Batch correction for SC-RNAseq data

Panagiotis Papasaikas

FMI Computational Biology

- **Batch effects in SC RNAseq data**

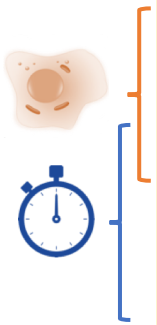
- **Data integration / batch correction**
  - Definition and objective

- **Common approaches**
  - Regression-based
  - Graph / MNN based

- **Batch correction evaluation**

- **Data integration with Deep Generative Models**
  - Variational Autoencoders

Sources of variance in SC-RNAseq data

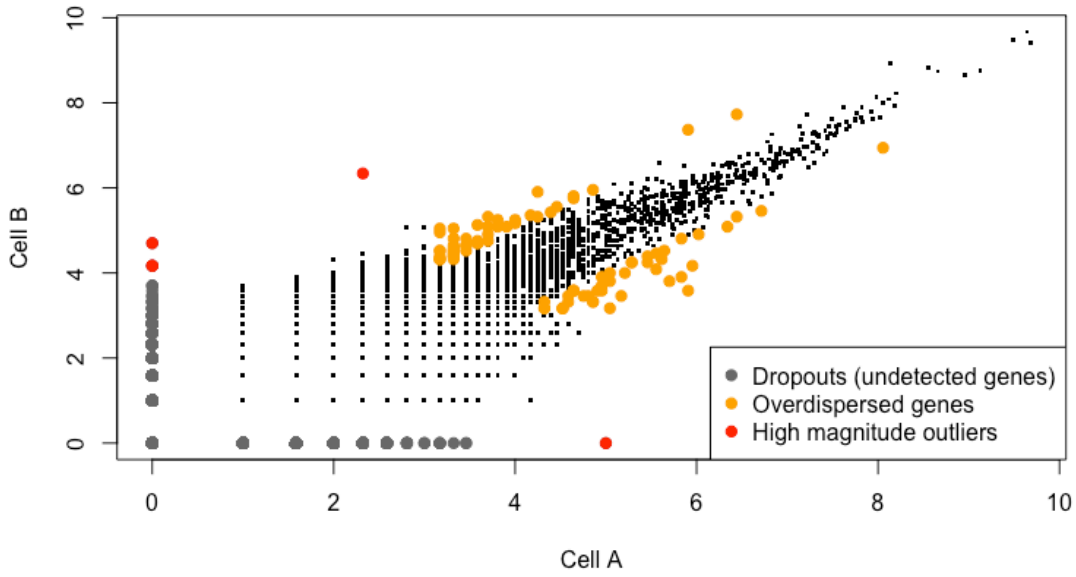


Biological:
- Cell Type Heterogeneity
- Genetics
- Cell State/Microenvir.
- GExpr Stochasticity
- Cell Cycle Dynamics
- Transcriptional Bursts
- Oscillations

...

Technical:
- Capture Efficiency
- Amplification Bias
- PCR artifacts
- Contamination
- Cell Doublets
- Cell Damage
- Sampling (Jackpot Effects)

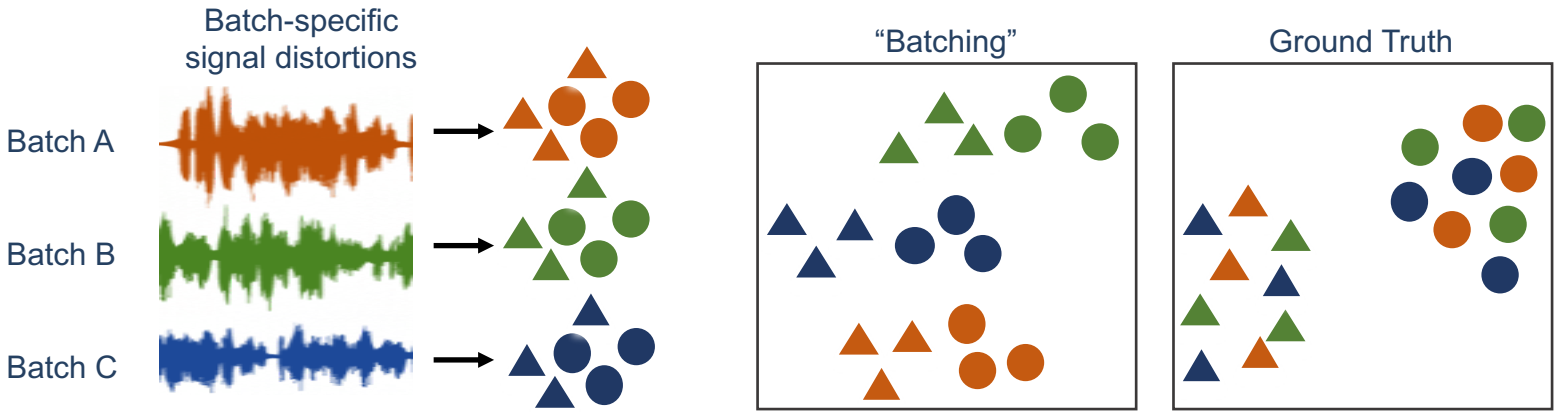
...



Reproduced according to Kharchenko et. al, Nat. Methods 2014

Batch effects

- Caused by technical sources of variation introduced to the dataset during handling/preparation/processing.
- Distortion signals of with different characteristics (e.g intensity, variance) are applied to each technical batch.
- The distortion can have different effects on each of the features (genes) of the dataset

