

Announcement

- ★ **Analysis assignment due Feb 28**
- ★ Optional midcourse feedback
(bonus point) due March 1
- ★ Discussion session for your project
on March 6, 12 - 1 PM

Statistical Methods for High-dimensional Biology



Functional interpretation of genomics analysis by enrichment analysis

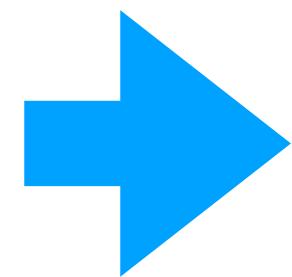
Yongjin Park, UBC Path&Lab, STAT, BC Cancer

Today's lecture: Enrichment Analysis

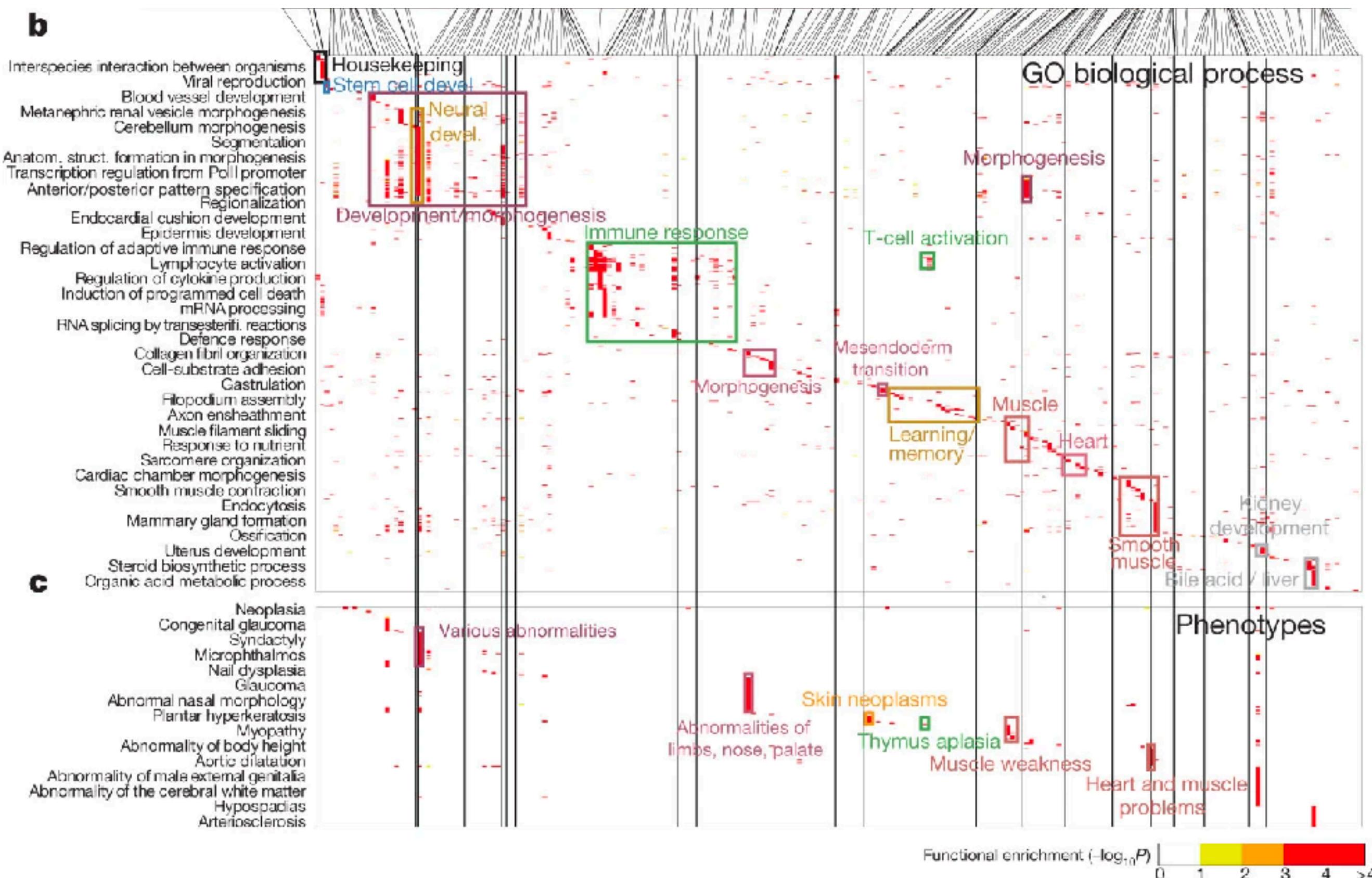
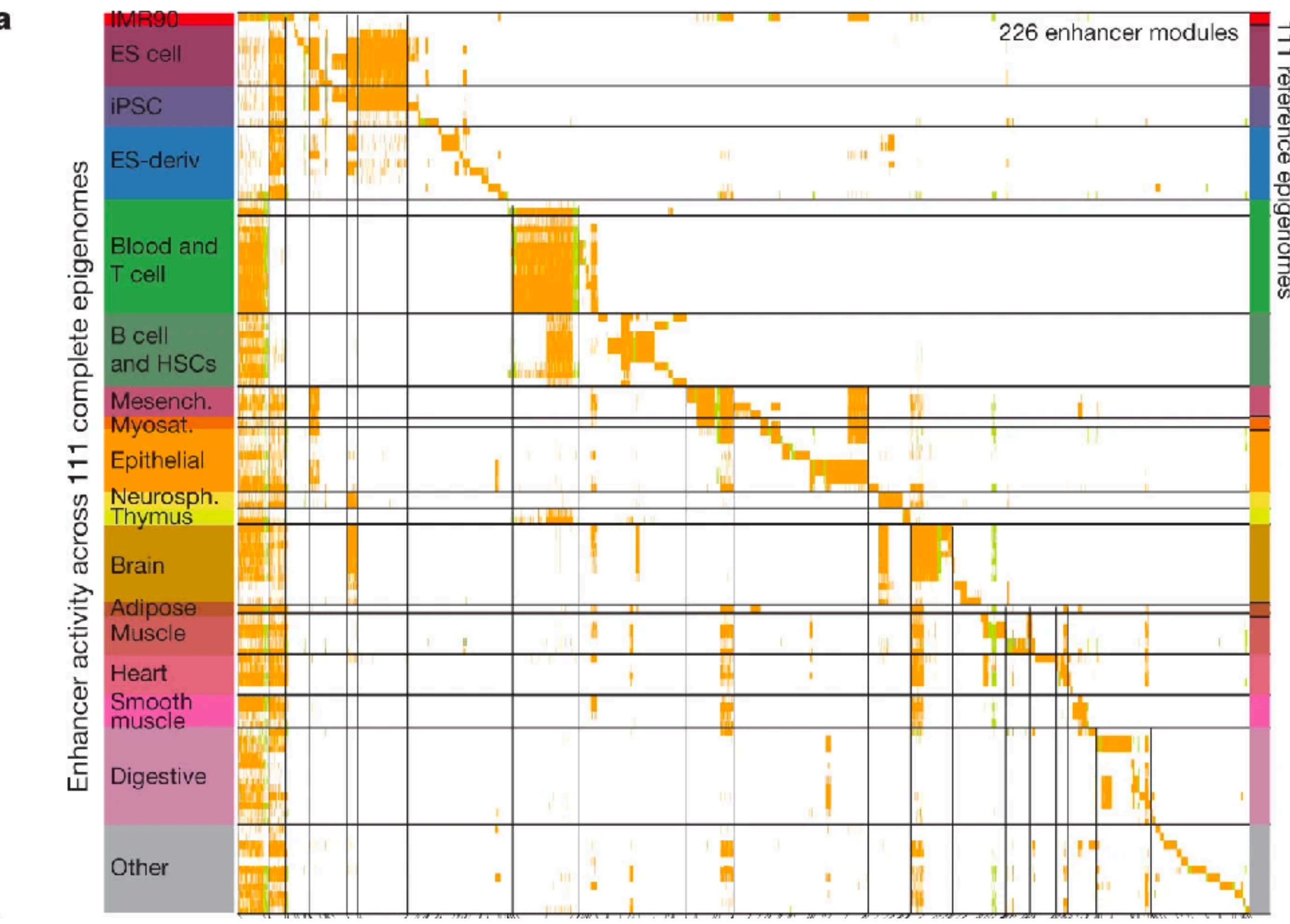
- **Motivations: What's next after genomics analysis?**
 - What have we learned?
 - How do we know that our discovery is meaningful?
- **Gene set enrichment analysis**
 - Set-based approach: Hypergeometric test
 - Rank-based approach: GSEA by KS statistic
- **Can we engineer new gene sets?**
 - Principal Component Analysis
 - Matrix factorization of count data

Biological mechanisms/processes are modular

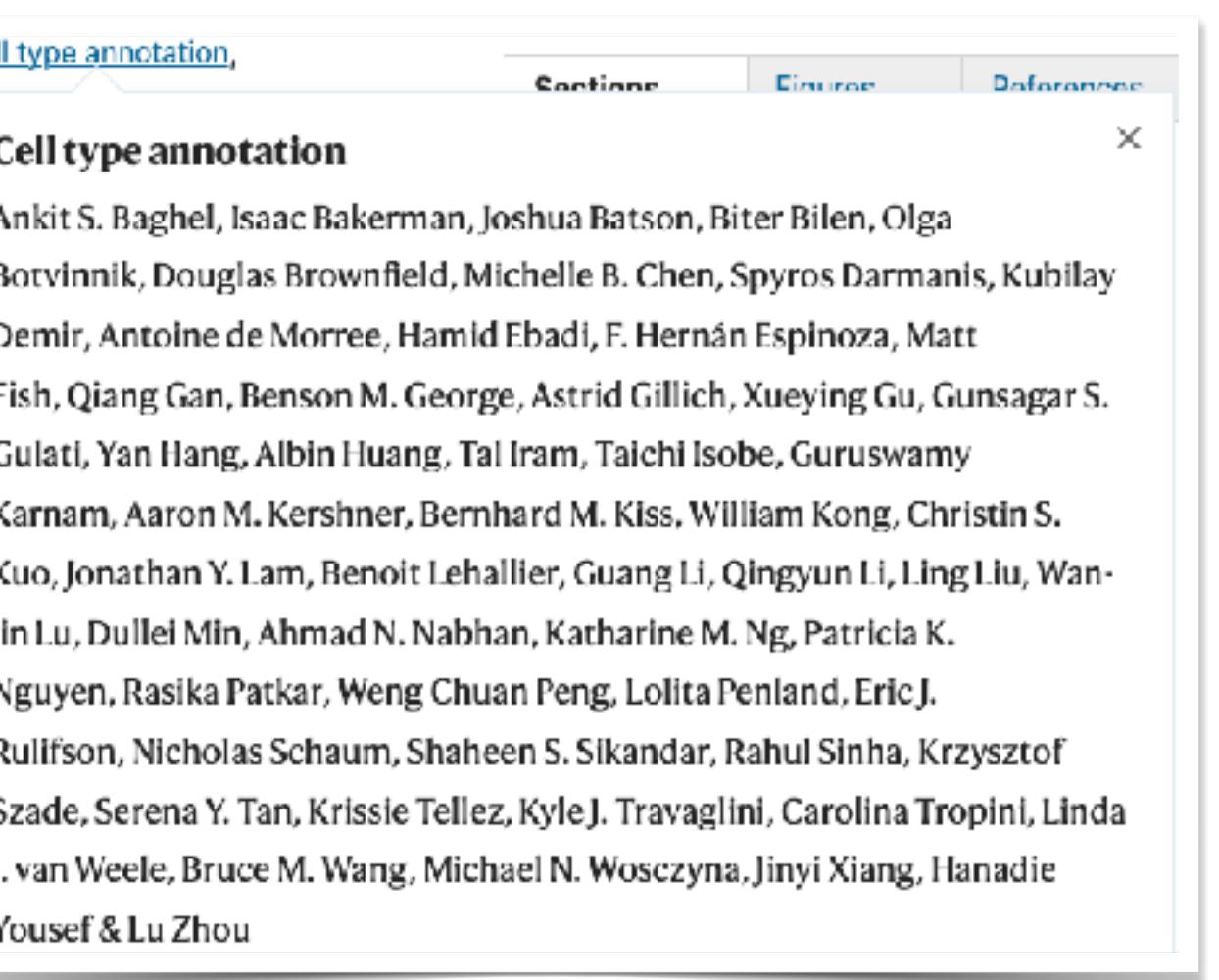
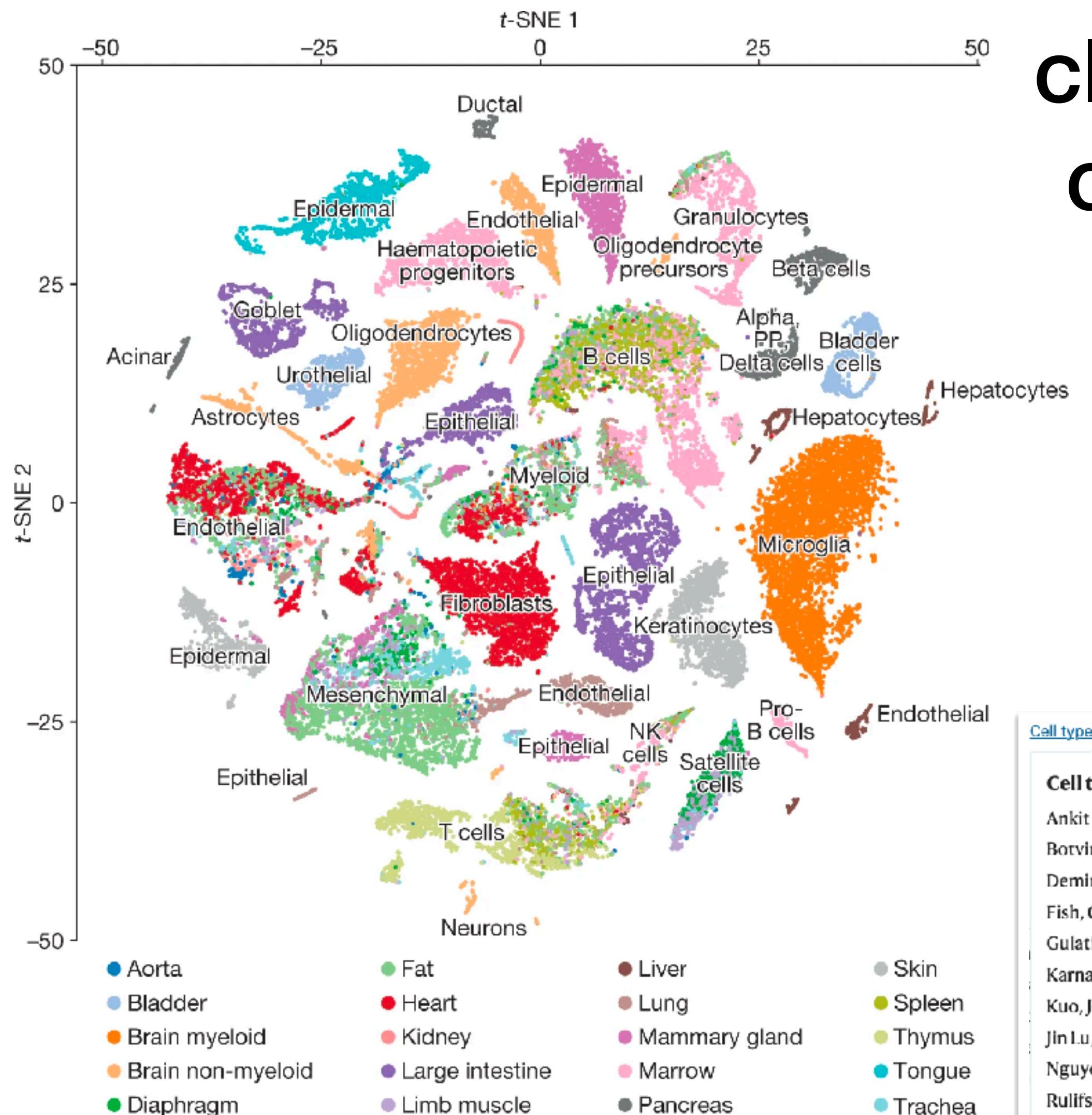
Epigenomic (enhancer element) modules across 111 tissues



Genie ontology information confirms cell-type- or tissue-specific biological pathways



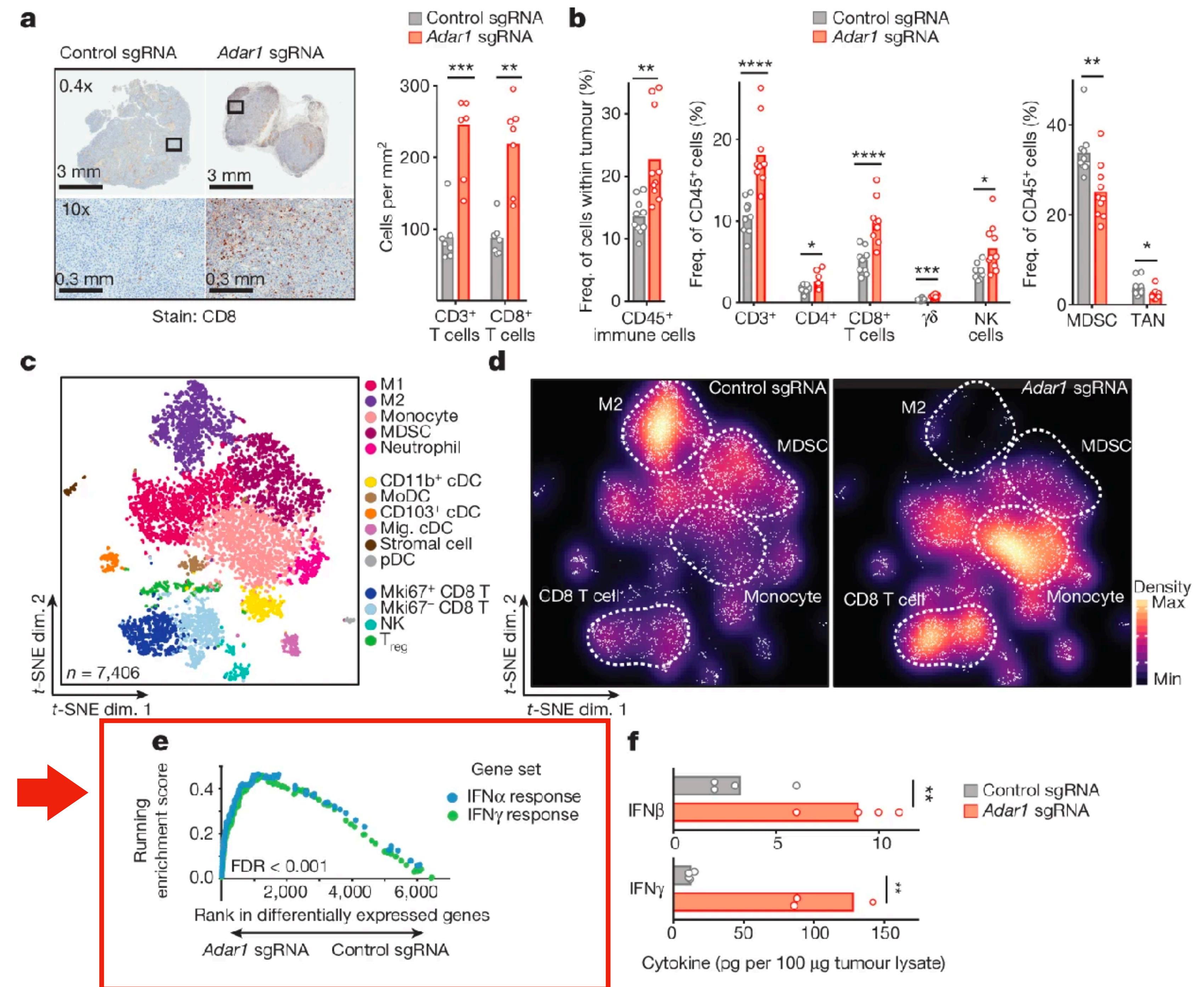
How do we know clusters of cells correspond to cell types?



Tabular Muris, *Nature* (2018)



Enrichment analysis is also embedded a larger analysis



Gene set enrichment analysis checks with previous knowledge

Today's lecture: Enrichment Analysis

- **Motivations: What's next after genomics analysis?**
 - What have we learned? How do we know that our discovery is meaningful?
- **Gene set enrichment analysis**
 - Set-based approach: Hypergeometric test
 - Rank-based approach: GSEA by KS statistic
- **Can we engineer new gene sets/scores?**
 - Principal Component Analysis
 - Matrix factorization of count data

What is Gene Set Analysis?

(Discrete) Gene Set Analysis

Input:

1. A dictionary of gene sets that map genes to sets (gene-to-set mapping)
2. A list of **top** genes identified in our own study (after FDR control)

Output:

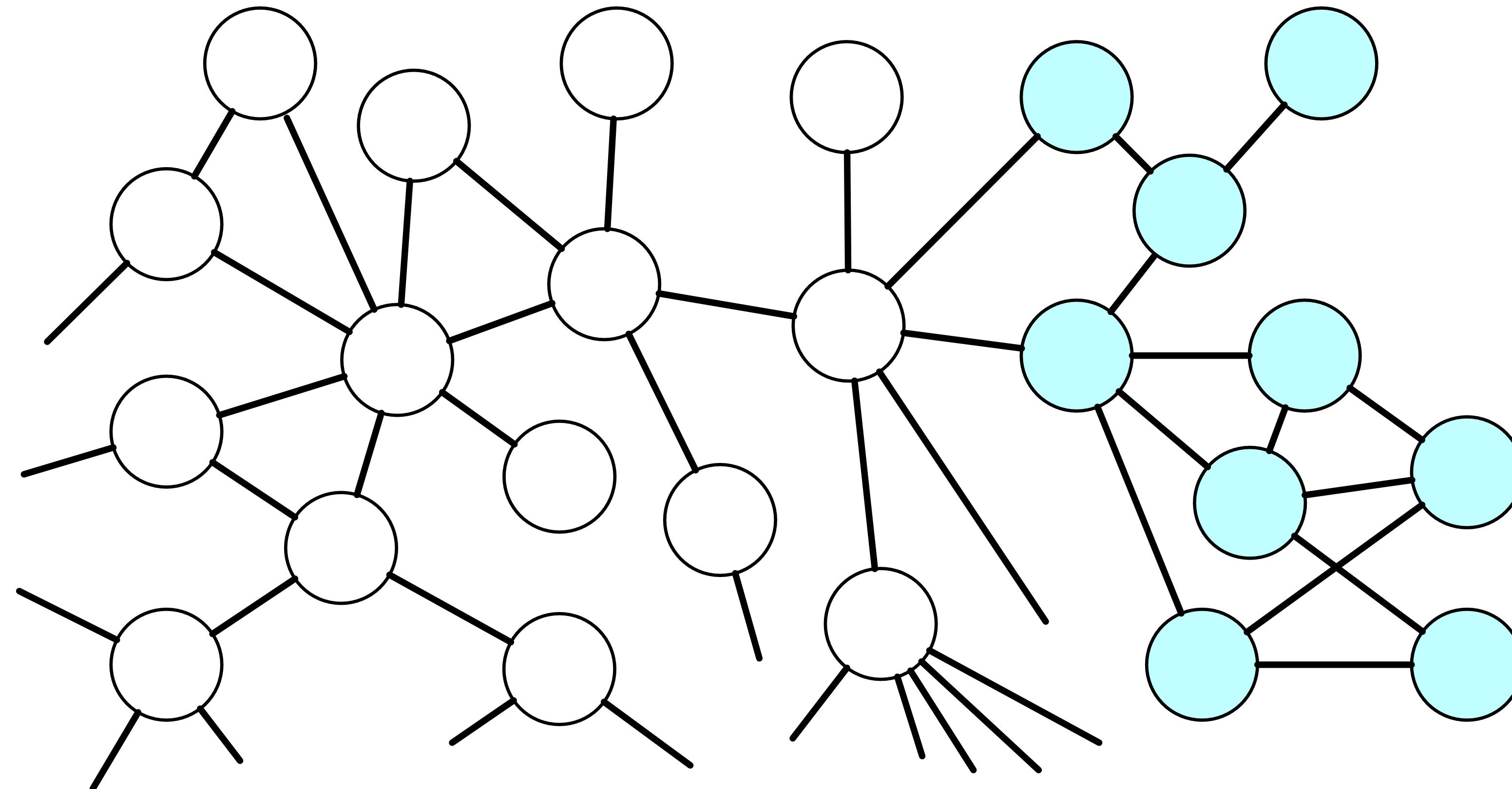
A table of scores for all the gene sets in the dictionary.

(Rank-based) Gene Set Enrichment

Input:

1. A dictionary of gene sets that map genes to sets
2. A **full** list of gene-level **scores** (e.g., p-values)

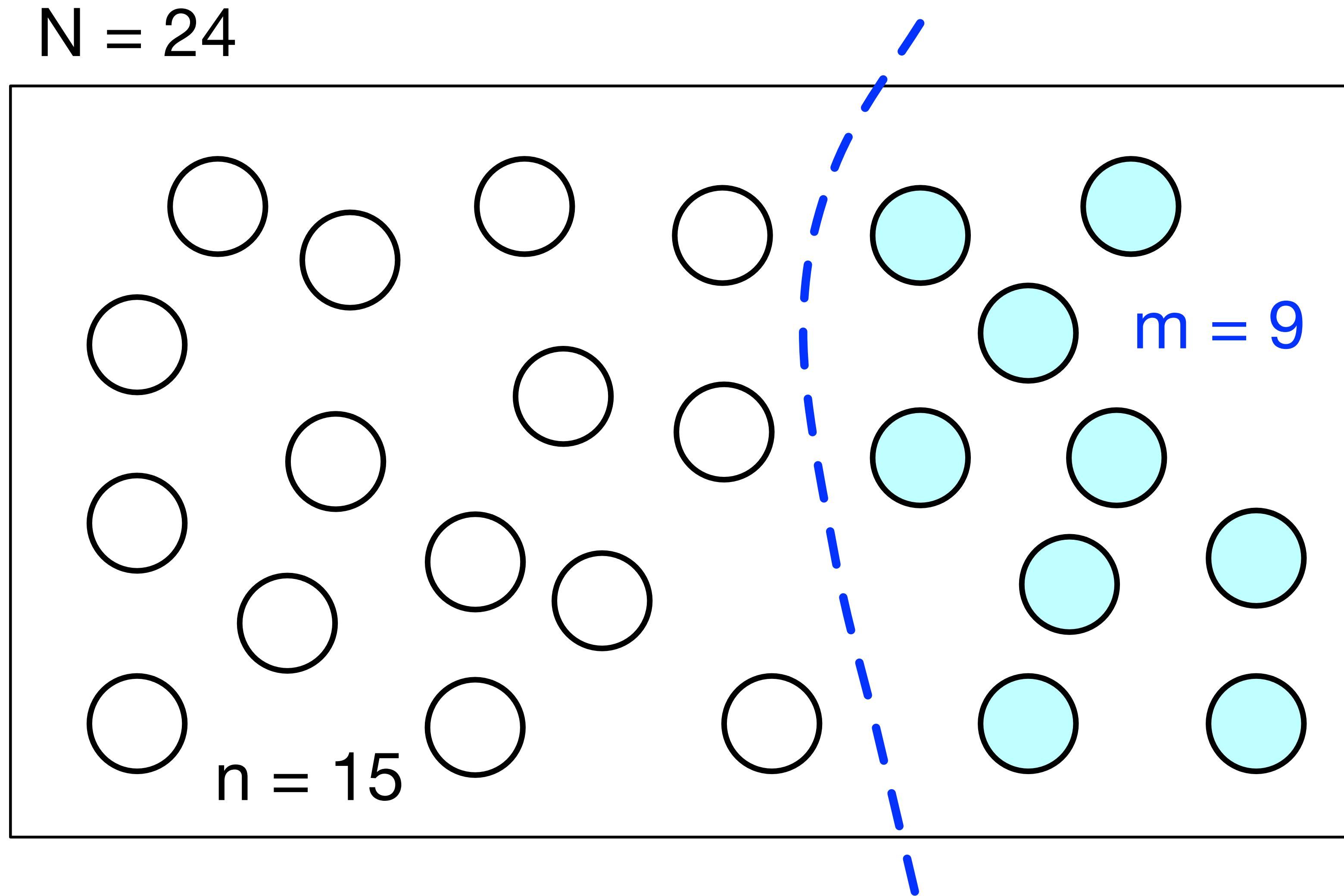
What have we learned from our differential expression analysis?



→ **coloured:** some pathway of interest

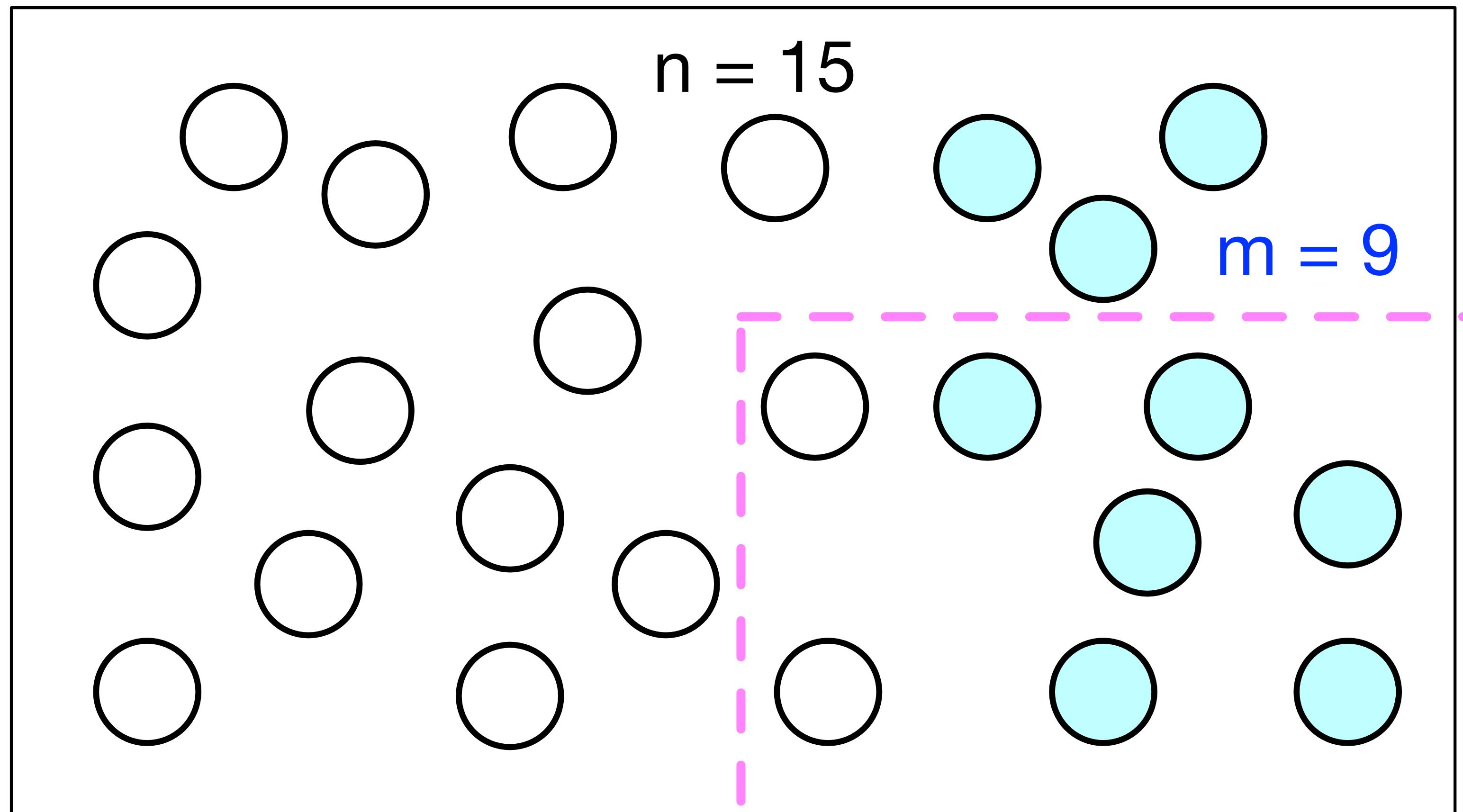
Let's drop the edges in biological networks

Prior Knowledge of biological pathways define a set of genes (coloured)



Does this pathway overlap with our DEG list?

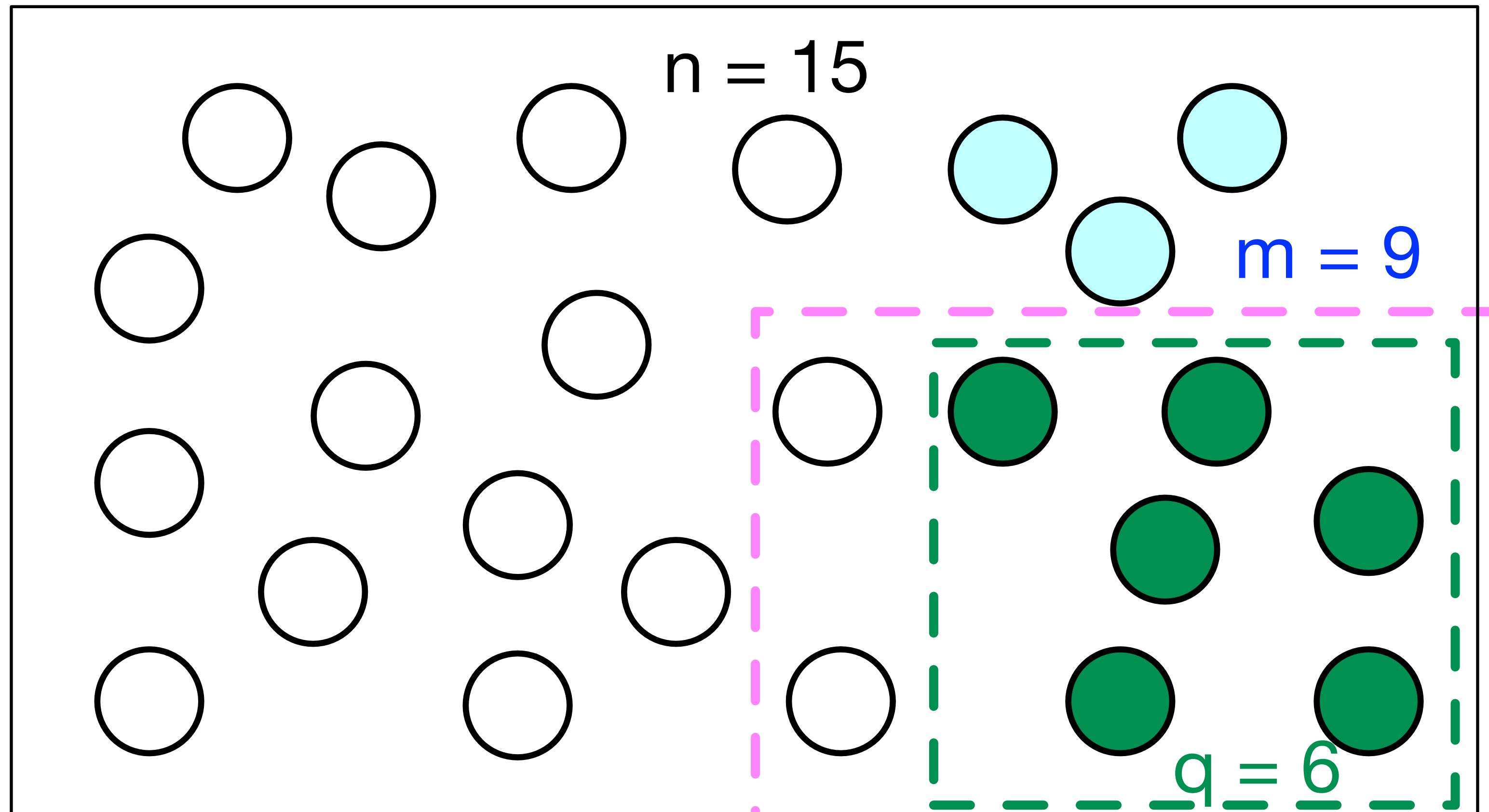
$N = 24$



$k = 8$

Say that we found q genes are overlapping with this pathway

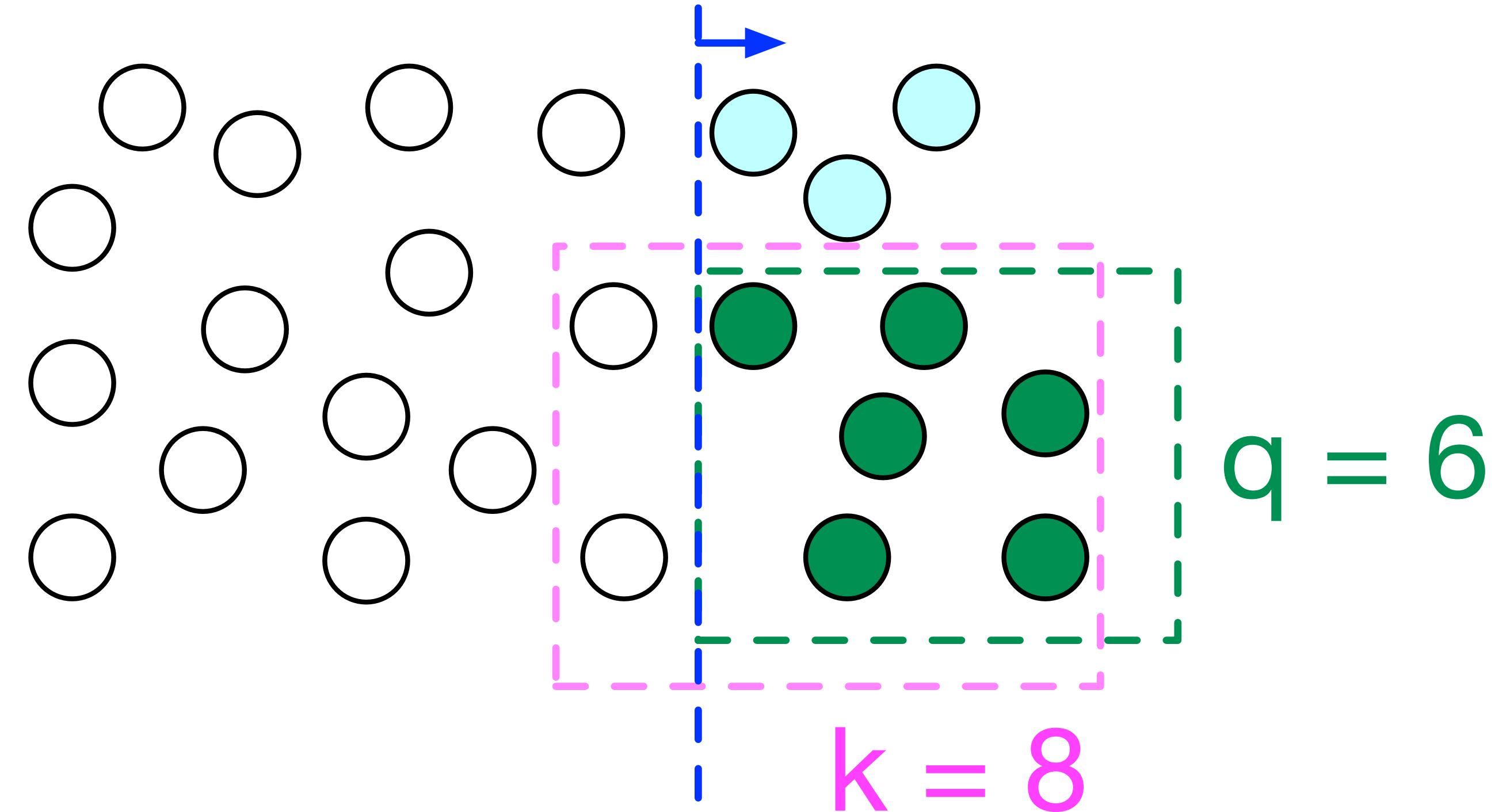
$$N = 24$$



$$k = 8$$

Gene Set Analysis testing over-representation of DEGs

$$N = 24, n = 15, m = 9$$

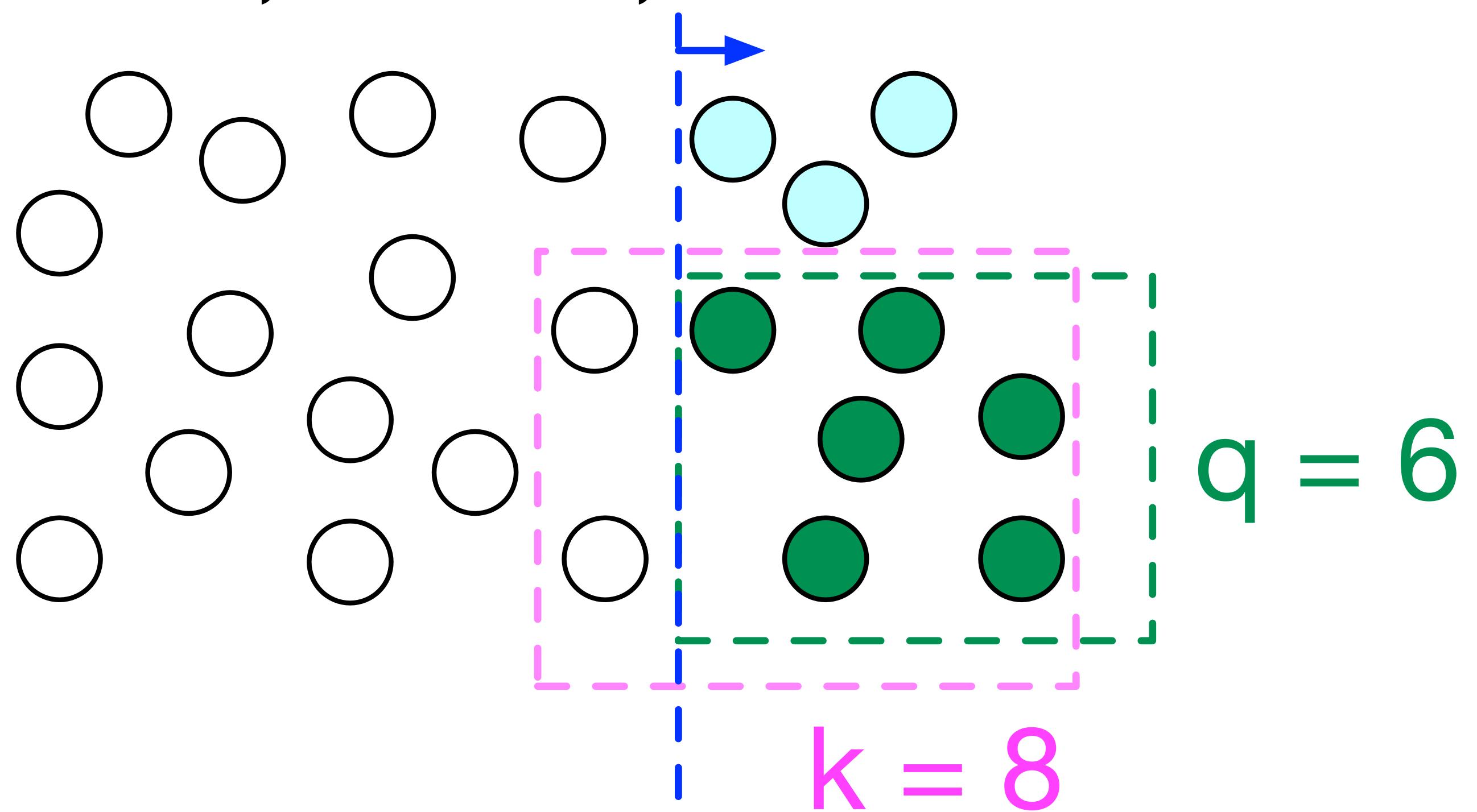


What are the numbers to count?

