Quoi de neuf pour l'analyse des données scRNAseq ?

Cathy Maugis-Rabusseau

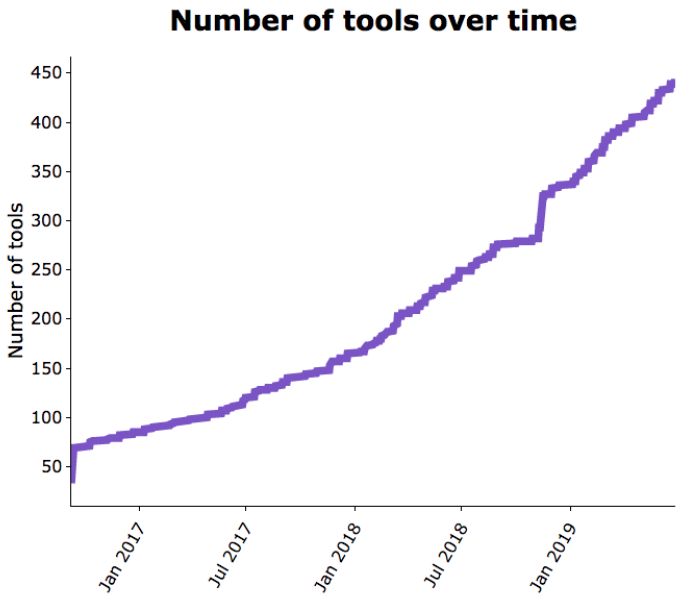
cathy.maugis@insa-toulouse.fr

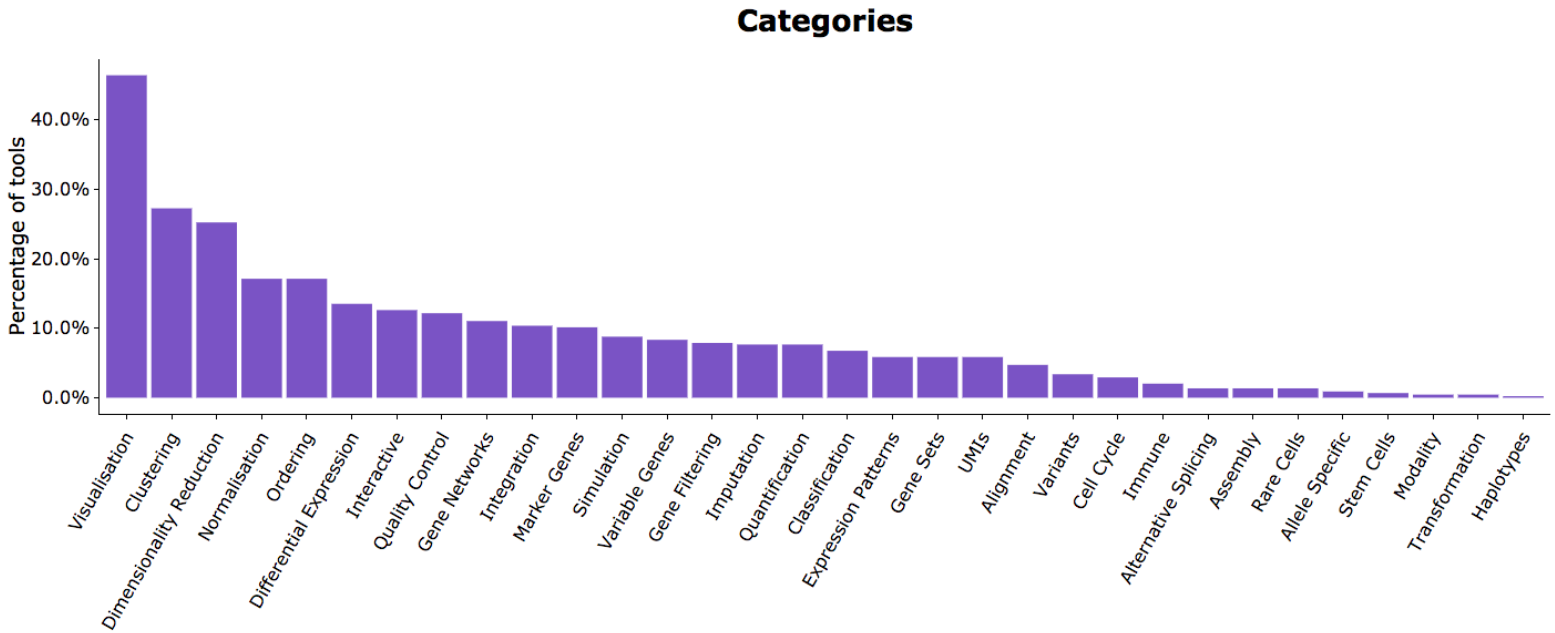
INSTITUT
de MATHEMATIQUES
de TOULOUSE

INSA
TOULOUSE

Plan

Evolution des packages pour scRNAseq





Nouveautés

- CellRanger 3.0 pour traiter les données issues de SingleCell 3' v3
- Seurat : nouvelle version depuis avril 2019
  - Amélioration des méthodes d'analyse
  - Amélioration de la normalisation (sctransform)
  - Modification de la structure de l'objet Seurat
- Scedar: package Python pour l'étude de données scRNA-seq [Zhang and Taylor, 2019]
- De nombreux packages sortent régulièrement pour l'analyse de données scRNAseq. On peut suivre les nouveautés dans la rubrique "Updates" de scRNA-tools. https://www.scrna-tools.org/updates

Plan

Correction effet batch

- Correction par Mutual Nearest Neighbors [ex: package **scran**]
- Correction par Canonical Correlation Analysis [ex: **Seurat v3**]
- Zinb-Wave : Effet batch pris en compte dans la modélisation GLM
- Autres méthodes / packages : SMNN, batchelor, mbkmeans, scBatch, ...
- Vignette générale Bioconductor sur la correction de l'effet batch

Correcting batch effects in single-cell RNA-seq data

Aaron T. L. Lun[1] ***and Michael D. Morgan***[2]

[1]Cancer Research UK Cambridge Institute, Li Ka Shing Centre, Robinson Way, Cambridge CB2 0RE, United Kingdom
[2]Wellcome Trust Sanger Institute, Wellcome Genome Campus, Hinxton, Cambridge CB10 1SA, United Kingdom