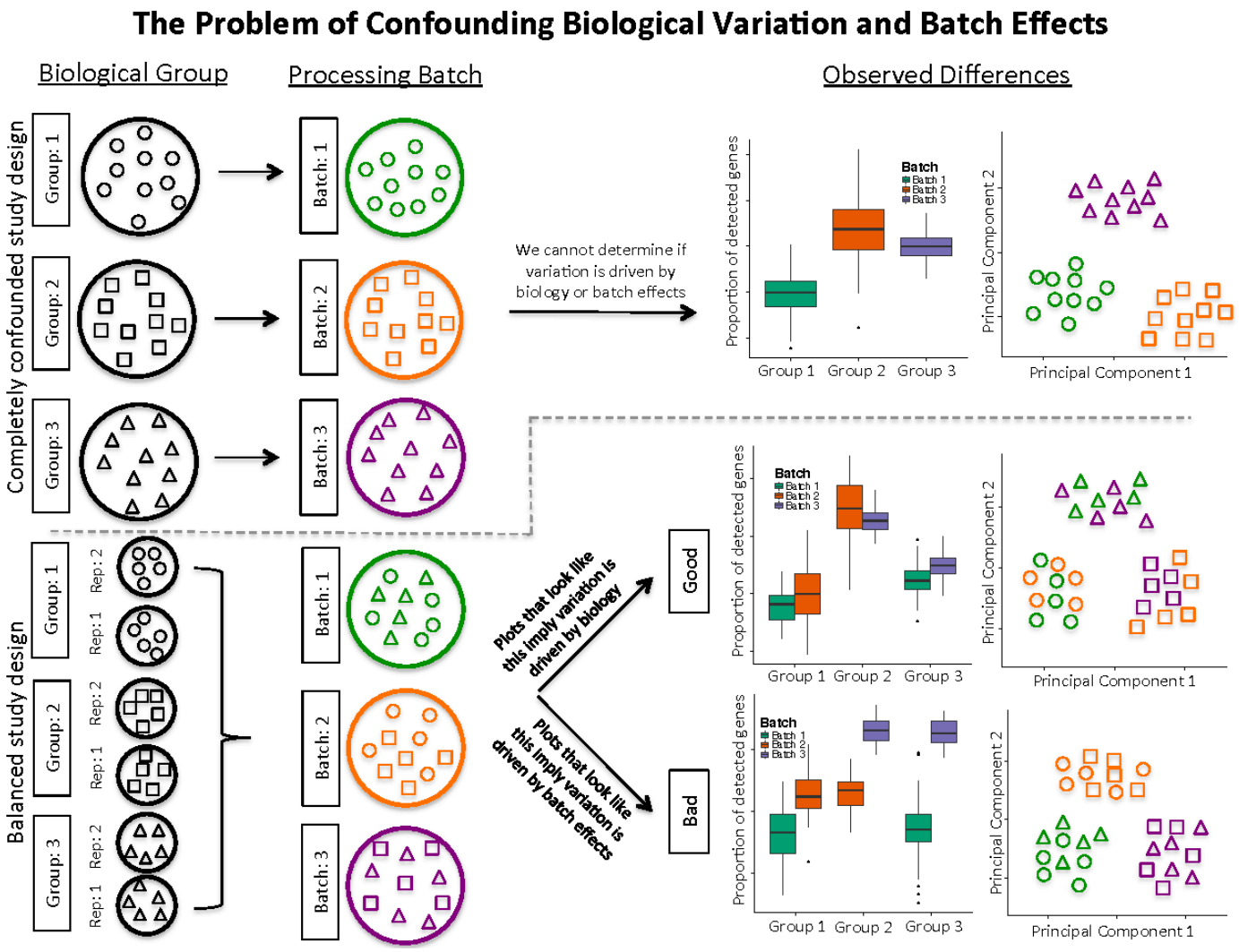


- In the case of **single cell sequencing** the distortions can have different effects on distinct cells within the same batch.
- Can be confounded with biology since batch populations are not identical in composition
- Single cell sequencing involves more, complex steps where batch effects can be introduced
- Due to the small starting sample batch effects are exaggerated.

Terminology disambiguation

Batch effects / batch correction	"Classical" bioinformatics terminology. Typically refers to technical sources of variance
Unwanted sources of variance (nuisance)	A more general, subjective term. Depends on context / goals. Can also be biological in nature (e.g cell cycle)
Data integration / fusion **<u>Manifold</u> alignment**	Typically reserved for more heterogeneous / complex datasets
Style transfer **Covariate shift**	Typically encountered in the machine learning / image processing literature

The importance of proper experimental design



Hicks et al DOI:10.1101/025528

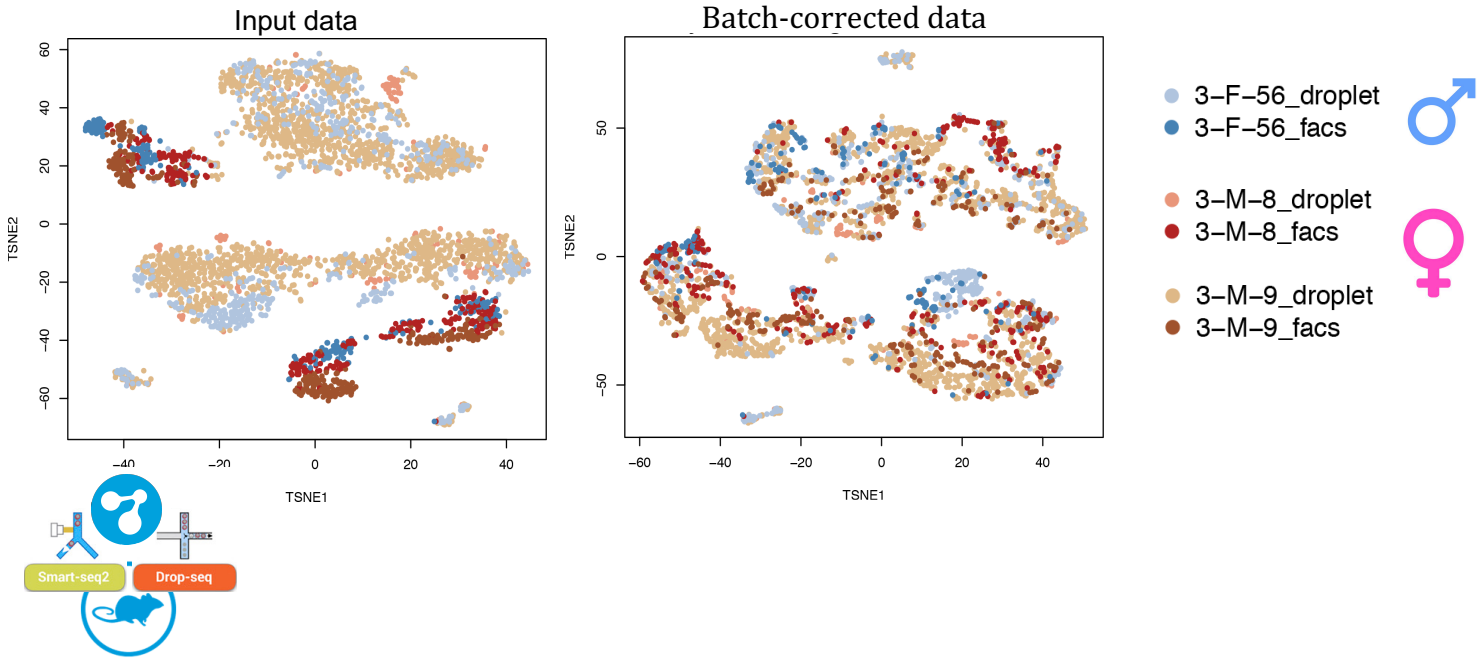
Objectives of batch-correction

We wish to obtain corrected data where the following goals are met:

