UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

CENTRE FOR ORGANISMAL STUDIES

VELTEN GROUP

HGS MathComp

Interdisciplinary Center for Scientific Computing

European Laboratory for Learning and Intelligent Systems

PROBABILISTIC FACTOR ANALYSIS FOR PERTURBATION RESPONSES IN SINGLE-CELL DATA

Francisca Gaspar Vieira

"Autumn School for Single Cell-ers"

October 2025

ABOUT ME

Integrated Master's in Biological Engineering, IST Lisboa

Phage Biocontrol Research Project, KU Leuven

Master's in Computational Biology and Bioinformatics

Researcher in projects: MONET, AI4MUFF and VOOmics

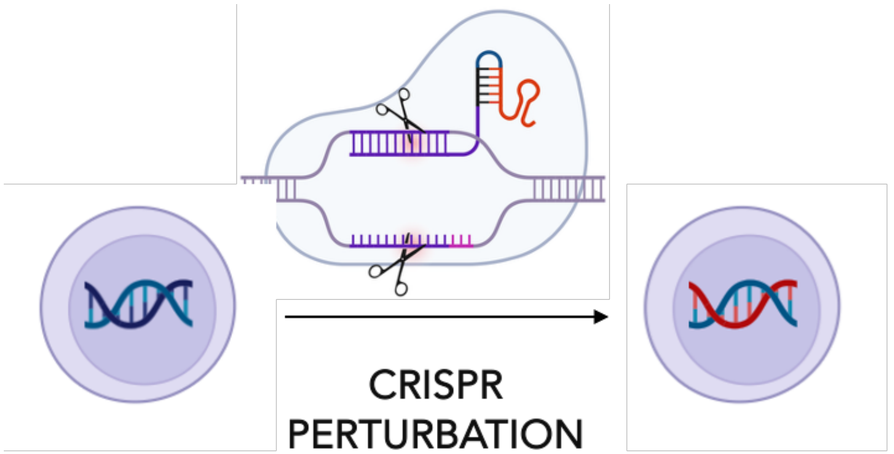
ELLIS PhD student, Heidelberg University + EMBL-EBI

Supervised by Dr. Britta Velten and Dr. Julio Saez-Rodriguez

1. Context
2. Approach using Factor Analysis
3. Integrating Uncertainty
4. Data Simulation
5. Application in Real Datasets
6. Conclusion and Future Perspectives

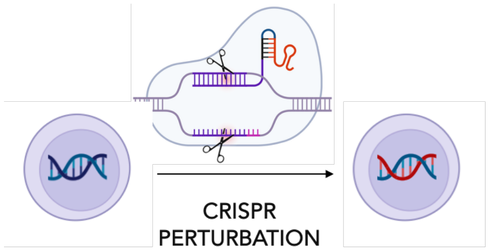
PERTURBATION RESPONSES IN SINGLE-CELL DATA

What does a certain gene actually do?

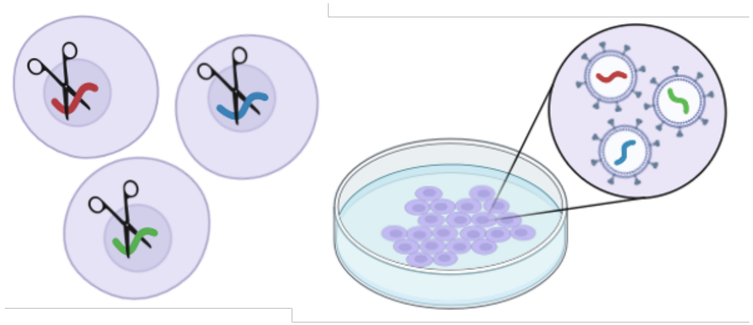


PERTURBATION RESPONSES IN SINGLE-CELL DATA

What does a certain gene actually do?



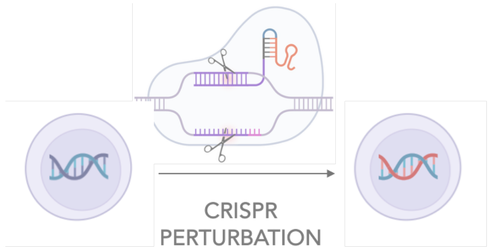
What about other genes?



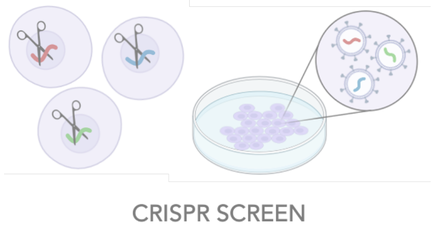
CRISPR SCREEN

PERTURBATION RESPONSES IN SINGLE-CELL DATA

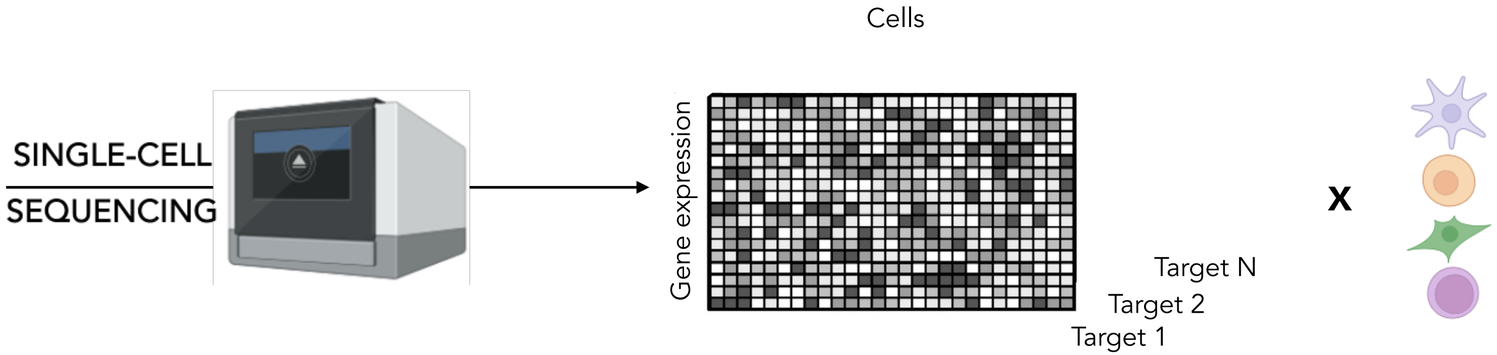
What does a certain gene actually do?



What about other genes?



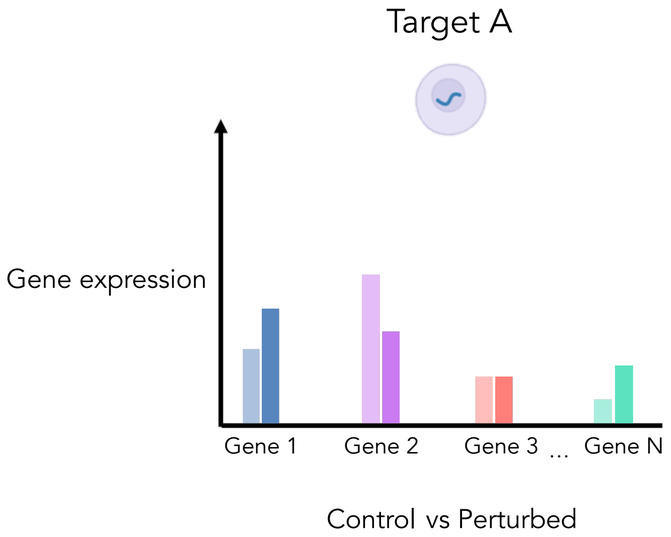
*How does knocking down **each gene***
*reshape the transcriptional landscape of **each cell**?*



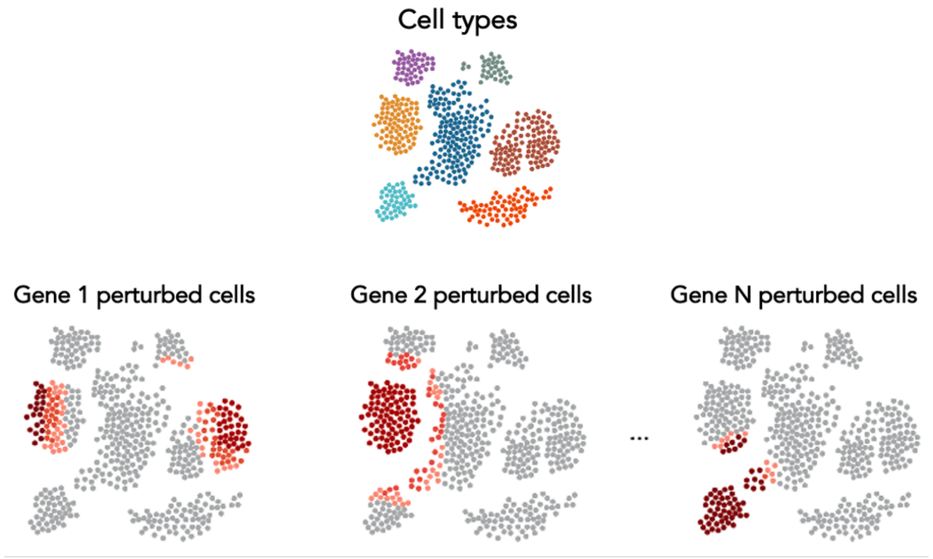
Do different cell types/lines/signaling contexts respond differently to the same perturbation - how can we detect these differences?

PERTURBATION RESPONSES IN SINGLE-CELL DATA

Gene effect estimation



Cell type-specific effects

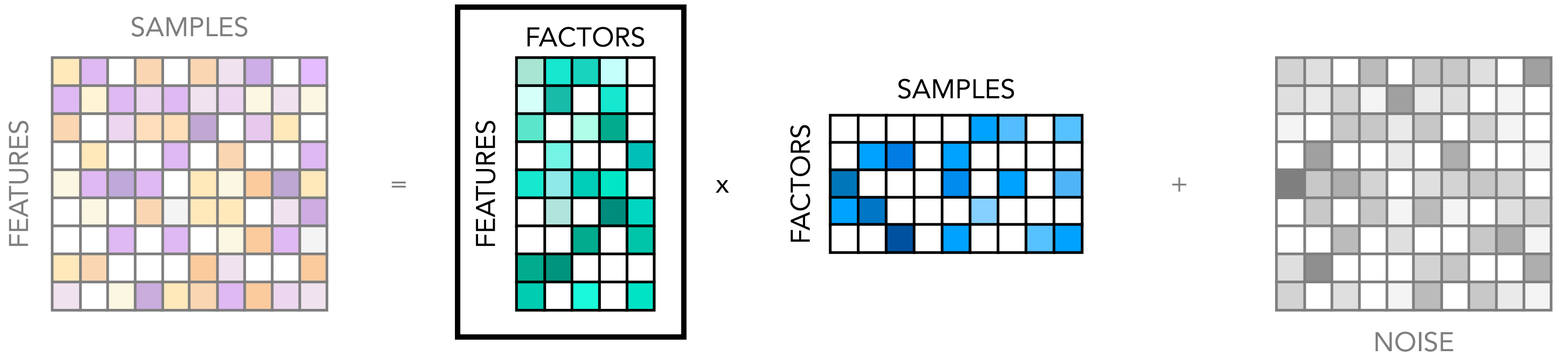


How to investigate shared effects between different biological contexts?
How to analyze all these results in a integrative way?

