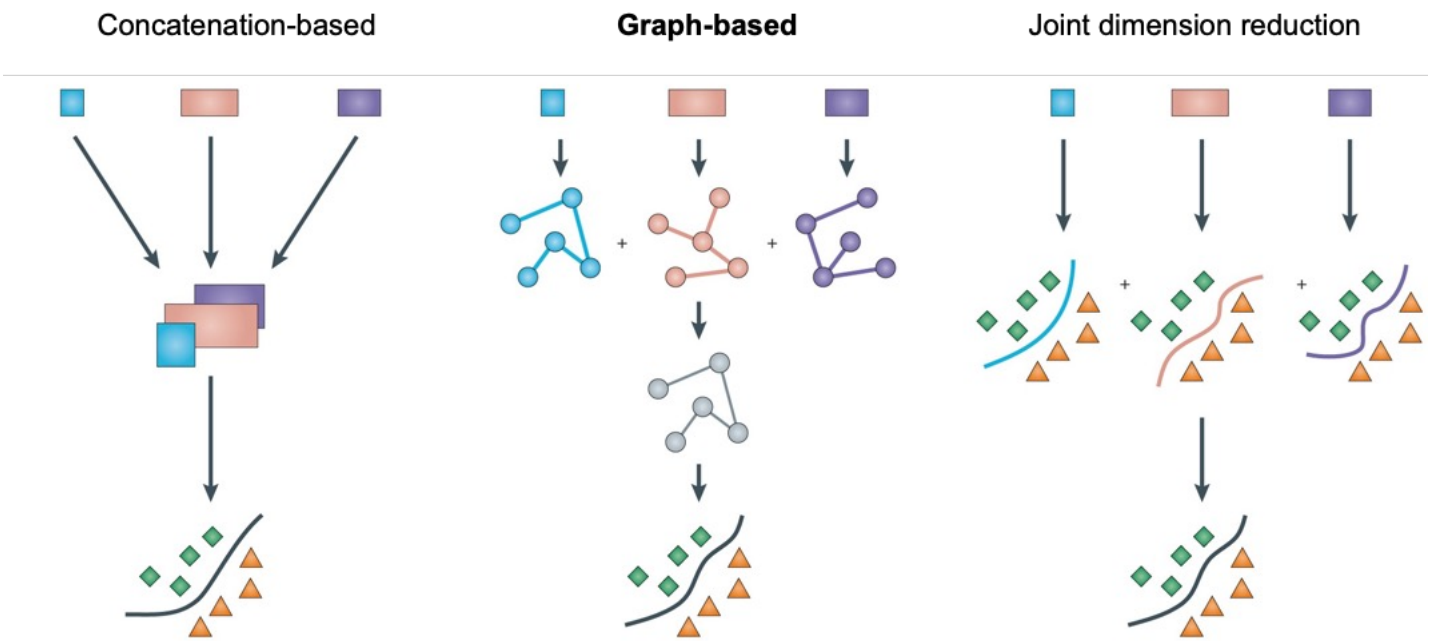
Adapted from Ritchie et al, Nat Rev Gen 2015

Global analysis



Adapted from Ritchie et al, Nat Rev Gen 2015

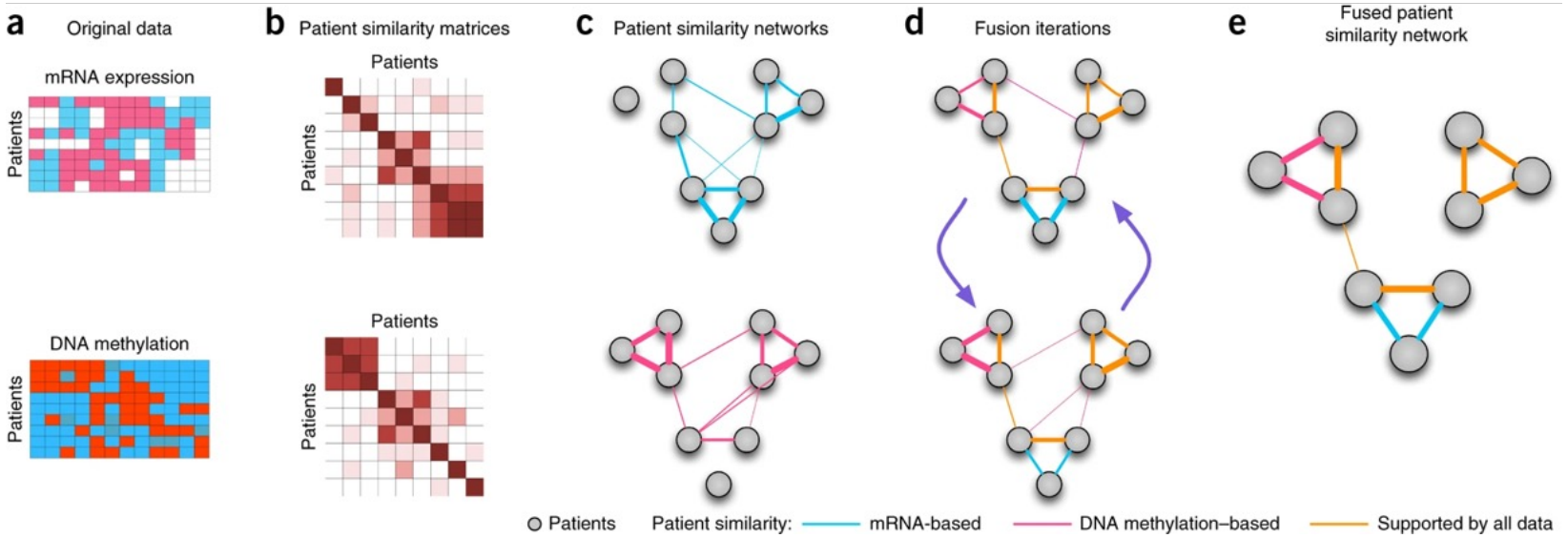
Global analysis



Adapted from Ritchie et al, Nat Rev Gen 2015

Examples for graph based-methods

Similarity network fusion (SNF)

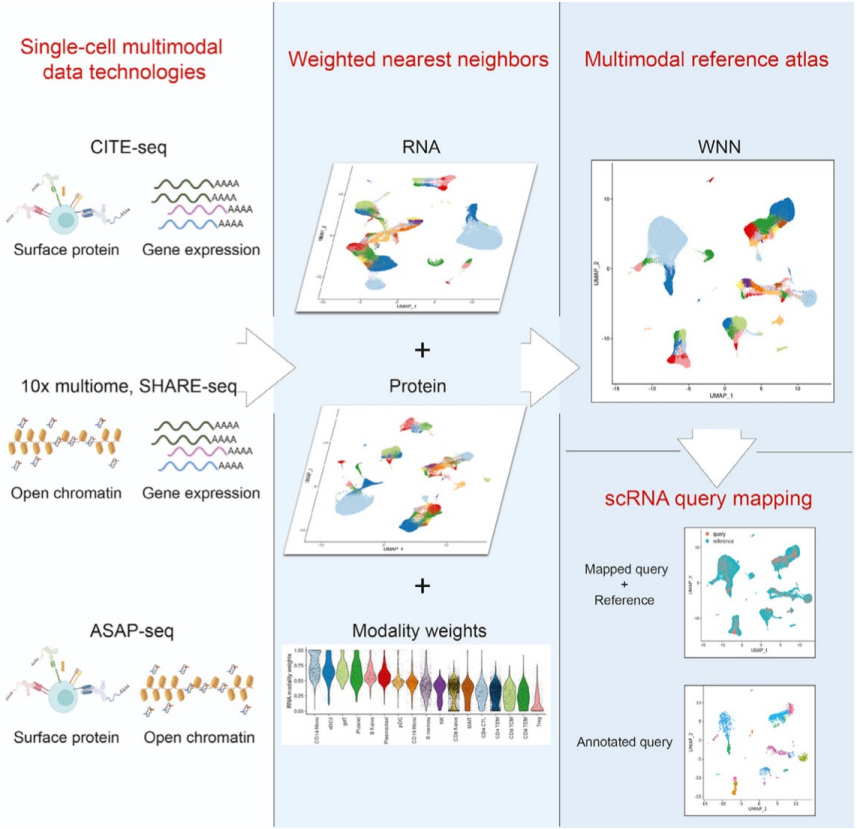


Wang et al 2014

Examples for graph based-methods

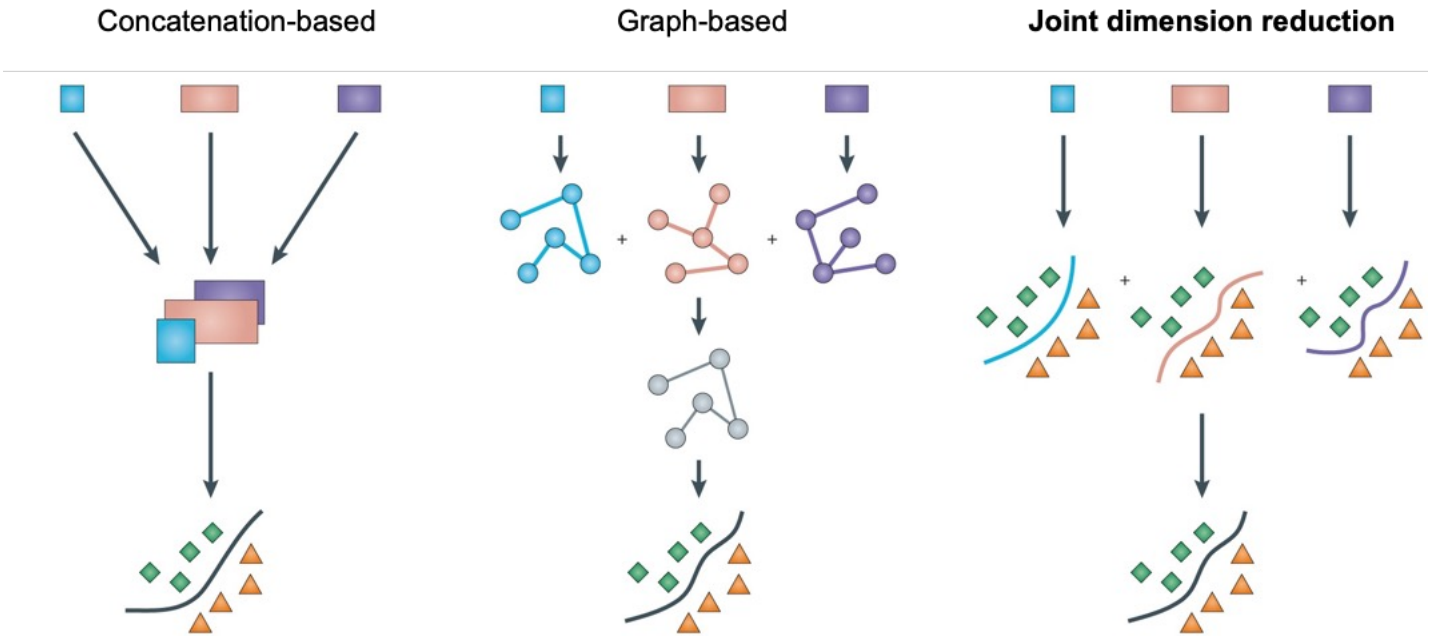
Weighted Nearest Neighbors

- Compare prediction performace in one modality using nearest neighbors from another modality
- Similarity graph based on resulting sample-wise weights