Goal:		What it practically means:
A. The batch-originating variance is erased	⇢	Similar cell types are intermixed across batches
B. Meaningful heterogeneity is preserved	⇢	We are not mixing distinct cell types (across or within batches)
C. No artefactual variance is introduced	⇢	We do not separate similar cells within batches

<u>Tabula-Muris bladder data</u>

2 Technologies (10x, Smartseq 2)
3 Mice (1 Male, 2 Female)
3 Main subpopulations per batch



Regression based batch effect removal

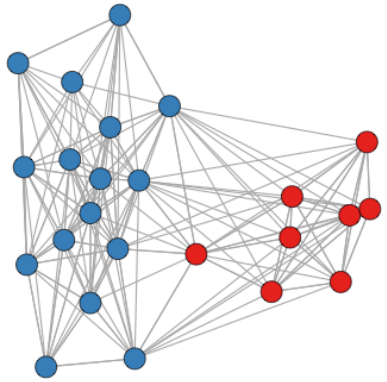
Regressing-out batch effects by specifying blocking factors.
e.g limma::removeBatchEffect() , sva::combat(),
batchelor::rescaleBatches() :

$$y_{ijk} = \mu + \alpha_i + b_{ij} + \epsilon_{ijk}$$

- Do not account for differences in population composition
- Assume batch effect is additive
- Prone to overcorrection (in cases of partial confounding)

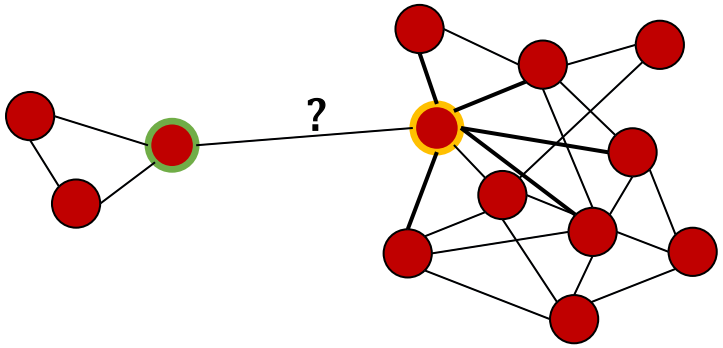
batchelor::rescaleBatches()
Preserves sparsity
Mitigates artificial variance differences

Graph-based approaches



- Cell are represented as vertices in a graph
- Weighted edges based on cell similarity
- Re-cast SC-RNAseq analyses problems as SNA/graph theoretic problems:
  e.g clustering -> Community detection

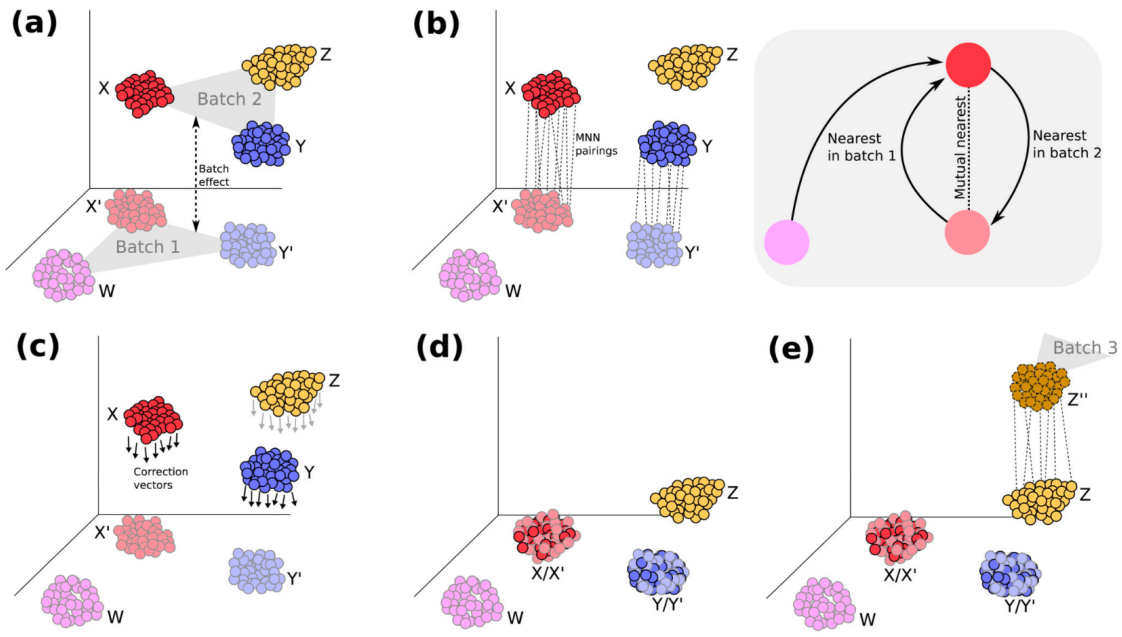
- Leverage local **neighborhood** information (power in numbers):
  Data are noisy / incomplete -> Allow information to be shared / flow /propagate in the graph
- **k-Mutual Nearest Neighbors** (kMNNs) -> A & B are neighbors iff $A \in KNN_B$ AND $B \in KNN_A$