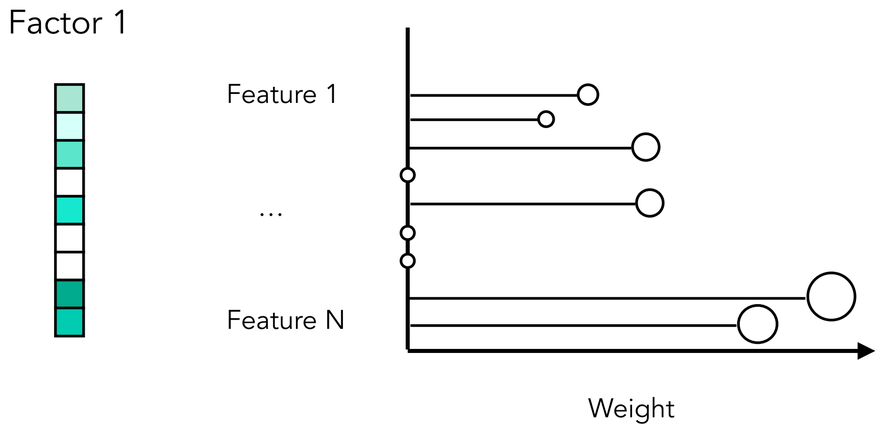
1. Context
2. Approach using Factor Analysis
3. Integrating Uncertainty
4. Data Simulation
5. Application in Real Datasets
6. Conclusion and Future Perspectives

APPROACH WITH FACTOR ANALYSIS

What is factor analysis?

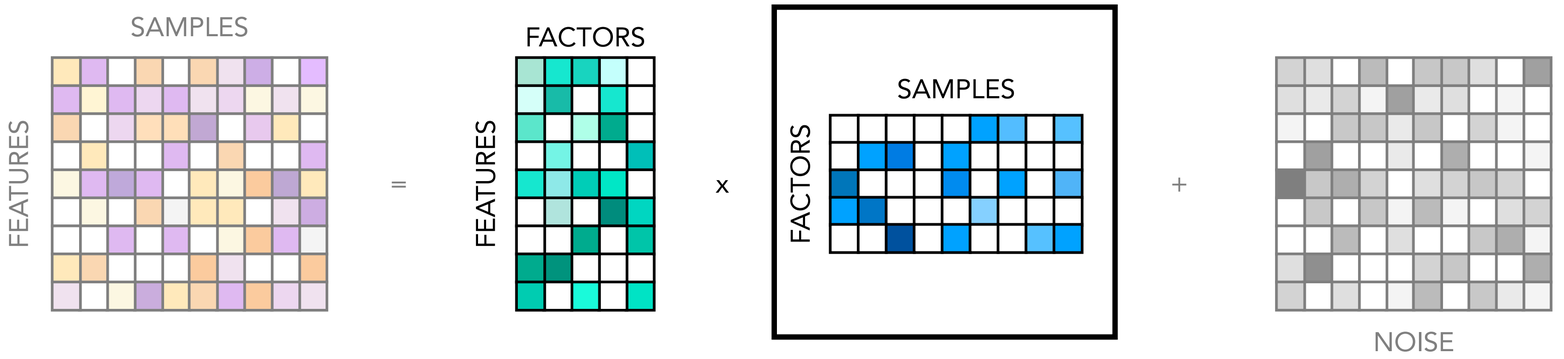


Why is it useful?

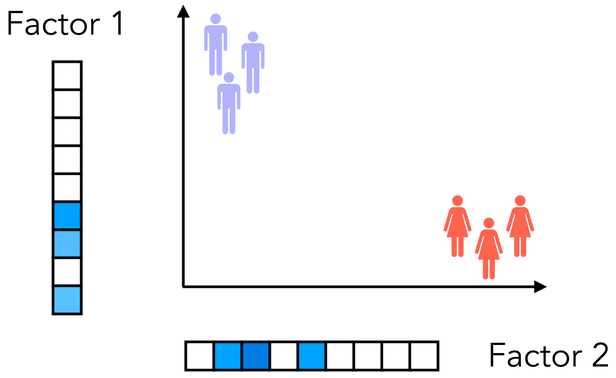


APPROACH WITH FACTOR ANALYSIS

What is factor analysis?

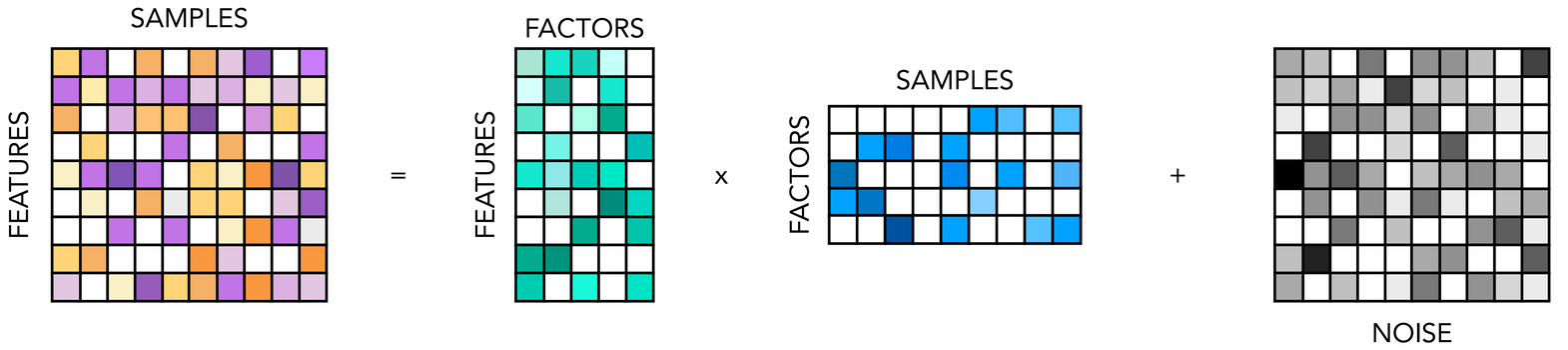


Why is it useful?



APPROACH WITH FACTOR ANALYSIS

What is behind probabilistic factor analysis?

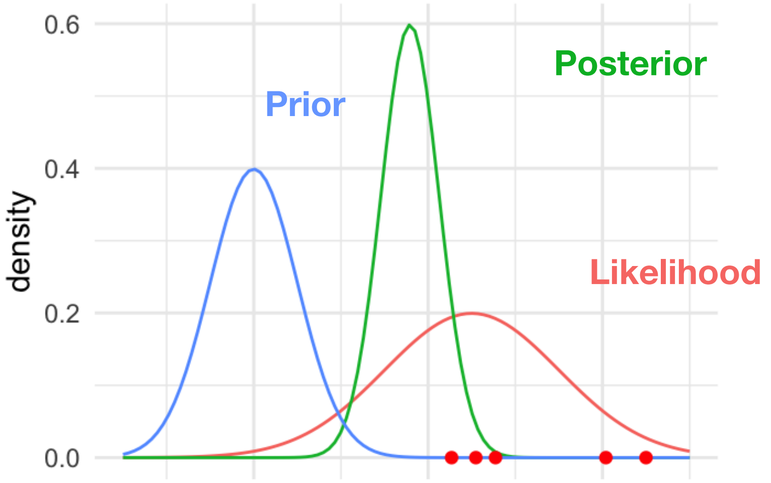


Probabilistic (Bayesian) models combine

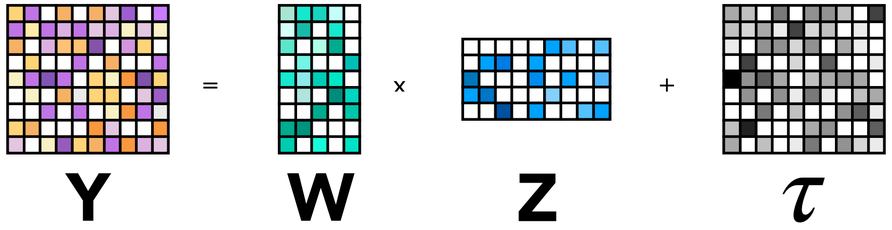
- a *likelihood function* (statistical model for the observed data)
- a *prior* (distribution of unobserved components)

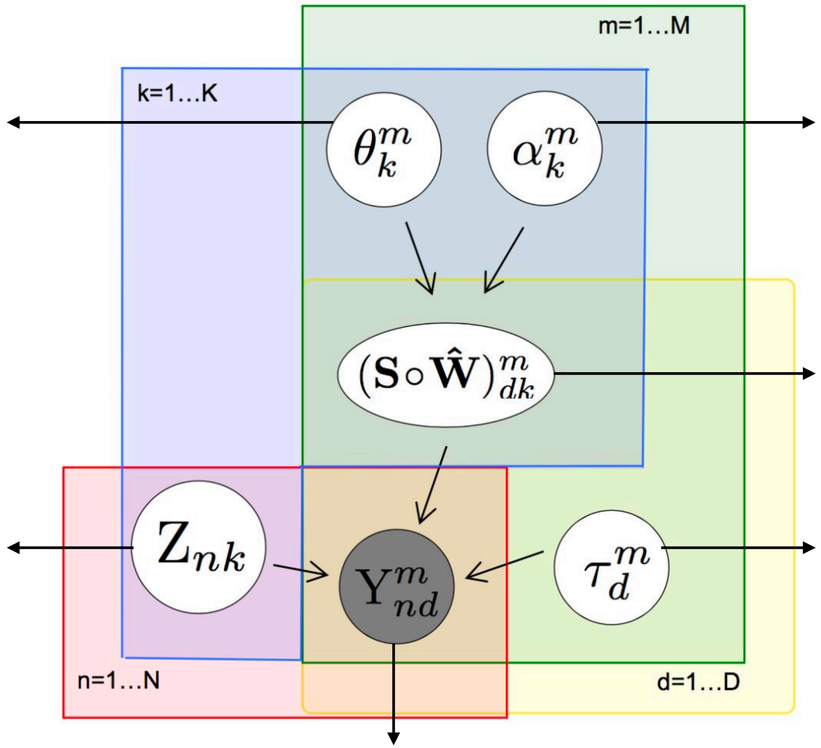
After seeing the data, we want to update our prior beliefs.
This updated belief is the **posterior.**

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



APPROACH WITH FACTOR ANALYSIS





Prior on the sparsity parameter

$$p(\theta_k^m) = \text{Beta}\left(\theta_k^m \mid a_0^\theta, b_0^\theta\right)$$

Prior on ARD precision parameter

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

Prior on the weights

$$p(\hat{w}_{d,k}^m, s_{d,k}^m) = \mathcal{N}(\hat{w}_{d,k}^m \mid 0, 1/\alpha_k^m)\,\text{Ber}(s_{d,k}^m \mid \theta_k^m)$$

Prior on latent factors

$$p(z_{nk}) = \mathcal{N}(z_{nk} \mid 0, 1)$$

Prior on noise

$$p(\tau_{md}) = \text{Gamma}(\tau_{md} \mid a_\tau^0, b_\tau^0)$$

$$a_\tau^0 = b_\tau^0 = 1 \times 10^{-14}$$

Likelihood

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

APPROACH WITH FACTOR ANALYSIS

Finding the posterior: An optimization problem

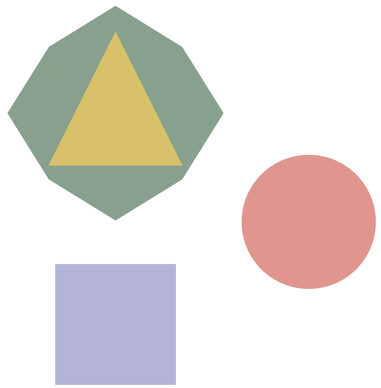
PROBLEM We want to know the hidden structure (**X**) behind our observed data (**Y**): meaning to calculate the posterior:

$$p(\mathbf{X} \mid \mathbf{Y}) = \frac{p(\mathbf{X}, \mathbf{Y})}{p(\mathbf{Y})} = \frac{p(\mathbf{X}, \mathbf{Y})}{\int p(\mathbf{X})\, p(\mathbf{Y} \mid \mathbf{X})\, d\mathbf{X}}$$ ⚠️

However, for most interesting models, the denominator becomes **intractable**.

CORE IDEA Approximate the true posterior with a simpler and tractable distribution q(**X**): turn inference into optimization.