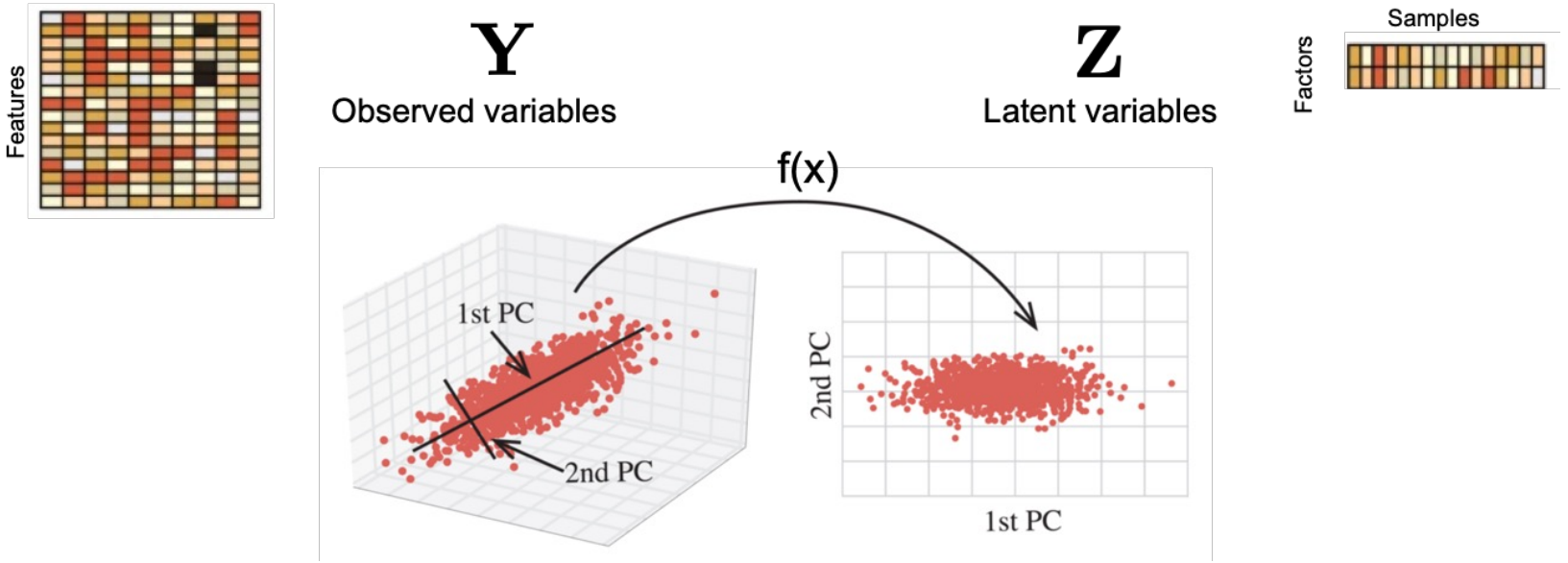


Hao, Hao et al, Cell 2021

Global analysis



Recap: Latent variable models



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

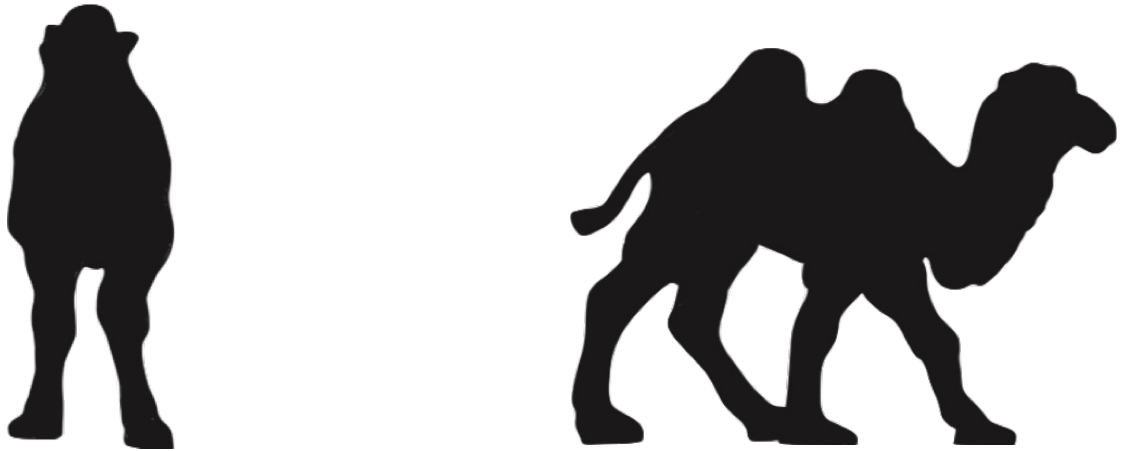
Common Aim

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

Adapted from Homes & Huber, Modern Statistics for Moden Biology (MSMB)

Common Aim

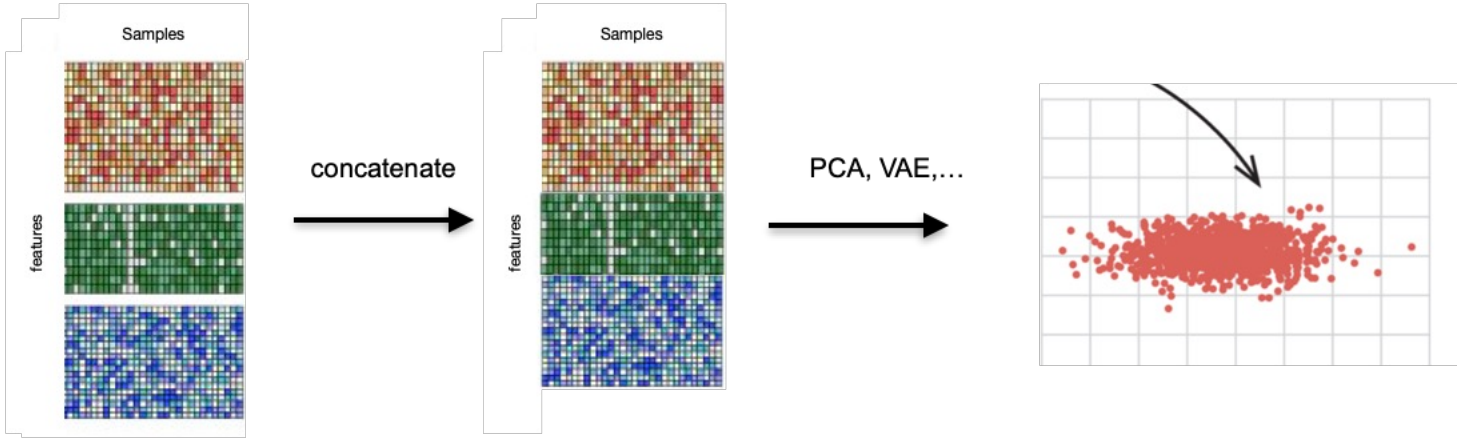
Maximize variance / minimize recontruction error in the low-dimensional space



Adapted from Homes & Huber, Modern Statistics for Moden Biology (MSMB)

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

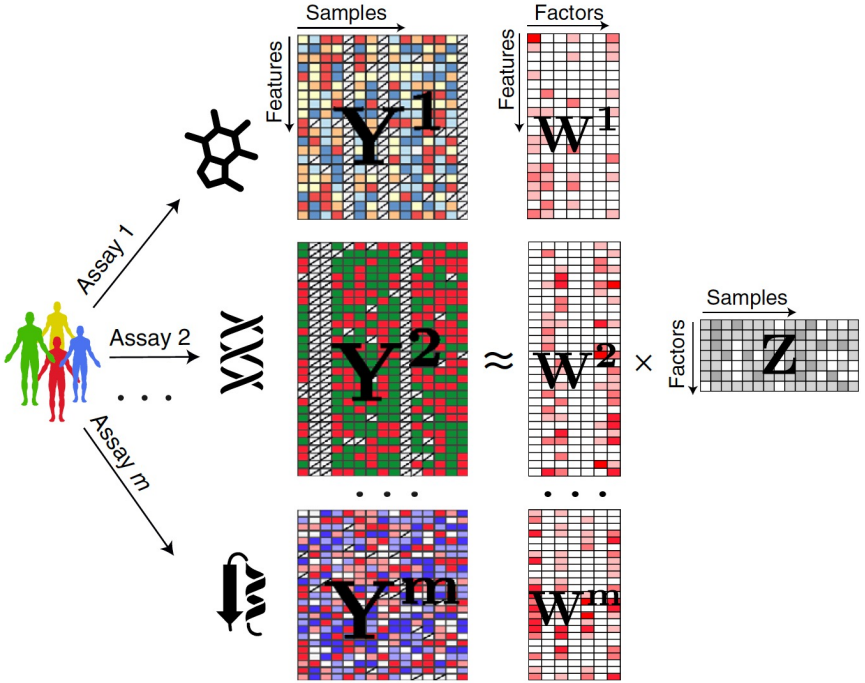
Consensus PCA

Partial least squares (PLS)

Multi-view variational auto-encoders

MOFA: Probabilistic factor model for multi-modal data

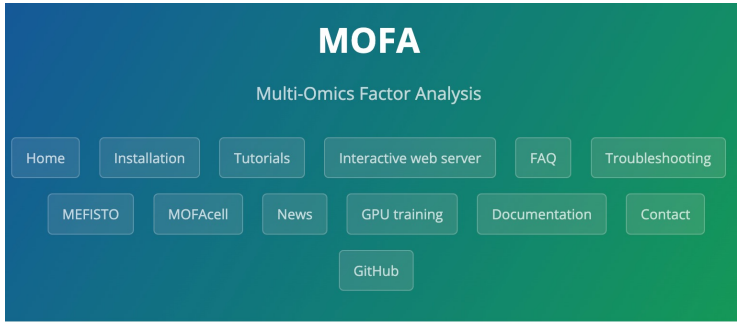
- joint **low-dimensional representation** in terms of (hidden) factors representing the principal axes of variation
- Interpretable model using a **two-level sparsity priors**
  - *Which modalities are important for a factor?*
  - *Which features are important for a factor?*
- **Scalable inference**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



MOFA: Argelaguet* & Velten* et al Mol Sys Bio. 2018
MOFA+: Argelaguet*, Arnol*, Bredikhin* et al Genome Biology. 2020