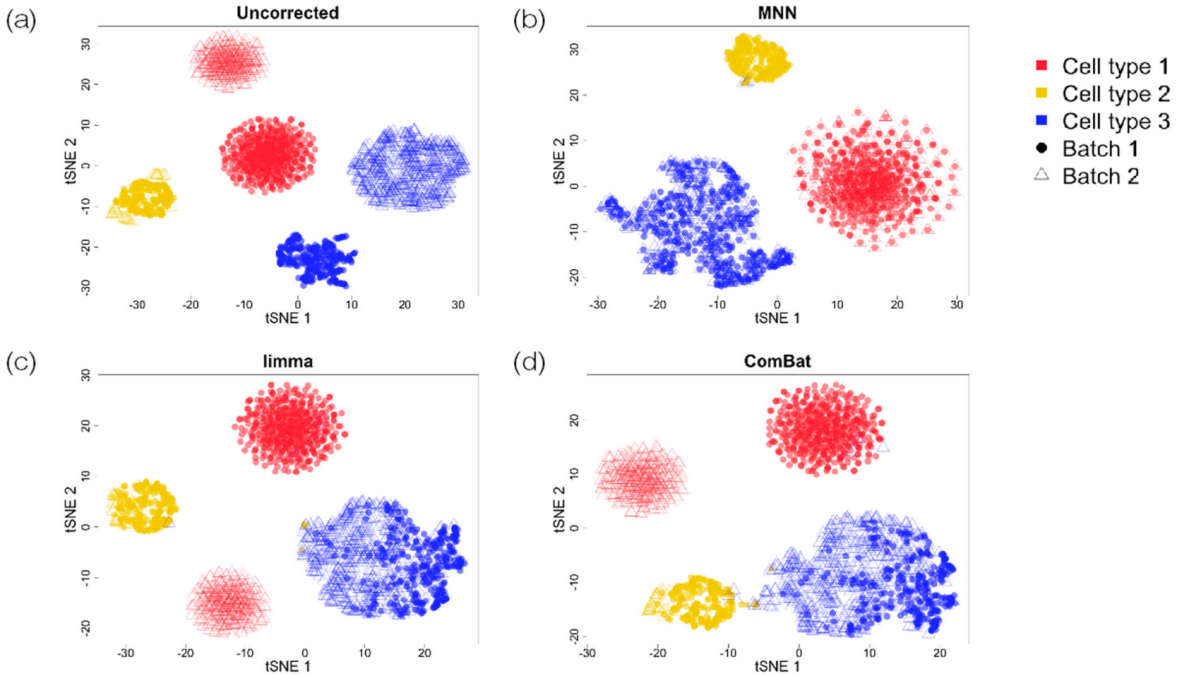


MNN batch correction : Nat Biotechnol. 2018 36(5):421-427
scran/batchelor::fastMNN()
Conos : Nature Methods 2019 16:695–698

Mutual-nearest-neighbors batch correction (MNN)

Haghverdi et al *Nat Biotechnol*. 2018





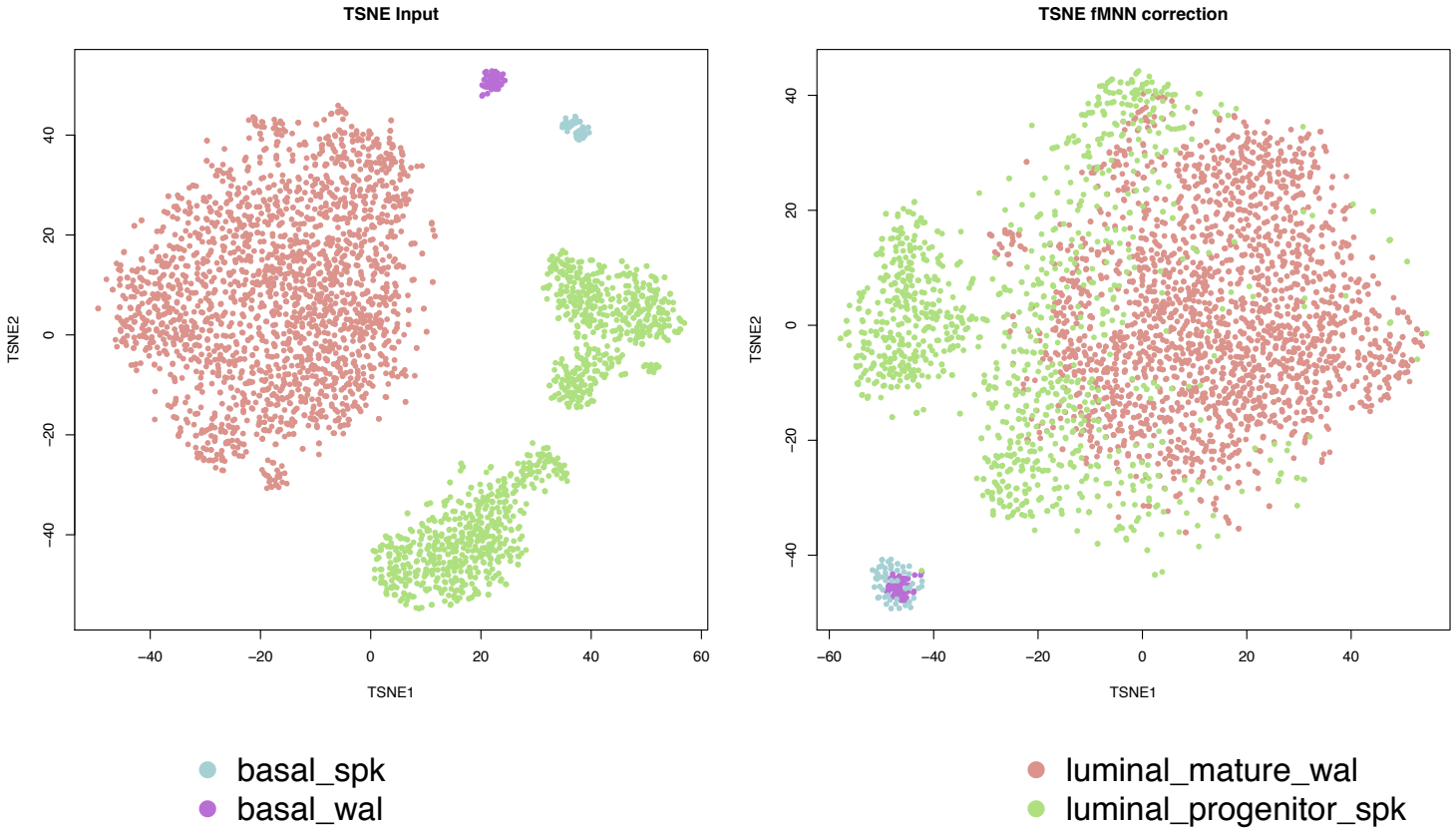
Model assumptions

1. There is at least one cell population that is present in both batches,
2. The batch effect is almost orthogonal to the biological subspace, and
3. batch effect variation is much smaller than the biological effect variation between different cell types

Violating the assumptions

2 batches (wal, spk)
1 cell type (basal) common in both batches
2 cell types constrained to one batch each

This violates the orthogonality assumption



-The higher the number of batches / the more complex the structure of the dataset the more probable some of the assumptions will be violated.
-The higher the number of batches the harder it become to find an optimal merging order.

Further improvements in batch correction pipelines

1. Batch-adapted selection of overdispersed genes:
Select union of overdispersed genes (too liberal, especially for high number of batches)
Select intersection of overdispsersed genes (too strict, especially for high number of batches)
Use scran::combineVar(): Averages variance components across batches

2. Batch-aware size factor calculation and normalization
multibatchNorm: Downscales all size factors to the ones calculated for the lowest coverage batch.