2019-05-03

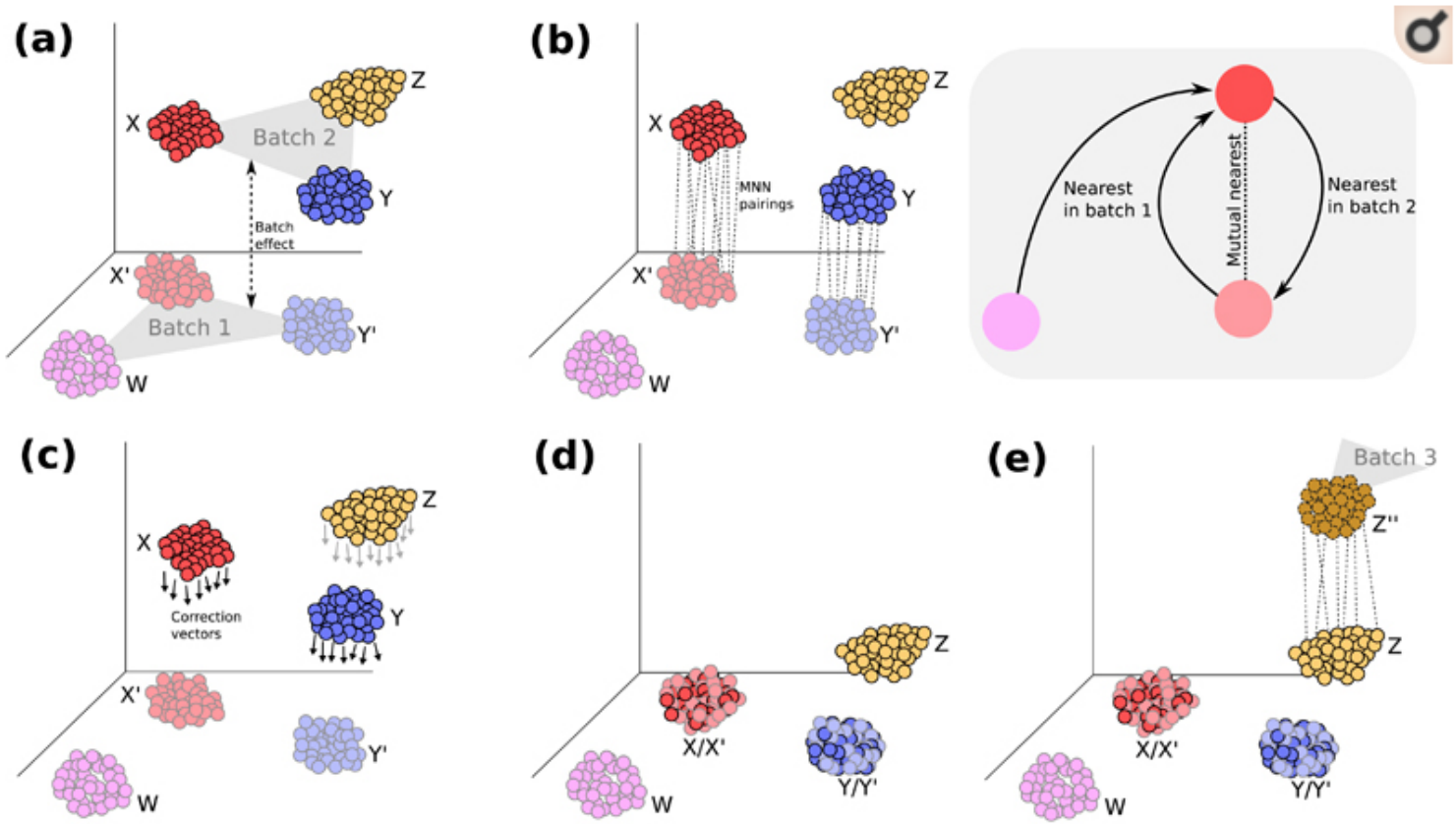
Mutual Nearest Neighbors (MNN) [Haghverdi et al., 2018]

- Transformation des données :

$$\tilde{X}_c = \frac{X_c}{\|X_c\|} \text{ où } X_c = \text{vecteur d'expression de la cellule } c$$

- Calcul de la distance euclidienne entre les $\tilde{X}_c$ des deux lots : $\|\tilde{X}_c - \tilde{X}_{c'}\|_2$
- Identification des MNN
- Calcul d'un facteur de correction de l'effet batch basé sur le vecteur des différences des profils et noyau gaussien
- Hyp : l'effet batch doit être "orthogonal" à l'effet biologique

Mutual Nearest Neighbors (MNN) [Haghverdi et al., 2018]



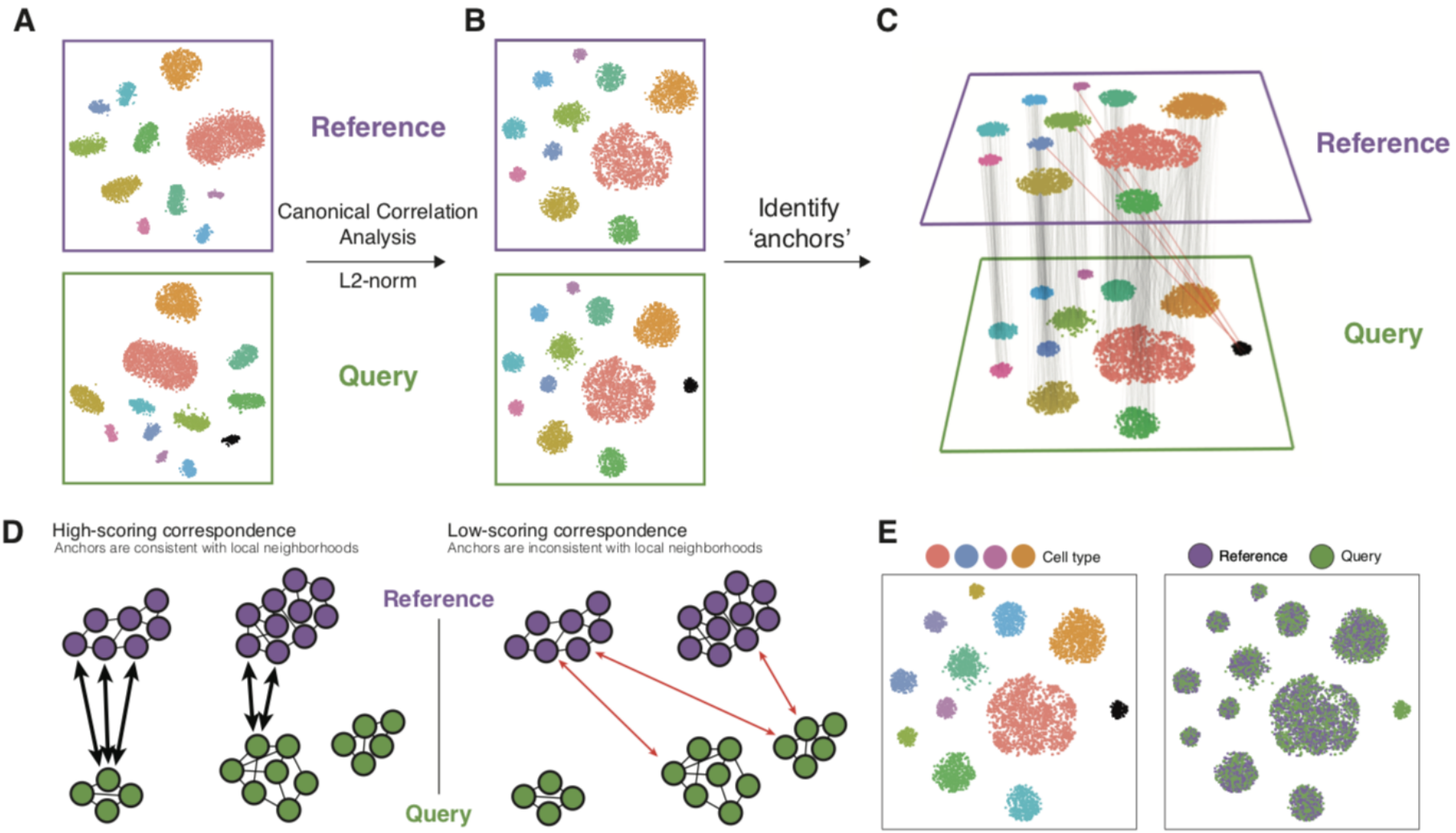
Canonical Correlation Analysis (CCA) [Stuart et al., 2019]

- Données : $X_{gc\ell}$ = mesure d'expression du gène g pour la cellule c du batch $\ell \in \{1, 2\}$ (Reference and query)
- Réduction de dimension des matrices $X_{..\ell}$ par CCA et normalisation L2 des vecteurs de CCA :

$$\underset{u,v}{argmax}\ u^T X_{..1}^T X_{..2} v \text{ st } \|X_{..1} u\|_2 \leq 1, \|X_{..2} v\|_2 \leq 1$$

- Détermination de MNN entre les deux batchs dans le sous-espace commun $\Rightarrow$ "anchors"
- Correction des "anchors" par score de correspondance
- Anchors + scores $\Rightarrow$ Correction des données du "query"

Canonical Correlation Analysis (CCA) [Stuart et al., 2019]