- ▶ N : # genes in this universe
- ▶ m : # genes in this set
- ▶ n : # genes *not* in this gene set
- ▶ k : # DEGs in our analysis
- ▶ q : # DEGs (of k) overlapping with the set of m genes

Is this overlap of $q = 6$ of $k = 8$ genes significant?

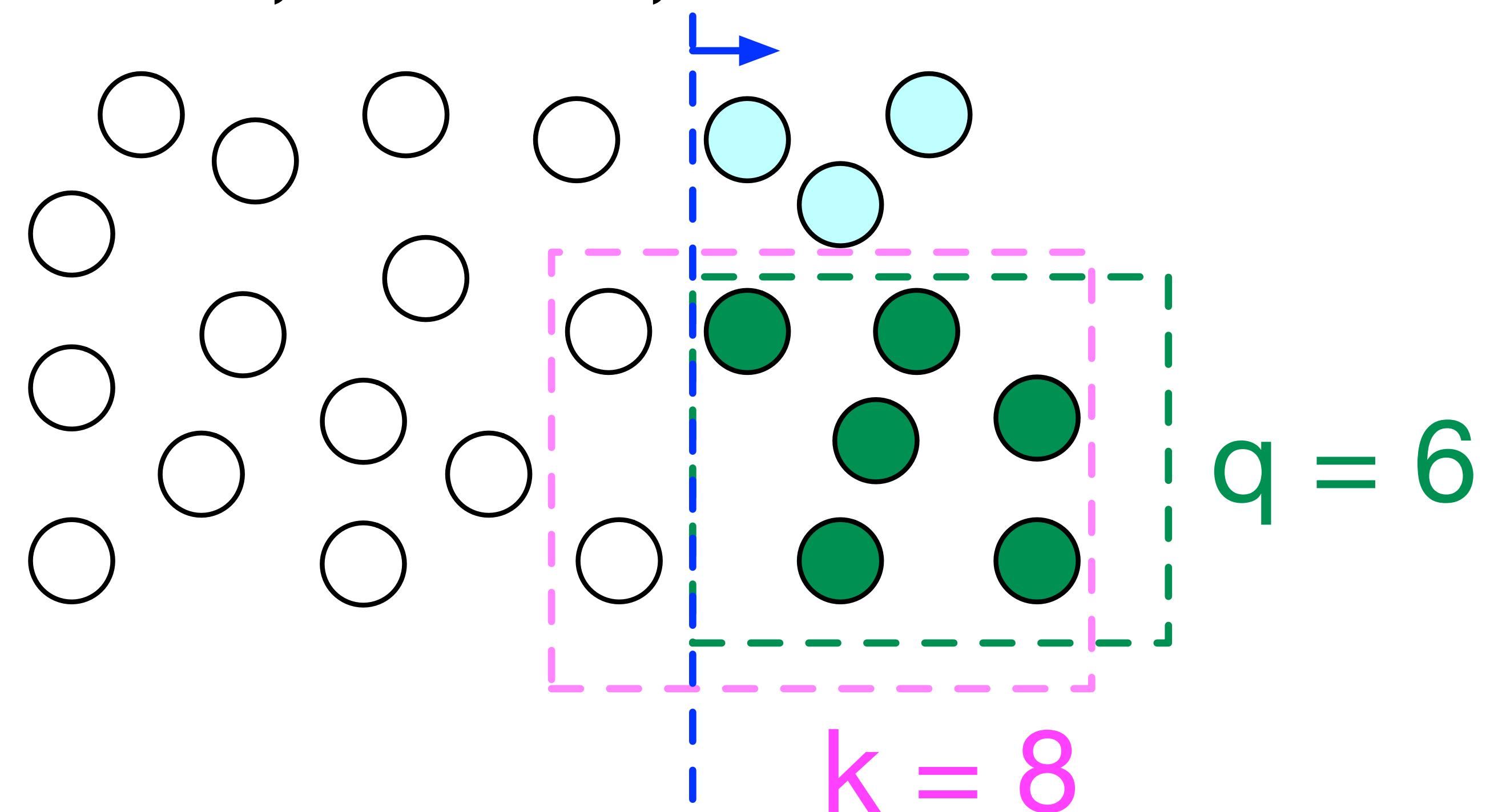
$N = 24, n = 15, m = 9$



- ▶ $N = 24$ # a total of 24 genes
- ▶ $m = 9$ genes in this set
- ▶ $n = N - m = 15$
- ▶ $k = 8$ DEGs
- ▶ $q = 6$ out of $k = 8$ overlap

Is this overlap of $q = 6$ of $k = 8$ genes significant?

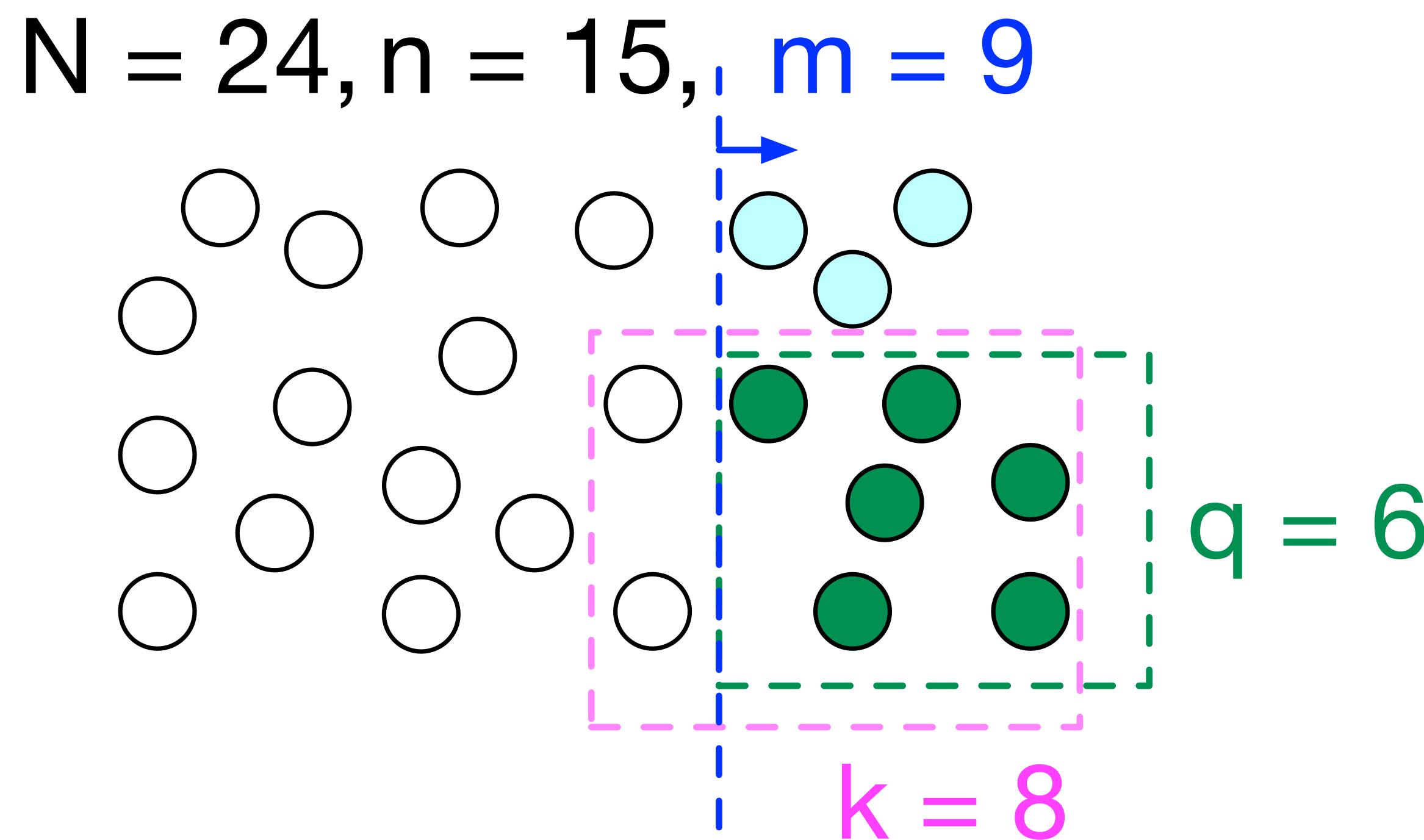
$N = 24, n = 15, m = 9$



Questions:

- ▶ Is it meaningful enough to report?
- ▶ Is it surprising enough that we recapitulated $6/9$ ($\sim 67\%$)?
- ▶ What is null distribution?
- ▶ What is the generative/simulation scheme?

How do we find q out of k DEGs overlapping with a gene set of m genes?



Under null of hypergeometric distribution

1. Sample k DEGs out of N genes
2. Of these k genes, q overlap with a gene set consisting of m genes
3. The rest $k - q$ genes overlap with genes outside of the gene set $N - m$

Review of binomial coefficient: Let's see if we can estimate null distribution by counting

- ▶ How many all possible ways to select $k = 8$ out of $N = 24$ genes, ignoring the order of k selected genes and $N - k$ *not* selected genes?
- ▶ We can think of this as three steps: (1) enumerating N genes, (2) partition them into the first k genes and the rest, (3) ignore the order within each partition.

$$\binom{24}{8} = \frac{\{\text{all possible ways to enumerate 24 genes}\}}{\{\text{enumerating 8 genes}\}\{\text{enumerating 16 genes}\}} = \frac{24!}{8!16!}$$

Hypergeometric distribution

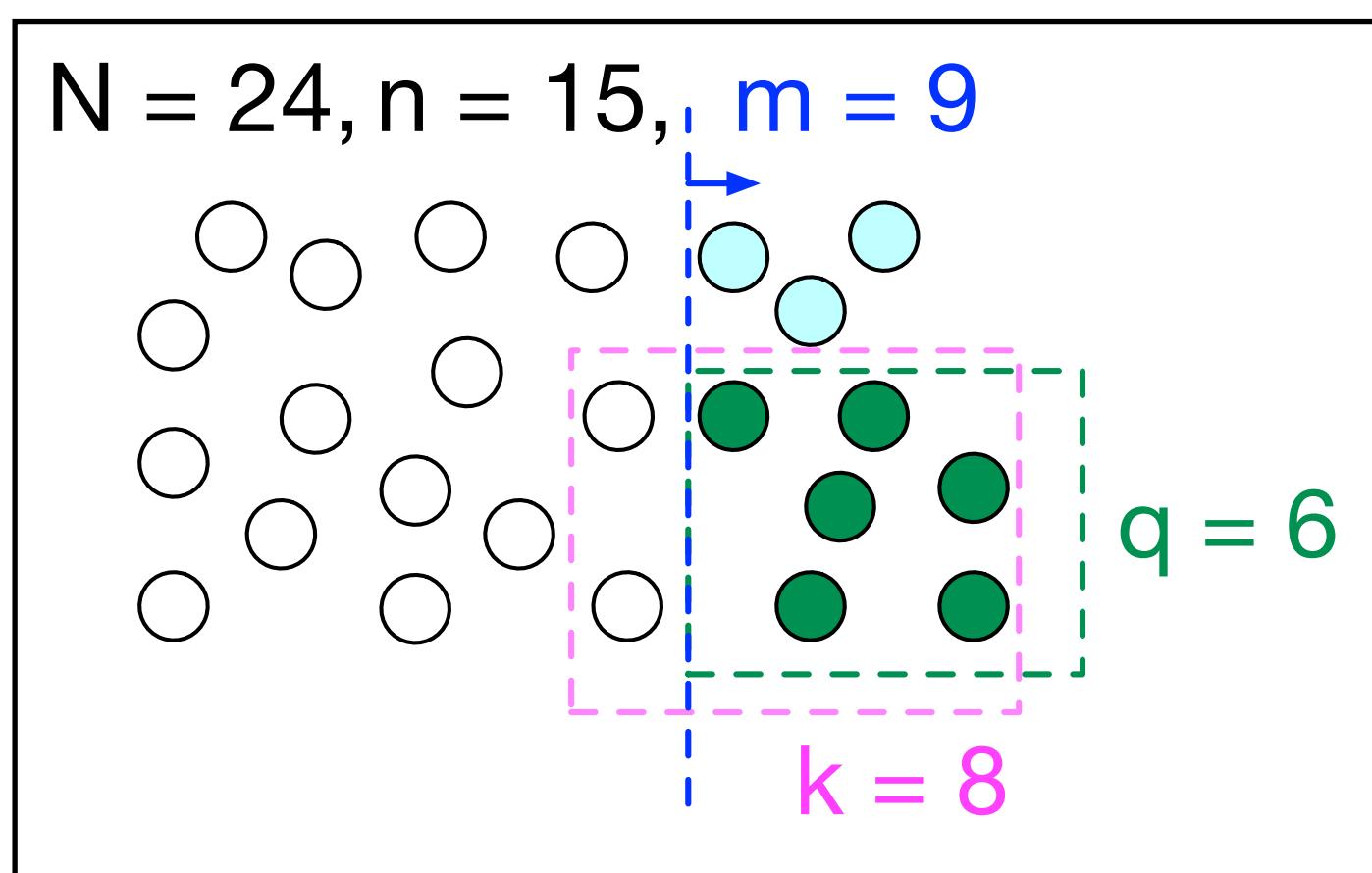
1. What is the probability of selecting k DEGs out of N genes?
2. How many possible ways of finding q DEGs overlapping with m genes in the gene set?
3. How many possible ways of finding the rest $k - q$ DEGs overlapping with $(N - m = n)$ genes in the gene set?

Hypergeometric distribution

$$P_0(q|N, m, k) = \sum \begin{matrix} \text{\# ways to select } q \text{ out of } m \\ \text{\# ways to select } (k - q) \text{ out of } N - m \end{matrix}$$

the probability
of having a set w/size k
out of N

=



Hypergeometric distribution

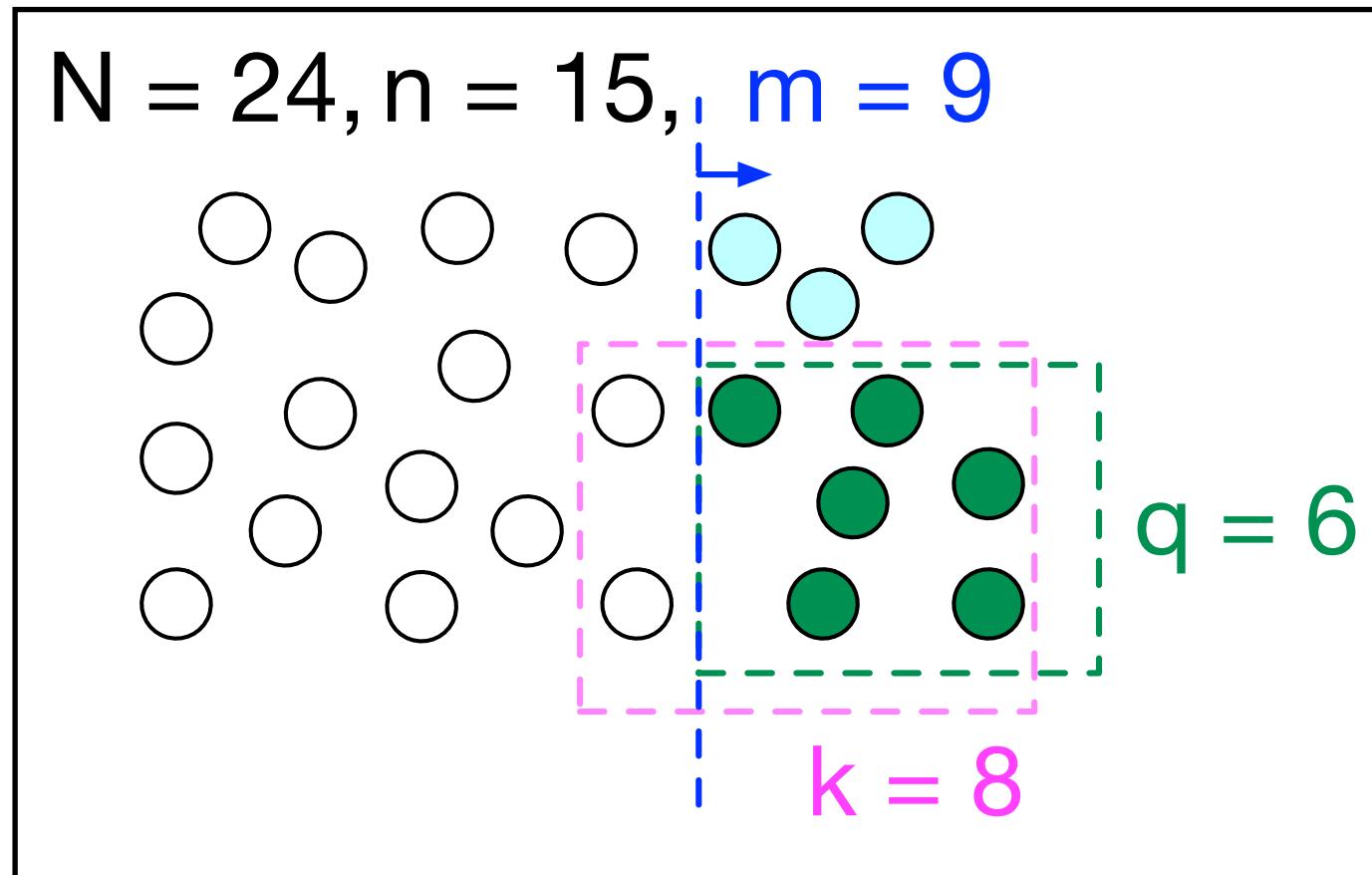
$$P_0(q|N, m, k) = \sum \frac{\# \text{ ways to select } q \text{ out of } m}{\# \text{ ways to select } (k - q) \text{ out of } N - m}$$

the probability
of having a set w/size k
out of N

=

$$\underbrace{\binom{m}{q}}$$

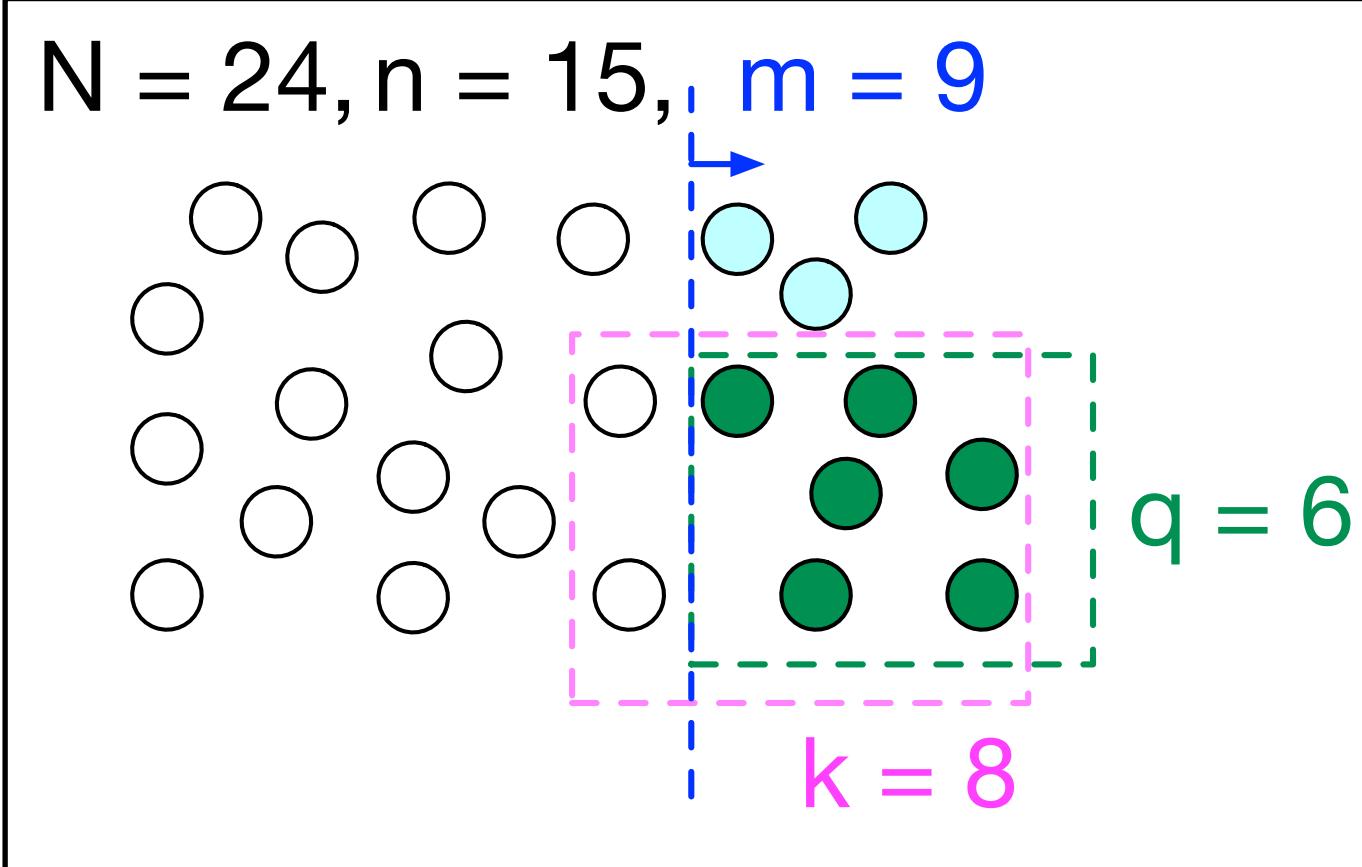
ways to choose
 q overlap out of m



Hypergeometric distribution

$$P_0(q|N, m, k) = \sum \begin{matrix} \text{\# ways to select } q \text{ out of } m \\ \text{\# ways to select } (k - q) \text{ out of } N - m \end{matrix}$$

the probability
of having a set w/size k
out of N



$$= \underbrace{\binom{m}{q}}_{\text{\# ways to choose } q \text{ overlap out of } m} \times \underbrace{\binom{N - m}{k - q}}_{\text{\# ways to choose } (k - q) \text{ out of } N - m}$$

Hypergeometric distribution

$$P_0(q|N, m, k) =$$

$$\sum$$

ways to select q out of m

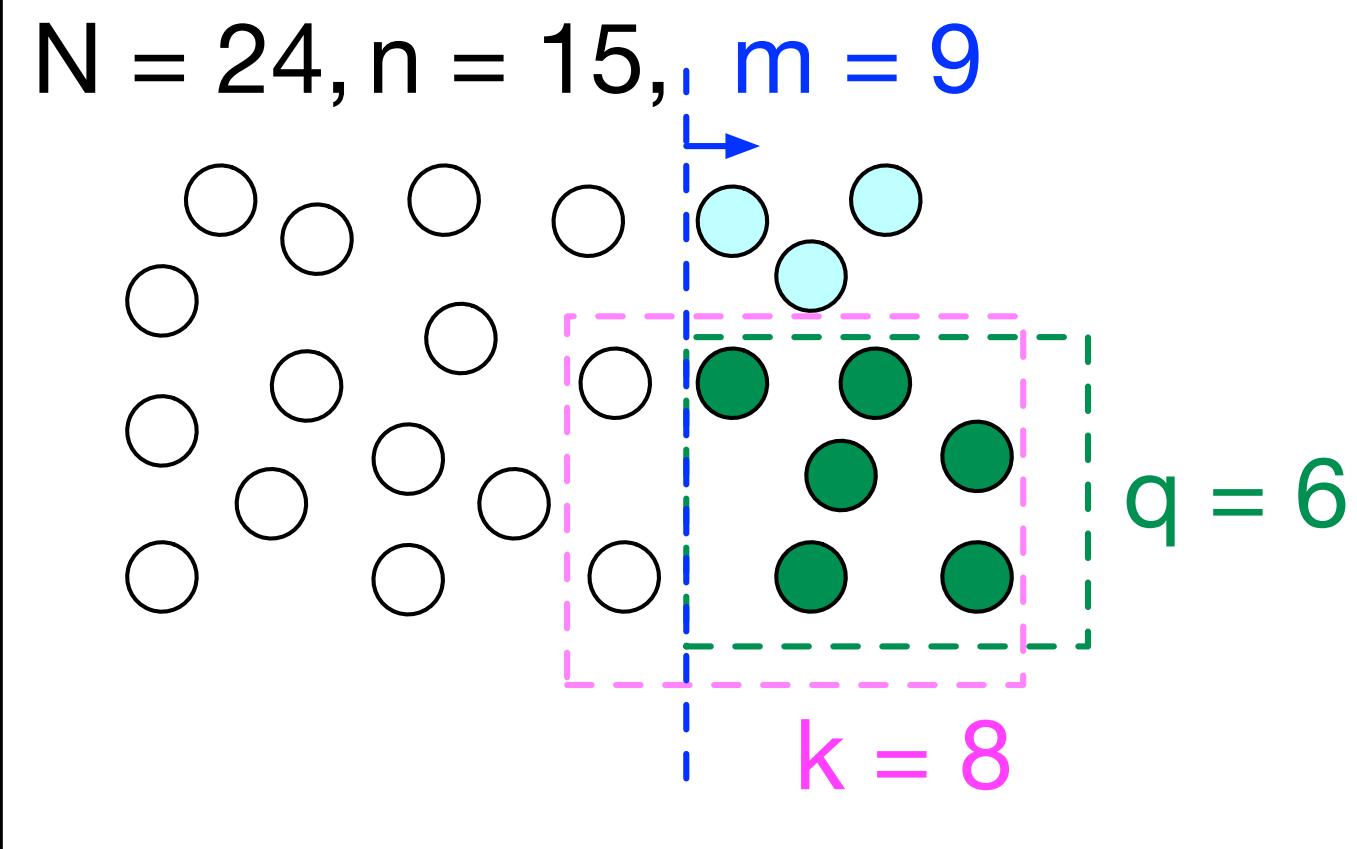
ways to select $(k - q)$ out of $N - m$

$$\binom{m}{q}$$

\times

$$\binom{N - m}{k - q}$$

$$\times \binom{N}{k}^{-1}$$



ways to choose
 q overlap out of m

ways to choose
 $(k - q)$ out of $N - m$

the probability
of having a set w/size k
out of N

What is the probability of k overlapping DEGs?

Hypergeometric PMF

$$p(x|N, m, k) = \frac{\binom{m}{x} \binom{N-m}{k-x}}{\binom{N}{k}}$$

Hypergeometric CDF

$$p(q|N, m, k) = \sum_{x=0}^q \frac{\binom{m}{x} \binom{N-m}{k-x}}{\binom{N}{k}}$$

Hypergeometric test for testing significant overlap

$$H_0 : x \leq q \quad \text{vs.} \quad H_1 : x > q$$

We may observe overlap q genes by random sampling of k genes **without** replacement.

Therefore, we can calculate the p-value:

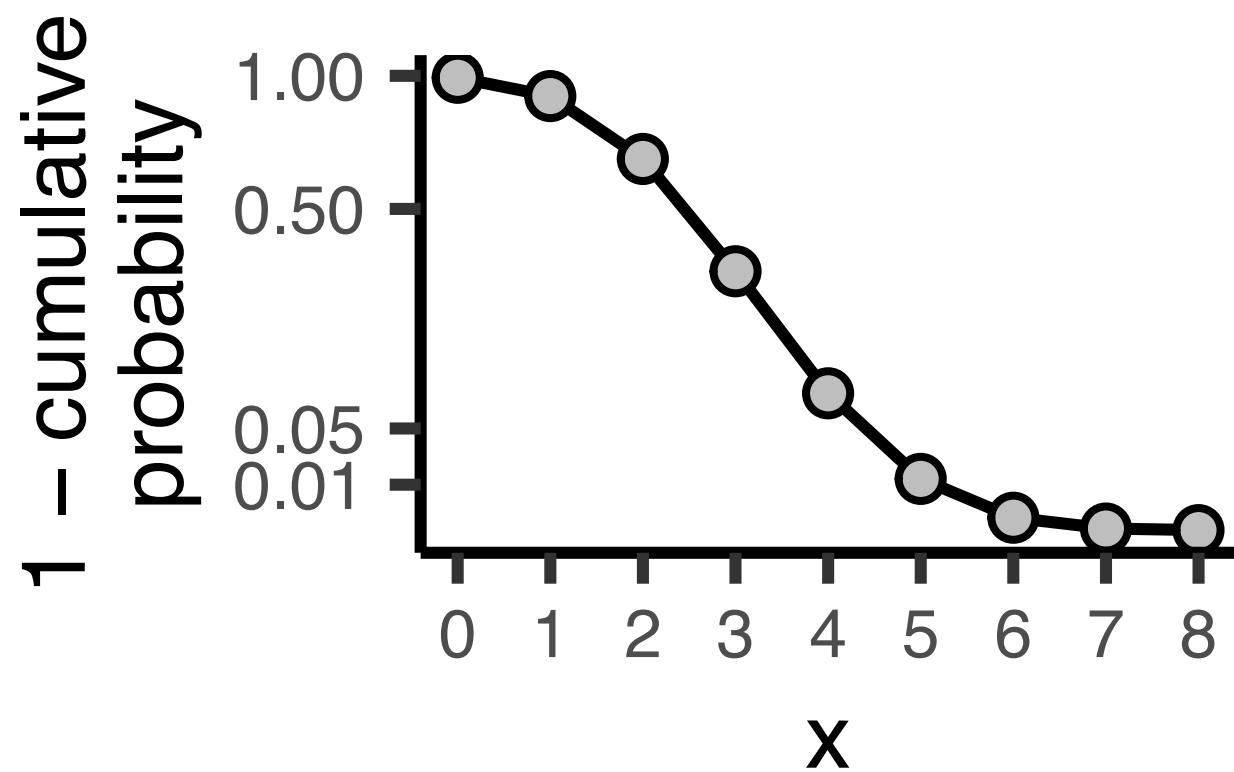
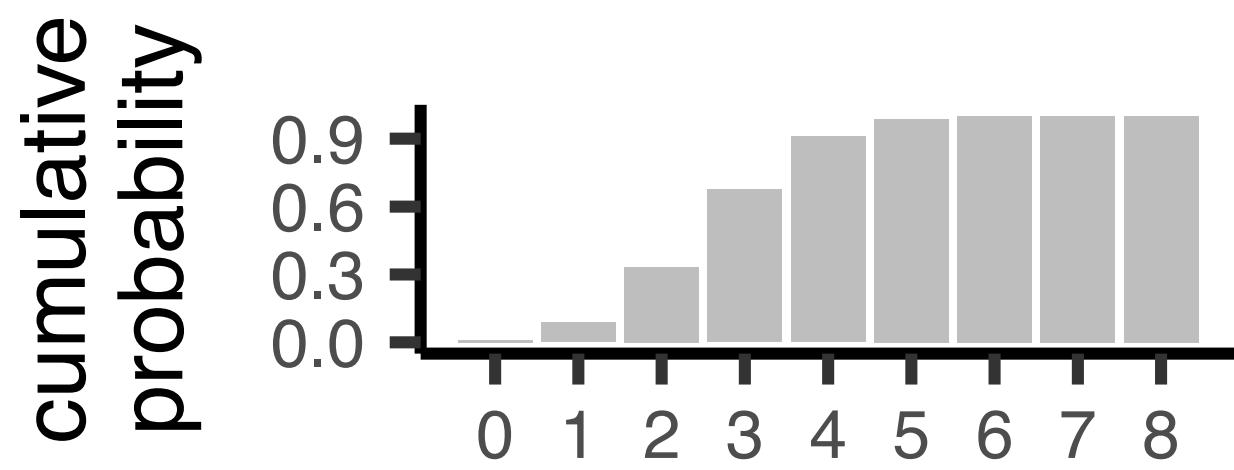
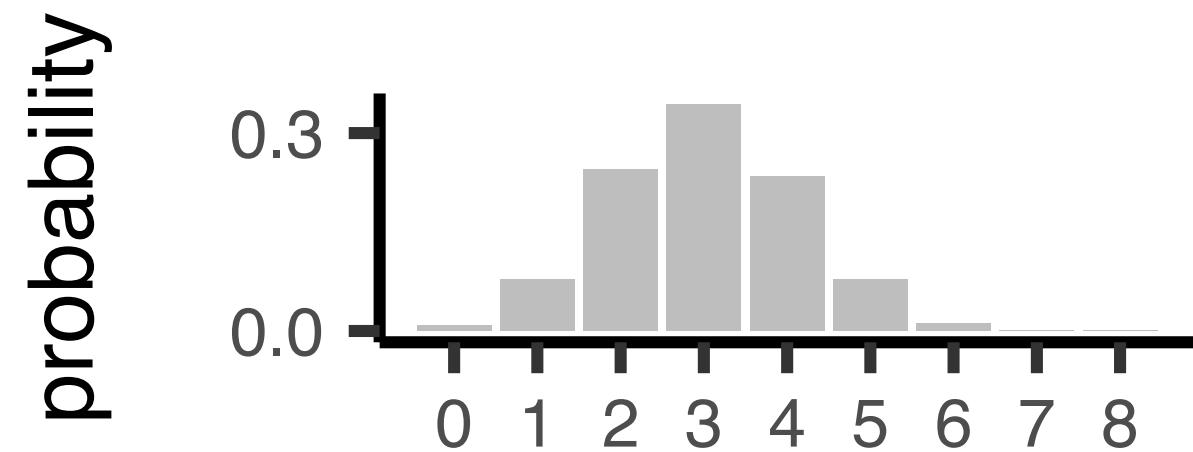
$$P(x > q | n, m, k) = 1 - \sum_{x=0}^q \frac{\binom{m}{x} \binom{n}{k-x}}{\binom{n+m}{k}}$$

```
phyper(q=6, m=9, n=15, k=8, lower.tail=FALSE)
```

```
## [1] 0.0007464604
```

How significant is q overlap in our discovery?

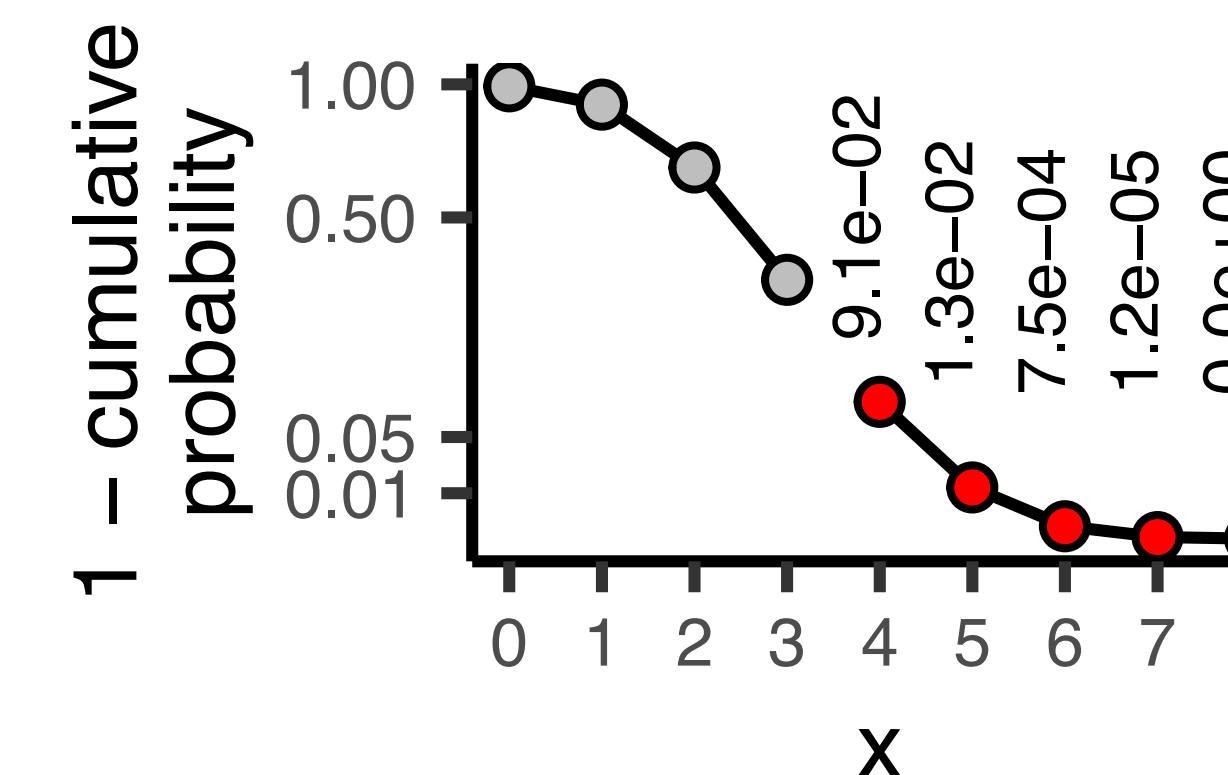
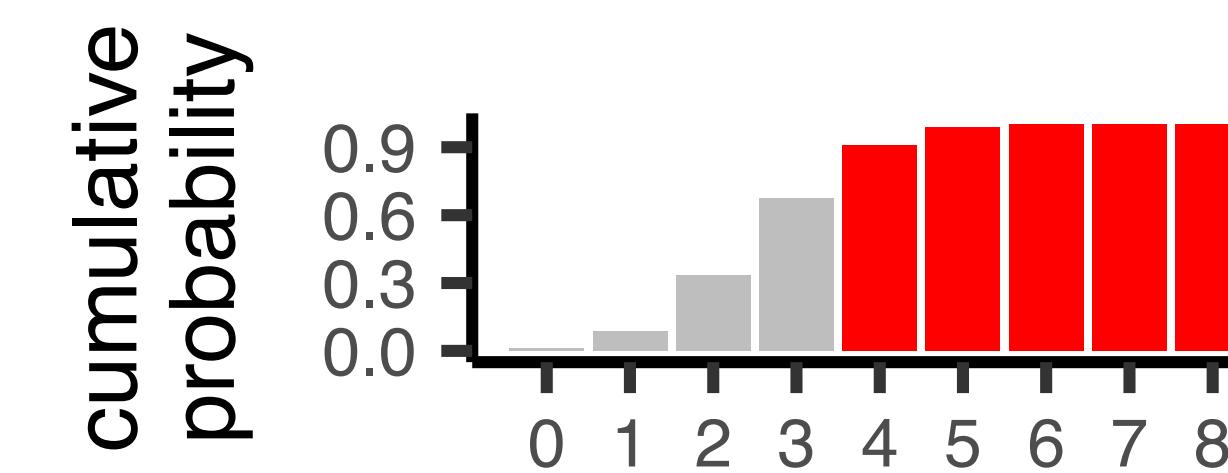
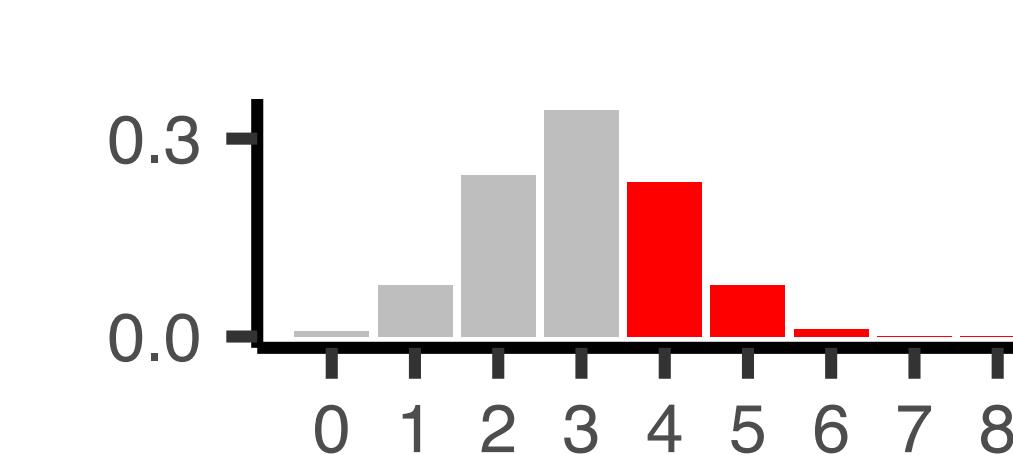
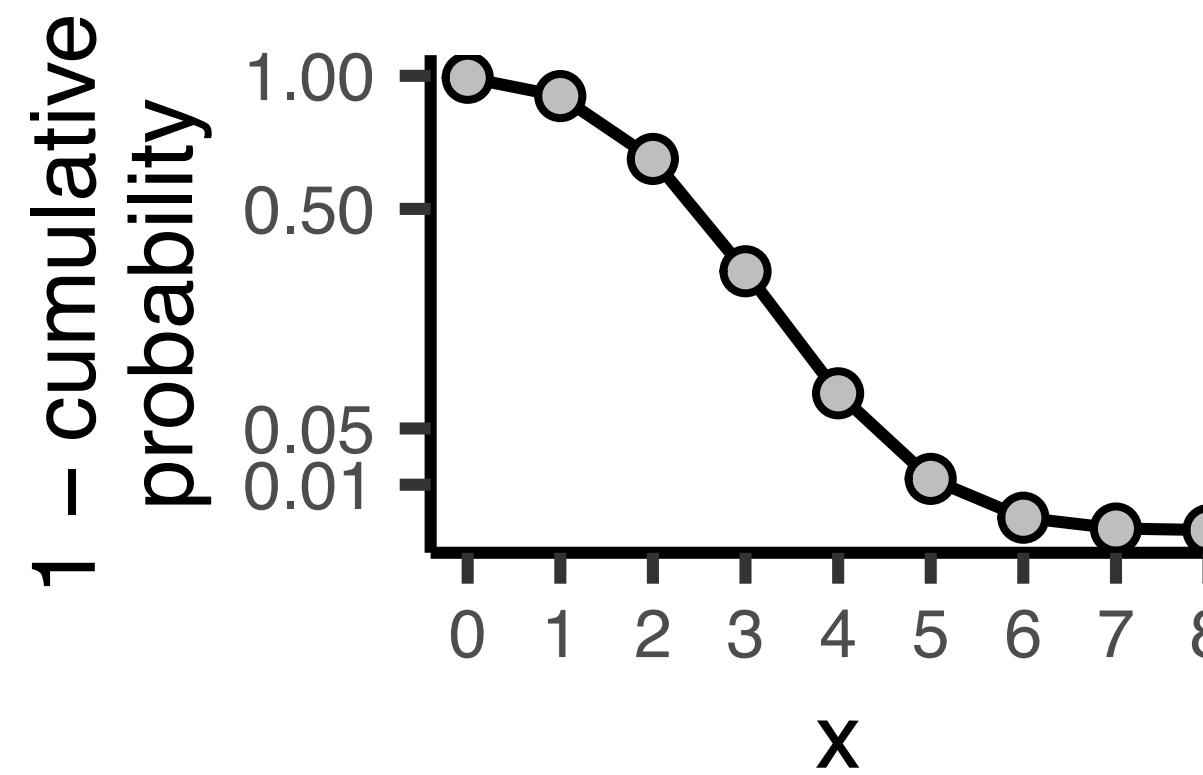
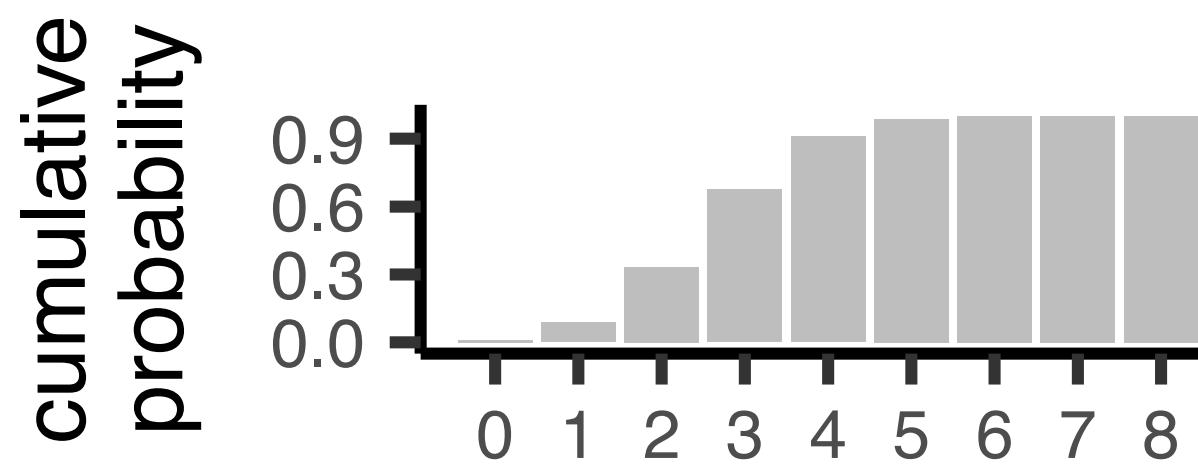
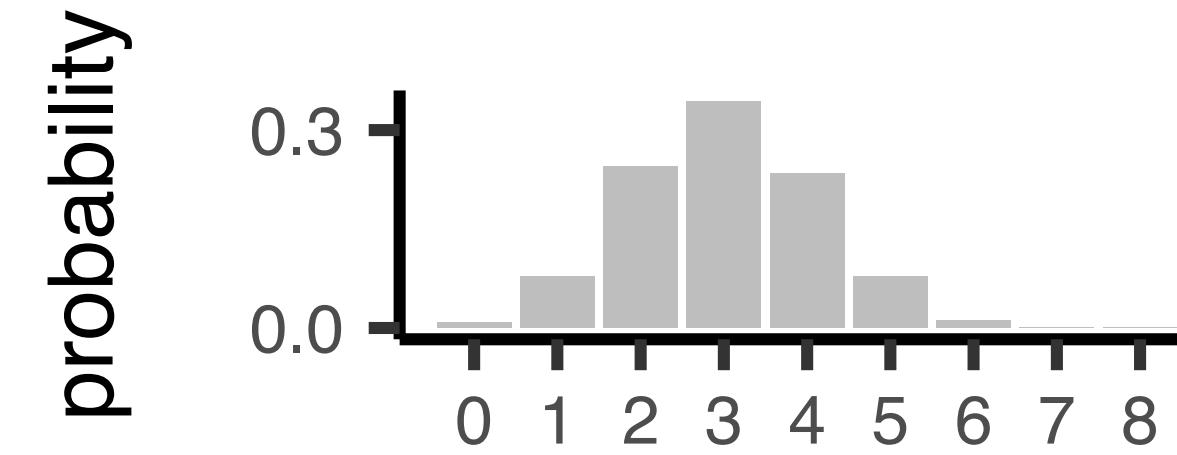
$m = 9$, $n = 15$, and $k = 8$,



How significant is q overlap in our discovery?

