Multivariate statistics for single-data data analysis

Zero-inflated count matrix factorization for data exploration and sparse PLS-based logistic regression for classification

Ghislain DURIF[1], Laurent Modolo[2], Jeff Mold[3],
Sophie Lambert-Lacroix[4], Franck Picard[5]

Talavera-Lopéz C, Reinius B, Réu P, Ståhl PL, Borgström E, Hård JL,
Picelli S, Blom K, Marquardt N, Andersson B, Sandberg R,
Michaelsson J and Frisén J

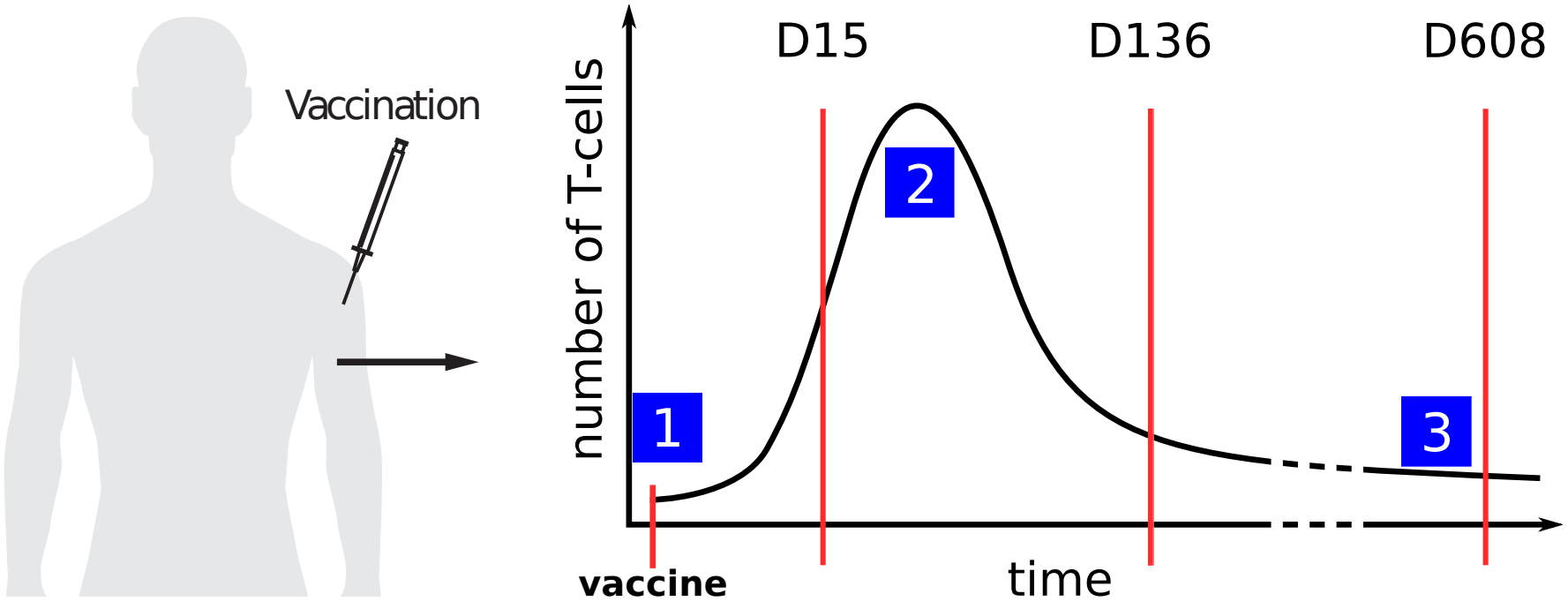
[1]INRIA Grenoble Alpes – THOTH Team
[2]LBMC UMR 5239 – CNRS/ENS Lyon
[3] Karolinska Institutet – Stockholm
[4]TIMC-IMAG UMR 5525 – Université Grenobles Alpes
[5]LBBE UMR 5558 – CNRS/Université Claude Bernard Lyon 1

Ascona Workshop 2017, 5 Mai 2017

Table of Contents

Table of Contents

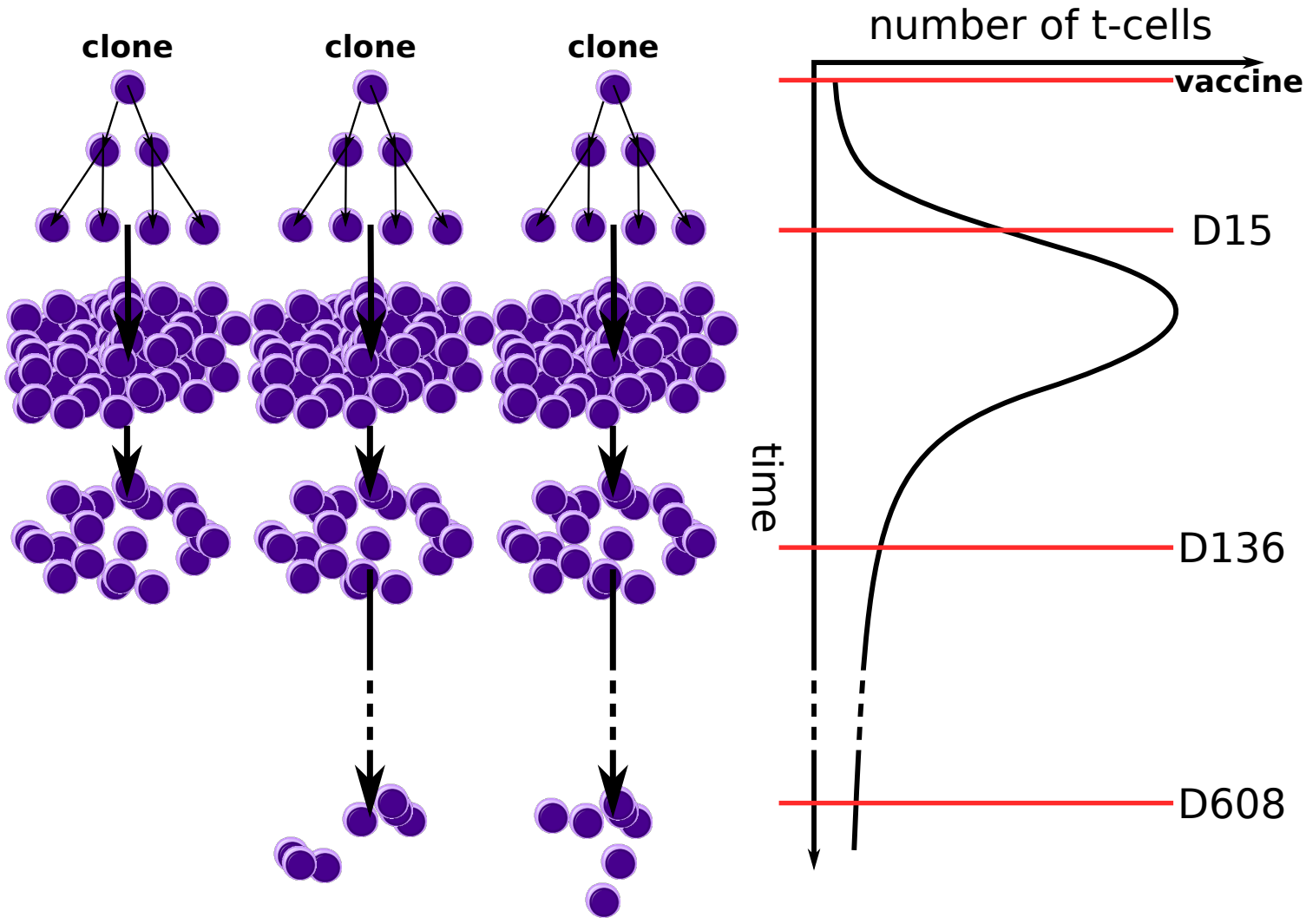
Immune response after a shot of yellow fever vaccine



T-cells:

1. **Naive** T-cells with a unique T-cell receptor
2. **Effector** T-cells multiply upon exposure to their cognate antigen
3. formation of long-lasting **memory** cells

Clonality in the T-cells immune response



Each clone is caracterized by an **unique T-Cell receptor (TCR)**.

Questions

Biological questions

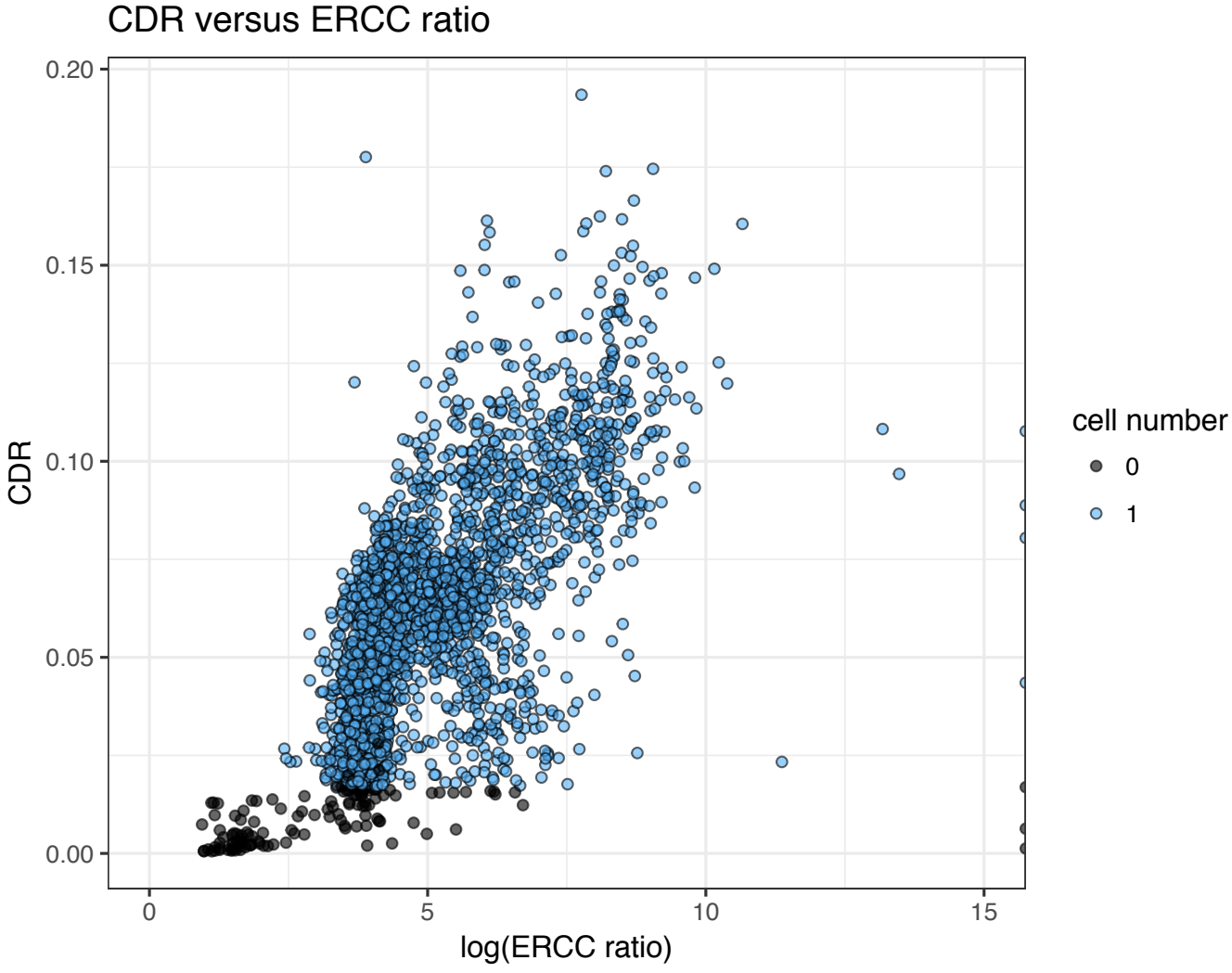
- Can we identify effector and memory cells?
- Can we identify effector and memory clones?

Methodological questions

- Quality control
- Identification of transcriptomic signatures

Table of Contents

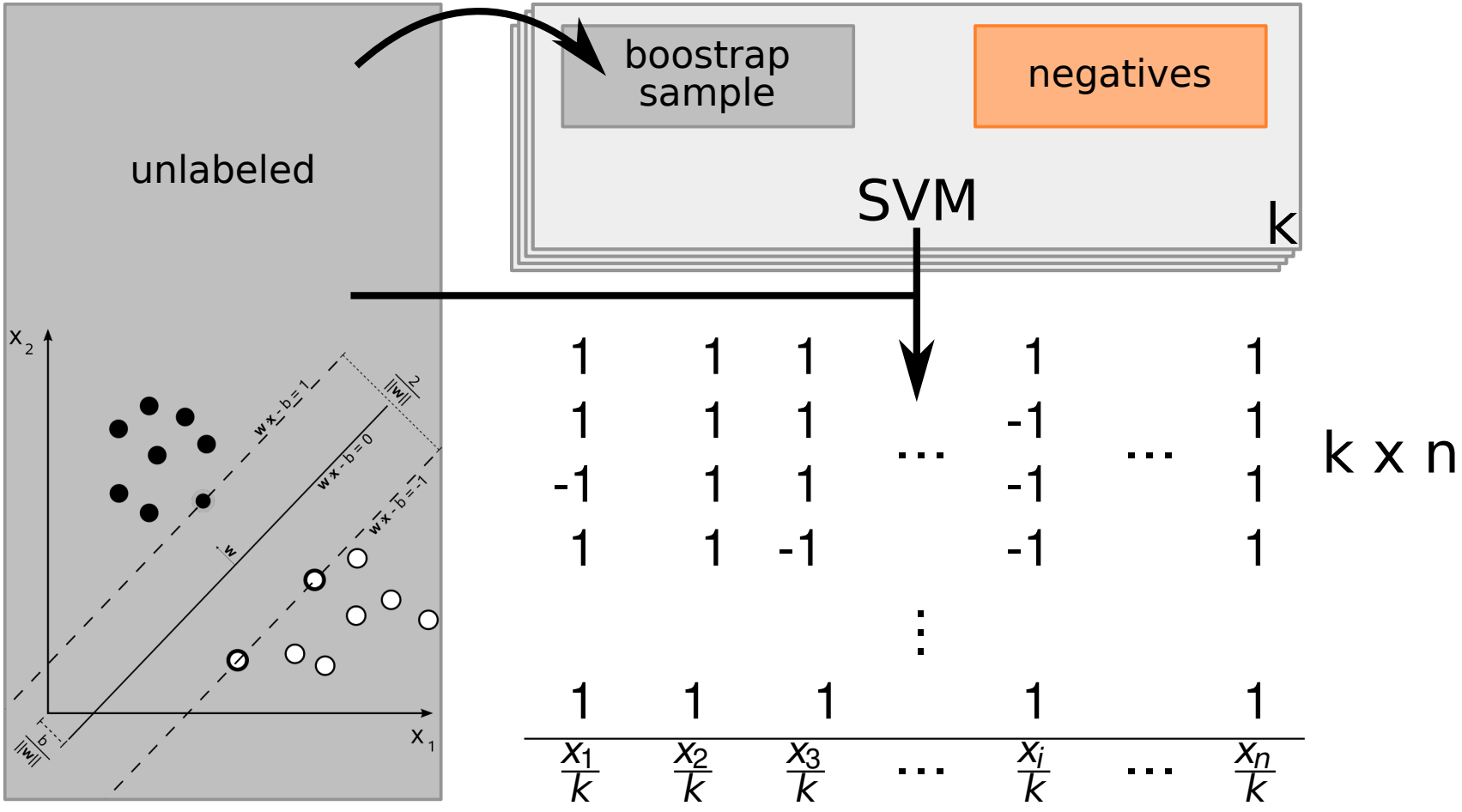
Quality control to remove the "bad cells"



