A review of popular methods for scRNA-seq data integration

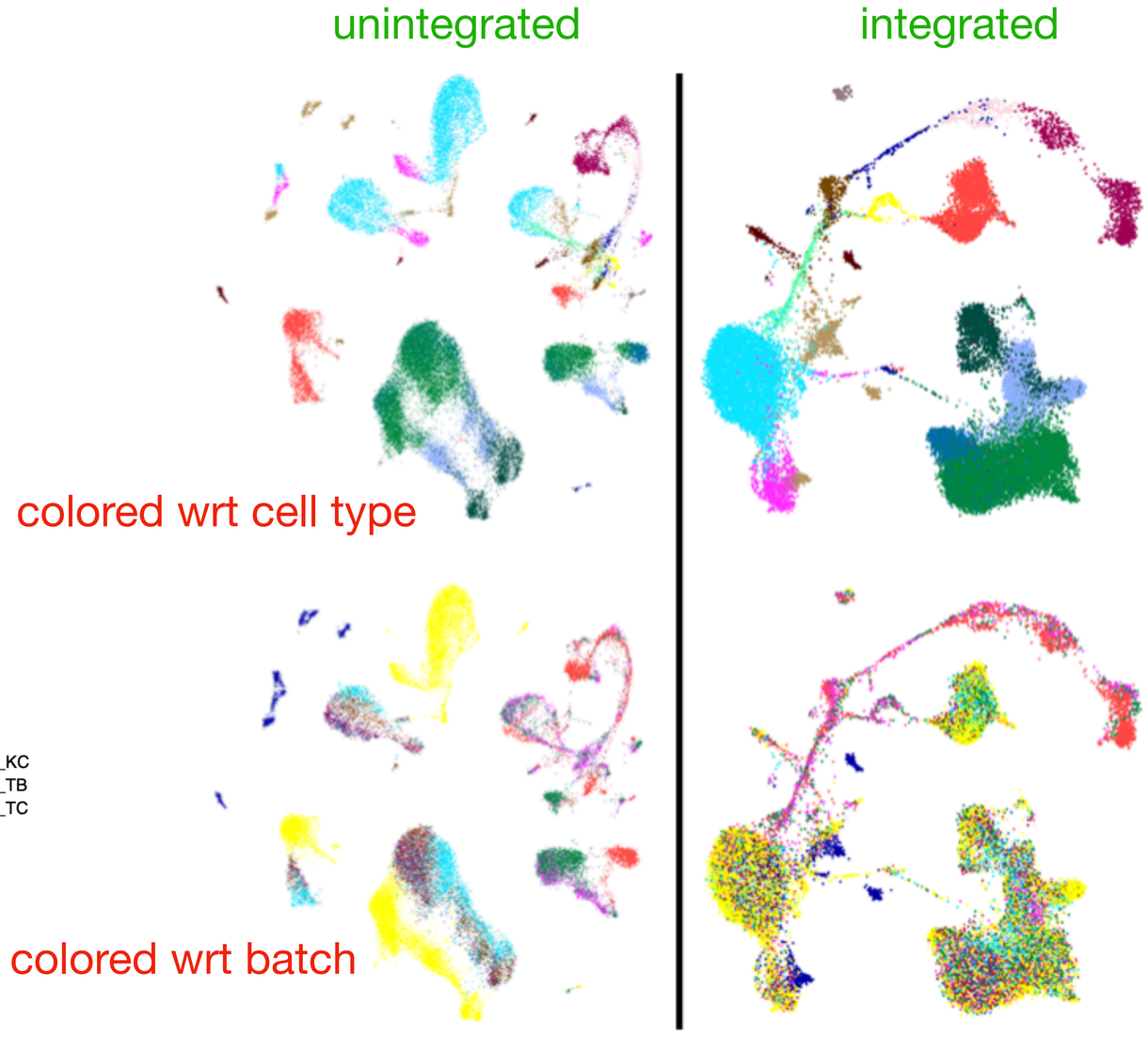
Cesim Erten, March 2023

What is scRNA-seq Data Integration?

Goal

- Integration of two or more transcriptomics datasets into a single dataset on which any downstream analysis can be applied.

Ex: Human immune cell integration
by Scanorama (Luecken et al.)
10 batches, datasets with cells from peripheral blood and bone marrow



What is scRNA-seq Data Integration?

Bulk vs Single-cell

- Some do not prefer the term ‘batch effect correction’, refer to it in the context of bulk RNA-seq data integration:
  - Emphasize datasets from very different sources (tech., lab)
  - Samples with different cell type composition
- scRNA-seq datasets:
  - generated from samples with distinctive characteristics
    e.g., cell counts, tissue types, healthy or diseased
  - using different experimental protocols
    e.g., cell isolation and handling protocols and library preparation methods or sequencing platforms.
  - Differences ⇒ unwanted technical and biological variations across datasets.

limma
Combat

Relevant Problem: Domain Adaptation

features of the data and the distribution of those features in the dataset

label space and an objective predictive function

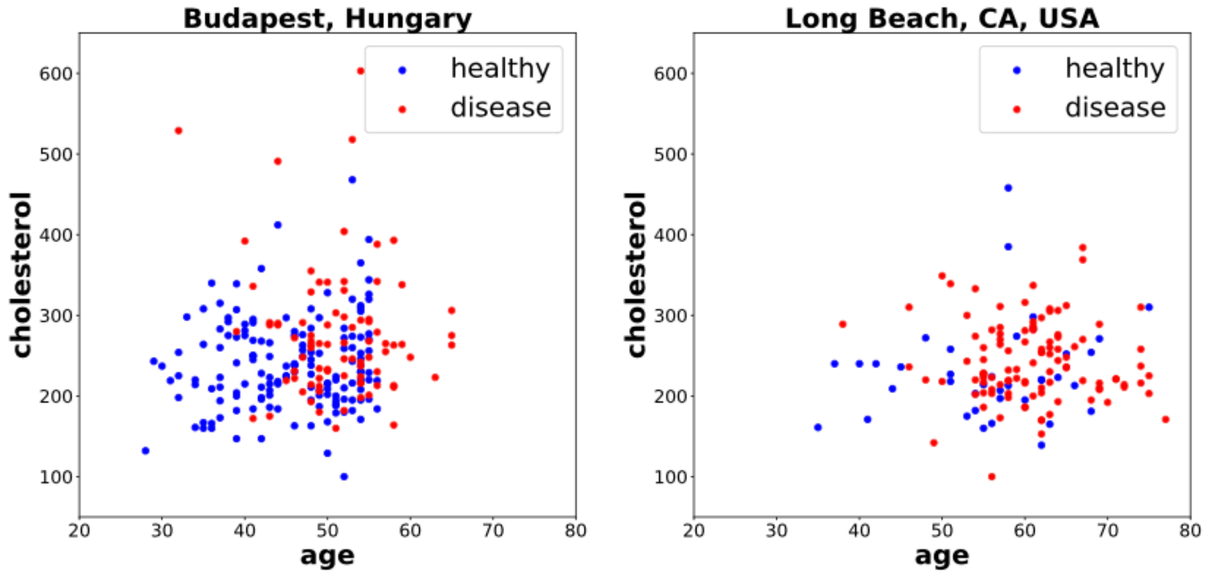
- Transfer learning:

  Model trained on a source domain or task and evaluated on a different but related target domain or task, where either the tasks or domains (or both) differ

- Domain Adaptation:

  Source & Target have the same tasks but differ in their domains.

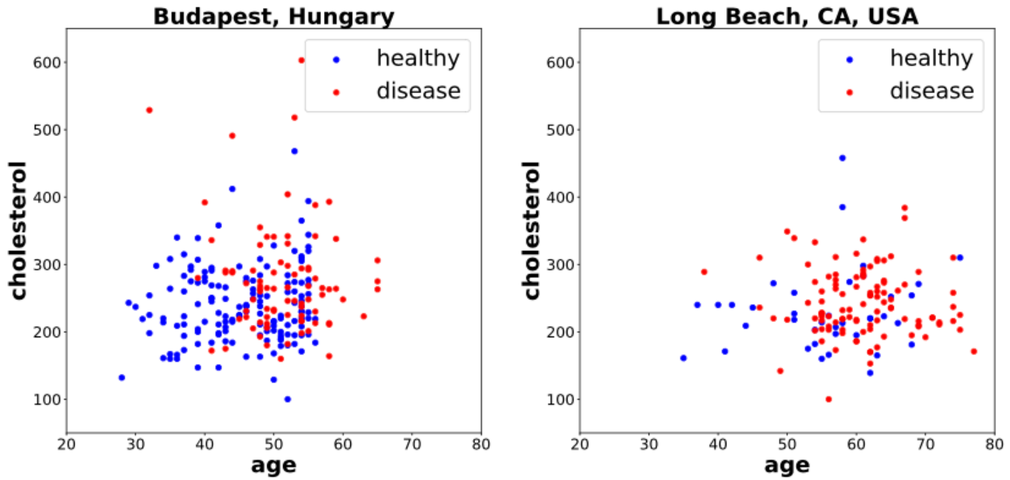
  e.g. can data collected in a Budapest hospital be used to train an intelligent diagnosis system for a hospital in Long Beach, CA?