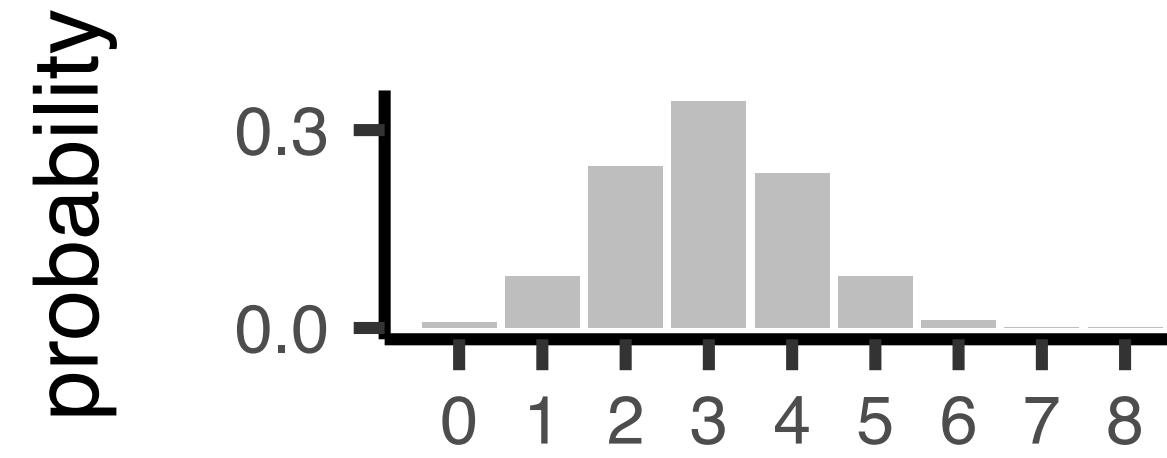
$m = 9$, $n = 15$, and $k = 8$,

If $q = 3$:

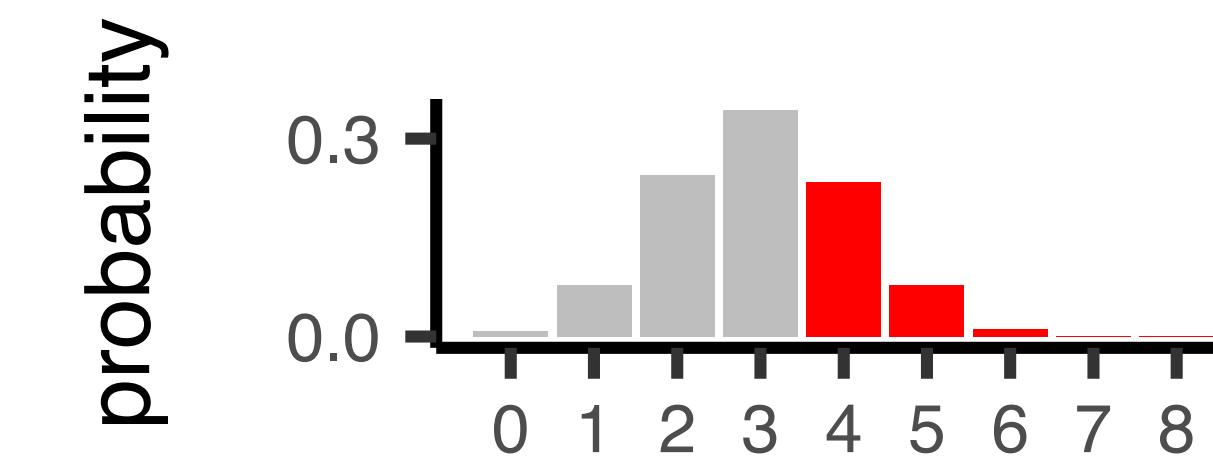


How significant is q overlap in our discovery?

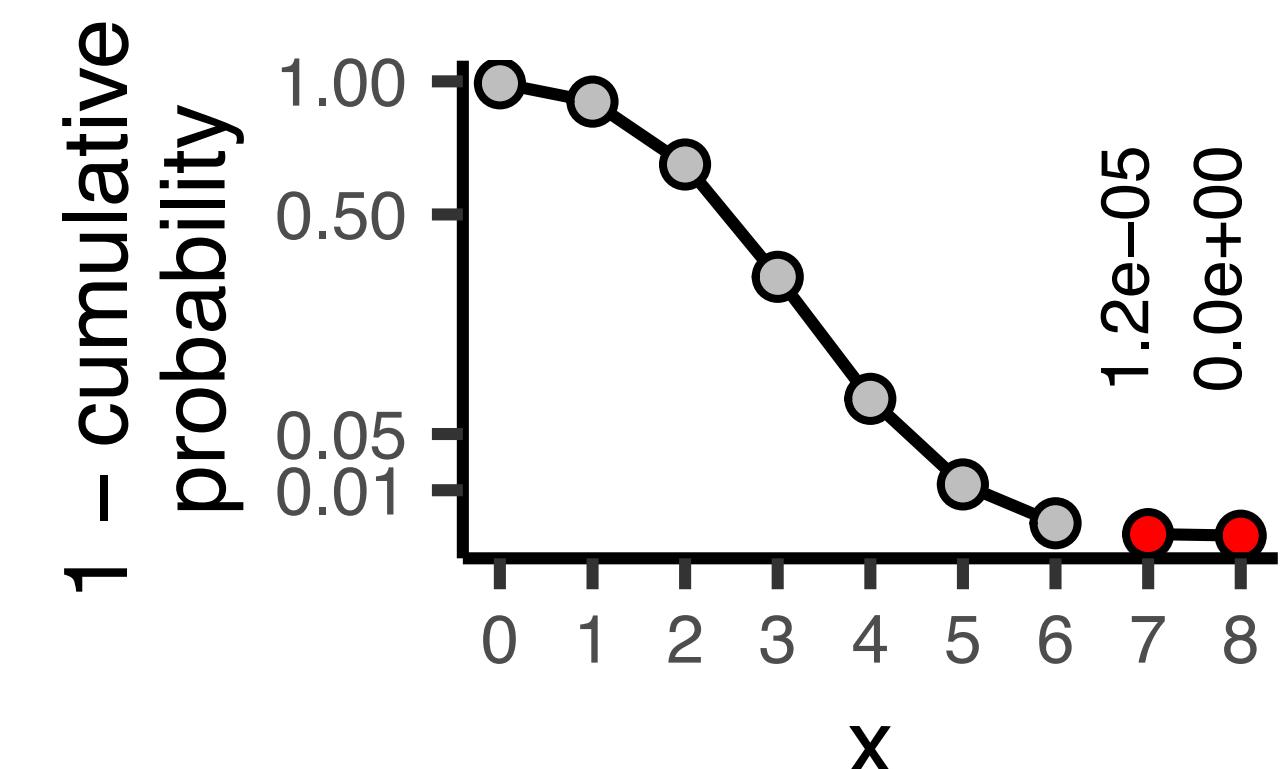
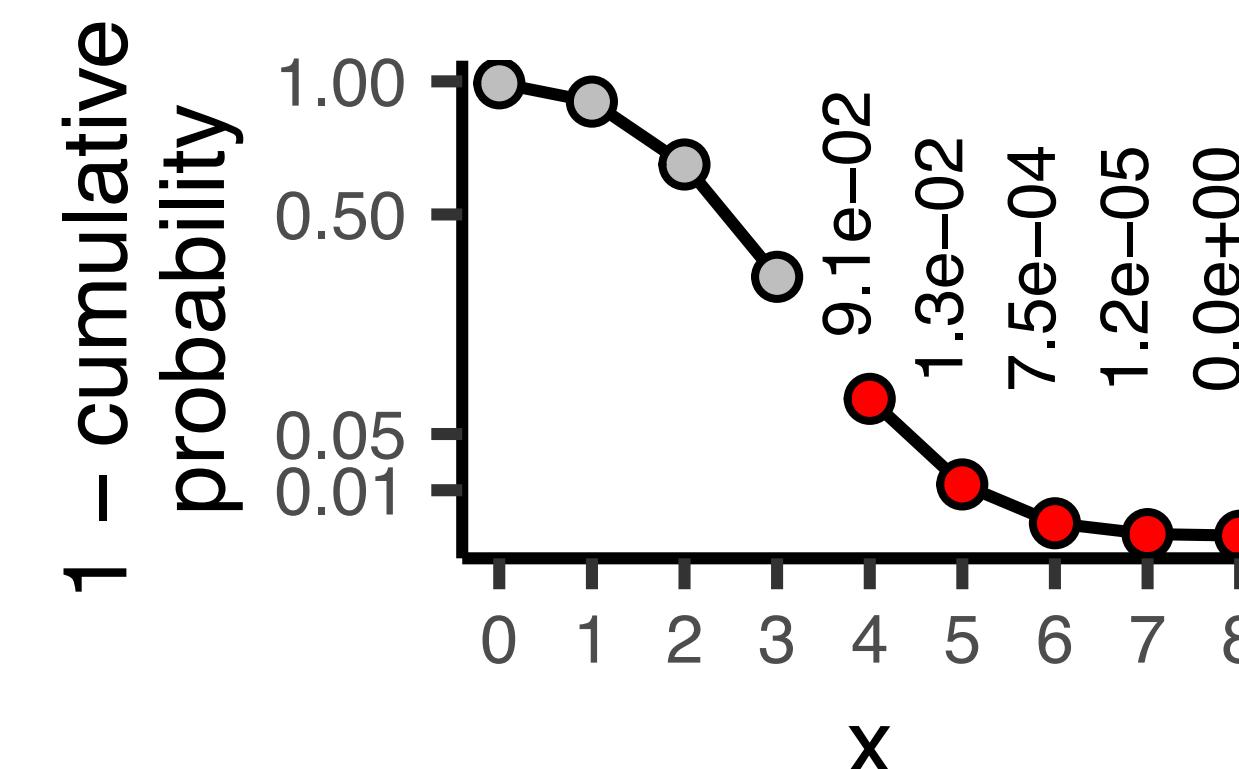
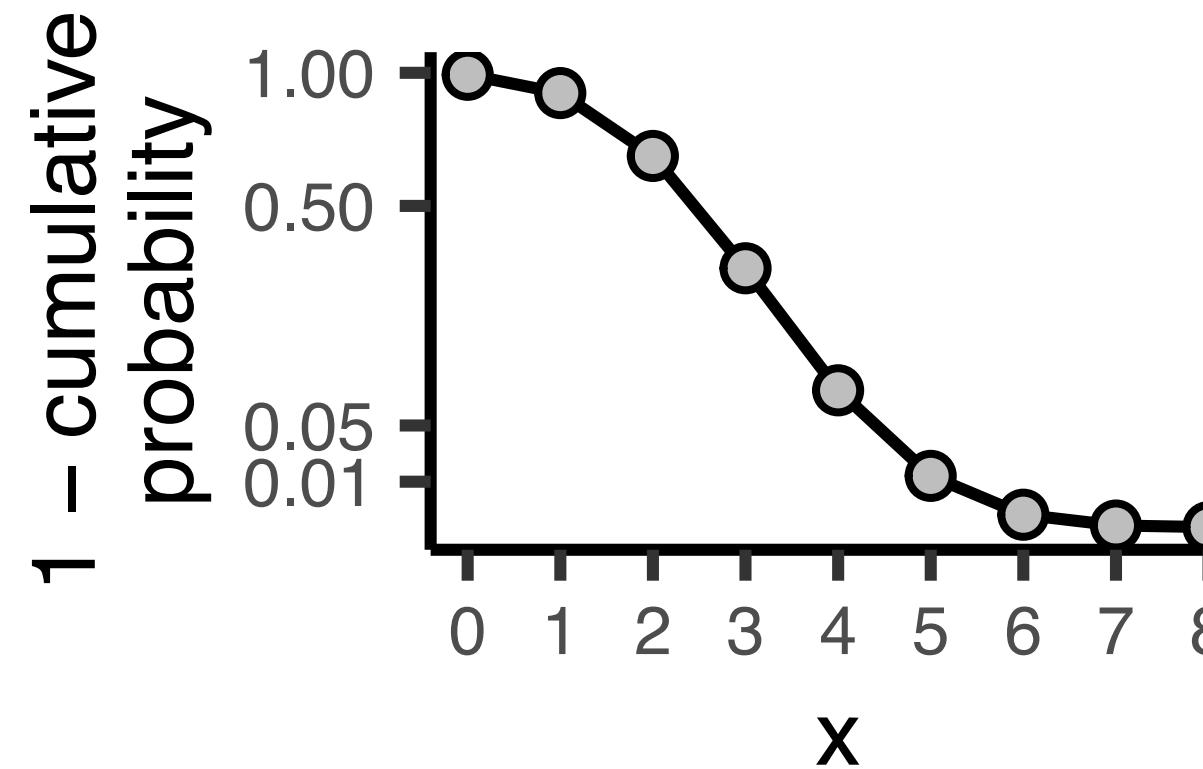
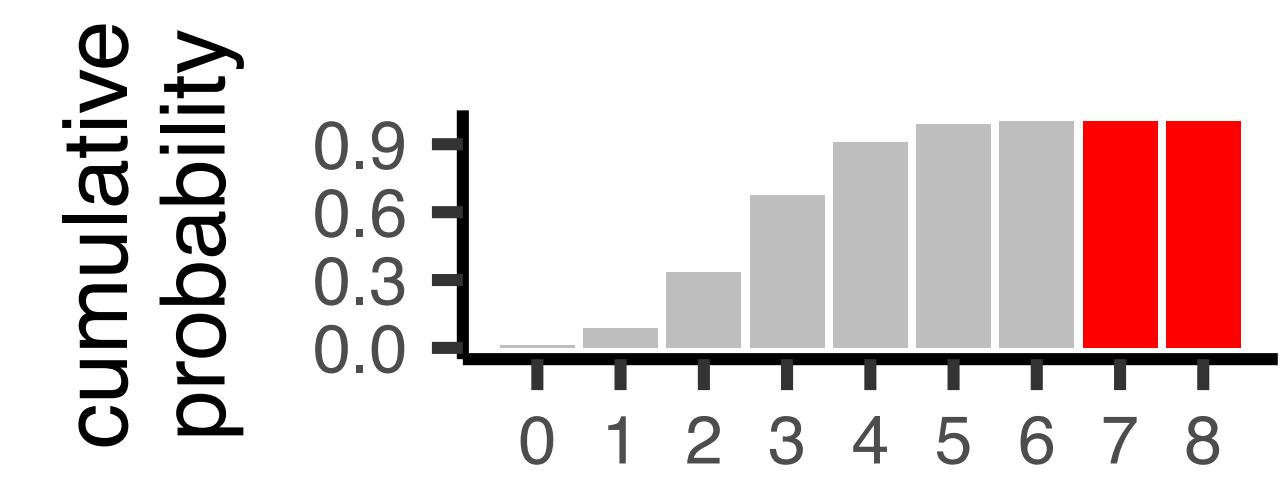
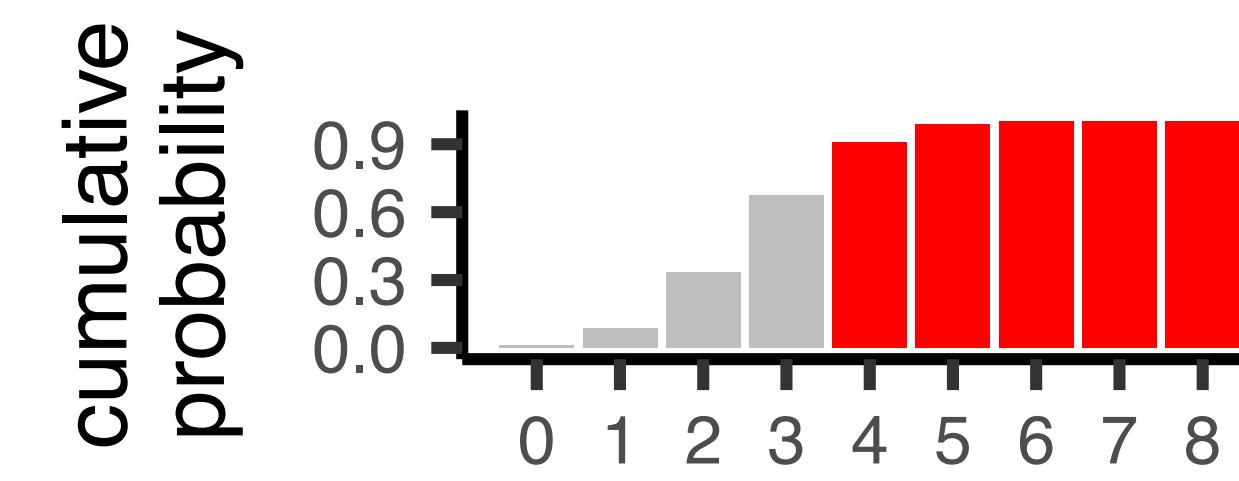
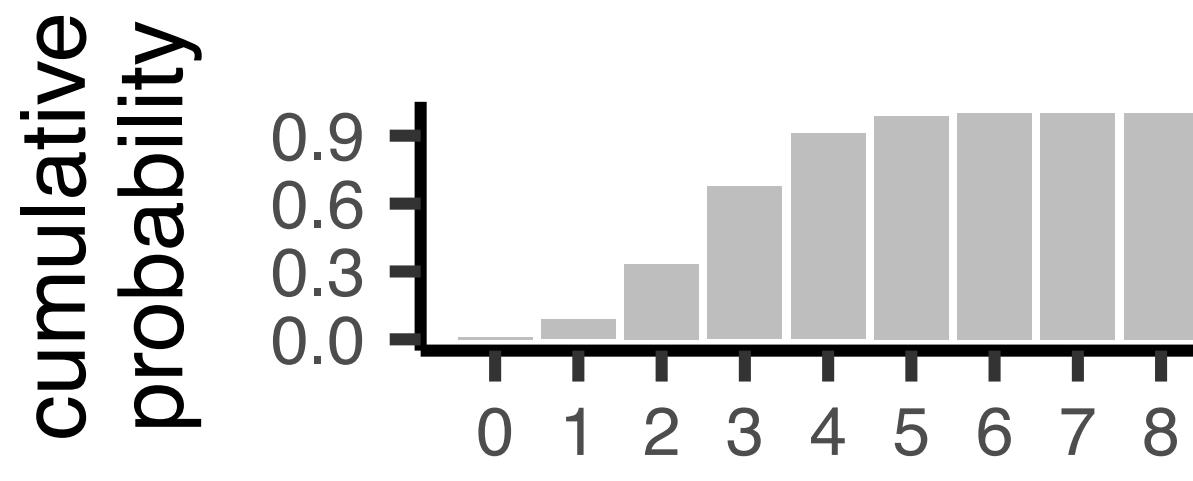
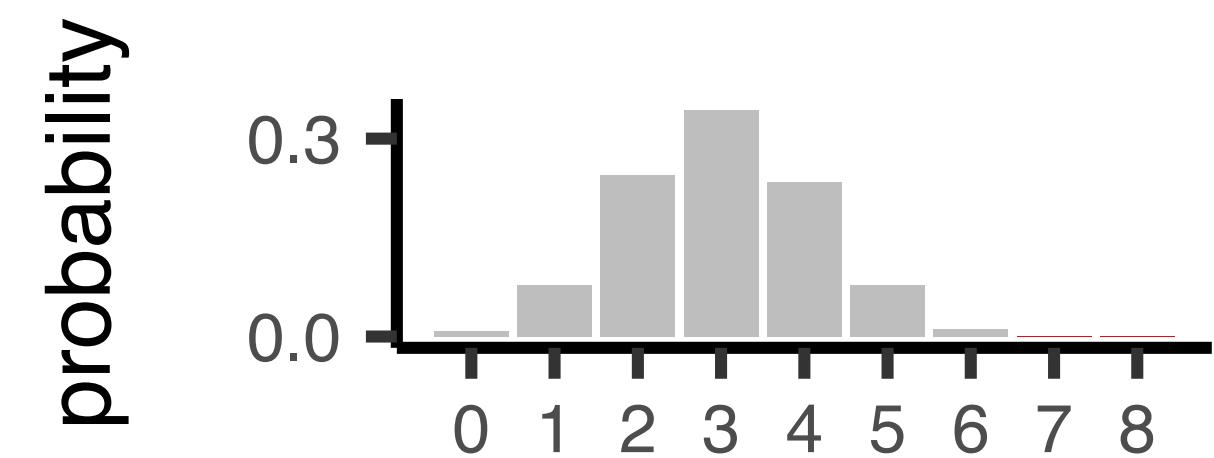
$m = 9$, $n = 15$, and $k = 8$,



If $q = 3$:



If $q = 6$:



Summary of the gene set analysis by hypergeometric test

When does it work?

- ▶ A routine to construct a set of differentially expressed genes by handling multiple hypothesis testing
- ▶ Gene sets are of similar sizes and *nearly* disjoint/independent from one another
- ▶ Genes are *nearly* independent (there is no overwhelmingly favourite genes)

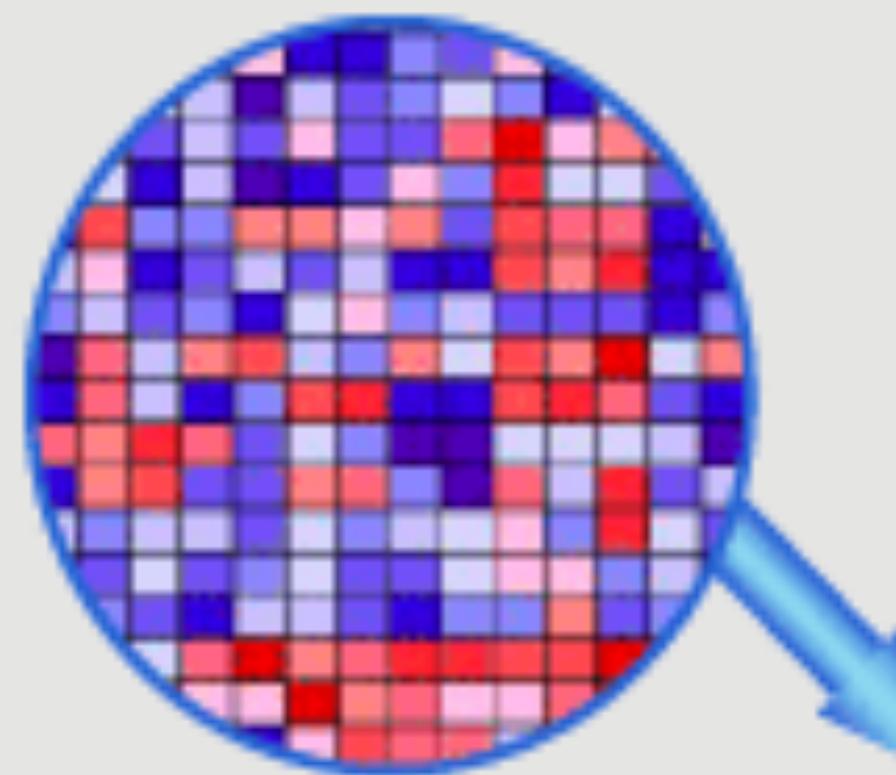
When does it not work?

- ▶ **Don't** have a good way to make a set:
 - ▶ Our discovery data may lack statistical power, i.e., no (or a few) significant genes left after multiple hypothesis correction
 - ▶ There is a hidden factor that can affect two steps: (1) gene set selection (annotations/knowledge) and (2) differential expression calling

Today's lecture: Enrichment Analysis

- **Motivations: What's next after genomics analysis?**
 - What have we learned? How do we know that our discovery is meaningful?
- **Gene set enrichment analysis**
 - Set-based approach: Hypergeometric test
 - Rank-based approach: GSEA by KS statistic
- **Can we engineer new gene sets/scores?**
 - Principal Component Analysis
 - Matrix factorization of count data

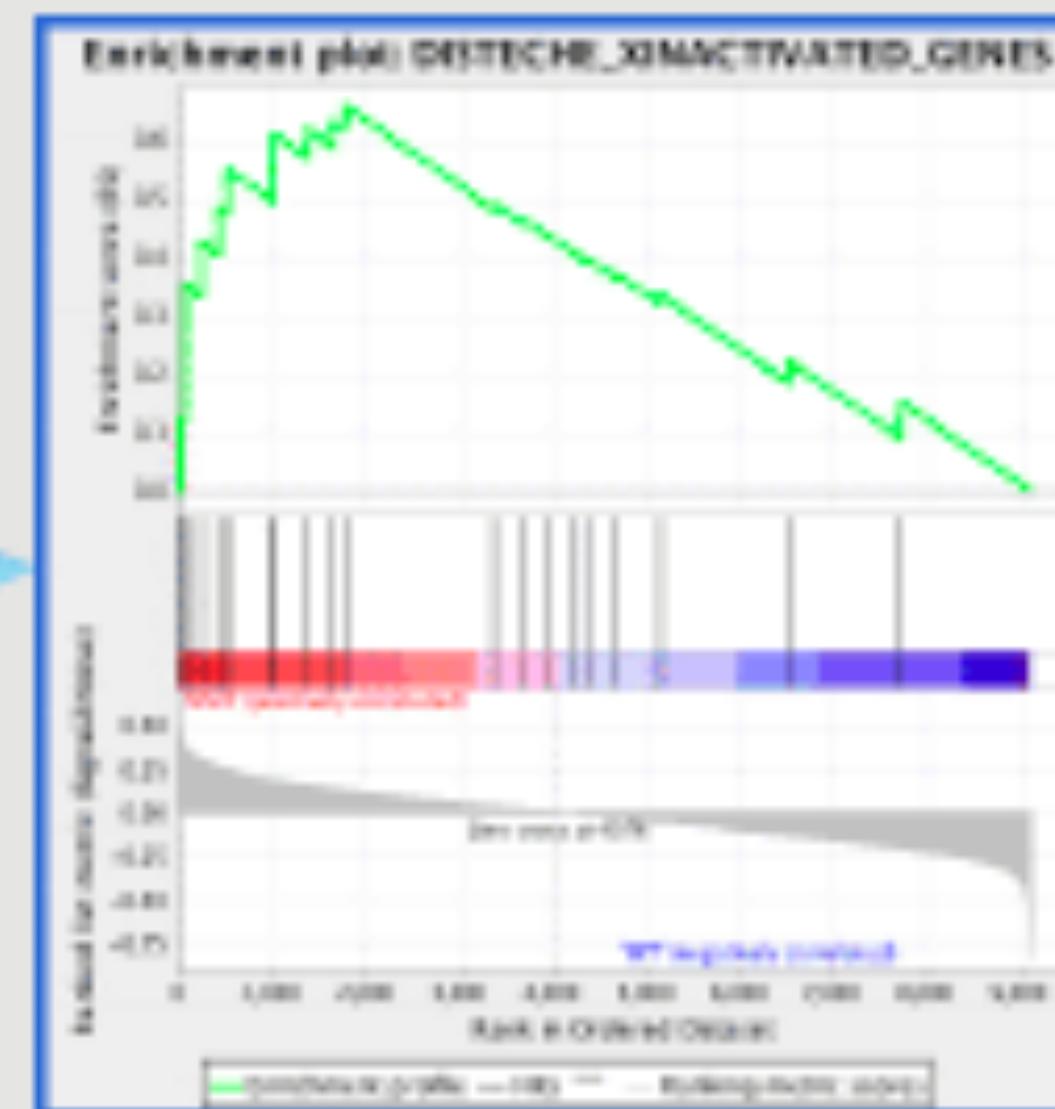
Molecular Profile Data



Gene Set Database

Run
GSEA

Enriched Sets



Rank-based Gene Set Enrichment Analysis method (Subramanian *et al.* 2005)

- ▶ A collection of gene-sets: $\mathcal{C}_1, \dots, \mathcal{C}_K$
- ▶ A vector of gene-level scores (G genes): z_1, \dots, z_G
- ▶ Each z_g could come from differential expression analysis

GSEA algorithm

- ▶ For each k , compute a set-level score $S_k(\mathbf{z}, \mathcal{C}_k)$
- ▶ E.g., Kolmogorov-Smirnov statistic comparing the distribution $\{z_g : g \in \mathcal{C}_k\}$ vs. $\{z_g : g \notin \mathcal{C}_k\}$

Gene Set Enrichment Analysis method (Subramanian *et al.* 2005)

- ▶ Construct null distribution of S_1, \dots, S_K by sample label (case-control) or gene-to-set membership permutation
- ▶ Using null distribution by permutation, estimate p-values and false discovery rates
- ▶ If we knew null distribution, we would not need expensive permutations.

Good:

- ▶ No cutoff/assumptions needed to estimate null distribution
- ▶ Aggregate scores across many genes! (boost the power)

Bad:

- ▶ What is an appropriate statistic?
- ▶ What should be permuted? For how long?

Simulation (Efron and Tibshirani 2007)

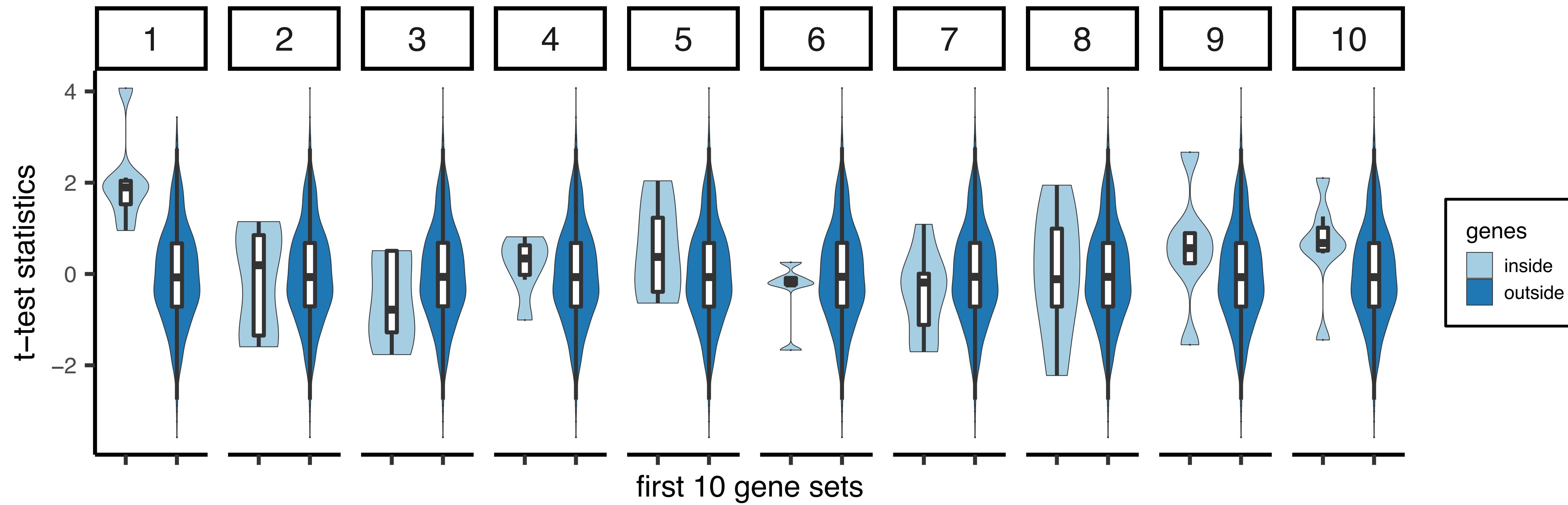
1. Generate basal gene expression

$$X_{i,g} \sim \mathcal{N}(0, 1)$$

2. Sample case vs. control membership (the rows of X) uniformly at random
3. Sample membership gene to gene set uniformly at random
4. For the first gene set, select a certain fraction of genes to perturb
5. For the selected genes g^* , add some Δ value to X_{i,g^*} if the sample i belongs to the control group

What will be a proper gene set score?

Gene-level scores across gene sets:



What will be a proper gene set score?

Can we simply aggregate gene-level z-scores (or t-statistics) within each set?

Irizarry *et al.* (2009), using Stouffer Z-score

$$S_k = \sum_{g \in \mathcal{C}_k} z_g / \sqrt{|\mathcal{C}_k|} \sim \mathcal{N}(0, 1)$$

if $Z_g \sim \mathcal{N}(0, 1)$, $\forall g$

```
geneset.score <- function(X, S, Y) {  
  z.genes <- run.t.test(X, Y)  
  n.sets <- apply(S, 1, sum)  
  z.sets <- (S %*% z.genes /  
             sqrt(n.sets))  
}
```

```
z.sets <- geneset.score(dat$X, dat$S, dat$Y)
```

What will be a proper gene set score?

Can we simply aggregate gene-level z-scores (or t-statistics) within each set?

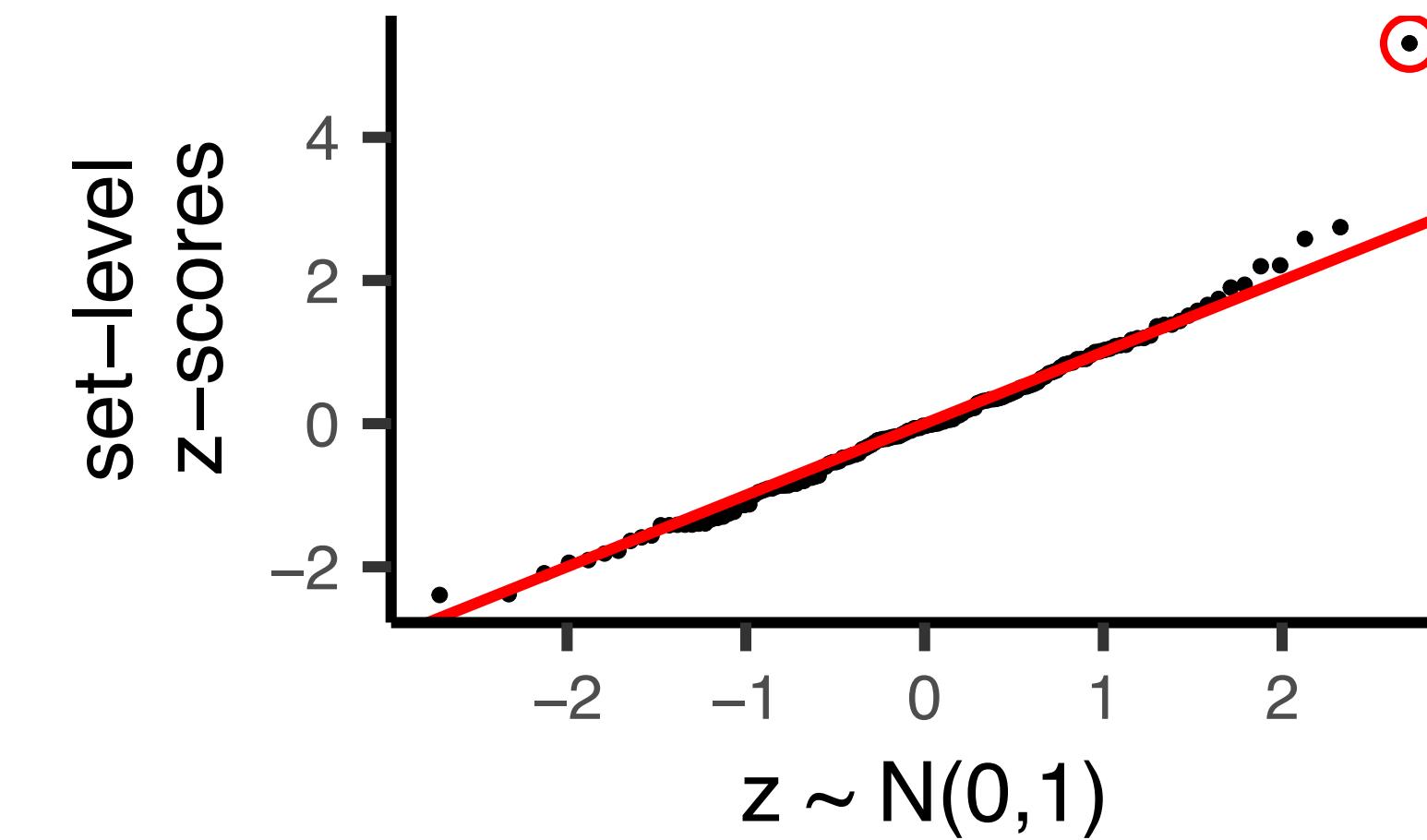
Irizarry *et al.* (2009), using Stouffer Z-score

$$S_k = \sum_{g \in \mathcal{C}_k} z_g / \sqrt{|\mathcal{C}_k|} \sim \mathcal{N}(0, 1)$$

if $Z_g \sim \mathcal{N}(0, 1), \forall g$

```
geneset.score <- function(X, S, Y) {  
  z.genes <- run.t.test(X, Y)  
  n.sets <- apply(S, 1, sum)  
  z.sets <- (S %*% z.genes /  
             sqrt(n.sets))  
}
```

```
z.sets <- geneset.score(dat$X, dat$S, dat$Y)
```



What will be a proper gene set score?

Can we simply aggregate gene-level z-scores (or t-statistics) within each set?