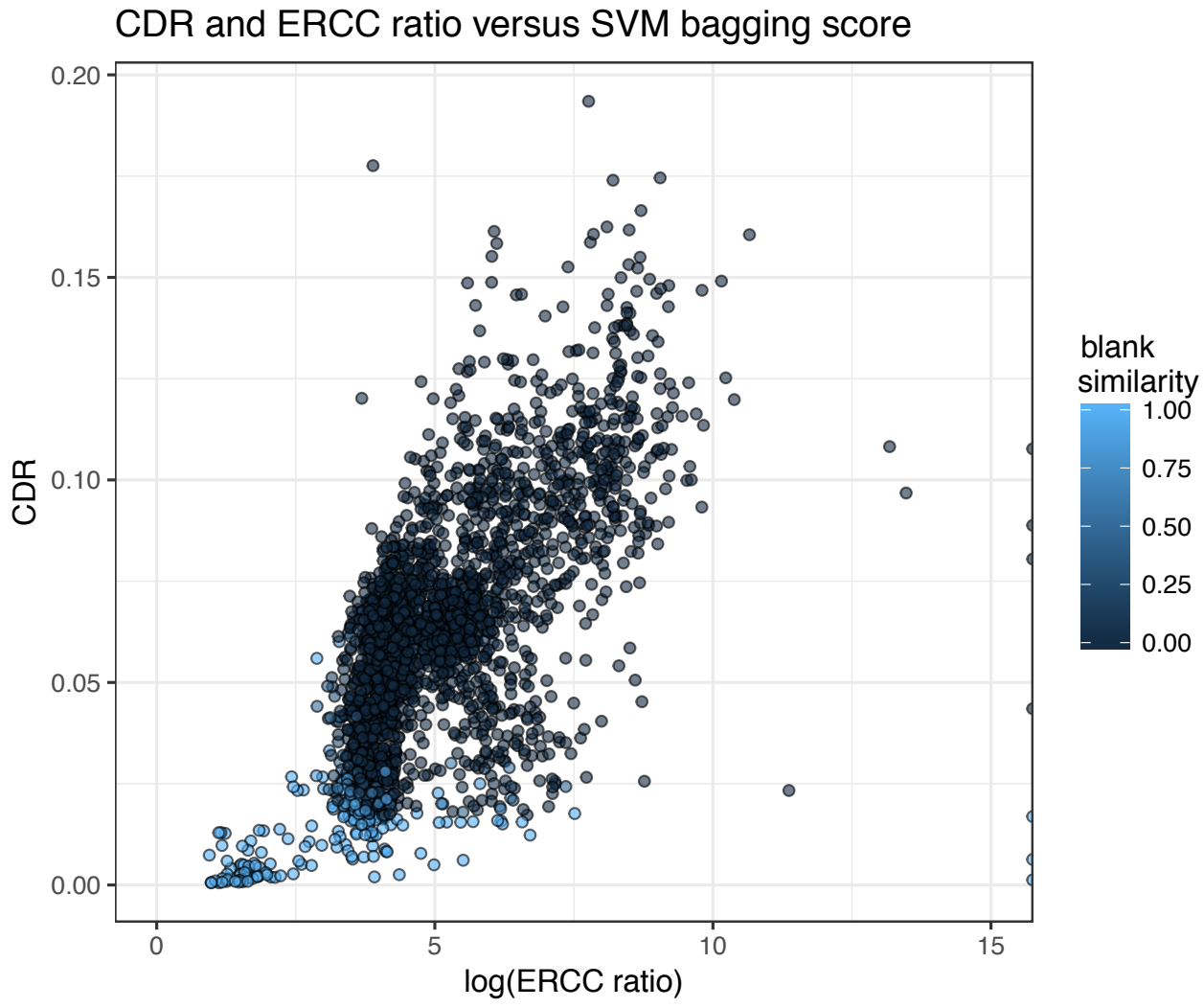
$$\text{CDR} = \frac{\#\text{ genes } >10 \text{ reads}}{\#\text{ genes}}, \quad \text{ERCC ratio} = \frac{\#\text{ total genes reads}}{\#\text{ total ERCC}^1\text{reads}}$$

Quality control to remove the "bad cells"

SVM-bagging algorithm (Mordelet et al,. 2014)

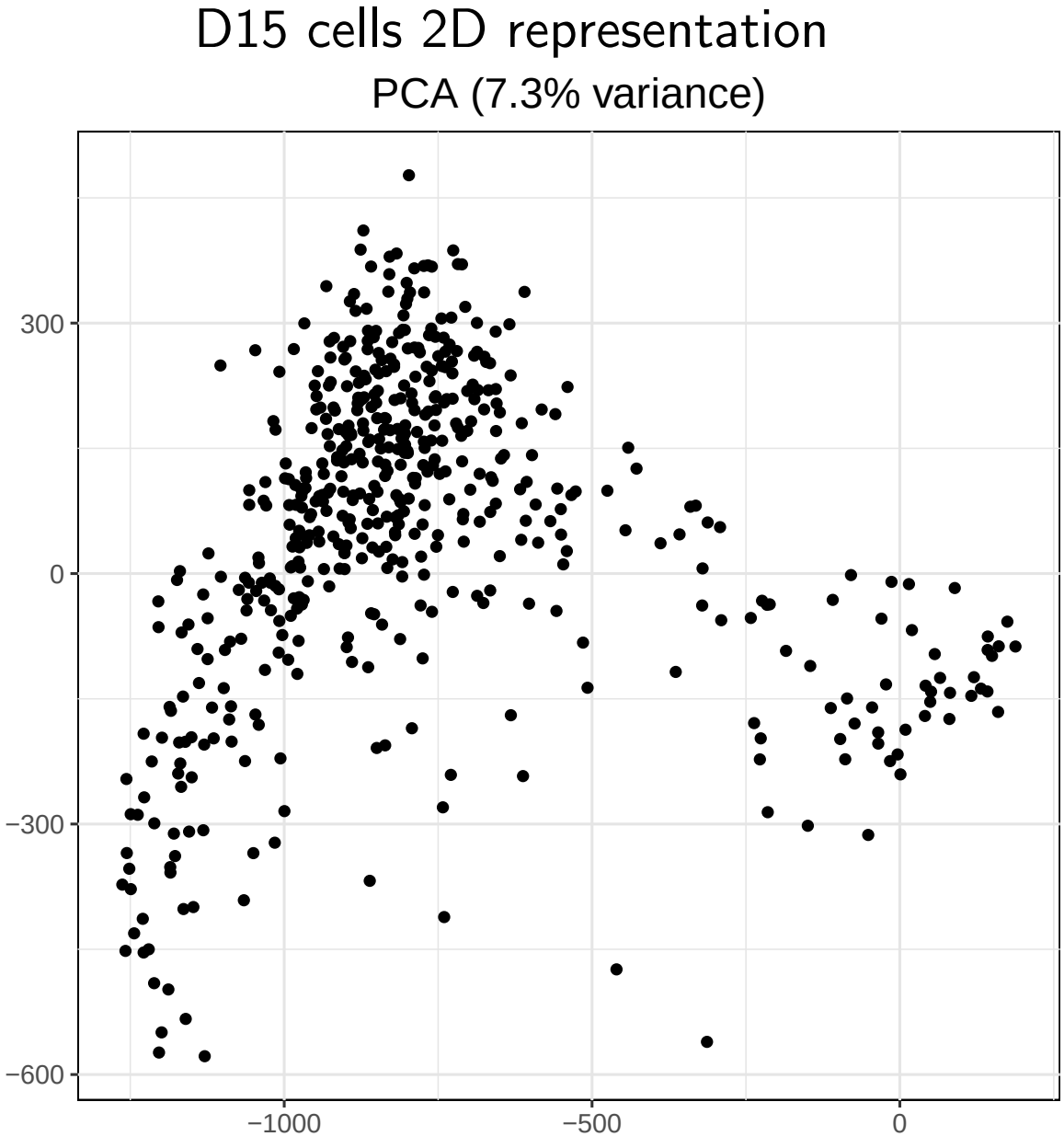


Quality control to remove the "bad cells"



373/2373 "bad cells"

Quality control to remove the "bad cells"



Quality control to remove the "bad cells"

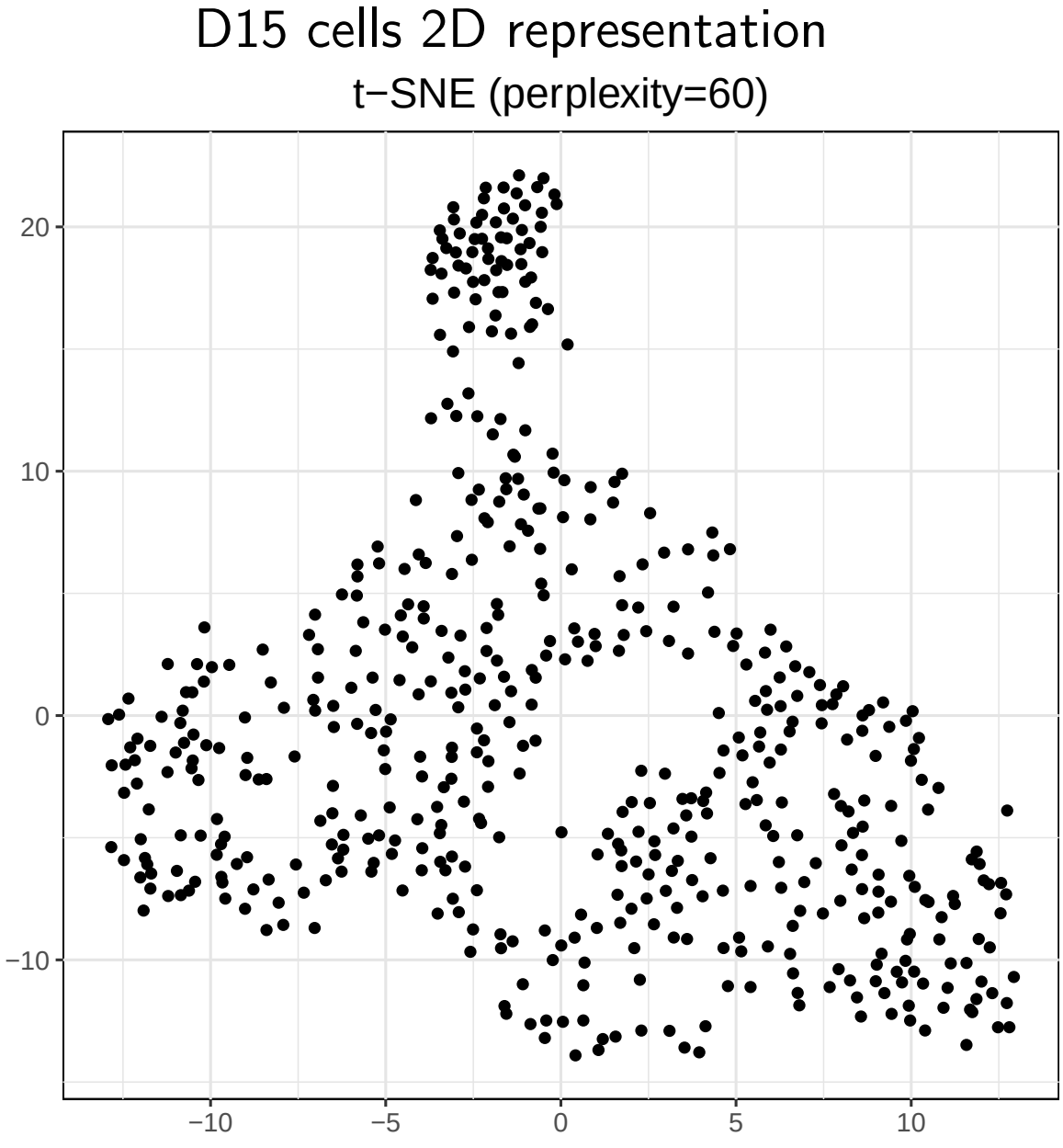


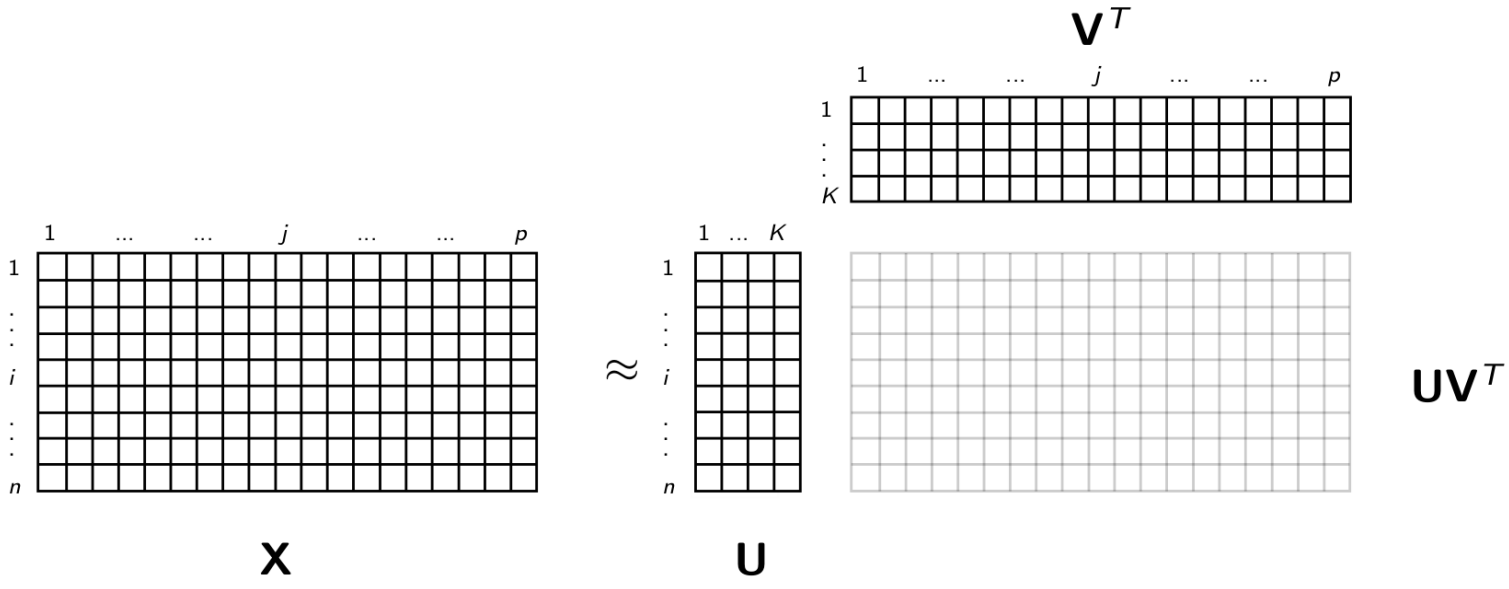
Table of Contents

Matrix factorization: $\mathbf{X} \approx \mathbf{U}\mathbf{V}^T$

Samples: $\mathbf{U} \in \mathbb{R}^{n \times K}$

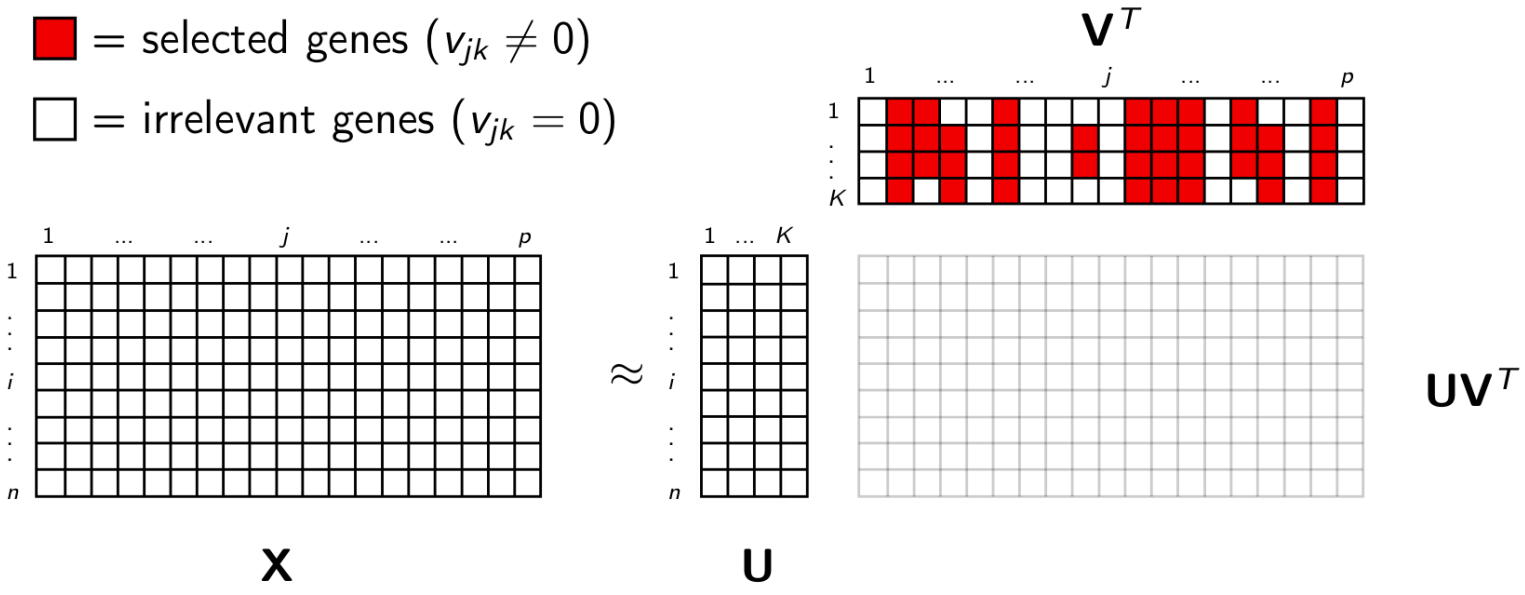
Variables: $\mathbf{V} \in \mathbb{R}^{p \times K}$