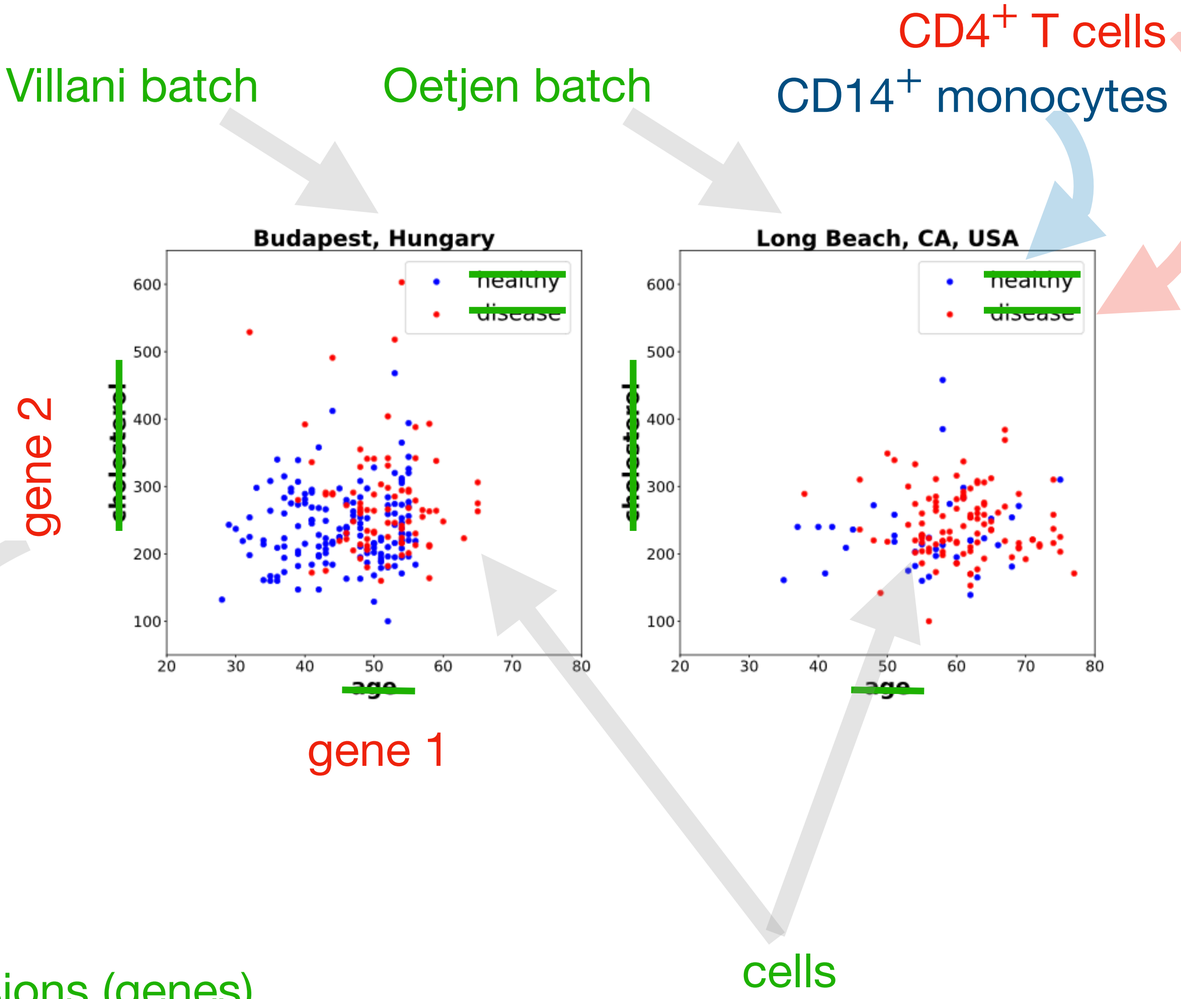


Kouw, Loog ‘21

Relevant Problem: Domain Adaptation

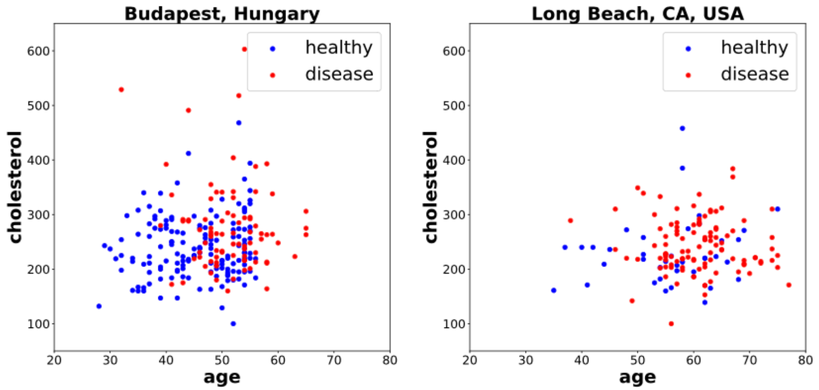
How does it Relate to scRNA-seq Data Integration?

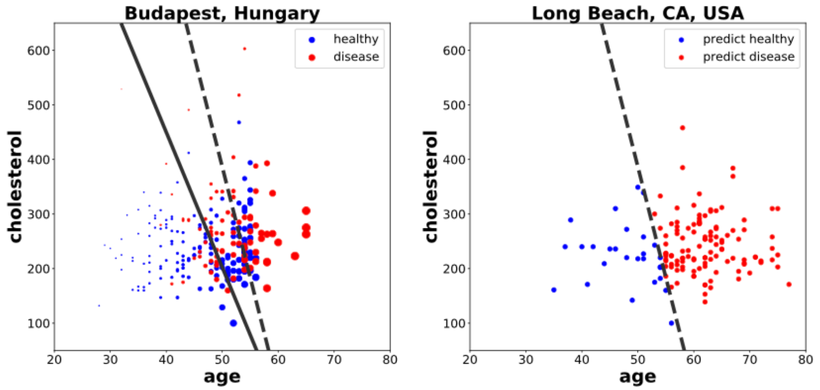




Relevant Problem: Domain Adaptation

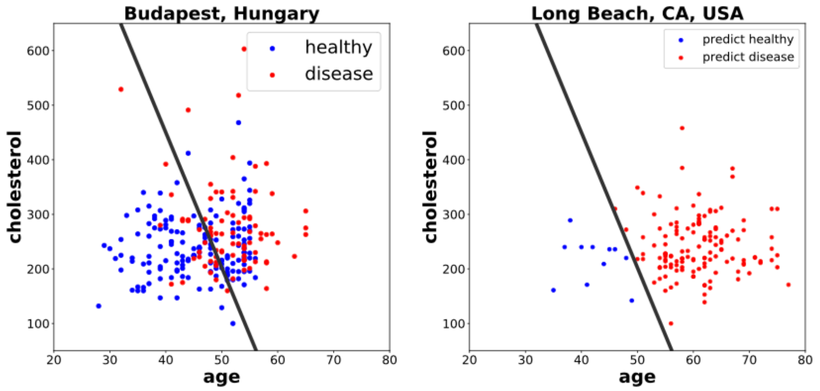
Some Techniques

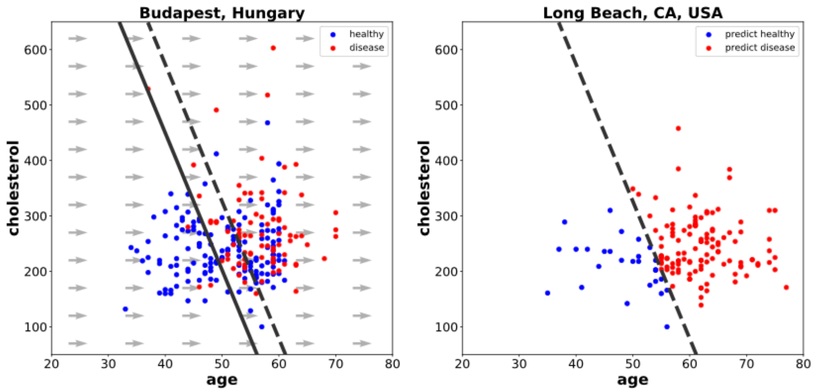




Seurat V4

Importance weighting





Many single-cell data integration methods

Linear classifier

Feature-based

Popular Methods for scRNA-seq Integration

- Many reviews including:
  Integration of Single-Cell RNA-Seq Datasets: A Review of Computational Methods, Ryu et al. ‘23
  Computational methods for the integrative analysis of single-cell data, Forcato et al. ‘21
  Integrative Cell Analysis, Stuart et al. ’19
- Many benchmarks including:
  Benchmarking atlas-level data integration in single-cell genomics, Luecken et al. ‘22
  A benchmark of batch-effect correction methods for single-cell RNA sequencing data, Tran et al. ‘20
  A multicenter study benchmarking single-cell RNA sequencing technologies using reference samples, Chen et al. ‘21
  A test metric for assessing single-cell RNA-seq batch correction, Buttner et al., ‘19
- Many different methods >50
- We talk about 5 of them:
  fastMNN, Seurat (V3), Scanorama, Harmony, scANVI

Seurat vs MNN

nature biotechnology

ANALYSIS

Integrating single-cell transcriptomic data across different conditions, technologies, and species

Andrew Butler[1,2], Paul Hoffman[1], Peter Smibert[1], Efthymia Papalexi[1,2] & Rahul Satija[1,2]

Received 4 July 2017; accepted 9 February 2018; published online 2 April 2018;
doi:10.1038/nbt.4096

NATURE BIOTECHNOLOGY VOLUME 36 NUMBER 5 MAY 2018

ANALYSIS

nature biotechnology

Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors

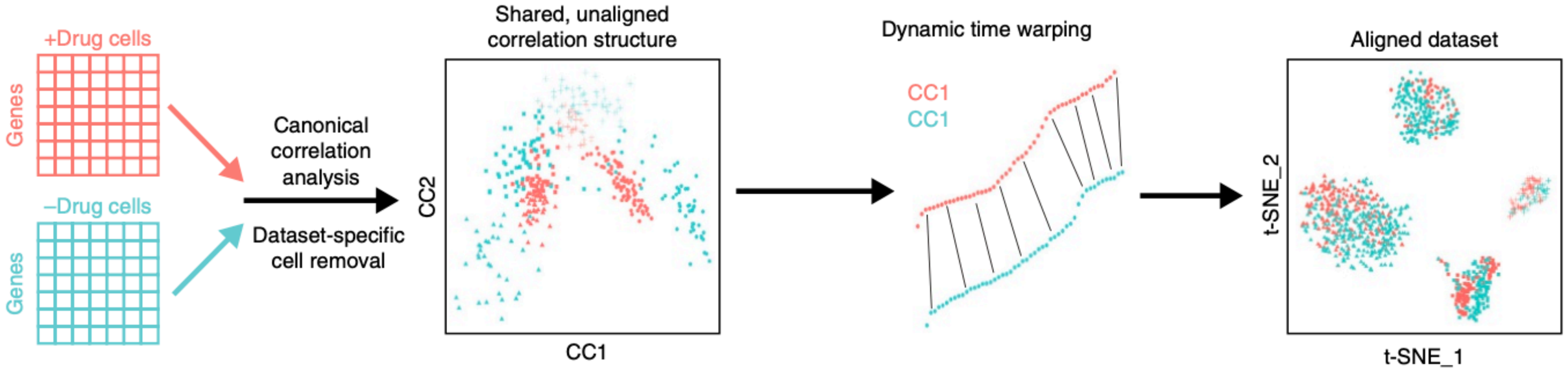
Laleh Haghverdi[1,2], Aaron T L Lun[3], Michael D Morgan[4] & John C Marioni[1,3,4]

Received 3 July 2017; accepted 1 February 2018; published online 2 April 2018;
doi:10.1038/nbt.4091