Zinb-Wave [Risso et al., 2017]

$$f_{ZINB}(x|\mu,\theta,\pi) = \pi\delta_0(x) + (1-\pi)f_{NB}(x|\mu,\theta)$$

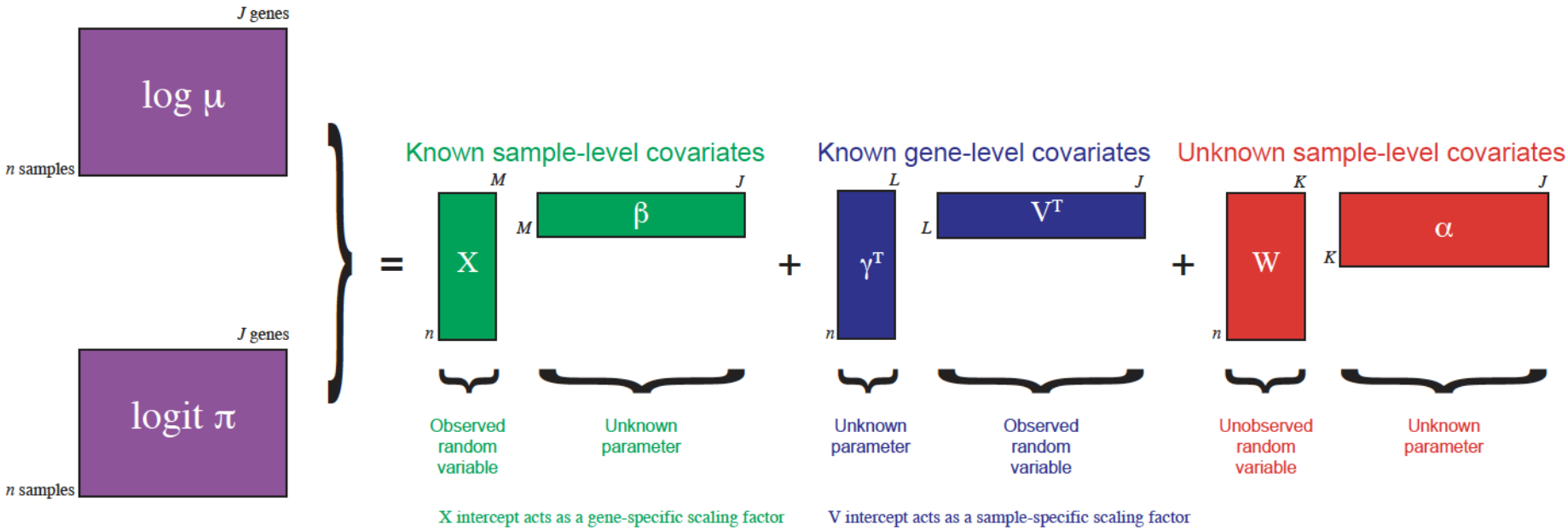
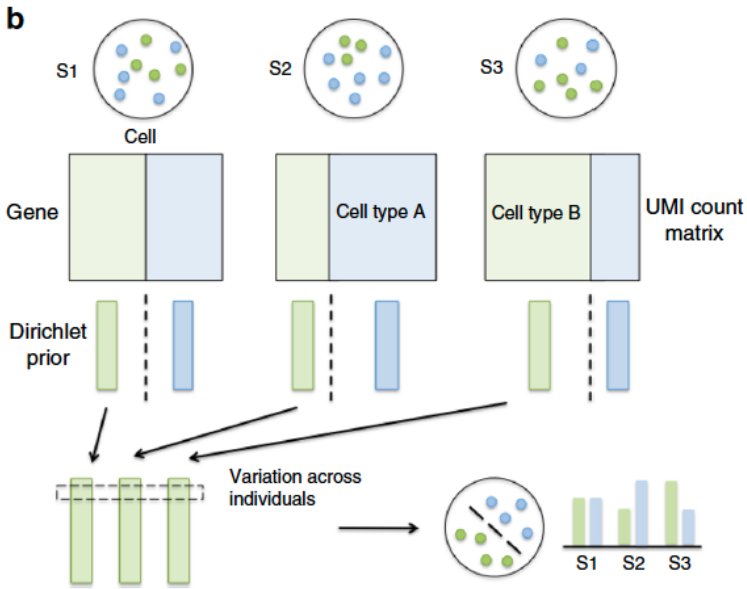


Figure 1: *Schematic view of the ZINB-WaVE model.* Given n cells and J genes, let Y_{ij} denote the count of gene j ($j = 1, \ldots, J$) for cell i ($i = 1, \ldots, n$) and Z_{ij} an unobserved indicator variable, equal to one if gene j is a dropout in cell i and zero otherwise. Then, $\mu_{ij} = E[Y_{ij}|Z_{ij} = 0, X, V, W]$ and $\pi_{ij} = Pr(Z_{ij} = 1|X, V, W)$. We model $\ln(\mu)$ and $\text{logit}(\pi)$ with the regression specified in the figure. Note that the model allows for different covariates to be specified in the two regressions; we have omitted the μ and π indices for clarity (see Methods for details).

BAMM-SC method [Sun et al., 2019]

BAMM-SC =Bayesian Hierarchical Dirichlet Multinomial Mixture Model

- Données : $X = [X_{gc\ell}]$ (gène g, cell. c, indiv ℓ)
- Hyp: Même nombre de classes K pour tous les individus
- Modèle : $Z = (z_{c\ell})$ latent clustering



$$X_{.c\ell} = (X_{gc\ell})_g \sim \text{Multi}\left(\sum_g X_{gc\ell}, p_{.c\ell}\right)$$

$$p_{.c\ell}|z_{c\ell} = k, \boldsymbol{\alpha}_{\ell,k} \sim \mathcal{D}(\boldsymbol{\alpha}_{\ell,k})$$

$$\boldsymbol{\alpha}_{g,.,k} = \prod_{\ell=1}^{L} \mathcal{LN}(\alpha_{g\ell,k}|\mu_{gk}, \sigma^2_{gk})$$

$\Rightarrow$ estimation de la distribution a posteriori par méthode de Gibbs

Plan

Détection des gènes marqueurs

- But : Après l'identification de groupes de cellules, on souhaite les caractériser en déterminant des gènes marqueurs pour chaque classe

- Procédures de test pour tester une différence d'expression de chaque gène dans les cellules d'un groupe par rapport à toutes les autres cellules
  (ex Seurat : Test de Wilcoxon (défaut), LRT, test de Welch, LRT-NB, MAST, DESeq2, ...)

  \+ Correction de tests multiples (ex: Bonferroni)

$\Rightarrow$ Mais la détection est biaisée par la procédure de classification sur laquelle on s'appuie

Differential analysis / Marker genes

Review: [Soneson and Robinson, 2018]