} Low dimensional representation



→ Low-rank representation of $\mathbf{X}$

Sparse matrix factorization



Penalization on ℓ_1 norm (Lasso):

$$\underset{\substack{\mathbf{u}\in\mathbb{R}^n \\ \mathbf{v}\in\mathbb{R}^p}}{\operatorname{argmin}} \left\{ \left\| \mathbf{X} - \mathbf{u}\mathbf{v}^T \right\|_F^2 + \lambda \sum_{j=1}^{p} |v_j| \right\}$$

$\rightarrow$ provides an easy interpretation of PCA axis

RNA-seq data = Counts

1) Interest for **lowly expressed genes** in single-cell

2) **Over-dispersion** in RNA-seq data $\rightarrow \text{Var}(X_{ij}) > \mathbb{E}[X_{ij}]$

3) Single-cell data: **zero-inflation** $\rightarrow \mathbb{P}(X_{ij} = 0) > e^{-\lambda}$

- **true zeros**
- transcription is **bursty** (cells are not synchronized)
- failure of the sequencing (**dropout events** = loss of the information)

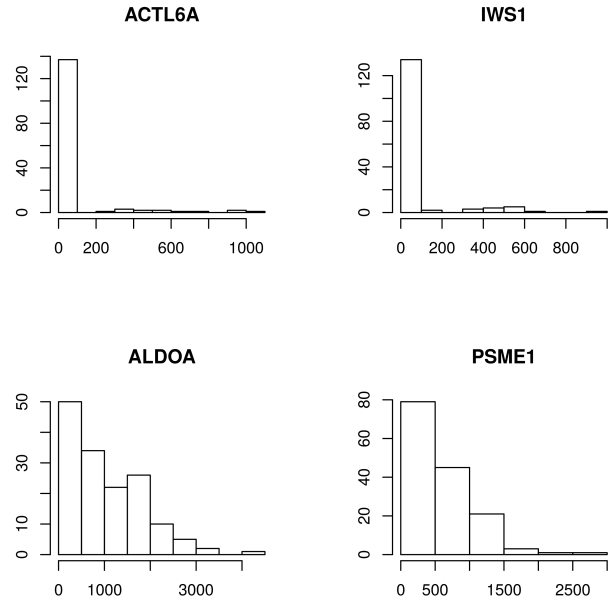
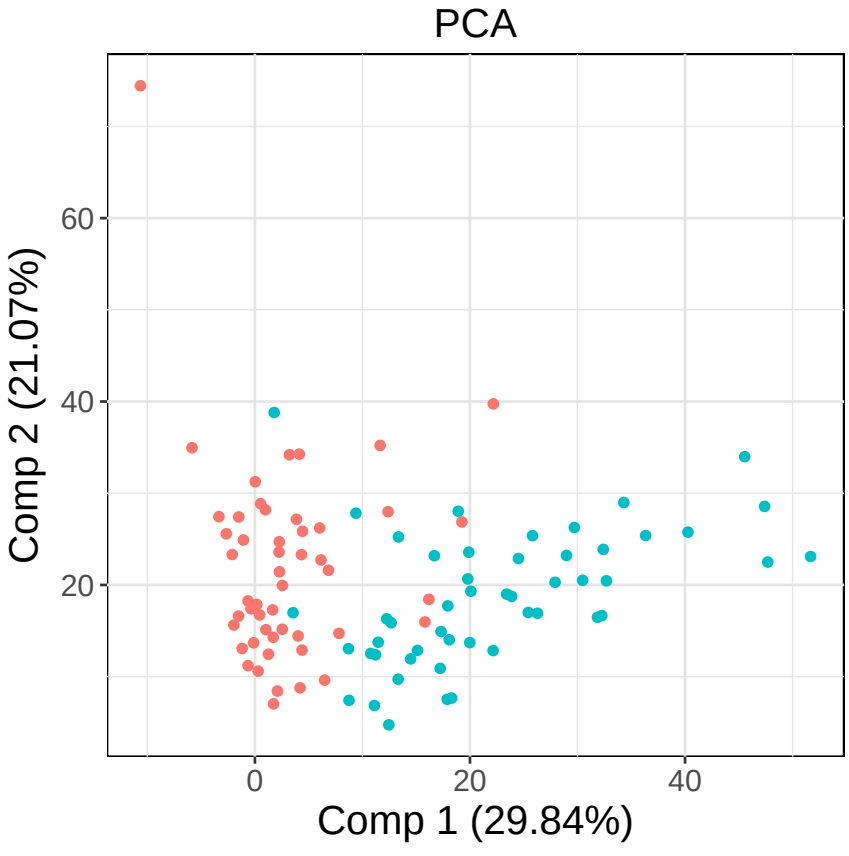


Figure: Count distribution for different genes

Appropriate geometry for count representation

High intensity Poisson data



Same data with zero-inflation

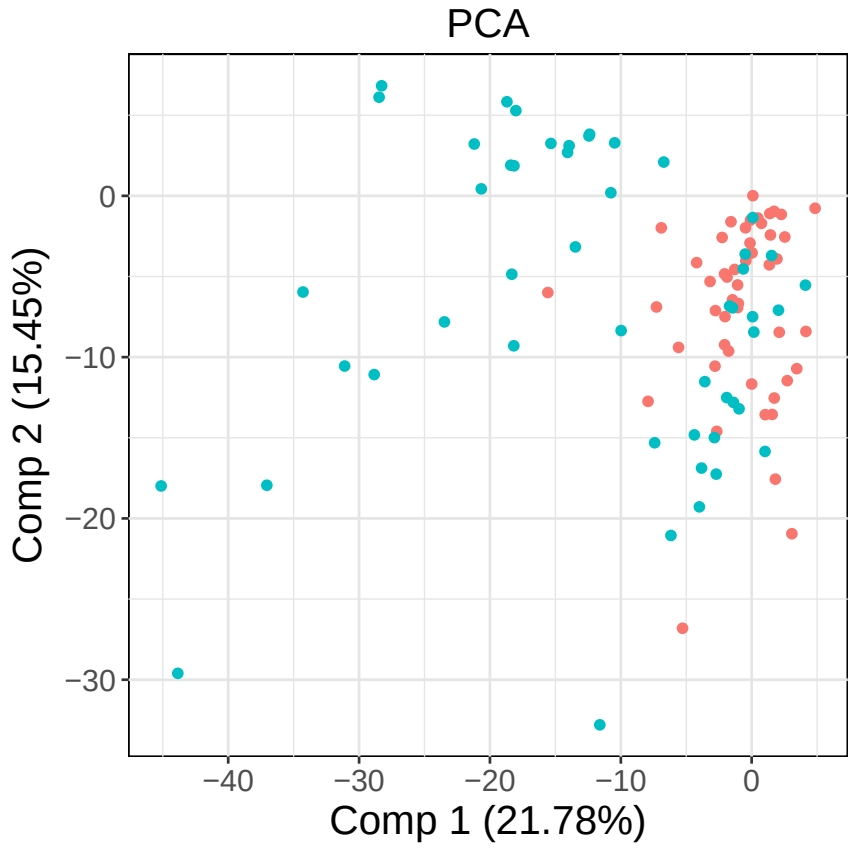


Table: Observations scores over first two principal components

Our contribution: probabilistic PCA for count data

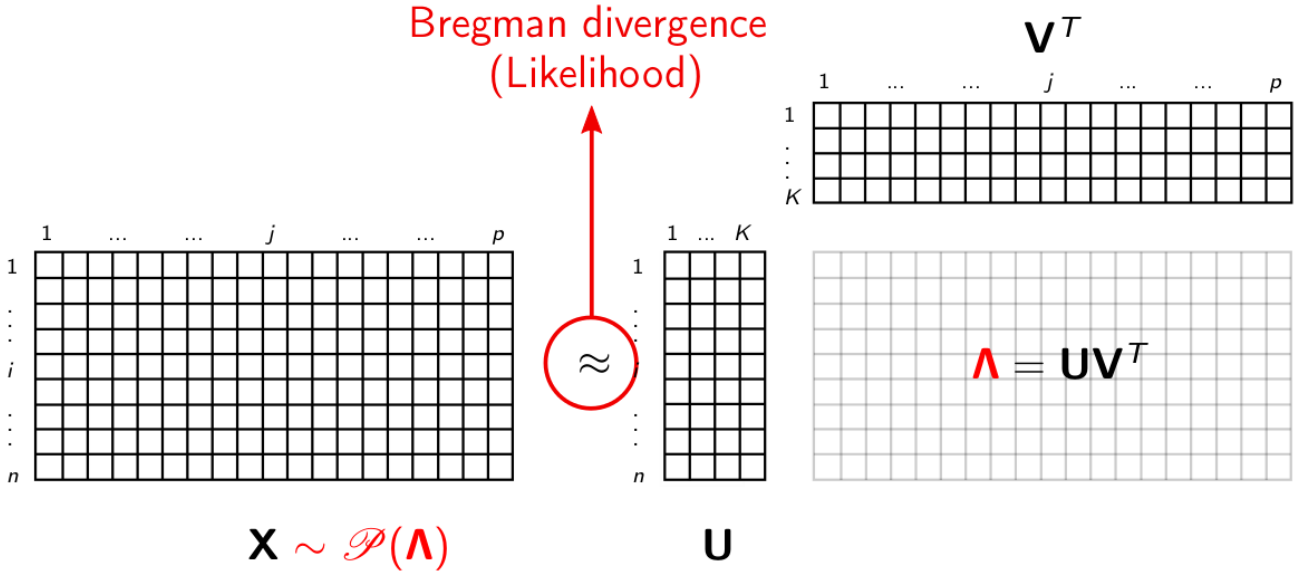
Count Matrix Factorization (CMF)

- Embed PCA with a **probabilistic model** (Collins et al. 2001)
  - $\rightarrow$ x_{ij} = over-dispersed, zero-inflated, count data
  - $\rightarrow$ $X_{ij} \sim$ probability distribution in the exponential family
  - $\rightarrow$ Replace $\|\cdot\|_2$ approximation by likelihood-based approaches
  - $\rightarrow$ Factorization of $\mathbb{E}[\mathbf{X}]$ rather than $\mathbf{X}$

Poisson Non-negative matrix factorization (NMF)

(Lee and Seung 1999)

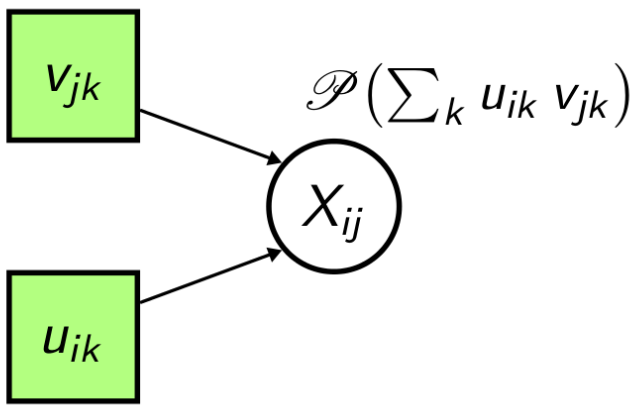
- $X_{ij} \sim \mathscr{P}(\lambda_{ij})$ with the Poisson rate matrix $\mathbf{\Lambda} = [\lambda_{ij}] \in (\mathbb{R}^+)^{n \times p}$
- Factorization: $\mathbb{E}[\mathbf{X}] = \mathbf{\Lambda} = \mathbf{U}\mathbf{V}^T \leftrightarrow \lambda_{ij} = \sum_k u_{ik}\, v_{jk}$
- Maximum Likelihood Estimation under non-negativity constraint over $\mathbf{U}$ and $\mathbf{V}$



Poisson Non-negative matrix factorization (NMF)

(Lee and Seung 1999)

- $X_{ij} \sim \mathscr{P}(\lambda_{ij})$ with the Poisson rate matrix $\mathbf{\Lambda} = [\lambda_{ij}] \in (\mathbb{R}^+)^{n \times p}$
- Factorization: $\mathbb{E}[\mathbf{X}] = \mathbf{\Lambda} = \mathbf{U}\mathbf{V}^T \leftrightarrow \lambda_{ij} = \sum_k u_{ik}\, v_{jk}$
- Maximum Likelihood Estimation under non-negativity constraint over $\mathbf{U}$ and $\mathbf{V}$



- $\mathbf{U}$ and $\mathbf{V}$ are parameters
- Optimization computationally expensive
- **Does not account for over-dispersion or zero-inflation**

Gamma-Poisson factor model

(Cemgil 2009)

- Independent Gamma prior distributions over $\mathbf{U}$ and $\mathbf{V}$:

$$U_{ik} \sim \Gamma(\alpha_{k,1}, \alpha_{k,2}) \quad \text{and} \quad V_{jk} \sim \Gamma(\beta_{k,1}, \beta_{k,2})$$

- Conditional Poisson distribution over the data $\mathbf{X}$:

$$X_{ij} \,|\, (U_{ik}, V_{jk})_{k=1:K} \sim \mathscr{P}(\textstyle\sum_k U_{ik}\, V_{jk})$$

Gamma-Poisson factor model

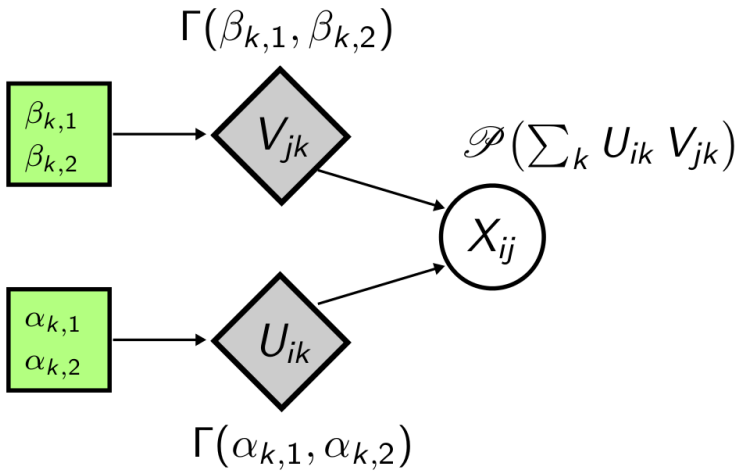
(Cemgil 2009)

- Independent Gamma prior distributions over $\mathbf{U}$ and $\mathbf{V}$:

$$U_{ik} \sim \Gamma(\alpha_{k,1}, \alpha_{k,2}) \quad \text{and} \quad V_{jk} \sim \Gamma(\beta_{k,1}, \beta_{k,2})$$

- Conditional Poisson distribution over the data $\mathbf{X}$:

$$X_{ij} \,|\, (U_{ik}, V_{jk})_{k=1:K} \sim \mathscr{P}(\textstyle\sum_k U_{ik}\, V_{jk})$$



- **Factors = latent variables**
- **Recover the posterior:**

$$\hat{\mathbf{U}} = \mathbb{E}[\mathbf{U} \,|\, \mathbf{X}] \text{ and } \hat{\mathbf{V}} = \mathbb{E}[\mathbf{V} \,|\, \mathbf{X}]$$

- **Marginal distribution is over-dispersed:**

$$\mathrm{Var}(X_{ij}) > \mathbb{E}[X_{ij}]$$

Sparse Gamma-Poisson model

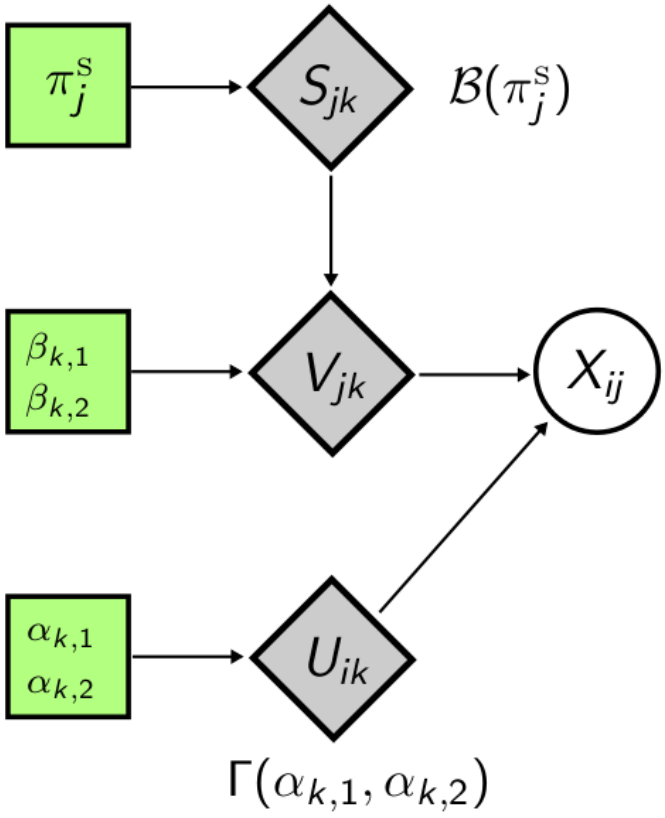
Sparsity on $\mathbf{V}$:

- Variable j contributes to factor k if $V_{jk} \neq 0$
- Objective: force the V_{jk}'s to be null for non pertinent genes

Sparse Gamma-Poisson model

Sparsity on $\mathbf{V}$:

- Variable j contributes to factor k if $V_{jk} \neq 0$
- Objective: force the V_{jk}'s to be null for non pertinent genes



- **Gamma-Dirac mixture**
  $V_{jk} \sim (1 - \pi_j^s)\,\delta_0 + \pi_j^s\,\Gamma(\beta_{k,1}, \beta_{k,2})$
- $\pi_j^s \in [0, 1]$ probability that gene j contributes to the model
- S_{jk} = sparsity indicator

"Zero-inflated" Gamma-Poisson factor model

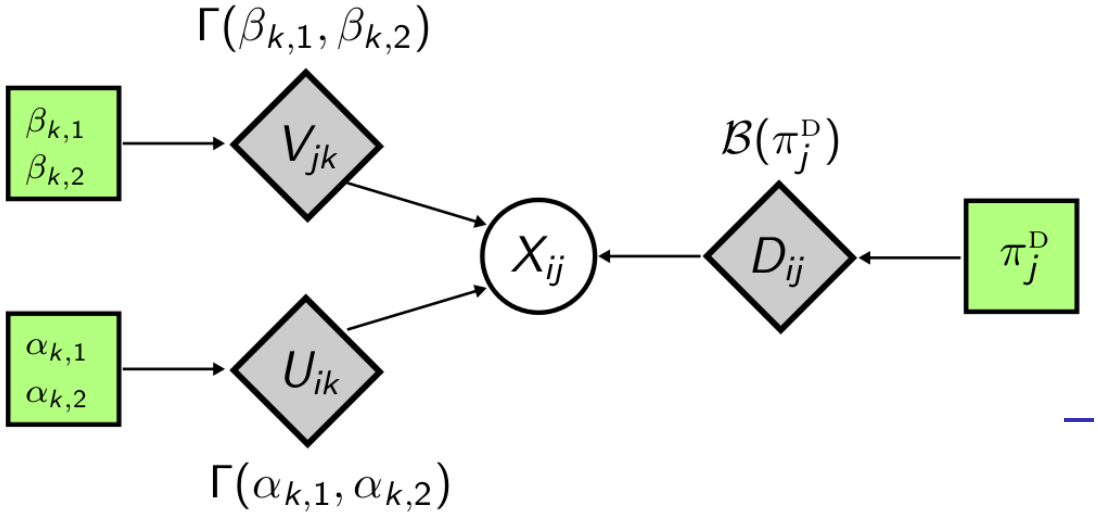
Poisson-Dirac mixture

- $X_{ij} \,|\, (U_{ik}, V_{jk})_{k=1:K} \sim (1 - \pi_j^{\mathsf{D}}) \times \delta_0 + \pi_j^{\mathsf{D}} \times \mathscr{P}(\lambda_{ij})$
- $1 - \pi_j^{\mathsf{D}} \in [0, 1]$ is the zero-inflation for gene j

"Zero-inflated" Gamma-Poisson factor model

Poisson-Dirac mixture

- $X_{ij} \,|\, (U_{ik}, V_{jk})_{k=1:K} \sim (1 - \pi_j^{\mathsf{D}}) \times \delta_0 + \pi_j^{\mathsf{D}} \times \mathscr{P}(\lambda_{ij})$
- $1 - \pi_j^{\mathsf{D}} \in [0, 1]$ is the zero-inflation for gene j

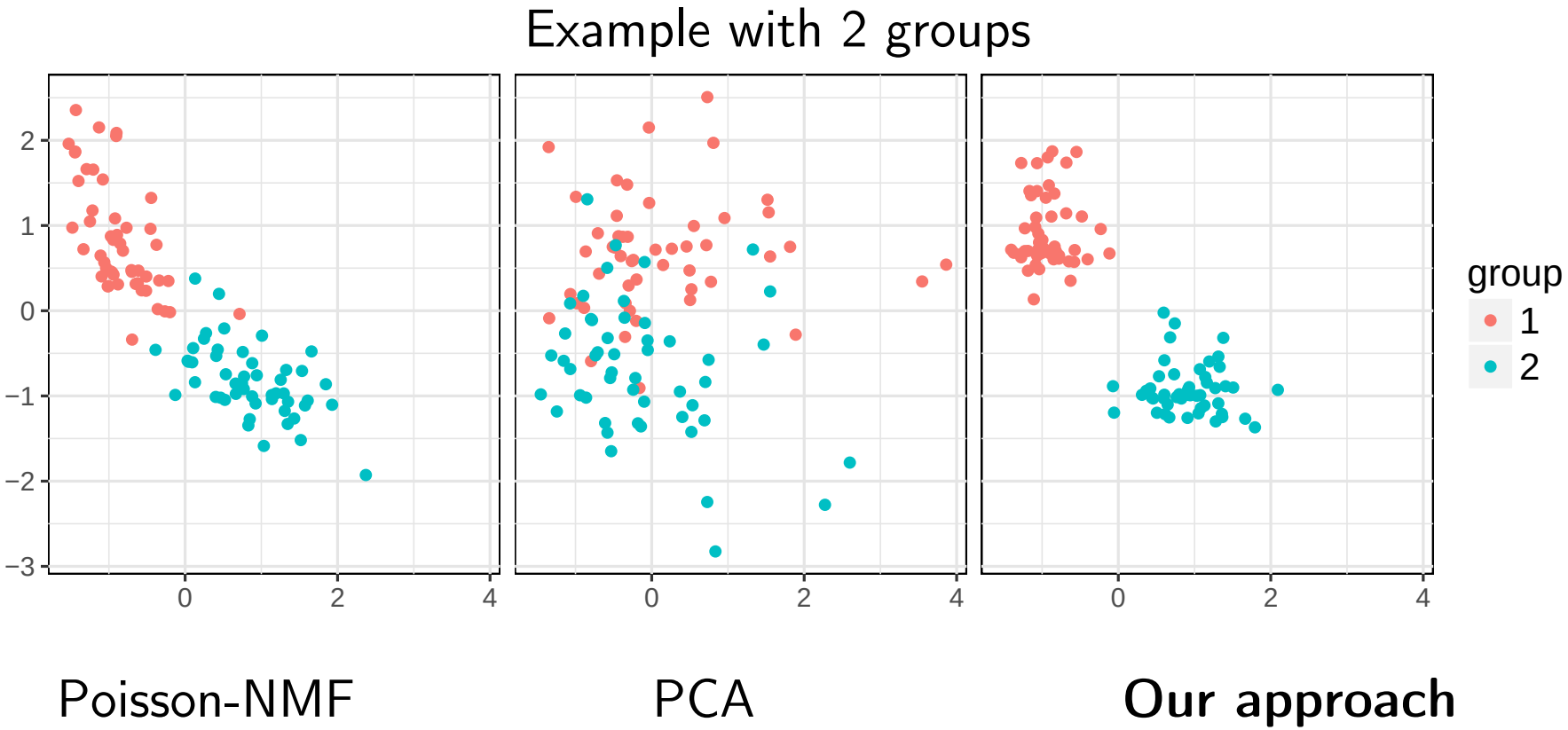


- D_{ij} = dropout indicator
- $\mathbb{P}(X_{ij} = 0 \,|\, \mathbf{U}, \mathbf{V}) > e^{-\lambda_{ij}}$

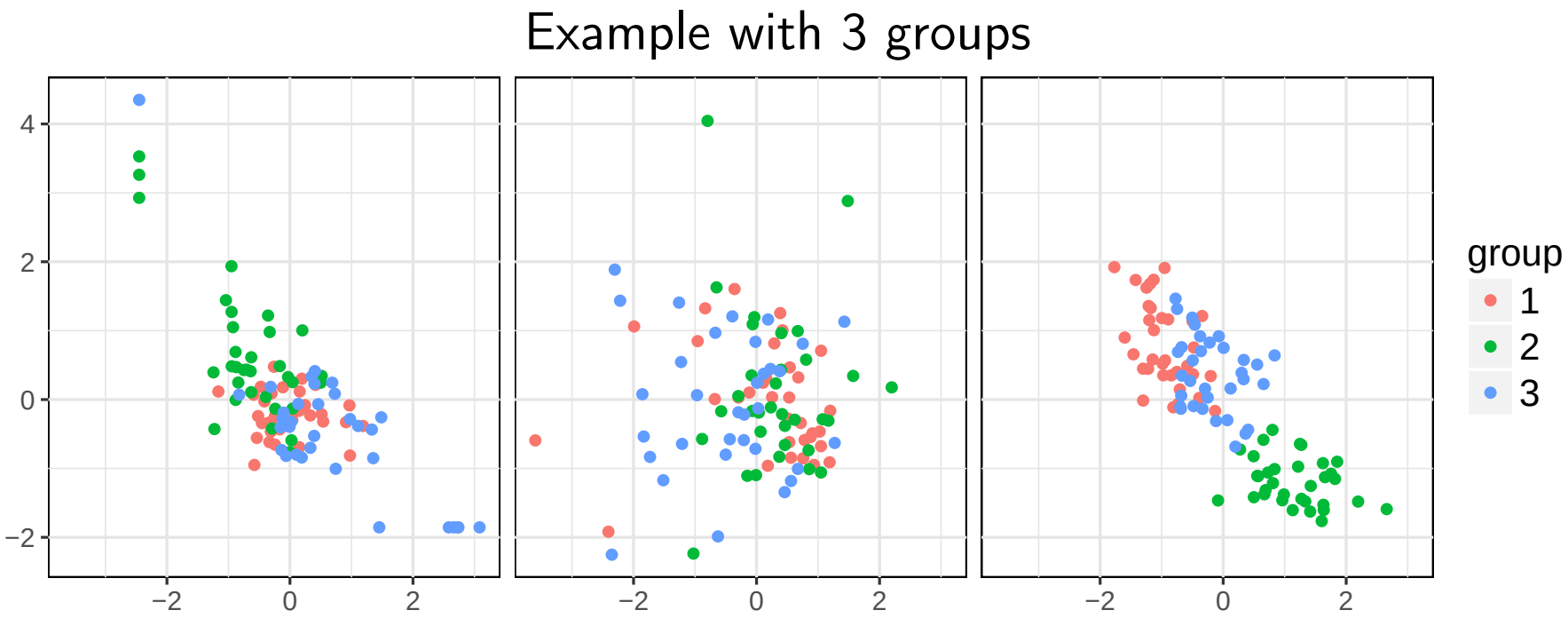
Gamma-Poisson model for matrix factorization

- Suitable for any count data, especially NGS data
- Accounts for
  - → Over-dispersion (Gamma-Poisson model)
  - → Zero-inflation (Poisson-Dirac mixture)
  - → sparsity in $\mathbf{V}$ (Gamma-Dirac mixture)
- Framework of variational inference
- Efficient implementation in `C++`, incorporated in a `R` package `CMF`

Visualization of zero-inflated over-dispersed count data



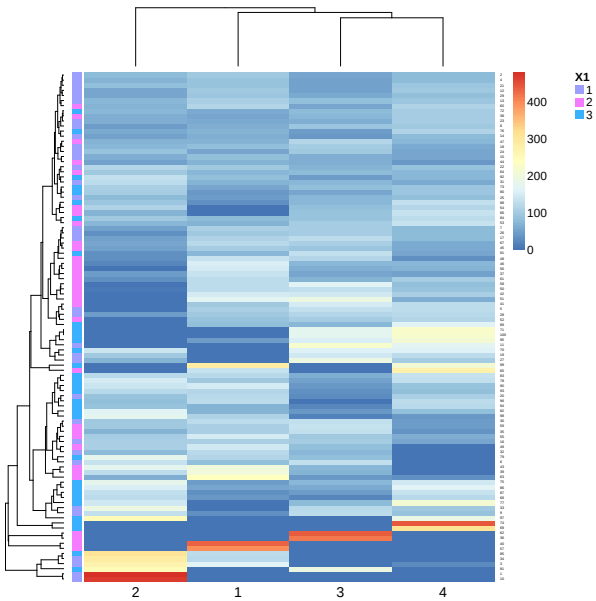
Visualization of zero-inflated over-dispersed count data