3. Batch-aware dimensionality reduction
multibatchPCA: - Every batch contributes equally to the basis vectors calculation
- Contributions to gene covariance matrix are normalized by number of cells.

commonPCA: - Simultaneous dimensionality reduction.
Joint NNMF
Generalized SVD

Batch-correction evaluation

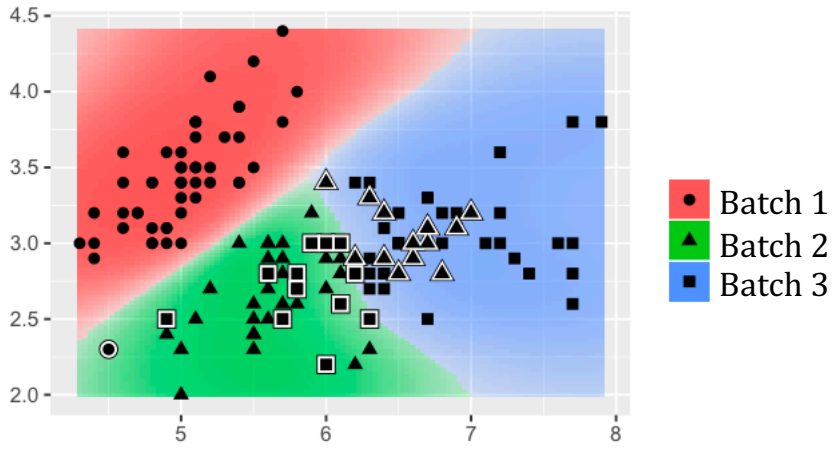
Objectives of batch-correction

We wish to obtain corrected data where the following goals are met:

	Goal:		What it practically means:
A.	The batch-originating variance is erased	⇢	Similar cell types are intermixed across batches
B.	Meaningful heterogeneity is preserved	⇢	We are not mixing distinct cell types (across or within batches)
C.	No artefactual variance is introduced	⇢	We do not separate similar cells within batches

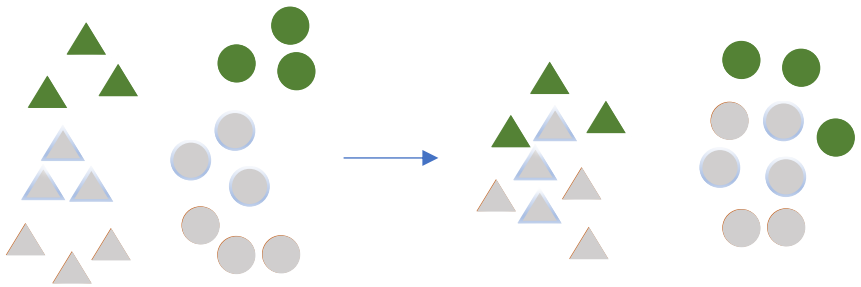
Batch-correction evaluation

1. Evaluate mixing efficiency (Goal A)
- How well mixed are the obtained clusters post-batch correction?
- How well does a classifier (eg SVM) perform pre/post-correction?

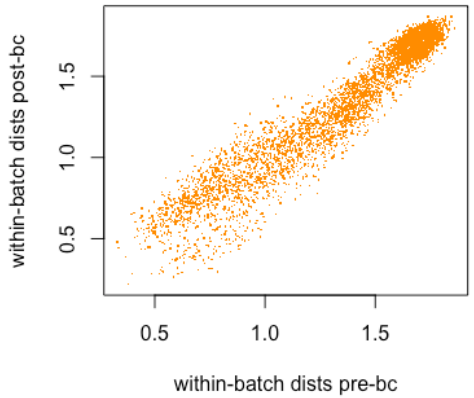


2. Evaluate preservation of remaining variance (Goals B, C)
- Evaluate proportion of removed variance, overlap of HVGs
- Evaluate preservation of within-batch cell topologies:

2A. Local neighborhood structure preservation



2B. Global structure preservation

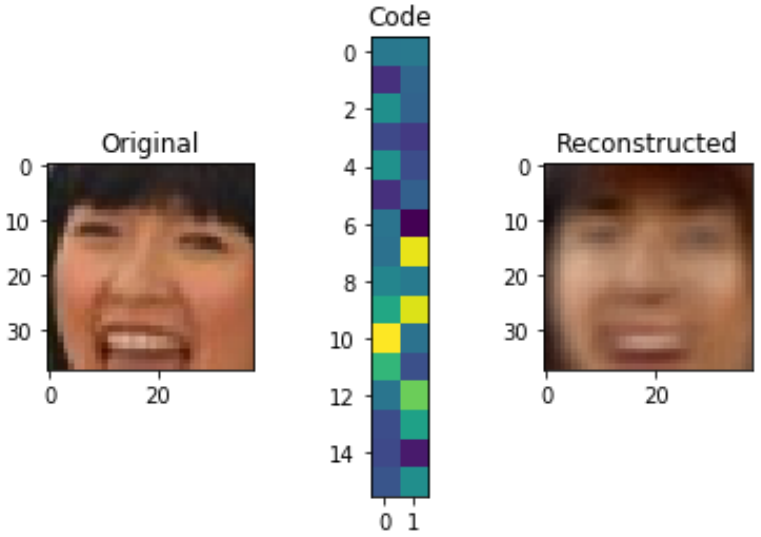


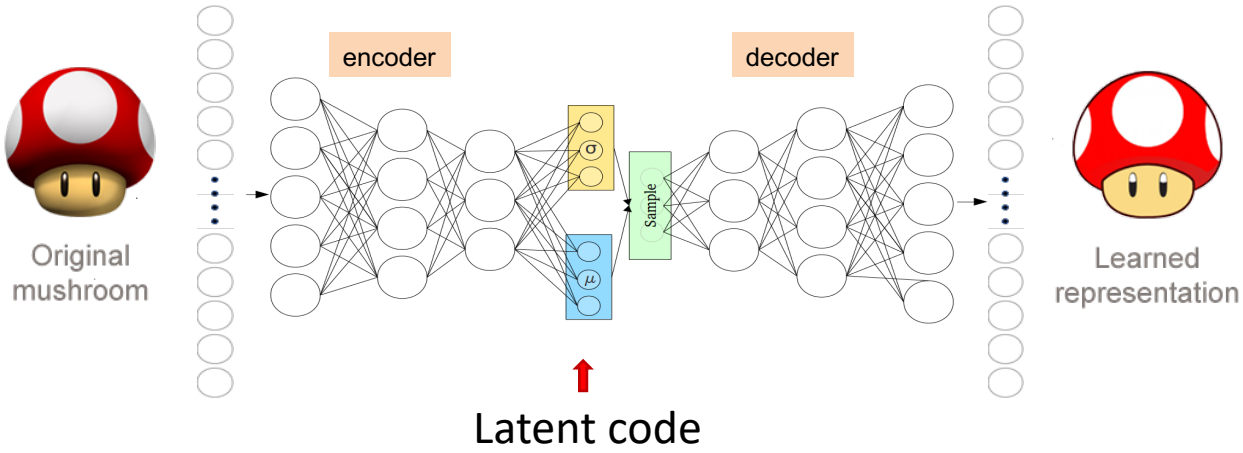
CellMixS: https://bioconductor.org/packages/release/bioc/manuals/CellMixS/man/CellMixS.pdf
kBet: *Nature Methods* **volume 16**, pages43–49 (2019), https://github.com/theislab/kBET