MOFA - Software

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

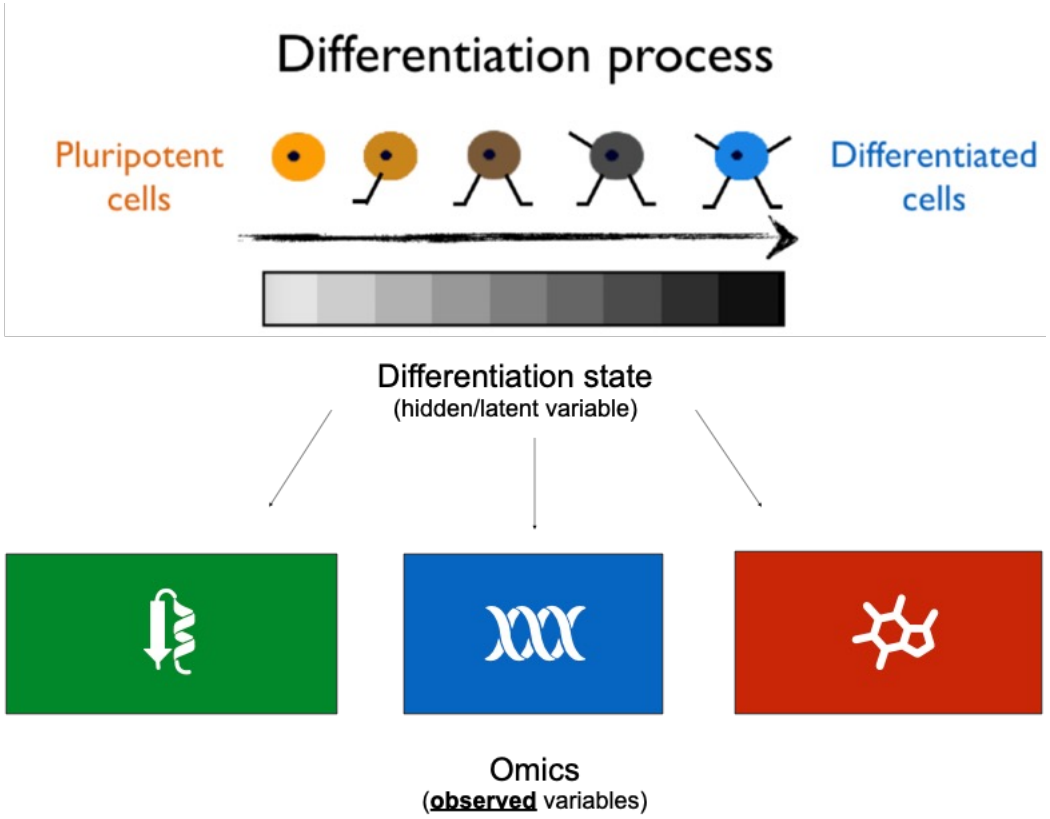


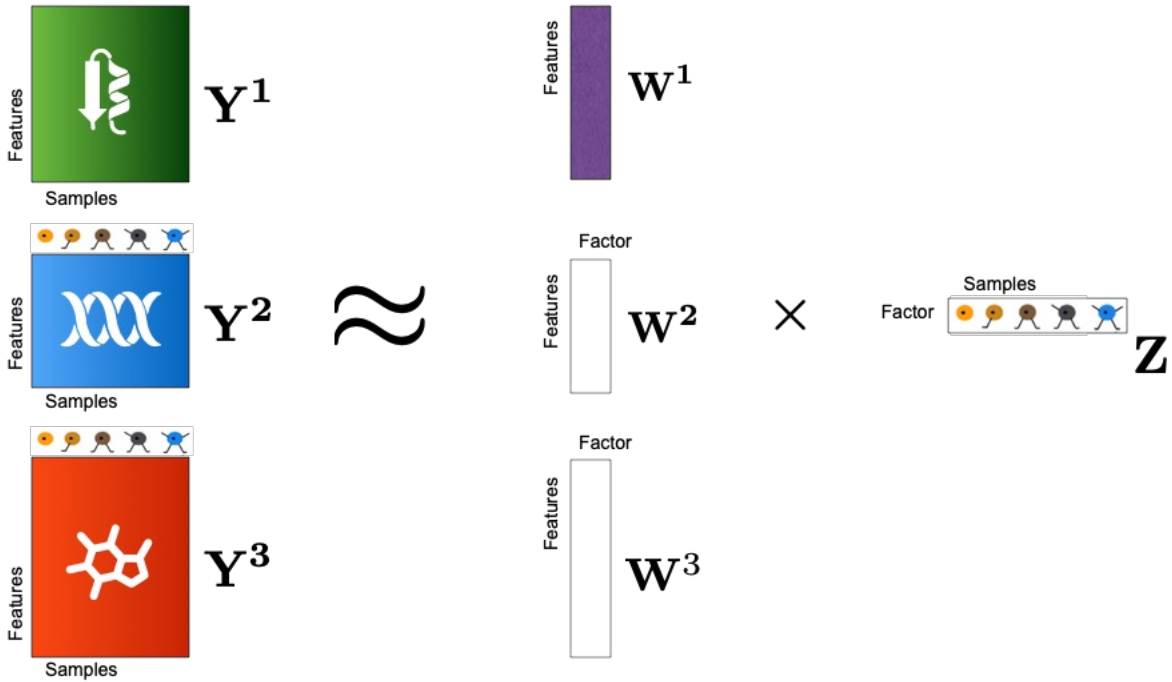
Overview

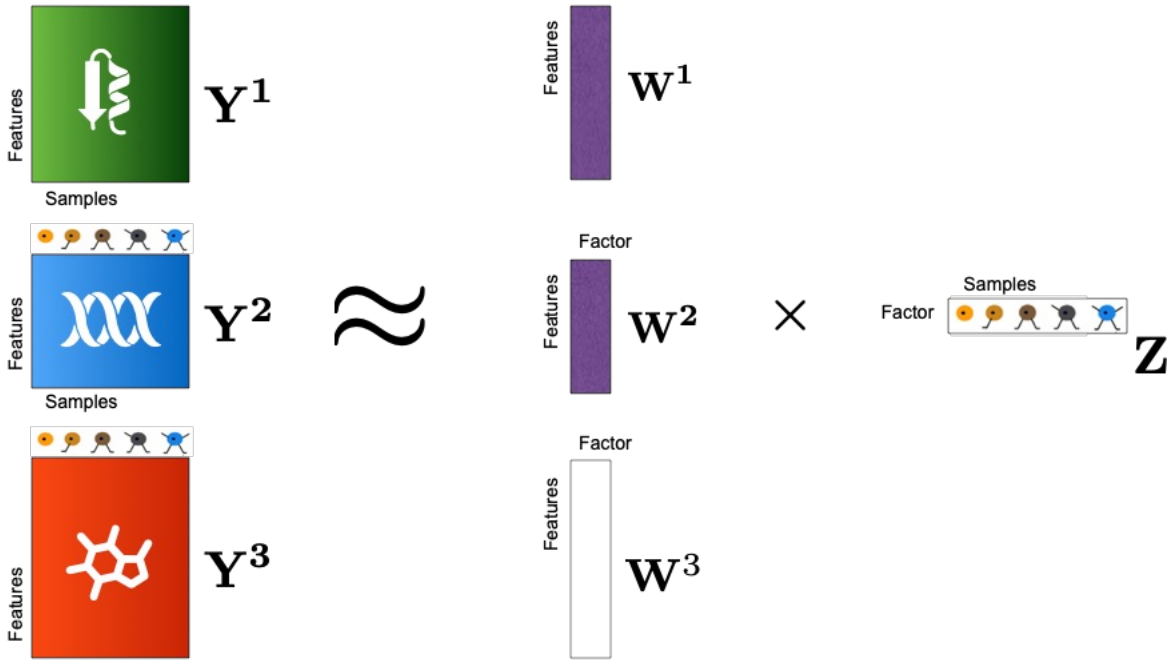
MOFA is a factor analysis model that provides a **general framework for the integration of multi-omic data sets** in an unsupervised fashion. Intuitively, MOFA can be viewed as a versatile and statistically rigorous generalization of principal component analysis to multi-omics data. Given several data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

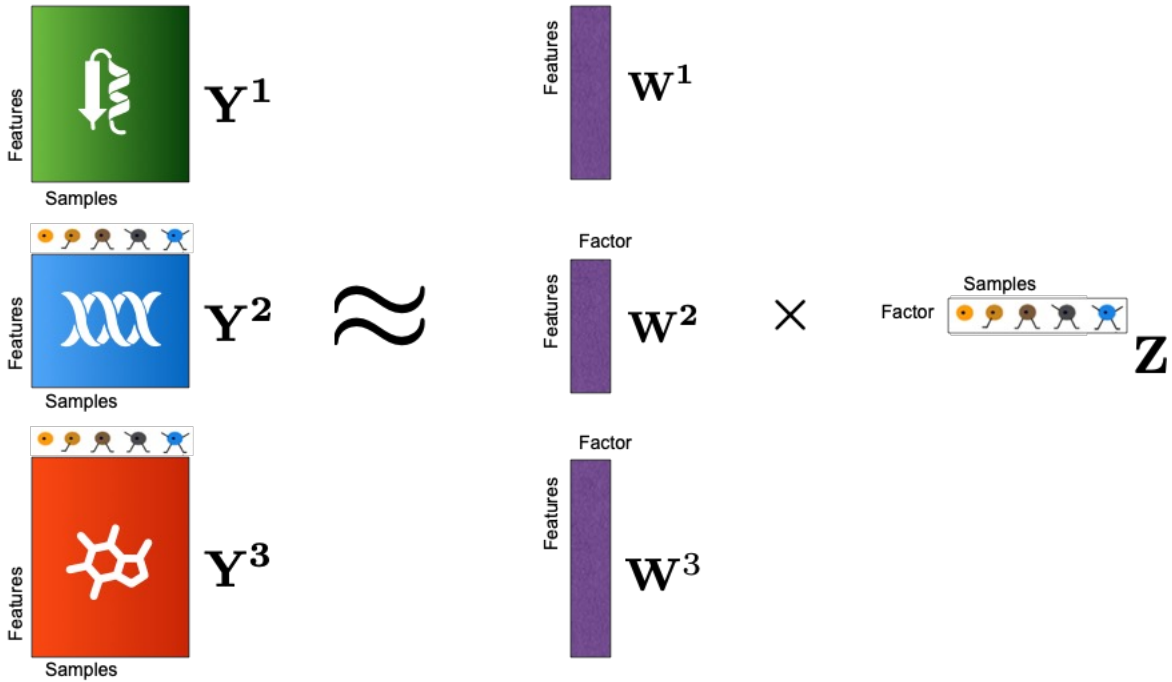
Python: mofapy2, muon
R: MOFA2

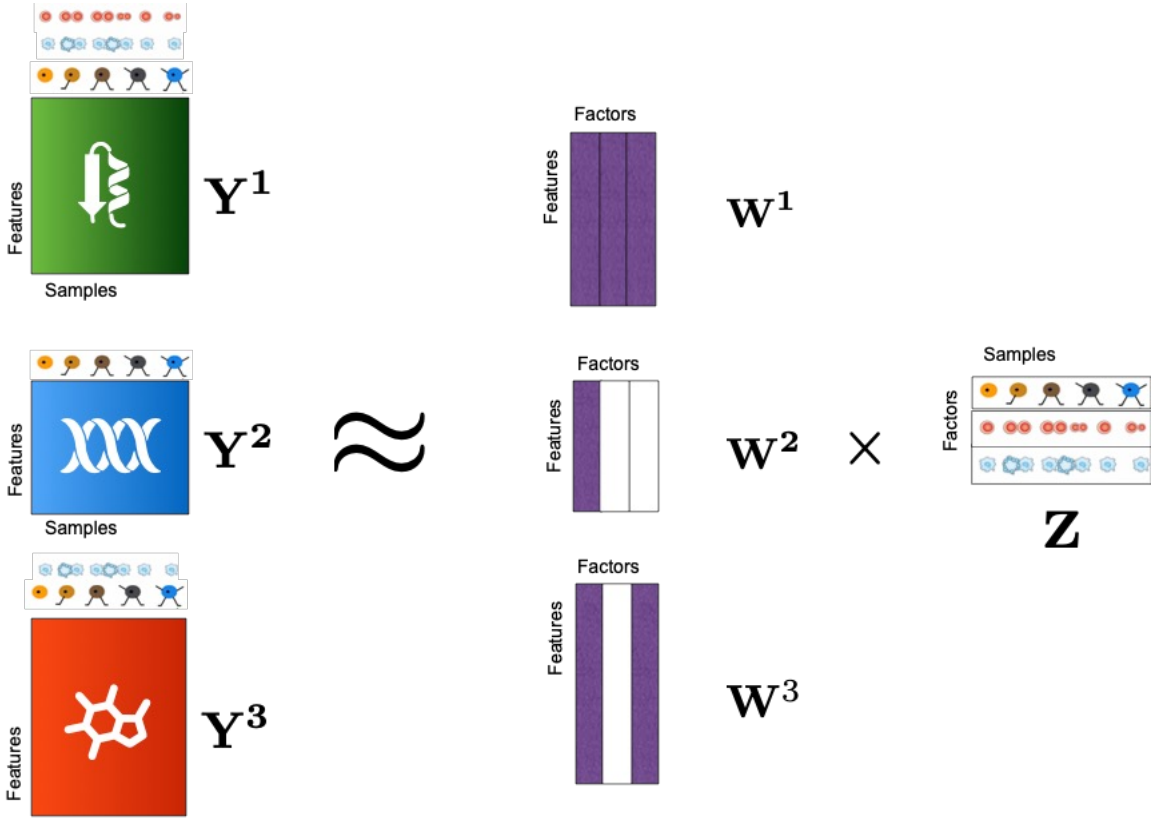
What is the intuition behind MOFA?


Differentiation process
Pluripotent cells
Differentiated cells
Differentiation state
(hidden/latent variable)
Omics
(observed variables)


Features
Samples
$\mathbf{Y^1}$
Features
$\mathbf{W^1}$
Features
Samples
$\mathbf{Y^2}$
$\approx$
Factor
Features
$\mathbf{W^2}$
$\times$
Samples
Factor
$\mathbf{Z}$
Features
Samples
$\mathbf{Y^3}$
Factor
Features
$\mathbf{W^3}$

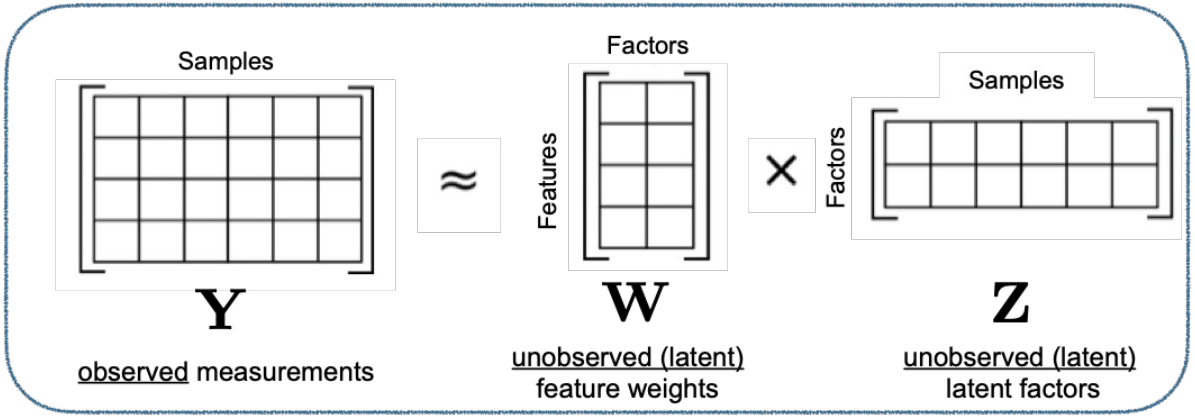

Features
Samples
$\mathbf{Y^1}$
Features
$\mathbf{W^1}$
Features
Samples
$\mathbf{Y^2}$
$\approx$
Factor
Features
$\mathbf{W^2}$
$\times$
Samples
Factor
$\mathbf{Z}$
Features
$\mathbf{Y^3}$
Factor
Features
$\mathbf{W^3}$


Features
Samples
$\mathbf{Y^1}$
Features
$\mathbf{W^1}$
Features
Samples
$\mathbf{Y^2}$
$\approx$
Factor
Features
$\mathbf{W^2}$
$\times$
Samples
Factor
$\mathbf{Z}$
Features
Samples
$\mathbf{Y^3}$
Factor
Features
$\mathbf{W^3}$


Features
Samples
$\mathbf{Y^1}$
Features
Samples
$\mathbf{Y^2}$
Features
Samples
$\mathbf{Y^3}$
$\approx$
Factors
Features
$\mathbf{W^1}$
Factors
Features
$\mathbf{W^2}$
Factors
Features
$\mathbf{W^3}$
$\times$
Samples
Factors
$\mathbf{Z}$

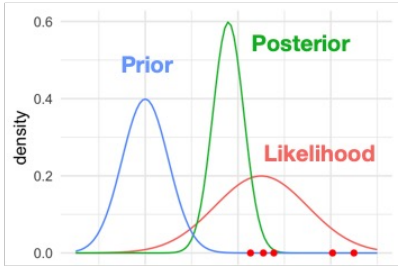
What is the mathematical model behind MOFA?