NATURE BIOTECHNOLOGY VOLUME 36 NUMBER 5 MAY 2018

Seurat

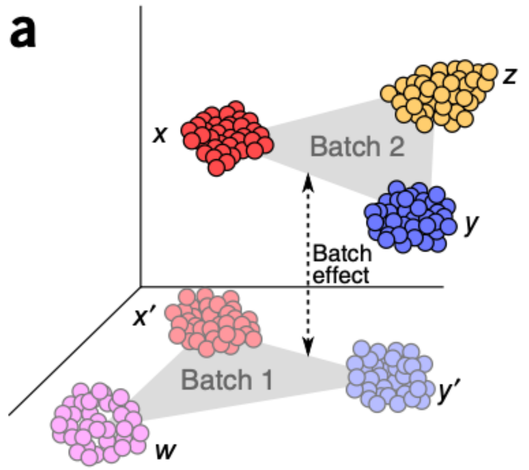
Overview

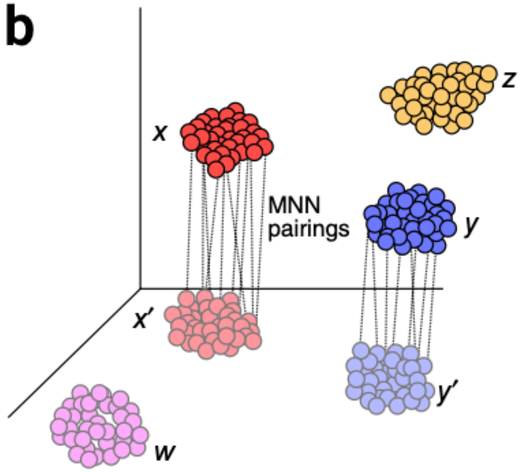


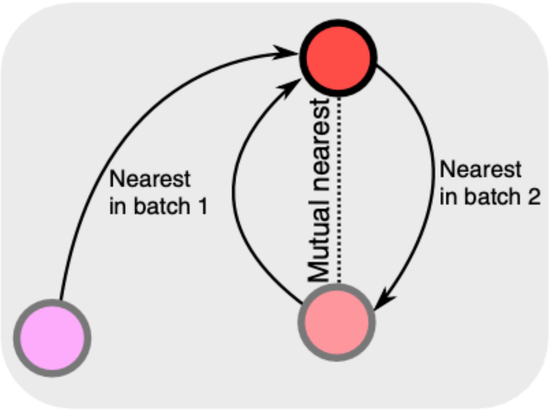
MNN

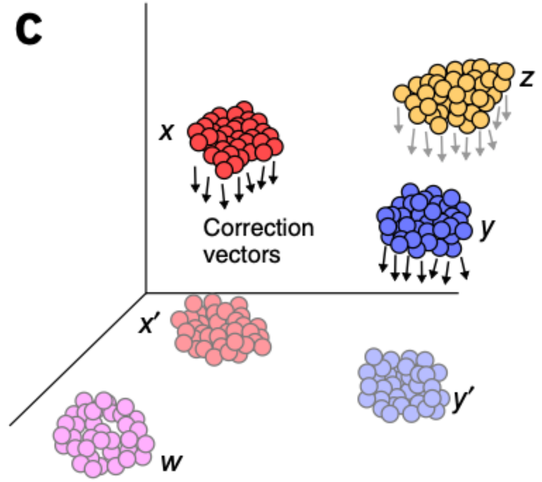
Overview

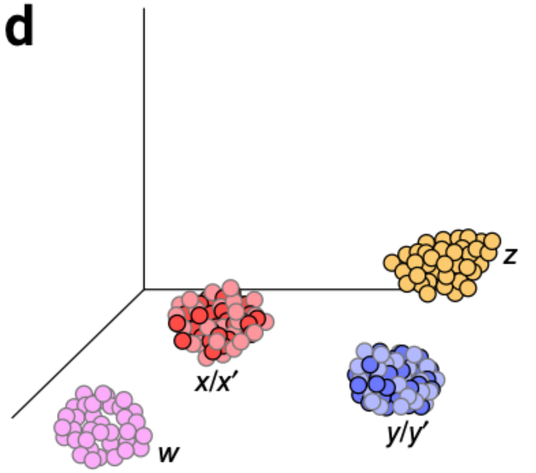
Differences between cells with mutually similar expression profiles in the high-dimensional gene expression space driven by batch effects

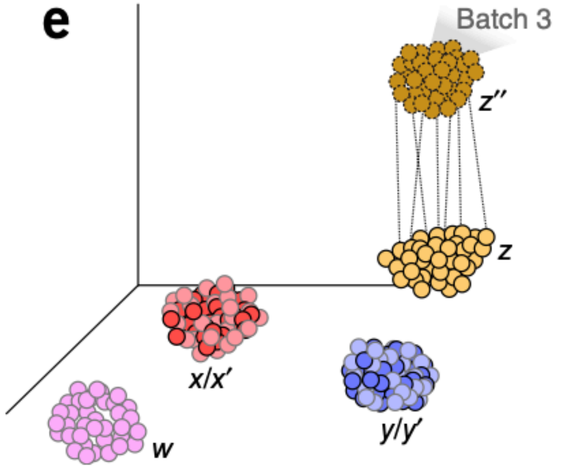












MNN

Main Steps

Brennecke et al.

INPUT: n batches $B_1, \cdots, B_n$ of correspondingly $N_1, \cdots, N_n$ cells

G_I inquiry genes, G_{HVG} highly variable genes.

OUTPUT: Batch corrected and integrated data C of $N_1 + \cdots + N_n$ cells and G_I

for each batch B_i **do**

Cosine normalize the expression data matrices

// B_1 is the first reference C

$C \leftarrow B_1$; $C_{HVG} \leftarrow B_{1,HVG}$; $C_I \leftarrow B_{1,I}$

for $i = 2$ to n **do**

// find MNN pairs between C and B_i, $k(c)^R$: k NN of c in batch R

for cells $l \in B_i, m \in C$,

l, m are MNN pairs if $l \in k(m)^{B_i}$ and $m \in k(l)^C$

for each cell $x \in B_i$ **do**

$\overrightarrow{u_I}(x) \leftarrow$ avg diff. of $\overrightarrow{x}, \overrightarrow{m}$, for m an MNN pair of x

$\overrightarrow{u_{HVG}}(x) \leftarrow$ avg diff. of $\overrightarrow{x}, \overrightarrow{m}$, for m an MNN pair of x

for each cell $x \in B_i$ **do**

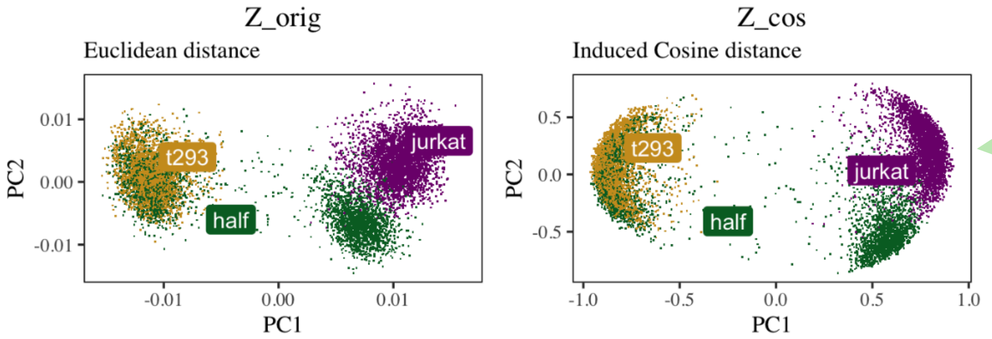
Gaussian kernel weighting $W_{HVG}(x, m)$, $\forall m$ in MNN of B_i

Smoothing on $\overrightarrow{u_I}(x)$, $\overrightarrow{u_{HVG}}(x)$

$x_I \leftarrow x_I - \overrightarrow{u_I}(x)$

$x_{HVG} \leftarrow x_{HVG} - \overrightarrow{u_{HVG}}(x)$

append $B'_{i,HVG}$ to C_{HVG} and $B'_{i,I}$ to C_I



$$Y_x \leftarrow \frac{Y_x}{||Y_x||}$$ (Y_x expression vector of cell x)

MNN

Main Steps

INPUT: n batches $B_1, \cdots, B_n$ of correspondingly $N_1, \cdots, N_n$ cells

G_I inquiry genes, G_{HVG} highly variable genes.

OUTPUT: Batch corrected and integrated data C of $N_1 + \cdots + N_n$ cells and G_I

for each batch B_i **do**

Cosine normalize the expression data matrices

// B_1 is the first reference C

$C \leftarrow B_1$; $C_{HVG} \leftarrow B_{1,HVG}$; $C_I \leftarrow B_{1,I}$

for $i = 2$ to n **do**

// find MNN pairs between C and B_i, $k(c)^R$: k NN of c in batch R

for cells $l \in B_i, m \in C$,

l, m are MNN pairs if $l \in k(m)^{B_i}$ and $m \in k(l)^C$

for each cell $x \in B_i$ **do**

$\overrightarrow{u_I}(x) \leftarrow$ avg diff. of $\overrightarrow{x}, \overrightarrow{m}$, for m an MNN pair of x

$\overrightarrow{u_{HVG}}(x) \leftarrow$ avg diff. of $\overrightarrow{x}, \overrightarrow{m}$, for m an MNN pair of x

for each cell $x \in B_i$ **do**

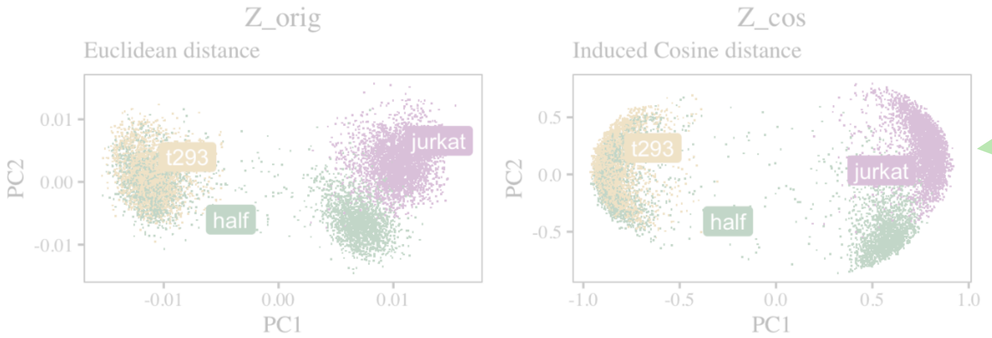
Gaussian kernel weighting $W_{HVG}(x, m)$, $\forall m$ in MNN of B_i

Smoothing on $\overrightarrow{u_I}(x)$, $\overrightarrow{u_{HVG}}(x)$

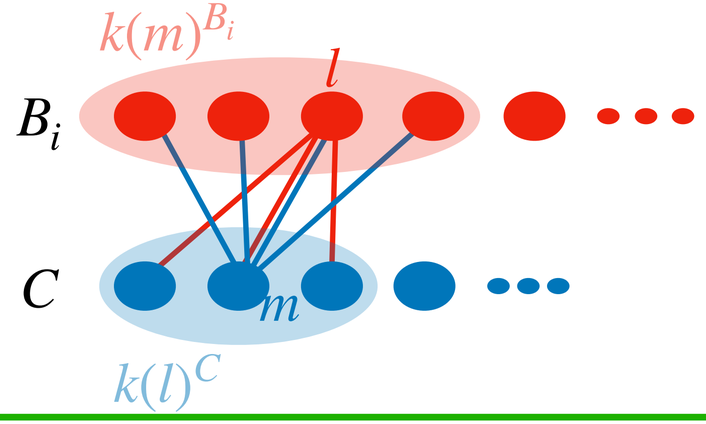
$x_I \leftarrow x_I - \overrightarrow{u_I}(x)$

$x_{HVG} \leftarrow x_{HVG} - \overrightarrow{u_{HVG}}(x)$

append $B'_{i,HVG}$ to C_{HVG} and $B'_{i,I}$ to C_I



$$Y_x \leftarrow \frac{Y_x}{||Y_x||} \quad (Y_x \text{ expression vector of cell } x)$$



Parameters:

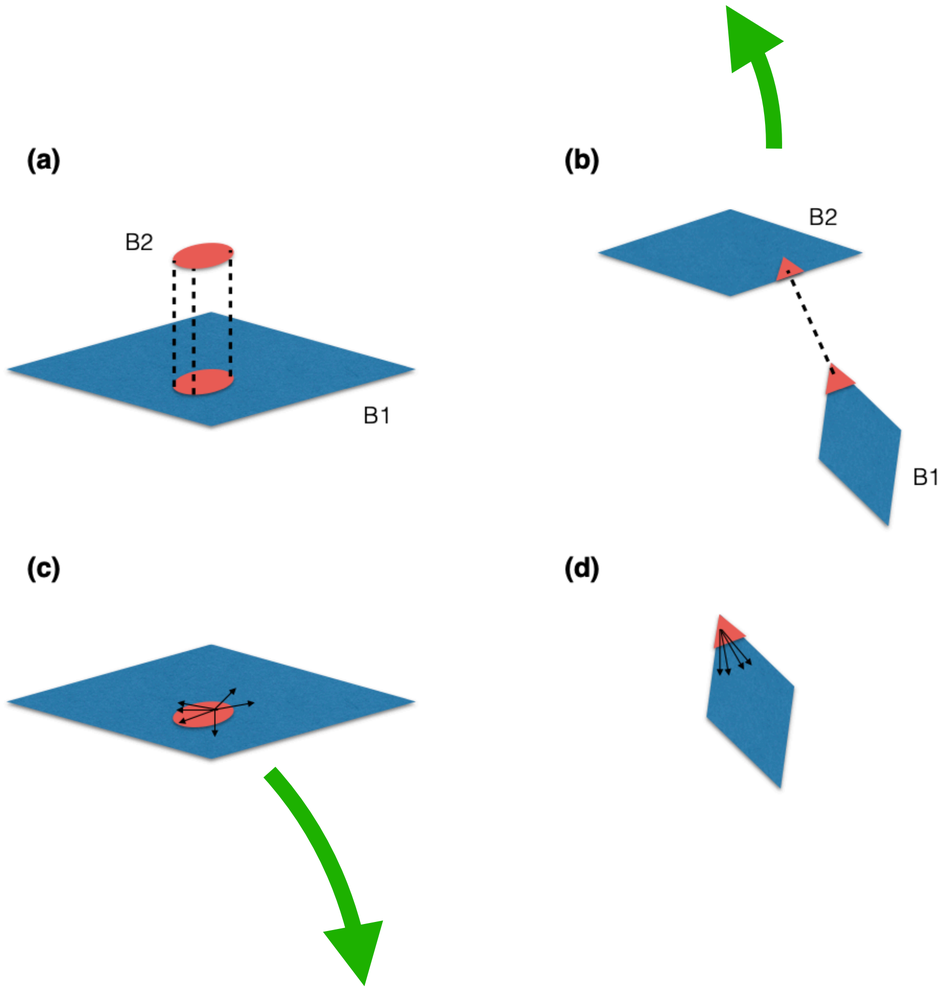
kernel width σ^2=1: the extent of smoothing

k=20

MNN

- Assumptions:
  - at least one cell population present in both batches (no point in merging the batches)

MNN pairs at the edges
(Easy to detect)

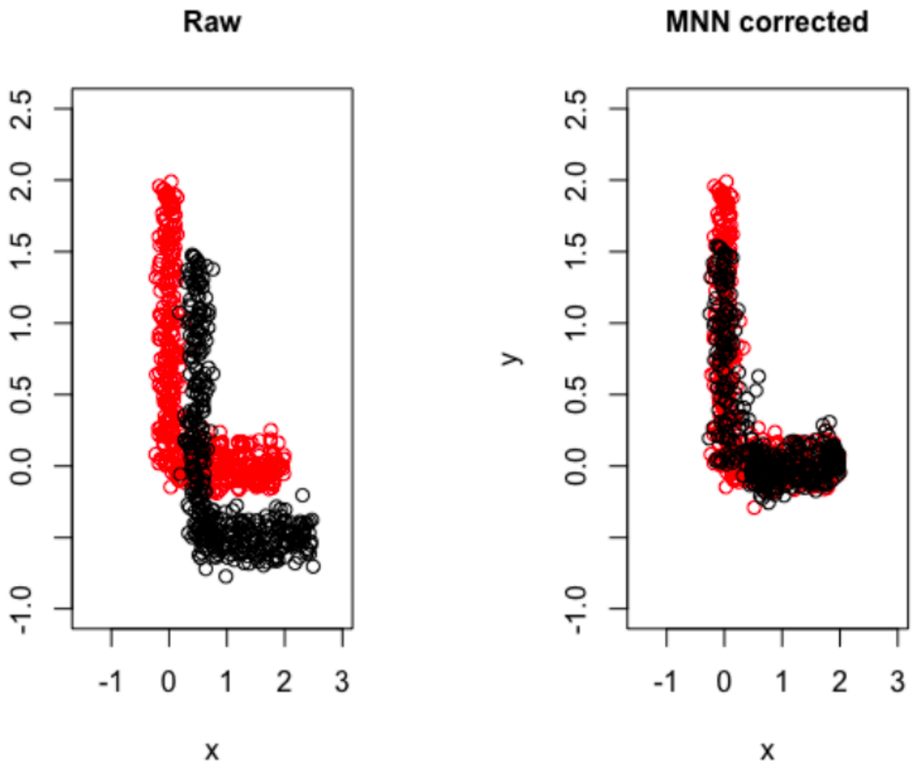


MNN pairs in the middle of the batch

MNN

- Assumptions:
  - batch effect almost orthogonal to the biological subspace

shift between the red and black batches
in the same plane as the L shape of each batch



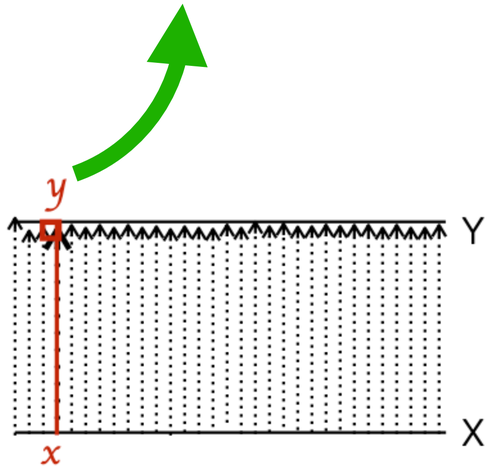
MNN

- Assumptions:
  - variation in batch effects across cells

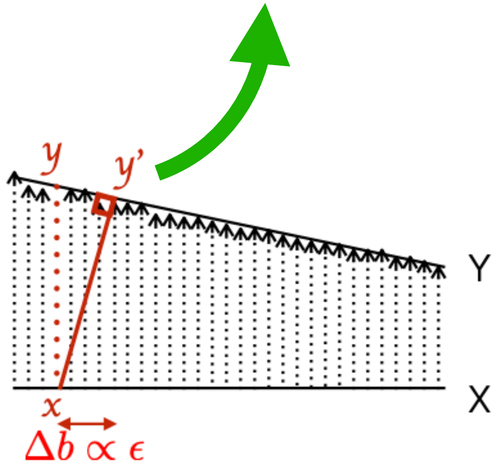
    $\ll$

    variation in biological effects between different cell types

True match found via MNN



False match found via MNN in the presence of batch effect variation



Seurat vs MNN

- Seurat:
  - Dimensionality reduction (CCA)
  - No concept to relate individual pairs of cells
  - Outputs integration in low-dimensions
- MNN:
  - No dimensionality reduction
  - Relates individual pairs of cells with the concept of MNN
  - Outputs integration & batch correction in original dimensions

fastMNN

Main Steps

for cell-level comparisons,
e.g., clustering and visualization.

# of PCs: 50 (default)

INPUT: n batches $B_1, \cdots, B_n$ of correspondingly $N_1, \cdots, N_n$ cells
G_I inquiry genes, G_{HVG} highly variable genes.

OUTPUT: Batch corrected and integrated data C of $N_1 + \cdots + N_n$ cells and G_I
(in low-dimensional representation)

- Multi-sample PCA on the (cosine-)normalized expression values (wrt HVG) to reduce dimensionality.
- Identify MNN pairs in the low-dimensional space between a reference batch and a target batch.

fastMNN

Main Steps

for cell-level comparisons,
e.g., clustering and visualization.

# of PCs: 50 (default)

INPUT: n batches $B_1, \cdots, B_n$ of correspondingly $N_1, \cdots, N_n$ cells
G_I inquiry genes, G_{HVG} highly variable genes.

OUTPUT: Batch corrected and integrated data C of $N_1 + \cdots + N_n$ cells and G_I
(in low-dimensional representation)

- Multi-sample PCA on the (cosine-)normalized expression values (wrt HVG) to reduce dimensionality.
- Identify MNN pairs in the low-dimensional space between a reference batch and a target batch.
- Remove variation along the average batch vector in both reference and target batches.

fastMNN

Main Steps

for cell-level comparisons,
e.g., clustering and visualization.

# of PCs: 50 (default)

INPUT: n batches $B_1, \cdots, B_n$ of correspondingly $N_1, \cdots, N_n$ cells
G_I inquiry genes, G_{HVG} highly variable genes.

OUTPUT: Batch corrected and integrated data C of $N_1 + \cdots + N_n$ cells and G_I
(in low-dimensional representation)

- Multi-sample PCA on the (cosine-)normalized expression values (wrt HVG) to reduce dimensionality.
- Identify MNN pairs in the low-dimensional space between a reference batch and a target batch.
- Remove variation along the average batch vector in both reference and target batches.

  correction vector of $l \in B_i$: average across l's MNNs
  average batch vector (estimate of overall batch effect): average over all correction vectors
  - project all cells in B_i onto average batch vector
  - adjust cell coordinates to the mean value within B_i

  Not batch correction! Simply remove variation within each batch to:
  - avoid the "kissing" problem: MNNs just identified on the surface of each subpopulation

fastMNN

Main Steps

for cell-level comparisons,
e.g., clustering and visualization.

# of PCs: 50 (default)

INPUT: n batches $B_1, \cdots, B_n$ of correspondingly $N_1, \cdots, N_n$ cells
G_I inquiry genes, G_{HVG} highly variable genes.

OUTPUT: Batch corrected and integrated data C of $N_1 + \cdots + N_n$ cells and G_I
(in low-dimensional representation)

- Multi-sample PCA on the (cosine-)normalized expression values (wrt HVG) to reduce dimensionality.
- Identify MNN pairs in the low-dimensional space between a reference batch and a target batch.
- Remove variation along the average batch vector in both reference and target batches.