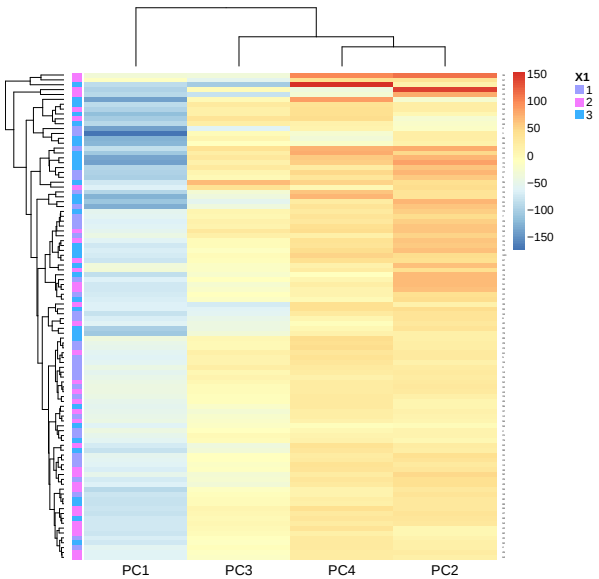
Poisson-NMF PCA **Our approach**

Clustering of the observations according to the matrix $\hat{\mathbf{U}}$

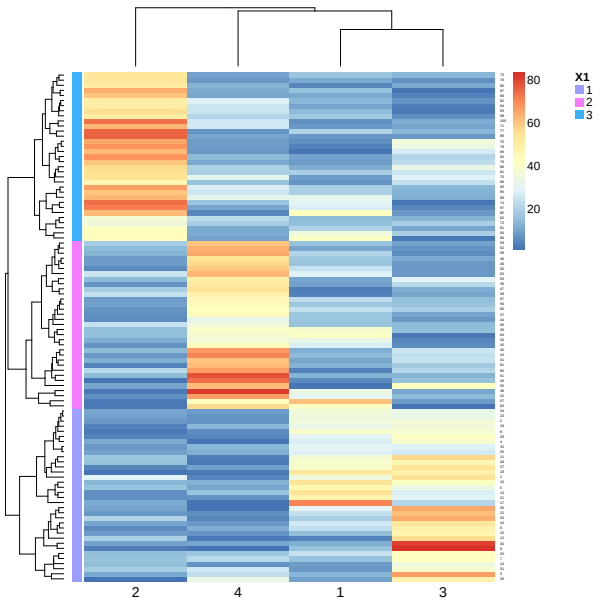
Example with 3 groups



Poisson-NMF



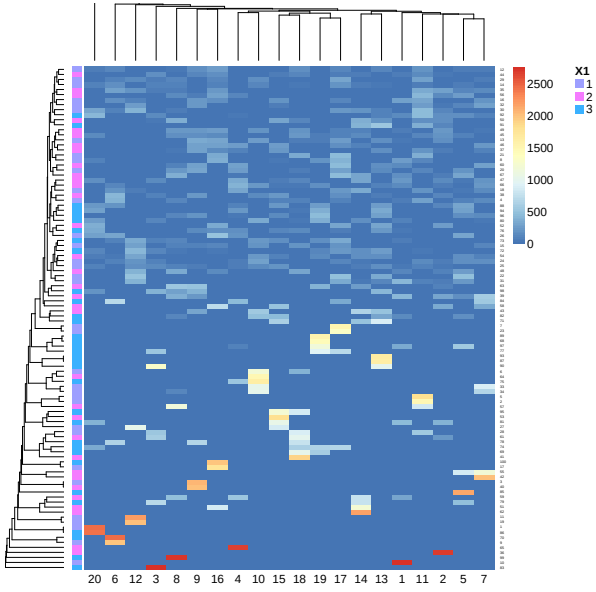
PCA



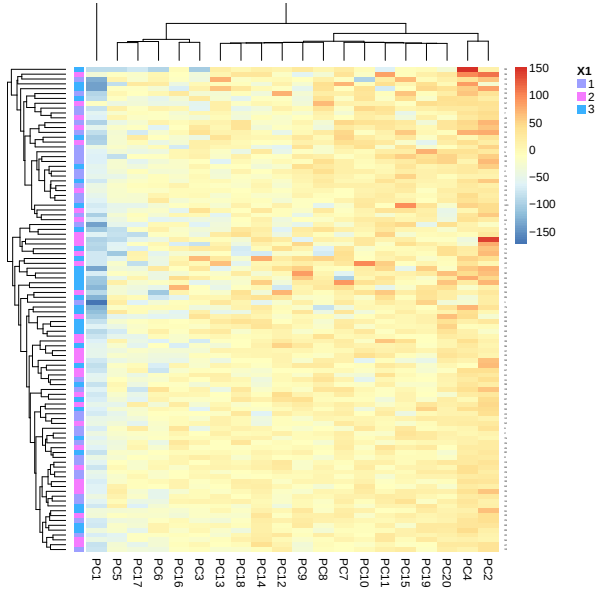
Our approach

Clustering of the observations according to the matrix $\hat{\mathbf{U}}$

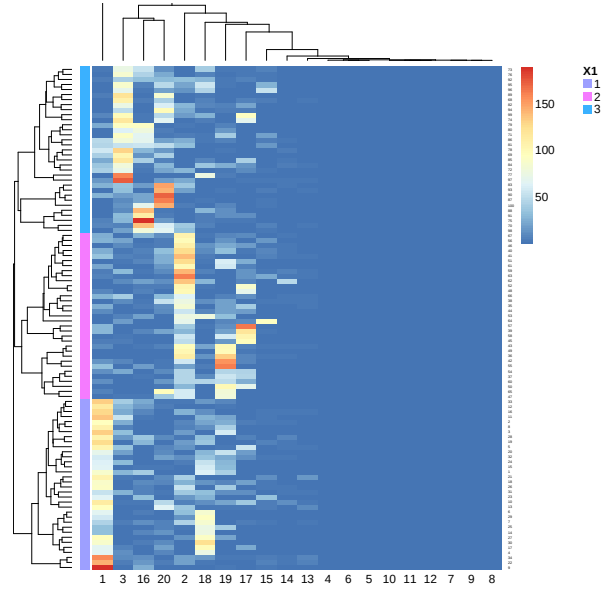
Example with 3 groups



Poisson-NMF



PCA

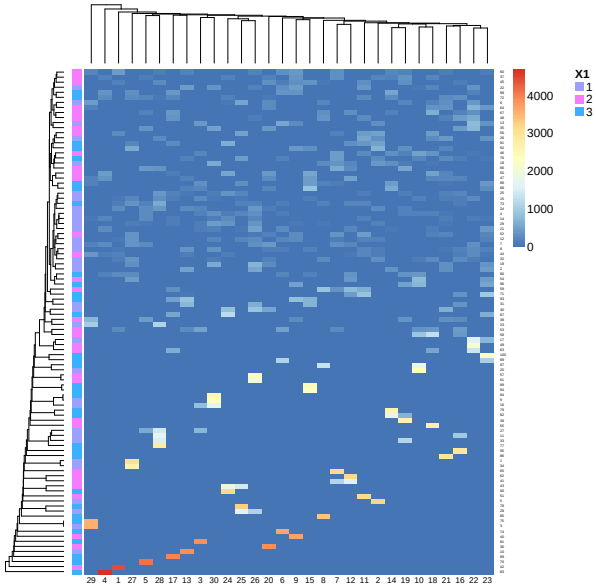


Our approach

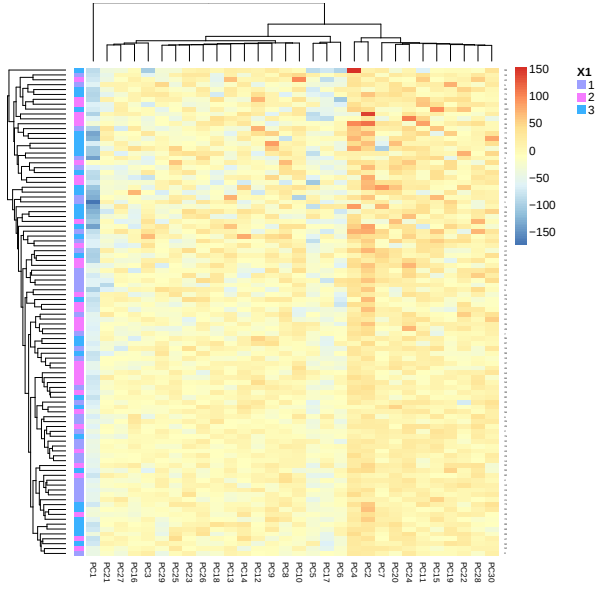
More robust to the choice of K

Clustering of the observations according to the matrix $\hat{\mathbf{U}}$

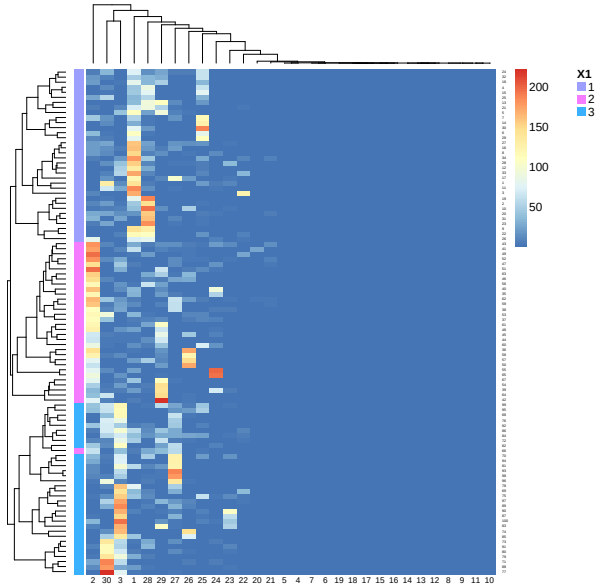
Example with 3 groups



Poisson-NMF



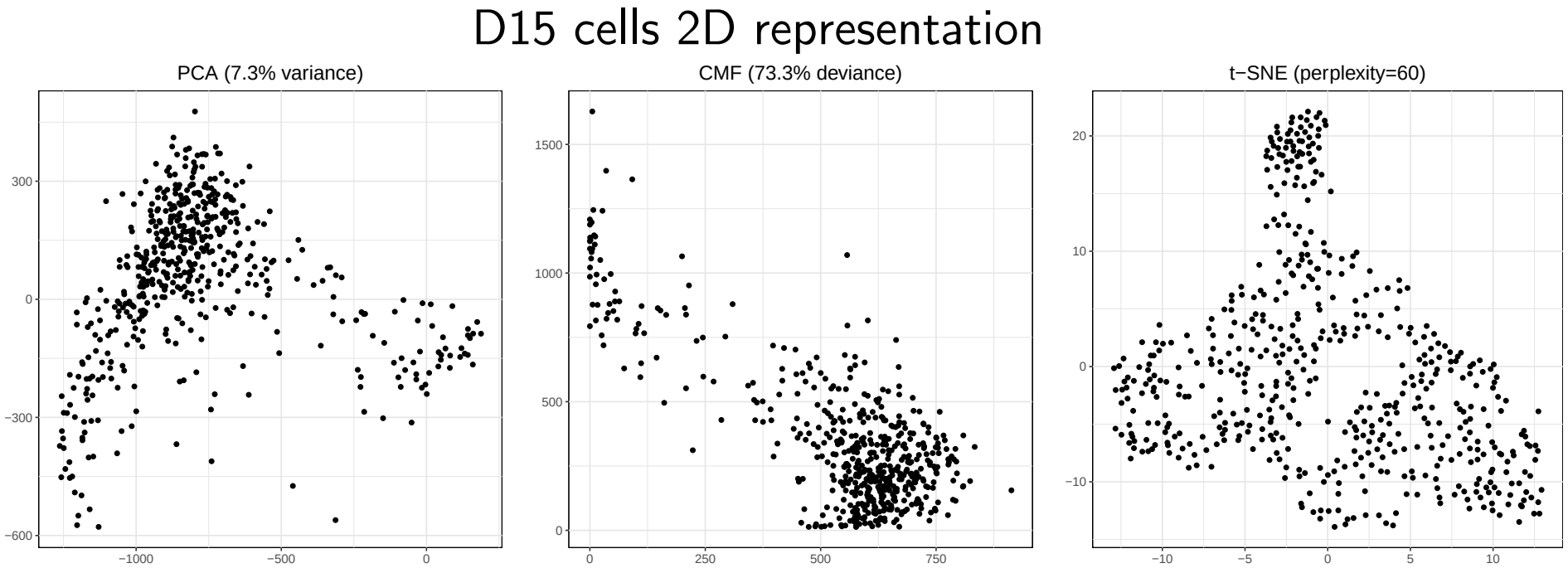
PCA



Our approach

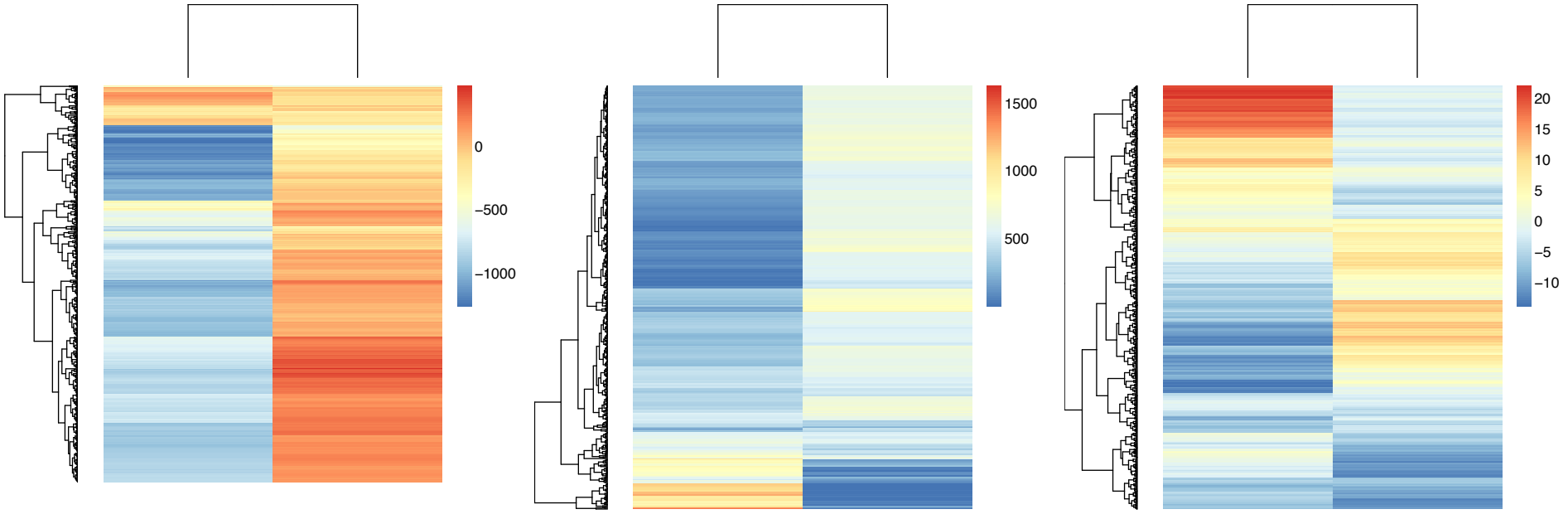
More robust to the choice of K

Quality control to remove the "bad cells"



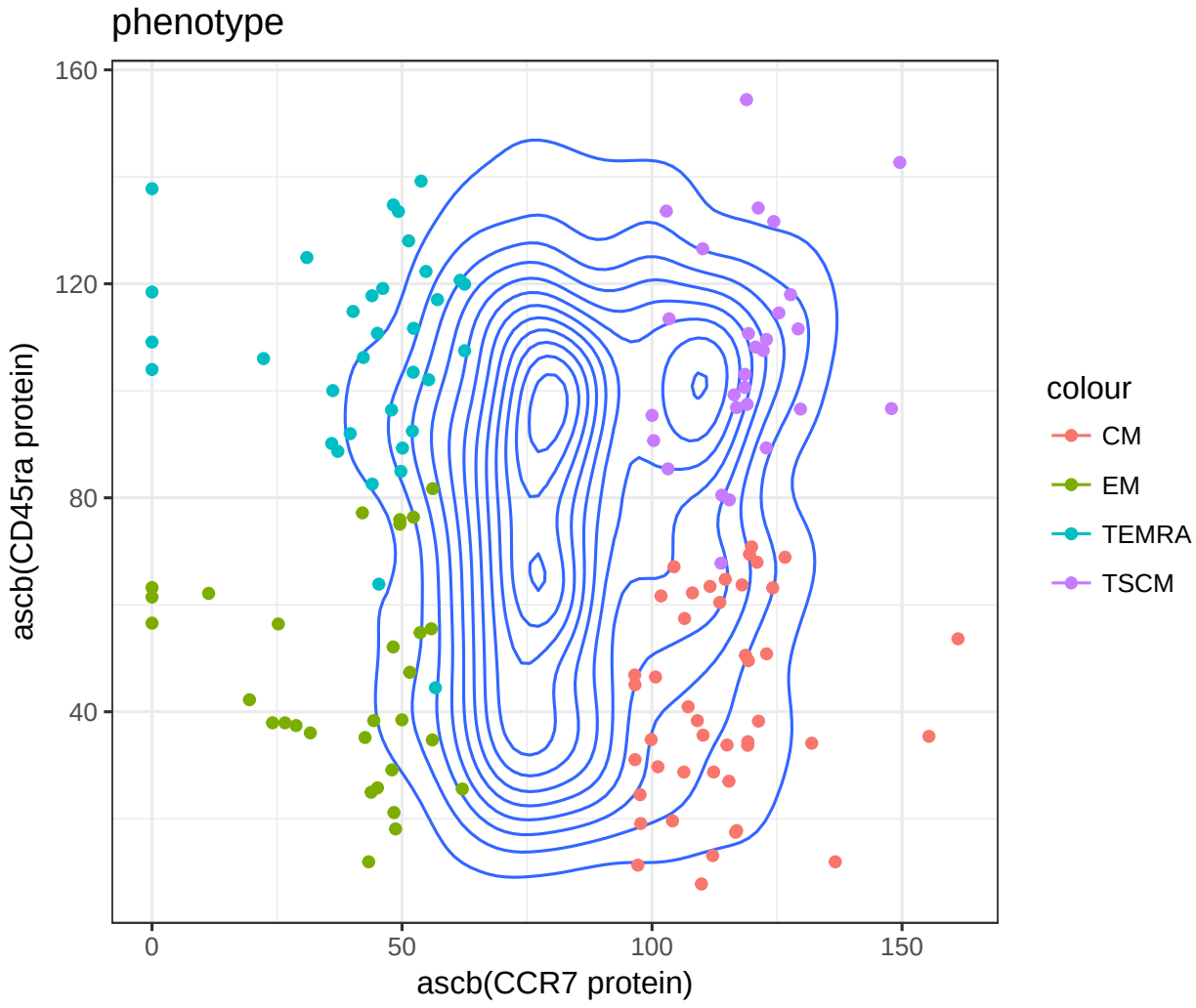
Quality control to remove the "bad cells"

D15 cells 2D representation



65 cells with TCR of poor quality

Effector versus Memory



Can we find effector and memory cells ?

Table of Contents

Supervised analysis of RNA-seq data

Consider labels on RNA-seq samples:

- $\rightarrow$ relate the expression of genes to a disease?
- $\rightarrow$ which genes predict the different types of the cells?

Supervised analysis of RNA-seq data

Consider labels on RNA-seq samples:

$\rightarrow$ relate the expression of genes to a disease?

$\rightarrow$ which genes predict the different types of the cells?

$$\mathbf{X}_{n\times p} = \underbrace{\begin{bmatrix} & & & & & & \\ & & & x_{ij} & & & \\ & & & & & & \end{bmatrix}}_{\textbf{Predictors}} \quad \text{and} \quad \boldsymbol{\xi} = \underbrace{\begin{bmatrix} \vdots \\ \xi_i \\ \vdots \end{bmatrix}}_{\textbf{Response}} \in \mathbb{R}^n$$

Linear regression problem (continuous response): $\xi_i = \mathbf{x}_i^T \boldsymbol{\beta} + \epsilon_i$

$\rightarrow$ find $\boldsymbol{\beta} \in \mathbb{R}^p$

Supervised analysis of RNA-seq data

Consider labels on RNA-seq samples:

$\rightarrow$ relate the expression of genes to a disease?