Probabilistic factor model

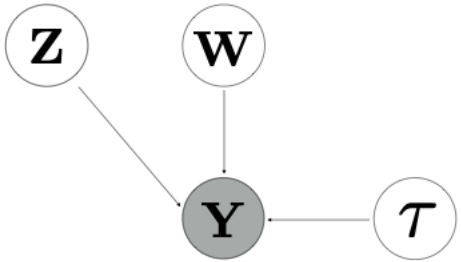


Probabilistic (Bayesian) models combine
- a *likelihood function* (statistical model for the observed data)
- a *prior* (distribution of unobserved components)

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



Likelihood model



Observed variable

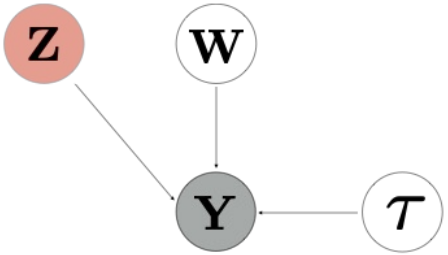
Unobserved variable

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

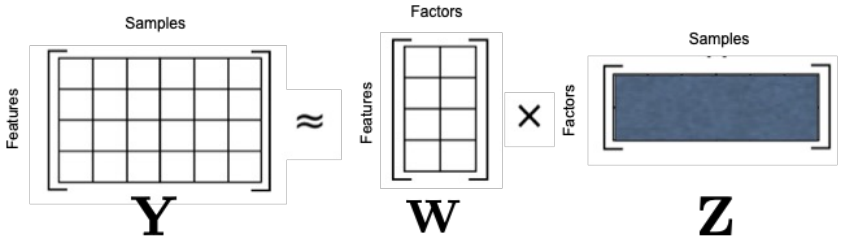
		Dimensions
$\mathbf{Y}$	Observation matrix	(N,D)
$\mathbf{Z}$	Latent variable matrix	(N,K)
$\mathbf{W}$	Weight matrix	(D,K)
τ	Noise vector	(D,)

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

Prior for factors

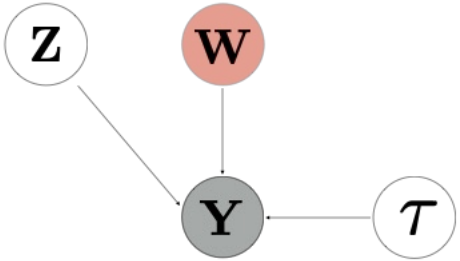


$$p(\mathbf{Z}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \mathcal{N}(0,1)$$



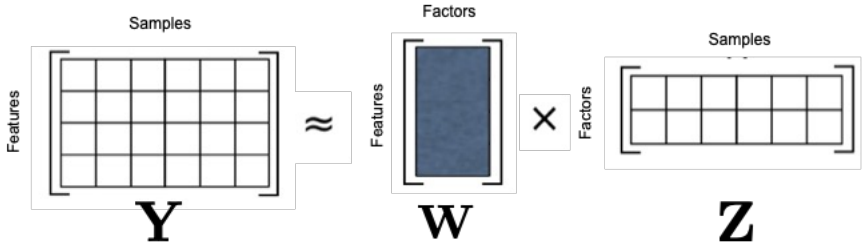
N is the number of samples
D is the number of features
K is the number of factors

Prior for weights



Automatic Relevance Determination (ARD)

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



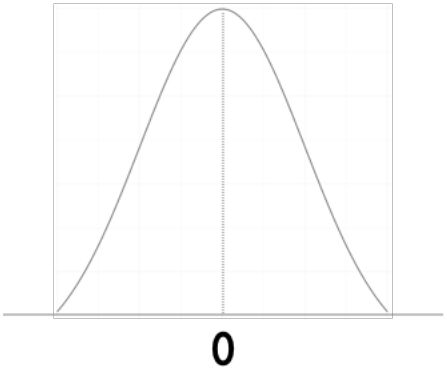
N is the number of samples
D is the number of features
K is the number of factors

Automatic relevance determination prior

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

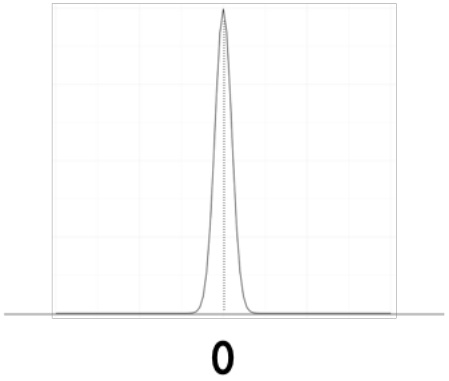
If precision α is small:

Broad distribution

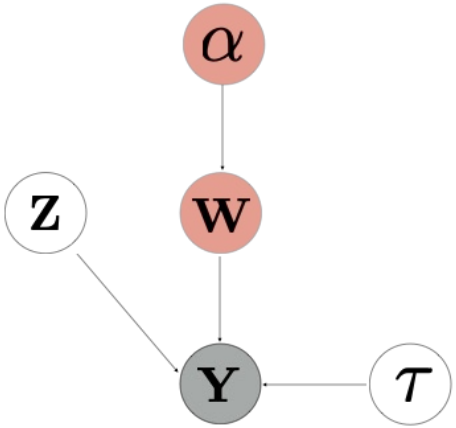


If precision α is large:

Distribution peaked at 0



Automatic relevance determination prior

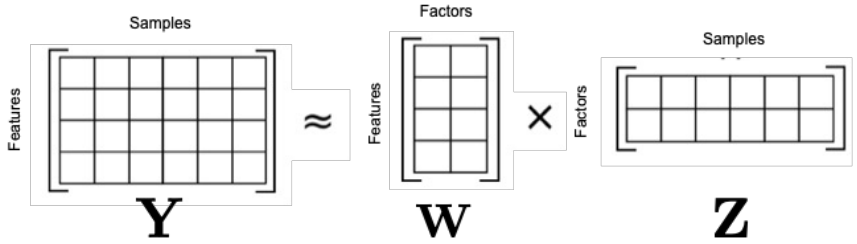


$$p(\mathbf{W}|\alpha) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{w}_{:,\mathbf{k}}|0, \alpha_k^{-1}\mathbf{I})$$

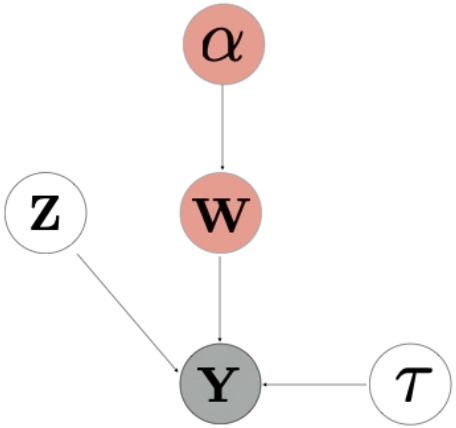
$$p(\alpha_k) = \mathcal{G}(a_0^{\alpha}, b_0^{\alpha})$$

Uninformative gamma prior

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



Automatic relevance determination prior

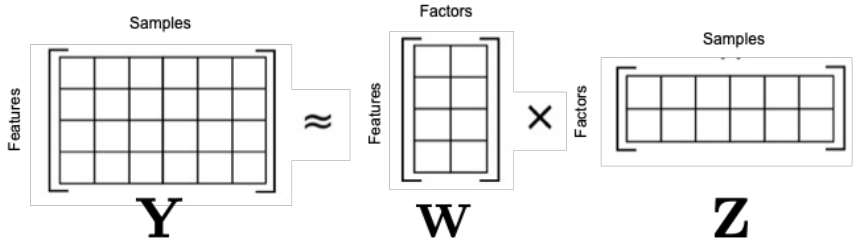


$$p(\mathbf{W}|\alpha) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{w}_{:,\mathbf{k}}|0, \alpha_k^{-1}\mathbf{I})$$

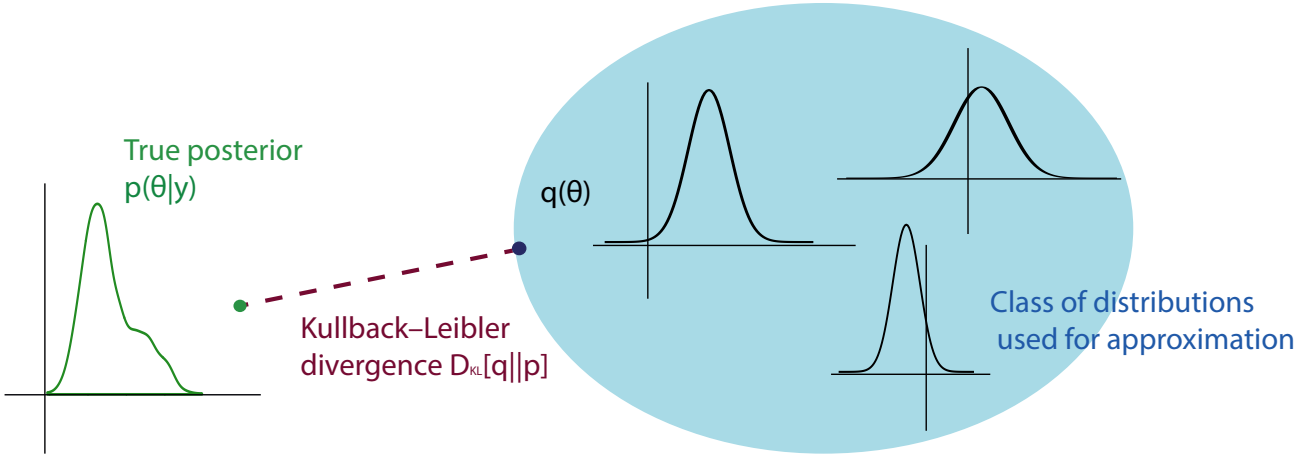
$$p(\alpha_k) = \mathcal{G}(a_0^{\alpha}, b_0^{\alpha})$$

Uninformative gamma prior

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

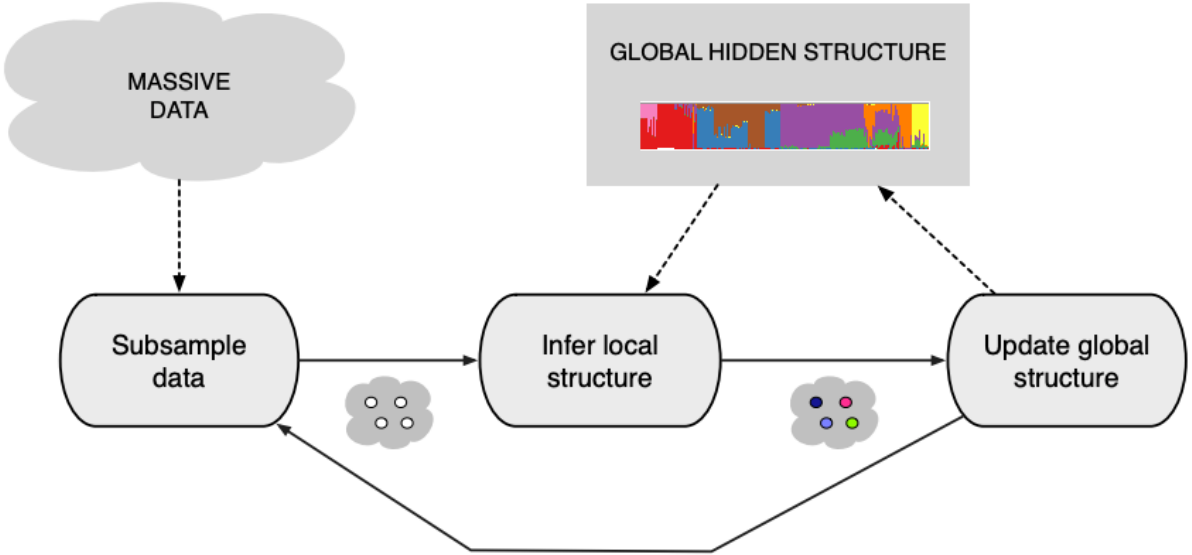


Finding the posterior $p(Z, W, \alpha|Y)$



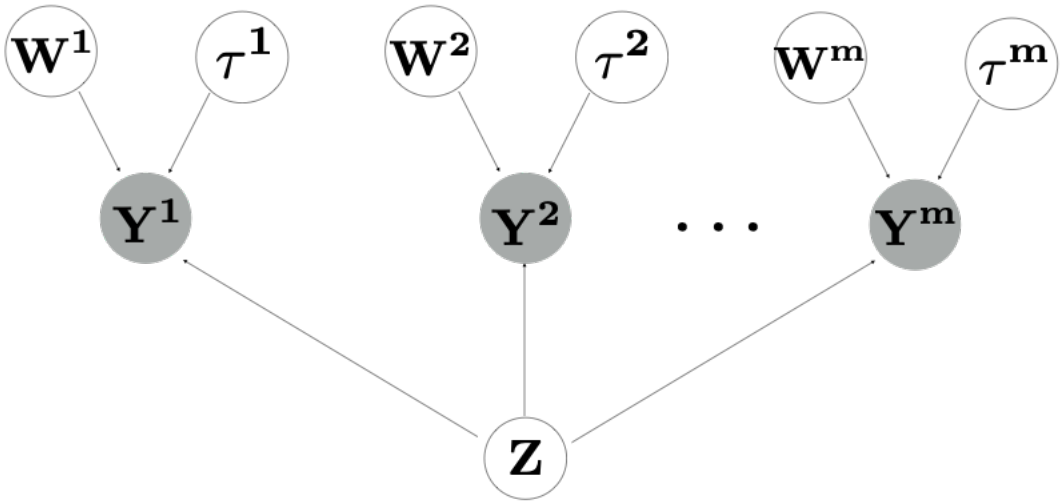
Variational inference provides an approximation of the posterior from a restricted class of distributions.