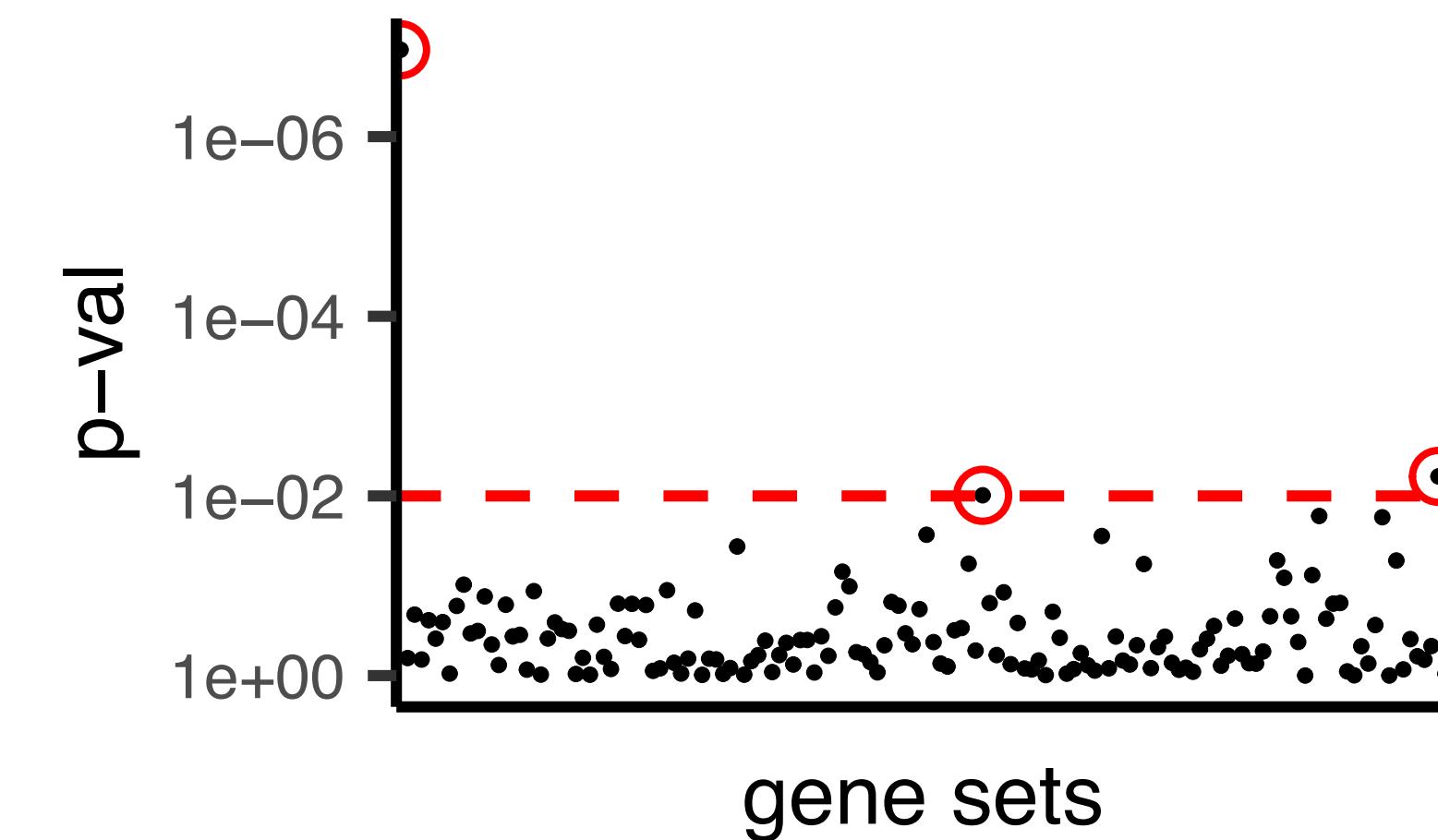
Irizarry *et al.* (2009), using Stouffer Z-score

$$S_k = \sum_{g \in \mathcal{C}_k} z_g / \sqrt{|\mathcal{C}_k|} \sim \mathcal{N}(0, 1)$$

if $Z_g \sim \mathcal{N}(0, 1), \forall g$

```
geneset.score <- function(X, S, Y) {  
  z.genes <- run.t.test(X, Y)  
  n.sets <- apply(S, 1, sum)  
  z.sets <- (S %*% z.genes /  
             sqrt(n.sets))  
}
```

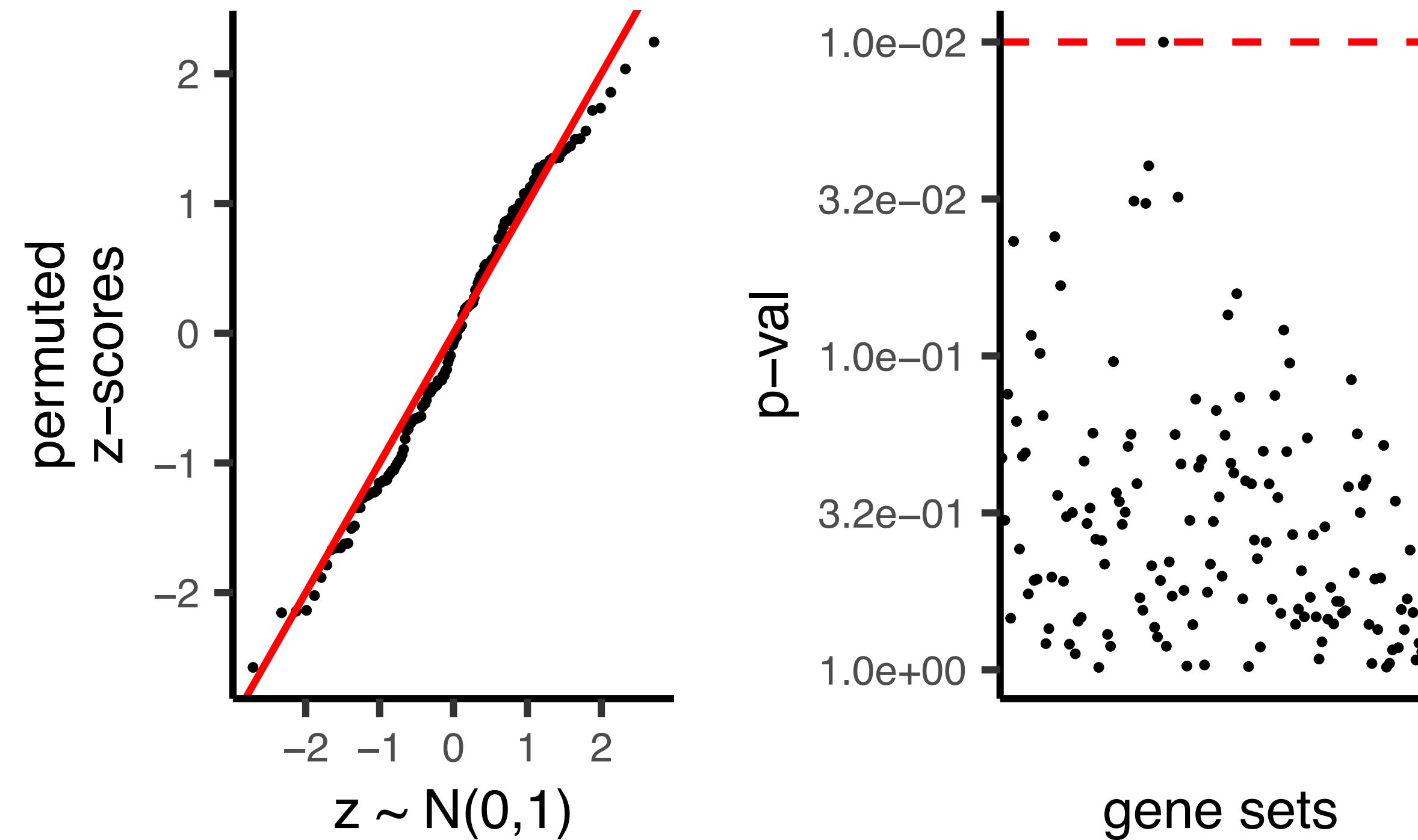
```
z.sets <- geneset.score(dat$X, dat$S, dat$Y)
```



Constructing null distribution by gene permutation

What if we don't know the distribution of set-wise scores?

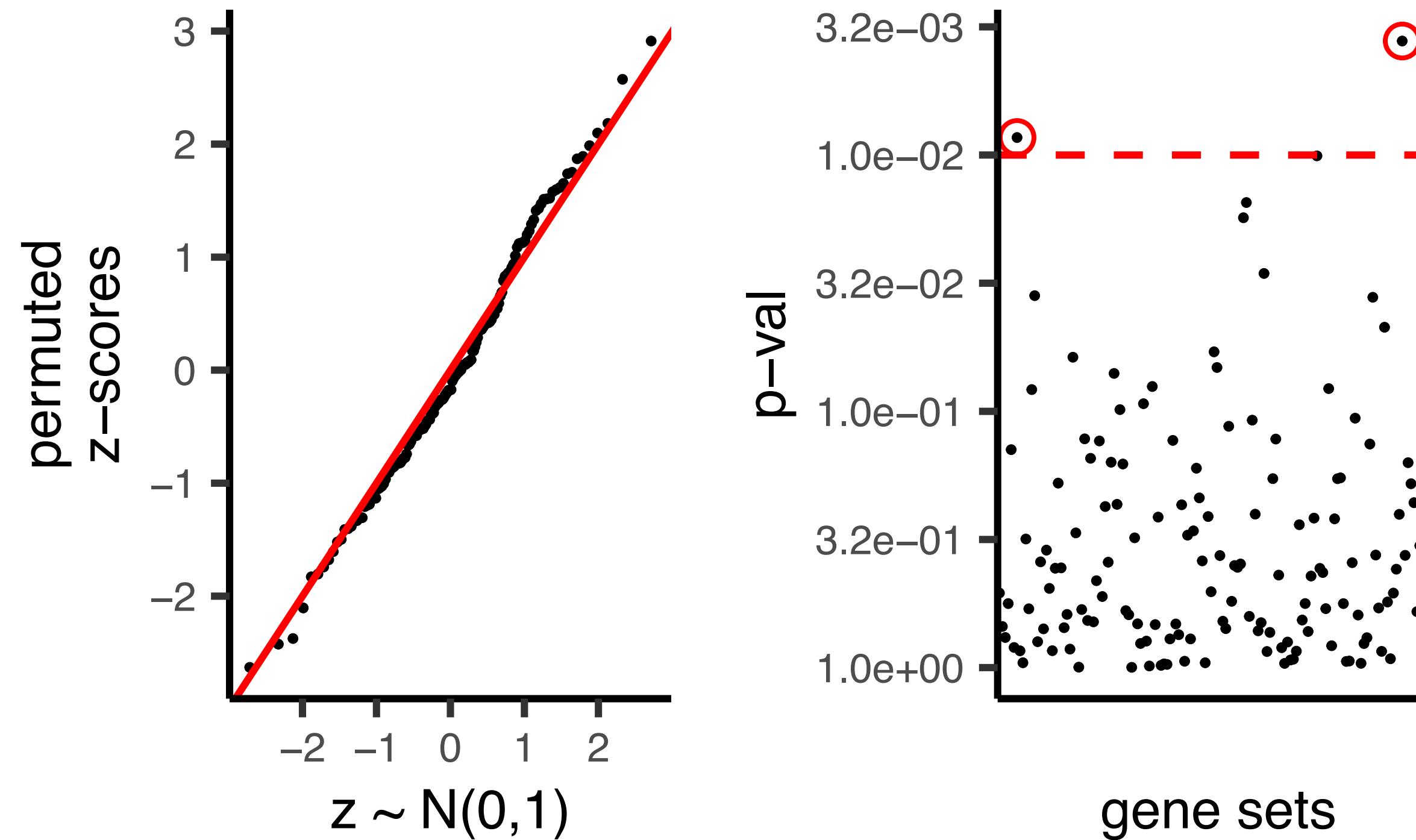
```
S.perm <- t(apply(dat$S, 1, sample))  
z.perm <- geneset.score(dat$x, S.perm, dat$y)
```



- ▶ Repeat the permutation of gene set membership matrix while preserving the number of genes within each set
- ▶ Compute set-level z-scores (or a similar kind) and construct null distribution
- ▶ Calculate p-values by counting the frequency of observed $S_k^* > S_k^{\text{perm}}$

Constructing null distribution by sample permutation

```
Y.perm <- sample(dat$Y)  
z.perm <- geneset.score(dat$x, S.perm, Y.perm)
```



- ▶ Repeat the permutation of case-control labels while preserving the same number of cases and controls
- ▶ Compute set-level z-scores (or a similar kind) and construct null distribution
- ▶ Calculate p-values by counting the frequency of observed $S_k^* > S_k^{\text{perm}}$

Today's lecture: Enrichment Analysis

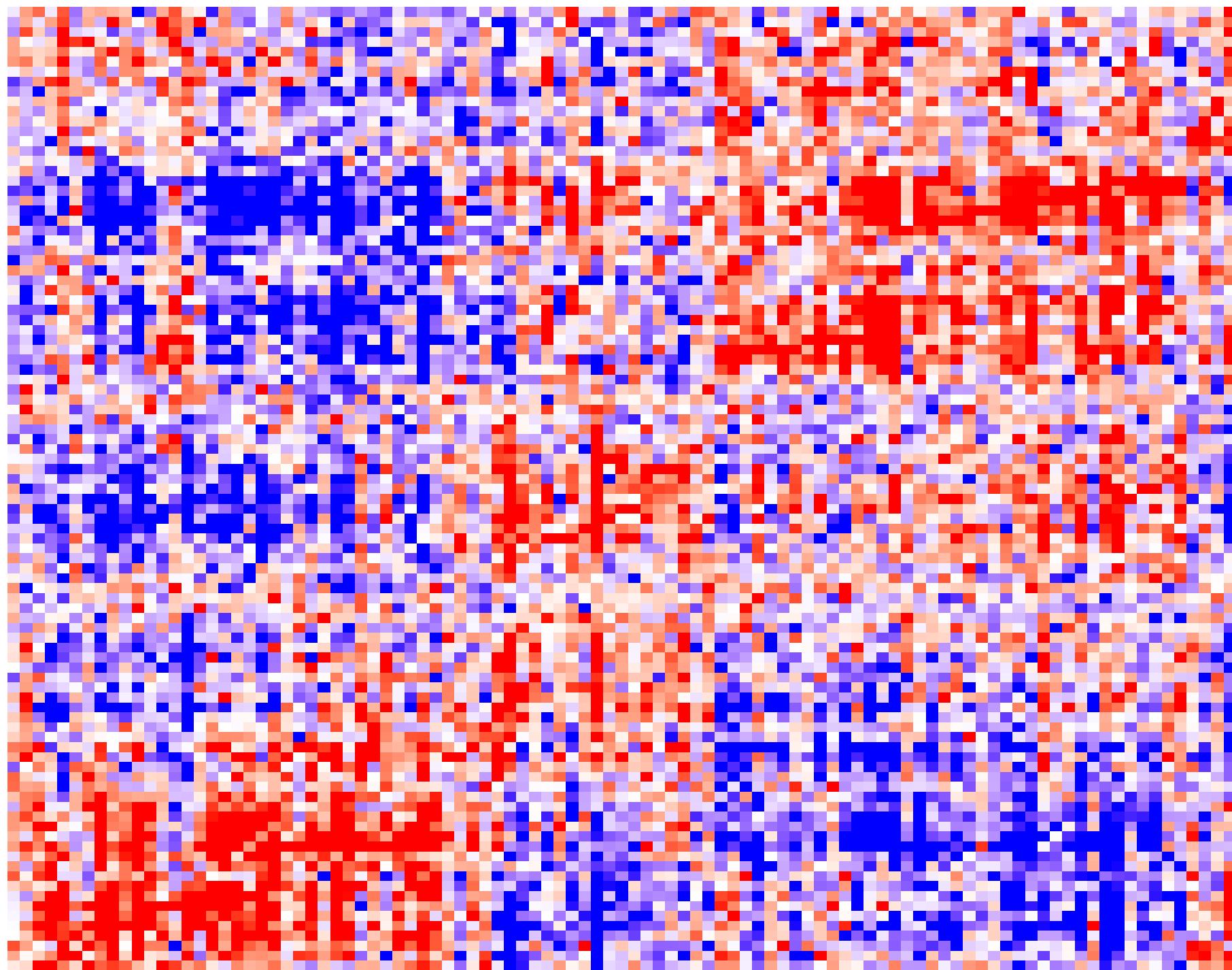
- **Motivations: What's next after genomics analysis?**
 - What have we learned? How do we know that our discovery is meaningful?
- **Gene set enrichment analysis**
 - Set-based approach: Hypergeometric test
 - Rank-based approach: GSEA by KS statistic
- **Can we engineer new gene sets/scores?**
 - Principal Component Analysis
 - Matrix factorization of count data

A toy example: Can we uncover hidden structures in a gene expression matrix?

Suppose the data matrix (gene expression data) were generated by the following:

$$\mathbf{x}_i = \mathbf{u}_i V^\top + \epsilon_i$$

for all i with $\epsilon_i \sim \mathcal{N}(\mathbf{0}, I)$



Can we reverse engineer the U and V matrices from the data X ?

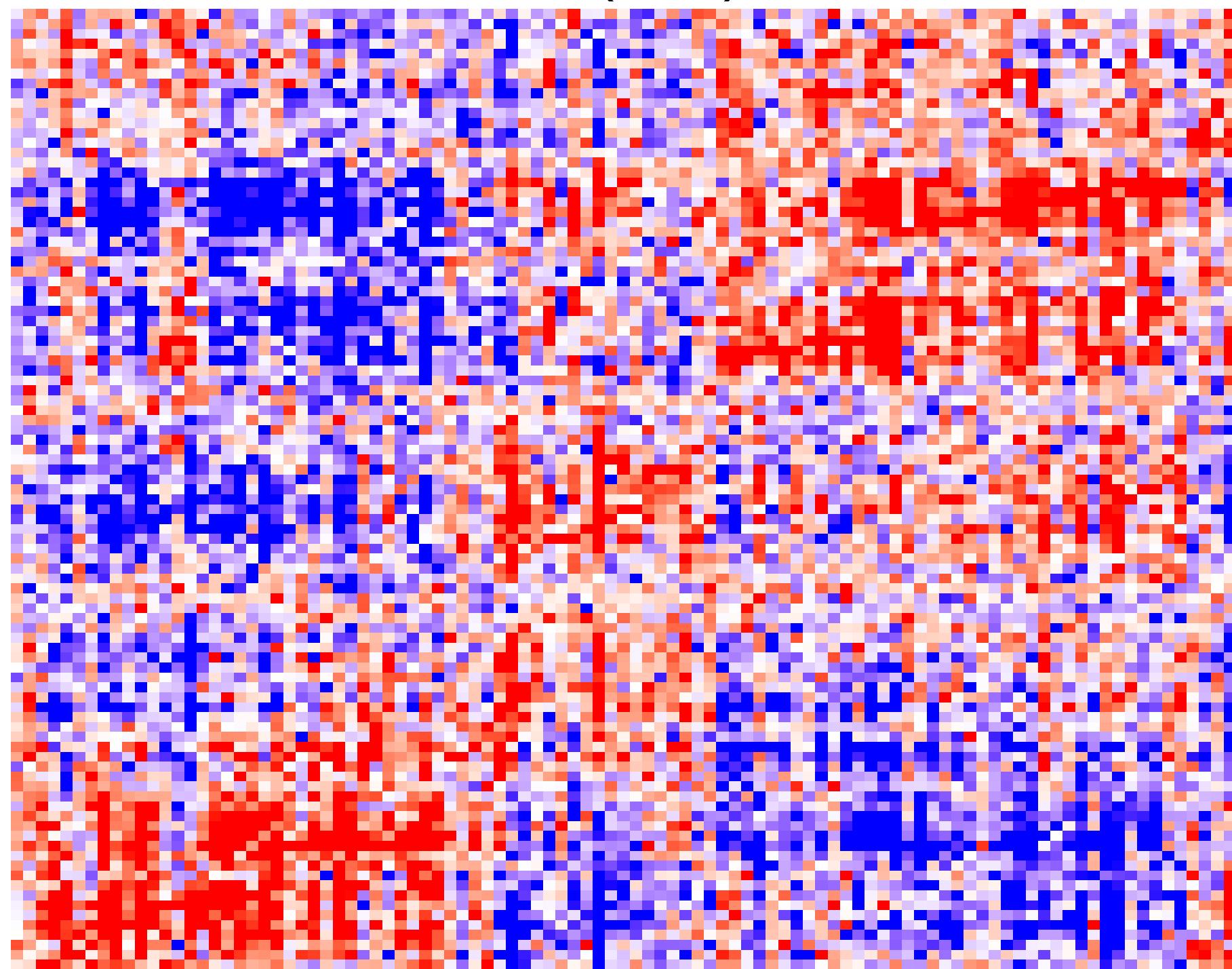
$$X \rightarrow UV^\top$$

A toy example: Can we uncover hidden structures in a gene expression matrix?

Suppose the data matrix (gene expression data) were generated by the following:

$$\mathbf{x}_i = \mathbf{u}_i V^\top + \epsilon_i$$

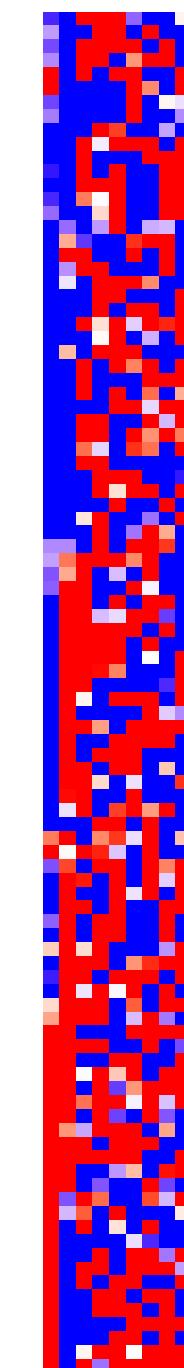
for all i with $\epsilon_i \sim \mathcal{N}(\mathbf{0}, I)$



Can we reverse engineer the U and V matrices from the data X ?

$$X \rightarrow UV^\top$$

features



feature loading

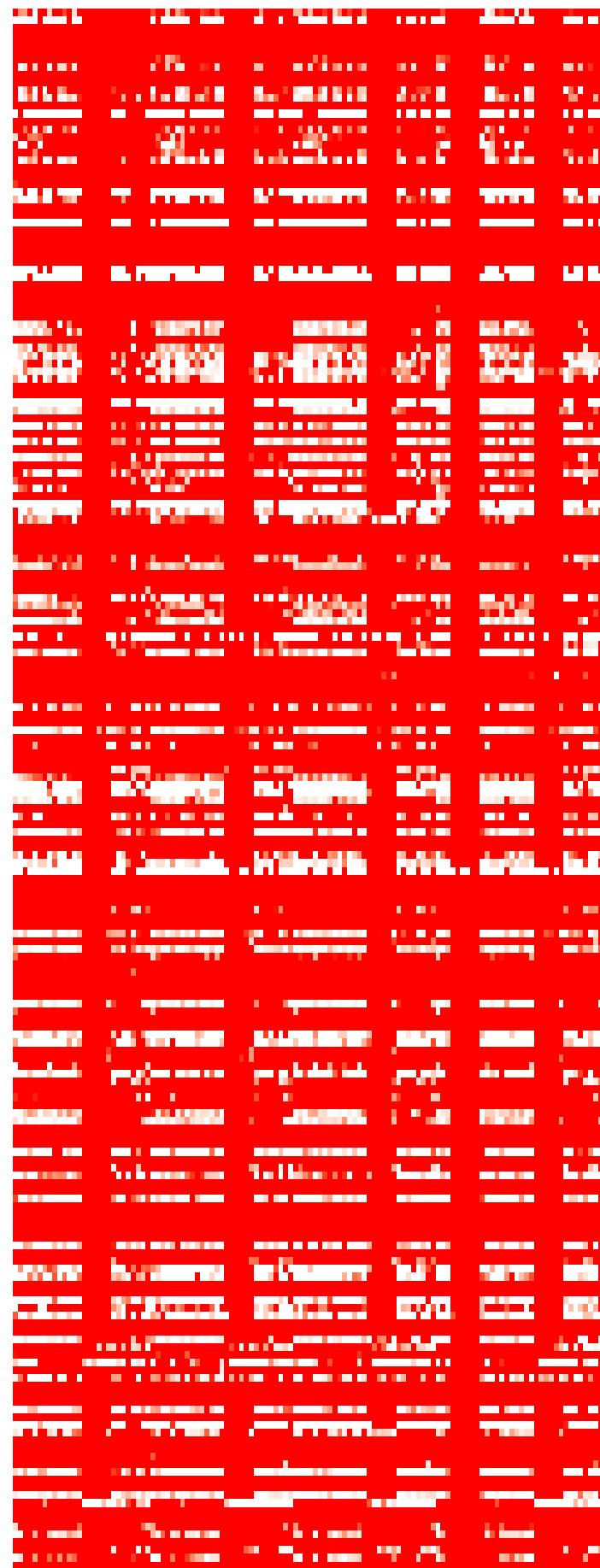
X

Today's lecture: Enrichment Analysis

- **Motivations: What's next after genomics analysis?**
 - What have we learned? How do we know that our discovery is meaningful?
- **Gene set enrichment analysis**
 - Set-based approach: Hypergeometric test
 - Rank-based approach: GSEA by KS statistic
- **Can we engineer new gene sets/scores?**
 - Principal Component Analysis
 - Matrix factorization of count data

Working example: GSE107011

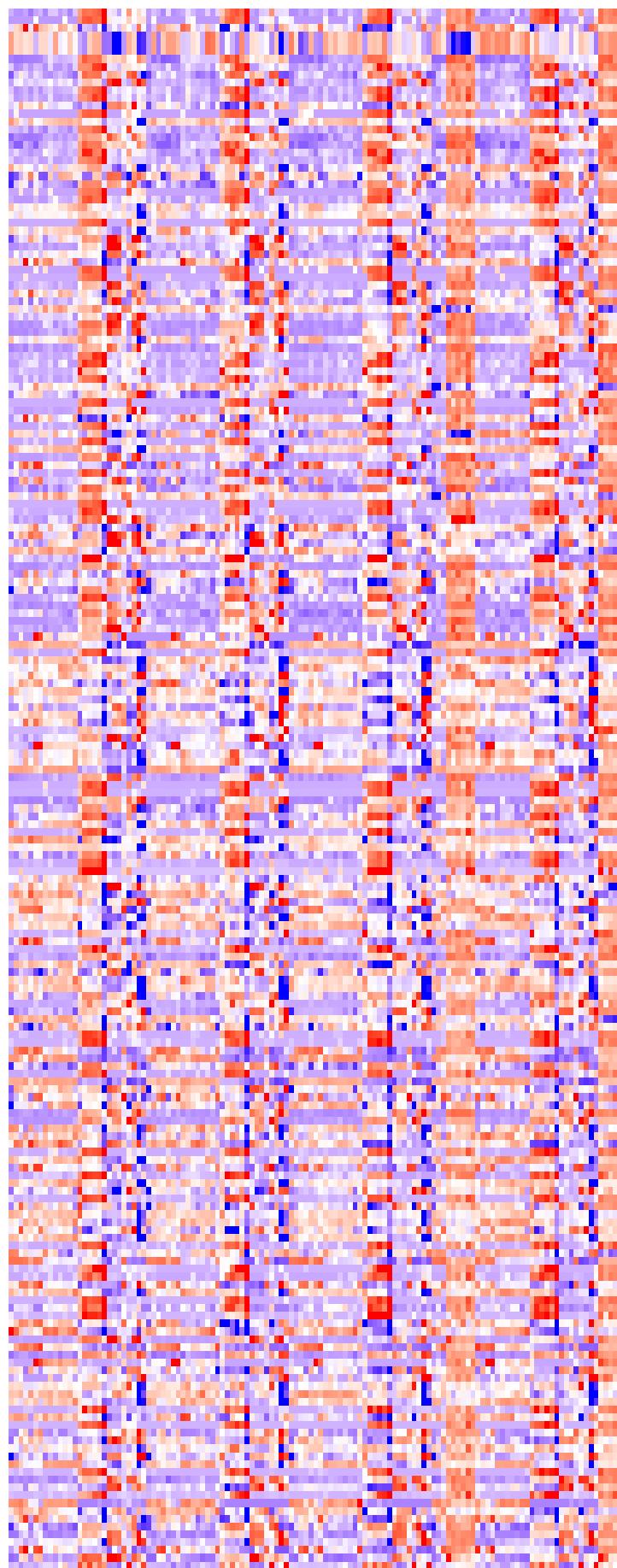
Top 200 most variable genes in the data matrix:



- ▶ We will call such a high-dimensional matrix X ($m \times n$)
- ▶ X has $m=58,311$ rows (transcripts/genes/features)
- ▶ X has $n=127$ columns (samples/#data points)
- ▶ The rows were log-transformed and scaled by `scale()` for visualization
- ▶ Each sample is a 58,311-dimensional vector!
- ▶ Each gene is a 127-dimensional vector...

Working example: GSE107011

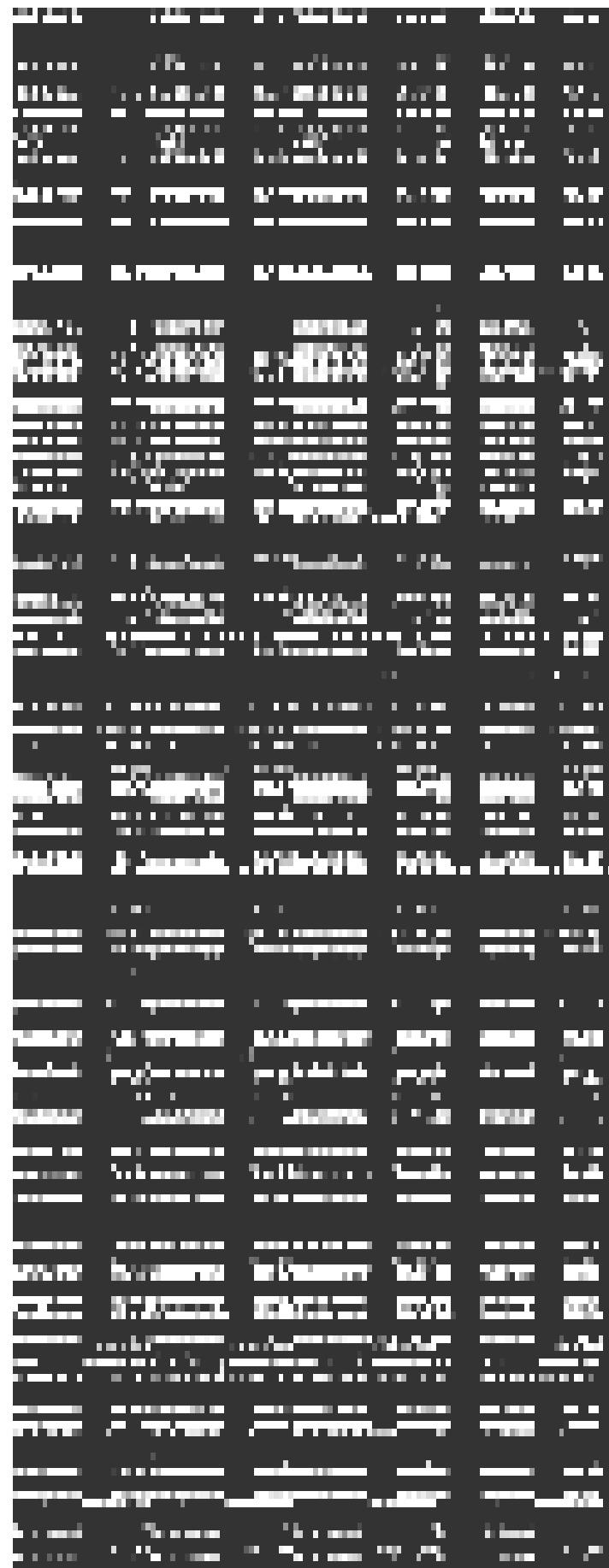
Top 200 most variable genes in the data matrix:



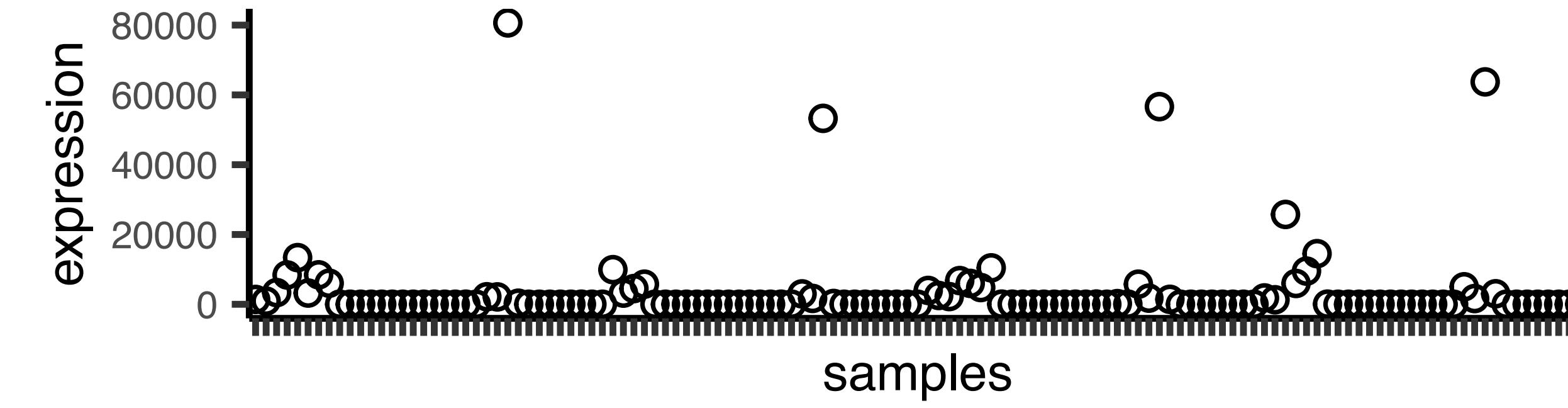
- ▶ We will call such a high-dimensional matrix X ($m \times n$)
- ▶ X has $m=58,311$ rows (transcripts/genes/features)
- ▶ X has $n=127$ columns (samples/#data points)
- ▶ The rows were log-transformed and scaled by `scale()` for visualization
- ▶ Each sample is a 58,311-dimensional vector!
- ▶ Each gene is a 127-dimensional vector...

1-dimensional representations/summary of data matrix

Top 200 genes:

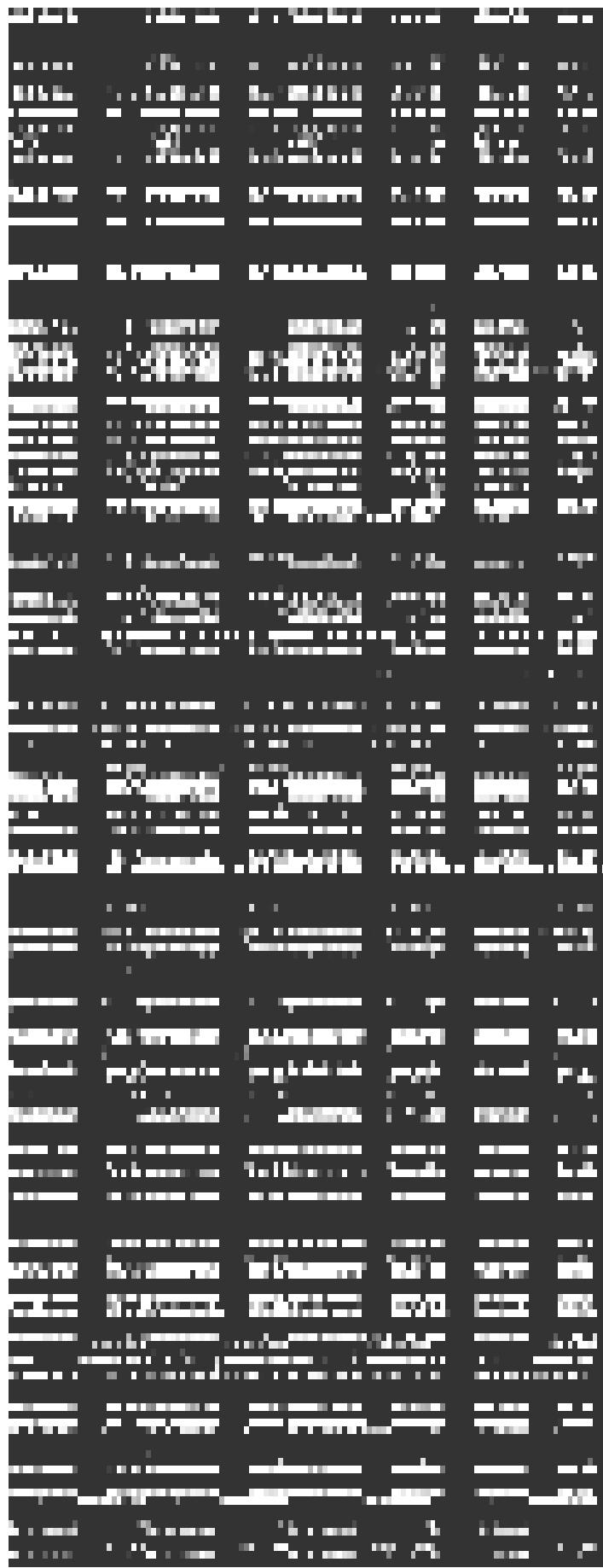


Each sample (column):

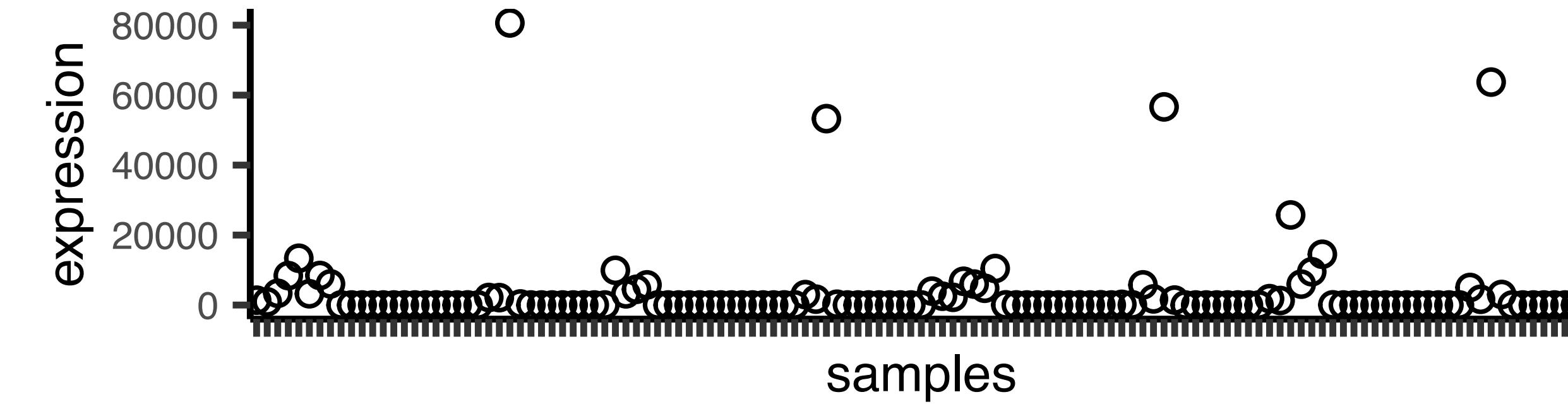


1-dimensional representations/summary of data matrix

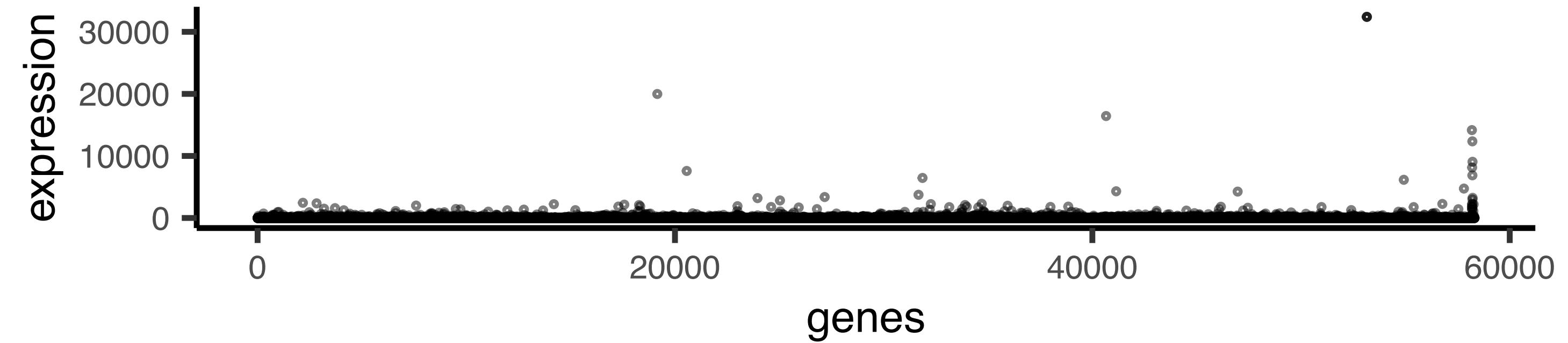
Top 200 genes:



Each sample (column):

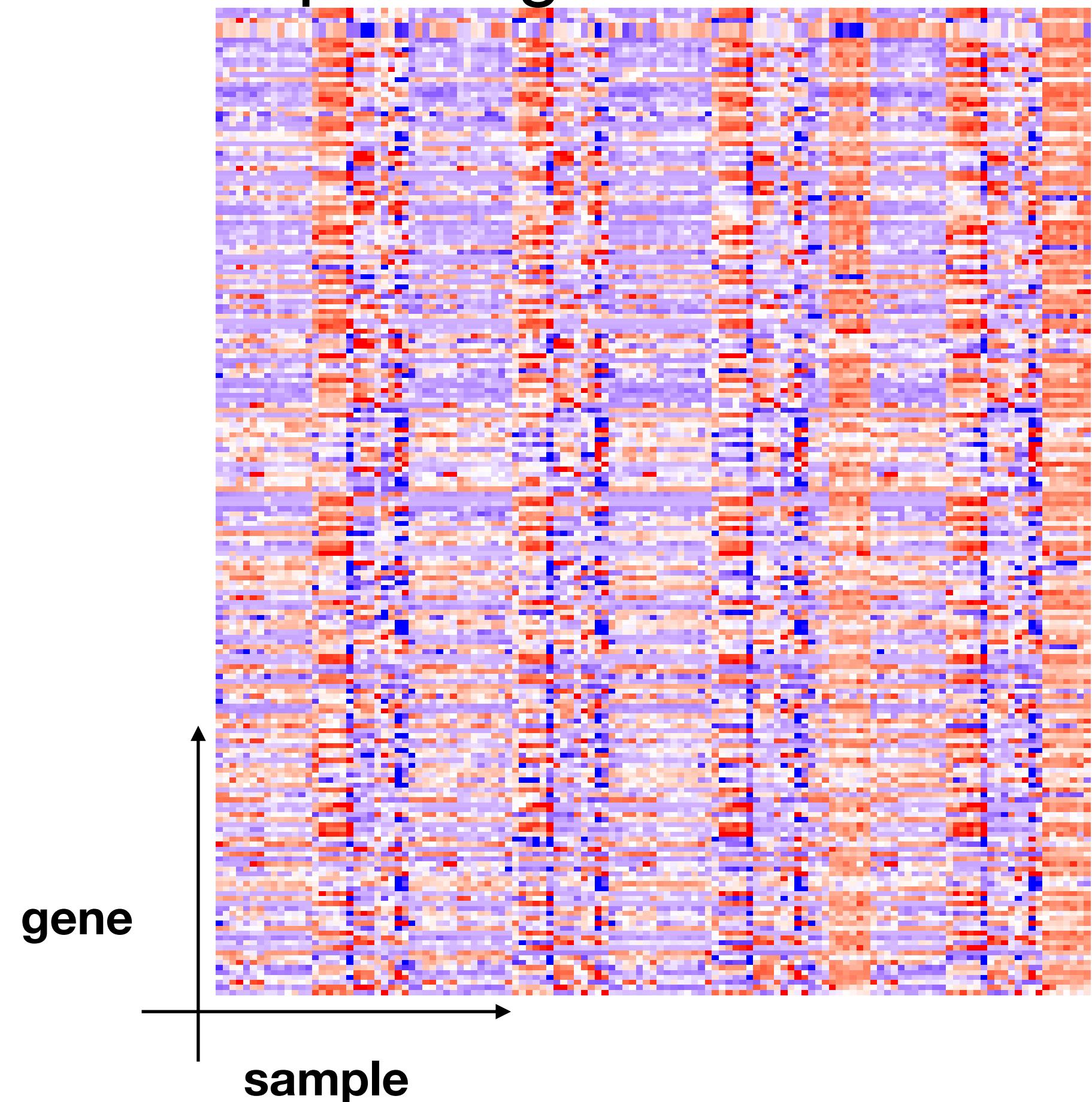


Each gene (row):



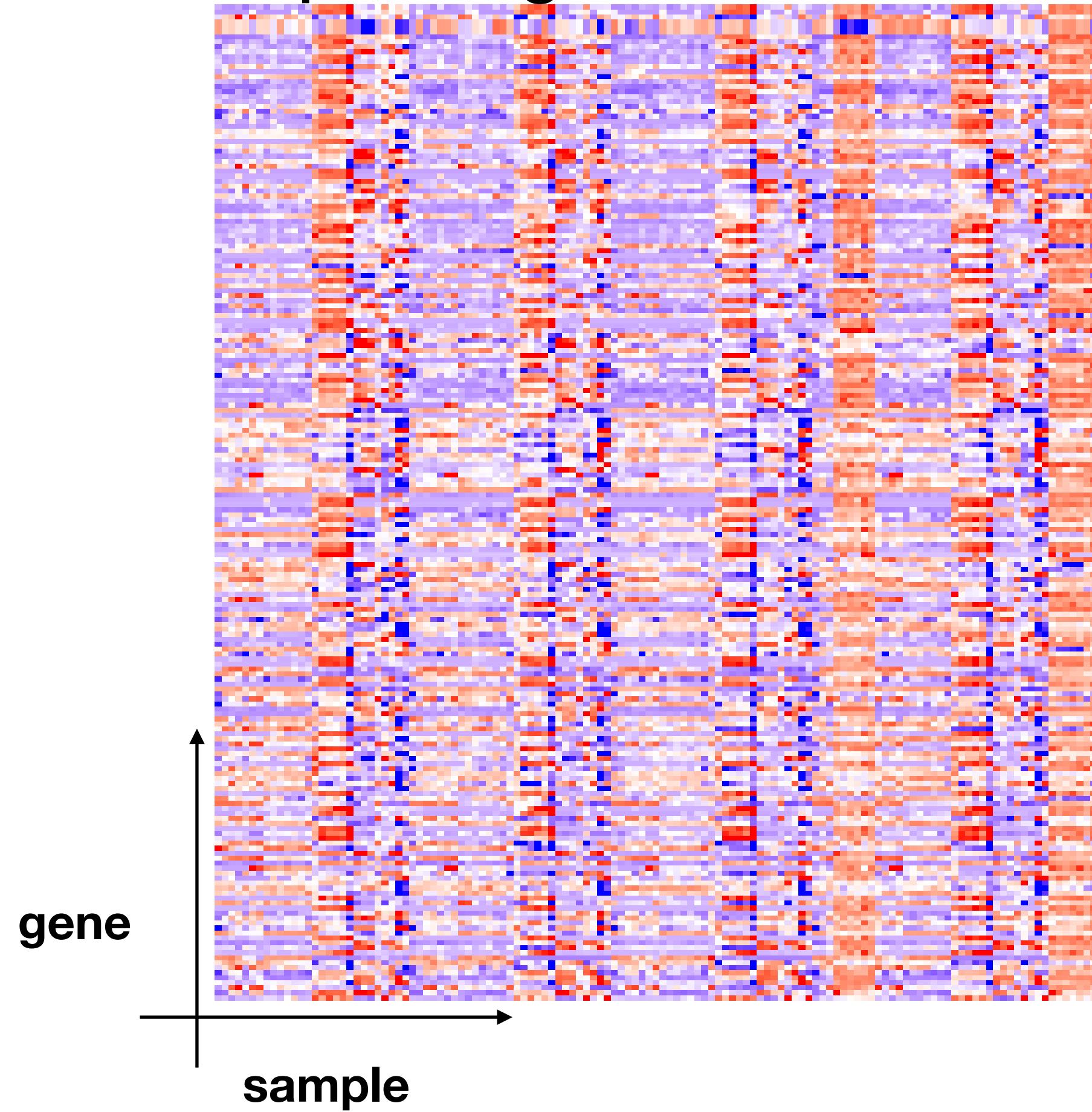
Can you see some common patterns?