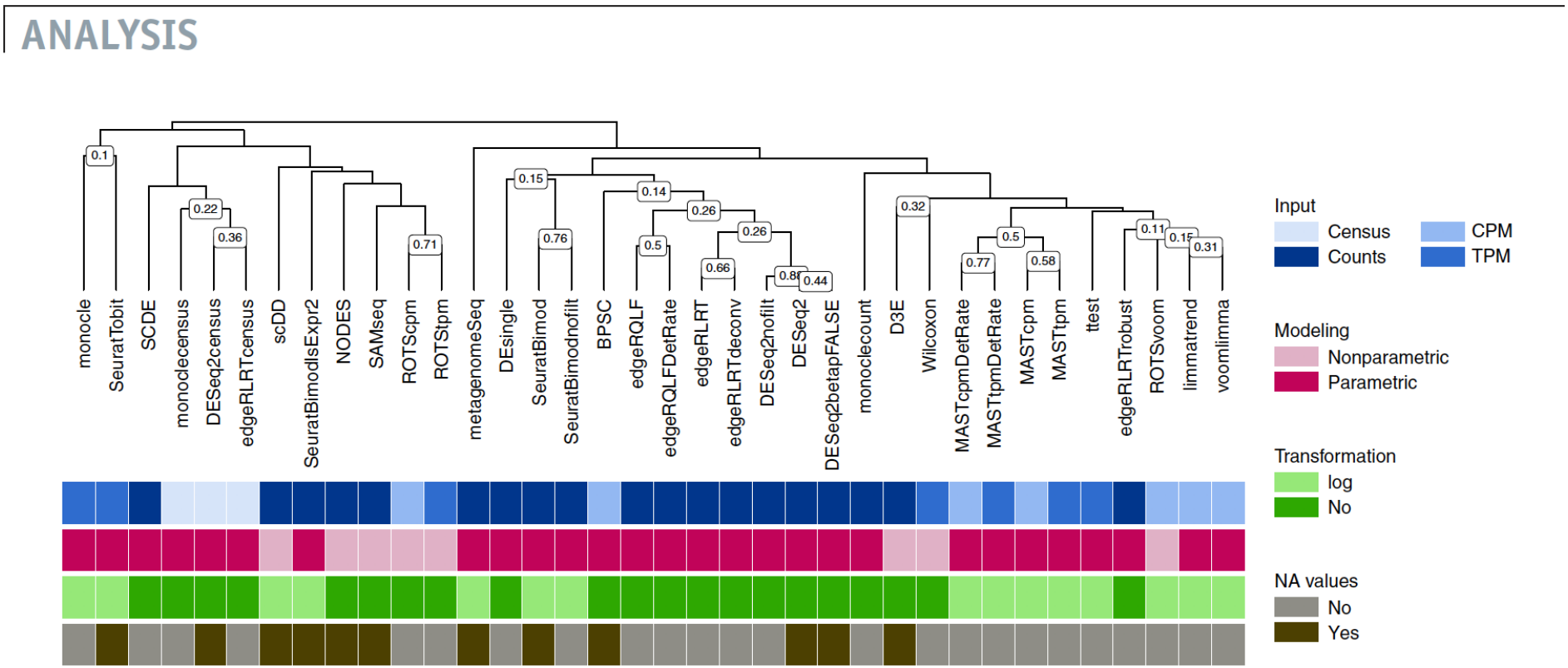
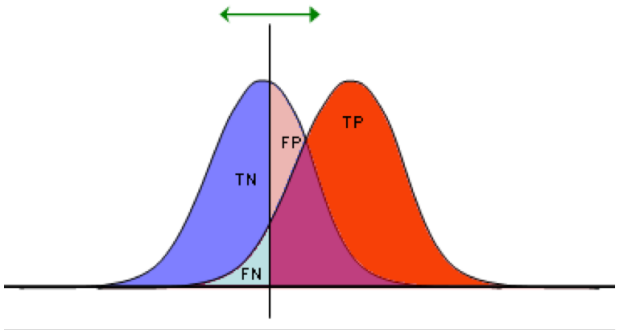
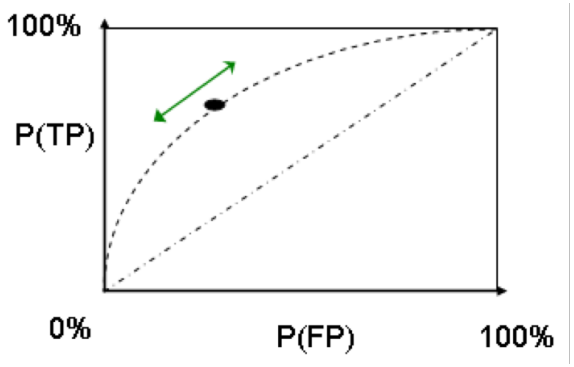


Figure 3 | Average similarities between gene rankings obtained by the evaluated DE methods. The dendrogram was obtained by complete-linkage hierarchical clustering based on the matrix of average AUCC values across all data sets. The labels of the internal nodes represent their stability across data sets (fraction of instances where they are observed). Only nodes with stability scores of at least 0.1 are labeled. Colored boxes represent method characteristics.

Détection des gènes marqueurs

- But : Après l'identification de groupes de cellules, on souhaite les caractériser en déterminant des gènes marqueurs pour chaque classe
- Procédures de test...
- Méthodes de classification supervisée
  - Calcul de AUC (area under the ROC curve)



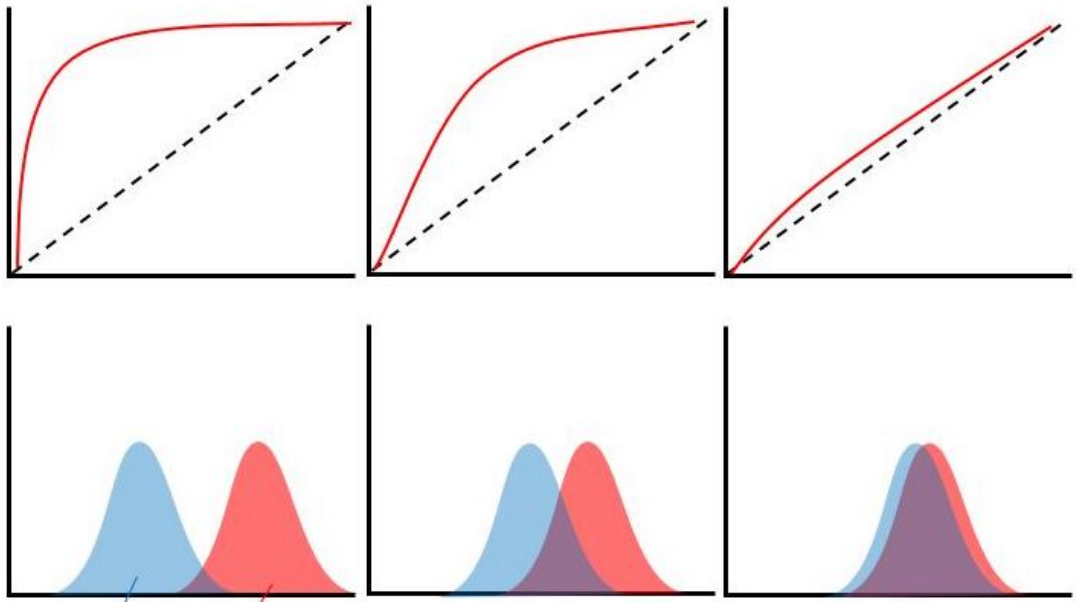


TP	FP
FN	TN
1	1

  - XGBoost dans Scedar (gènes rangés par importance)

Détection des gènes marqueurs

- But : Après l'identification de groupes de cellules, on souhaite les caractériser en déterminant des gènes marqueurs pour chaque classe
- Procédures de test...
- Méthodes de classification supervisée
  - Calcul de AUC (area under the ROC curve)



  - XGBoost dans Scedar (gènes rangés par importance)