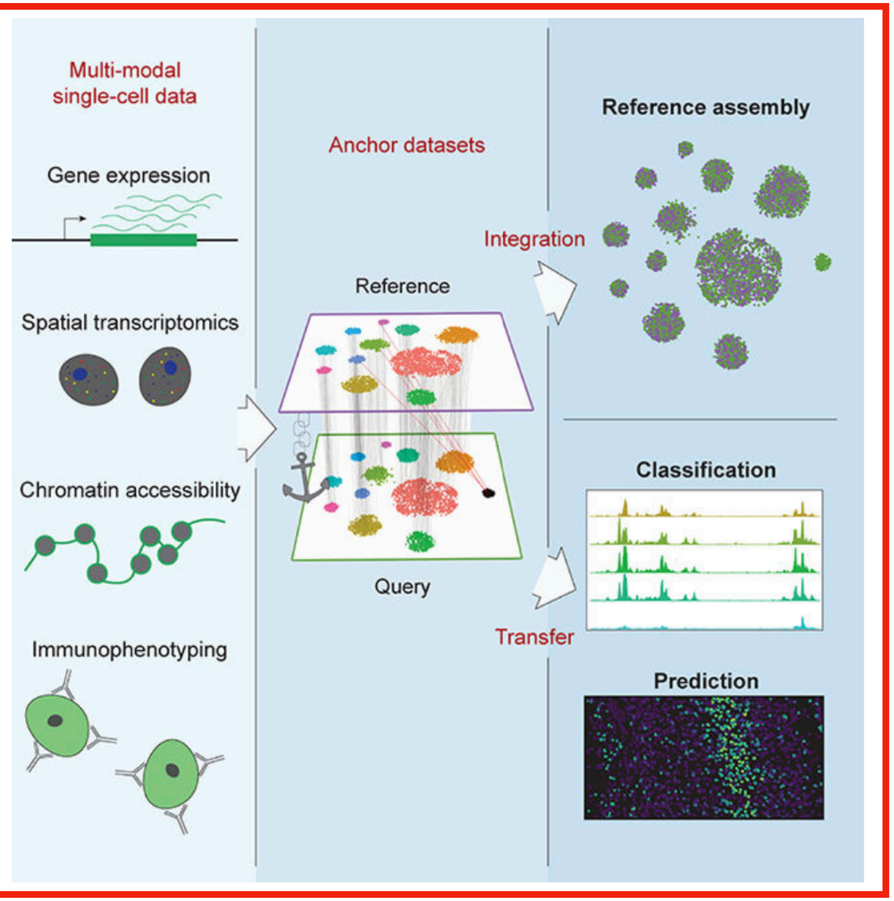
  correction vector of $l \in B_i$: average across l's MNNs
  average batch vector (estimate of overall batch effect): average over all correction vectors
  - project all cells in B_i onto average batch vector
  - adjust cell coordinates to the mean value within B_i

  Not batch correction! Simply remove variation within each batch to:
  - avoid the "kissing" problem: MNNs just identified on the surface of each subpopulation

- Correct the target batch towards the reference with locally weighted correction vectors
  (Merge the corrected target batch with the reference, repeat with new target batch if multi-batch)

Seurat V3

Overview

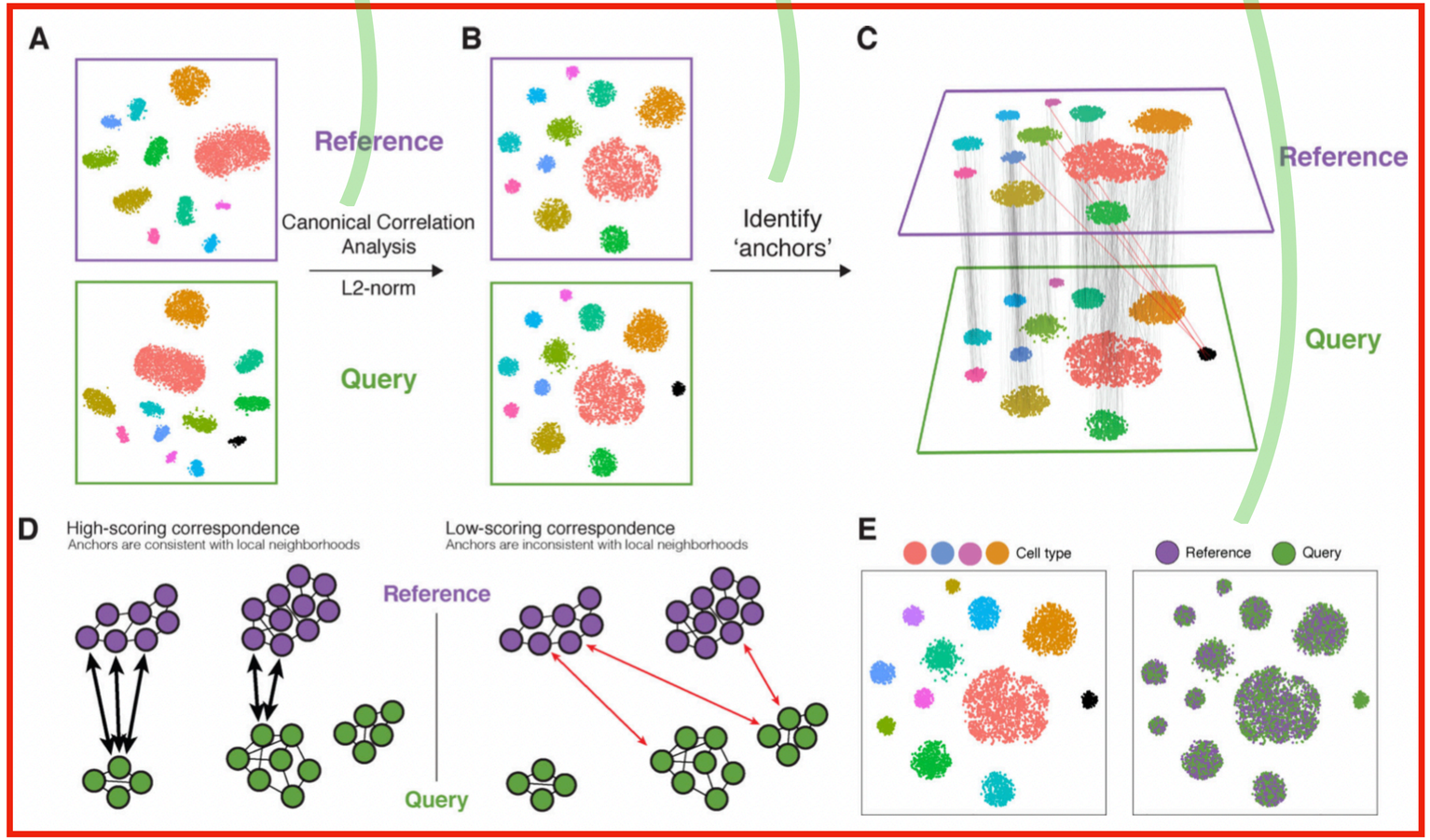


New Motivation: Multiple modalities

From Seurat (original)

Inspired by MNN

Transforms in original dims



Overview

Seurat V3

Main Steps

- Normalization

log-normalization with a size factor of 10000
z-score transform: expression values for each gene across all cells

Seurat V3

Main Steps

- Normalization
- Feature Selection

log-normalization with a size factor of 10000
z-score transform: expression values for each gene across all cells

- mean and variance of each gene using the unnormalized data (i.e. UMI or counts matrix) & log-transform
- fit a curve to predict the σ_i as a function of $\overline{x_i}$, for each gene i
- $z_{ij} = \frac{x_{ij} - \overline{x_i}}{\sigma_i}$
- for each gene i find variance of z_{ij} across all cells j
- use variance to rank & select top 2000 highly variable genes

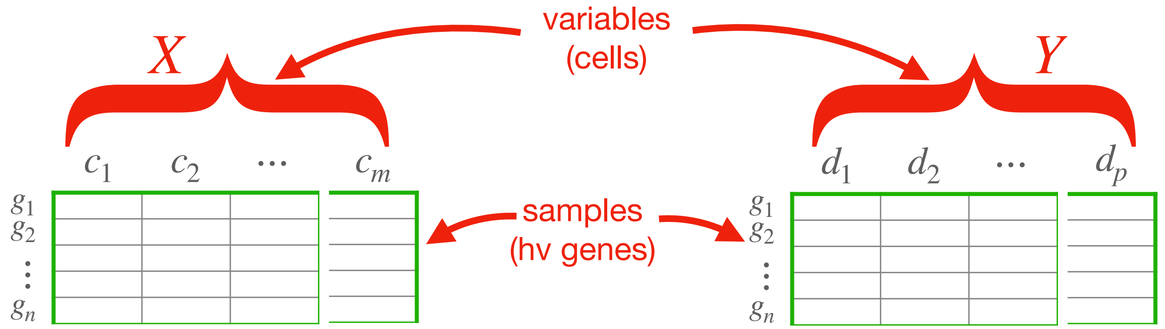
Rank hvgs of all datasets with respect to
the number of datasets it appears as hvg & collect top 2000

Seurat V3

Main Steps

- Normalization
- Feature Selection
- Datasets in shared low-dimensional space: CCA

Idea: PCA may identify sources of variation even if only present in one dataset, but CCA emphasizes shared variation across datasets.



- find linear combination of the x variables and that of the y variables that are most highly correlated.
- find other linear combinations of the x's and y's with high correlations
- each pair of combinations mutually uncorrelated with the rest except for their “partner” combination!

Seurat V3

Main Steps

- Normalization
- Feature Selection
- Datasets in shared low-dimensional space: CCA
- Construct MNN with k-NN of each cell within paired dataset

- Termed ‘anchors’
- In L2 normalized shared low-dimensional embedding
- k=5, as compared to 20 in MNN

Seurat V3

Main Steps

- Normalization
- Feature Selection
- Datasets in shared low-dimensional space: CCA
- Construct MNN with k-NN of each cell within paired dataset
- Filter out & score remaining anchors

- Termed ‘anchors’
- In L2 normalized shared low-dimensional embedding
- k=5, as compared to 20 in MNN

- find closest neighbors of each query cell c_i in reference Y (k.filter=200, done in L2 normalized original space using top max.features=200 genes with strongest association with the CCV)
- if paired anchor reference cell not in closest, remove anchor

Seurat V3

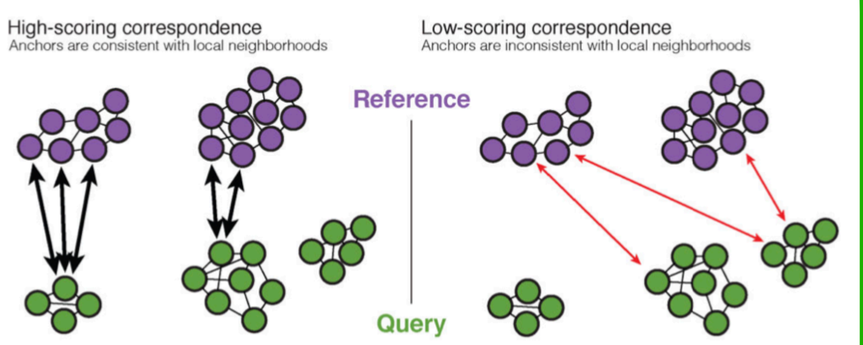
Main Steps

- Normalization
- Feature Selection
- Datasets in shared low-dimensional space: CCA
- Construct MNN with k-NN of each cell within paired dataset
- Filter out & score remaining anchors

- Termed ‘anchors’
- In L2 normalized shared low-dimensional embedding
- k=5, as compared to 20 in MNN

- find closest neighbors of each query cell c_i in reference (k.filter=200, done in L2 normalized original space using top max.features=200 genes with strongest association with the CCV)
- if paired anchor reference cell not in closest, remove anchor

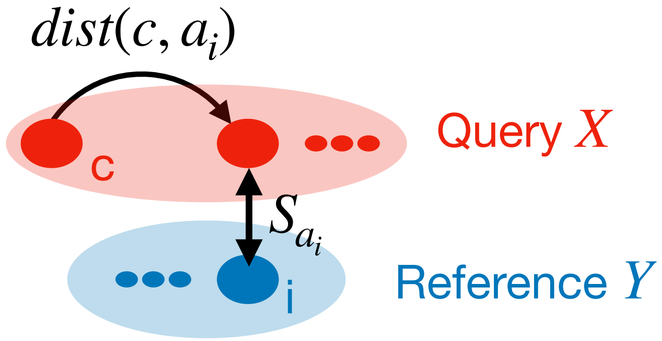
- SNN graph (k.score=30 closest neighbors; within-dataset, between-dataset 4 graphs combined
- anchor-score: shared neighbor overlap