Stochastic variational inference



Adapted from David Blei, NIPS tutorial 2016

Multi-view probabilistic factor analysis



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

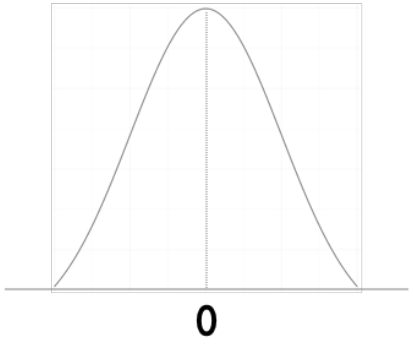
M is the number of “views” (omics)

Determining activity of factors per view

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

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

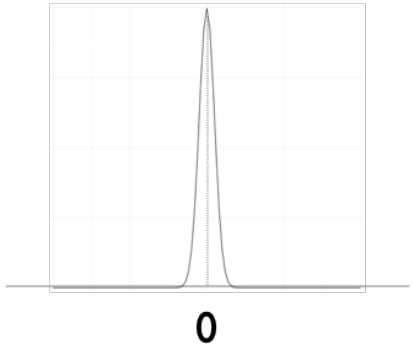
Broad distribution



Factor k is active in view m

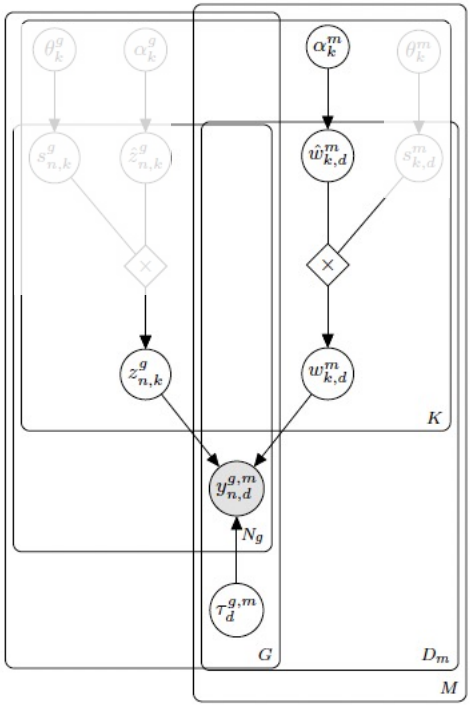
If precision $\alpha_k^{(m)}$ is large:

Distribution peaked at 0



Factor k is not active in view m

Plate notation of graphical model

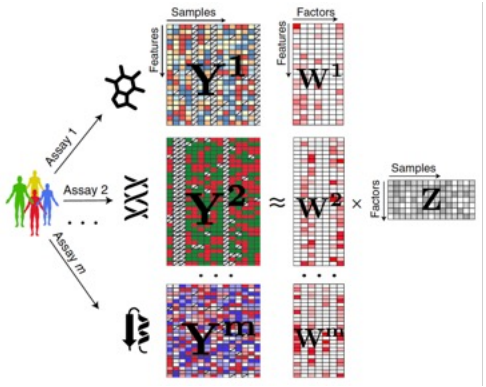


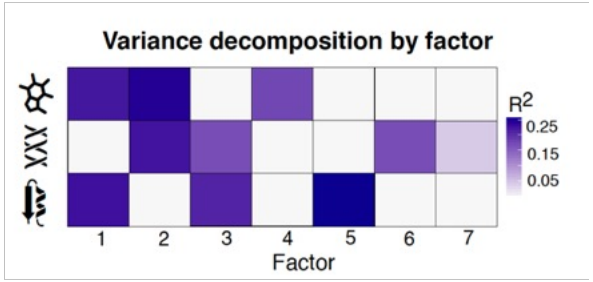
pyro


```python
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_featu
        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
```

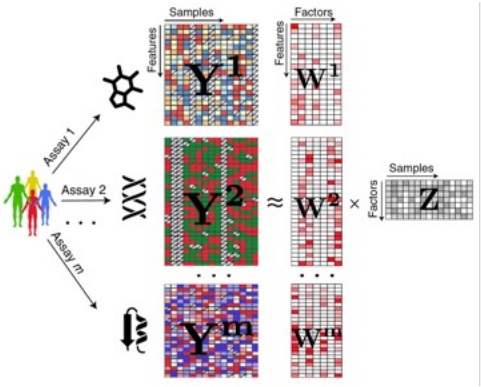
What can be done with such models?

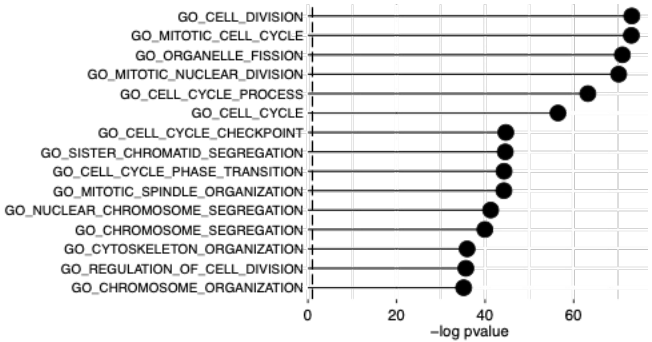
Variance decomposition



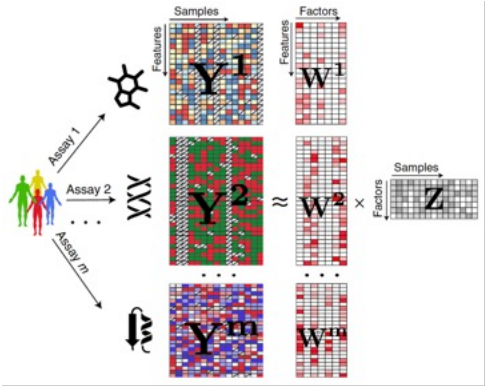


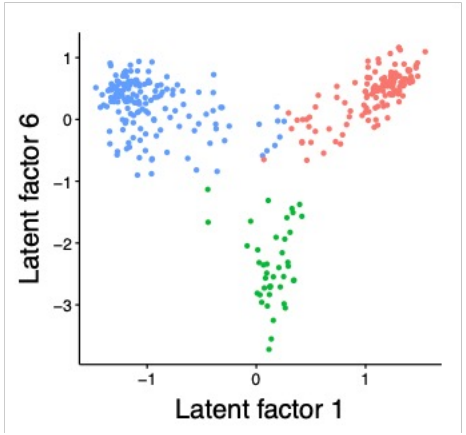
Inspection of weights



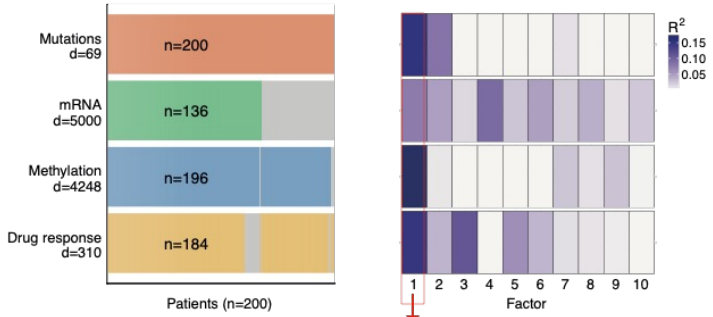


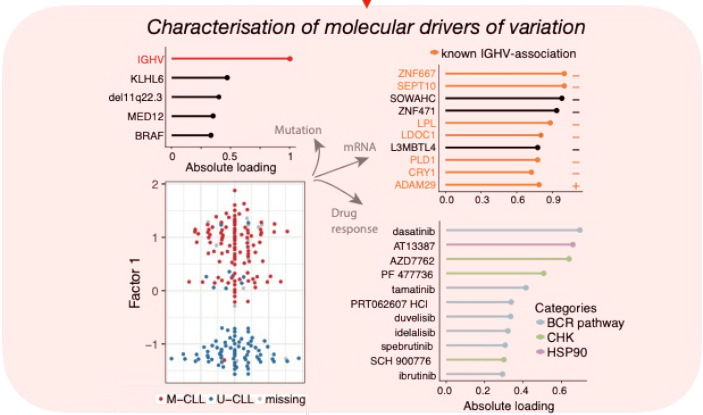
Visualisation of samples





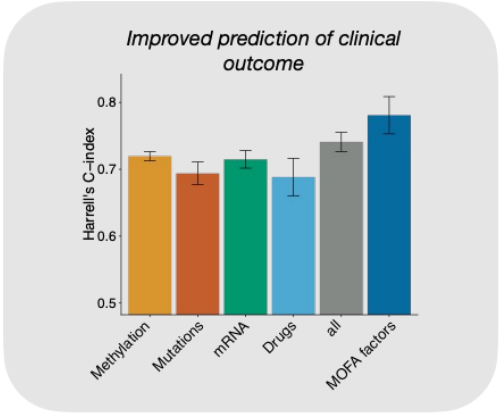
Example I: Precision medicine





200 chronic lymphocytic leukaemia samples (incompletely) characterised by

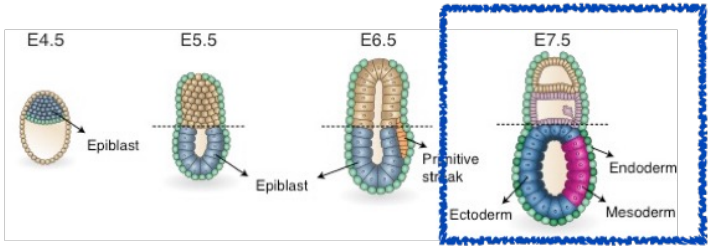
- genomic sequencing
- RNA-seq
- methylation arrays
- ex-vivo drug response assays



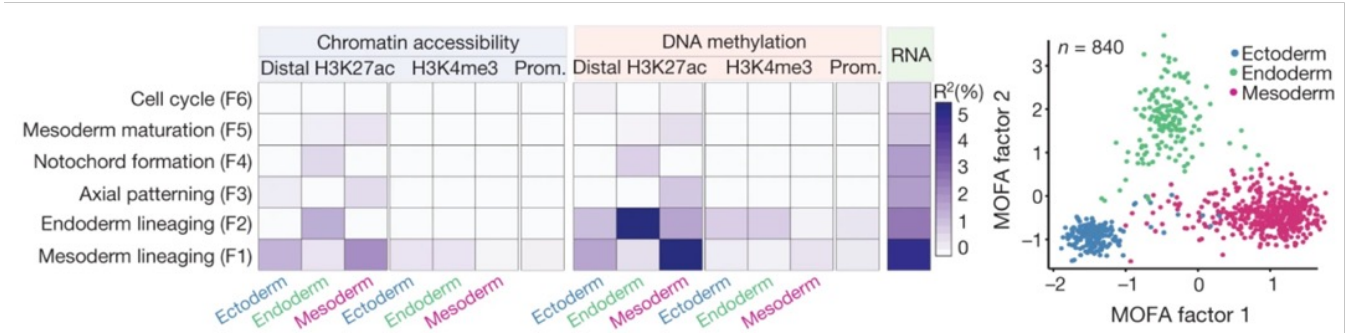
Argelaguet* & Velten* et al Mol Sys Bio. 2018

Example II: Single cell studies of development

scNMT-seq on cells from day E7.5 of mouse development provides information on DNA methylation, chromatin accessibility and transcription for each cell.



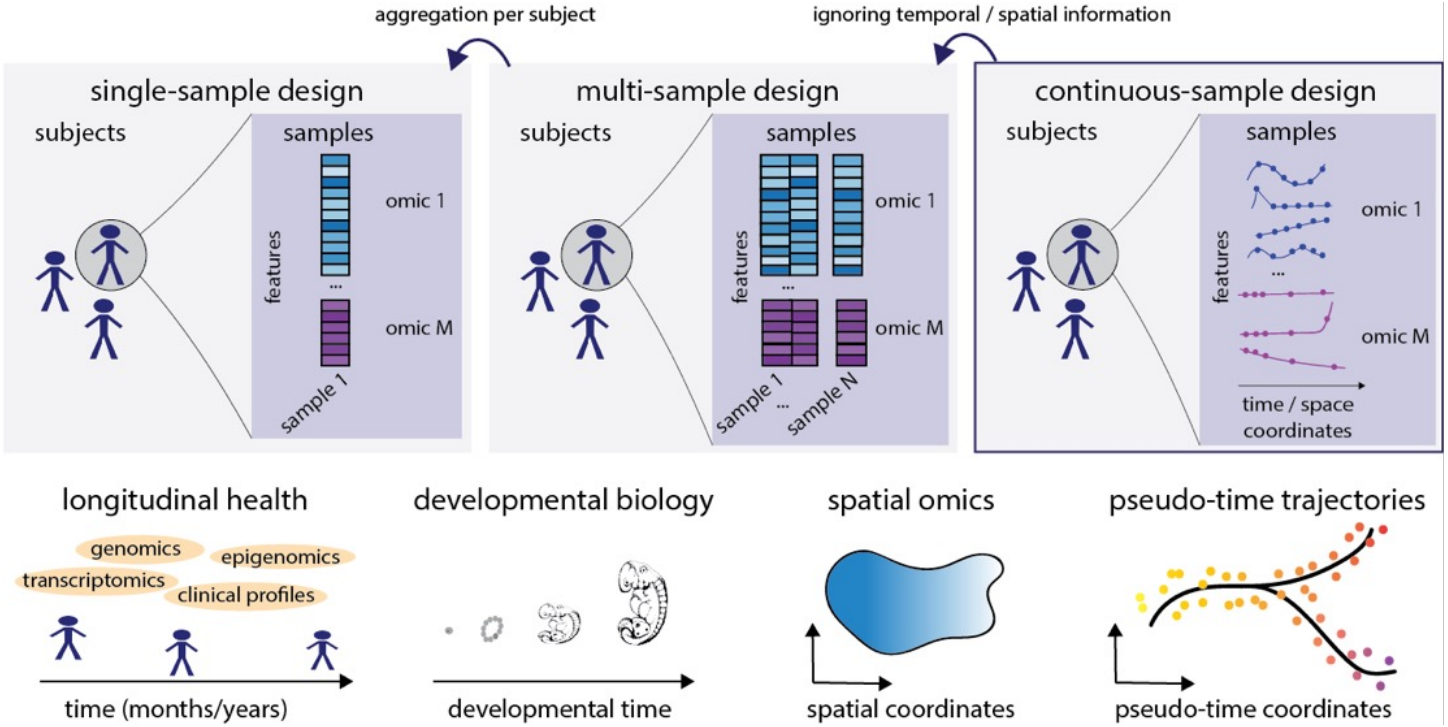
MOFA factors 1 and 2 capture the emerge of different germ layers and reveal coordinated transcriptional and epigenetic variation (latter mostly present at enhancers sites).