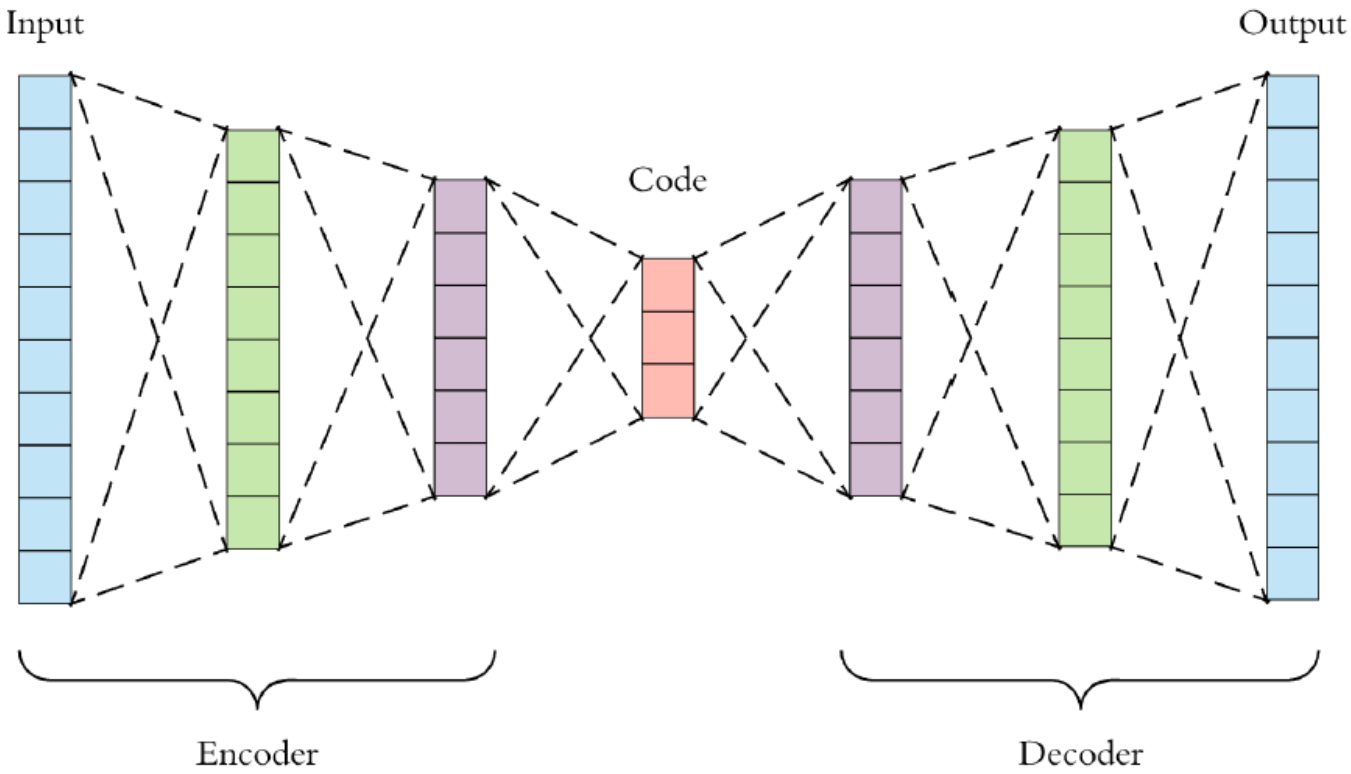
Détection des gènes marqueurs

- But : Après l'identification de groupes de cellules, on souhaite les caractériser en déterminant des gènes marqueurs pour chaque classe
- Procédures de test...
- Méthodes de classification supervisée
- Autres indicateurs :
  - pct1 et pct2 : % de cellules où le gène est exprimé dans chaque groupe
  - avg_logFC = log fold-change des expressions moyennes entre les deux groupes
  - Specificity score et Gene Specificity Shannon Index (package genesorteR)
  - ...

Plan

Principe d'un AutoEncoder

- Un autoencoder est un réseau de neurone qui essaie d'apprendre une représentation sparse des données et de capturer des structures latentes
- Il apprend à compresser les données (encodeur) puis il décompresse pour reconstruire quelque chose proche des données d'origine (décodeur).

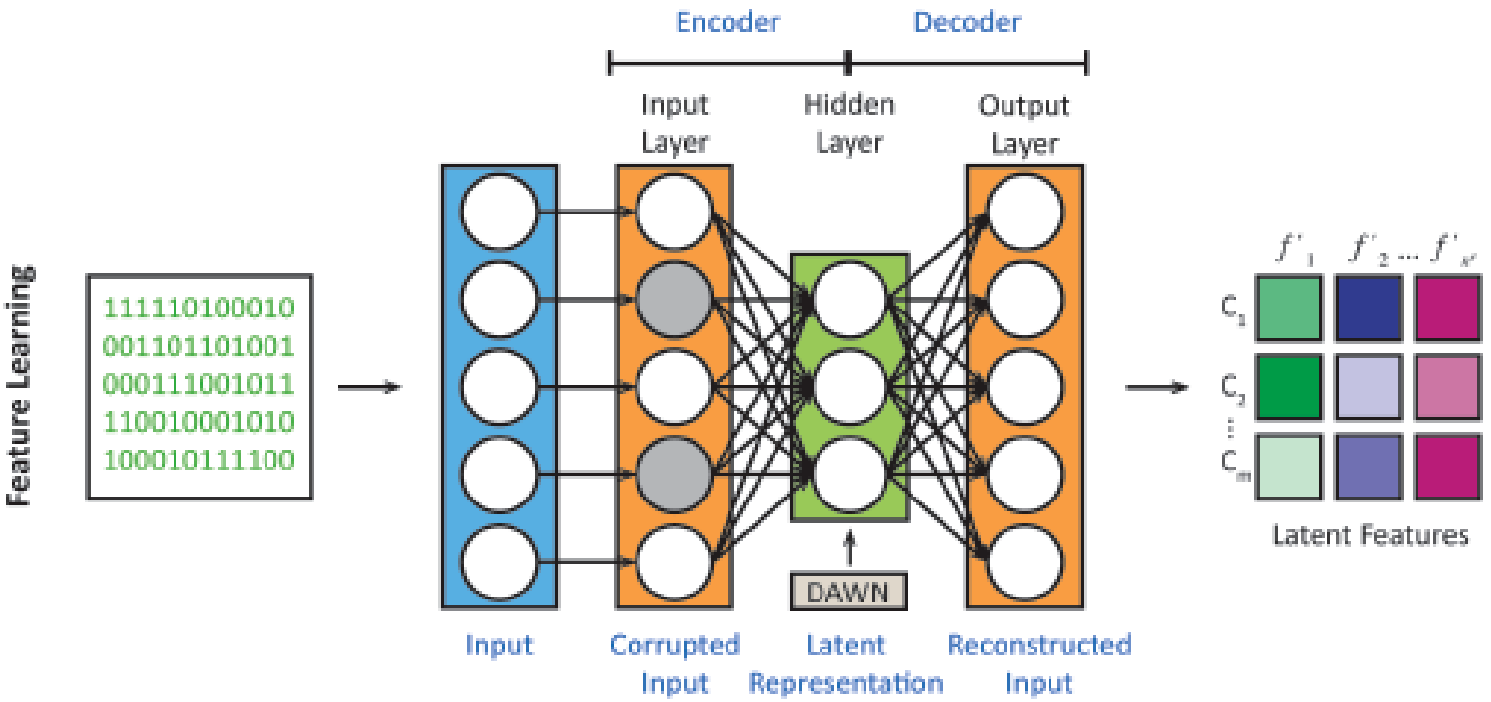


Principe d'un AutoEncoder

- Pour définir un autoencodeur :
  - Une fonction d'encodage : $z = f(x) = \phi(Wx + b)$
  - Une fonction de décodage : $y = g(z) = \psi(W'z + b')$
  - Paramètre $\theta = (W, W', b, b')$
  - Une fonction de perte à minimiser $L(x, g(f(x)))$+ sparsité
- Denoising autoencoder :
  on bruite les données d'entrée $x \to \tilde{x}$ et on minimise $L(x, g(f(\tilde{x})))$

DUSC [Srinivasan et al., 2019]

- DUSC = Deep Unsupervised Single-cell Clustering
- Réduction de dimension : Utilisation d'un "Denoising Autoencoder with Neural Network" (DAWN) pour obtenir un sous-espace latent Z
- Clustering dans le sous-espace latent Z avec un algorithme de type EM



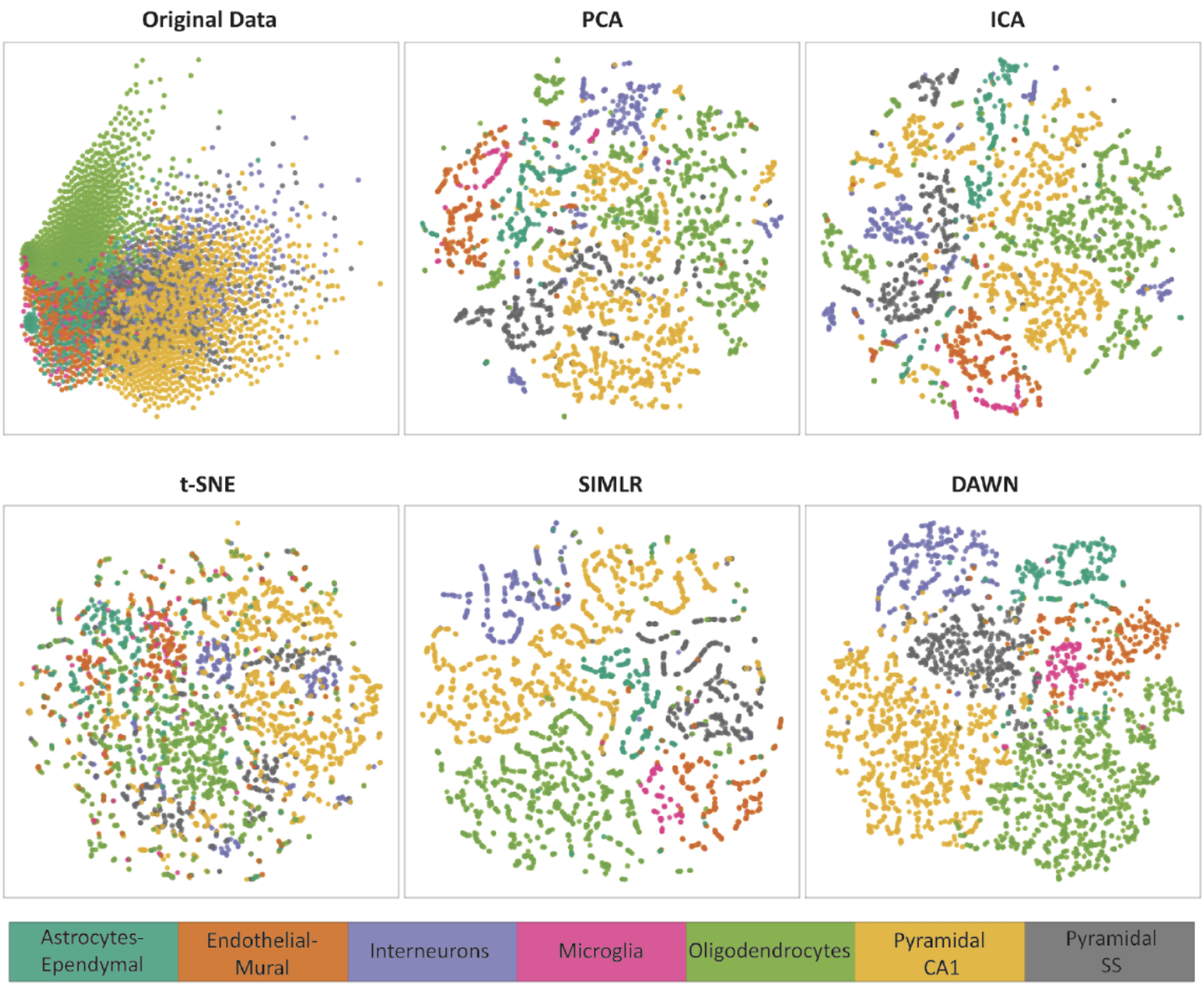
DUSC [Srinivasan et al., 2019]