Seurat V3

Main Steps

- Normalization
- Feature Selection
- Datasets in shared low-dimensional space: CCA
- Construct MNN with k-NN of each cell within paired dataset
- Filter out & score remaining anchors
- Anchor weights & final correction

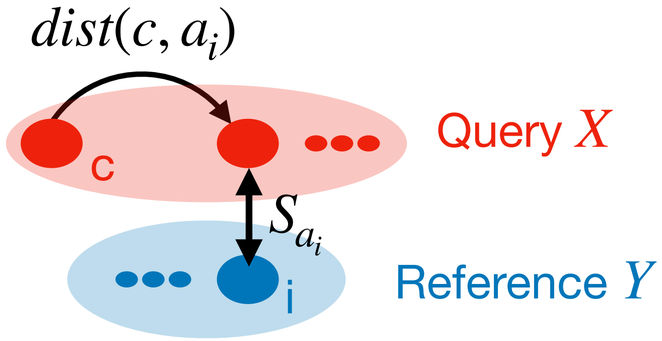


- in a way, extending the effects of the anchors
- for each query cell c_i in X,
  find k.weight=100 nearest anchors cells in X (in PCA space)
  compute $D_{c,i} = (1 - \frac{dist(c, a_i)}{dist(c, a_{k.weight})})S_{a_i}$
- Gaussian kernel on $D_{c,i}$ (bandwidth sd set to 1) gives $\tilde{D}_{c,i}$
- normalizing $\tilde{D}_{c,i}$ gives $W_{c,i}$ with dimensions $|X| \times |Y|$

Seurat V3

Main Steps

- Normalization
- Feature Selection
- Datasets in shared low-dimensional space: CCA
- Construct MNN with k-NN of each cell within paired dataset
- Filter out & score remaining anchors
- Anchor weights & final correction

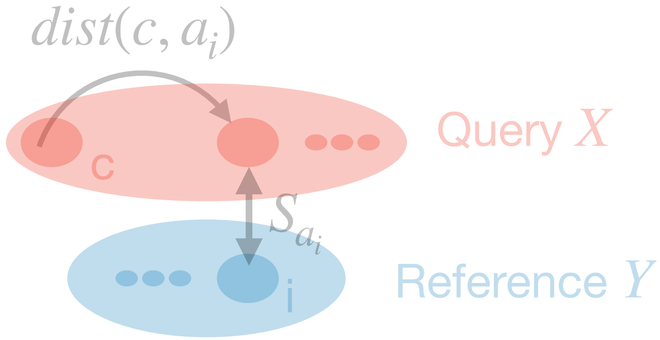


- in a way, extending the effects of the anchors
- for each query cell c_i in X,
  find k.weight=100 nearest anchors cells in X (in PCA space)
  compute $D_{c,i} = (1 - \frac{dist(c, a_i)}{dist(c, a_{k.weight})})S_{a_i}$
- Gaussian kernel on $D_{c,i}$ (bandwidth sd set to 1) gives $\tilde{D}_{c,i}$
- normalizing $\tilde{D}_{c,i}$ gives $W_{c,i}$ with dimensions $|X| \times |Y|$

- Similar to MNN, only weighted, that is,
  difference between expression of a query cell c and anchor i is multiplied with the weight, the resulting transformation matrix is subtracted from the original expression matrix

Seurat V3

Main Steps



- Normalization
- Feature Selection
- Datasets in shared low-dimensional space: CCA
- Construct MNN with k-NN of each cell within paired dataset
- Filter out & score remaining anchors
- Anchor weights & final correction
- Extension to multiple batches

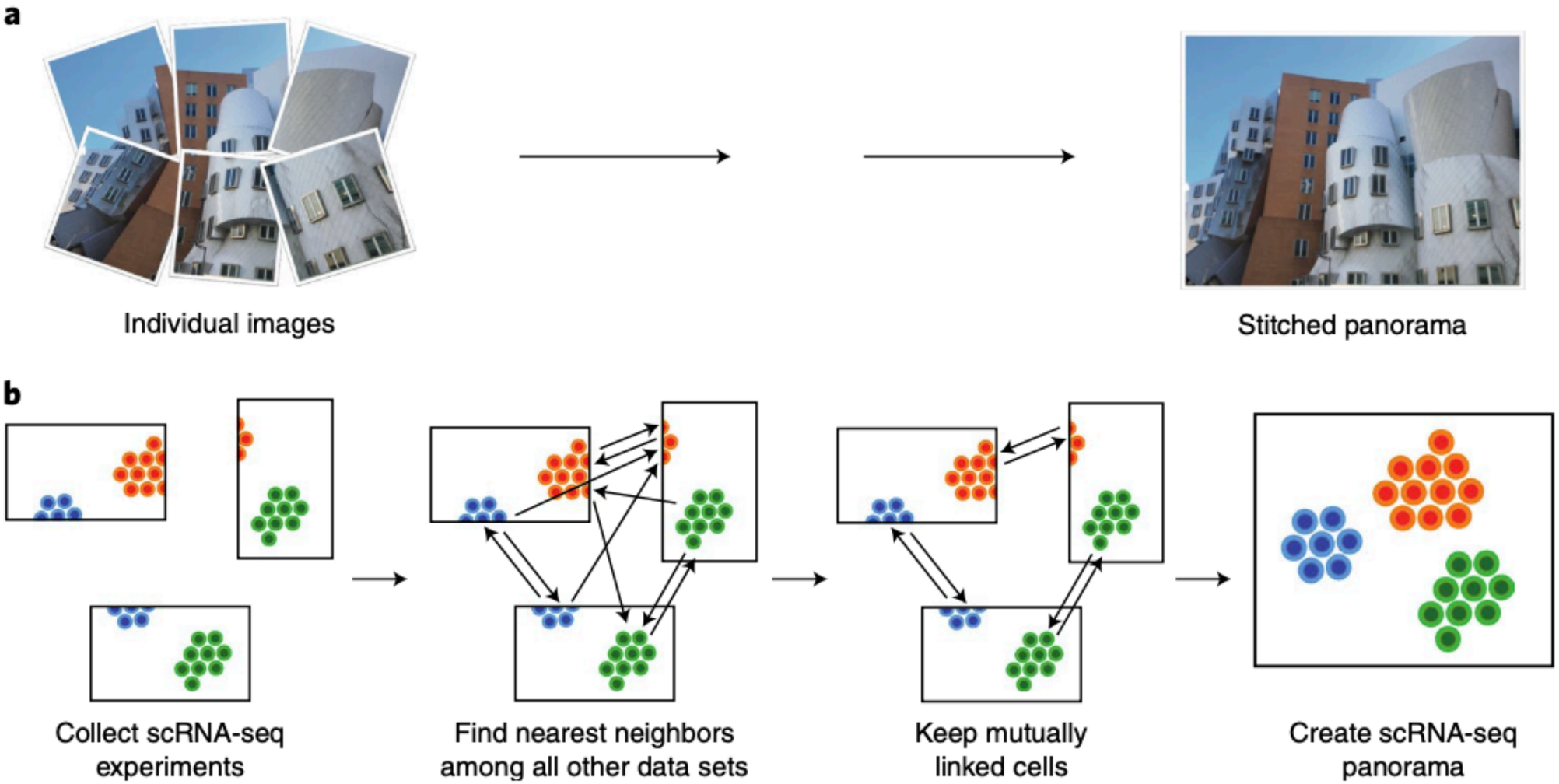
- in a way, extending the effects of the anchors
- for each query cell c_i in X,
  find k.weight=100 nearest anchors cells in X (in PCA space)
  compute $D_{c,i} = (1 - \frac{dist(c, a_i)}{dist(c, a_{k.weight})})S_{a_i}$
- Gaussian kernel on $D_{c,i}$ (bandwidth sd set to 1) gives $\tilde{D}_{c,i}$
- normalizing $\tilde{D}_{c,i}$ gives $W_{c,i}$ with dimensions $|X| \times |Y|$

- Similar to MNN, only weighted, that is,
  difference between expression of a query cell c and anchor i is multiplied with the weight, the resulting transformation matrix is subtracted from the original expression matrix

- Hirerarchical clustering:
  $dist(X, Y) = min(|X||Y|)/|Anchors(X, Y)|$

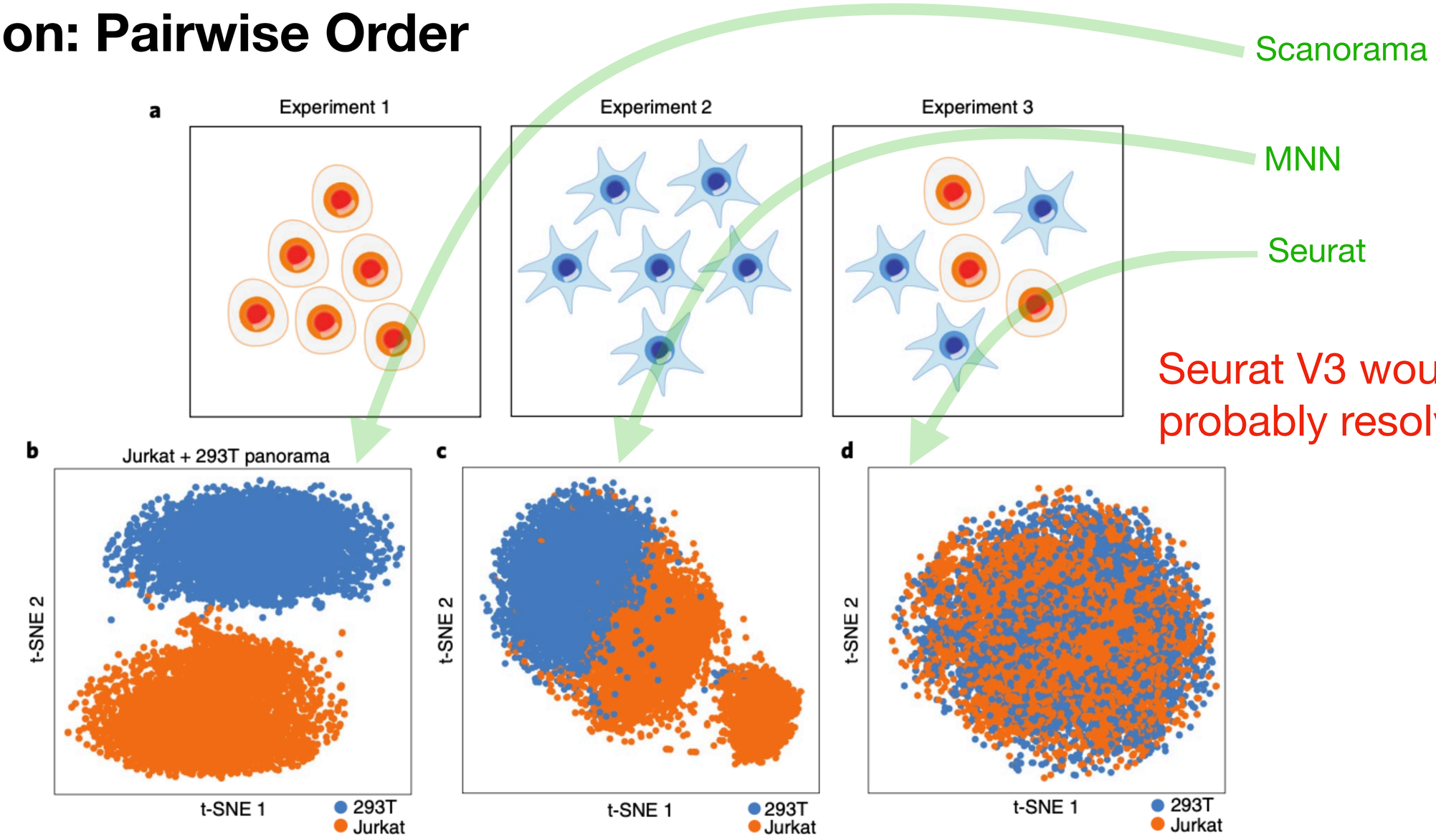
Scanorama

Overview



Scanorama

Motivation: Pairwise Order



Scanorama

Main Steps

- Preprocessing & Normalization

- remove low-quality cells: < 600 identified genes
- only consider genes present in all datasets
- l_2-normalize expression values for each cell