Argelaguet, Mohammed, Clark et al. Nature 2019

Multi-modal data with time and space resolution

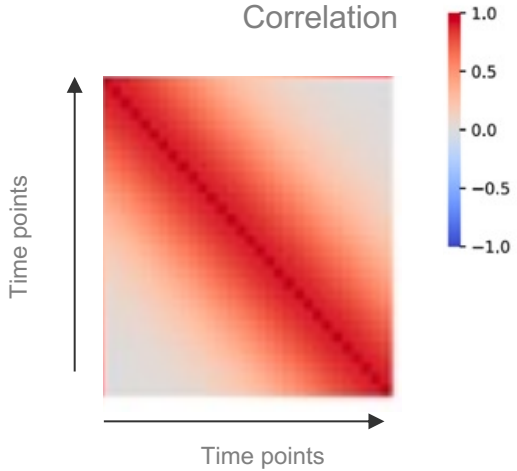
Temporal or spatial resolution

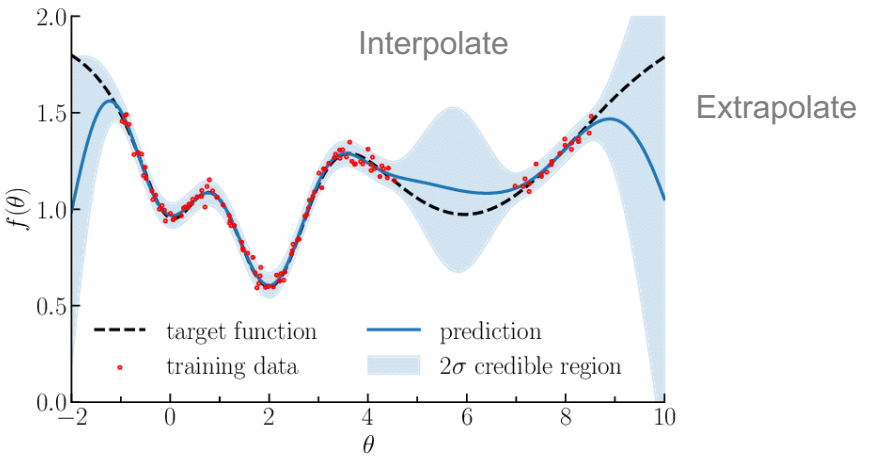


Velten & Stegle, Nath Methods 2023

What changes?

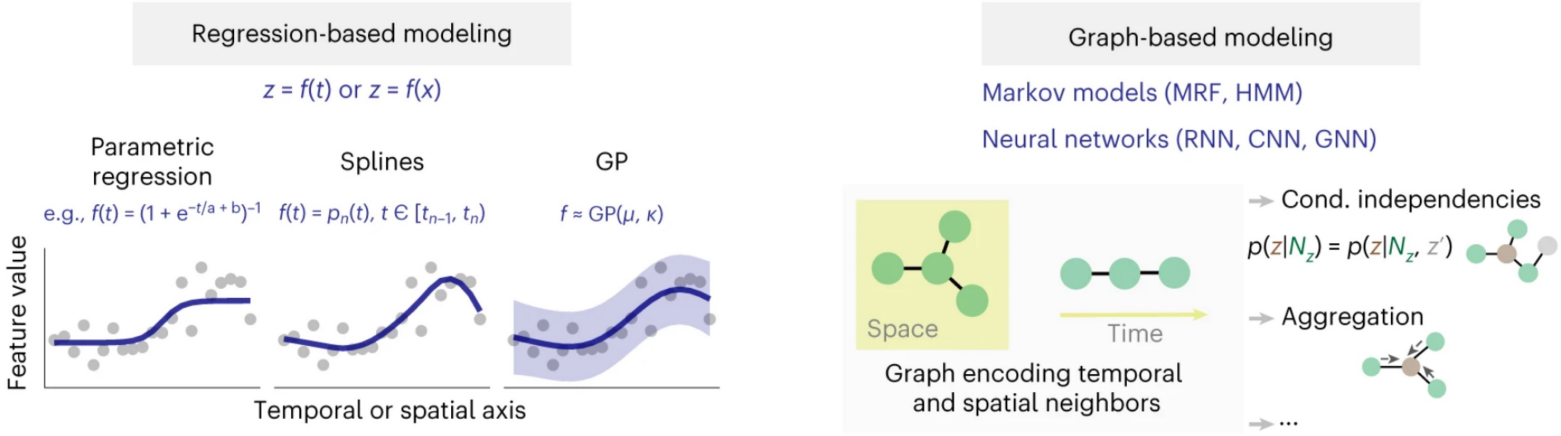
- Samples are non-i.i.d: nearby time points or spatial positions can be correlated
- Assumptions of smoothness along time or space can help to
  - mitigate noise
  - interpolate or extrapolate to unmeasured time points or positions





Florent Leclercq, 2018

Avenues for accounting for time & space



Velten & Stegle, Nath Methods 2023

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 μ and covariance function k is denoted by

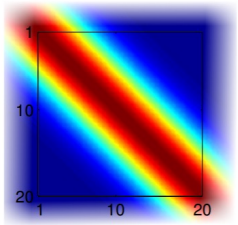
$$f \sim GP(\mu, k)$$

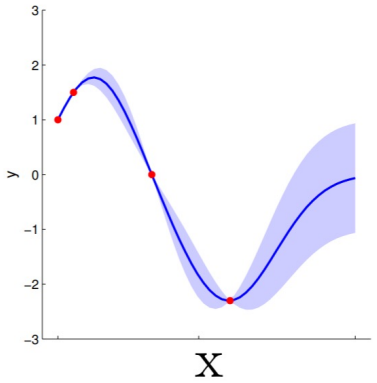
→ Probabilistic non-parametric method for learning functiona l relationships

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

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

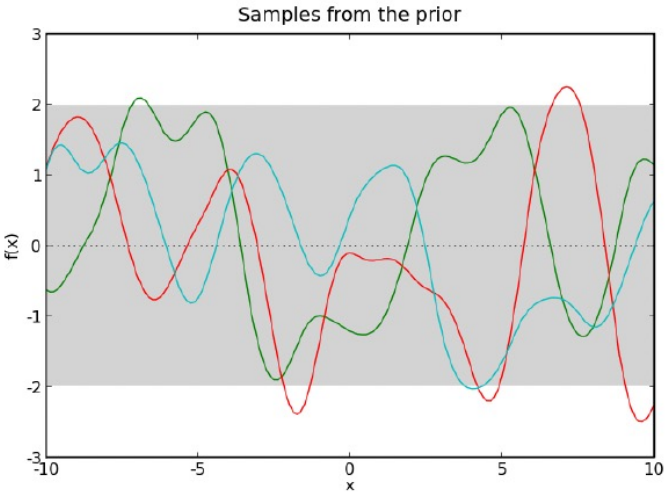
$\Sigma =$



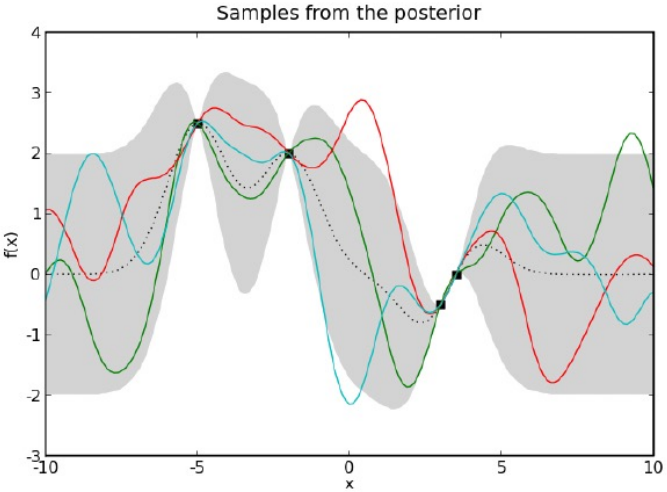


Gaussian processes

Before observing any data:

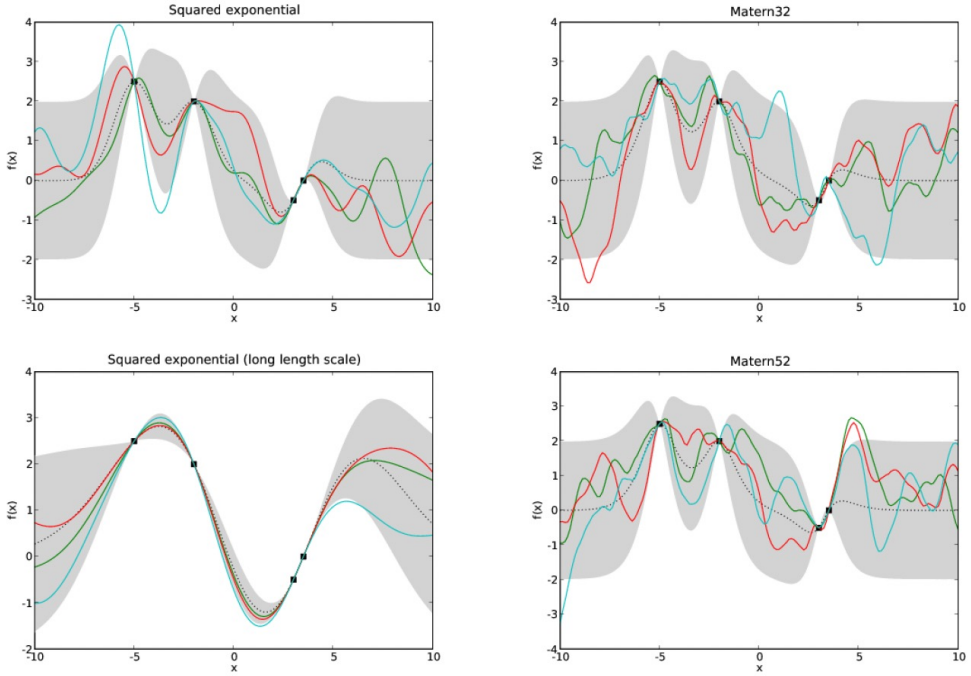


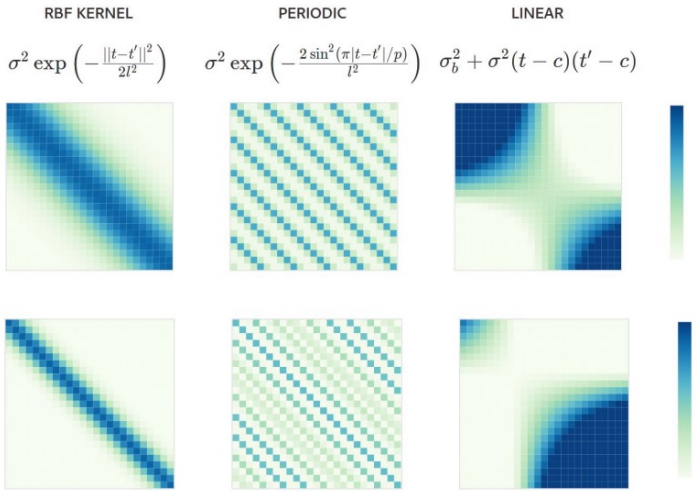
After measurements:



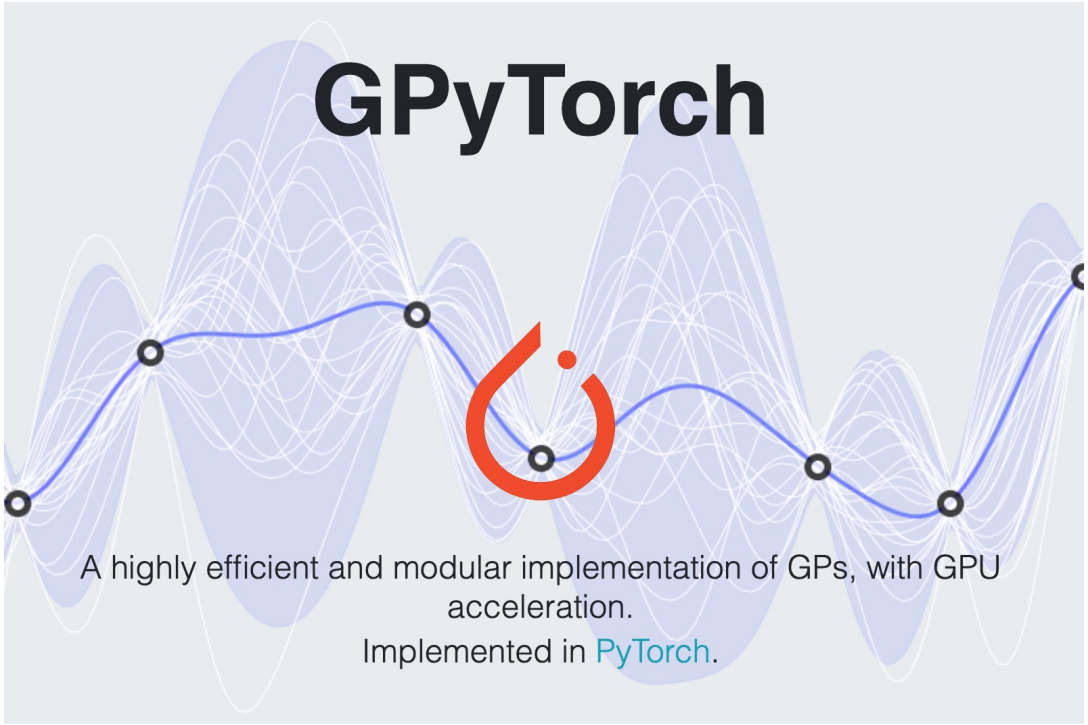
Gaussian processes

Choice of kernel / covariance function determines smoothness

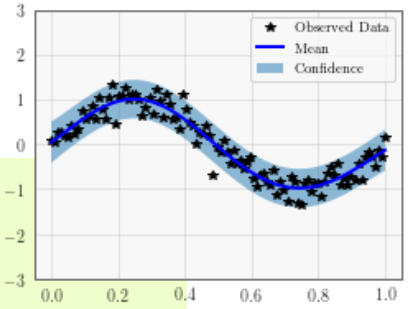




gpytorch


GPyTorch
A highly efficient and modular implementation of GPs, with GPU acceleration.
Implemented in PyTorch.