- **Data integration with Deep Generative Models**
  Variational Autoencoders

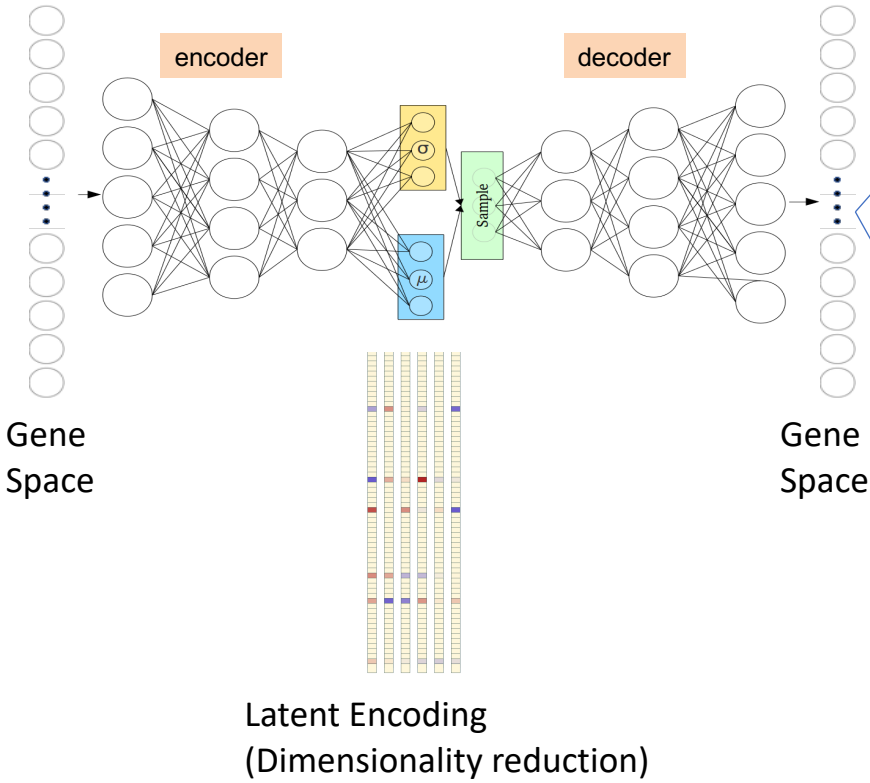
Variational Autoencoders

- Unsupervised representation learning
- Objective is to obtain an output that matches the input.
- Data are “squeezed” through successive layers of decreasing dimensions
- The middle hidden layer is a **code** (latent code) that **represents** the input:



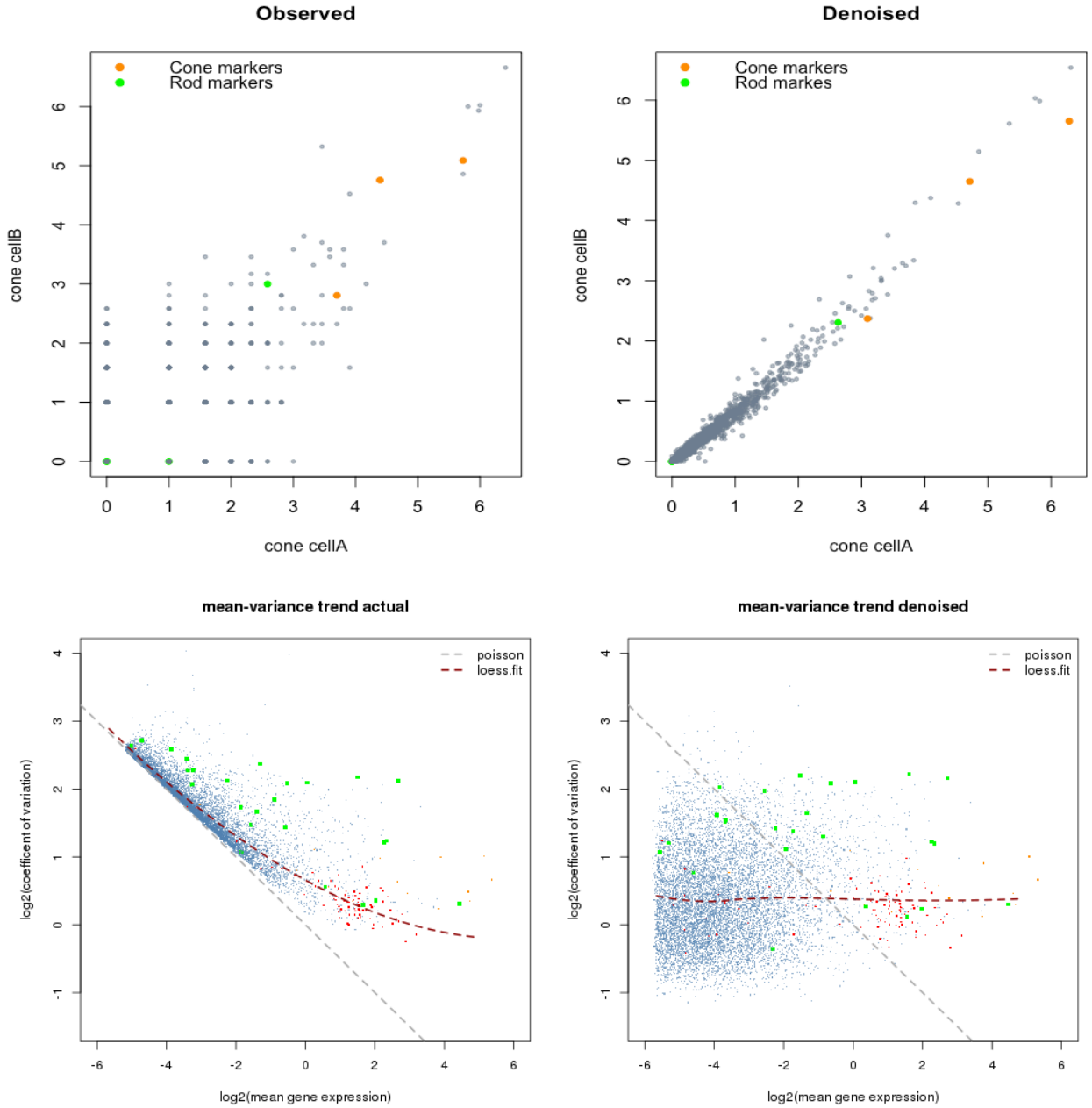


VAEs in single cell data

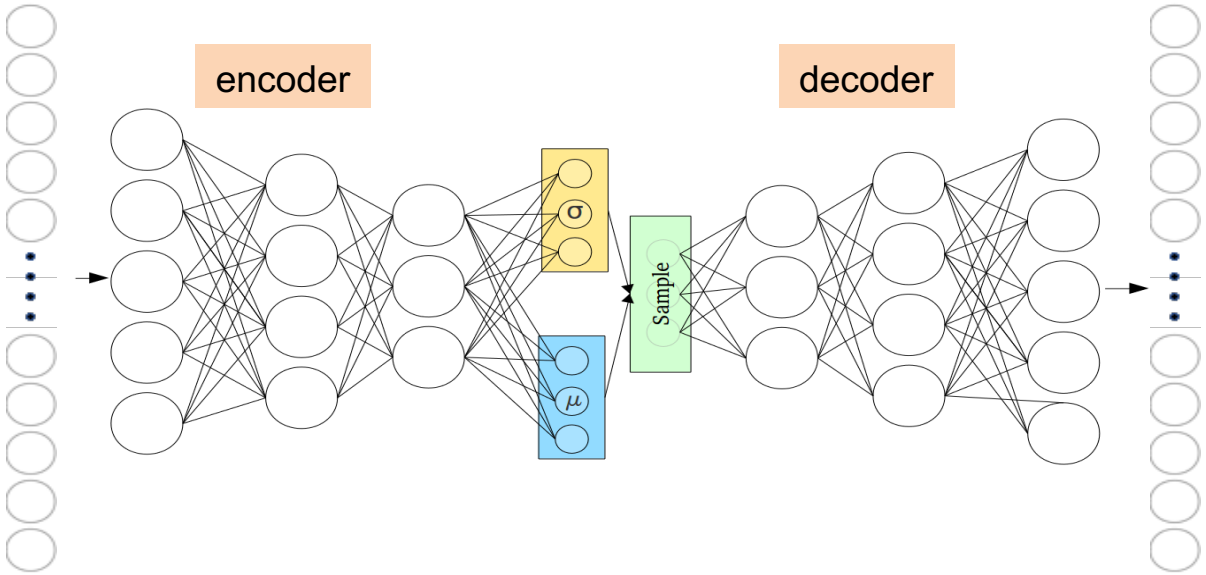


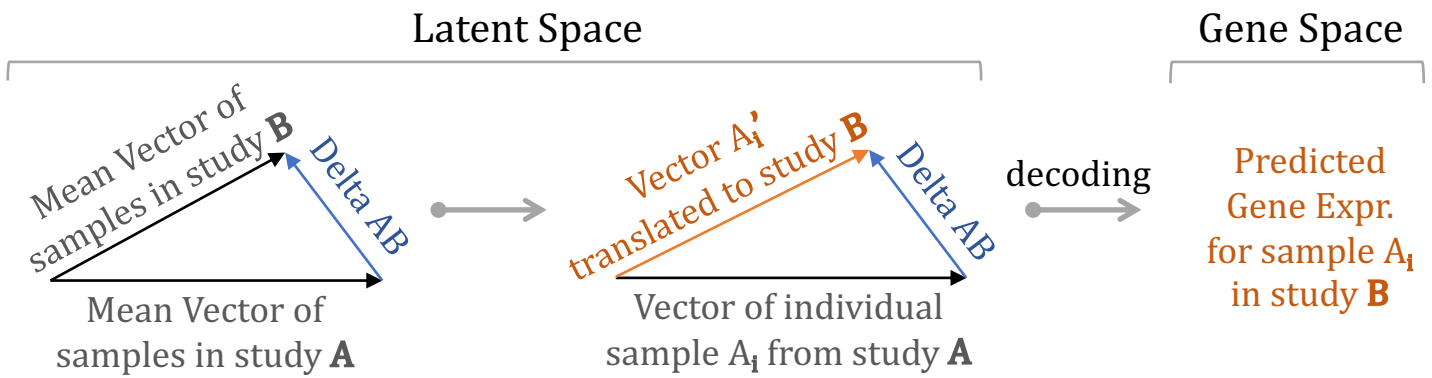
Denoising
(Imputation)