$\rightarrow$ which genes predict the different types of the cells?

$$\mathbf{X}_{n\times p} = \underbrace{\begin{bmatrix} & & & & & & \\ & & & x_{ij} & & & \\ & & & & & & \end{bmatrix}}_{\textbf{Predictors}} \quad \text{and} \quad \boldsymbol{\xi} = \underbrace{\begin{bmatrix} \vdots \\ \xi_i \\ \vdots \end{bmatrix}}_{\textbf{Response}} \in \mathbb{R}^n$$

Linear regression problem (continuous response): $\xi_i = \mathbf{x}_i^T \boldsymbol{\beta} + \epsilon_i$

$\rightarrow$ find $\boldsymbol{\beta} \in \mathbb{R}^p$

Issue = **high dimension**

Sparse Partial Least Squares regression (Sparse PLS)

Purpose: find latent directions that explain the response

	PCA	**PLS**
Components	$\mathbf{t}_k = \mathbf{X}\mathbf{w}_k \in \mathbb{R}^n$	
Criterion	$\text{Var}(\mathbf{X}\mathbf{w}_k)$	$\text{Cov}(\mathbf{X}\mathbf{w}_k, \boldsymbol{\xi})$

Sparse Partial Least Squares regression (Sparse PLS)

Purpose: find latent directions that explain the response

	PCA	PLS
Components	$\mathbf{t}_k = \mathbf{X}\mathbf{w}_k \in \mathbb{R}^n$	
Criterion	$\text{Var}(\mathbf{X}\mathbf{w}_k)$	$\text{Cov}(\mathbf{X}\mathbf{w}_k, \boldsymbol{\xi})$

Penalized covariance maximization:

$$\begin{cases} \underset{\mathbf{w}\in\mathbb{R}^p}{\text{argmin}} \left\{ -\mathbf{w}^T \mathbf{X}_c^T \boldsymbol{\xi}_c + \lambda_S \|\mathbf{w}\|_1 \right\} \\ \|\mathbf{w}\|_2 = 1 \end{cases}$$

$\rightarrow$ to select the genes

Our approach logit-SPLS

1. Ridge IRLS algorithm (Eilers et al., 2001)

   $\rightarrow$ **Ensure the convergence**

2. Estimate $\boldsymbol{\beta}$ with **adaptive sparse PLS** regression of $\boldsymbol{\xi}$ over $\mathbf{X}$

   $\rightarrow$ sparse dimension reduction

Our approach logit-SPLS

1. Ridge IRLS algorithm (Eilers et al., 2001)

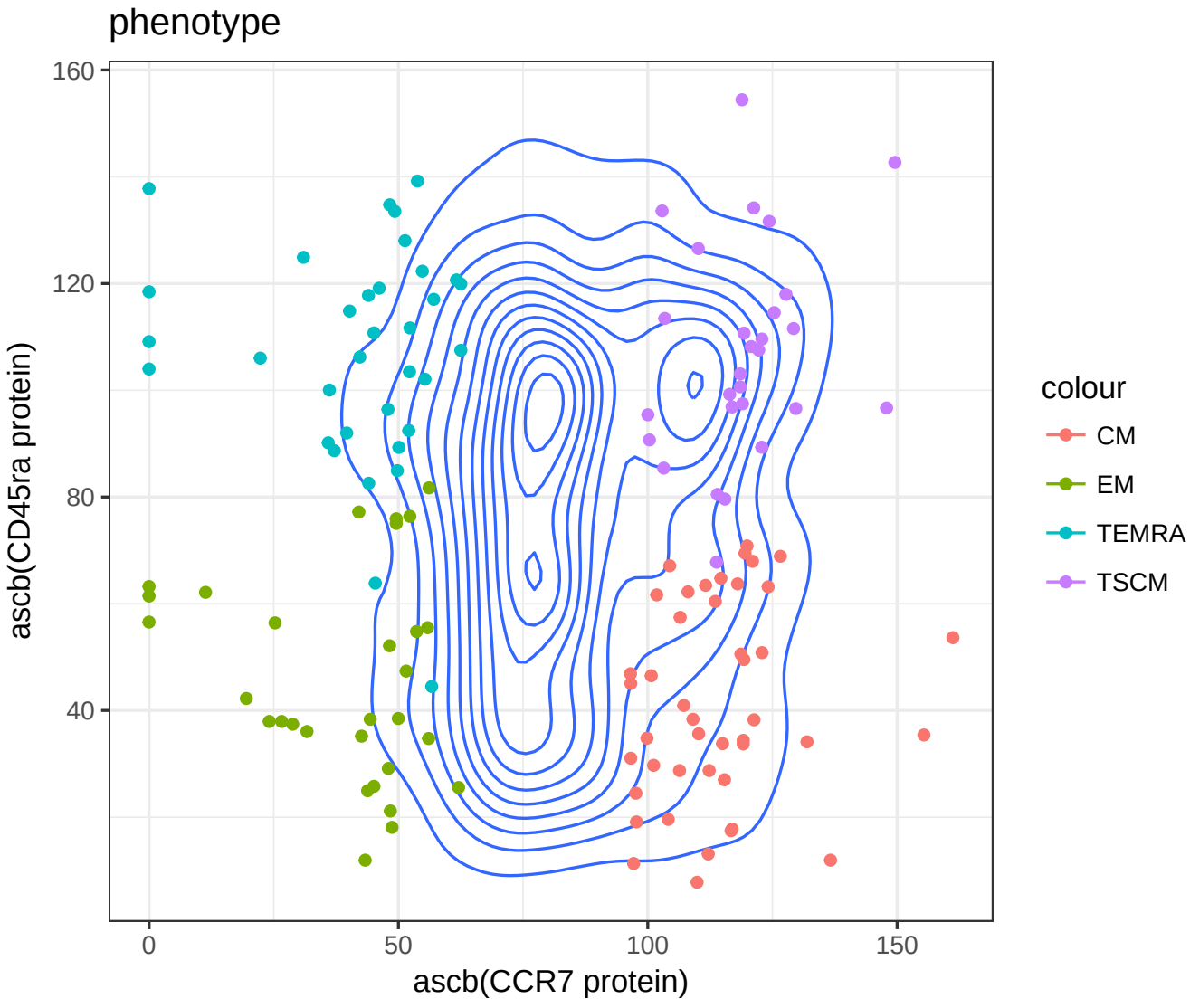
   $\rightarrow$ **Ensure the convergence**

2. Estimate $\boldsymbol{\beta}$ with **adaptive sparse PLS** regression of $\boldsymbol{\xi}$ over $\mathbf{X}$

   $\rightarrow$ sparse dimension reduction

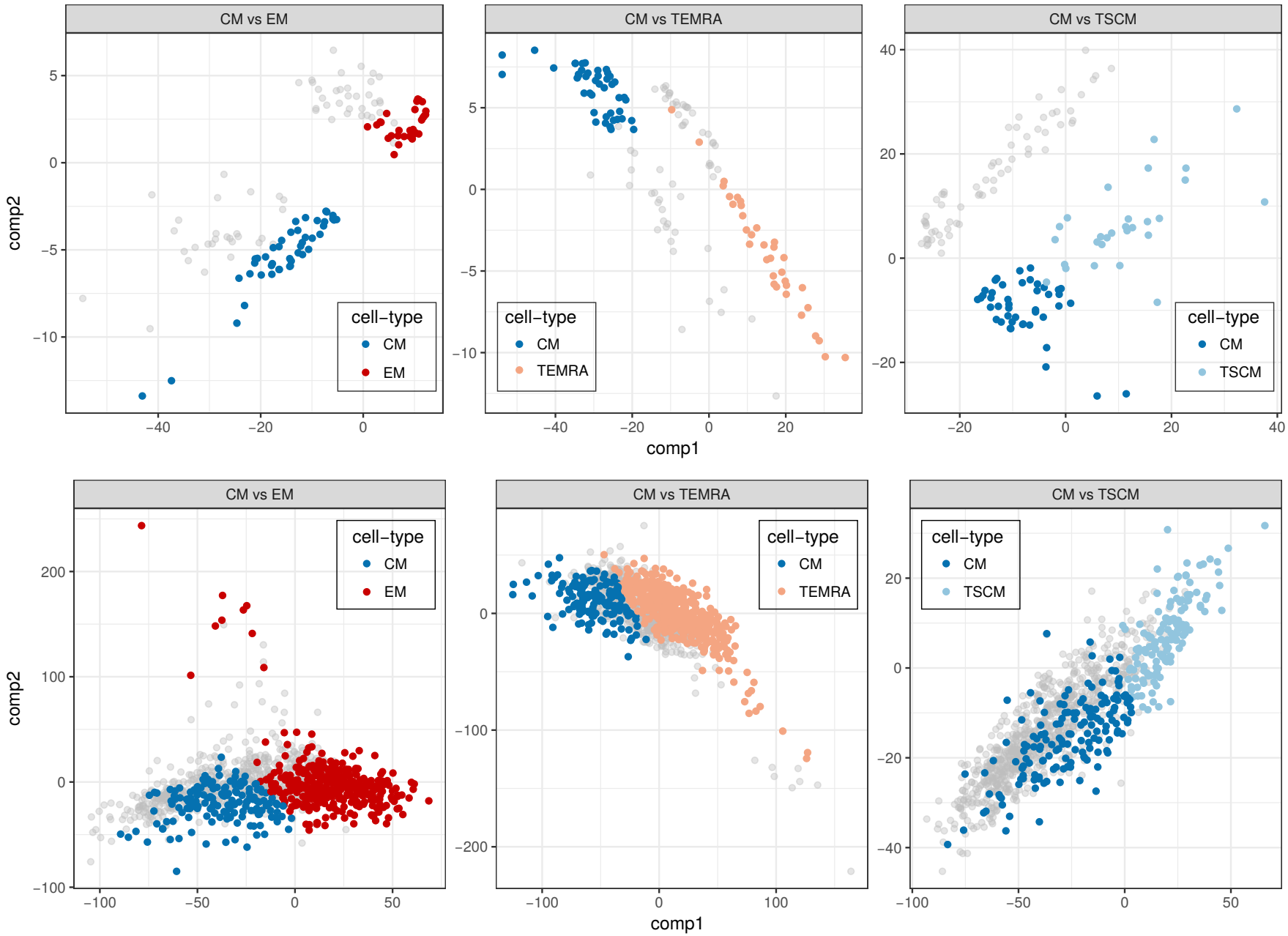
- Performance in prediction and selection accuracy similar or better to state-of-the-art approaches
- Fast convergence of the algorithm (contrary to other sparse PLS based approaches)
- Calibration of λ_S
  - cross-validation is more precise
  - stability selection (Meinshausen and Bühlmann, 2010)

Effector versus Memory

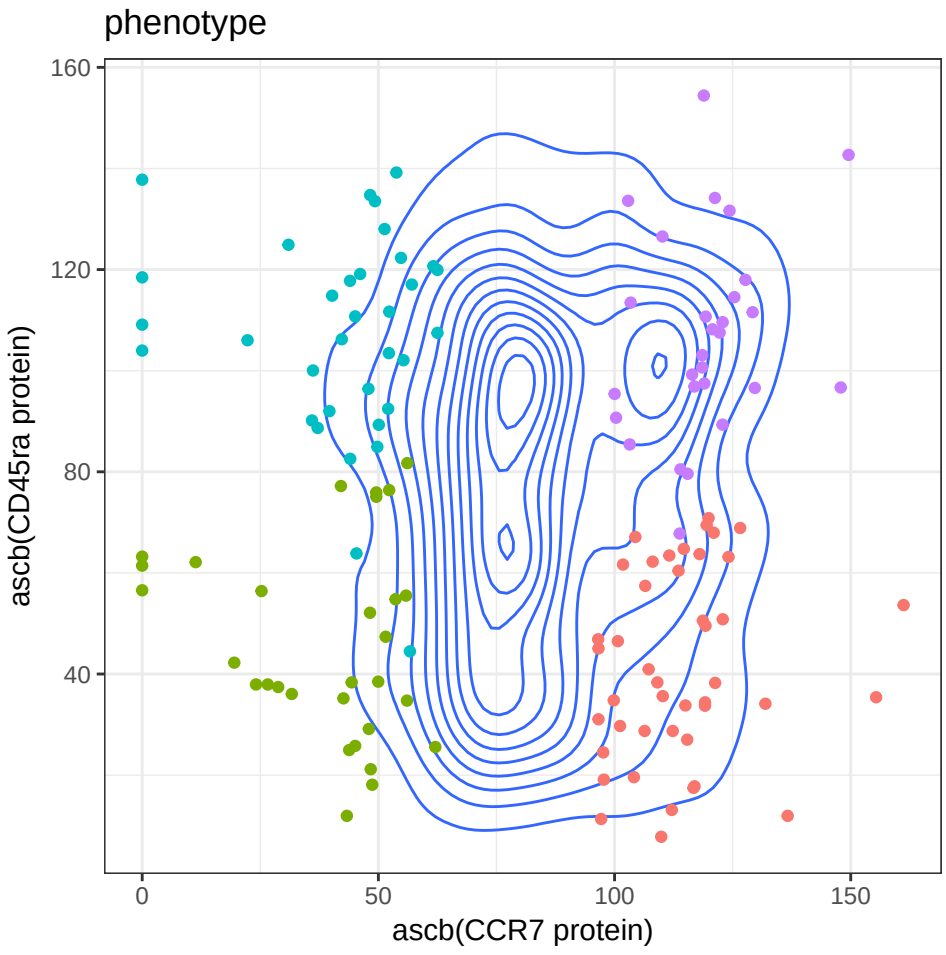


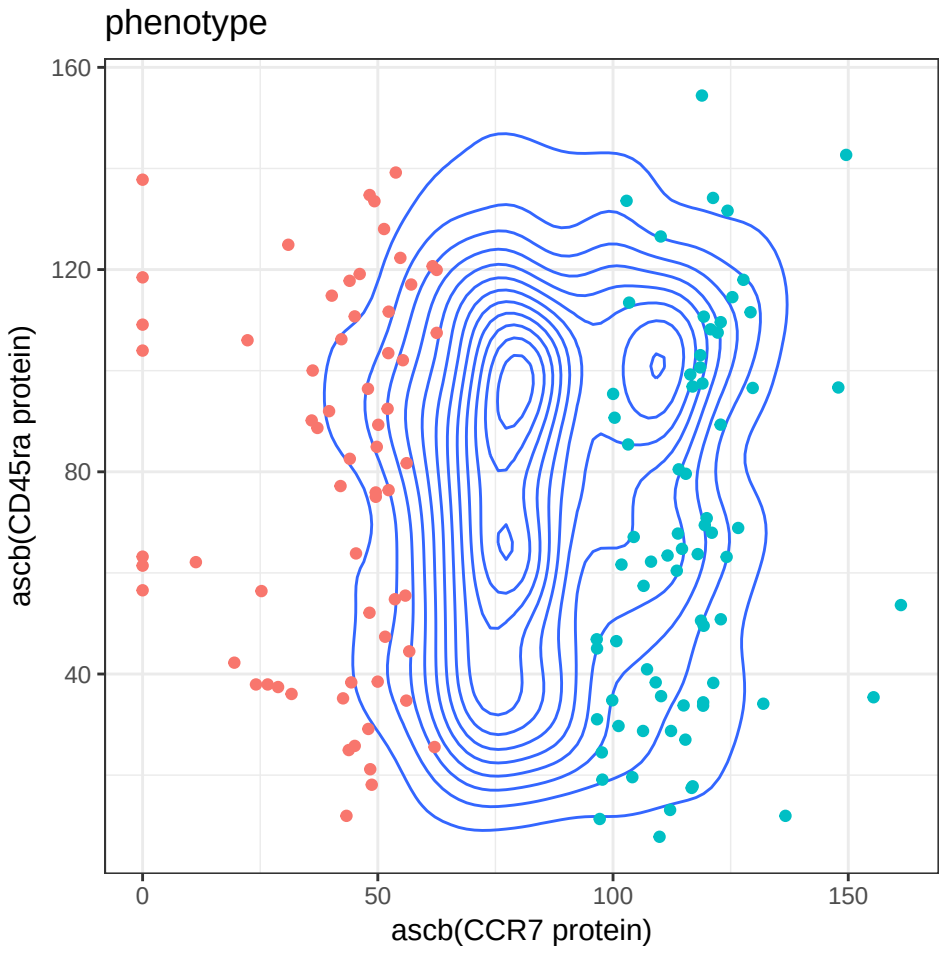
train on 11 cellular markers and corresponding genes