Example: Regression in gpytorch



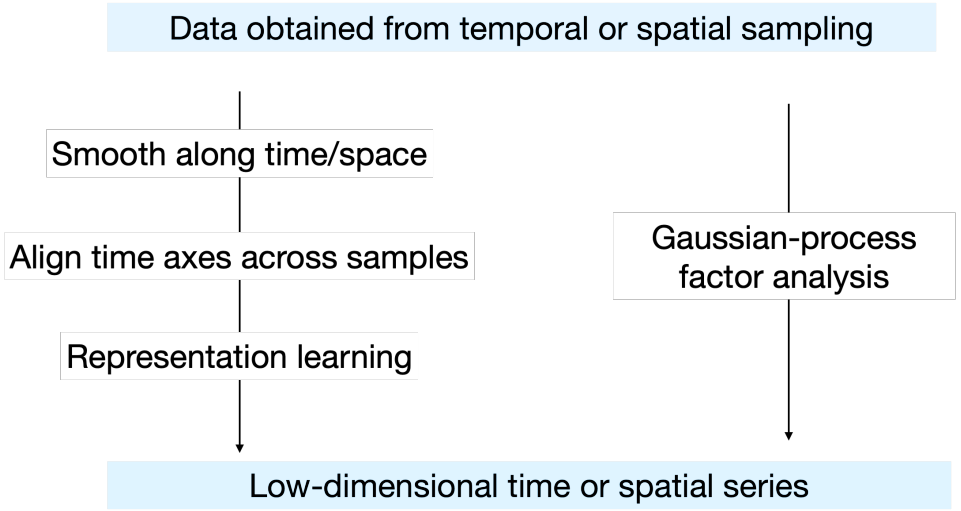

```python
class ExactGPModel(gpytorch.models.ExactGP):
    def __init__(self, train_x, train_y, likelihood):
        super(ExactGPModel, self).__init__(train_x, train_y, likelihood)
        self.mean_module = gpytorch.means.ConstantMean()
        self.covar_module = gpytorch.kernels.ScaleKernel(gpytorch.kernels.RBFKernel())

    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)
```

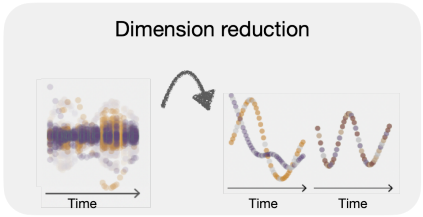
→ pytorch training loop

How to use Gaussian processes for dimenstion reduction and data integration?

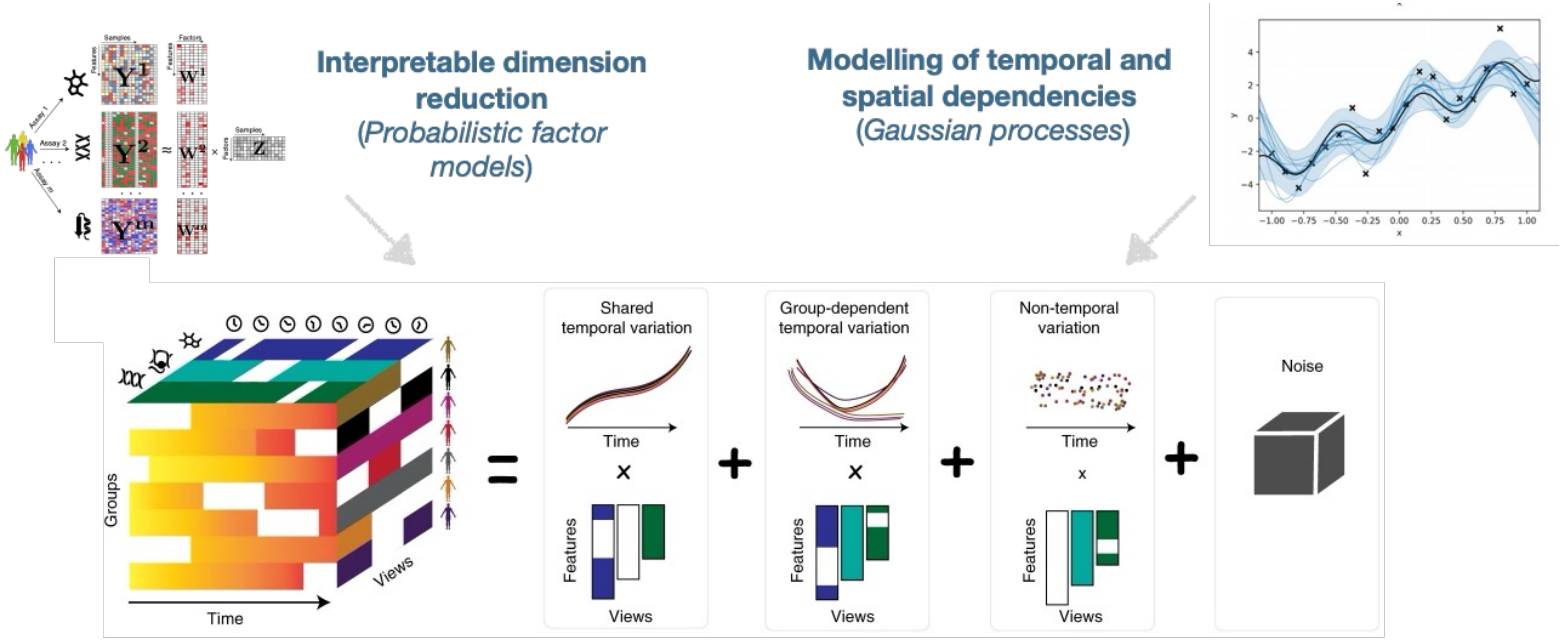
Gaussian process factor models (GPFA)

- Gaussian process priors in the latent space (factors)
- Joint dimension reduction and smoothing



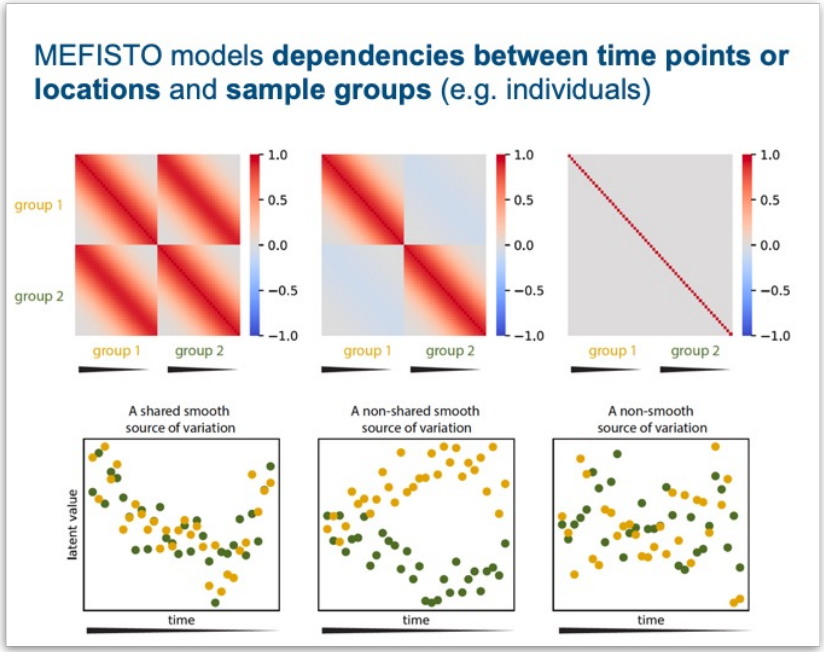


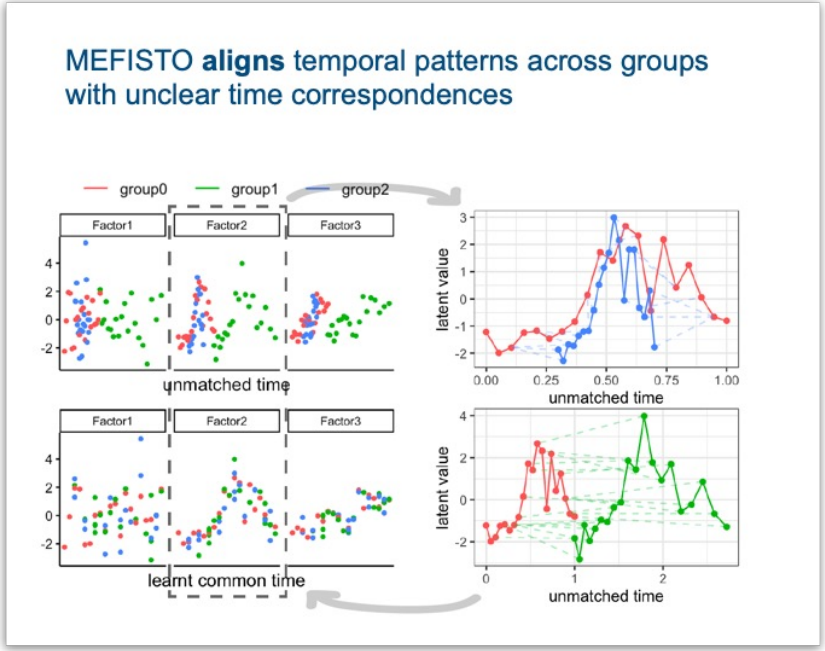
GPFA for multi-modal data: MEFISTO



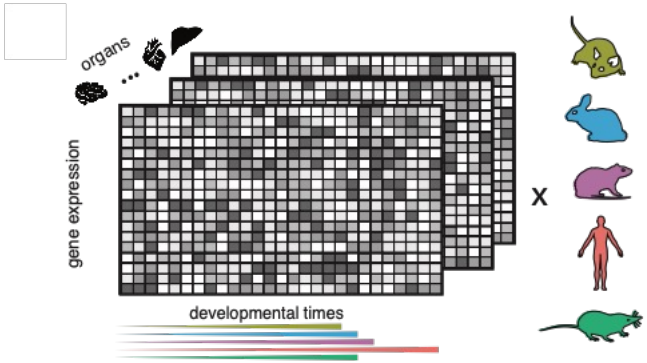
Integrated alignment of time points across individuals enables a meaningful comparisons of temporal patterns.

GPFA for multi-modal data: MEFISTO





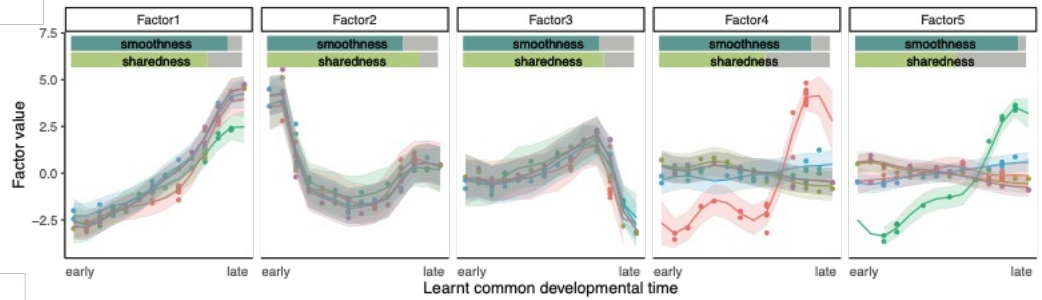
Example: Evo-devo data



Gene expression data for
- 5 species
- 5 organs
- 14-23 time points with unclear correspondences

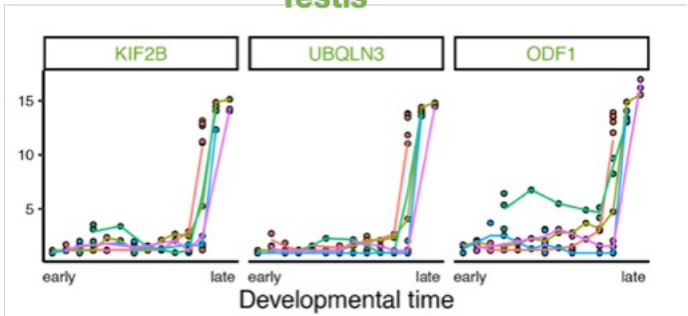
MEFISTO

Analysis of top weights in MEFISTO



Conserved developmental programs

Non-conserved developmental programs



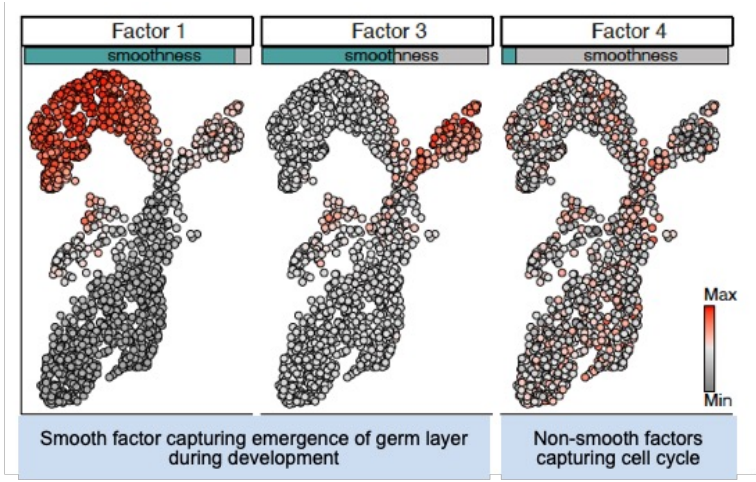
Velten et al, Nat Methods 2022

Example: Single cell data

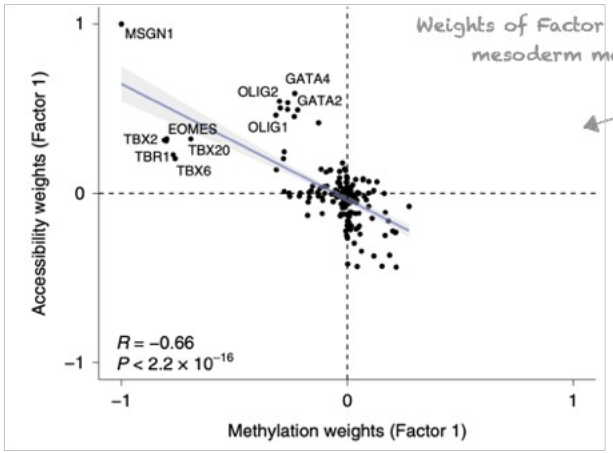
Data

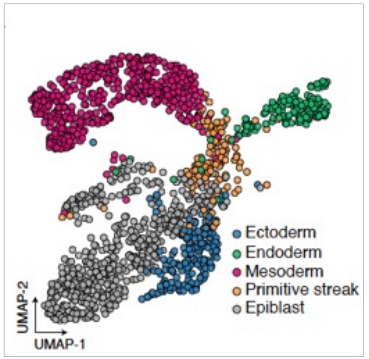
- 1,518 cells from multiple stages of early mouse development
- 3 omic layers (nucleosome, methylation, transcriptome)
- ~66 % of the cells only had transcriptome data