Top 200 genes

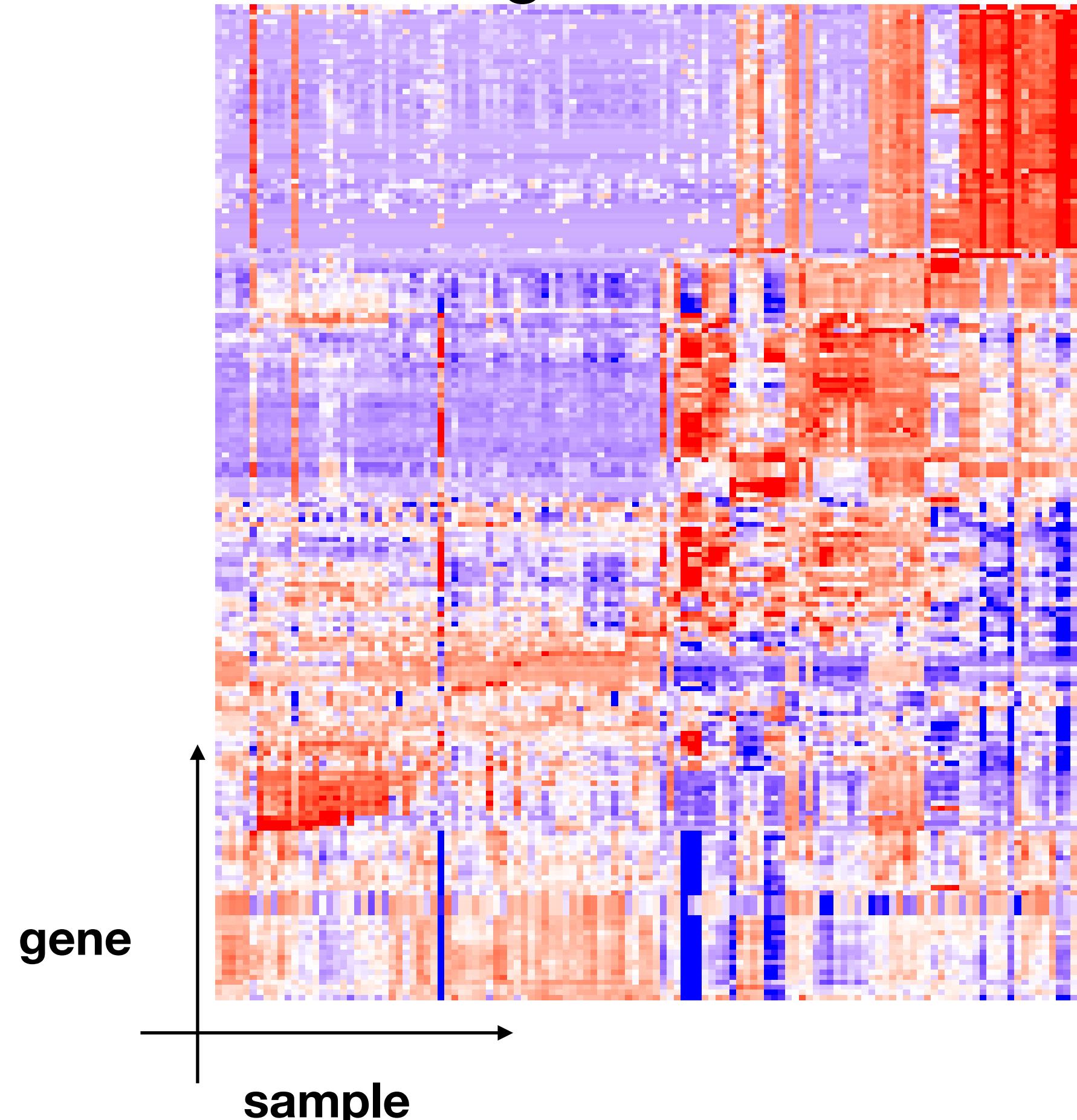


Can you see some common patterns?

Top 200 genes



Rearrange them...



PCA: How do we recover “common” patterns from data?

Principal Component Analysis

(Pearson 1901)

- ▶ Projection [of the original data] that minimizes the projection cost between the original and projected
- ▶ The cost = mean squared distance

(Hotelling 1933)

- ▶ Orthogonal projection of data into a lower-dimensional **[principal]** sub-space,
- ▶ such that the total **variation of the projected** is maximized

What do we mean by projection to a low-dimensional space?

Recap: We learned that a multivariate linear regression projects a data vector \mathbf{x} onto the column space of the design matrix U :

$$\begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{pmatrix} = \begin{pmatrix} U_{11} & \dots & U_{1k} \\ U_{21} & \dots & U_{2k} \\ \vdots & \vdots & \vdots \\ U_{m1} & \dots & U_{mk} \end{pmatrix} \begin{pmatrix} W_1 \\ \vdots \\ W_k \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_m \end{pmatrix}$$

or

$$\mathbf{x} = U\mathbf{w} + \boldsymbol{\epsilon}, \text{ for many columns, } \mathbf{X} = UW + E.$$

- ▶ If we **knew** U , we would be able to solve weights W . If $k = 2$, it would result in a projection to the 2D space.
- ▶ Unlike regression, we need ask: How do we know this unknown U matrix?

What is a projection matrix?

The purpose of a multivariate regression-based prediction:

$$\mathbf{x} \rightarrow \hat{\mathbf{x}} = U\mathbf{w}$$

The least-square solution for the \mathbf{W} :

$$\hat{\mathbf{w}} = (U^\top U)^{-1} U^\top \mathbf{x}$$

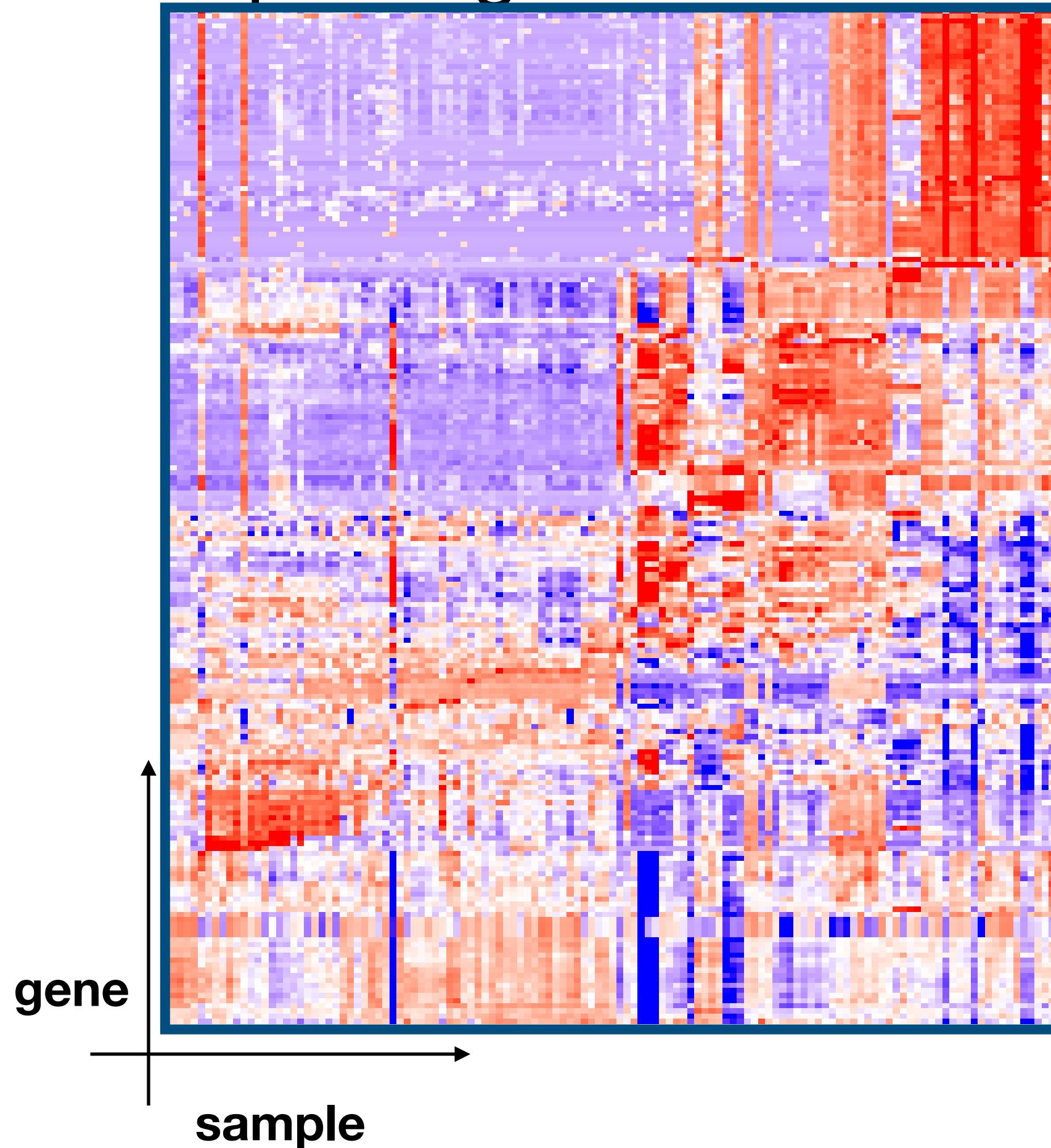
Then we have

$$\hat{\mathbf{x}} = \underbrace{U(U^\top U)^{-1} U^\top}_{\text{projection matrix}} \mathbf{x}$$

You can think of “projection” as “**prediction**” in the linear space defined by the columns of a design matrix.

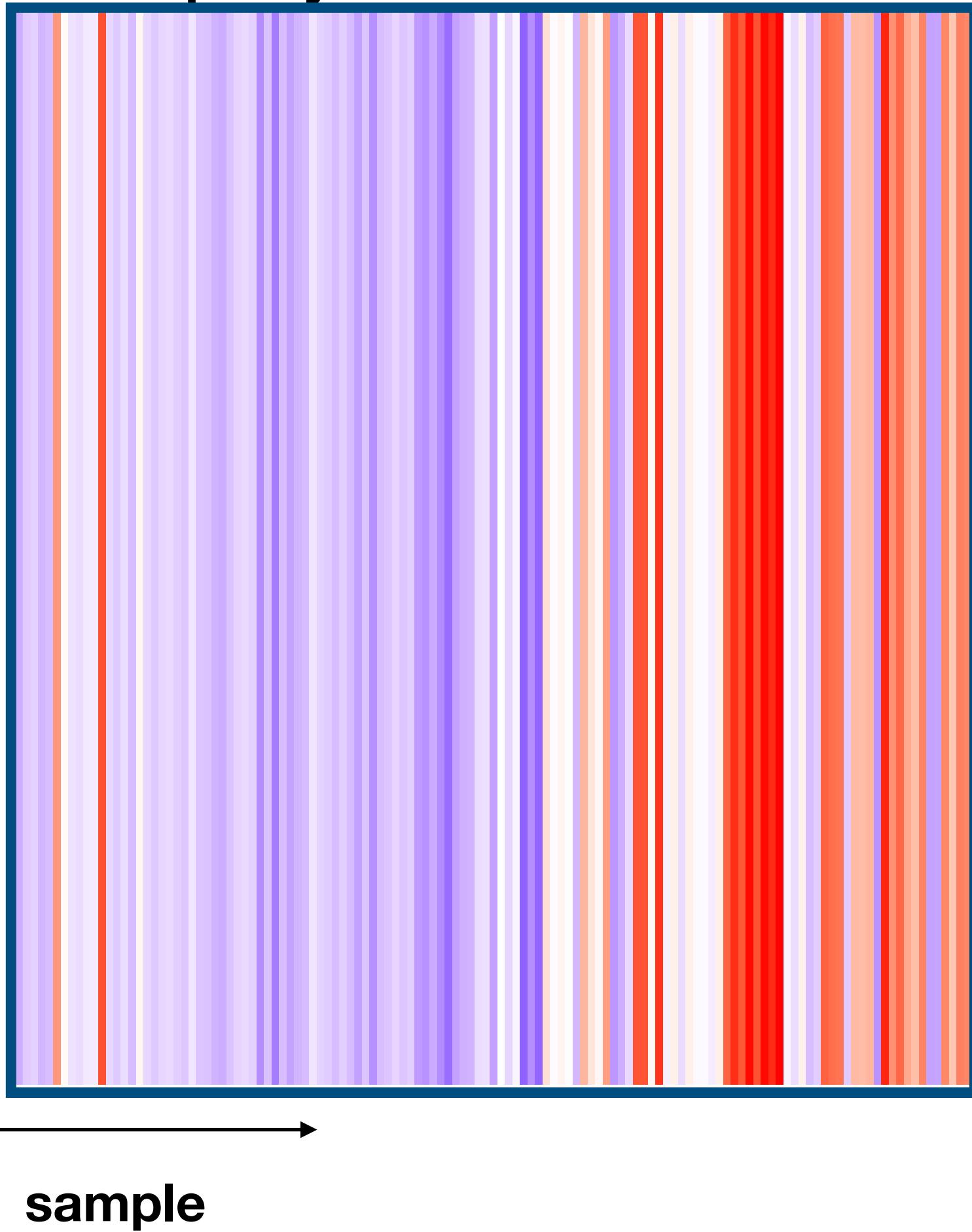
What will be “good” features (the U matrix) to regress on?

Top 200 genes



What will be “good” features (the U matrix) to regress on?

one proj.



=

1's

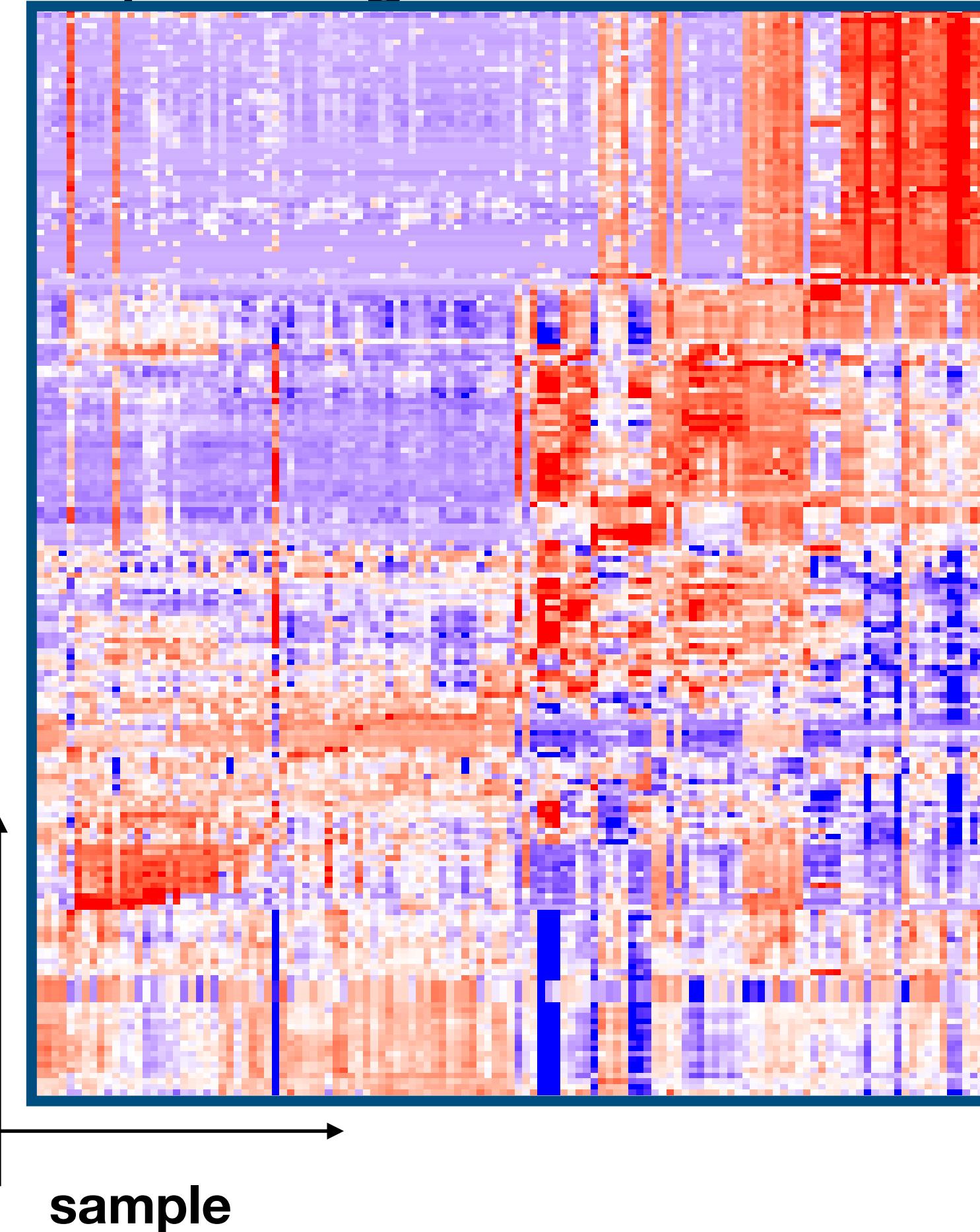
how much contribution?

X

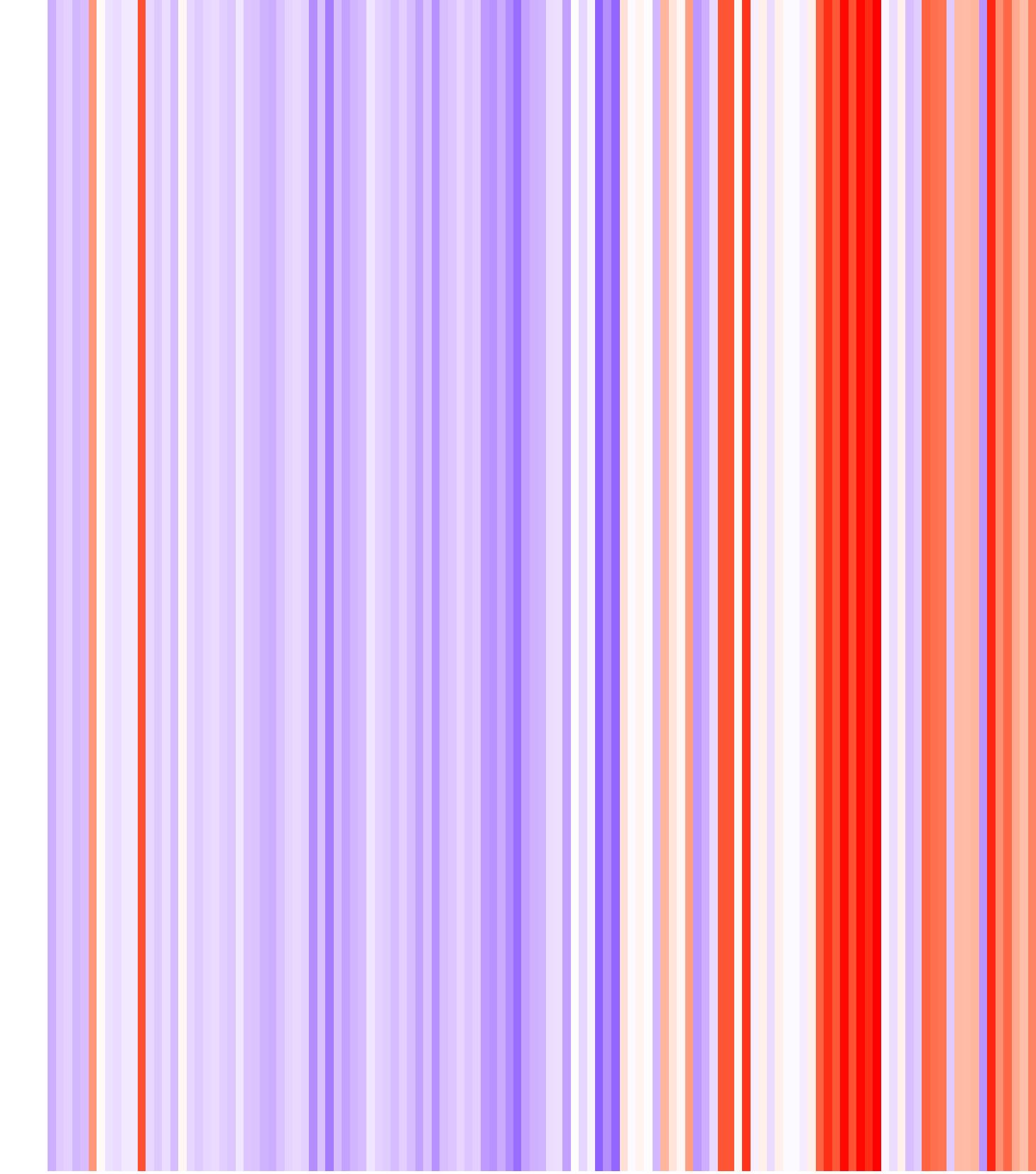


What will be “good” features (the U matrix) to regress on?

Top 200 genes

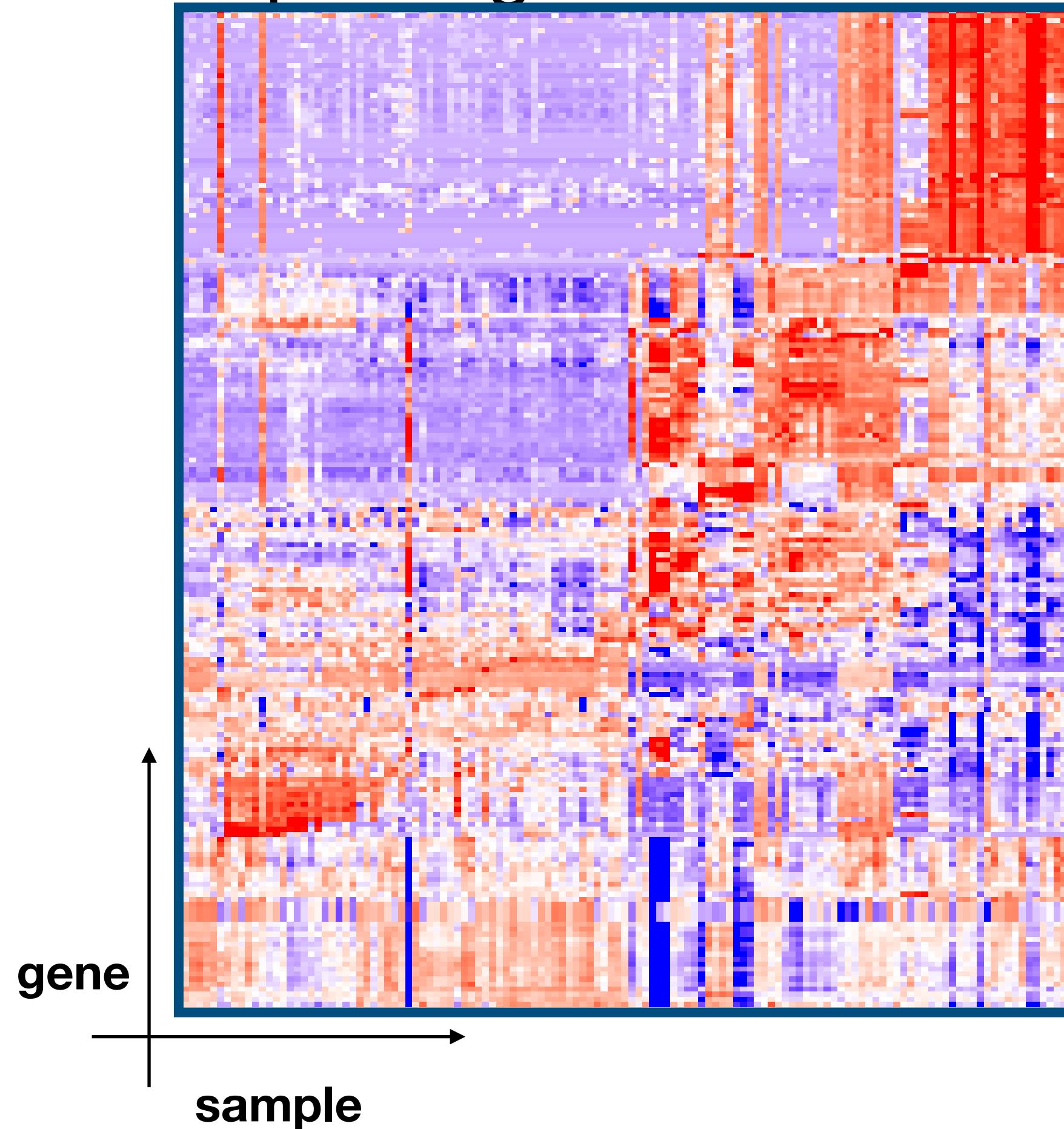


1% var. explained

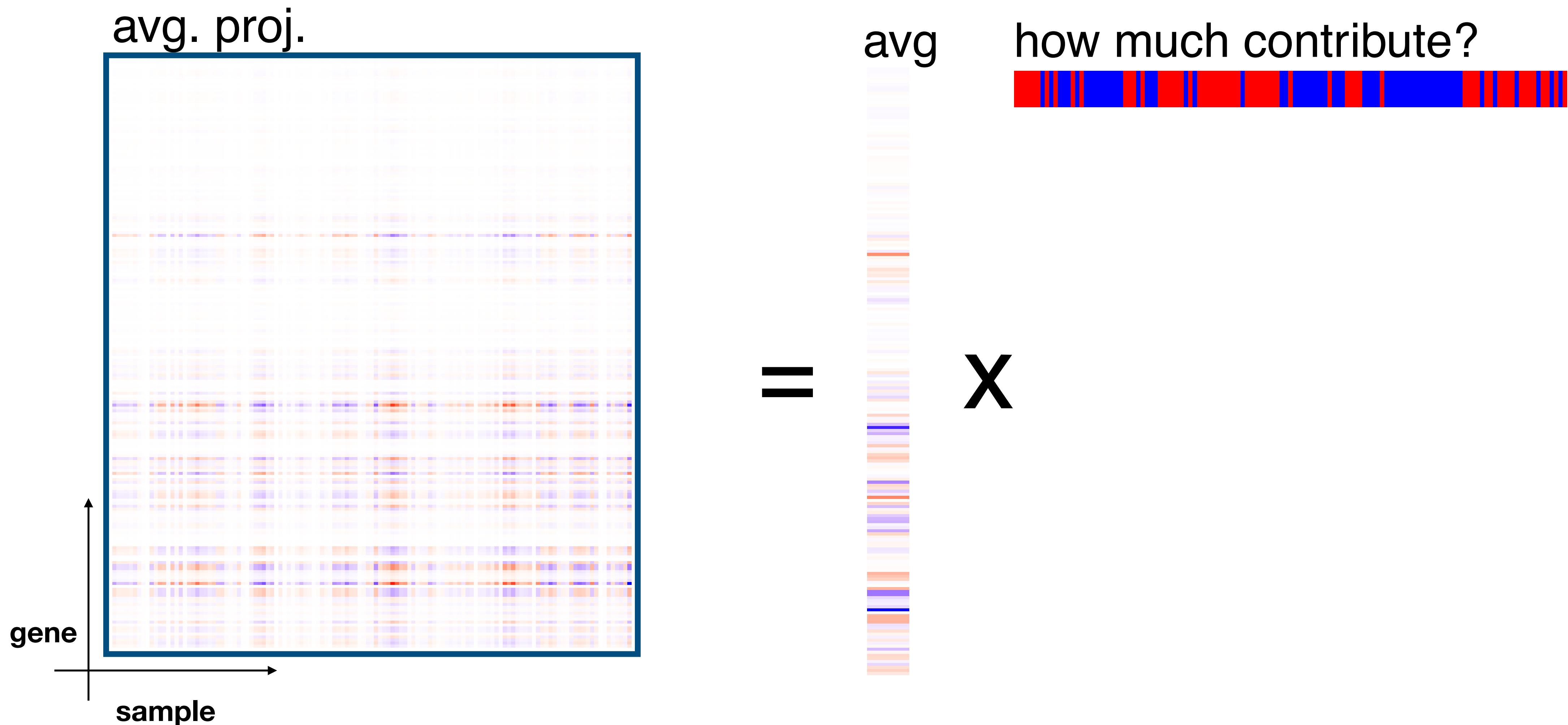


How about an average vector?

Top 200 genes

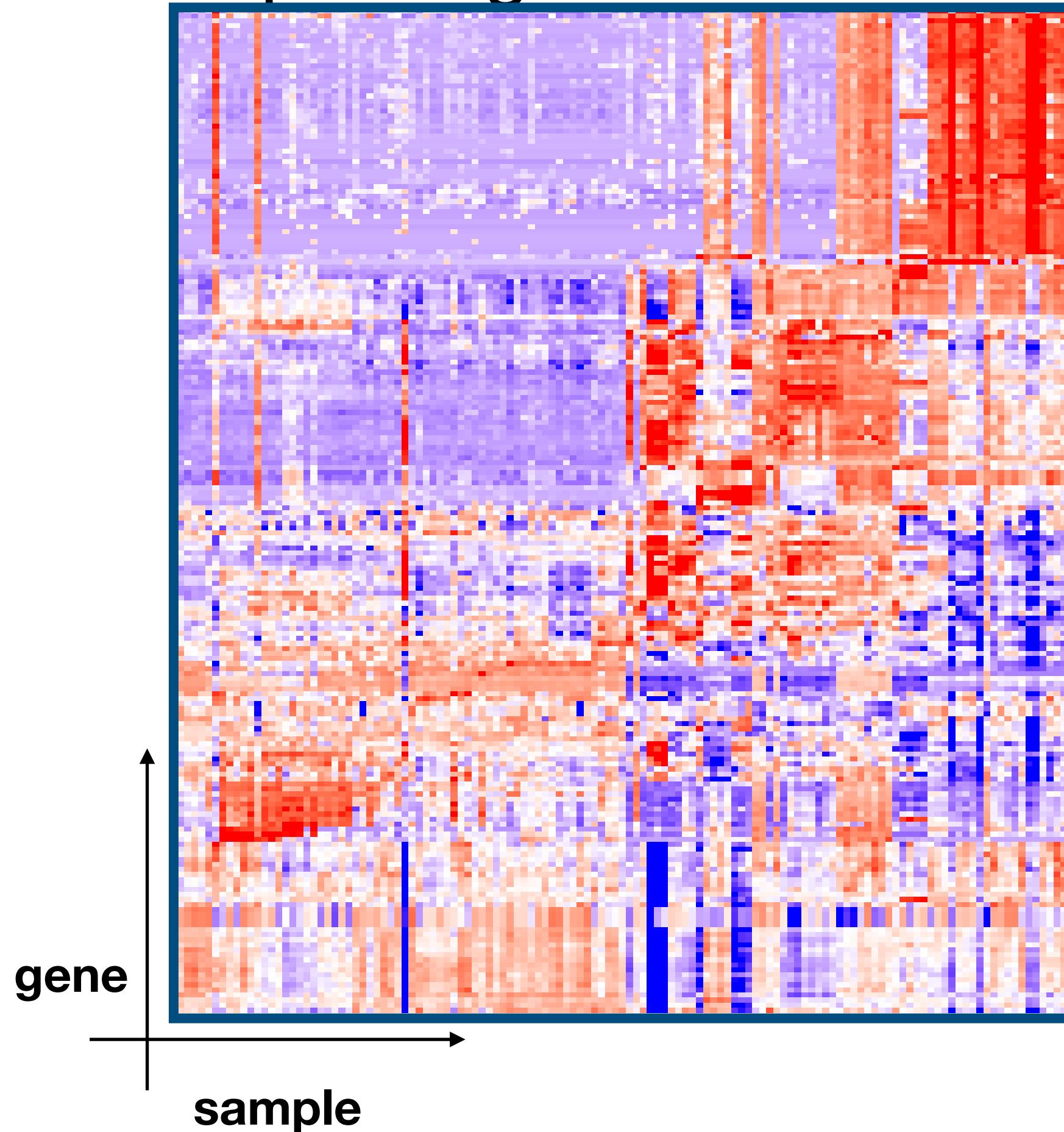


How about an average vector?

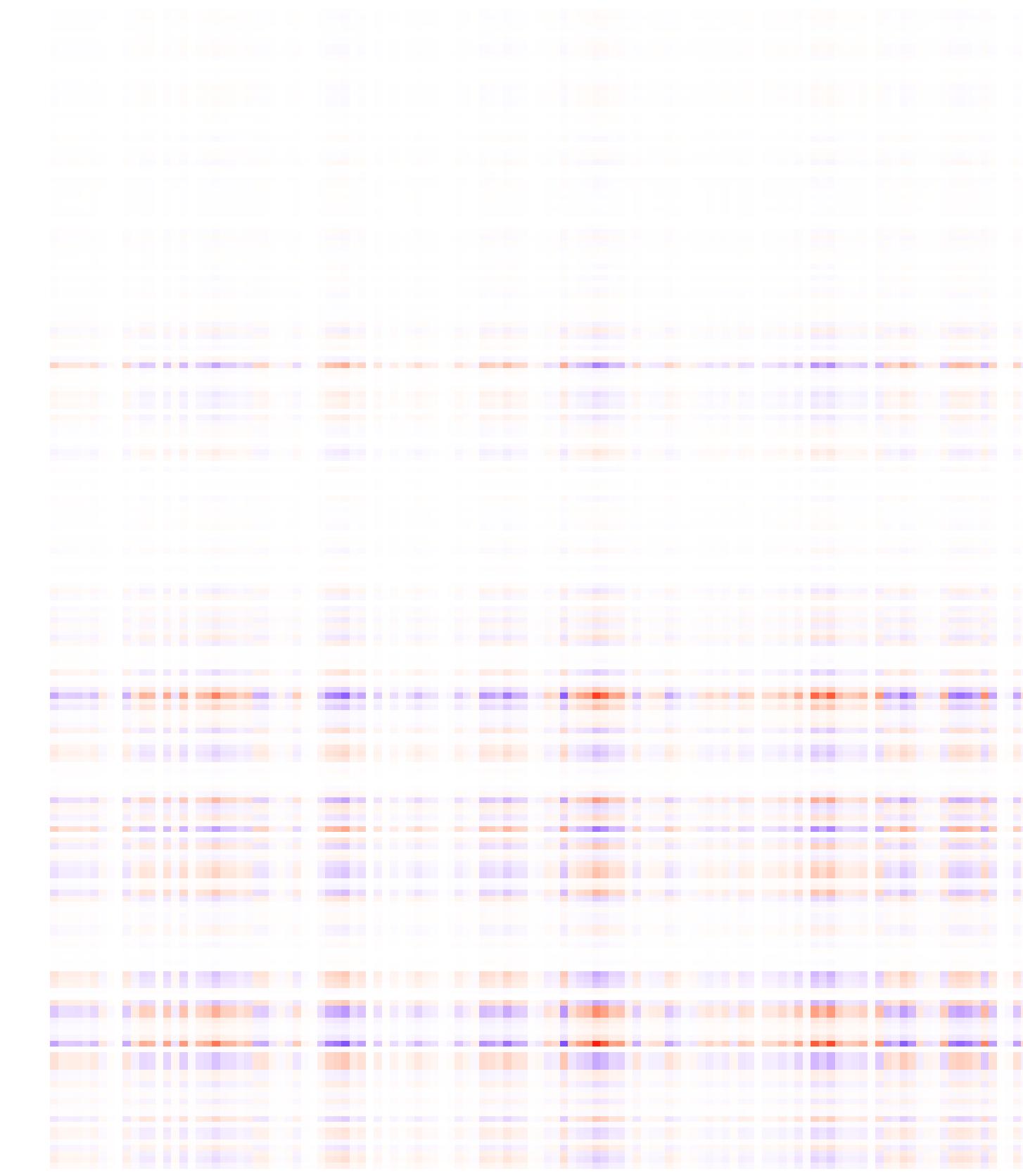


How about an average vector?