Imagine trying to fit a complex shape with a simple one



APPROACH WITH FACTOR ANALYSIS

Finding the posterior: An optimization problem

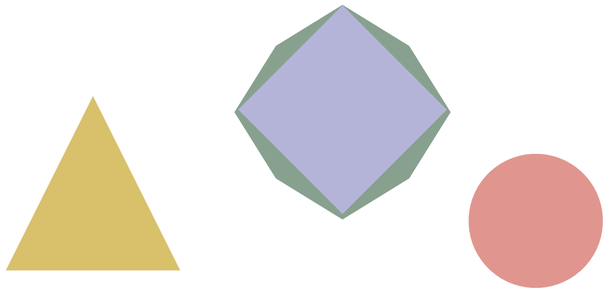
PROBLEM We want to know the hidden structure (**X**) behind our observed data (**Y**): meaning to calculate the posterior:

$$p(\mathbf{X} \mid \mathbf{Y}) = \frac{p(\mathbf{X}, \mathbf{Y})}{p(\mathbf{Y})} = \frac{p(\mathbf{X}, \mathbf{Y})}{\int p(\mathbf{X})\, p(\mathbf{Y} \mid \mathbf{X})\, d\mathbf{X}}$$ ⚠️

However, for most interesting models, the denominator becomes **intractable**.

CORE IDEA Approximate the true posterior with a simpler and tractable distribution q(**X**): turn inference into optimization.

Imagine trying to fit a complex shape with a simple one



APPROACH WITH FACTOR ANALYSIS

Finding the posterior: An optimization problem

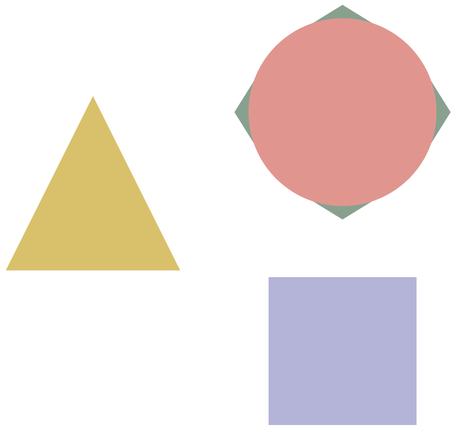
PROBLEM We want to know the hidden structure (**X**) behind our observed data (**Y**): meaning to calculate the posterior:

$$p(\mathbf{X} \mid \mathbf{Y}) = \frac{p(\mathbf{X}, \mathbf{Y})}{p(\mathbf{Y})} = \frac{p(\mathbf{X}, \mathbf{Y})}{\int p(\mathbf{X})\, p(\mathbf{Y} \mid \mathbf{X})\, d\mathbf{X}}$$ ⚠️

However, for most interesting models, the denominator becomes **intractable**.

CORE IDEA Approximate the true posterior with a simpler and tractable distribution q(**X**): turn inference into optimization.

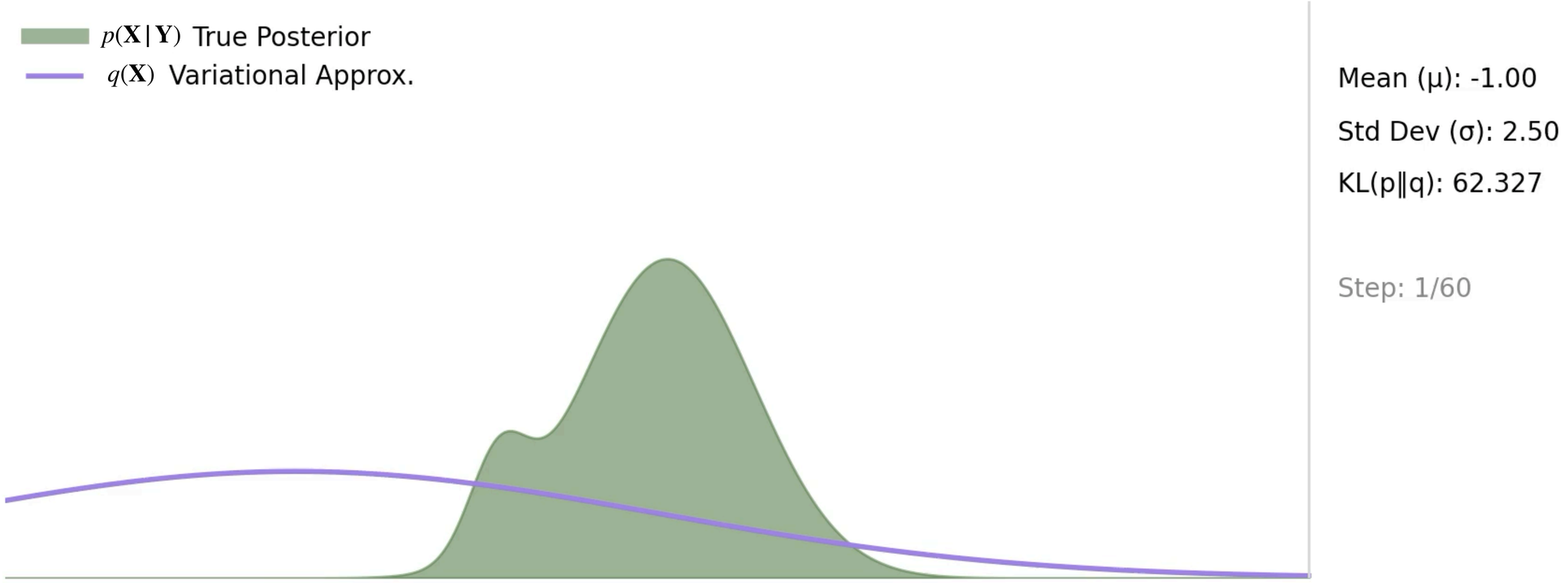
Imagine trying to fit a complex shape with a simple one



APPROACH WITH FACTOR ANALYSIS

Finding the posterior: An optimization problem

1. Pick a flexible, tractable family of distributions (e.g., Gaussians)
2. Define objective/ loss function for how well $q(\mathbf{X})$ approximates $p(\mathbf{X}|\mathbf{Y})$
3. Optimize: Adjust parameters of $q(\mathbf{X})$ to minimize the mismatch

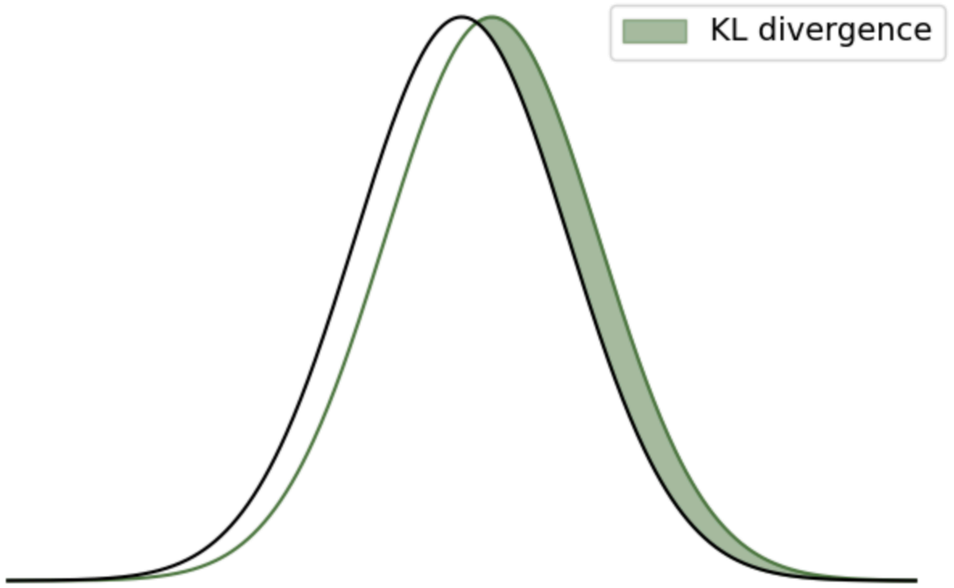


APPROACH WITH FACTOR ANALYSIS

Kullback–Leibler divergence

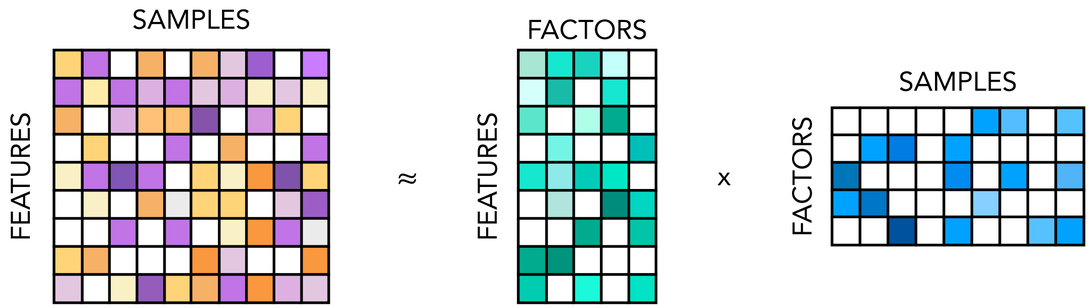
The Kullback-Leibler (KL) divergence is the standard distance measure choice when working with probability distributions:

$$\mathrm{KL}(q(\mathbf{X}) \,\|\, p(\mathbf{X} \,|\, \mathbf{Y})) = \int q(\mathbf{X}) \log \frac{q(\mathbf{X})}{p(\mathbf{X} \,|\, \mathbf{Y})} \, d\mathbf{z}$$



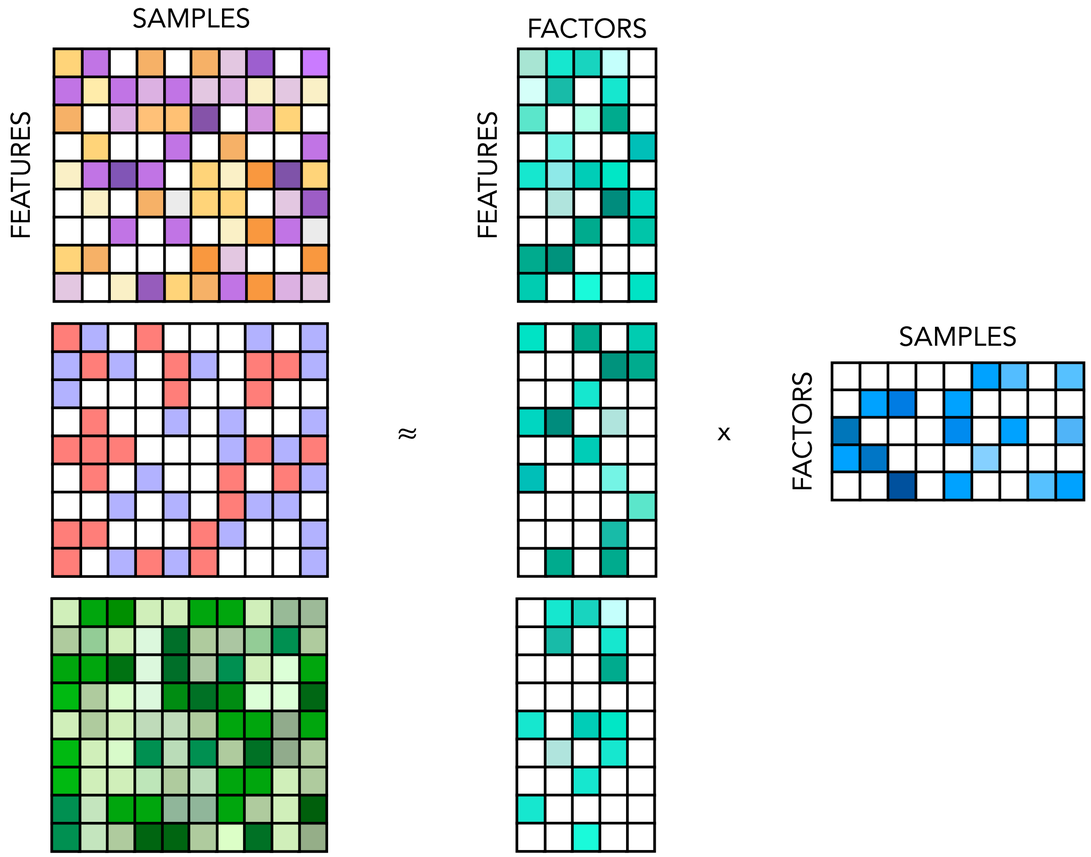
APPROACH WITH FACTOR ANALYSIS

What is factor analysis?