Scanorama

Main Steps

- Preprocessing & Normalization
- Dimensionality reduction: Randomized SVD

- remove low-quality cells: < 600 identified genes
- only consider genes present in all datasets
- l_2-normalize expression values for each cell

- an approximate SVD is employed on the ‘merged’ gene-by-cell expression matrix
- k=reduced dimension=100

Scanorama

Main Steps

- Preprocessing & Normalization
- Dimensionality reduction: Randomized SVD
- Find MNNs

- remove low-quality cells: < 600 identified genes
- only consider genes present in all datasets
- l_2-normalize expression values for each cell

- an approximate SVD is employed on the ‘merged’ gene-by-cell expression matrix
- k=reduced dimension=100

- find k-NN of cell $c \in D_i$ against all cells in all datasets
- repeat for all datasets D_i
- k=20
- due to prohibitive running time: approximation via locality sensitive hashing with random hyperplanes
- MNN definition is the same as before:
  M_{ij} denotes MNN pairs between D_i, D_j

Scanorama

Main Steps

- Preprocessing & Normalization
- Dimensionality reduction: Randomized SVD
- Find MNNs
- Score pairs of datasets

- dataset panorama graph G:
  For dataset pair D_i, D_j, we have $(i, j) \in E(G)$
  with weight r_{ij} iff $r_{ij} = max(n_i^M / n_i, n_j^M / n_j) \geq \alpha$

Scanorama

Main Steps

- Preprocessing & Normalization
- Dimensionality reduction: Randomized SVD
- Find MNNs
- Score pairs of datasets

# of cells in D_i with MNNs in D_j

10% of all weights

- dataset panorama graph G:
  For dataset pair D_i, D_j, we have $(i, j) \in E(G)$
  with weight r_{ij} iff $r_{ij} = max(n_i^M / n_i, n_j^M / n_j) \geq \alpha$
- Panoramas $\leftrightarrow$ connected components of G

Scanorama

Main Steps

- Preprocessing & Normalization
- Dimensionality reduction: Randomized SVD
- Find MNNs
- Score pairs of datasets
- Successively merge panoramas in the order of edge weights

# of cells in D_i with MNNs in D_j

10% of all weights

- dataset panorama graph G:
  For dataset pair D_i, D_j, we have $(i, j) \in E(G)$
  with weight r_{ij} iff $r_{ij} = max(n_i^M / n_i, n_j^M / n_j) \geq \alpha$
- Panoramas $\leftrightarrow$ connected components of G

- Merging: Similar to MNN method
- for D_i, D_j let E_i^{match}, E_j^{match} denote cells in MNNs of M_{ij}
- for any cell a in D_i and matched cell b in D_i

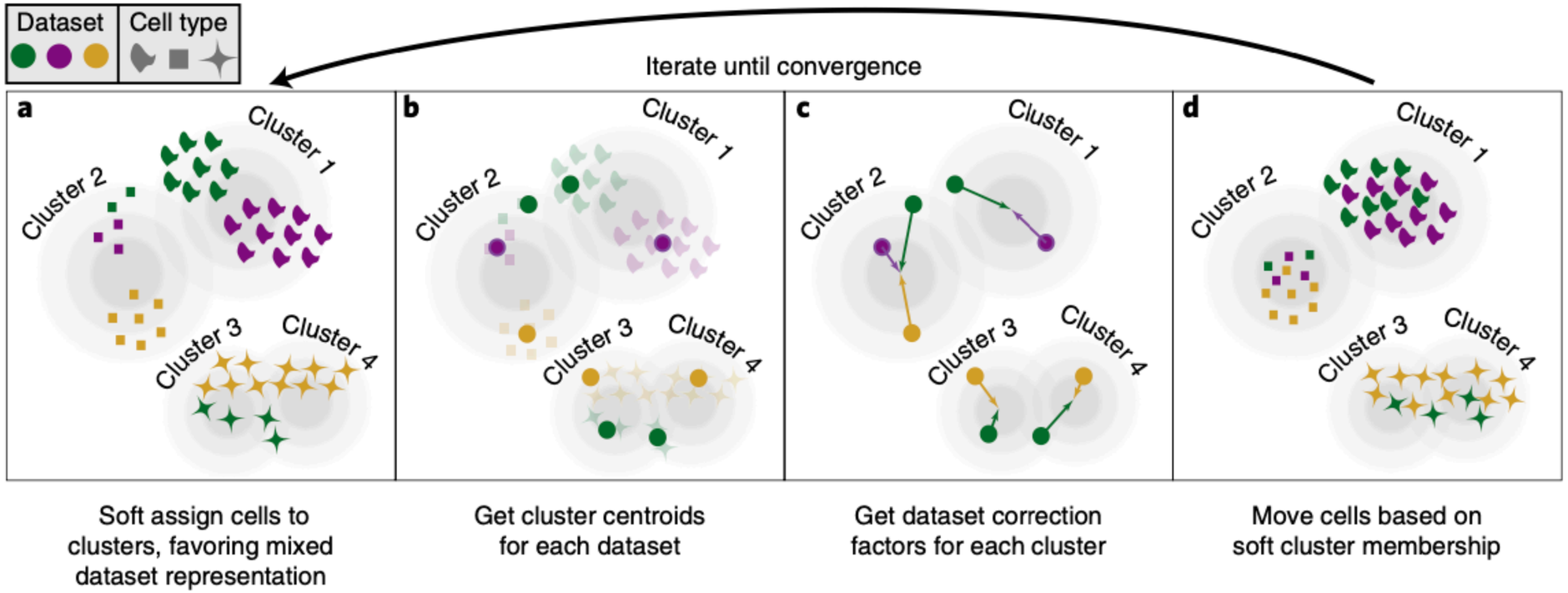
$$[\Gamma_i]_{ab} = \exp\left(-\frac{\sigma}{2}\left\|[E_i]_{a,:} - \left[E_i^{match}\right]_{b,:}\right\|_2^2\right)$$

  where $[\Gamma_i]_{ab}$ denotes Gaussian smoothed weight
- v_a the translation vector of a is a linear combination of the matching vectors where the Gaussian kernel up-weights the matching vectors closest to a:

$$v_a = \frac{[\Gamma_i]_{a,:}\left(E_j^{match} - E_i^{match}\right)}{\sum_{b \in [|M_{ij}|]} [\Gamma_i]_{a,b}}$$

Harmony

Overview



Fuzzy clustering

Global + dataset-specific centroids per cluster

Correction factors based on centroids

Correction: linear combination of dataset correction factors weighted by the soft-cluster assignments

Harmony

Main Steps

Algorithm 1 Harmony

function HARMONIZE (Z, ϕ)

$\hat{Z} \leftarrow Z$

repeat

$R \leftarrow$ CLUSTER $(\hat{Z}, \phi)$

$\hat{Z} \leftarrow$ CORRECT (Z, R, ϕ)

until convergence

return $\hat{Z}$

PCA embedding of cells

batch assignments of cells

correction is a linear model of the original embedding
(overcorrection could arise if previous iteration's $\hat{Z}$ is used)

output in low-dimensional embedding

Harmony

Main Steps

Algorithm 1 Harmony

function HARMONIZE (Z, ϕ)

$\hat{Z} \leftarrow Z$

repeat

$R \leftarrow$ CLUSTER $(\hat{Z}, \phi)$

$\hat{Z} \leftarrow$ CORRECT (Z, R, ϕ)

until convergence

return $\hat{Z}$

$$\min_{R,Y} \sum_{i,k} R_{ki} \|Z_i - Y_k\|^2$$

$$\text{s.t.} \forall_i \forall_k \, R_{ki} \in \{0, 1\}$$

Ordinary k-means

$$\min_{R,Y} \sum_{i,k} R_{ki} \|Z_i - Y_k\|^2 + \sigma R_{ki} \log R_{ki}$$

$$\text{s.t.} \, \forall_i \forall_k \, R_{ki} > 0, \forall_i \sum_{k=1}^{K} R_{ki} = 1$$

Soft k-means with entropy regularization

$$\min_{R,Y} \sum_{i,k} R_{ki} \|Z_i - Y_k\|^2 + \sigma R_{ki} \log R_{ki} + \sigma \theta R_{ki} \log \left(\frac{O_{ki}}{E_{ki}} \right) \phi_i$$

$$\text{s.t.} \, \forall_i \forall_k \, R_{ki} > 0, \forall_i \sum_{k=1}^{K} R_{ki} = 1$$

Maximum diversity clustering

Mixture of experts:
Conditioned on experts (clusters) the model assumes linear relationship between response and independent variables