Top 200 genes

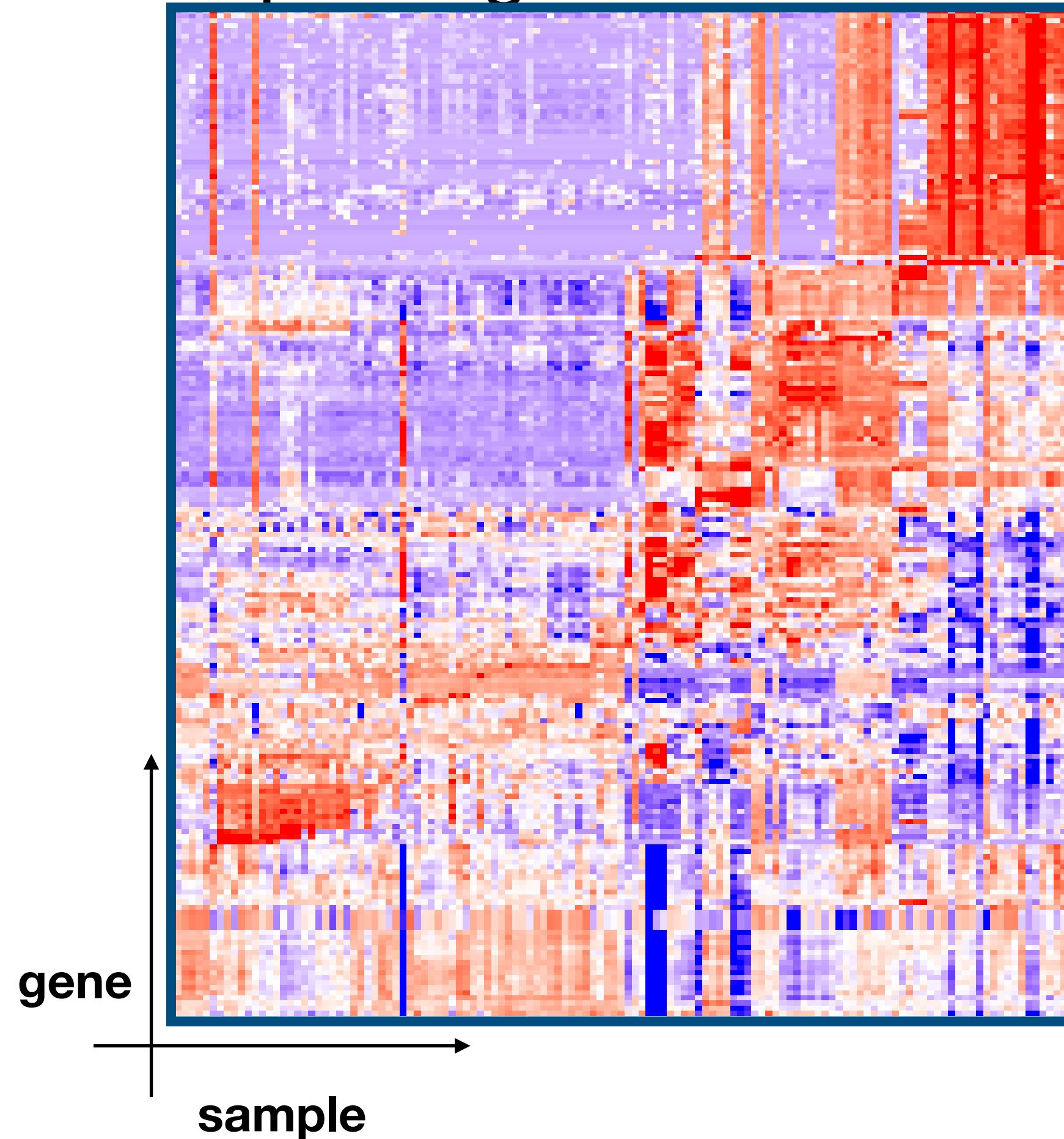


1%



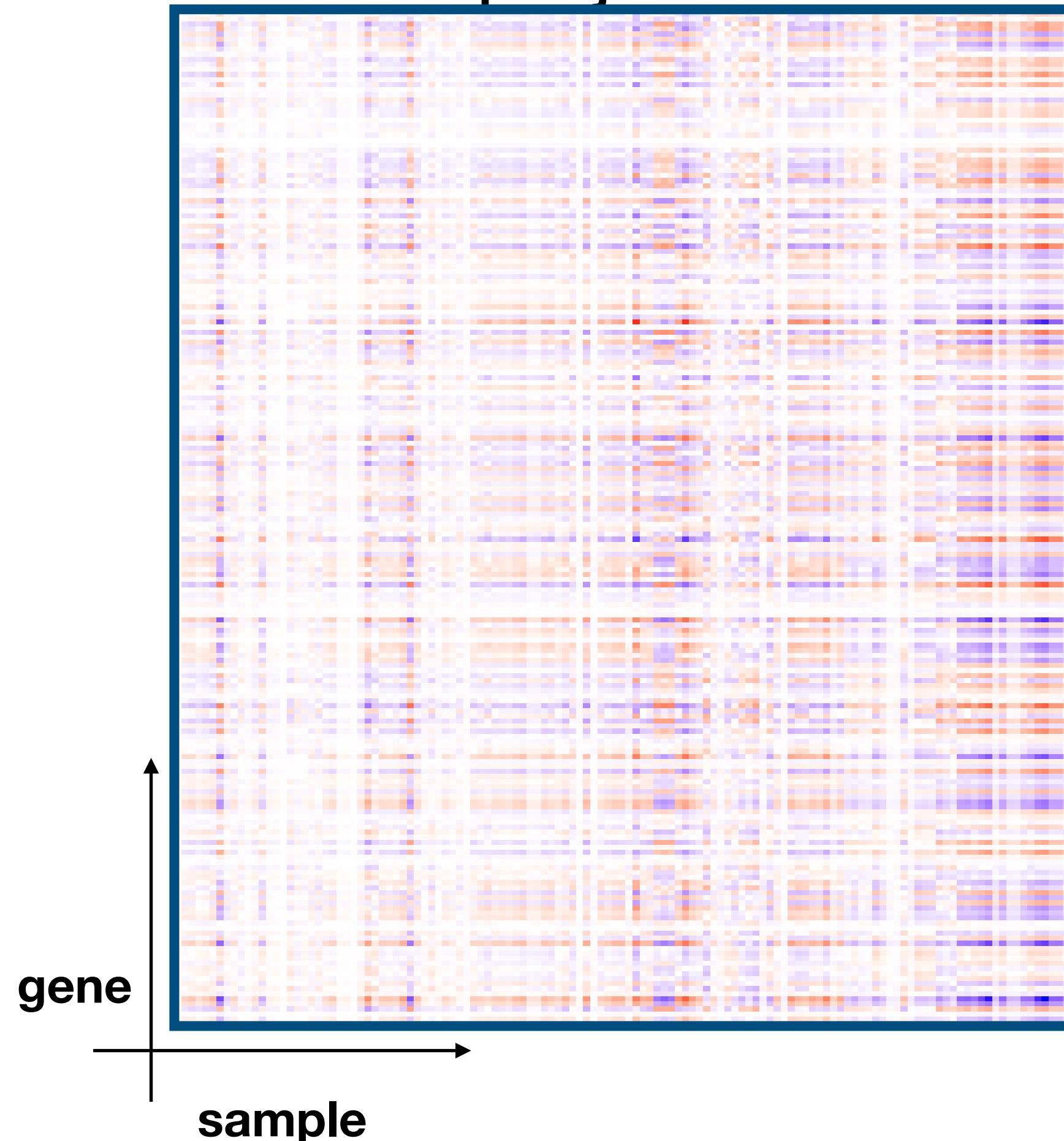
Let's try out random projection

Top 200 genes



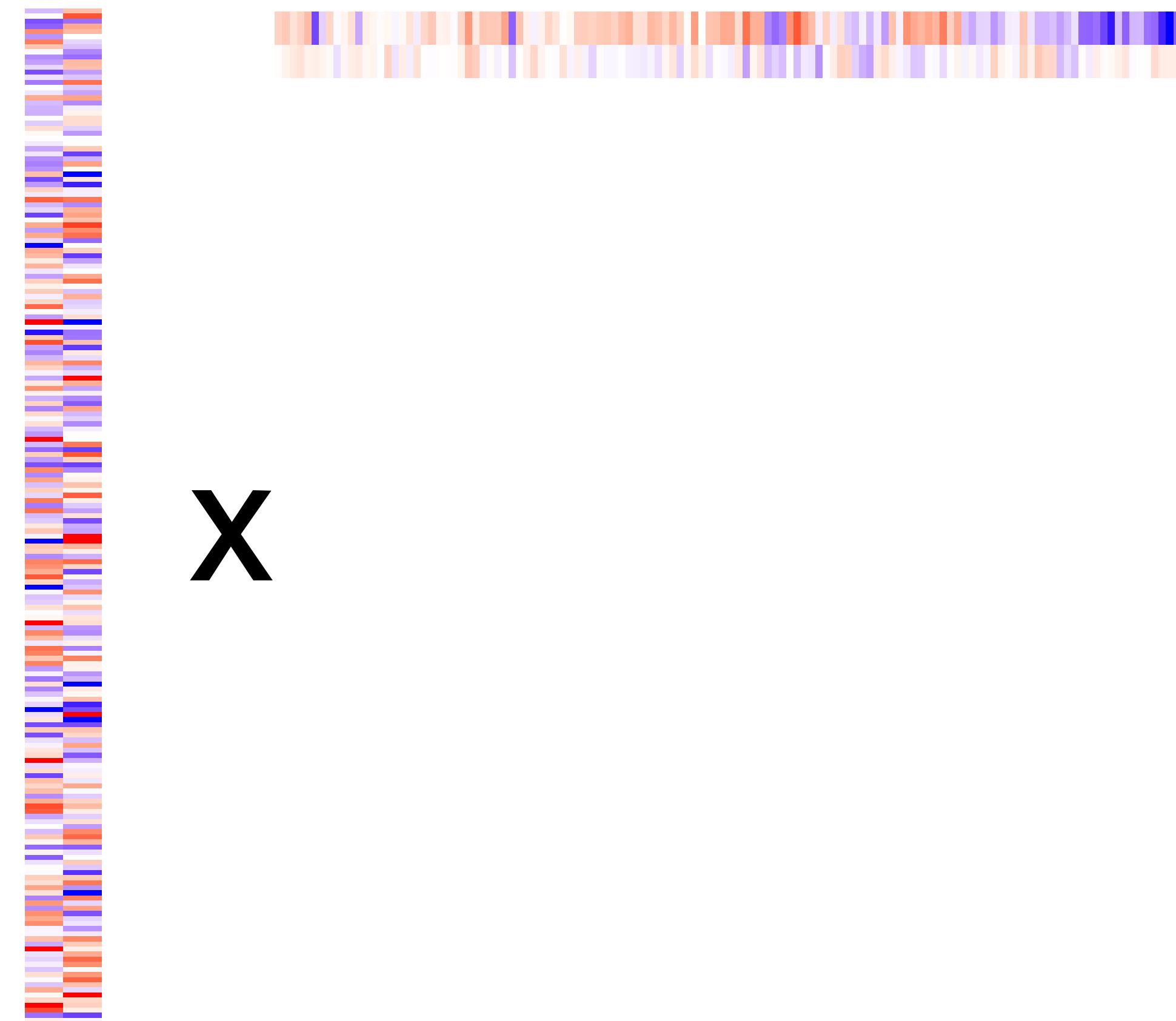
Let's try out random projection

random proj.



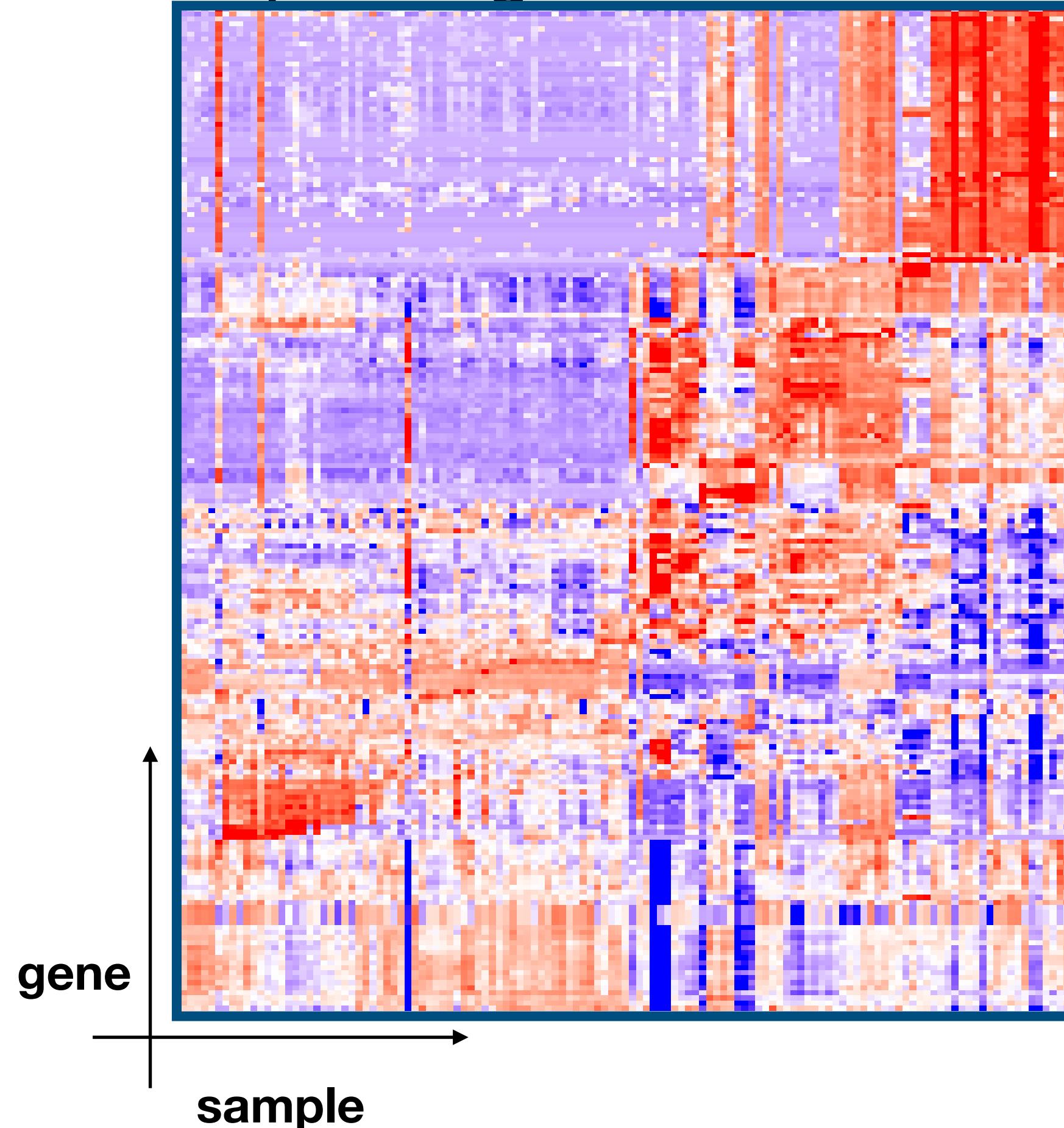
rand how much contribute?

$$= \times$$

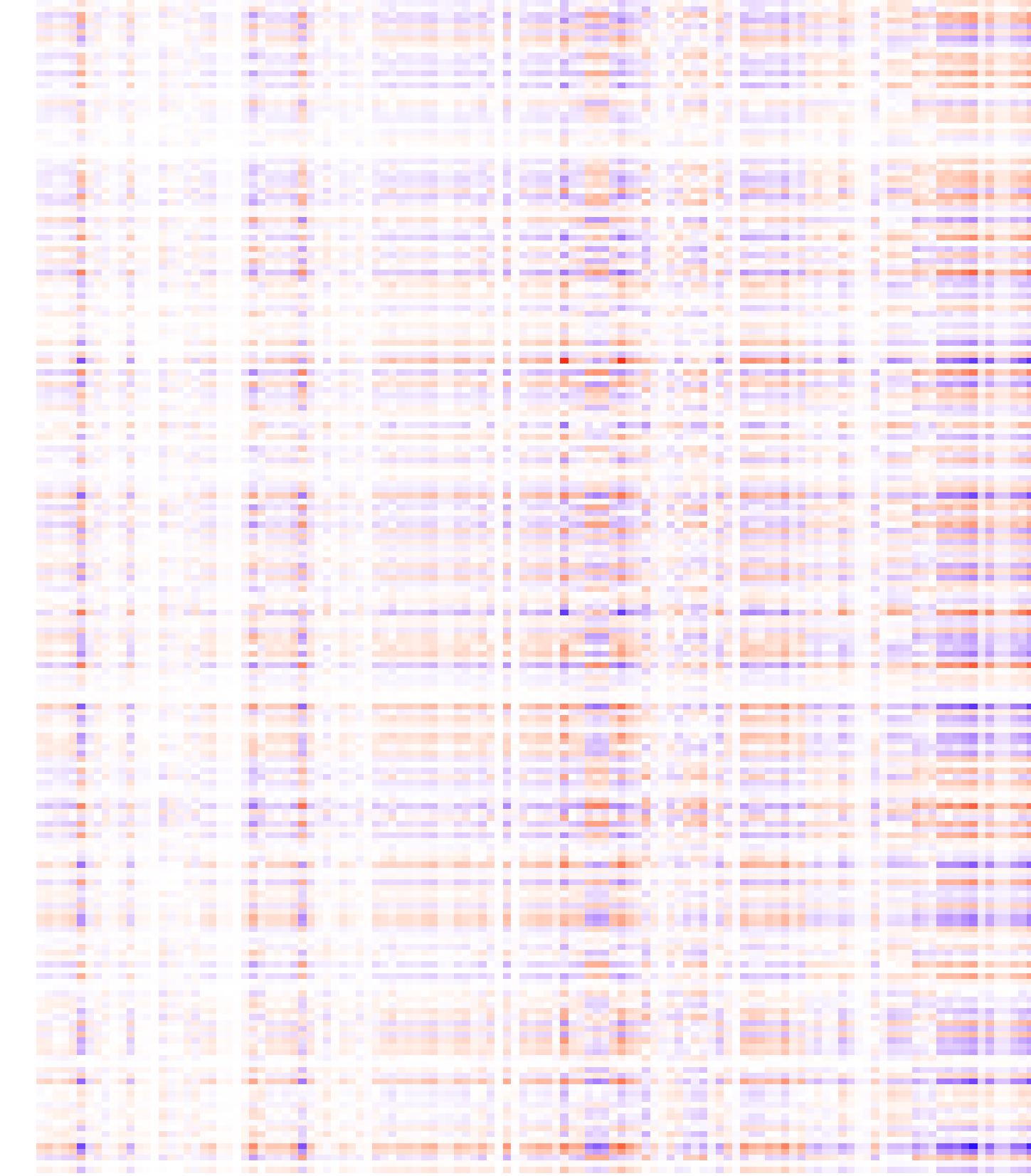


Let's try out random projection

Top 200 genes



1%



ARE YOU PREMATURELY
OPTIMIZING OR JUST TAKING
TIME TO DO THINGS RIGHT?

ARE YOU CONSULTING A
FLOWCHART TO ANSWER
THIS QUESTION?

YES

YOU ARE
PREMATURELY
OPTIMIZING

Can we find a set of “good” vectors to maximize the explained variability?

Recap: Sample covariance matrix

- ▶ Sample mean: $\bar{X}_i = \sum_{j=1}^m X_{ji}/m$
- ▶ Sample variance: $\sum_{j=1}^m (X_{ji} - \bar{X}_i)^2/(m - 1)$
- ▶ Sample covariance between i and k :

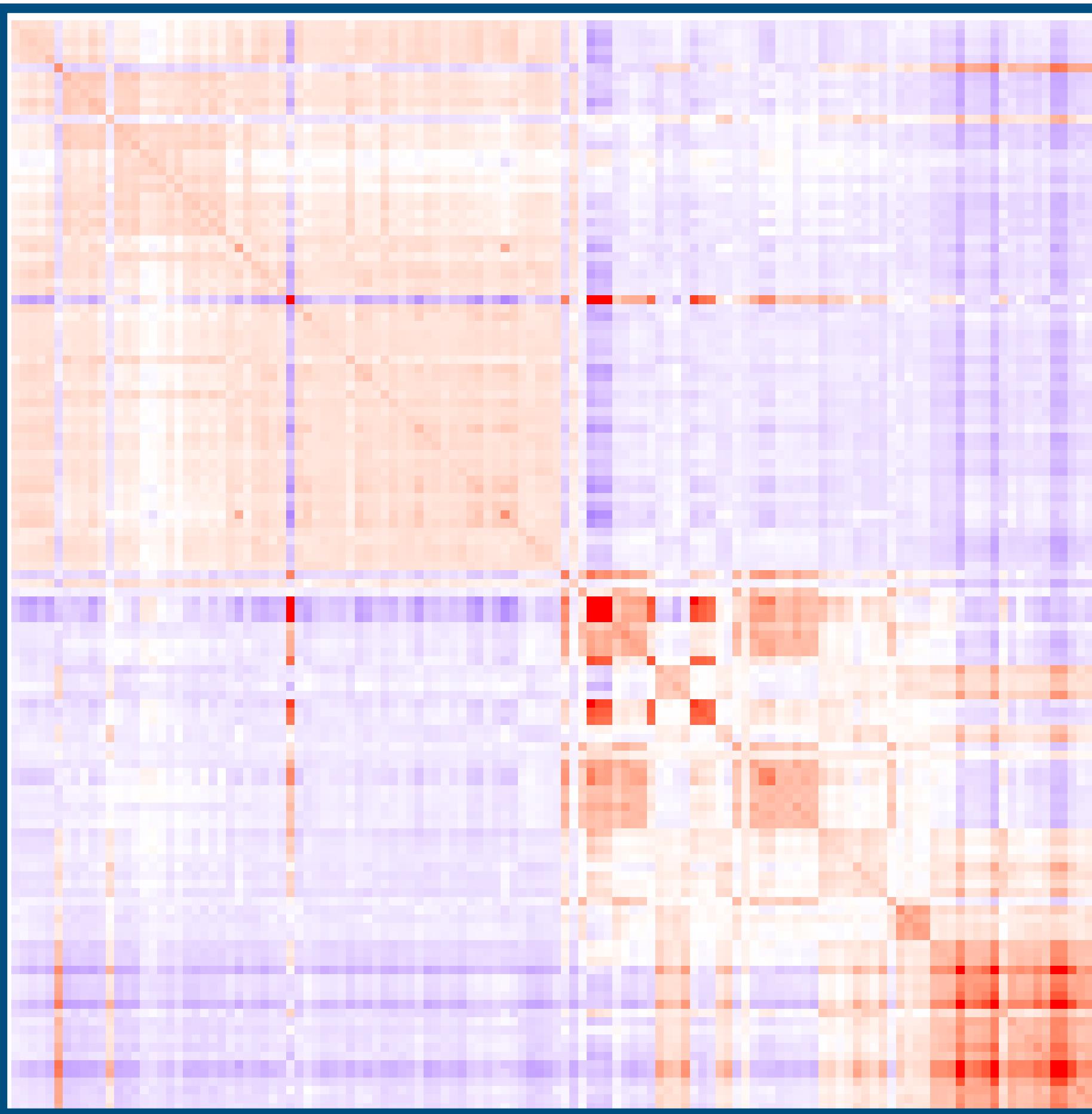
$$\frac{1}{m-1} \sum_{j=1}^m (X_{ji} - \bar{X}_i)(X_{jk} - \bar{X}_k)$$

If all the column vectors \mathbf{x}_i are standardized, the column-by-column covariance $\mathbf{X}^\top \mathbf{X}/(m - 1)$.

If all the row vectors \mathbf{x}_d are standardized, the row-by-row covariance $\mathbf{X}\mathbf{X}^\top/(n - 1)$.

Sample covariance matrix

$$\hat{\Sigma} = X^\top X / (m - 1)$$



Theory: Total variance of the projected data

Total variance

Given the projected, $\hat{X} = \mathbf{u}_1 \cdot (W_{11}, \dots, W_{1n})$, our goal is

$$\max \mathbb{V}[\hat{X}]$$

- ▶ We have two unknown variables U and W
- ▶ There are an infinite number of solutions.

Theory: Total variance of the projected data

Constrained total variance

Given the projected, $\hat{X} = \mathbf{u}_1 \mathbf{w}$, and a unit vector, namely $\|\mathbf{u}_1\| = 1$, our goal is equivalent to

$$\max \mathbf{u}^\top X X^\top \mathbf{u}$$

Because each \hat{W}_i is the solution to the least-square problem:

$$\hat{W}_i = \arg \min \|\mathbf{x}_i - \mathbf{u} W_i\|$$

by solving the least square:

$$\hat{W}_i = \mathbf{x}_i^\top \mathbf{u} / \mathbf{u}^\top \mathbf{u}, \forall i$$

Theory: Why is Principal Component Analysis an eigen value problem?

PCA

Letting the feature-by-feature sample covariance matrix $\hat{\Sigma} = \mathbf{X}\mathbf{X}^\top/(n - 1)$, we want to find a unit vector \mathbf{u} by

$$\max \mathbf{u}^\top \hat{\Sigma} \mathbf{u}$$

subject to $\mathbf{u}^\top \mathbf{u} = 1$.

Theory: Why is Principal Component Analysis an eigen value problem?

Eigen value problem

Given the covariance matrix $\hat{\Sigma}$, we can resolve an eigen-value λ and the corresponding eigen-vector \mathbf{u} such that

$$\hat{\Sigma}\mathbf{u} = \lambda\mathbf{u}$$

(“eigen” mean “own” in German).

Theory: Why is Principal Component Analysis an eigen value problem?

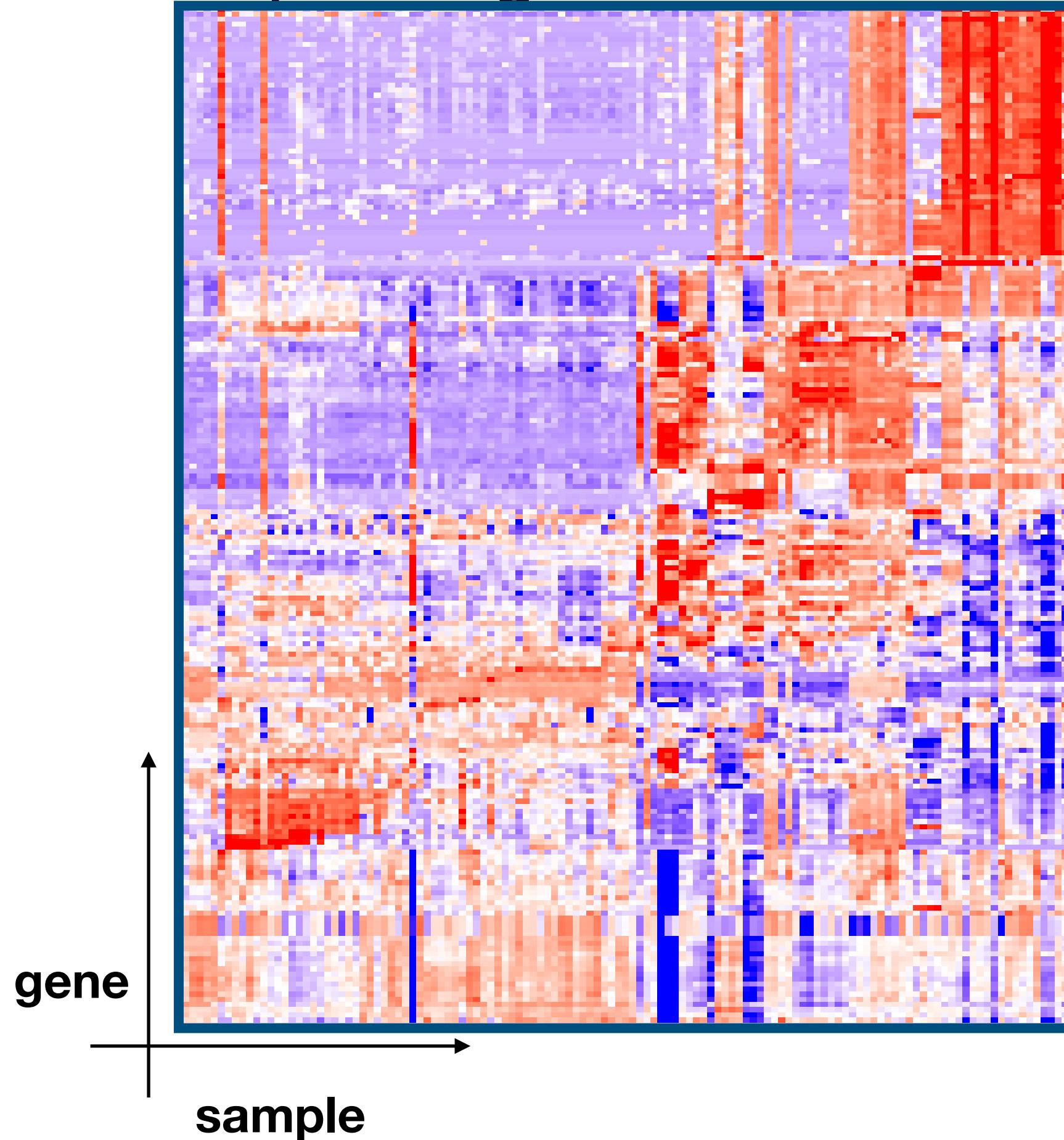
Why are they equivalent? Solving the PCA problem is

$$\iff \max_{\mathbf{u}} \underbrace{\mathbf{u}^\top \hat{\Sigma} \mathbf{u}}_{\text{total variation}} + \underbrace{\lambda (1 - \mathbf{u}^\top \mathbf{u})}_{\text{constraint}}, \lambda > 0 \quad \text{a.k.a. Lagrangian}$$

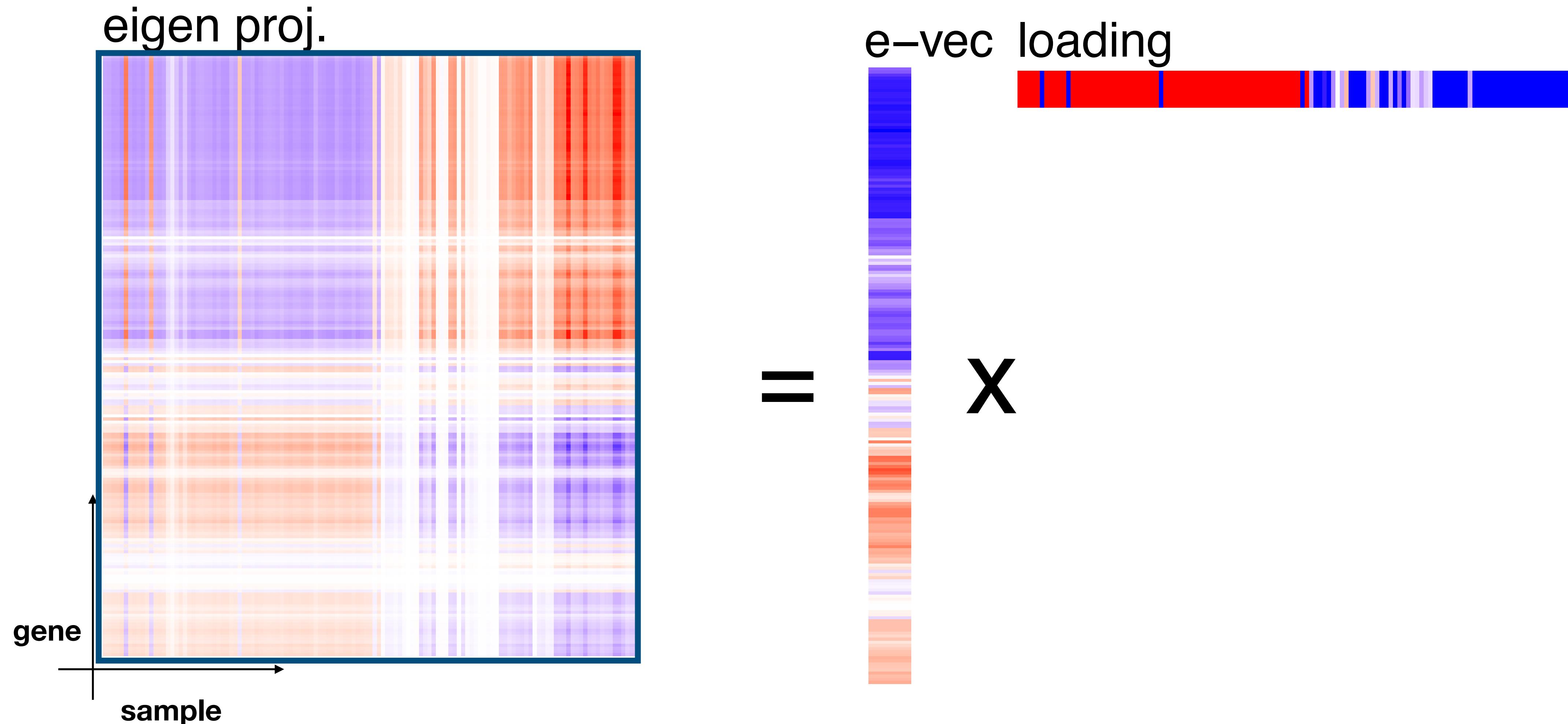
Taking the derivative with respect to \mathbf{u} and setting it to zero, we get the eigen-value problem.