APPROACH WITH FACTOR ANALYSIS

*What is **multi view-** factor analysis?*



APPROACH WITH FACTOR ANALYSIS

Method

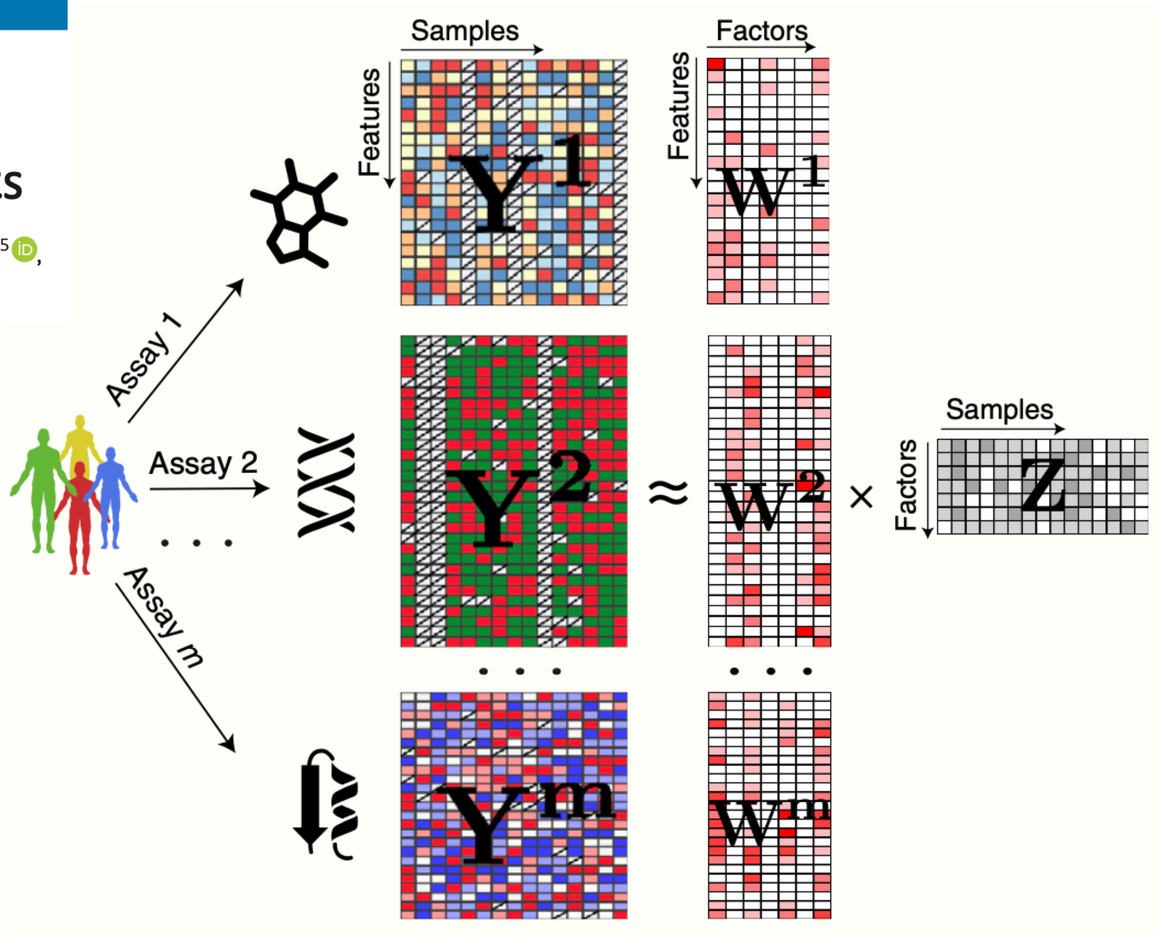
TRANSPARENT PROCESS OPEN ACCESS

molecular systems biology

Multi-Omics Factor Analysis—a framework for unsupervised integration of multi-omics data sets

Ricard Argelaguet[1,†], Britta Velten[2,†], Damien Arnol[1], Sascha Dietrich[3], Thorsten Zenz[3,4,5], John C Marioni[1,6,7], Florian Buettner[1,8,*], Wolfgang Huber[2,**] & Oliver Stegle[1,2,***]

- MOFA infers an interpretable low-dimensional data representation in terms of (hidden) factors that capture major sources of variation across data modalities
- The inferred factor loadings can be sparse, thereby facilitating the linkage between the factors and the most relevant molecular features.
- **MOFA disentangles to what extent each factor is unique to a single data modality or is manifested in multiple modalities**



APPROACH WITH FACTOR ANALYSIS

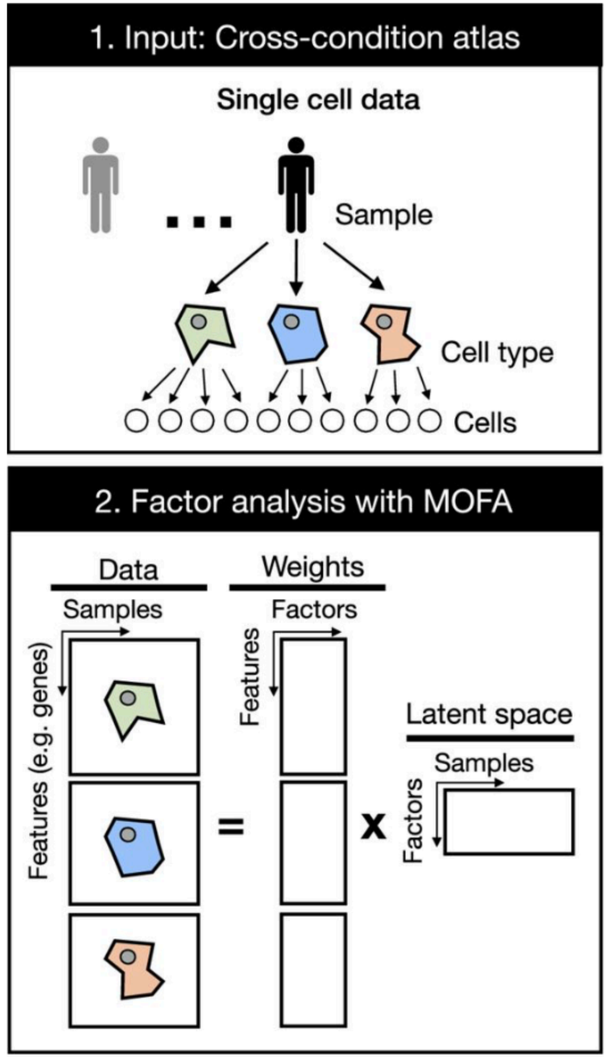
eLife

RESEARCH ARTICLE

Multicellular factor analysis of single-cell data for a tissue-centric understanding of disease

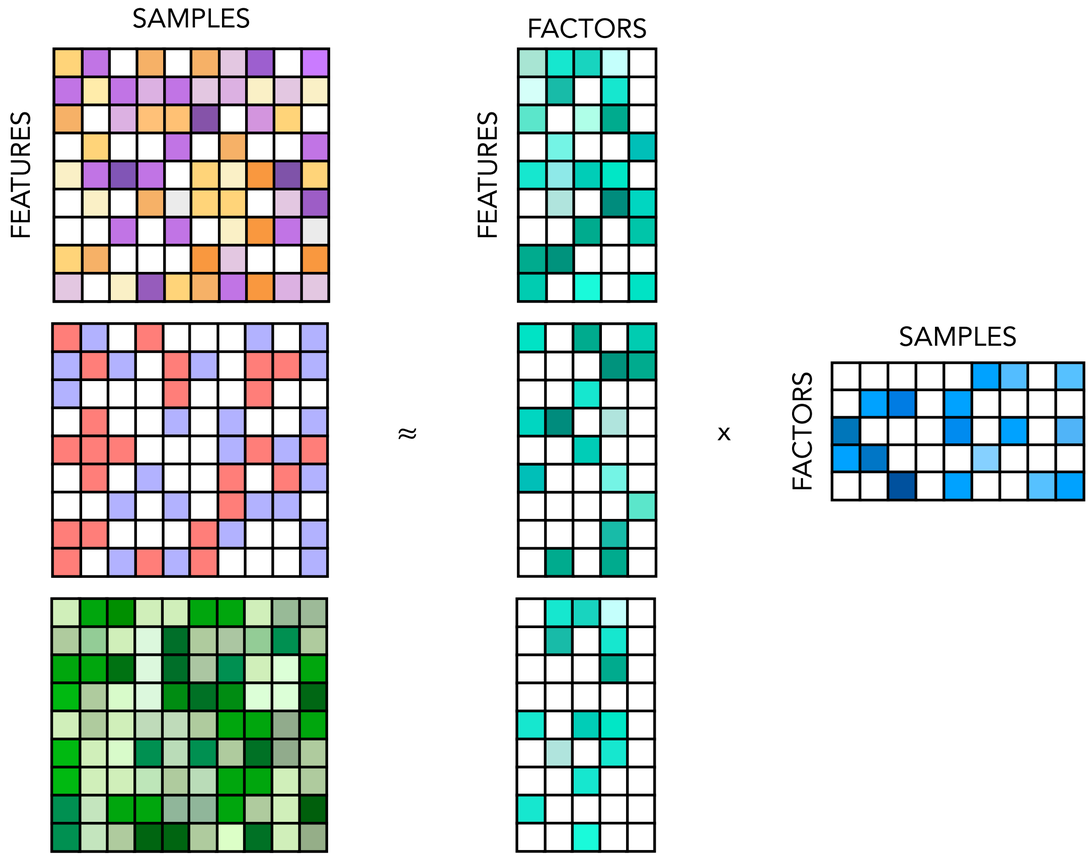
Ricardo Omar Ramirez Flores[1]*, Jan David Lanzer[1], Daniel Dimitrov[1], Britta Velten[2], Julio Saez-Rodriguez[1]*

- Repurposes multi-omics factor analysis (MOFA) to simultaneously **decompose the variability of multiple cell types** and create a latent space that recovers multicellular transcriptional programs.
- The variables that form this latent space can be **interpreted as coordinated transcriptional changes** occurring in multiple cells - multicellular programs, providing a tissue-centric understanding of the analyzed sample.



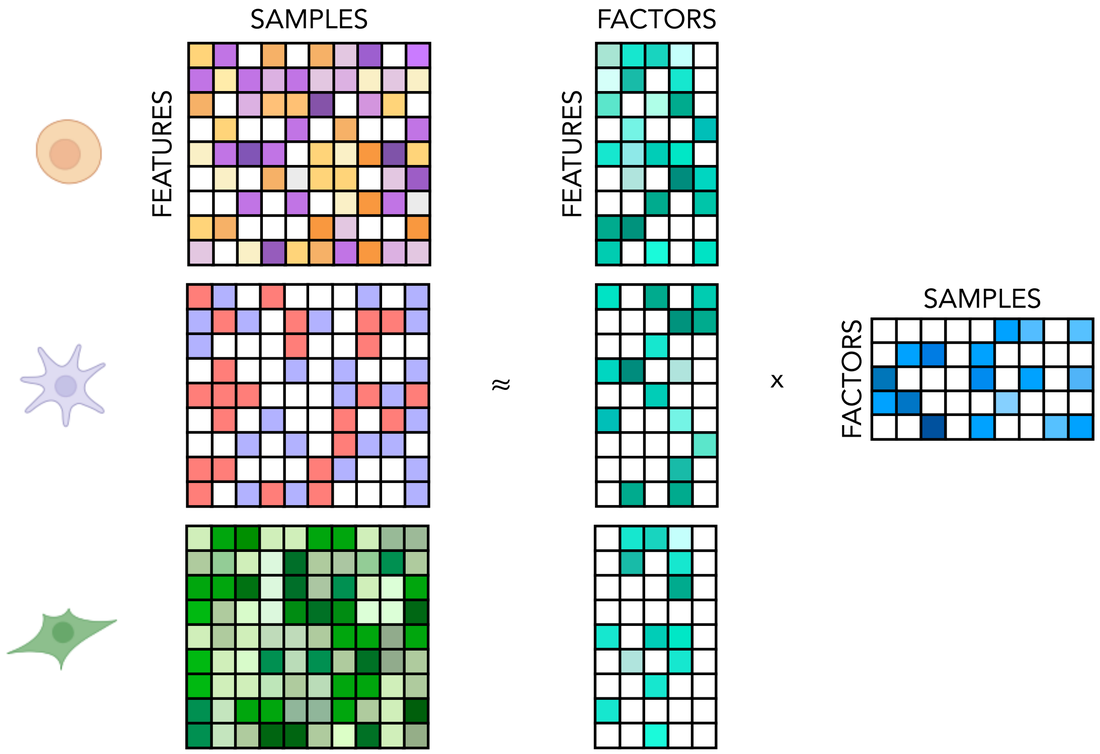
APPROACH WITH FACTOR ANALYSIS

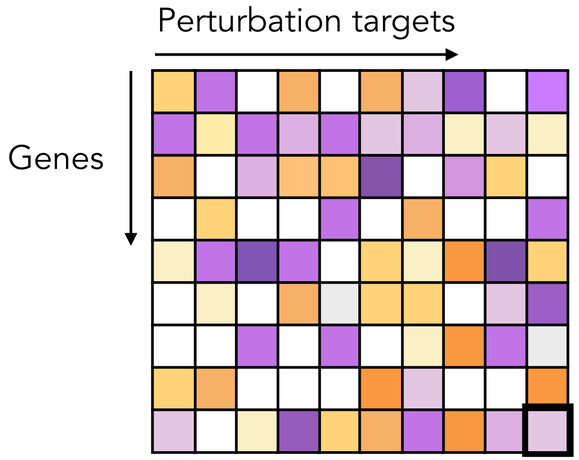
How to use/repurpose it for perturbation responses?