Effector versus Memory

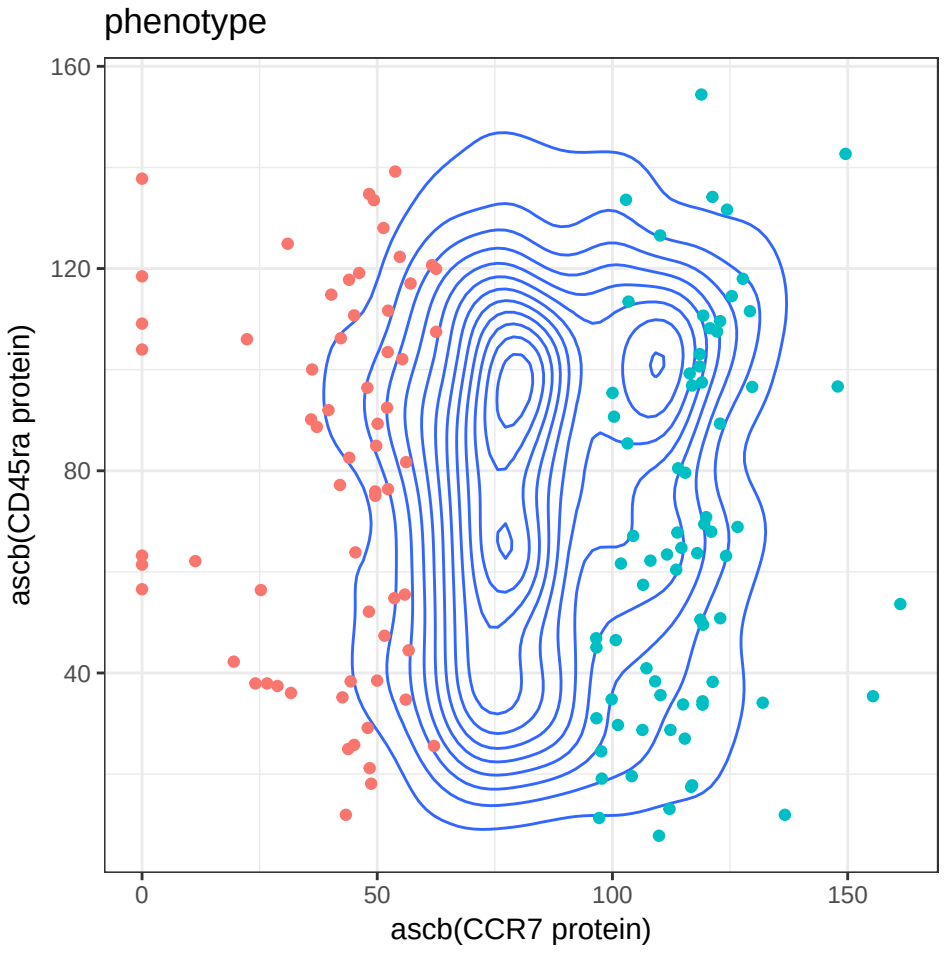


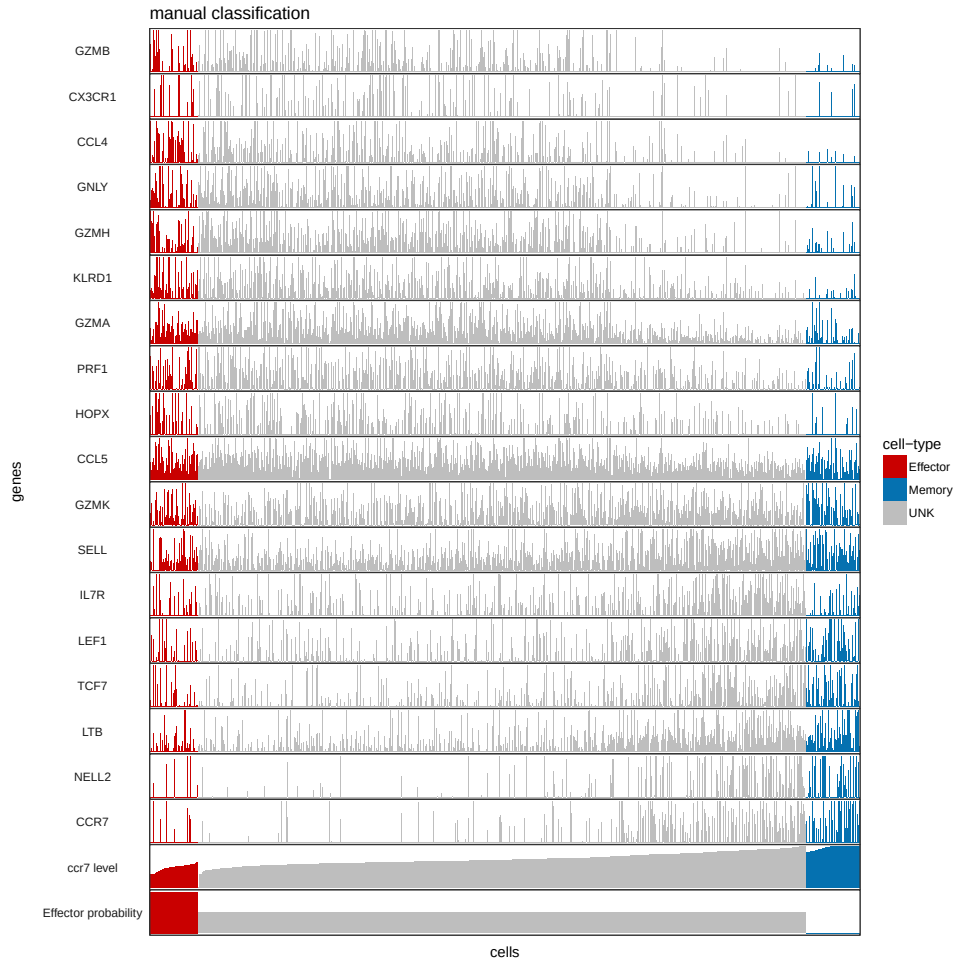
Effector versus Memory





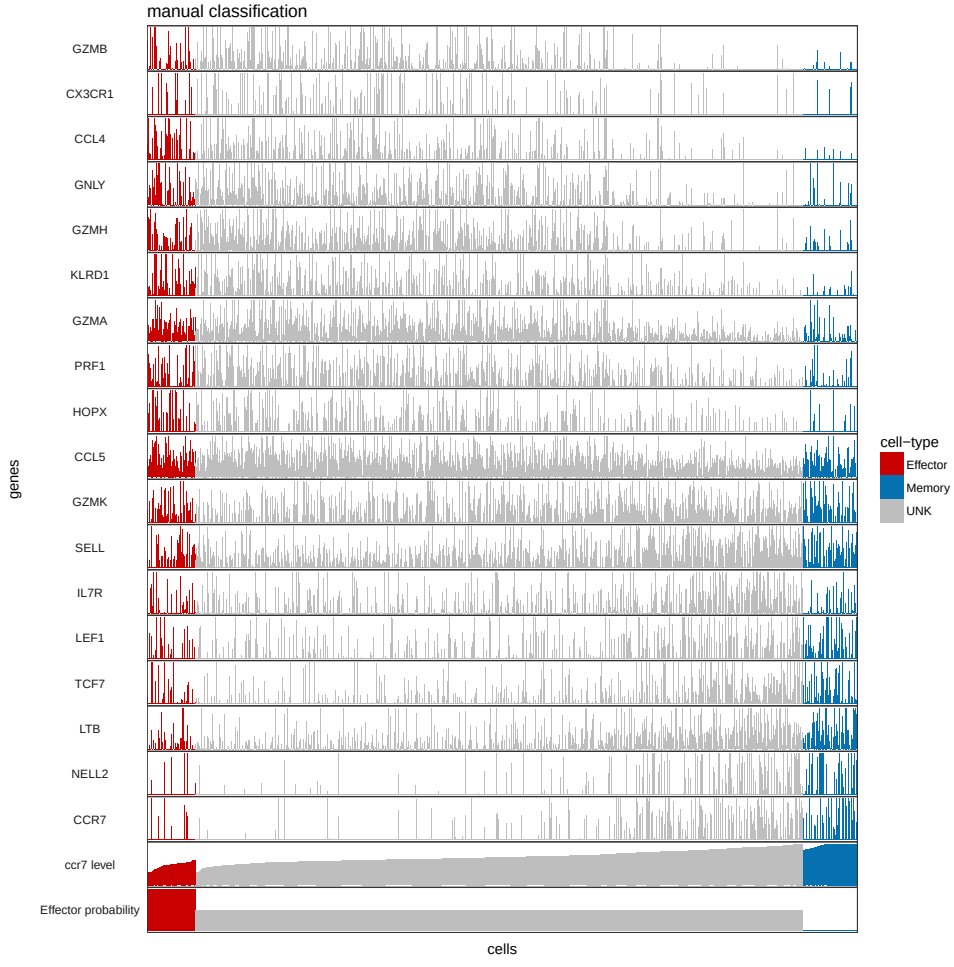
Effector versus Memory

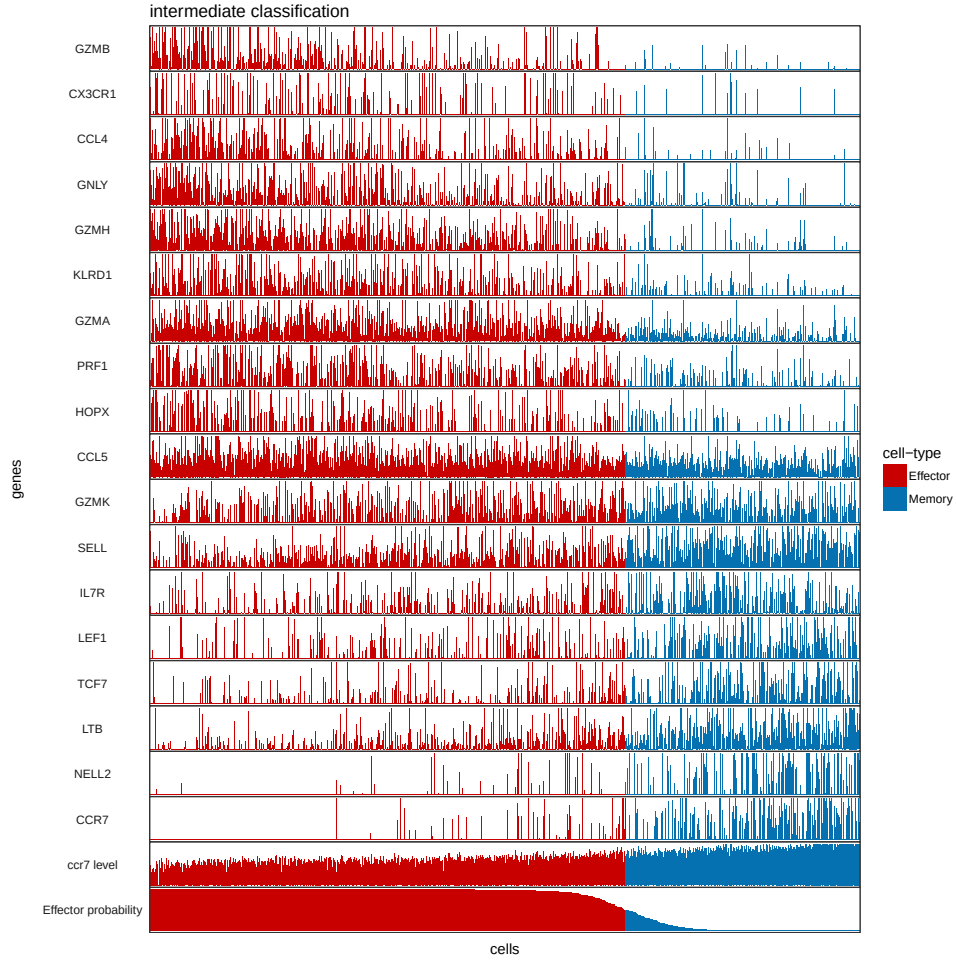




train on 11 cellular markers and corresponding genes

Effector versus Memory

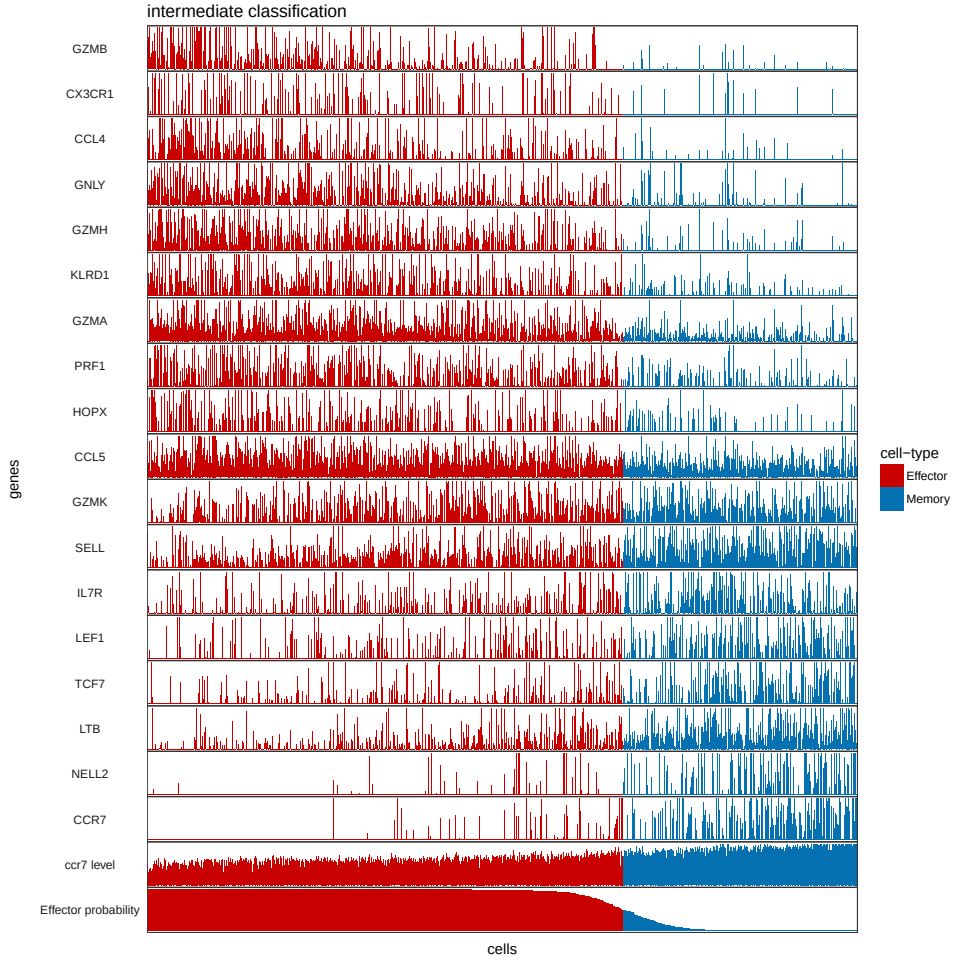


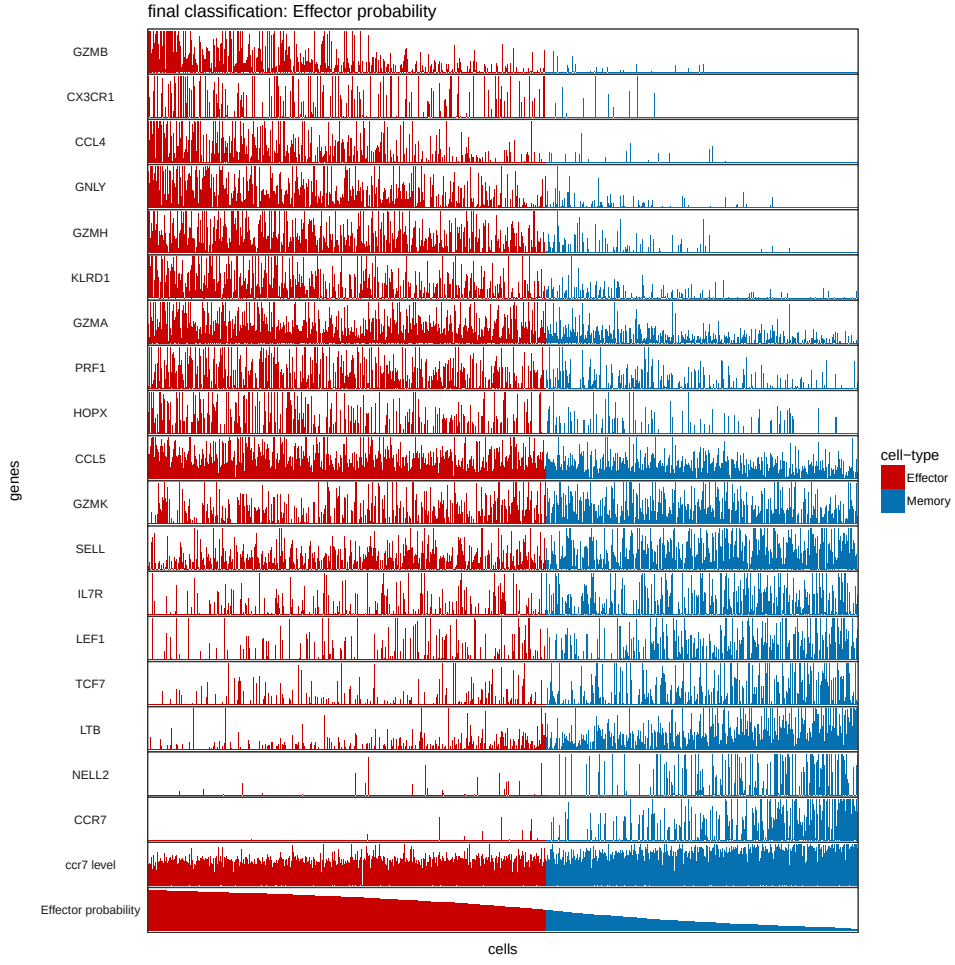


predict effector and memory groups

DEA on group effect for each time-points

Effector versus Memory





train on the 64 DE genes of the intersect between time-points
predict effector and memory groups