- DUSC = Deep Unsupervised Single-cell Clustering
- Réduction de dimension : Utilisation d'un "Denoising Autoencoder with Neural Network" (DAWN) pour obtenir un sous-espace latent Z
  - Transformation des données : $\tilde{x}_c = (x_c - x_{min})/(x_{max} - x_{min}) \in [0, 1]$
  - Encoder: $z = \phi(Wx + b)$ où $\phi(x) = (1 + e^{-x})^{-1}$ fonction sigmoide
  - Decoder: $y = \psi(W'z + b')$ avec $W' = W^T$
  - Fonction de perte = cross-entropy of reconstruction

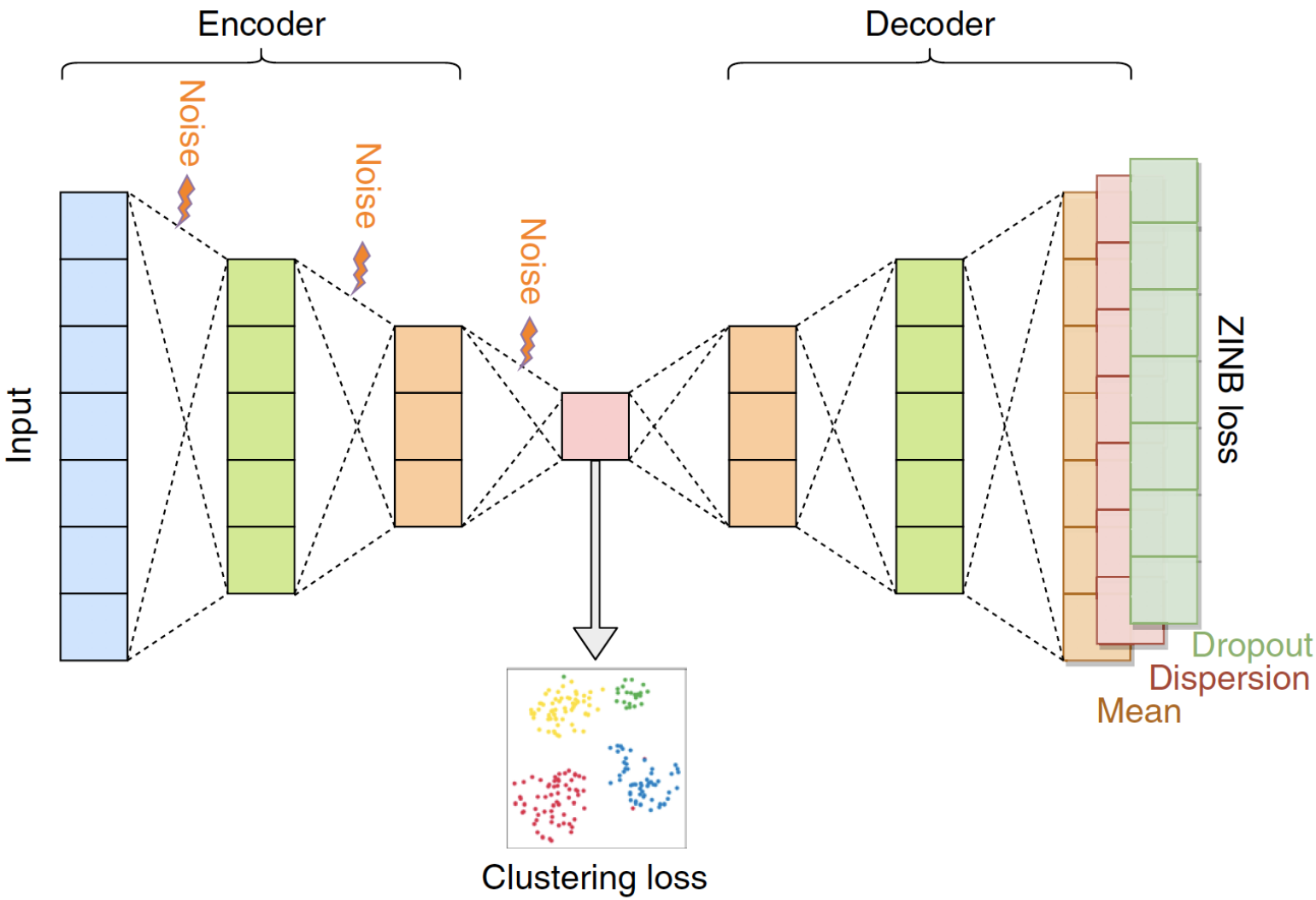
$$L(x, y) = \sum_{k=1}^{d} [x_k log(y_k) + (1 - x_k) log(1 - y_k)]$$

- Clustering dans le sous-espace latent Z avec un algorithme de type EM

DUSC [Srinivasan et al., 2019]



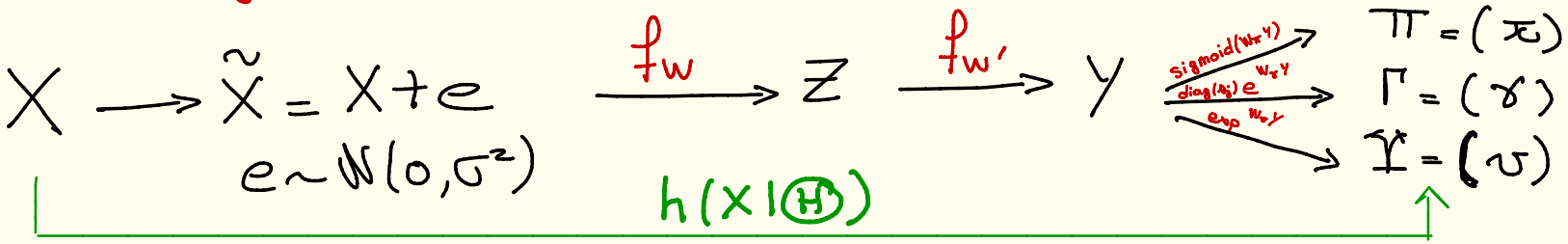
scDeepCluster [Tian et al., 2019]



scDeepCluster [Tian et al., 2019]

Prétraitement: logcomptages normalisés centrés réduits

Denoising ZINB model-based auto encoder



$$\Theta = \text{argmin} \; -\log\left[f_{ZINB}(X \mid \Pi, \Gamma, \Upsilon)\right]$$

clustering

$$\Theta \longrightarrow Z = f_W(X) \xrightarrow{\text{Kmeans}} M = \{m_k\}_{k=1}^{K} \longrightarrow Q, P$$