APPROACH WITH FACTOR ANALYSIS

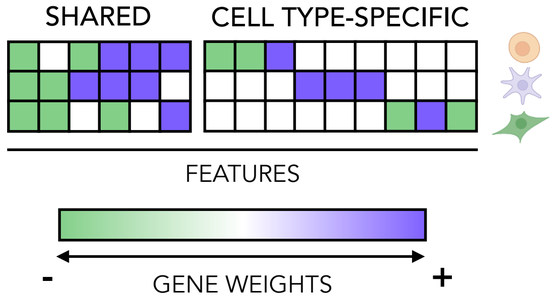
How to use/repurpose it for perturbation responses?





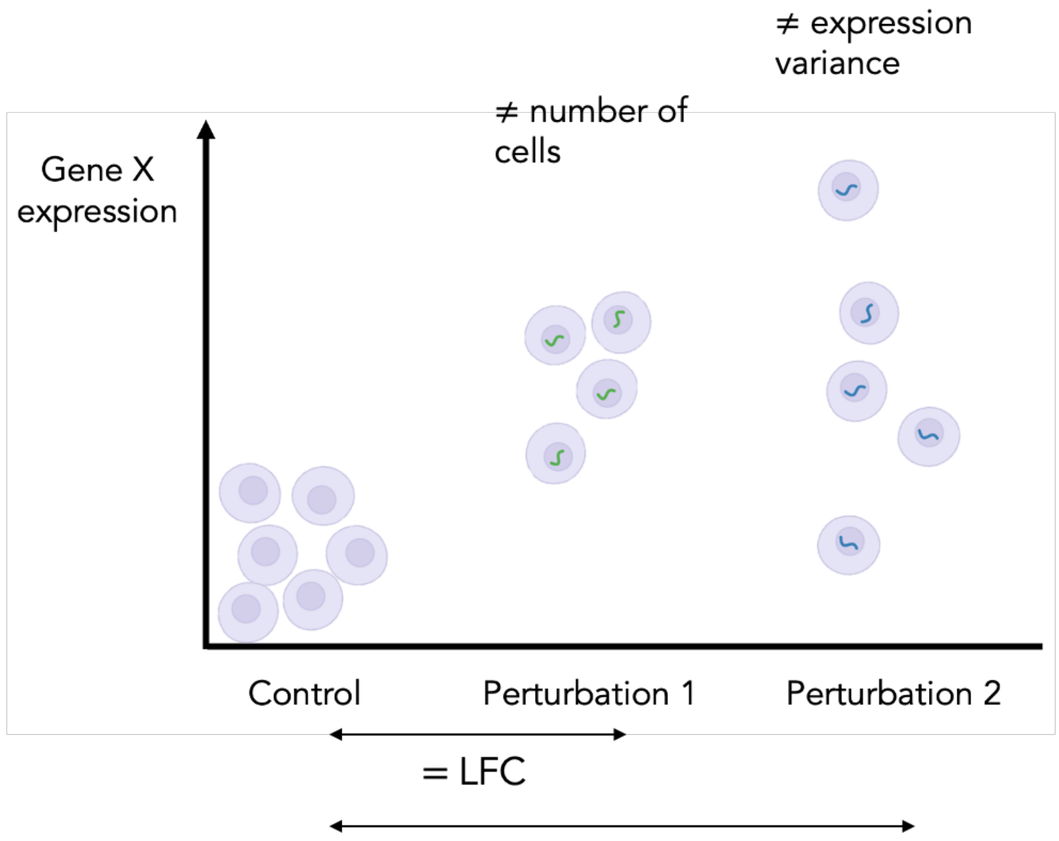
Use Log Fold Changes (LFC) for each gene under each perturbation (control vs perturbed)

Use multi-view and multi-group to handle multiple cell types and different biological contexts



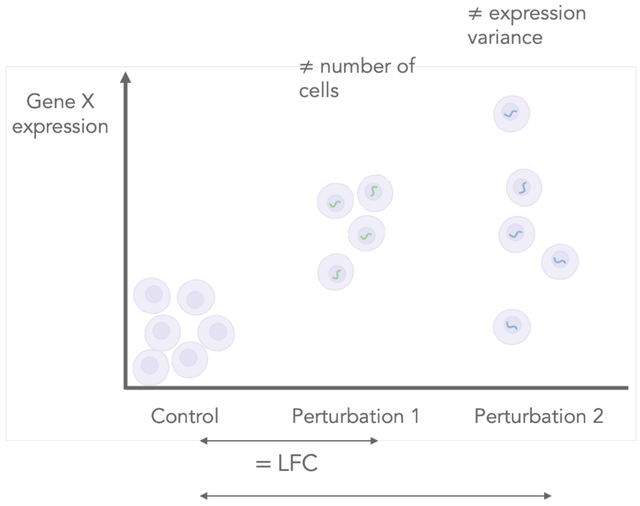
1. Context
2. Approach using Factor Analysis
3. Integrating Uncertainty
4. Data Simulation
5. Application in Real Datasets
6. Conclusion and Future Perspectives

INTEGRATING UNCERTAINTY



How to incorporate LFC uncertainty in the model?

INTEGRATING UNCERTAINTY



How to incorporate LFC uncertainty in the model?

DATA AND ASSUMPTION

For each gene *g* and perturbation *p:* $\hat{b}_{gp} \sim \mathbb{F}_{gp}, \quad \mathbb{E}[\hat{b}_{gp}] = b_{gp}$

We do not observe b_{gp} directly. Instead, we observe a noisy estimate obtained from some inference method (e.g. DESeq2, GLM, Seurat,…)

$\hat{b}_{gp}$: Estimated effect (LFC)

b_{gp}: True, unknown effect

$\mathbb{F}_{gp}$: Distribution with known info:

$$\mathrm{Var}(\hat{b}_{gp}) = \sigma^2_{gp}$$

$$\pi_{gp} = \mathbb{P}\left(\left|\hat{B}_{gp}\right| \geq \left|\hat{b}_{gp}\right| \,\middle|\, b_{gp} = 0\right)$$

We assume that the matrix of perturbation effects has a **low-rank structure:**

$$b_p = W z_p + \varepsilon_p, \quad \varepsilon_p \sim \mathcal{N}(0, \Psi)$$

$$b_p \sim \mathcal{N}(0, W^T W + \Psi)$$

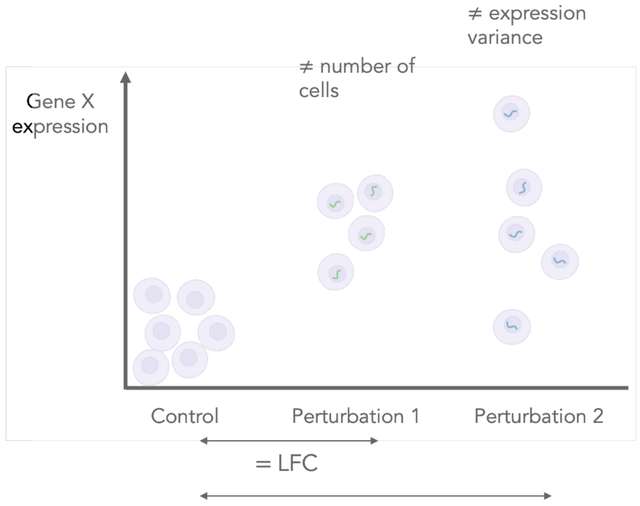
$$\Psi = \mathrm{diag}(\phi_1^2, ..., \phi_G^2)$$

$W \in \mathbb{R}^{G \times K}$: Gene loadings on latent factors (K)

$z_p \in \mathbb{R}^K$: Perturbation's latent factor

ε_p: Gene-specific residuals

INTEGRATING UNCERTAINTY



How to incorporate LFC uncertainty in the model?

DATA AND ASSUMPTION

For each gene *g* and perturbation *p*: $\hat{b}_{gp} \sim \mathbb{F}_{gp}, \quad \mathbb{E}[\hat{b}_{gp}] = b_{gp}$

$\hat{b}_{gp}$: Estimated effect (LFC)

b_{gp}: True, unknown effect

$\mathbb{F}_{gp}$: Distribution with known info:

$$\mathrm{Var}(\hat{b}_{gp}) = \sigma^2_{gp}$$

$$\pi_{gp} = \mathbb{P}\left(\left|\hat{B}_{gp}\right| \geq \left|\hat{b}_{gp}\right| \,\middle|\, b_{gp} = 0\right)$$

We assume that the matrix of perturbation effects has a **low-rank structure:**

$$b_p = W z_p + \varepsilon_p, \quad \varepsilon_p \sim \mathcal{N}(0, \Psi)$$

$$b_p \sim \mathcal{N}(0, W^T W + \Psi)$$

$\Psi = \mathrm{diag}(\phi_1^2, ..., \phi_G^2)$

$W \in \mathbb{R}^{G \times K}$: Gene loadings on latent factors (K)

$z_p \in \mathbb{R}^K$: Perturbation's latent factor

ε_p: Gene-specific residuals

APPROACHES

1. Direct naïve application

$$\hat{b}_{gp} = \sum_k w_{gk} z_{kp} + \varepsilon_g$$

2. Variance weighting (z-score)

$$\frac{\hat{b}_{gp}}{\sigma_{gp}} = \sum_k w_{gk} z_{kp} + \varepsilon_g$$

3. P-value weighting

$$\hat{b}_{gp} \cdot (-\log_{10} \pi_{gp}) = \sum_k w_{gk} z_{kp} + \varepsilon_g$$

4. Decomposition of noise

$$\hat{b}_{gp} = \sum_k w_{gk} z_{kp} + \varepsilon_g + \zeta_{gp}$$

INTEGRATING UNCERTAINTY

APPROACHES

1. Direct naïve application

$$\hat{b}_{gp} = \sum_k w_{gk} z_{kp} + \varepsilon_g$$

2. Variance weighting (z-score)

$$\frac{\hat{b}_{gp}}{\sigma_{gp}} = \sum_k w_{gk} z_{kp} + \varepsilon_g$$

3. P-value weighting

$$\hat{b}_{gp} \cdot (-\log_{10} \pi_{gp}) = \sum_k w_{gk} z_{kp} + \varepsilon_g$$