Let's see how much the first eigenvector can explain

Top 200 genes

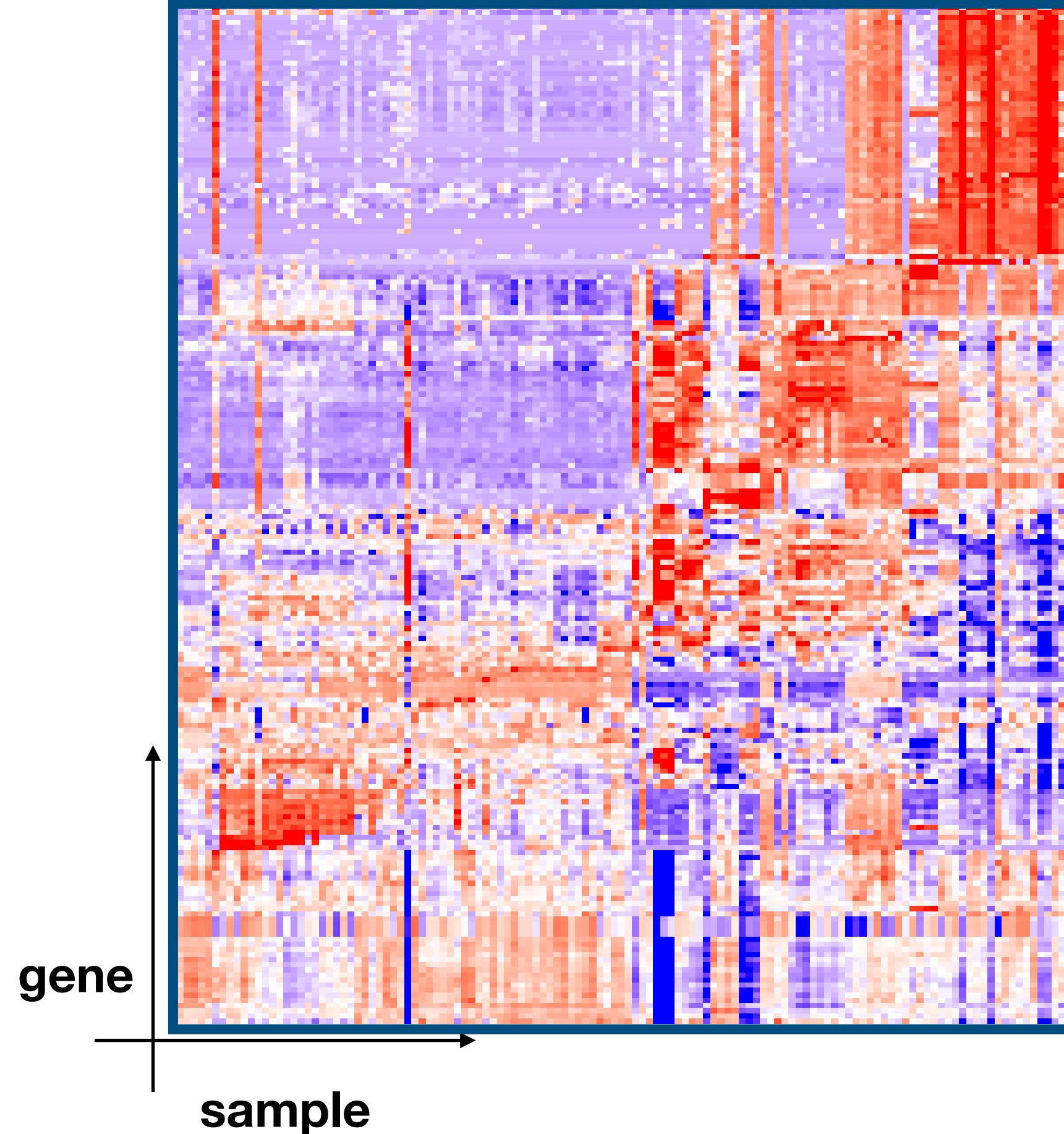


Let's see how much the first eigenvector can explain

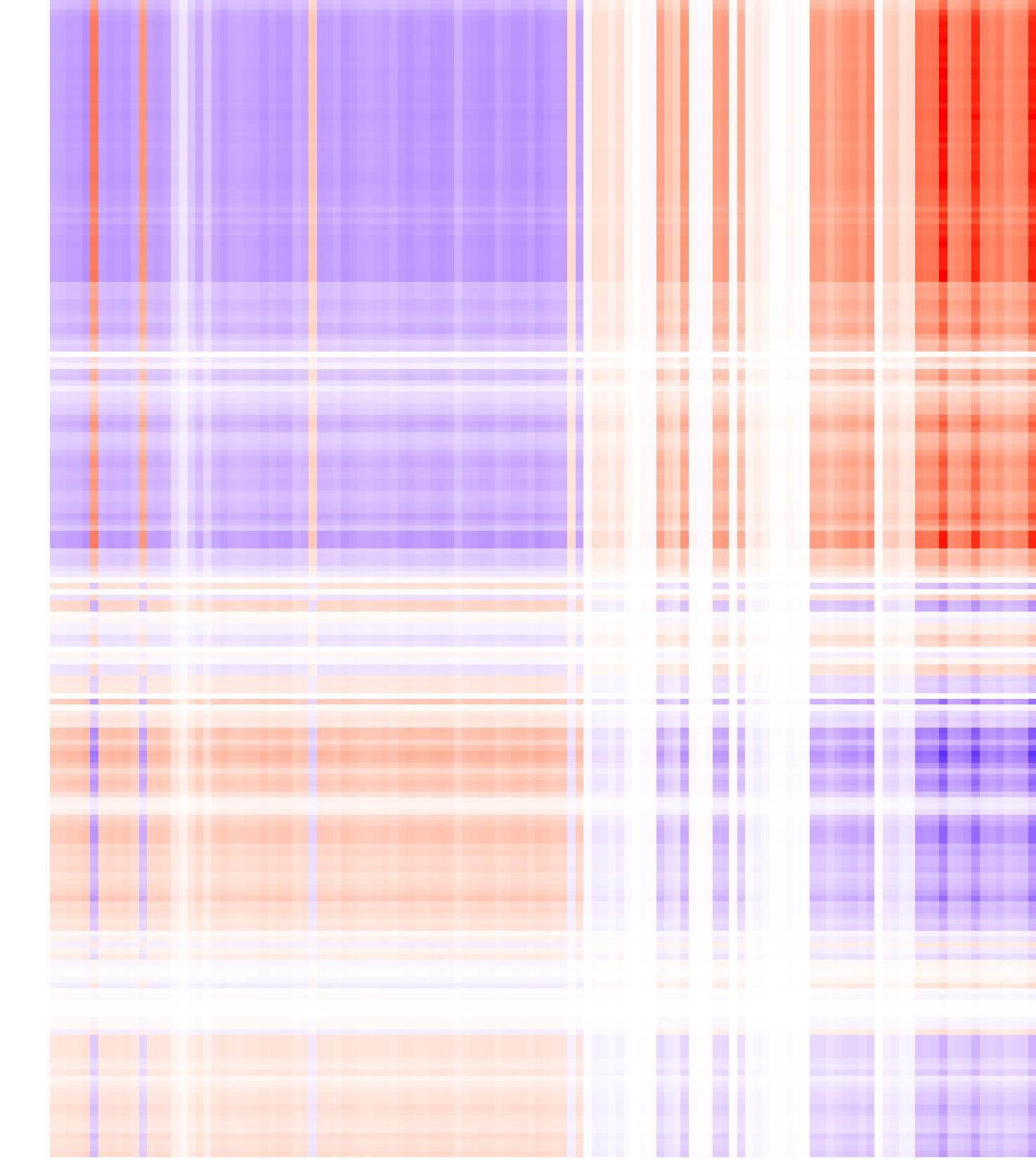


Let's see how much the first eigenvector can explain

Top 200 genes

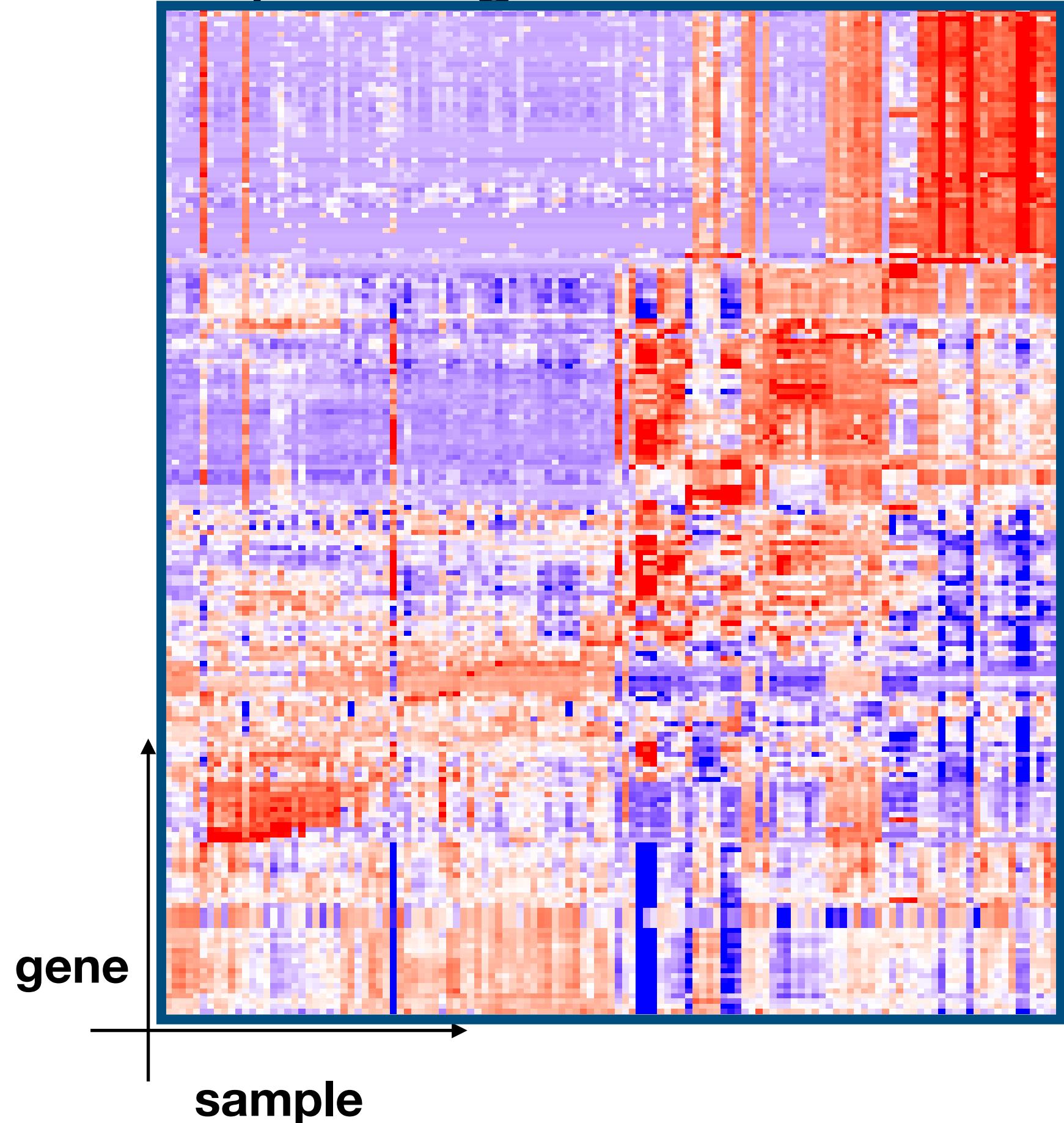


42%

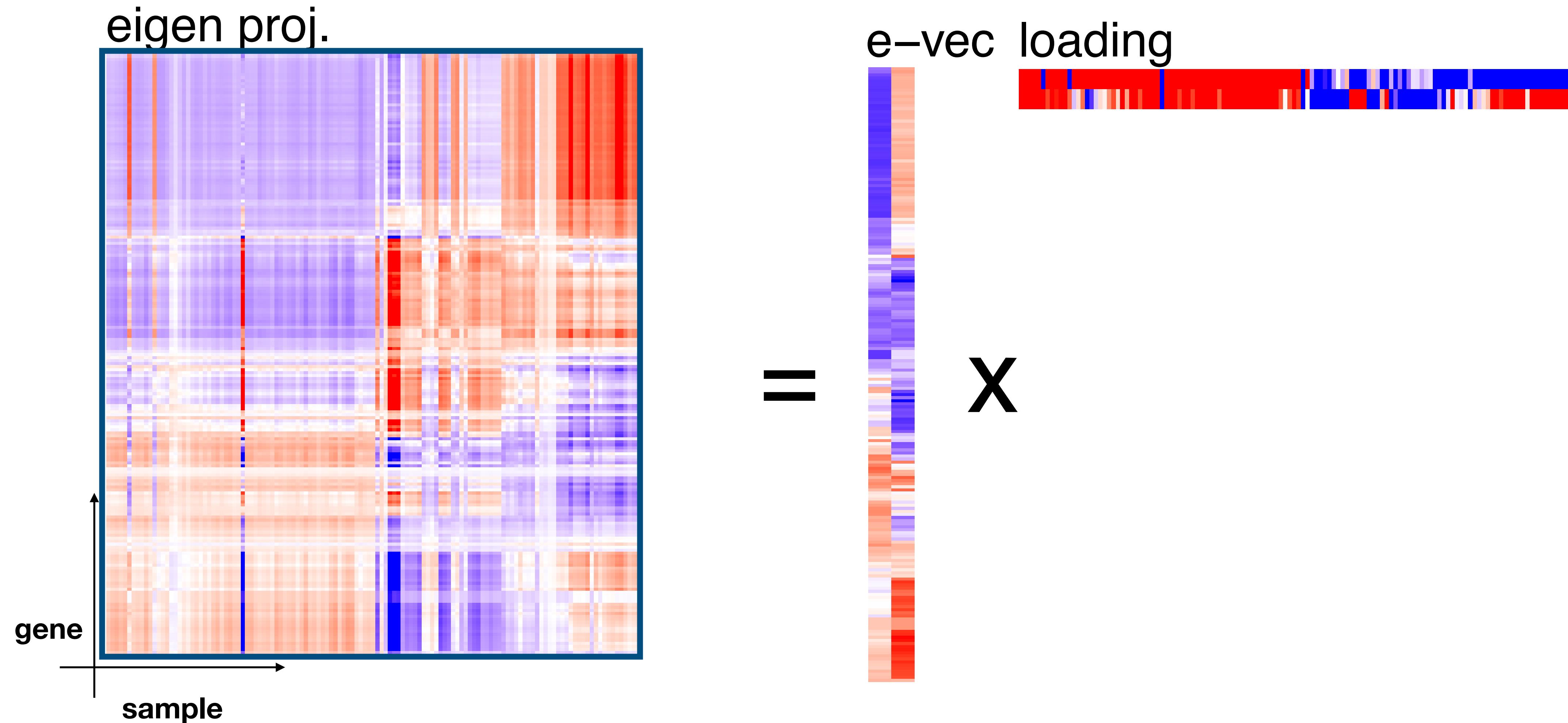


Top two eigen-vectors

Top 200 genes

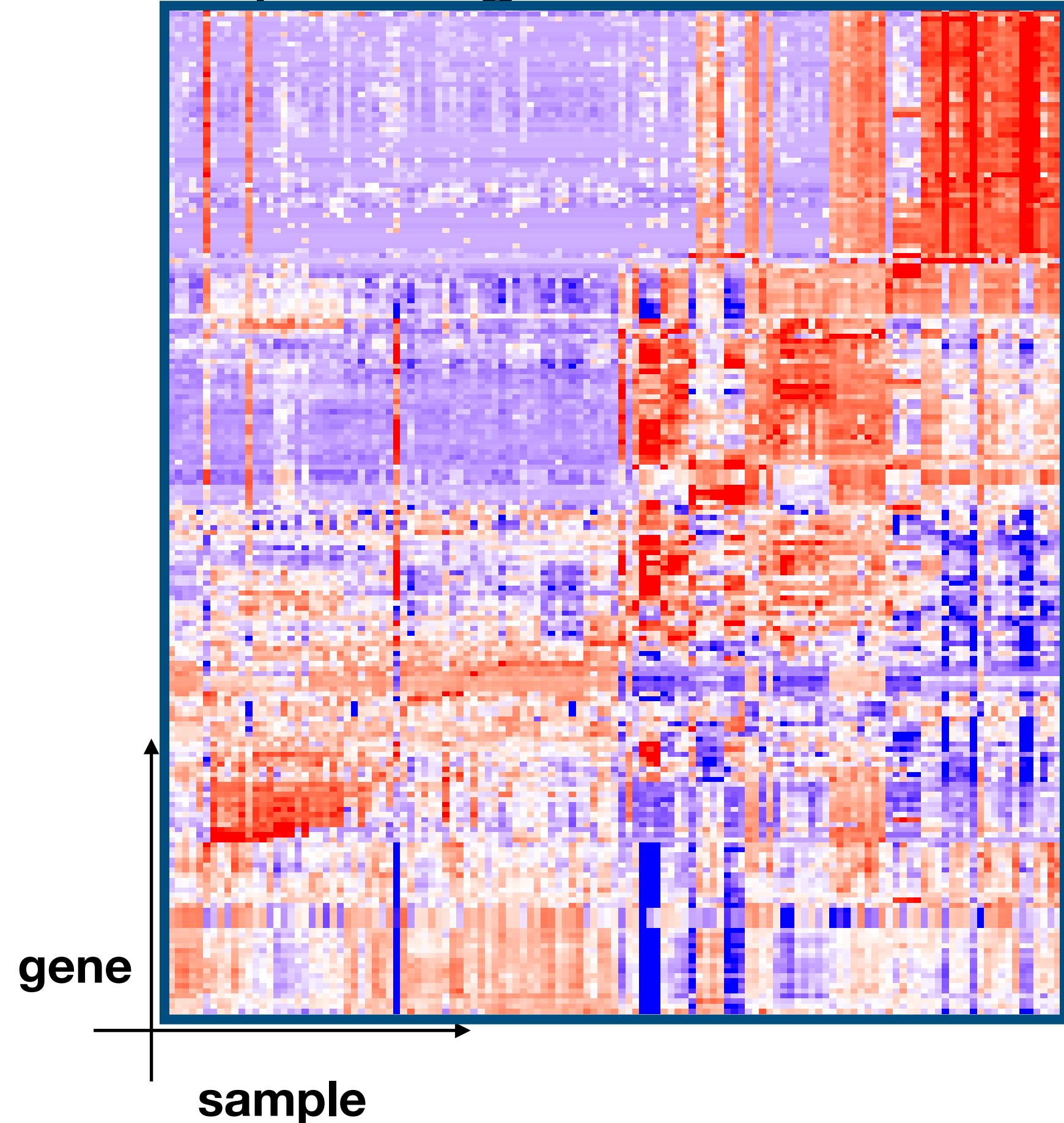


Top two eigen-vectors

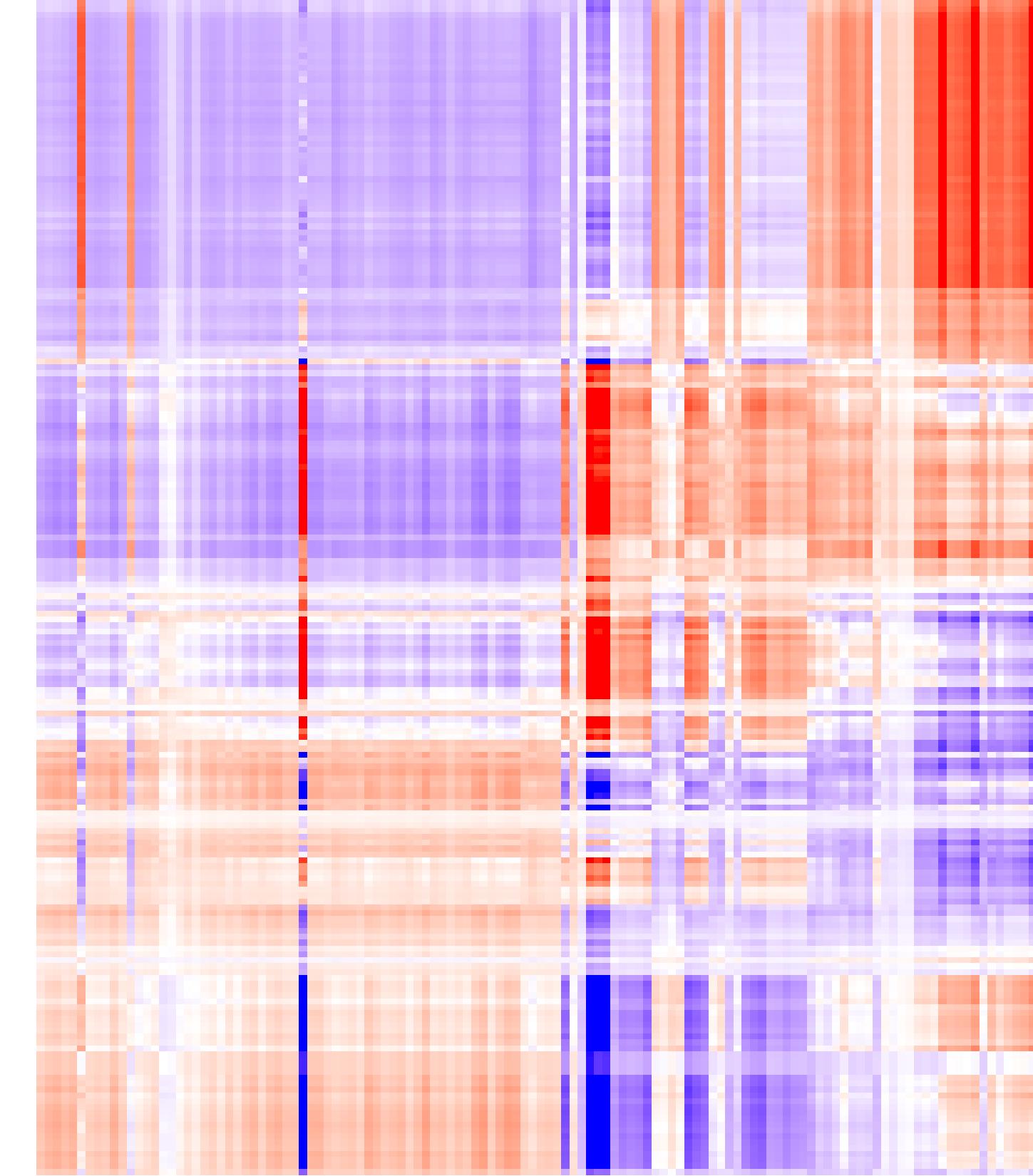


Top two eigen-vectors

Top 200 genes

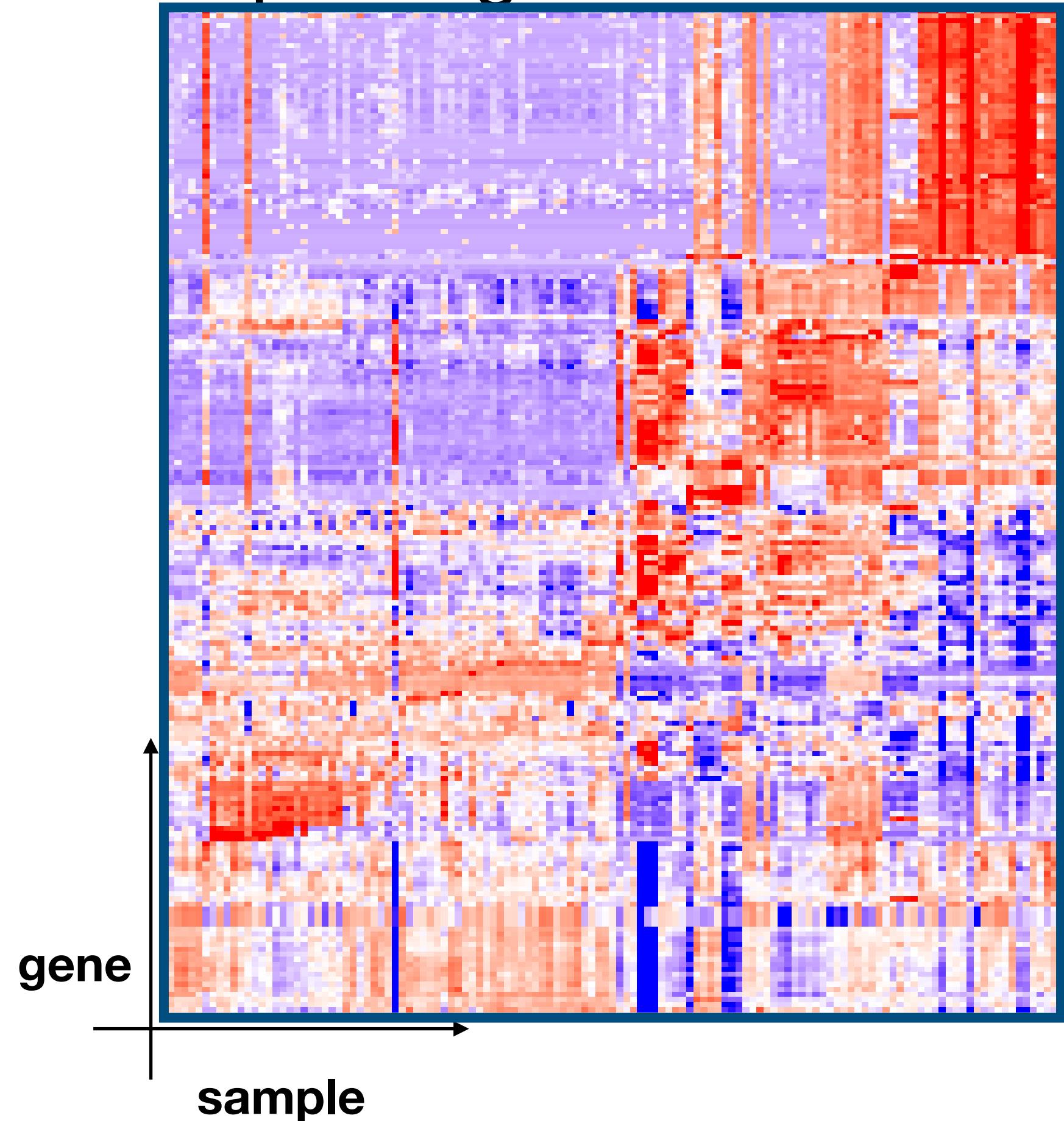


58%

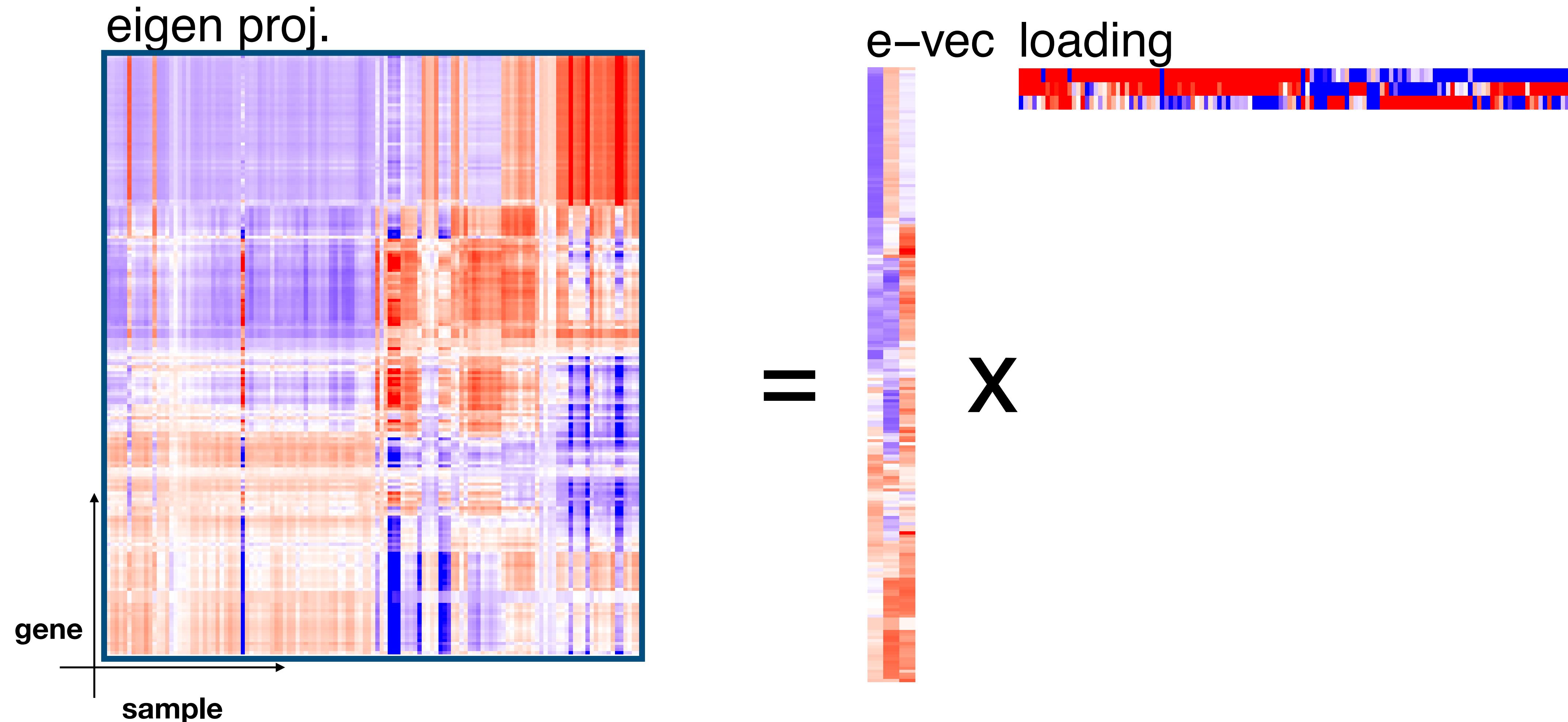


Top three eigen-vectors

Top 200 genes

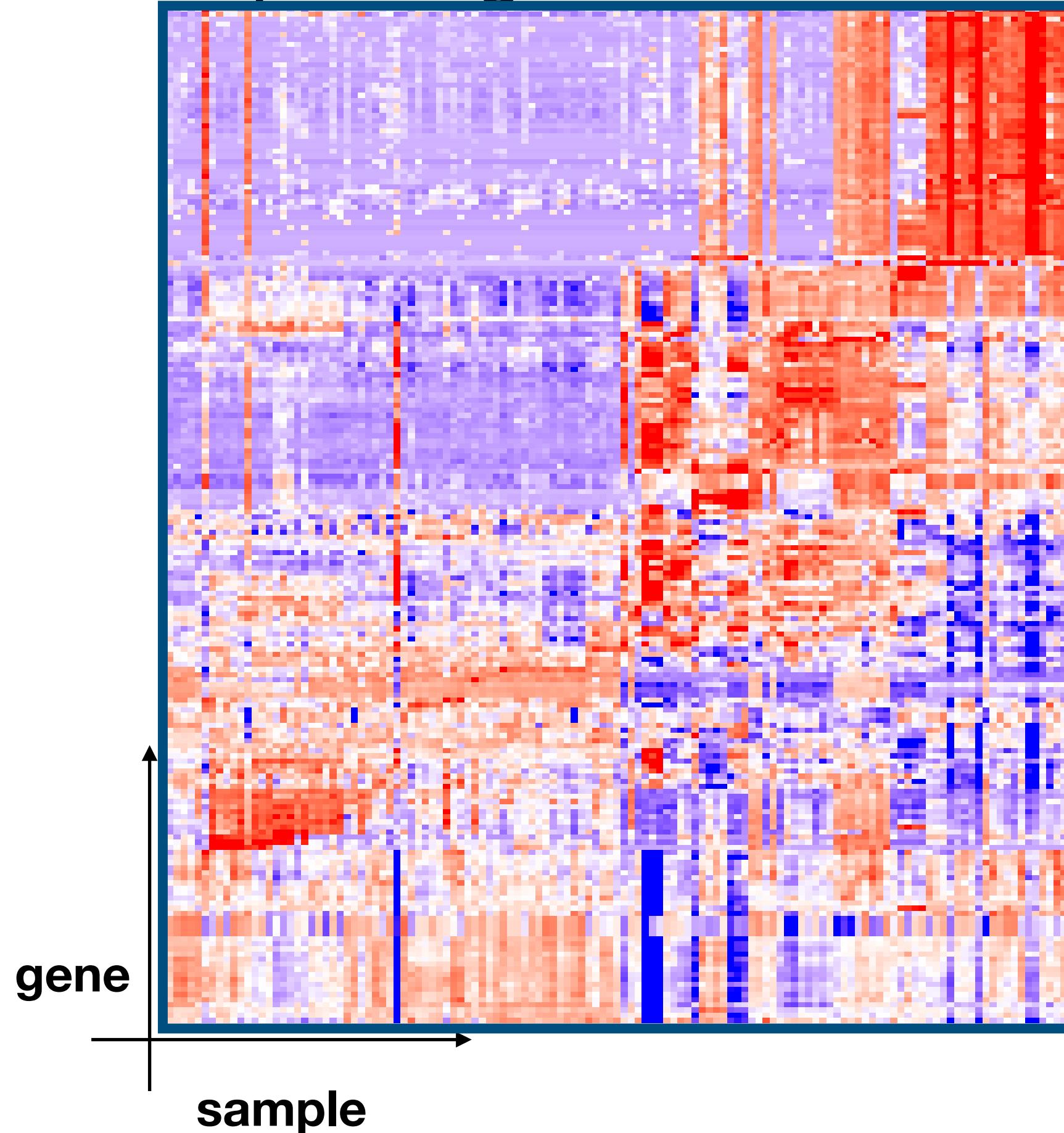


Top three eigen-vectors

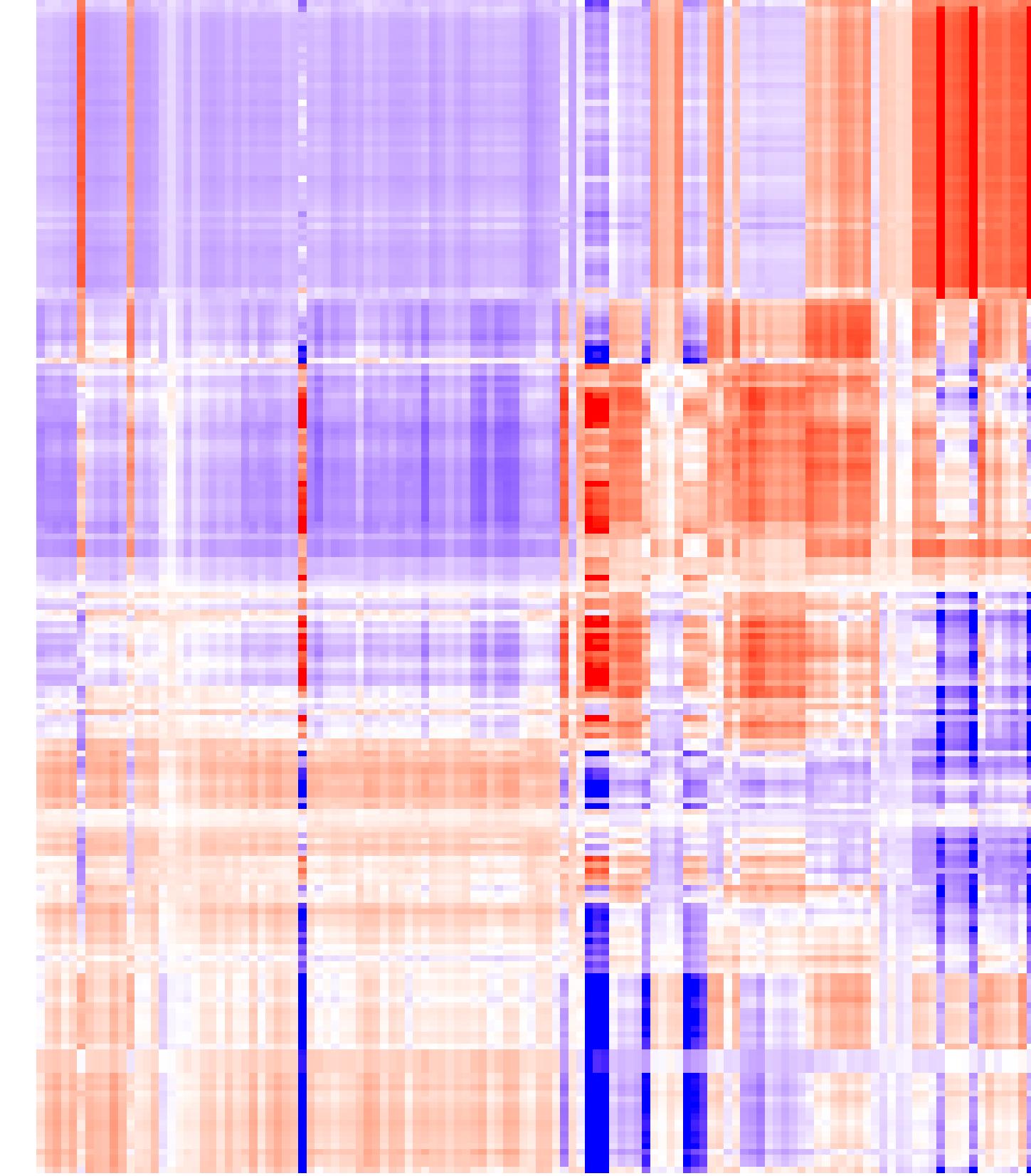


Top three eigen-vectors

Top 200 genes



59%



Singular Value Decomposition can find a PCA solution

Singular Value Decomposition

SVD identifies three matrices of X :

$$X = UDV^\top$$

where both U and V vectors are orthonormal, namely,

- ▶ $U^\top U = I$, $\mathbf{u}_k^\top \mathbf{u}_k = 1$ for all k ,
- ▶ $V^\top V = I$, $\mathbf{v}_k^\top \mathbf{v}_k = 1$ for all k .

SVD: another equivalent method for PCA

Singular Value Decomposition

SVD identifies three matrices of X :

$$X = UDV^\top$$

where both U and V vectors are orthonormal, namely,

- ▶ $U^\top U = I$, $\mathbf{u}_k^\top \mathbf{u}_k = 1$ for all k ,
- ▶ $V^\top V = I$, $\mathbf{v}_k^\top \mathbf{v}_k = 1$ for all k .

Covariance by SVD

Covariance across the columns (samples)

$$X^\top X / (m - 1) = VD^2V^\top / (m - 1)$$

Covariance across the rows (genes)

$$XX^\top / (n - 1) = UD^2U^\top / (n - 1)$$

Remark: standardized matrix

How can SVD find an equivalent solution for PCA?

$$\underbrace{\left(\frac{1}{m-1} X^\top X \right)}_{\text{sample covariance}} \mathbf{v}_1 = \frac{1}{m-1} (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k) \begin{pmatrix} D_1^2 & 0 & \dots & \dots \\ 0 & D_2^2 & 0 & \dots \\ 0 & \dots & \ddots & 0 \\ 0 & \dots & 0 & D_k^2 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \\ \vdots \\ \mathbf{v}_k^\top \end{pmatrix} \mathbf{v}_1$$

How can SVD find an equivalent solution for PCA?

$$\underbrace{\left(\frac{1}{m-1} X^\top X \right)}_{\text{sample covariance}} \mathbf{v}_1 = \frac{1}{m-1} (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k) \begin{pmatrix} D_1^2 & 0 & \dots & \dots \\ 0 & D_2^2 & 0 & \dots \\ 0 & \dots & \ddots & 0 \\ 0 & \dots & 0 & D_k^2 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \\ \vdots \\ \mathbf{v}_k^\top \end{pmatrix} \mathbf{v}_1$$
$$= \frac{1}{m-1} (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k) \begin{pmatrix} D_1^2 & 0 & \dots & \dots \\ 0 & D_2^2 & 0 & \dots \\ 0 & \dots & \ddots & 0 \\ 0 & \dots & 0 & D_k^2 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

How can SVD find an equivalent solution for PCA?

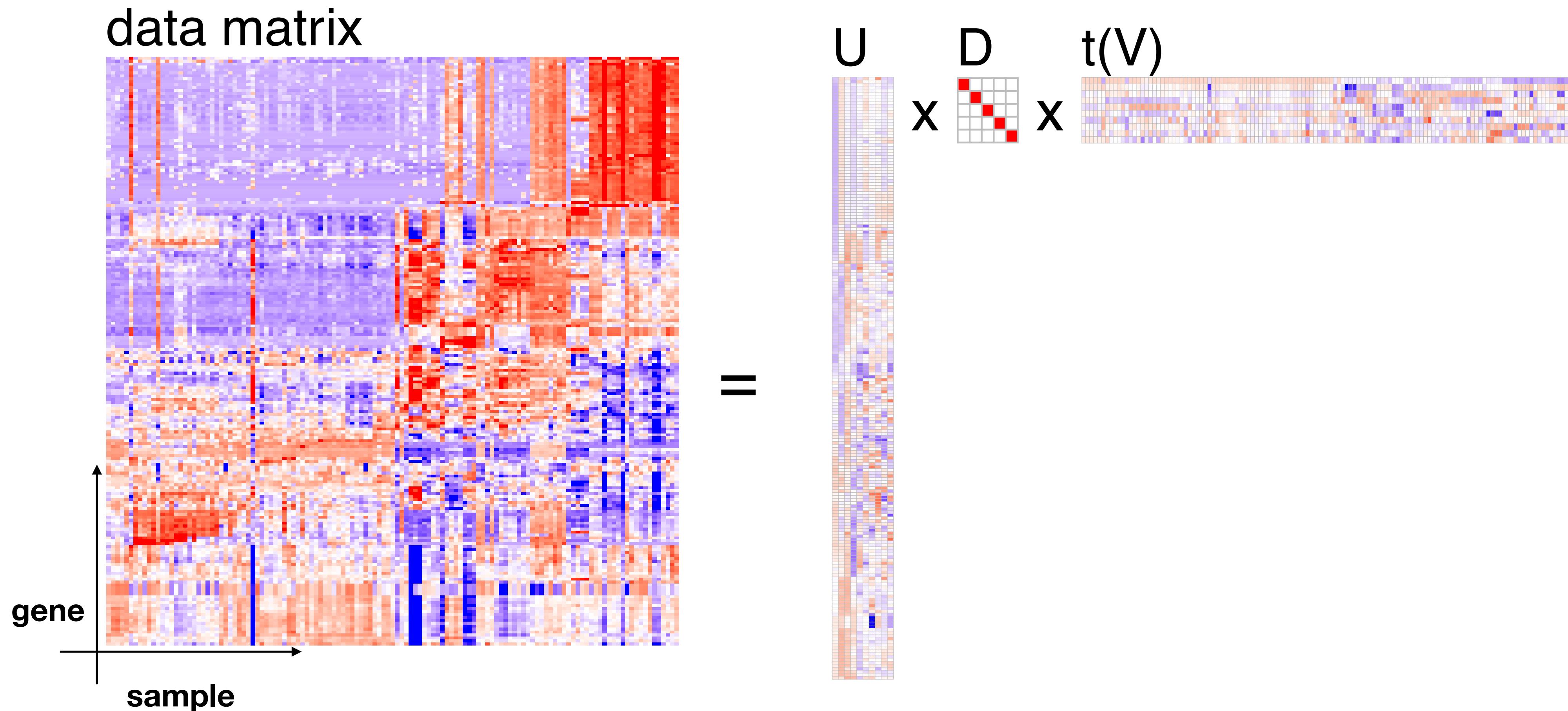
$$\begin{aligned} \underbrace{\left(\frac{1}{m-1} X^\top X \right)}_{\text{sample covariance}} \mathbf{v}_1 &= \frac{1}{m-1} (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k) \begin{pmatrix} D_1^2 & 0 & \dots & \dots \\ 0 & D_2^2 & 0 & \dots \\ 0 & \dots & \ddots & 0 \\ 0 & \dots & 0 & D_k^2 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \\ \vdots \\ \mathbf{v}_k^\top \end{pmatrix} \mathbf{v}_1 \\ &= \frac{1}{m-1} (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k) \begin{pmatrix} D_1^2 & 0 & \dots & \dots \\ 0 & D_2^2 & 0 & \dots \\ 0 & \dots & \ddots & 0 \\ 0 & \dots & 0 & D_k^2 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\ &= \frac{1}{m-1} (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k) \begin{pmatrix} D_1^2 \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{aligned}$$

How can SVD find an equivalent solution for PCA?

$$\begin{aligned}
 \underbrace{\left(\frac{1}{m-1} X^\top X \right)}_{\text{sample covariance}} \mathbf{v}_1 &= \frac{1}{m-1} (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k) \begin{pmatrix} D_1^2 & 0 & \dots & \dots \\ 0 & D_2^2 & 0 & \dots \\ 0 & \dots & \ddots & 0 \\ 0 & \dots & 0 & D_k^2 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \\ \vdots \\ \mathbf{v}_k^\top \end{pmatrix} \mathbf{v}_1 \\
 &= \frac{1}{m-1} (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k) \begin{pmatrix} D_1^2 & 0 & \dots & \dots \\ 0 & D_2^2 & 0 & \dots \\ 0 & \dots & \ddots & 0 \\ 0 & \dots & 0 & D_k^2 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\
 \hat{\Sigma} \mathbf{v}_1 &= \frac{1}{m-1} (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k) \begin{pmatrix} D_1^2 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \underbrace{\frac{D_1^2}{m-1}}_{\text{eigenvalue}} \underbrace{\mathbf{v}_1}_{\text{eigenvector}}
 \end{aligned}$$

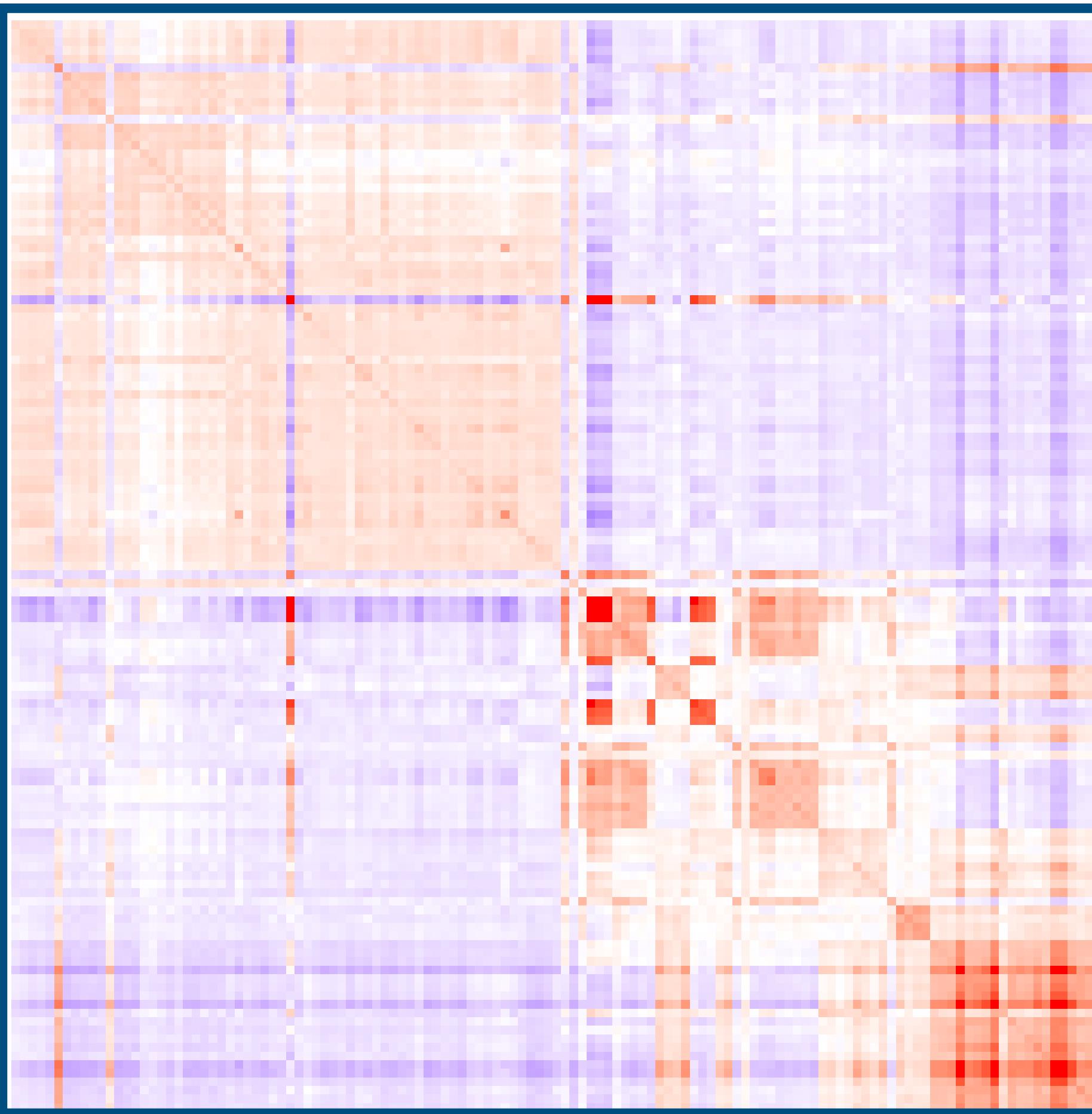
Run SVD to find principal components

```
svd.out <- svd(x.sub, nu = 10, nv = 10)  
U <- svd.out$u; D <- diag(svd.out$d[1:5]); V <- svd.out$v
```



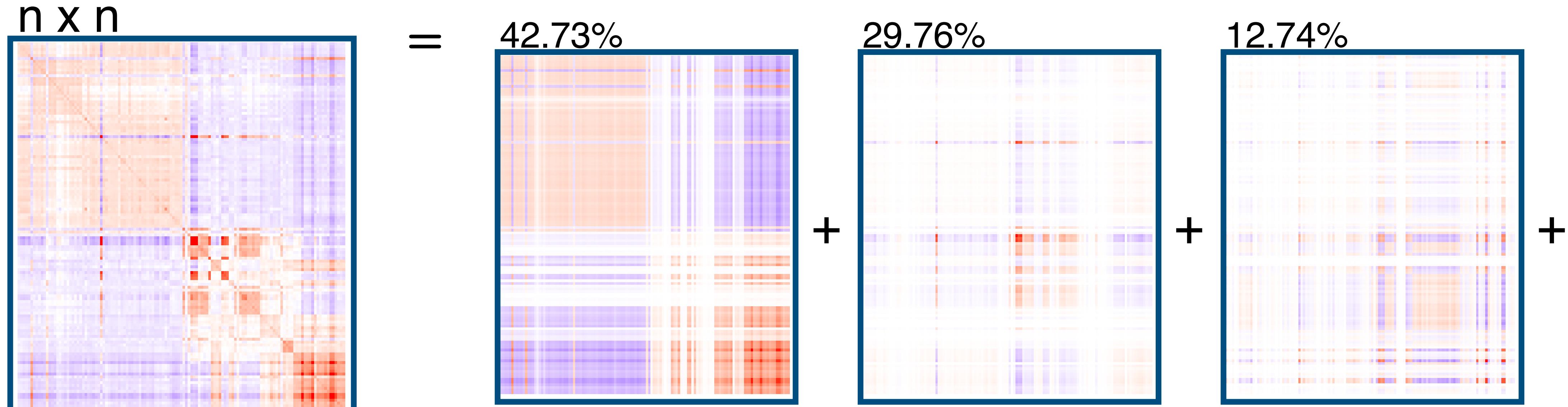
Sample covariance matrix

$$\hat{\Sigma} = X^\top X / (m - 1)$$



Principal components decompose the covariance matrix

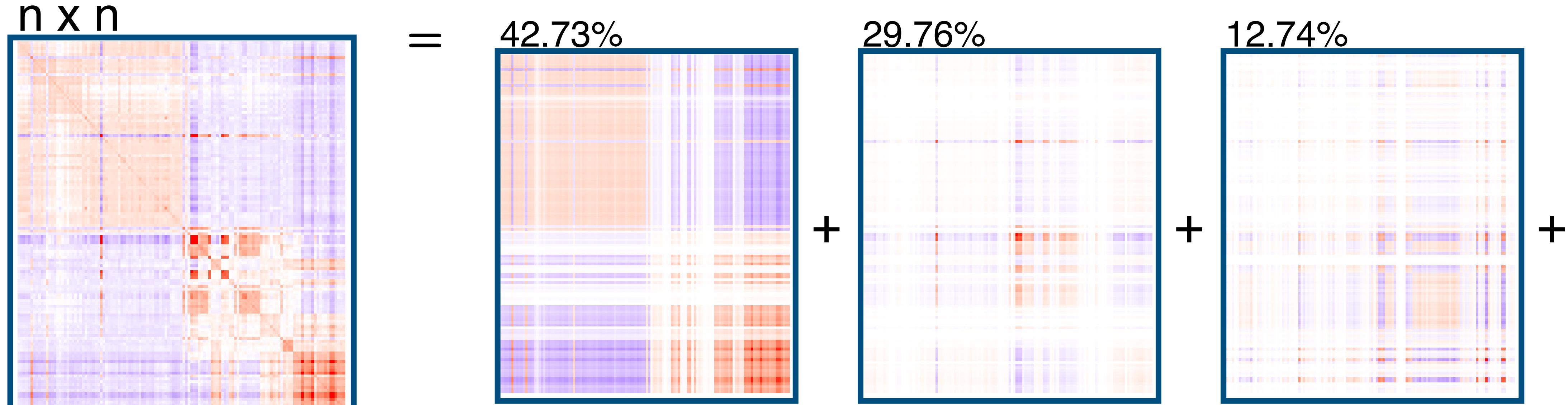
$$\hat{\Sigma} = \frac{X^T X}{m - 1}$$



- ▶ How much variance is explained by each component?

Principal components decompose the covariance matrix

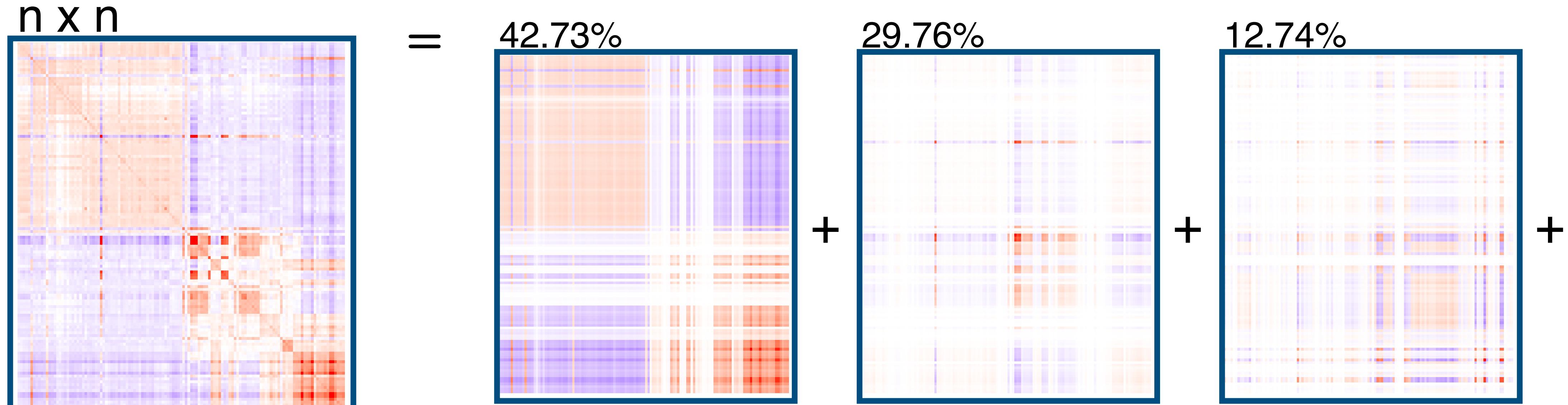
$$\hat{\Sigma} = \frac{X^T X}{m - 1} = V \frac{D^2}{m - 1} V^T$$



- ▶ How much variance is explained by each component?