Variance
stabilization



Batch correction using latent vector arithmetic

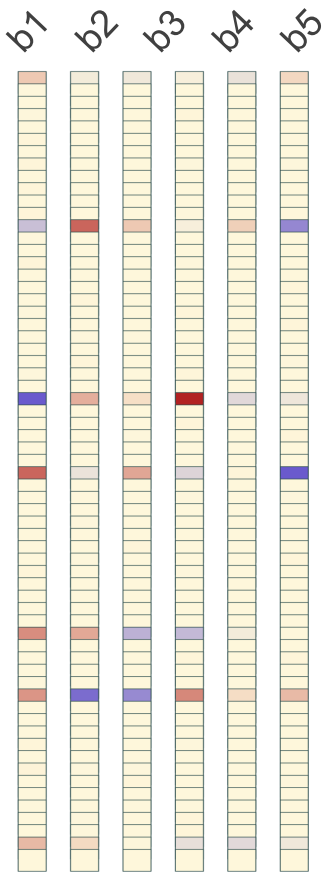




calculate delta vector for batch variance

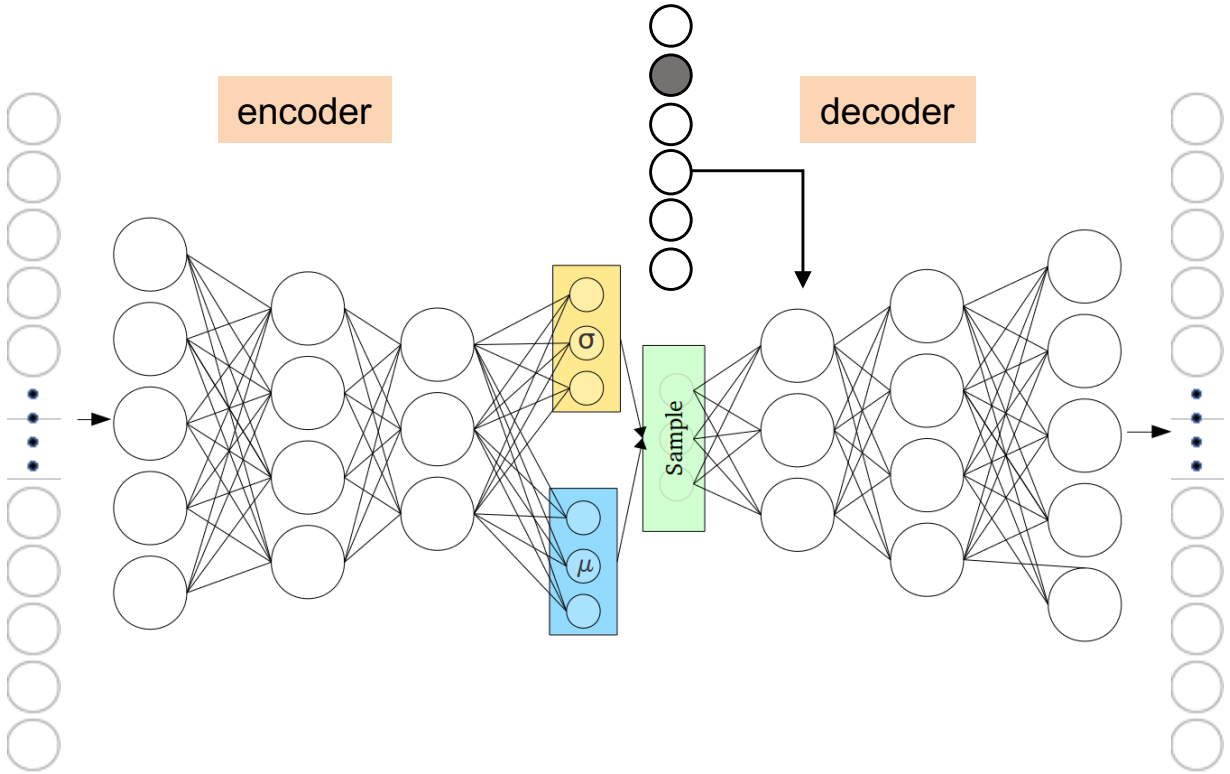
apply delta vector to individual Samples of one study

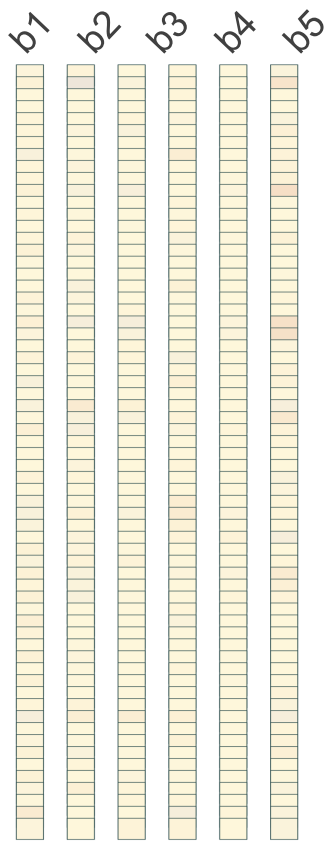
decode translated latent vectors back to gene space



Latent vectors of technical batches normalized to the dataset mean

Batch correction using explicit style encoding





Latent vectors of technical batches normalized to the dataset mean

Applicability range of VAEs for batch correction

Require large training sets

Great at generalizing: The more complex the dataset structure the bigger the advantage over traditional approaches

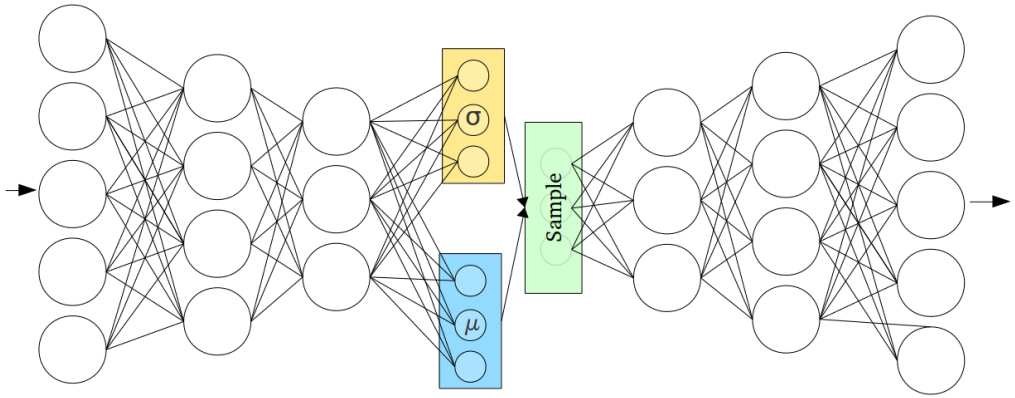
Smaller amount of hand-tunable parameters

Extremely flexible: They can deal with a wide-arrange of of data-integration problems with minimal architectural changes

As a bonus they provide solutions for other SC analysis tasks under a single framework e.g dimensionality reduction, imputation, variance stabilization,

The end ☺

Variational Autoencoders



- VAEs generalize AEs adding stochasticity
- Encourage a continuous latent manifold
- Robustness + valid decoding
- Allows interpolation and exploration

$$\mathcal{L}_\beta = \underbrace{\frac{1}{N}\sum_{n=1}^{N}\left(\mathbb{E}_q[\log p(x_n|z)]\right.}_{\text{Reconstruction}} - \underbrace{\beta\, \mathrm{D}_{\mathrm{KL}}\left(q(z|x_n)||p(z)\right)}_{\text{Distance to latent prior}}$$

- *$\beta = 1$: ELBO (Evidence Lower Bound, standard VAE)*
- *$\beta < 1$: Partially regularized VAE (Liang et al. 2018)*
- *$\beta > 1$: Disentangling Autoencoders (β –VAE, Higgins et al. 2017)*