M : centres des classes

$Q = \text{fct}^\circ(M, Z)$

$P = \text{fct}^\circ(Q)$

on veut minimiser KL entre les lois Q et P

$\Rightarrow$ Fonction objective de scDeepCluster

$$L(\Theta, M) = L_{ZINB}(\Theta) + c\, L_c(M, \Theta)$$

Plan

Problématique

- Plusieurs tableaux de données contenant l'expression des gènes dans différentes cellules issues de plusieurs conditions

$$X_s = (X_{gcs})_{g=1,\dots,G,c=1,\dots,N_s},\ s = 1, \dots, S$$

- But : obtenir une classification de l'ensemble des cellules en tenant compte des S conditions

- Idées :
  - Classification des X_s séparément puis retrouver un lien entre les classes
  - Classification simultanée prenant en compte les S conditions

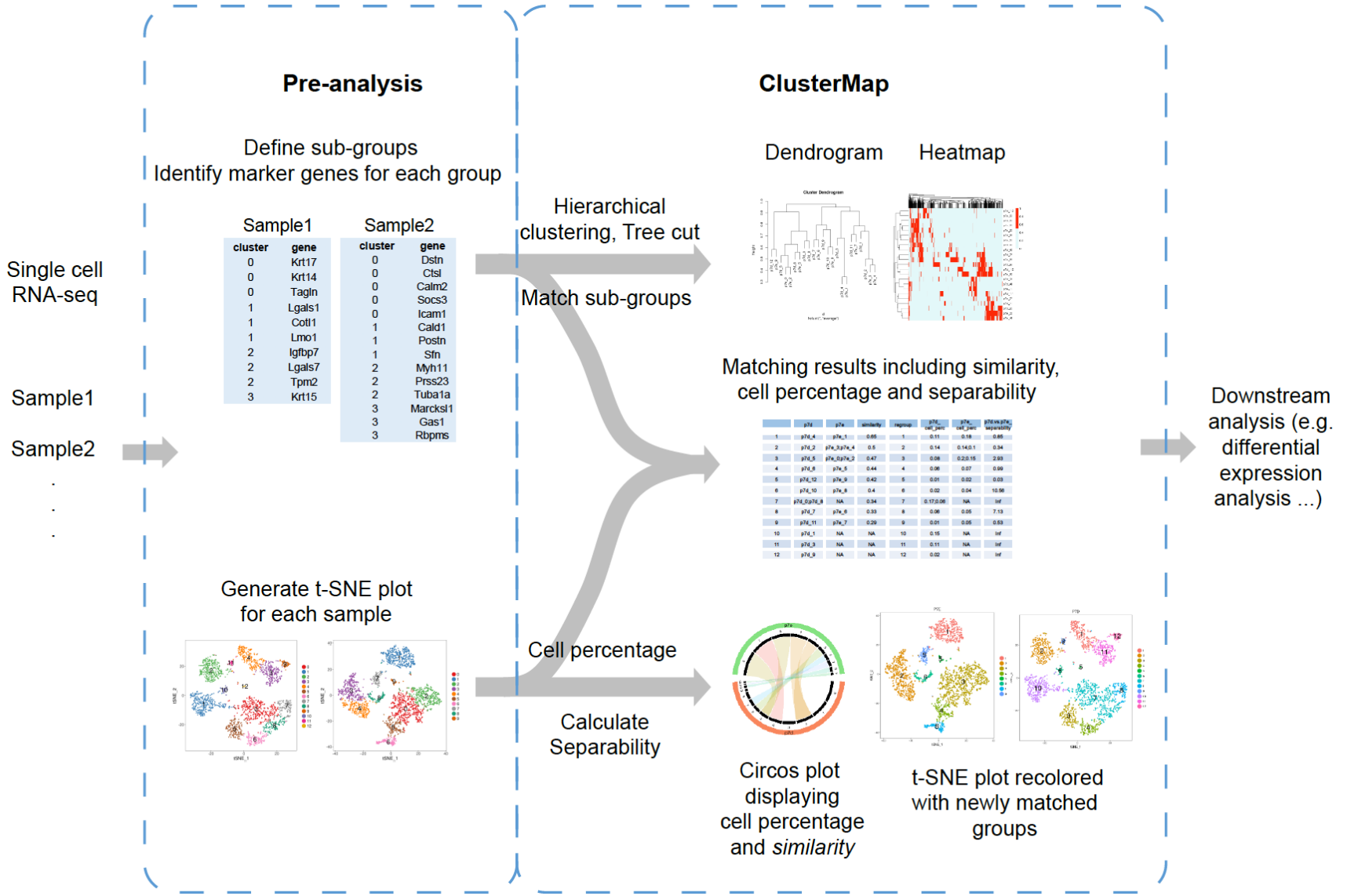
ClusterMap [Gao et al., 2018]

- Entrée : Les classifications $\left\{\mathcal{C}_k^{(s)}\right\}_{s=1,\ldots,S,k=1,\ldots,K(s)}$,
  la liste des gènes marqueurs de chaque classe et la réduction de dimension.
  $\Rightarrow$ tableau binaire de présence d'un gène comme marqueur pour chaque classe

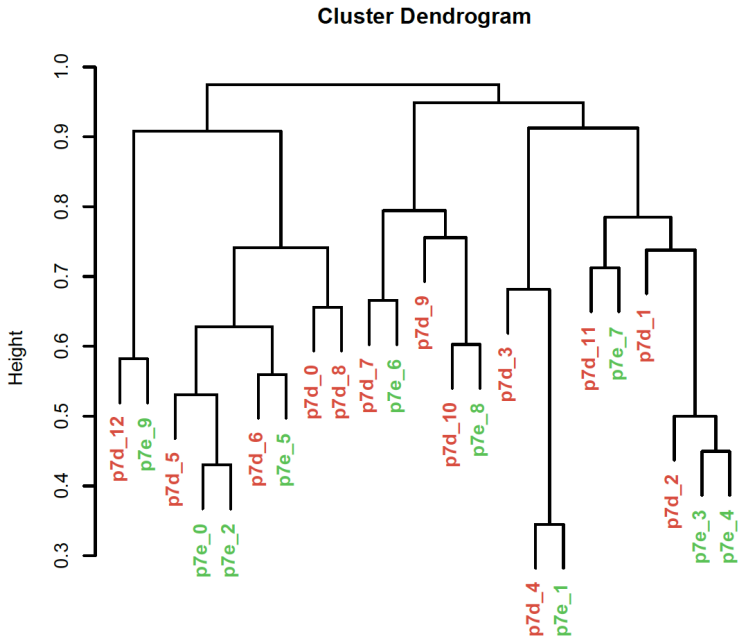
- CAH avec la distance de Jaccard et le lien moyen pour l'ensemble des classes $\mathcal{C}_k^{(s)}$

- Purity Tree Cut : Agrégation du dendrogramme selon un indice de pureté d'un noeud et la longueur entre deux branches

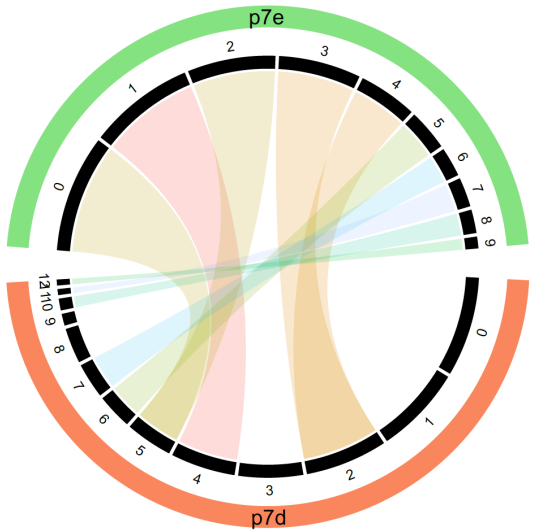
ClusterMap [Gao et al., 2018]



ClusterMap [Gao et al., 2018]



p7d	p7e	similarity	regroup	p7d_ cell_perc	p7e_ cell_perc	p7d.vs.p7e_ separability
p7d_4	p7e_1	0.65	1	0.11	0.18	0.85
p7d_2	p7e_3;p7e_4	0.5	2	0.14	0.14;0.1	0.34
p7d_5	p7e_0;p7e_2	0.47	3	0.08	0.2;0.15	2.93
p7d_6	p7e_5	0.44	4	0.06	0.07	0.99
p7d_12	p7e_9	0.42	5	0.01	0.02	0.03
p7d_10	p7e_8	0.4	6	0.02	0.04	10.56
p7d_0;p7d_8	NA	0.34	7	0.17;0.06	NA	Inf
p7d_7	p7e_6	0.33	8	0.06	0.05	7.13
p7d_11	p7e_7	0.29	9	0.01	0.05	0.53
p7d_1	NA	NA	10	0.15	NA	Inf
p7d_3	NA	NA	11	0.11	NA	Inf
p7d_9	NA	NA	12	0.02	NA	Inf