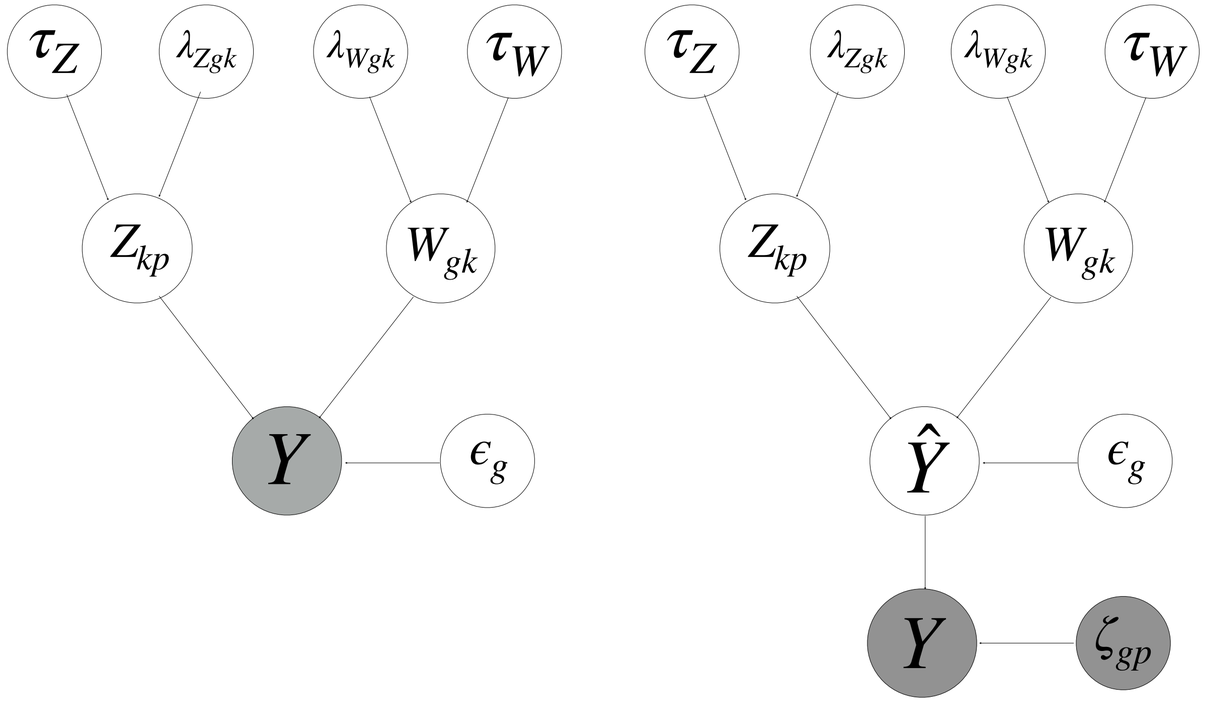
4. Decomposition of noise

$$\hat{b}_{gp} = \sum_k w_{gk} z_{kp} + \varepsilon_g + \zeta_{gp}$$

IMPLEMENTATION

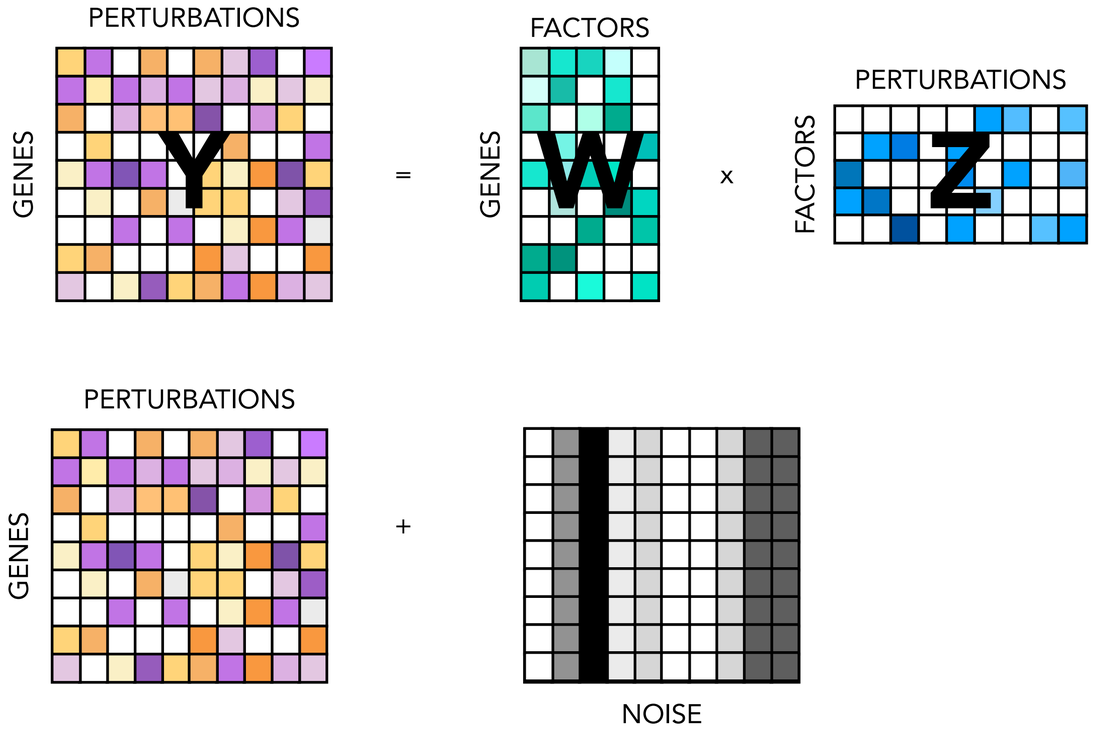
PYRO

Implementation of two different models in Pyro:



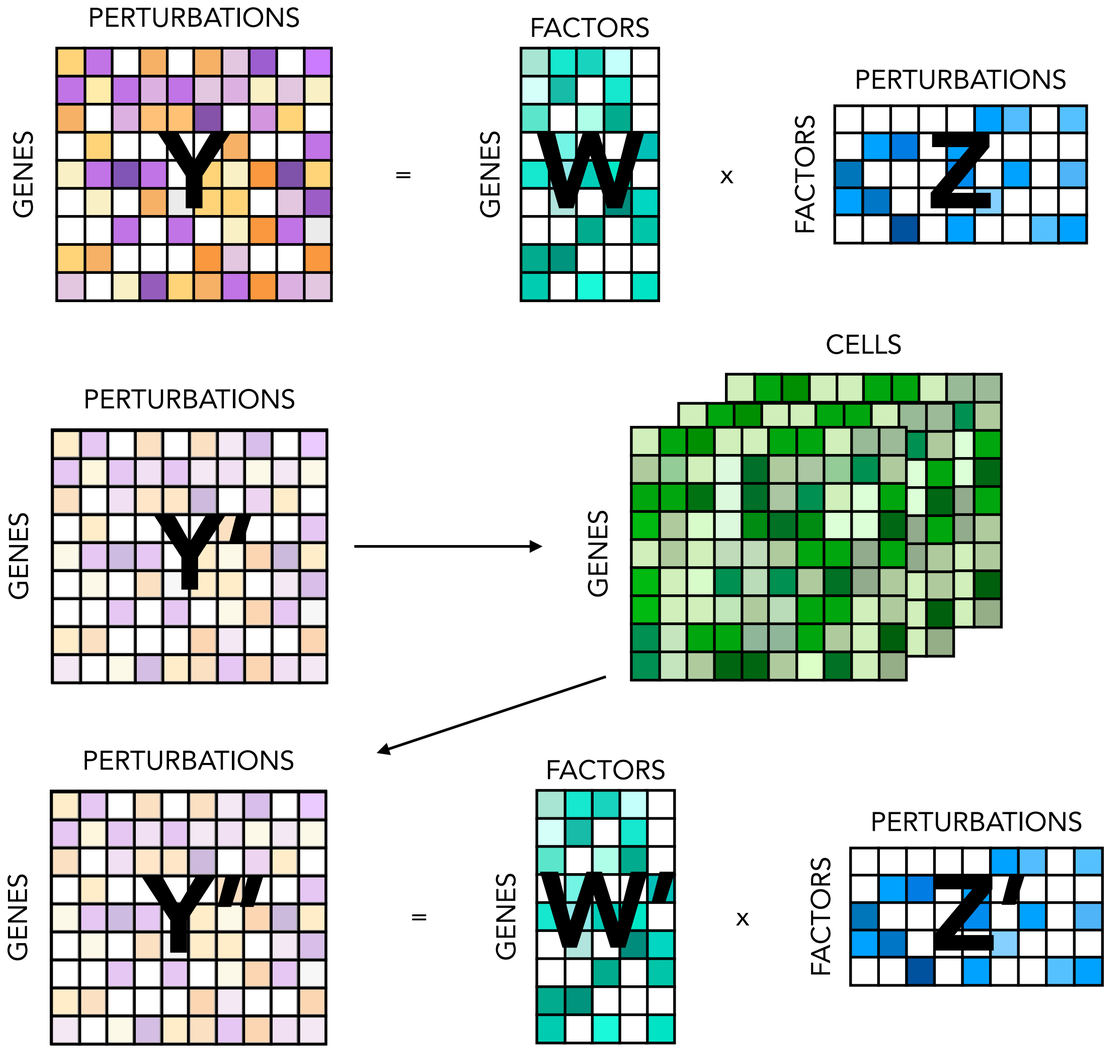
1. Context
2. Approach using Factor Analysis
3. Integrating Uncertainty
4. Data Simulation
5. Application in Real Datasets
6. Conclusion and Future Perspectives

DATA SIMULATION



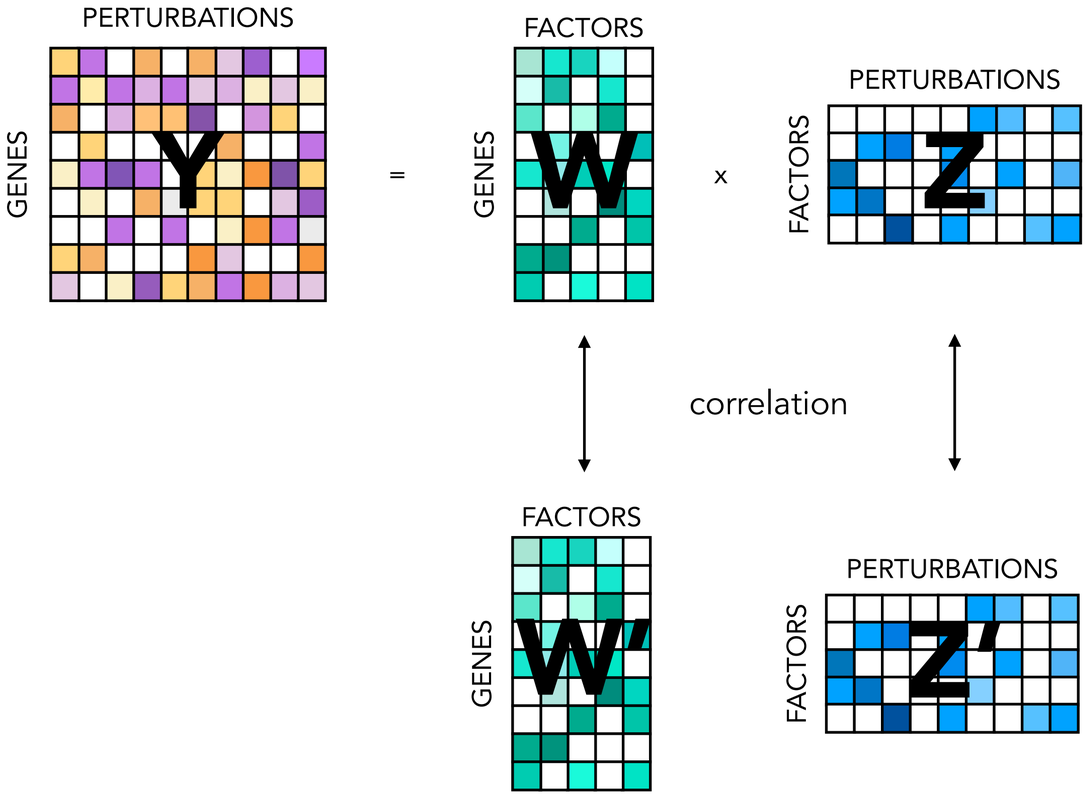
1. Simulate sparse W and Z;
2. Compute Y;
3. Add noise:
    - independent
    - gene-specific
    - perturbation-specific