Penalize dependence between batch identity & cluster asignment

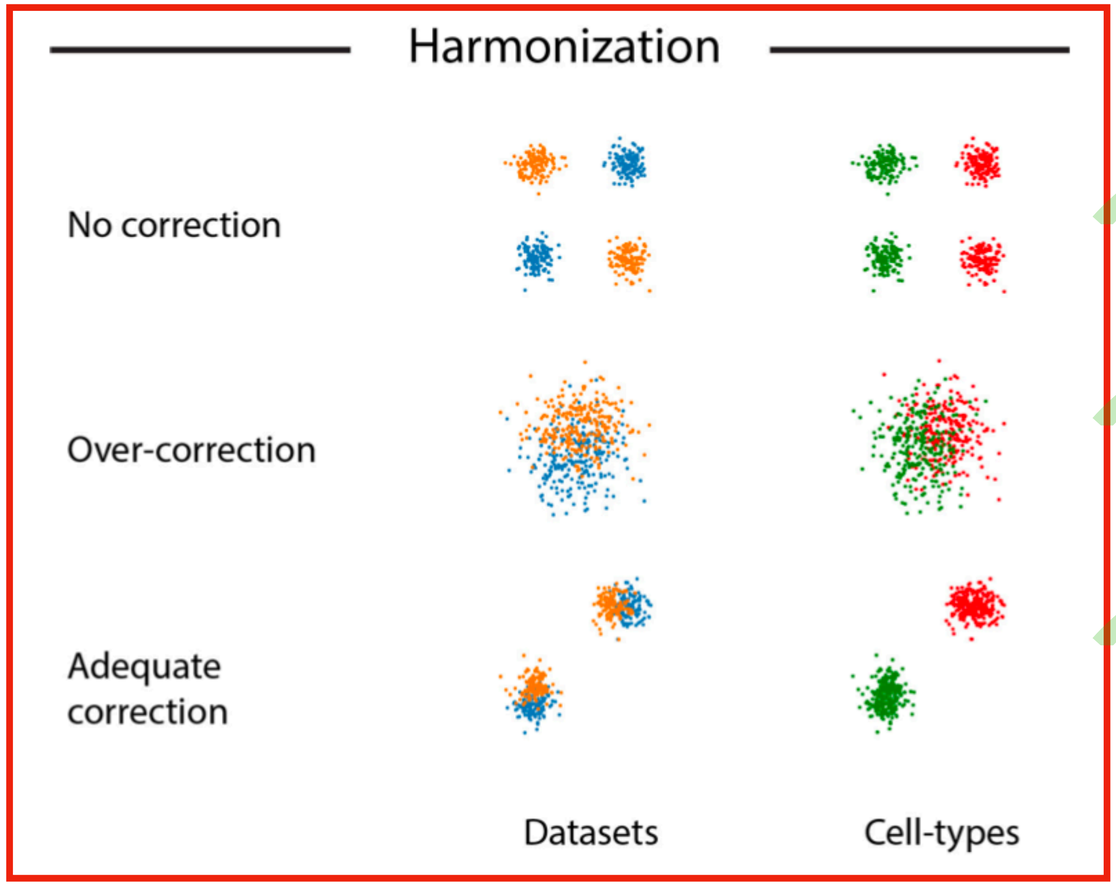
O_{kb}: observed co-occurence counts of cluster k, batch b

E_{kb}: expected counts under independence

ScANVI

Motivation Emphasis

two objectives:
(1) mixing the two datasets well and
(2) retaining the original structure in each dataset



(1) not satisfied, (2) satisfied

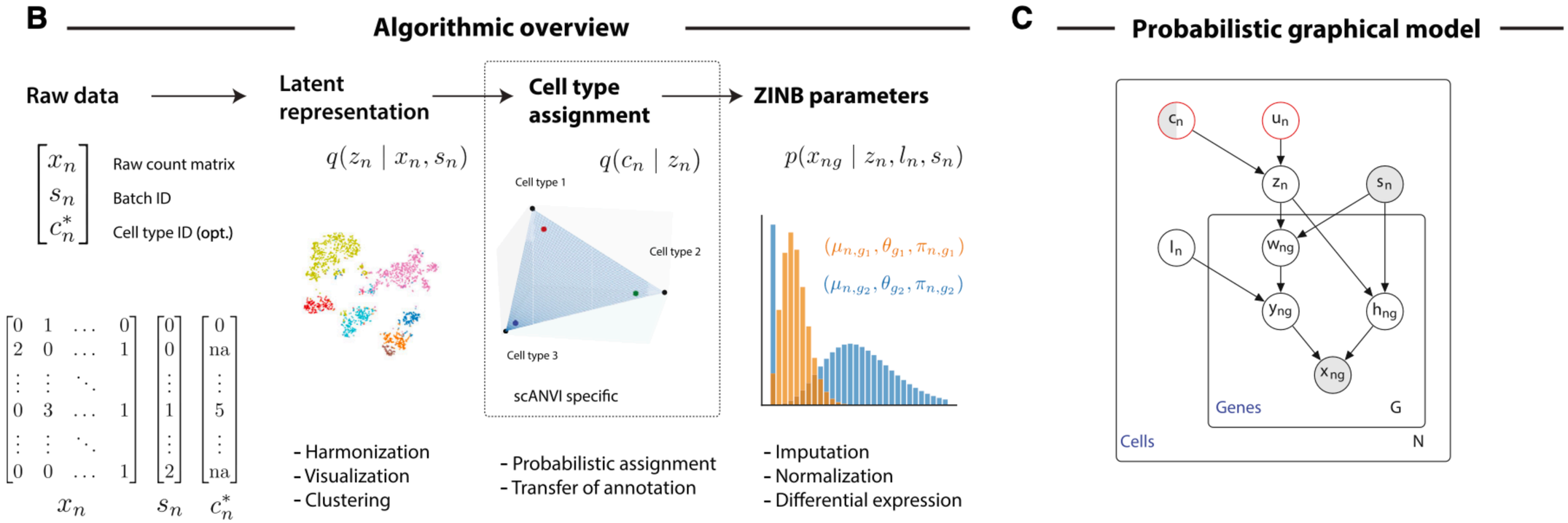
(2) not satisfied, (1) satisfied

Both satisfied

Available annotations may guide the inference of an informative latent representation.

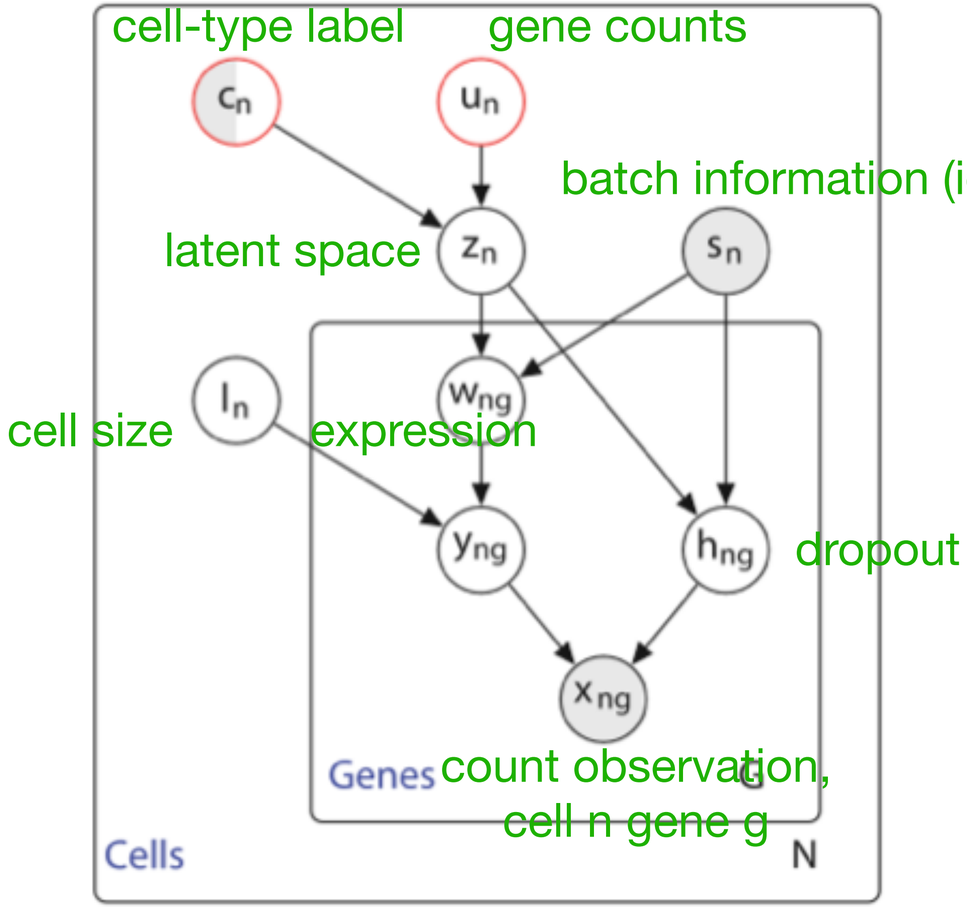
ScANVI

Overview



ScANVI

Main Idea



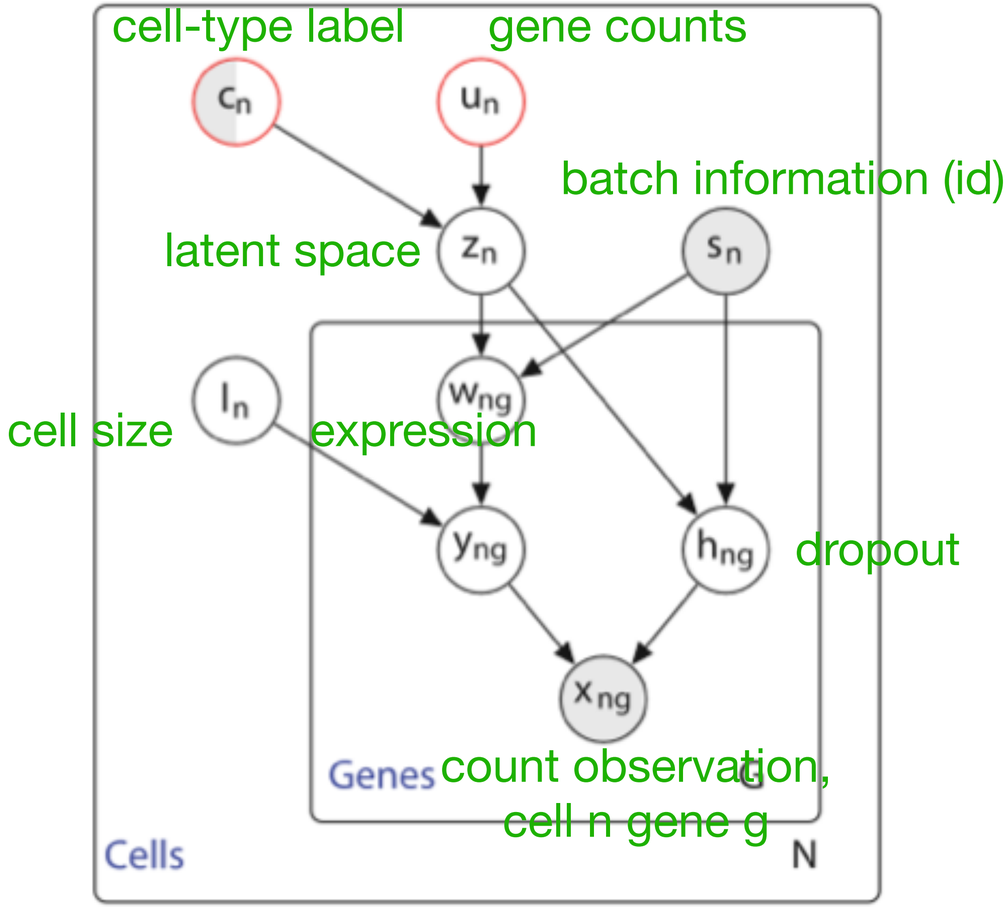
factorization for the inference model

$$q_\eta(z_n, \ell_n, u_n, c_n \mid x_n) = q_\eta(z_n \mid x_n) q_\eta(\ell_n \mid x_n) q_\eta(c_n \mid z_n) q_\eta(u_n \mid c_n, z_n)$$

variational distributions parameterized by neural networks

ScANVI

Main Idea



Provides output in low-dimensional embedding, but similar model can be executed for analysis such as differential gene expression.

factorization for the inference model

$$q_\eta(z_n, \ell_n, u_n, c_n \mid x_n) = q_\eta(z_n \mid x_n) q_\eta(\ell_n \mid x_n) q_\eta(c_n \mid z_n) q_\eta(u_n \mid c_n, z_n)$$

variational distributions parameterized by neural networks

Thank You