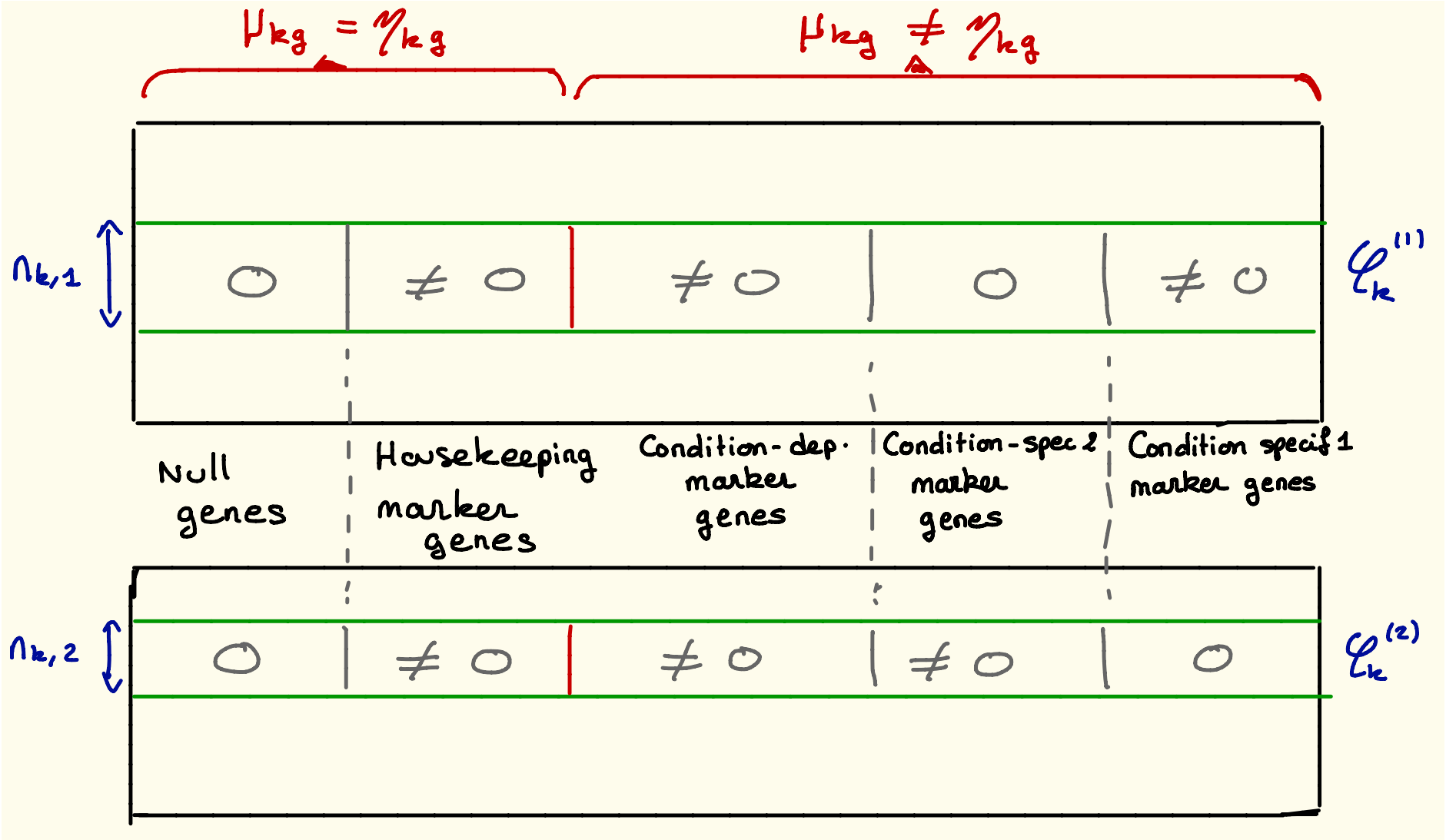


sparseDC [Barron et al., 2017]

- Objectif : obtenir une classification des cellules issues de $S = 2$ conditions et déterminer un rôle aux gènes (pour remplacer la partie détection de gènes marqueurs)
- Hyp : $K(1) = K(2)$ et $\mathcal{C}_k^{(1)}$ et $\mathcal{C}_k^{(2)}$ sont les mêmes types de cellules (possible vide)
- Données : $(\tilde{X}_{gcs})$ les log-comptages normalisés et centrés par gène (2 cond. confondues).
- On minimise $T(\mathcal{C}^{(1)}, \mathcal{C}^{(2)}, \mu, \eta) = \sum_{g=1}^{G} \sum_{k=1}^{K} \square_{gk}$

$$\begin{aligned} \square_{gk} &= \sum_{c \in \mathcal{C}_k^{(1)}} (\tilde{X}_{gc1} - \mu_{kg})^2 + \sum_{c \in \mathcal{C}_k^{(2)}} (\tilde{X}_{gc2} - \eta_{kg})^2 \\ &\quad + \lambda_1 \left(\sqrt{n_{k,1}} |\mu_{kg}| + \sqrt{n_{k,2}} |\eta_{kg}| \right) + \lambda_2 \left(\sqrt{n_{k,1}} + \sqrt{n_{k,2}} \right) |\mu_{kg} - \eta_{kg}| \end{aligned}$$

Interprétation du rôle des gènes en comparant les μ_{kg} et η_{kg} obtenus



Plan

Clustering ensemble / Consensus clustering

- Objectif : on a obtenu Q classifications des mêmes données (avec nb de classes différents). On souhaite résumer ces résultats dans une classification globale
- **SAFE** : HGPA, MCLA et CSPA (3 méthodes basées sur des hypergraphes)
- **clusterExperiment** : CAH sur matrice de consensus + procédure de fusion (proportion de gènes DE entre les noeuds enfants)
- **SAME** : Utilisation de mélanges de lois multinomiales
- ....

References I

Barron, M., Zhang, S., and Li, J. (2017).
A sparse differential clustering algorithm for tracing cell type changes via single-cell rna-sequencing data.
Nucleic acids research, 46(3):e14–e14.

Gao, X., Hu, D., Gogol, M., and Li, H. (2018).
Clustermap: Comparing analyses across multiple single cell rna-seq profiles.
bioRxiv.

Haghverdi, L., Lun, A. T., Morgan, M. D., and Marioni, J. C. (2018).
Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors.
Nature biotechnology, 36(5):421.

Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S., and Vert, J.-P. (2017).
Zinb-wave: A general and flexible method for signal extraction from single-cell rna-seq data.
bioRxiv.

Soneson, C. and Robinson, M. (2018).
Bias, robustness and scability in single-cell differential expression analysis.
Nature Methods, 15:255–261.

Srinivasan, S., Johnson, N. T., and Korkin, D. (2019).
A hybrid deep clustering approach for robust cell type profiling using single-cell rna-seq data.
bioRxiv, page 511626.

Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck III, W. M., Hao, Y., Stoeckius, M., Smibert, P., and Satija, R. (2019).
Comprehensive integration of single-cell data.
Cell.

Sun, Z., Chen, L., Xin, H., Jiang, Y., Huang, Q., Cillo, A. R., Tabib, T., Kolls, J. K., Bruno, T. C., Lafyatis, R., et al. (2019).
A bayesian mixture model for clustering droplet-based single-cell transcriptomic data from population studies.
Nature communications, 10(1):1649.

References II

Tian, T., Wan, J., Song, Q., and Wei, Z. (2019).

Clustering single-cell rna-seq data with a model-based deep learning approach.
Nature Machine Intelligence, 1(4):191.

Zhang, Y. and Taylor, D. M. (2019).

Scedar: a scalable python package for single-cell rna-seq exploratory data analysis.
bioRxiv.