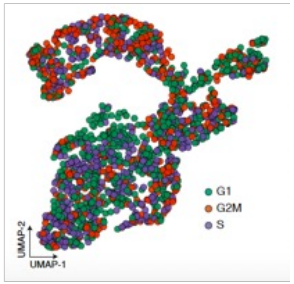
MEFISTO



Weights of Factor 1 pinpoint mesoderm makers

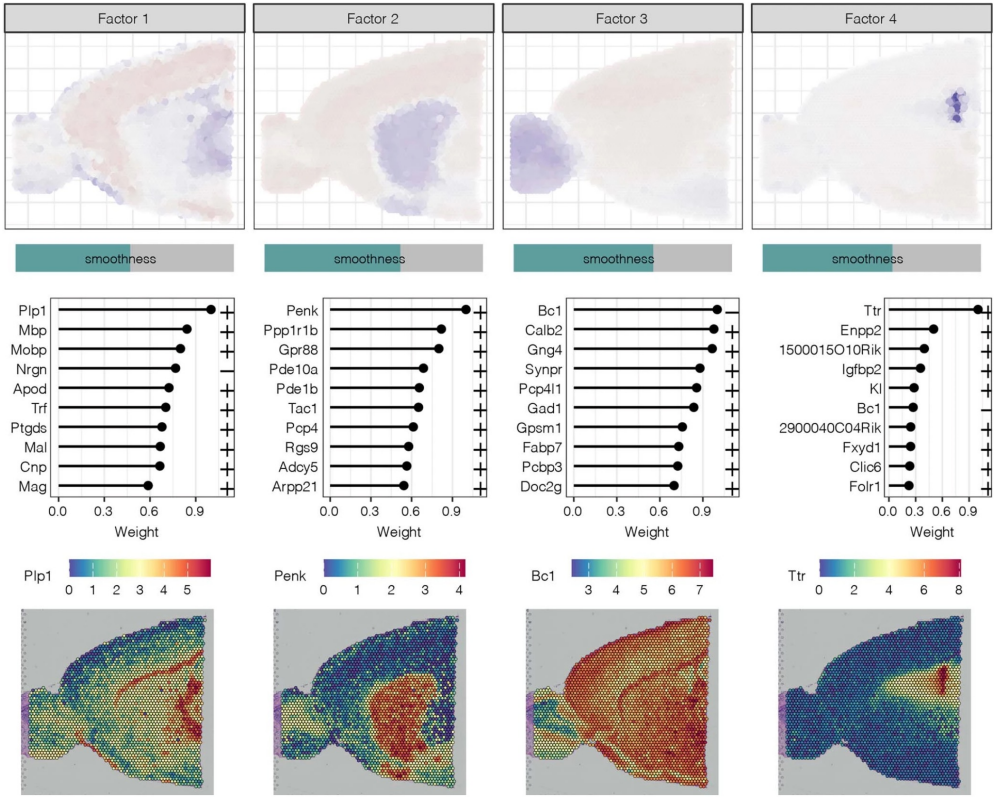






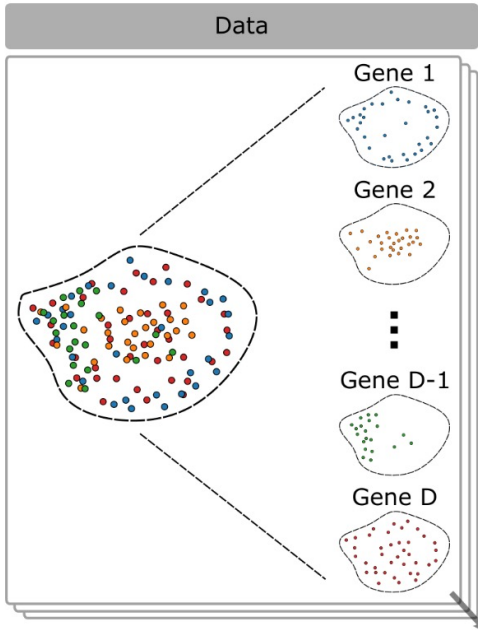
Velten et al, Nat Methods 2022

Example: Spatial data



Velten et al, Nat Methods 2022

Different likelihood models

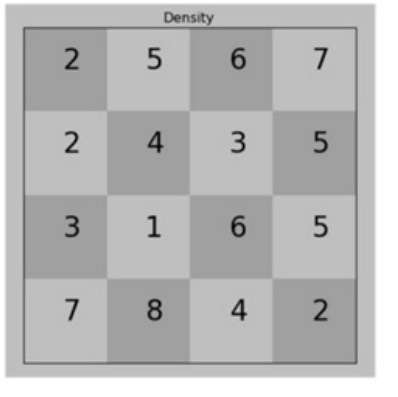


Cell with RNA molecule coordinates

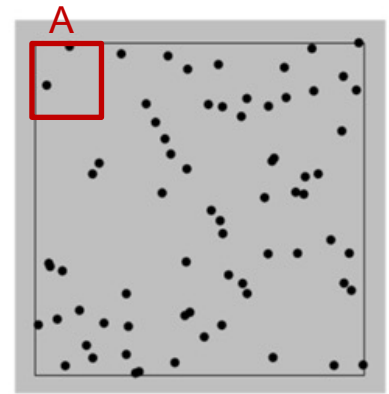
Walter et al, Bioinformatics 2023

Poisson point processes

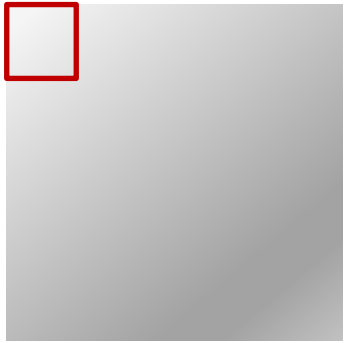
Spatial binning & Poisson likelihood



Poisson point process



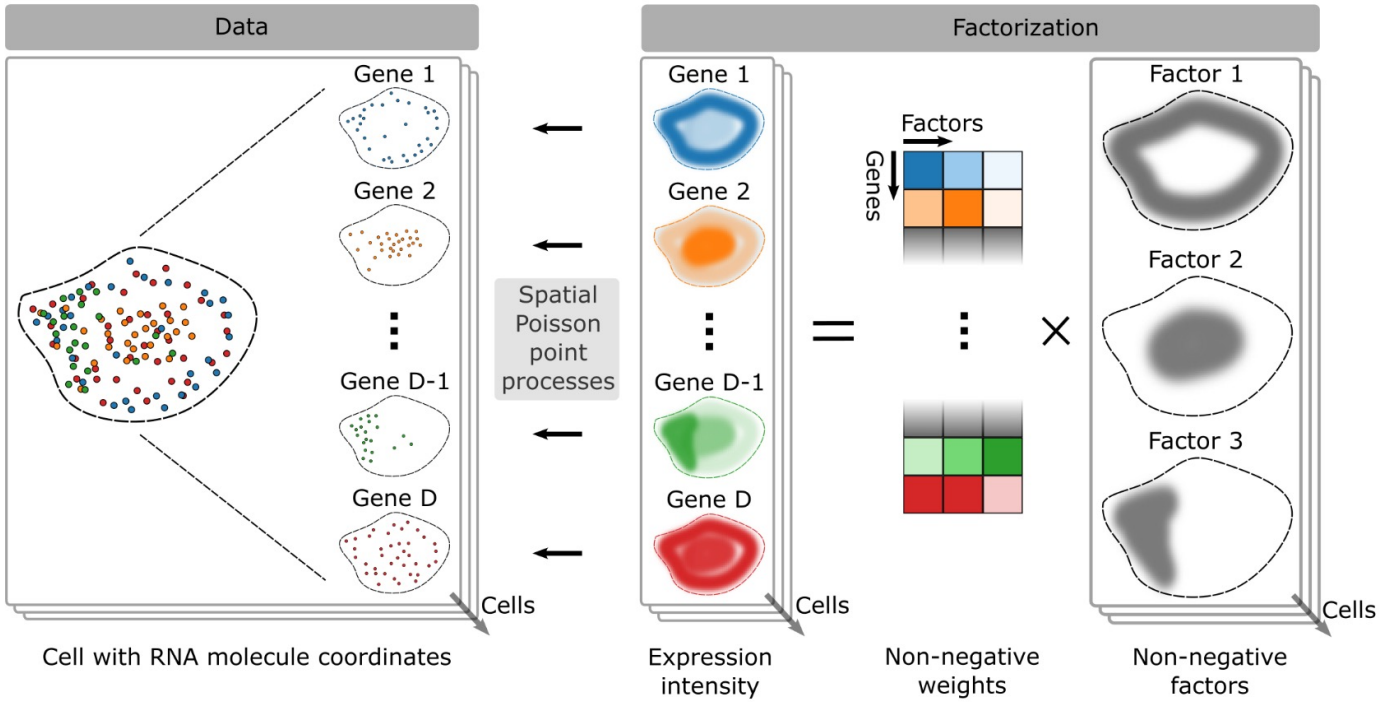
Intensity function λ



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

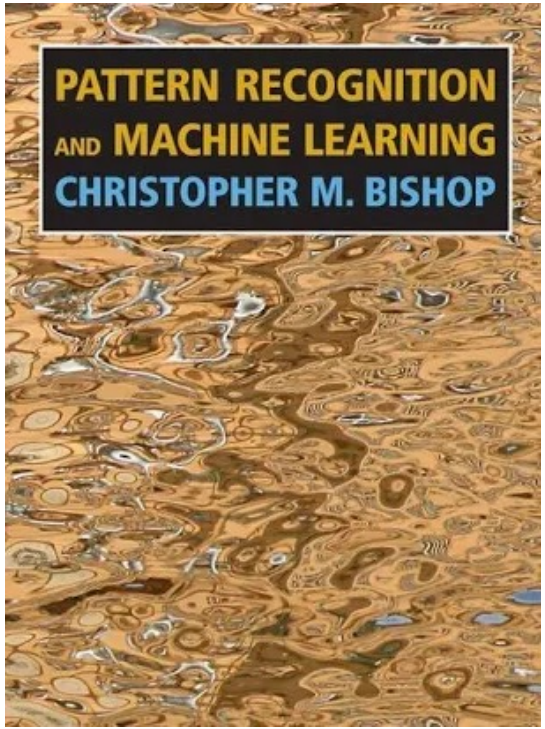
Adapted from here

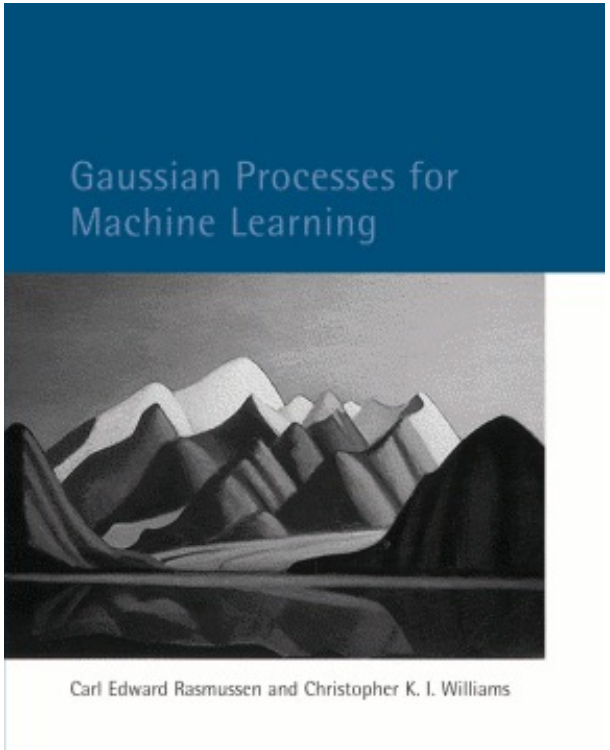
PP-GPFA for FISH data: FISHFactor



Walter et al, Bioinformatics 2023

Further reading





Questions?