Questions

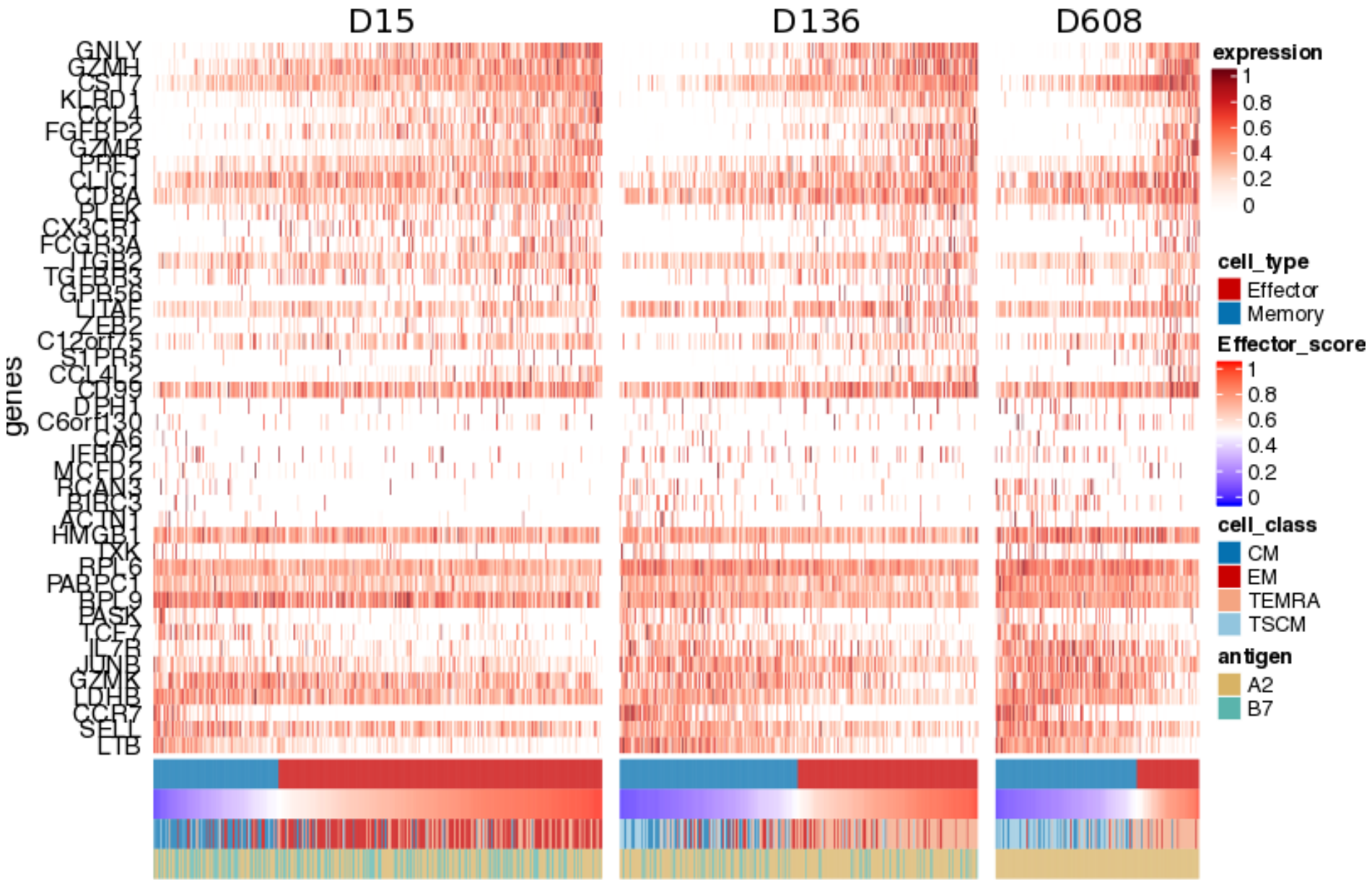
- **Can we find effector and memory cells ?**

  We have a continuum between cells that are more effector and cells that are more memory

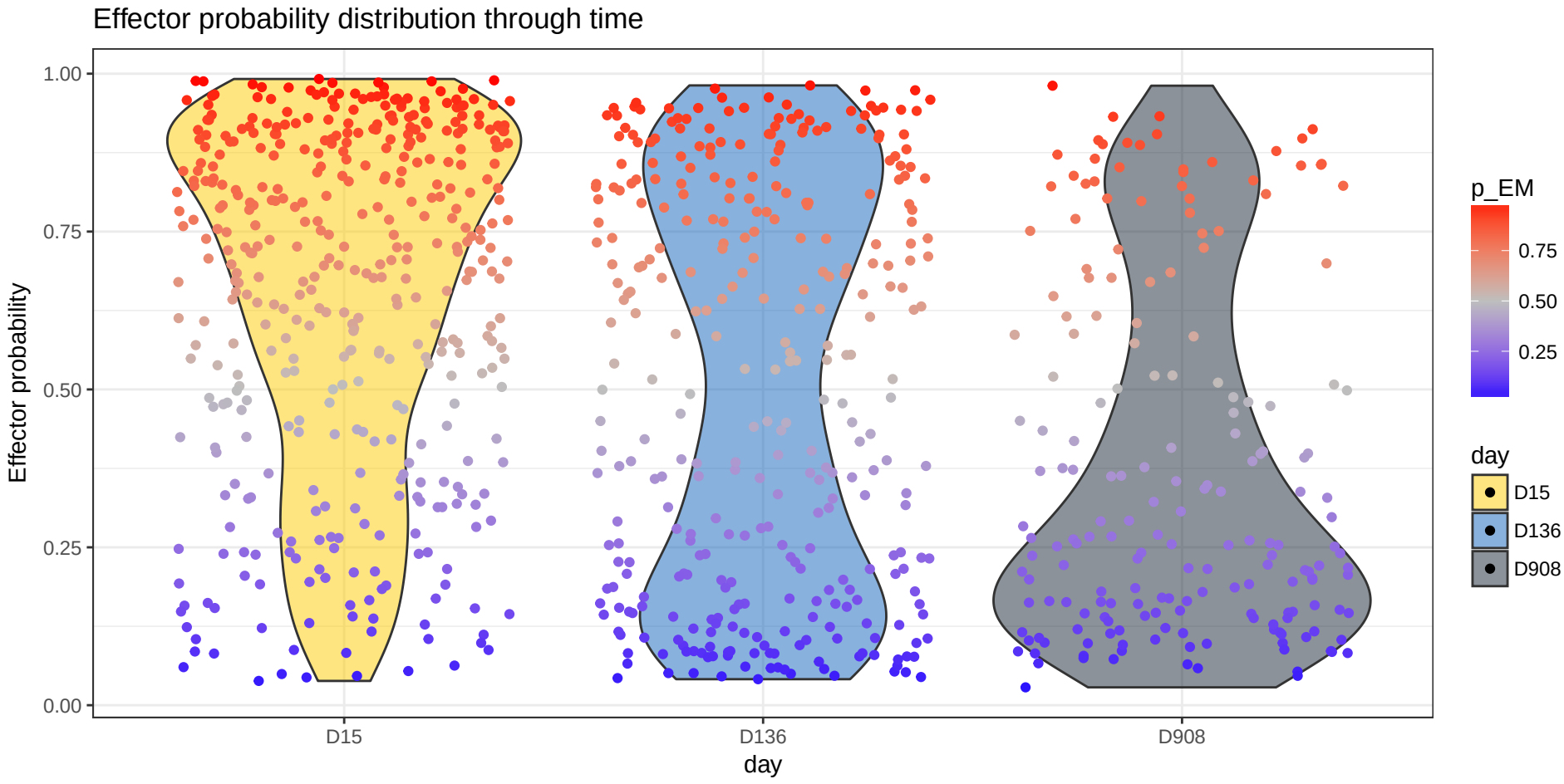
- Are they effector clones and memory clones ?

Effector versus Memory through time

DEA for cell-type effect accounting for batch effect at each time-point.



Cell-type identity through time



The proportion of memory cells increase with time.

Questions

- **Can we find effector and memory cells ?**

  We have a continuum between cells that are more effector and cells that are more memory.
  The proportion of memory cells increases with time.

- Are they effector clones and memory clones ?

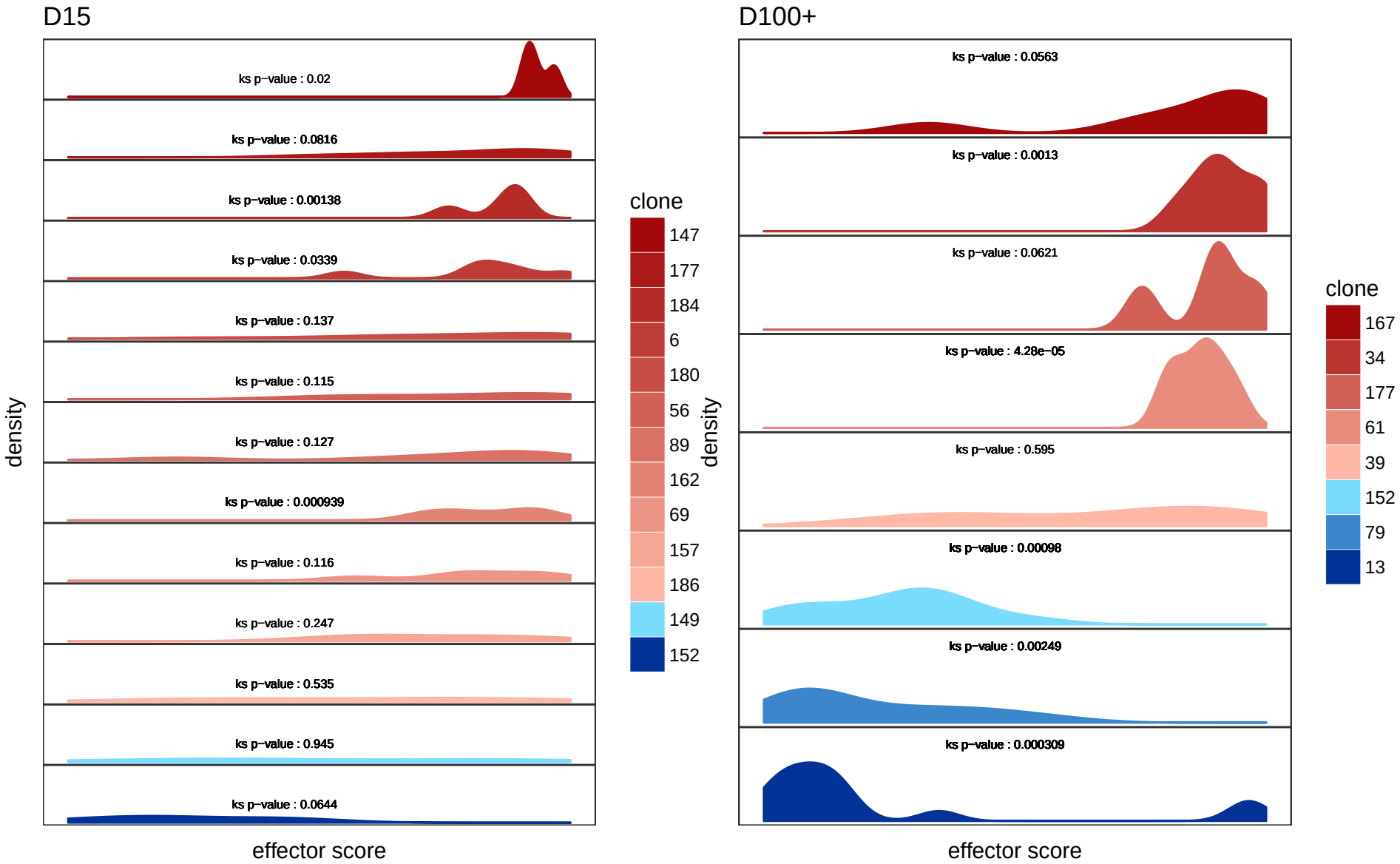
Questions

- Can we find effector and memory cells ?

  We have a continuum between cells that are more effector and cells that are more memory.
  The proportion of memory cells increases with time.

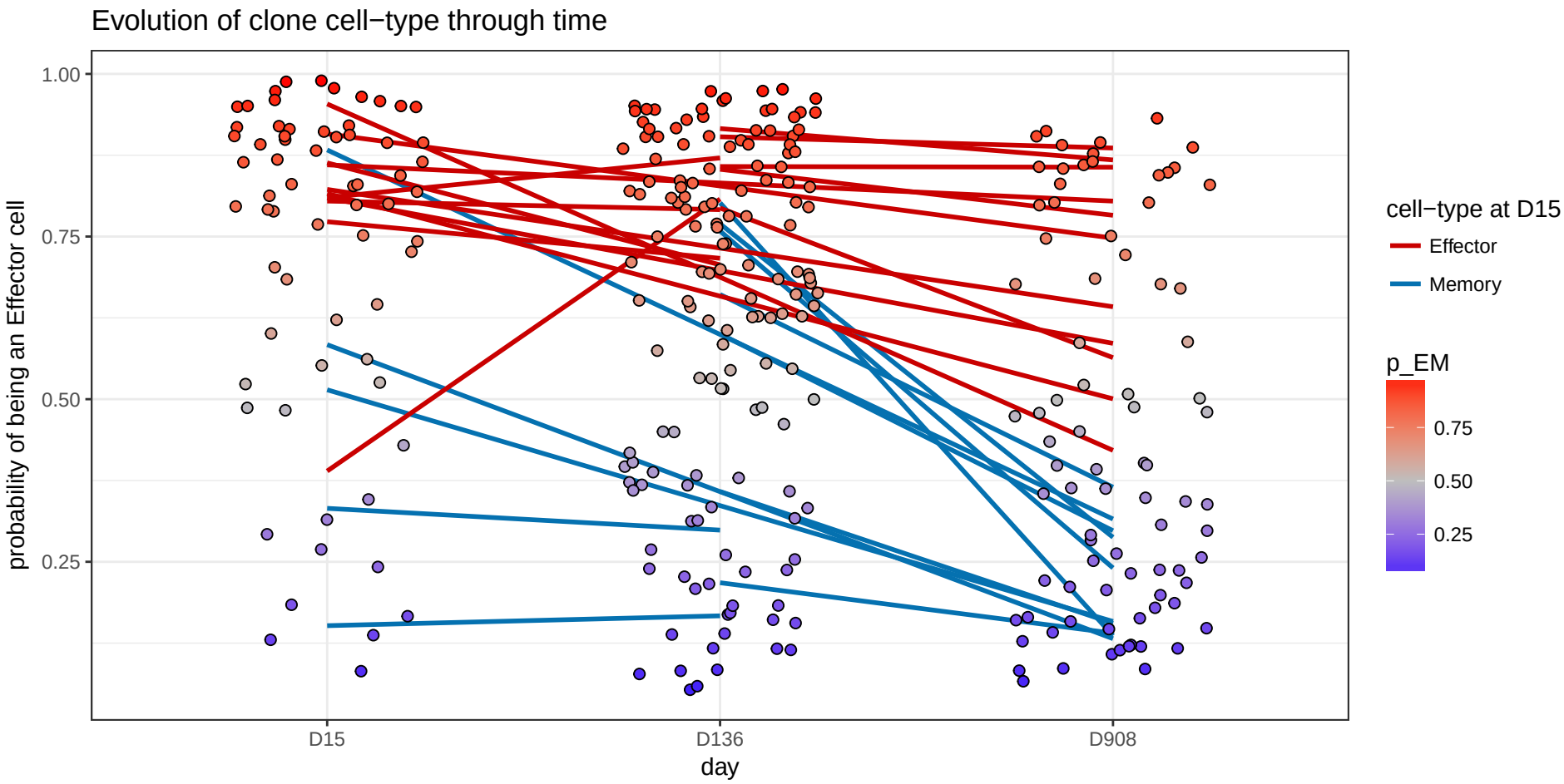
- **Are they effector clones and memory clones ?**

Cell-type identity of clones



There is a full range of clone cell-type identity

Cell-type identity of clones through time



While the proportion of memory cells increase with time, clones tend to keep their identity.

Questions

- Can we find effector and memory cells ?

  We have a continuum between cells that are more effector and cells that are more memory.
  The proportion of memory cells increases with time.

- **Are they effector clones and memory clones ?**

  Yes, but also memory-effector clones.

Take-home message

Count Matrix Factorization (CMF): Data exploration (unsupervized)

- zero-inflated over-dispersed counts
- Variables selection (sparcity on $\hat{\mathbf{V}}$)
- Interpretability of components (clustering on $\hat{\mathbf{U}}$)
- Efficient implementation in `C++`, incorporated in a `R` package `CMF`

Sparse multinomial PLS: Prediction (supervized)

- discrete response
- Variables selection (genes selection)
- Stability of the procedure (reproducibility, cross validation, ...)
- `R` package `plsgenomics`

In the future

Count Matrix Factorization (CMF)

- Model selection criterion (choice of K)
- Stochastic procedure to improve the optimization
- Extension to account for covariates in the model

Sparse multinomial PLS

- Efficient implementation in `C++`, incorporated in a `R` package `CMF`

Acknowledgment

project "*Investissement d'Avenir*"
ABS4NGS

Karolinska Institutet

You for your attention !

sPLS: https://arxiv.org/abs/1502.05933
cran: `plsgenomics`
CMF: `ghislain.durif@inria.fr`