DATA SIMULATION



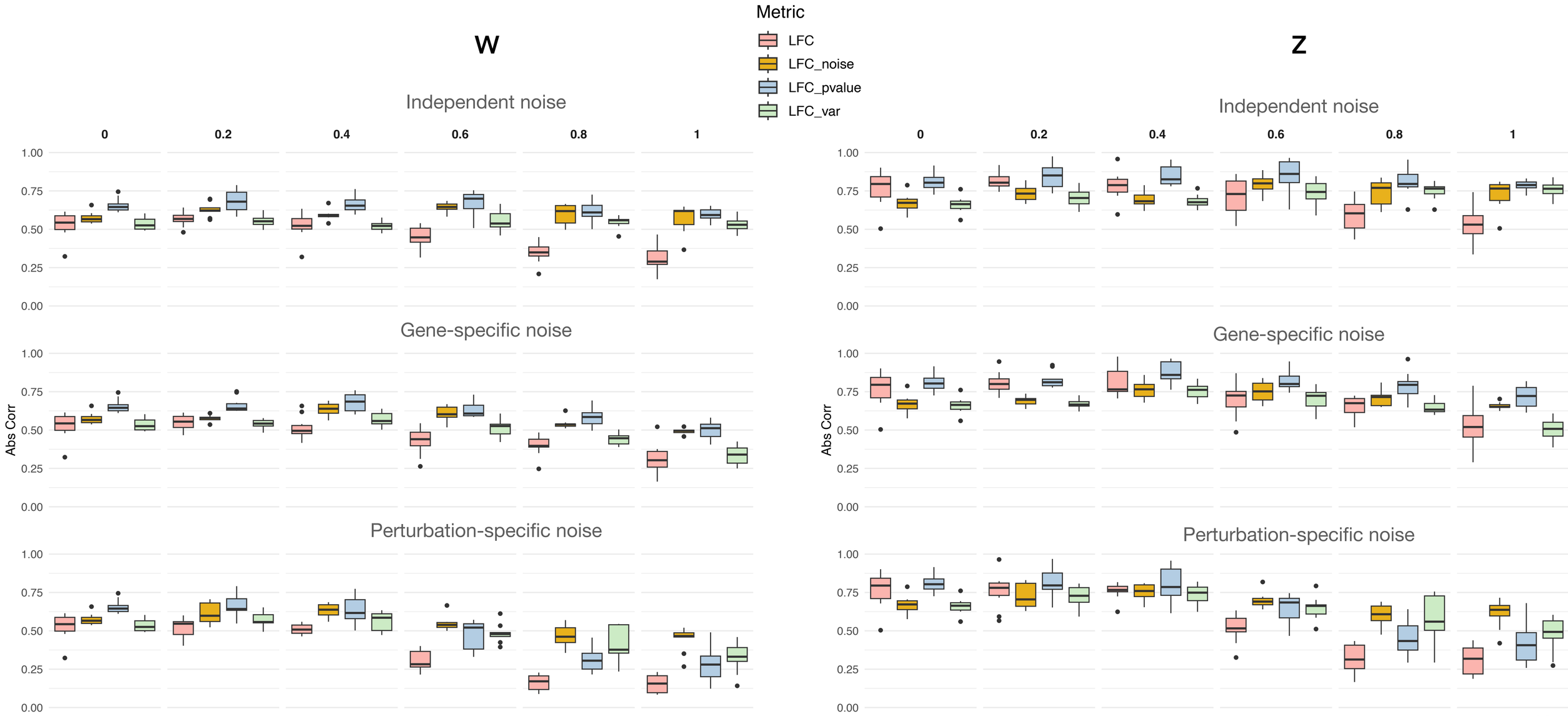
1. Simulate sparse W and Z;
2. Compute Y;
3. Add noise:
    - independent
    - gene-specific
    - perturbation-specific
4. Generate control and perturbed single-cell expression based on Y (with adaptation from splatter);
5. Calculate new LFCs and weighted LFCs;
6. Use FA models to get new estimated W' and Z';

DATA SIMULATION



1. Simulate sparse W and Z;
2. Compute Y;
3. Add noise:
    - independent
    - gene-specific
    - perturbation-specific
4. Generate control and perturbed single-cell expression based on Y (with adaptation from splatter);
5. Calculate new LFCs and weighted LFCs;
6. Use FA models to get new estimated W' and Z';
7. Compare estimated with simulated matrices.

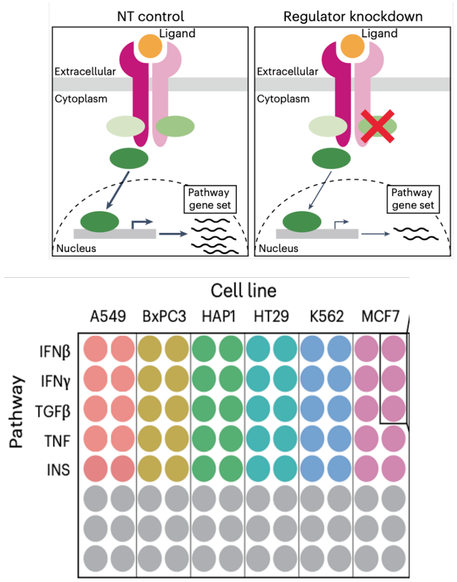
DATA SIMULATION



1. Context
2. Approach using Factor Analysis
3. Integrating Uncertainty
4. Data Simulation
5. Application in Real Datasets
6. Conclusion and Future Perspectives

REAL DATASETS

Jiang et al. 2025

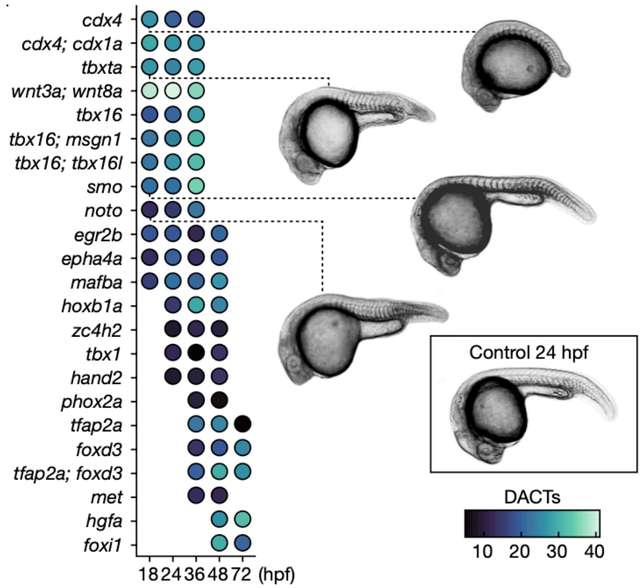


- 6 human cancer cell lines:
  lung, breast, colon, bone marrow, pancreas
- 5 cytokine stimuli for major pathways:
  IFNβ, IFNγ, TGFβ, TNF, INS
- 44–61 literature-curated regulators per pathway

~ 2.6 million cells profiled

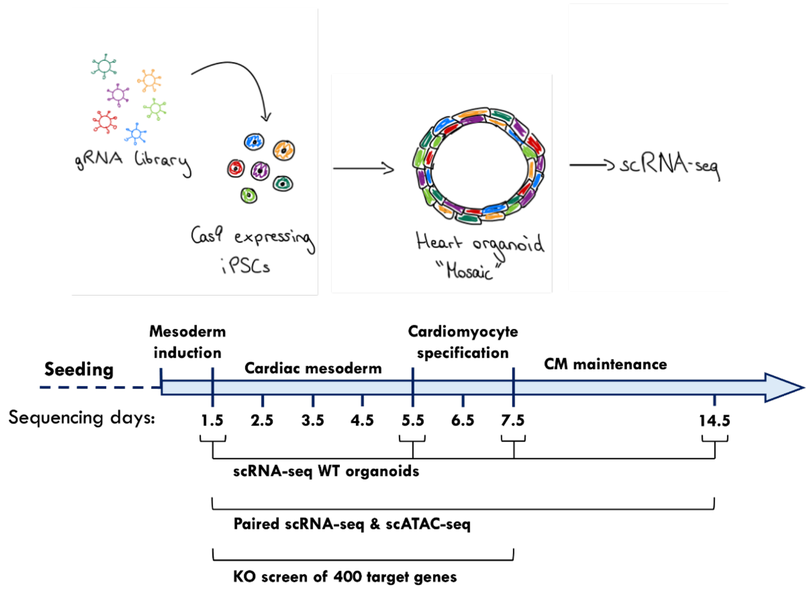
~30K genes

Saunders et al. 2023



- 1812 zebrafish embryos
- 33 major tissues, 99 broad cell types
- 23 genetic perturbations
- 3.2 million cells
- >8 embryos/condition
- 19 timepoints

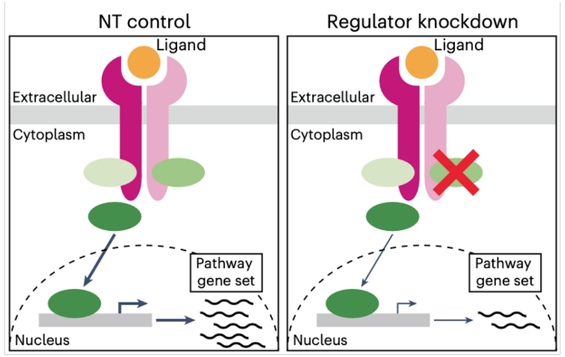
Heart Organoid dataset, Stegle Group, DKFZ (still to be published)

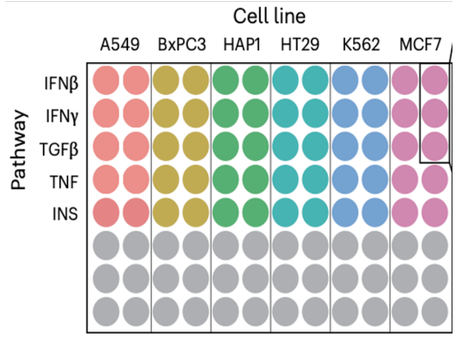


- 400 genetic perturbations
- 15 cell types
- ~1 million cells
- ~20K genes
- 7 timepoints

REAL DATASETS

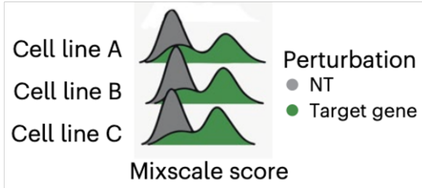
Jiang et al. 2025





Authors strategy:

Weighted differential expression test + MultiCCA decomposition method



DE gene vector for *i*th cell:

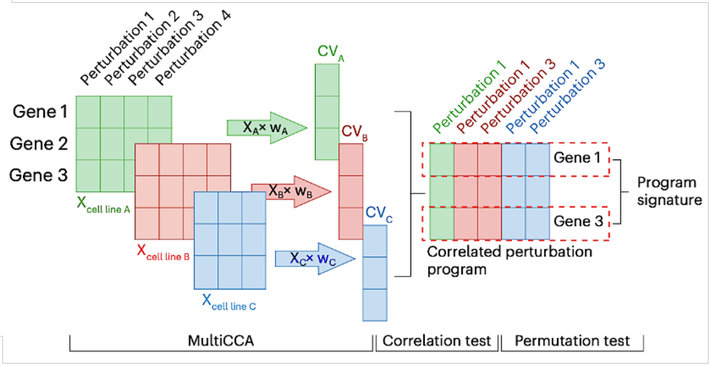
$$\vec{x}_i = [x_{i1}, x_{i2}, \ldots, x_{im}]$$

Mixscale score:

$$s_i = \vec{x}_i \quad (\vec{x}^{P} - \vec{x}^{NP})$$

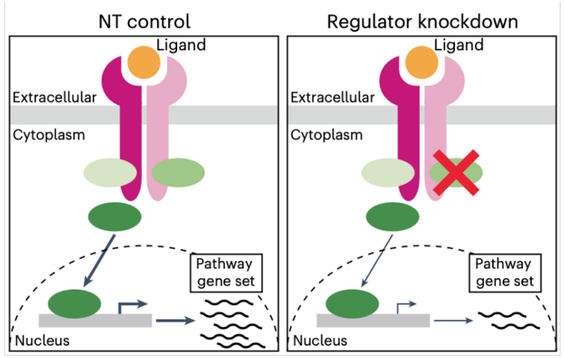
Weighted DE test:

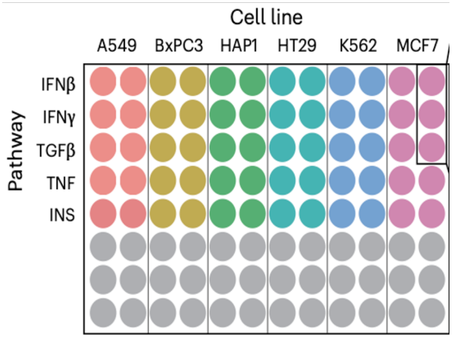
$$\log(E(x)) = \beta \times (sx) + \text{covariates}$$



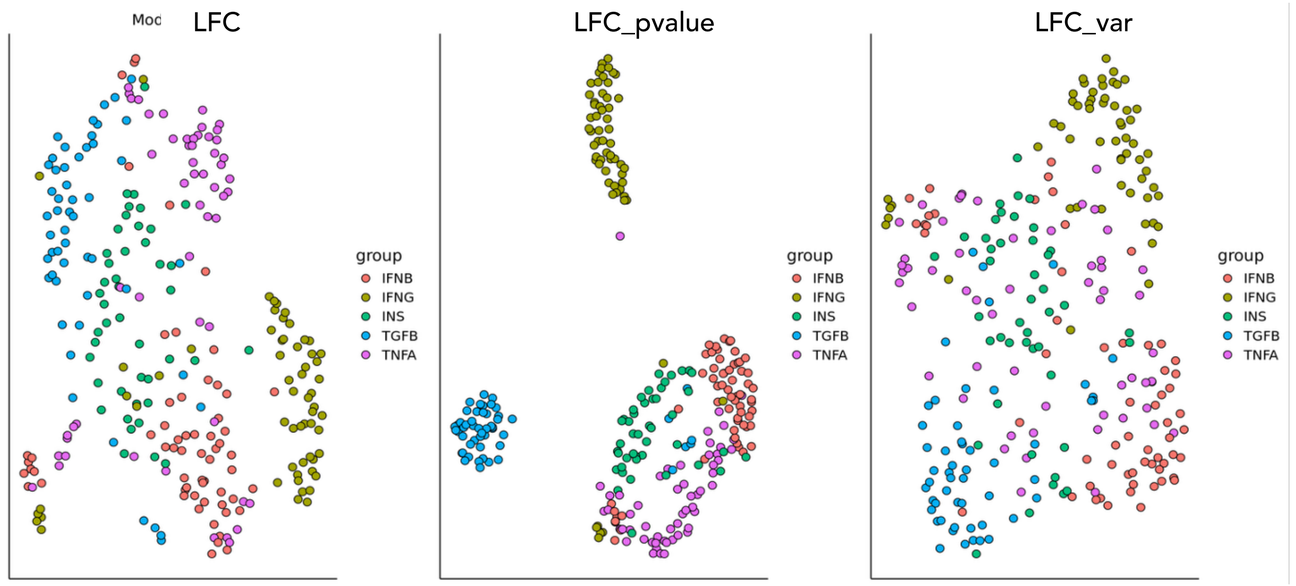
REAL DATASETS

Jiang et al. 2025





Application of MOFA:

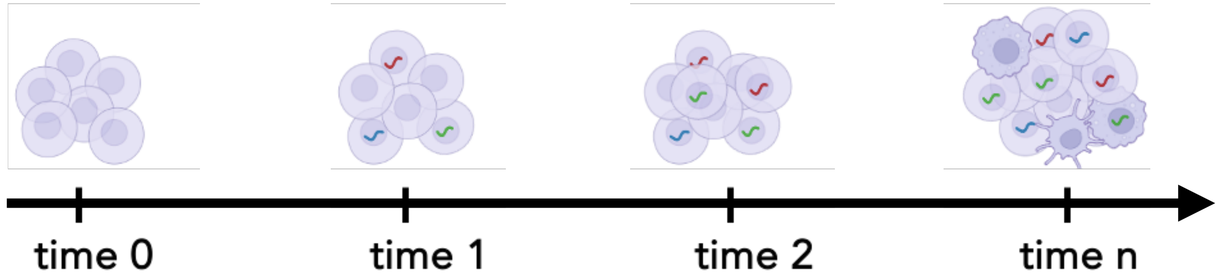


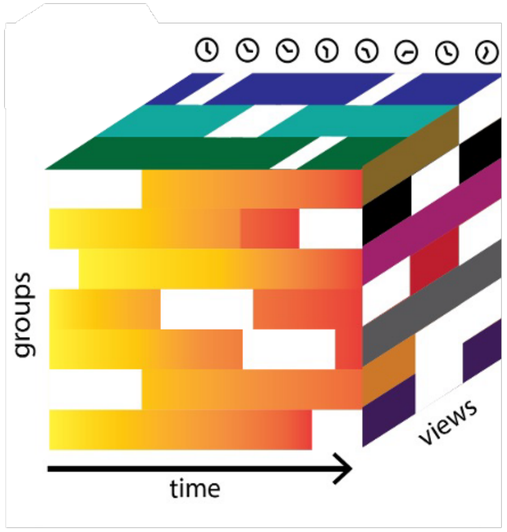
1. Context
2. Approach using Factor Analysis
3. Integrating Uncertainty
4. Data Simulation
5. Application in Real Datasets
6. Conclusion and Future Perspectives

CONCLUSION

- Single-cell CRISPR screens offer rich, context-specific insights—but are still challenging to analyze and interpret;
- Multi-view/group factor models (like MOFA) might allow a more comprehensive analysis of perturbation responses across several biological contexts;
- Integrating uncertainty into the models can improve the interpretability and reliability of inferred responses:
  - Initial simulations suggest that uncertainty-informed models recover more relevant and structured latent factors.
  - Real-data applications are underway - early results look promising.

FUTURE STEPS





ACKNOWLEDGMENTS

AG Velten

JunProf Dr. Britta Velten
Dr. Nikolai Köhler
Jana Braunger
Zehua Zhang
Purusharth Saxena
Wangjun Hu
Dr. Stijn Hawinkel
Jan Sprengel
Fabian Linsenmeier
Thomas Greulich
Dr. Yen-Hsi Beyer

PhD advisors and TAC Members

JunProf Dr. Britta Velten
Prof Dr. Julio Saez-Rodriguez
JunProf Dr. Lauren Saunders
Dr. Ricardo Ramirez-flores

PAGES

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

CENTRE FOR ORGANISMAL STUDIES

VELTEN GROUP

HGS MathComp

IWR Interdisciplinary Center for Scientific Computing

DFG

ellis European Laboratory for Learning and Intelligent Systems

APPENDIX

ADAPTED SPLATTER

1. **Simulate Gene Mean Expression**
   - Sample gene means λ_i from a **Gamma distribution**:

$$\lambda_i \sim \text{Gamma}(\alpha, \beta)$$

2. **Introduce High Expression Outliers**
   - With probability π_O, a gene is an outlier.
   - Replace its mean with an inflated value:

$$\lambda_i' = \text{median}(\lambda) \times F$$

   - Inflation factor F sampled from a **log-normal distribution**:

$$F \sim \text{LogNormal}(\mu_O, \sigma_O)$$

3. **Adjust for Library Size**
   - Simulate library sizes L_j from a **log-normal distribution**:

$$L_j \sim \text{LogNormal}(\mu_L, \sigma_L)$$

   - Adjust mean expression per cell:

$$\lambda_{i,j} = \lambda_i \times \frac{L_j}{\sum L_j}$$

4. **Enforce Mean-Variance Trend**
   - Sample the **biological coefficient of variation (BCV)** from a **scaled inverse chi-squared distribution**:

$$\text{BCV}_i \sim \text{Scaled-Inv-}\chi^2(\nu, s^2(\lambda_i))$$

   - Adjust gene means using a **Gamma distribution**:

$$\lambda_{i,j} \sim \text{Gamma}\left(\frac{1}{\text{BCV}_i^2}, \frac{\lambda_i}{\text{BCV}_i^2}\right)$$

5. **Generate Count Matrix**
   - Sample counts from a **Poisson distribution**:

$$Y_{i,j} \sim \text{Poisson}(\lambda_{i,j})$$

6. **Introduce Dropout (Zero Inflation)**
   - Define dropout probability using a **logistic function**:

$$P(\text{zero}|\lambda_{i,j}) = \frac{1}{1 + e^{-k(\lambda_{i,j} - x_0)}}$$

   - Replace values with zero based on **Bernoulli sampling**:

$$Z_{i,j} \sim \text{Bernoulli}(P(\text{zero}|\lambda_{i,j}))$$

STEPS

1. Estimate α and β from control cells in real dataset
2. Generate control dataset (I am not using outliers nor dropout), with random number of cells.
3. Use simulated Y values to get gene means for each perturbation: control_means * 2^logFC
4. Generate perturbation dataset with random number of cells for each perturbation:
   - Input gene means instead of sampling from gamma

APPENDIX

What is BBVI and why do we need it?

CORE IDEA

BBVI is a flexible alternative that uses **sampling and gradients** to estimate and optimize the ELBO, without **model-specific derivations.**

GOAL

We want to compute the gradient: $$\nabla_\phi \mathcal{L}(\phi) = \nabla_\phi \mathbb{E}_{z\sim q_\phi(z)} \left[\log \frac{p(x,z)}{q_\phi(z)}\right]$$

PROBLEM

Exact integration is intractable, we need to **approximate the expectation using sampling**
But z **is sampled from** $q_\phi(z)$ **which depends on** ϕ, so we can't move the gradient inside naïvely

$$\nabla_\phi \mathbb{E}_{z\sim q_\phi(z)}[f(z)] \neq \mathbb{E}_{z\sim q_\phi(z)}[\nabla_\phi f(z)]$$

APPROACH

Rewrite the expression as an **expectation of the gradient** using tricks

Score-Function Estimator (REINFORCE)

Reparameterization Trick

APPENDIX

BBVI in practice

Pseudocode:

1. **Initialize** variational parameters λ
2. **Repeat until convergence**:
   - **Sample** $z^{(1)}, \ldots, z^{(S)} \sim q_\lambda(z)$
     → use reparameterization if available, otherwise sample directly
   - For each sample $z^{(s)}$, compute:
     - $\log p(x, z^{(s)})$
     - $\log q_\lambda(z^{(s)})$
     - **If using score-function**:

$$\nabla_\lambda \log q_\lambda(z^{(s)})$$

     - **If using reparameterization**:

$$\nabla_\lambda \left(\log p(x, z^{(s)}) - \log q_\lambda(z^{(s)}) \right)$$

   - **Estimate gradient of ELBO**:
     - Score-function:

$$\nabla_\lambda \mathcal{L} \approx \frac{1}{S} \sum_{s=1}^{S} \left(\log p(x, z^{(s)}) - \log q_\lambda(z^{(s)}) \right) \cdot \nabla_\lambda \log q_\lambda(z^{(s)})$$

     - Reparameterization:

$$\nabla_\lambda \mathcal{L} \approx \frac{1}{S} \sum_{s=1}^{S} \nabla_\lambda \left(\log p(x, z^{(s)}) - \log q_\lambda(z^{(s)}) \right)$$

   - **Update** λ using gradient ascent (e.g. Adam)

You write only the model and the guide.

Pyro takes care of ELBO gradients, chooses the right estimator, and runs the optimizer in just a few lines of code.

Model $p(x, z)$	`def model(x): ...` `pyro.sample(...)`
Guide $q_\phi(z)$	`def guide(x): ...` `pyro.sample(...)`
ELBO Objective	`loss=Trace_ELBO()`
Gradient Ascent on ELBO	`svi.step(x)`

❌ No ELBO derivation

❌ No gradient estimator coding

❌ No optimizer steps written by hand