Principal components decompose the covariance matrix

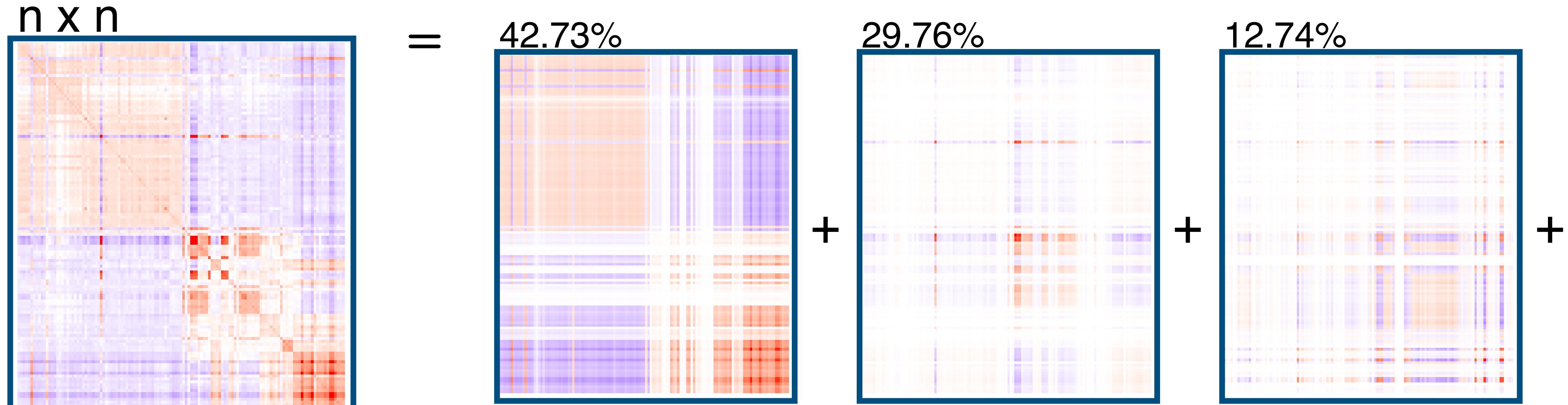
$$\hat{\Sigma} = \frac{X^\top X}{m-1} = V \frac{D^2}{m-1} V^\top = \sum_{k=1} \lambda_k \mathbf{v}_k \mathbf{v}_k^\top, \quad \lambda_k$$



- ▶ How much variance is explained by each component?

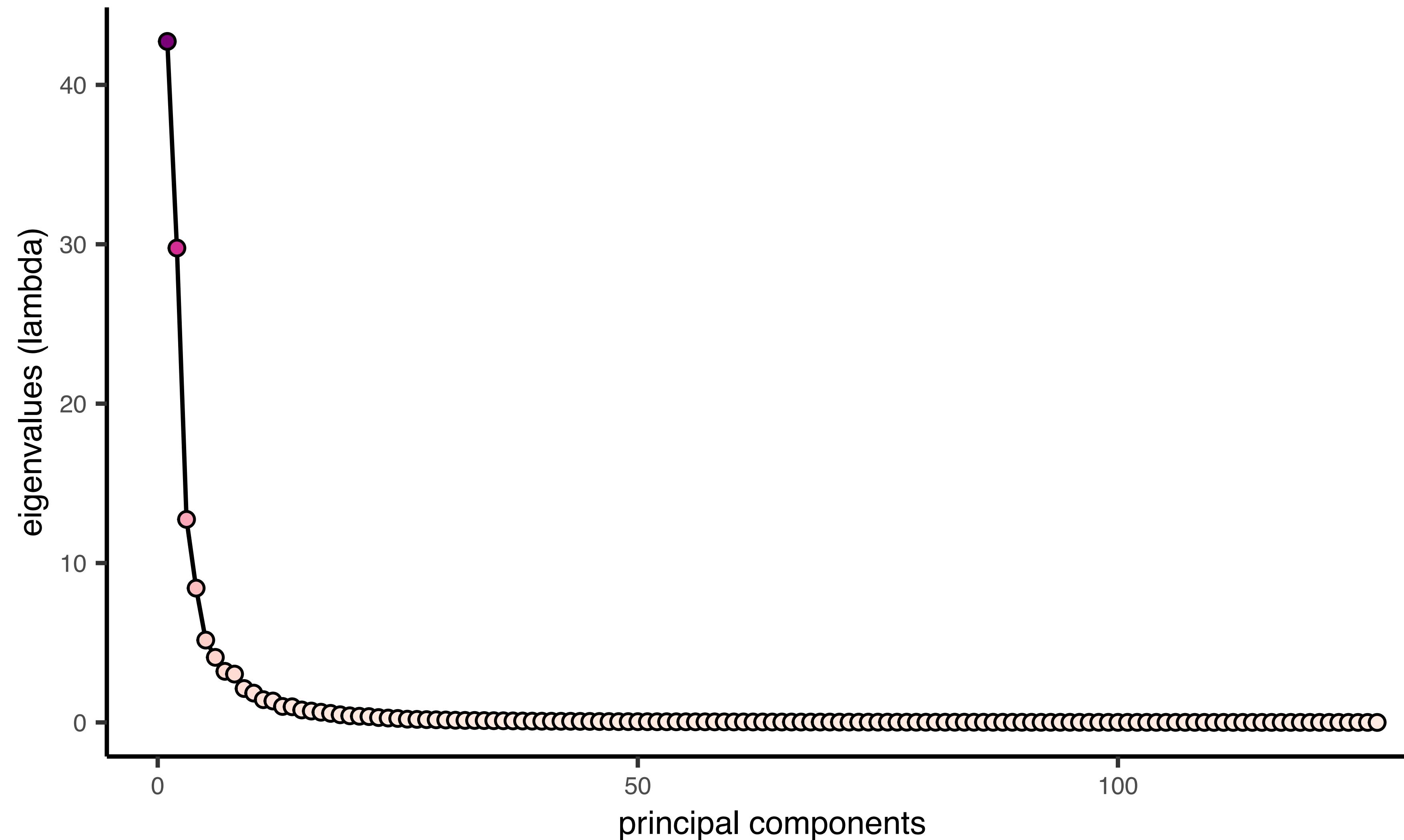
Principal components decompose the covariance matrix

$$\hat{\Sigma} = \frac{X^\top X}{m-1} = V \frac{D^2}{m-1} V^\top = \sum_{k=1} \lambda_k \mathbf{v}_k \mathbf{v}_k^\top, \quad \lambda_k = \frac{D_k^2}{m-1}$$

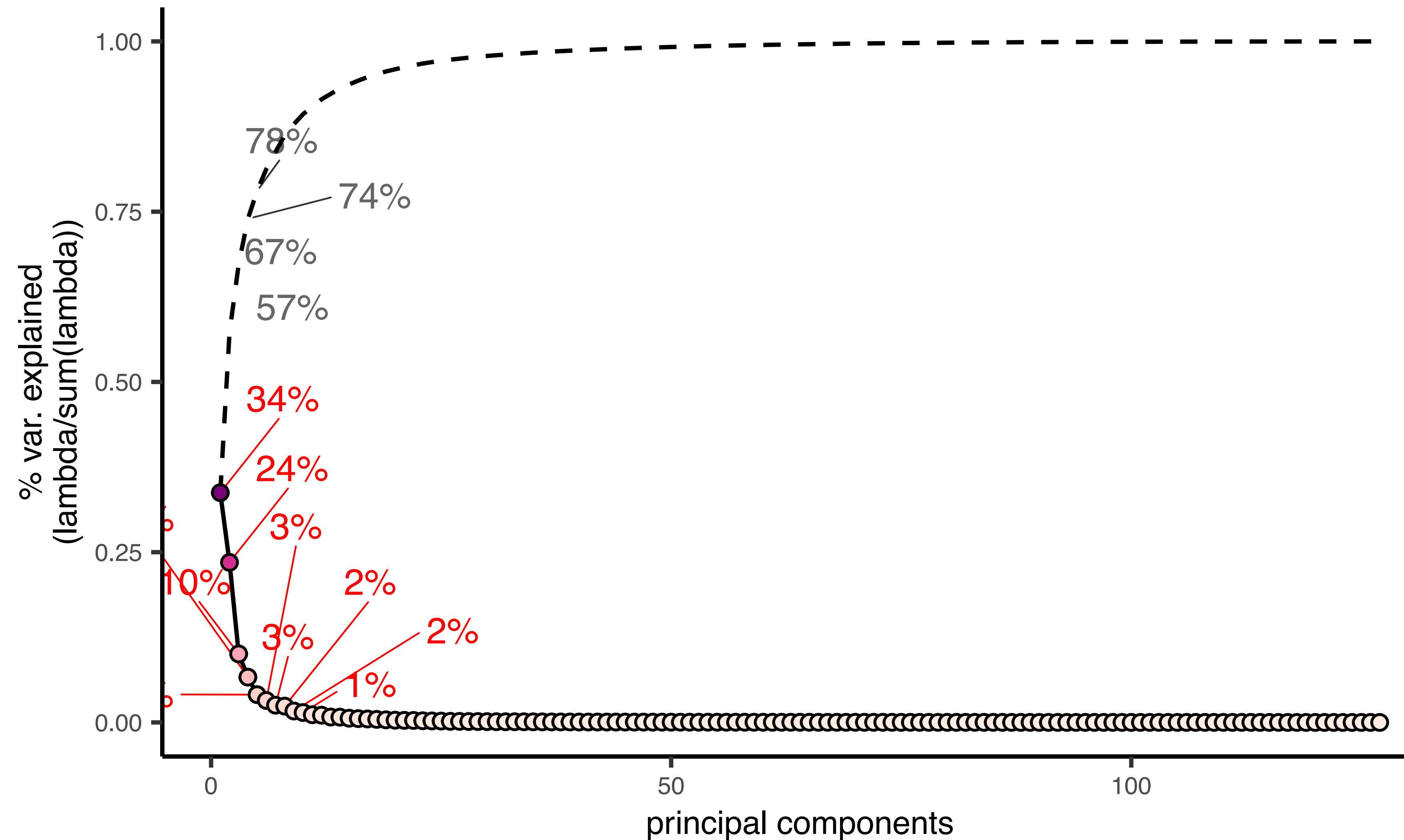


- ▶ How much variance is explained by each component?

How much variance is explained by each principal component?



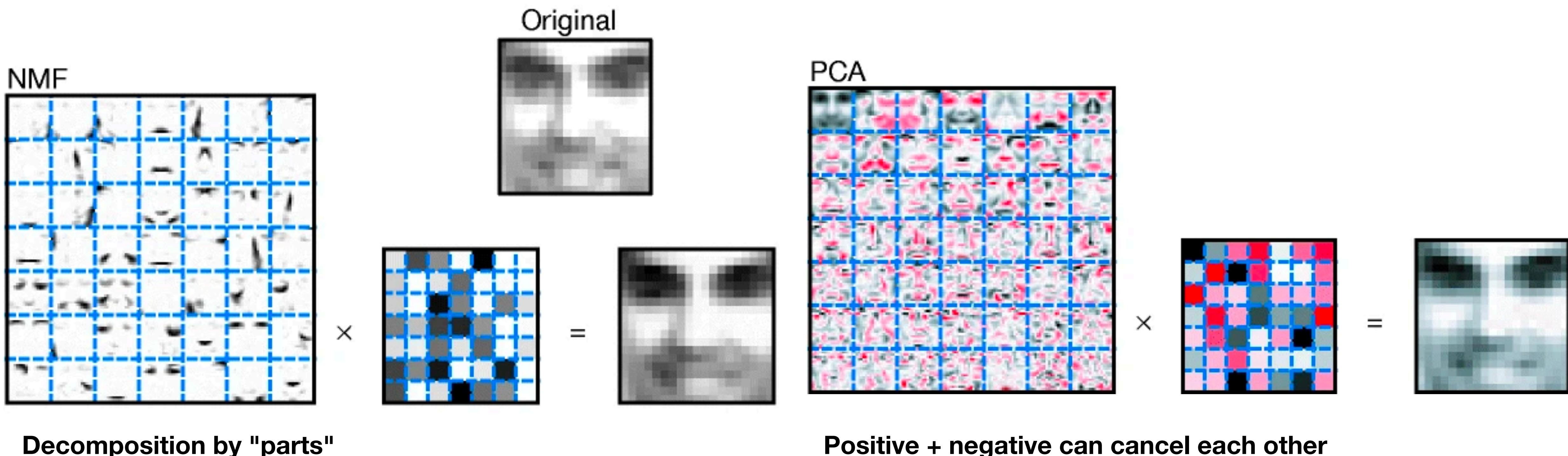
How much variance is explained by each principal component?



Today's lecture: Enrichment Analysis

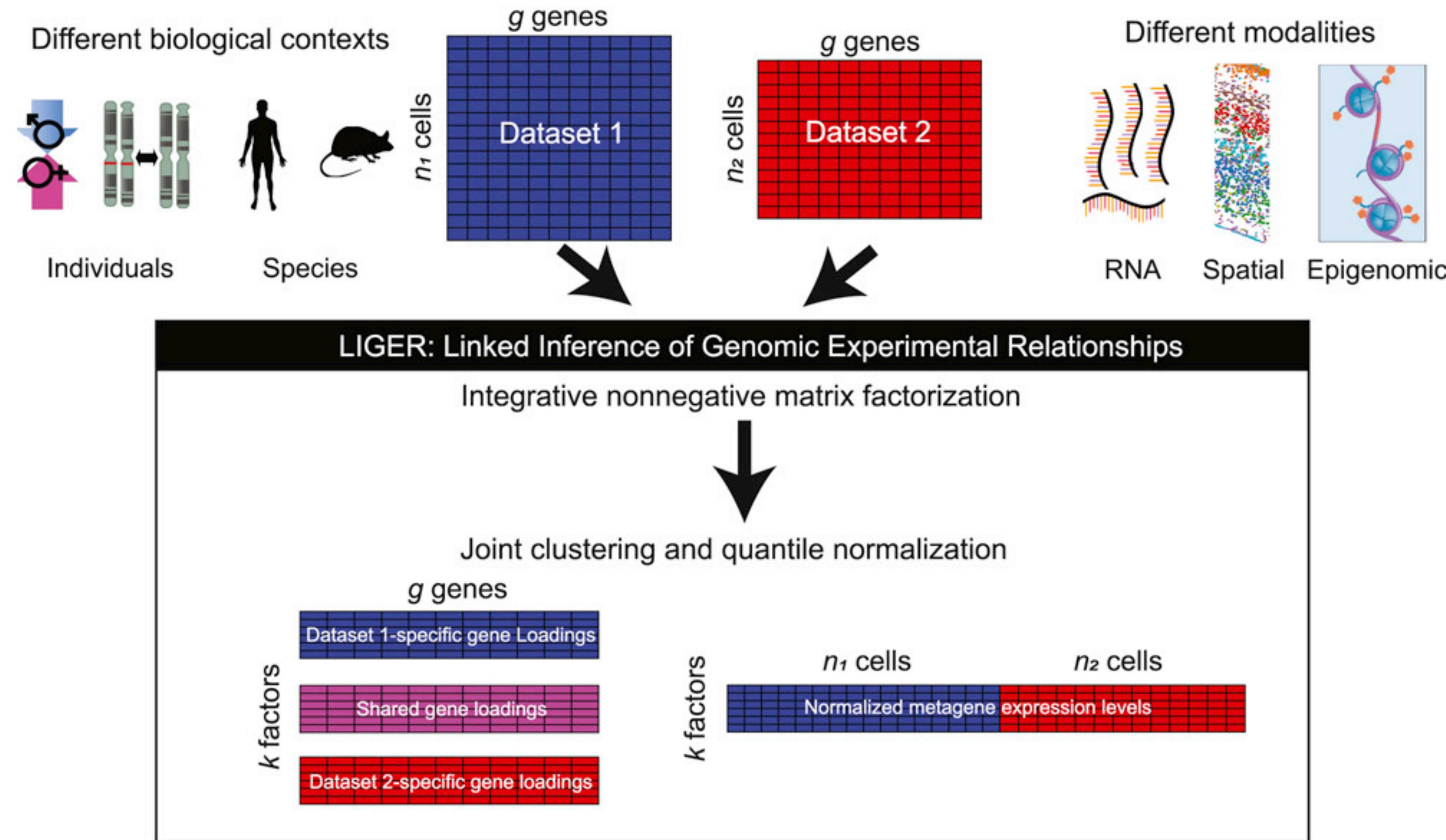
- **Motivations: What's next after genomics analysis?**
 - What have we learned? How do we know that our discovery is meaningful?
- **Gene set enrichment analysis**
 - Set-based approach: Hypergeometric test
 - Rank-based approach: GSEA by KS statistic
- **Can we engineer new gene sets/scores?**
 - Principal Component Analysis
 - Matrix factorization of count data

Why another factorization method?



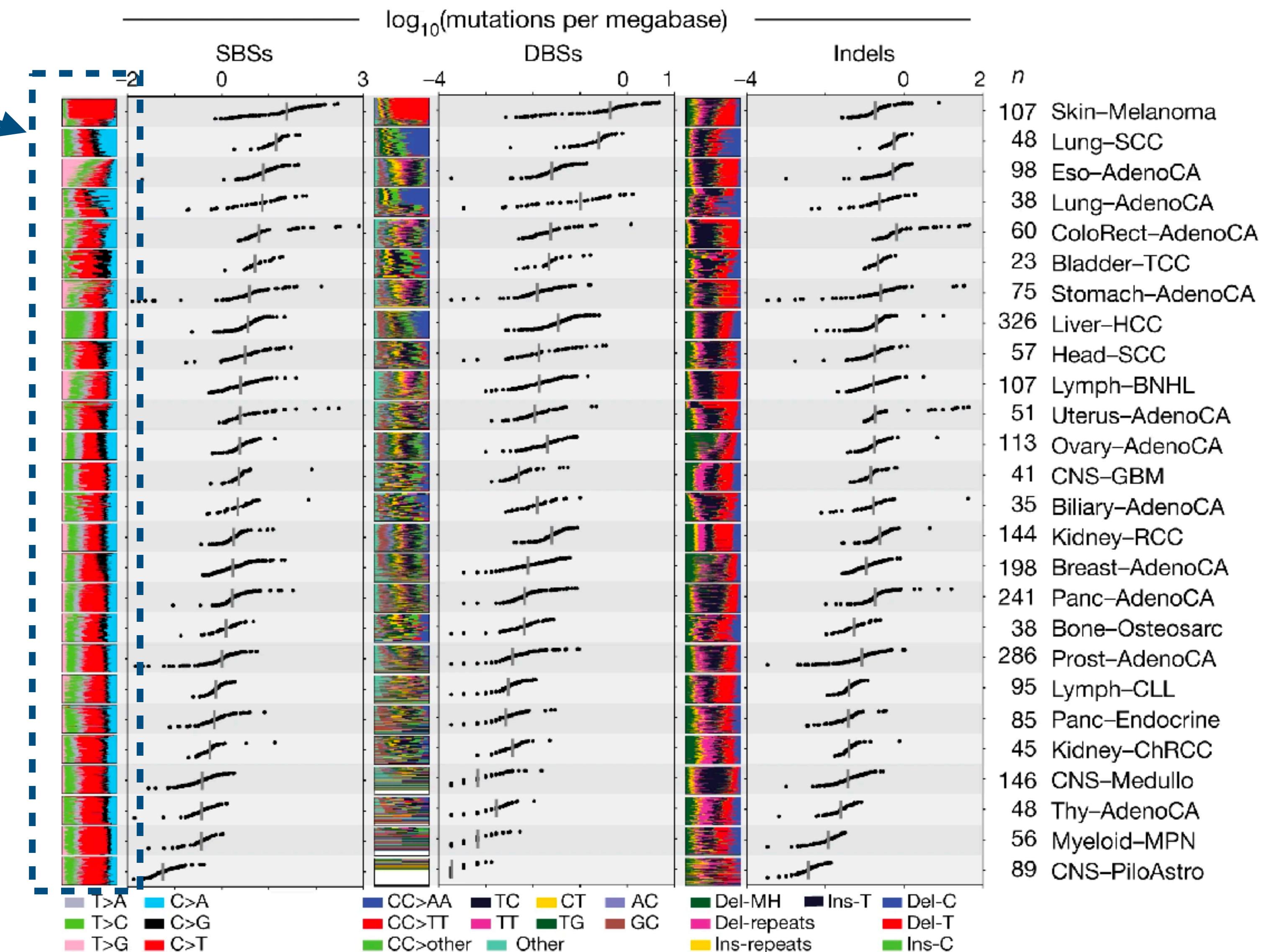
Lee and Seung, *Nature* (1999)

NMF-based method can facilitate data integration

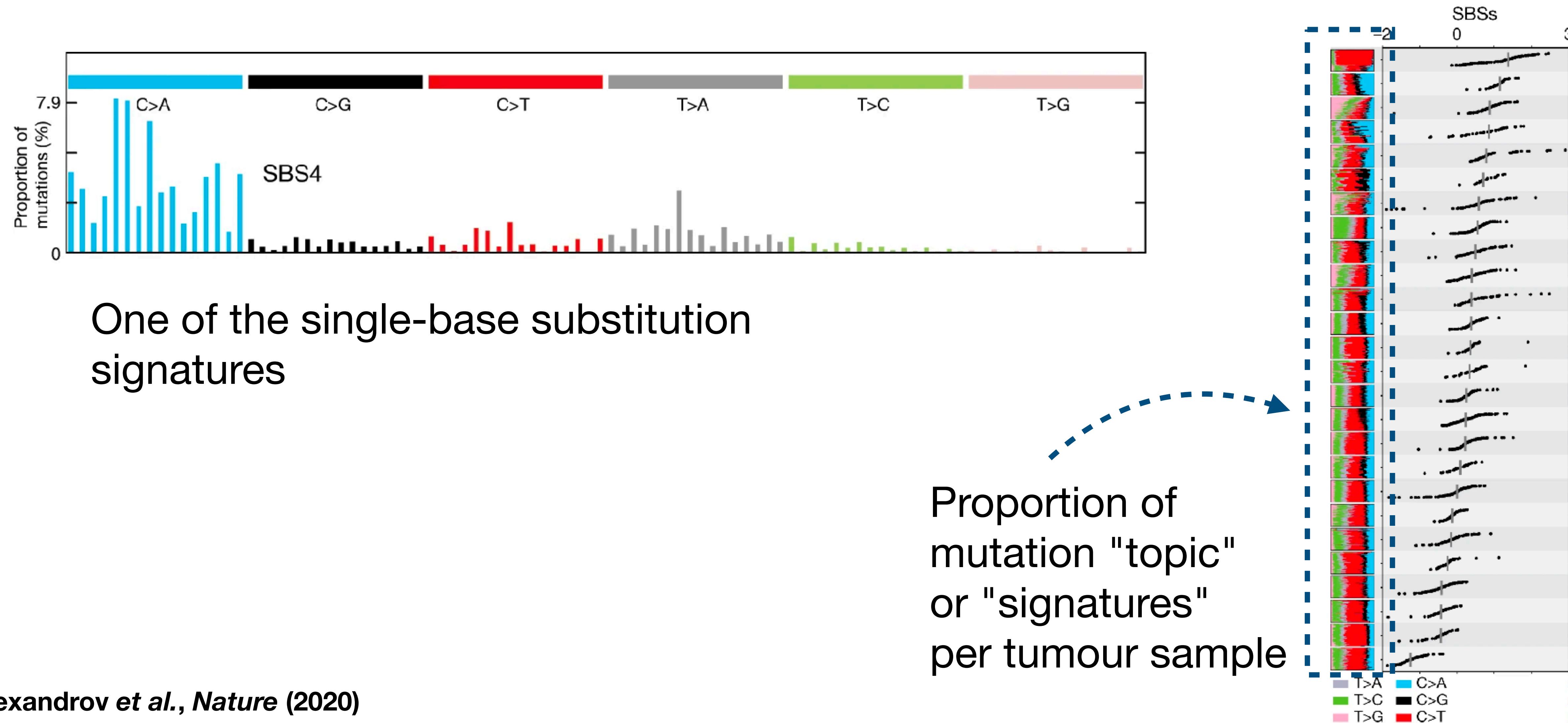


NMF can identify hidden patterns in high-dimensional count data

Proportion of mutation "topic" or "signatures" per tumour sample

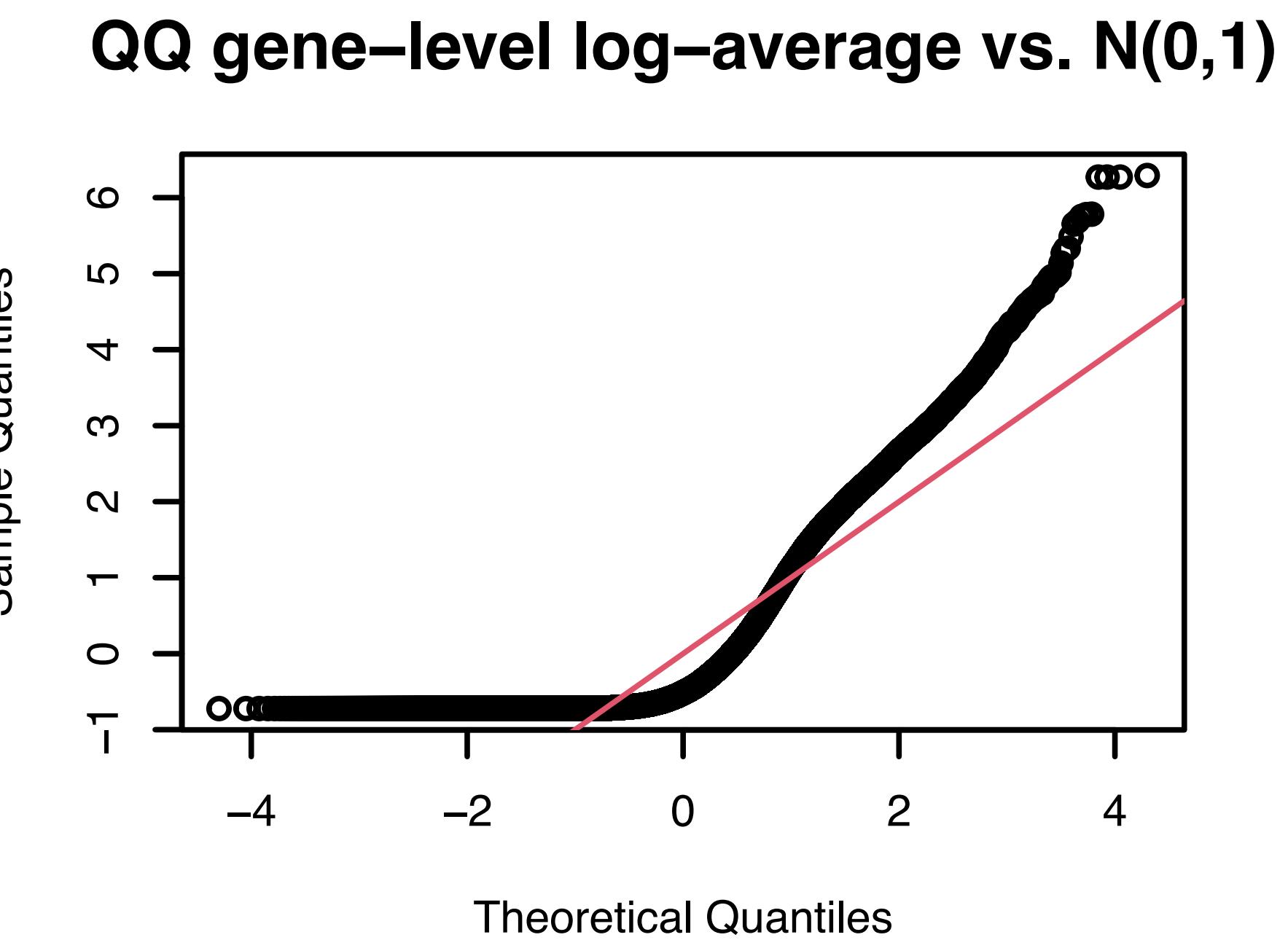
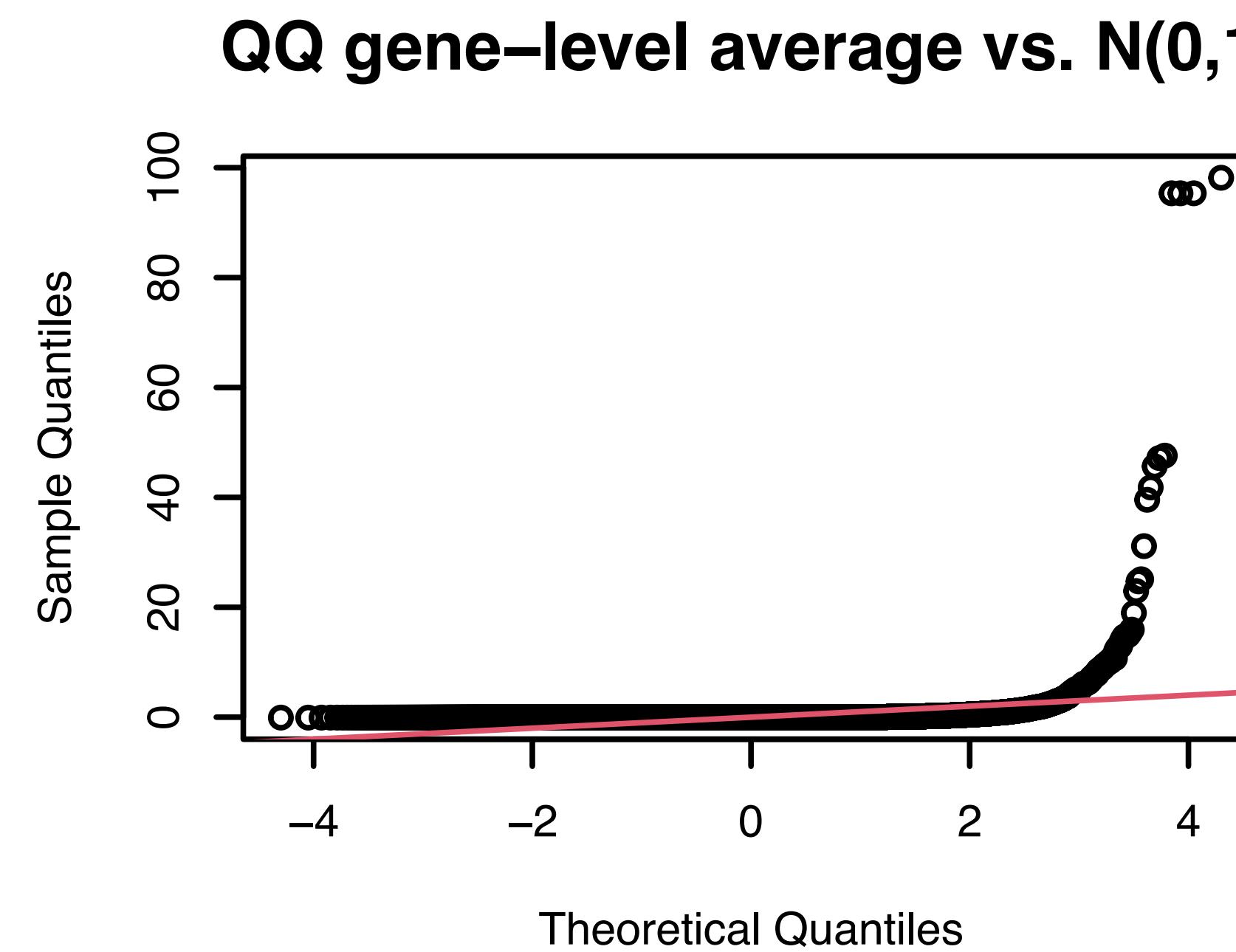


NMF can identify hidden patterns in high-dimensional count data



Why another factorization method for genomics data?

Most genomics data were measured by counting the number of short reads. It is generally hard to make it look “normal,” and the normality assumption may introduce unwanted bias.



What is NMF? How can we find NMF solutions?

For each **non-negative** element X_{ij} of a matrix, we want to find non-negative U and V (factors and factor-loading values):

$$X_{ij} \approx \sum_{k=1}^K U_{ik} V_{kj}$$

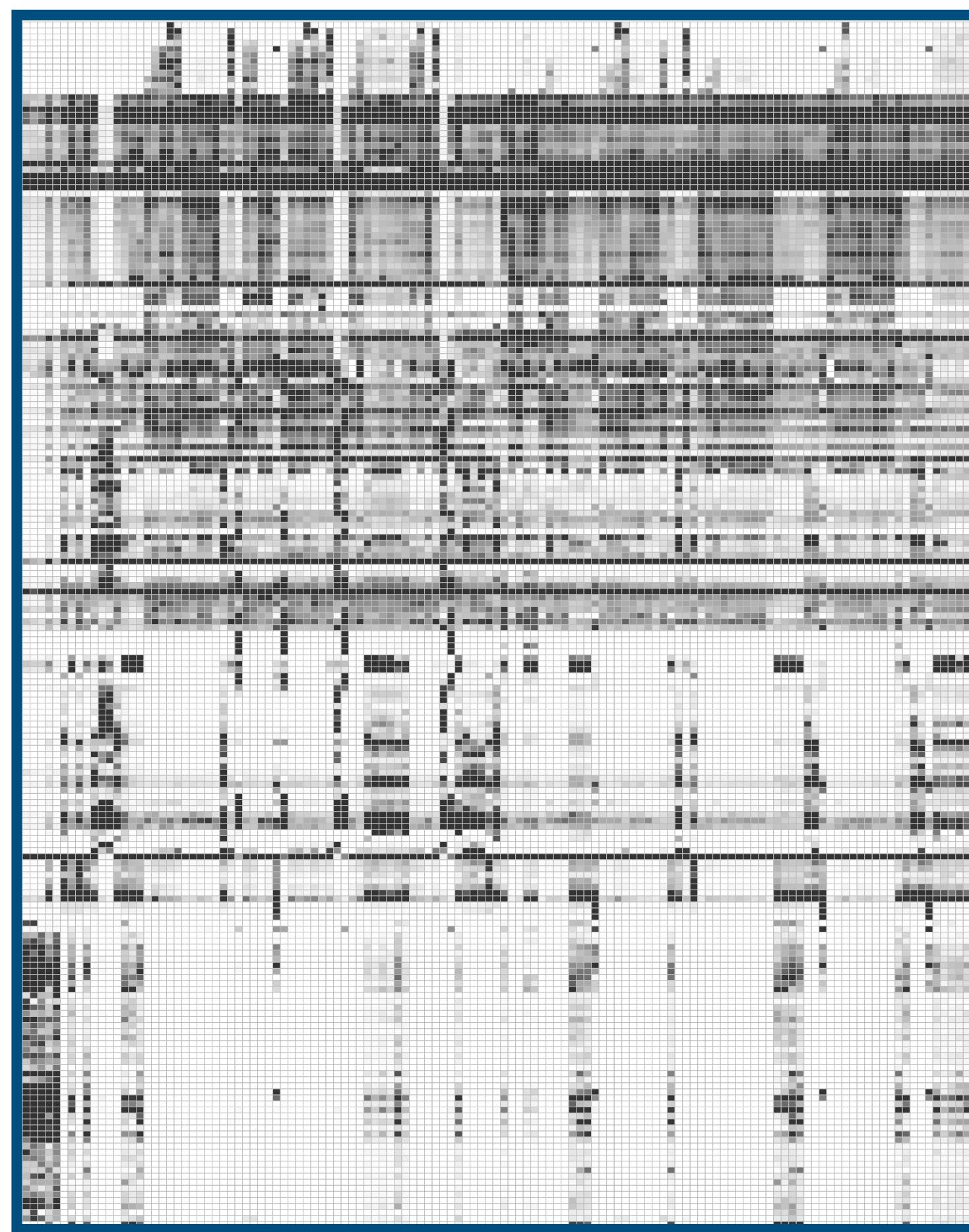
with

$$U_{ik} \geq 0, V_{kj} \geq 0.$$

We can use some packages e.g., NMF in R.

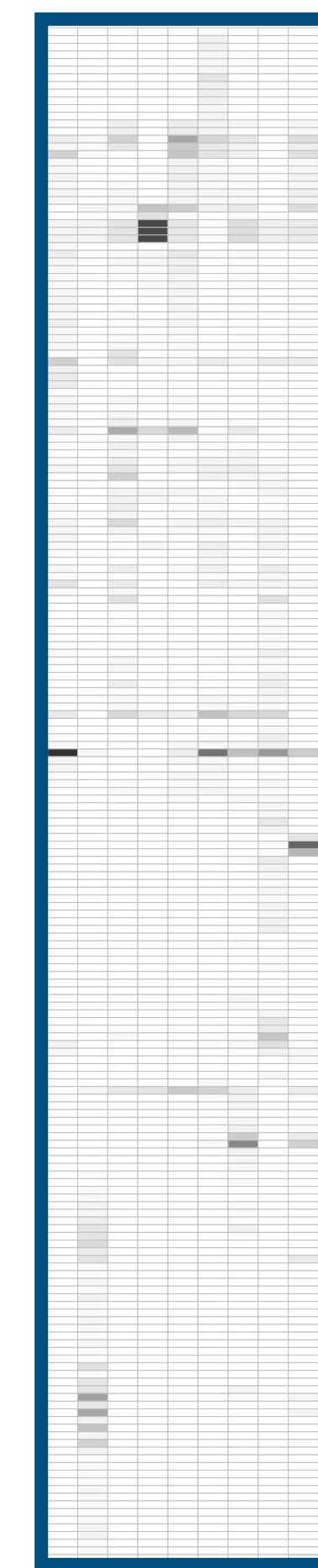
```
library(NMF)
out <- nmf(x.sub, rank=10)
U <- basis(out)
V <- t(coef(out))
```

NMF naturally induces (1) gene sets (2) sample loading



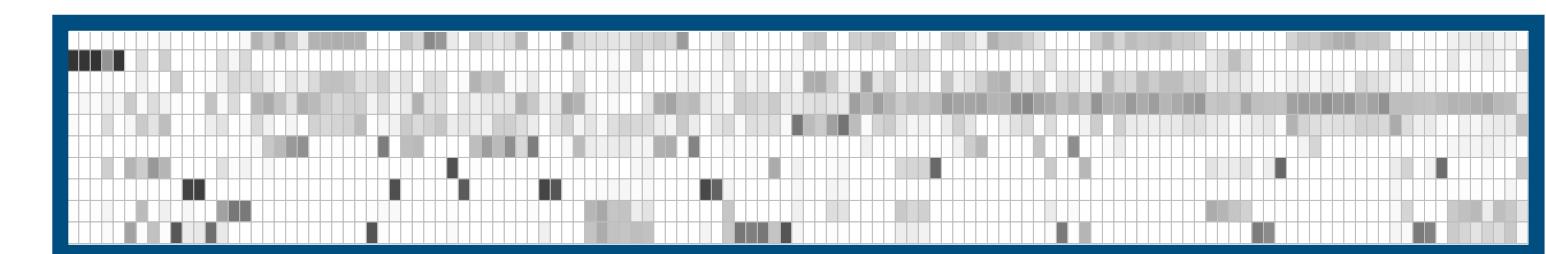
Y

?



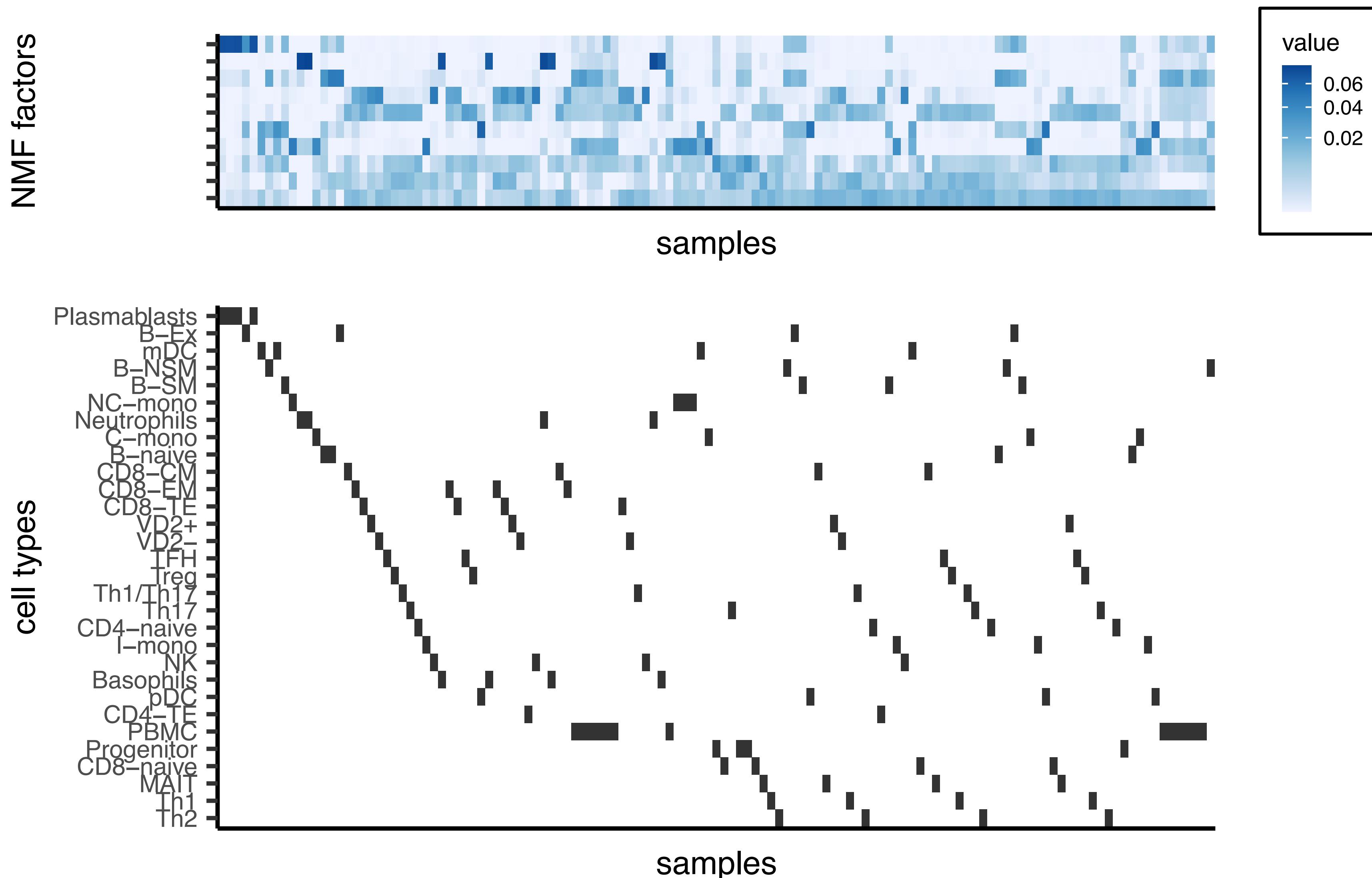
U

×

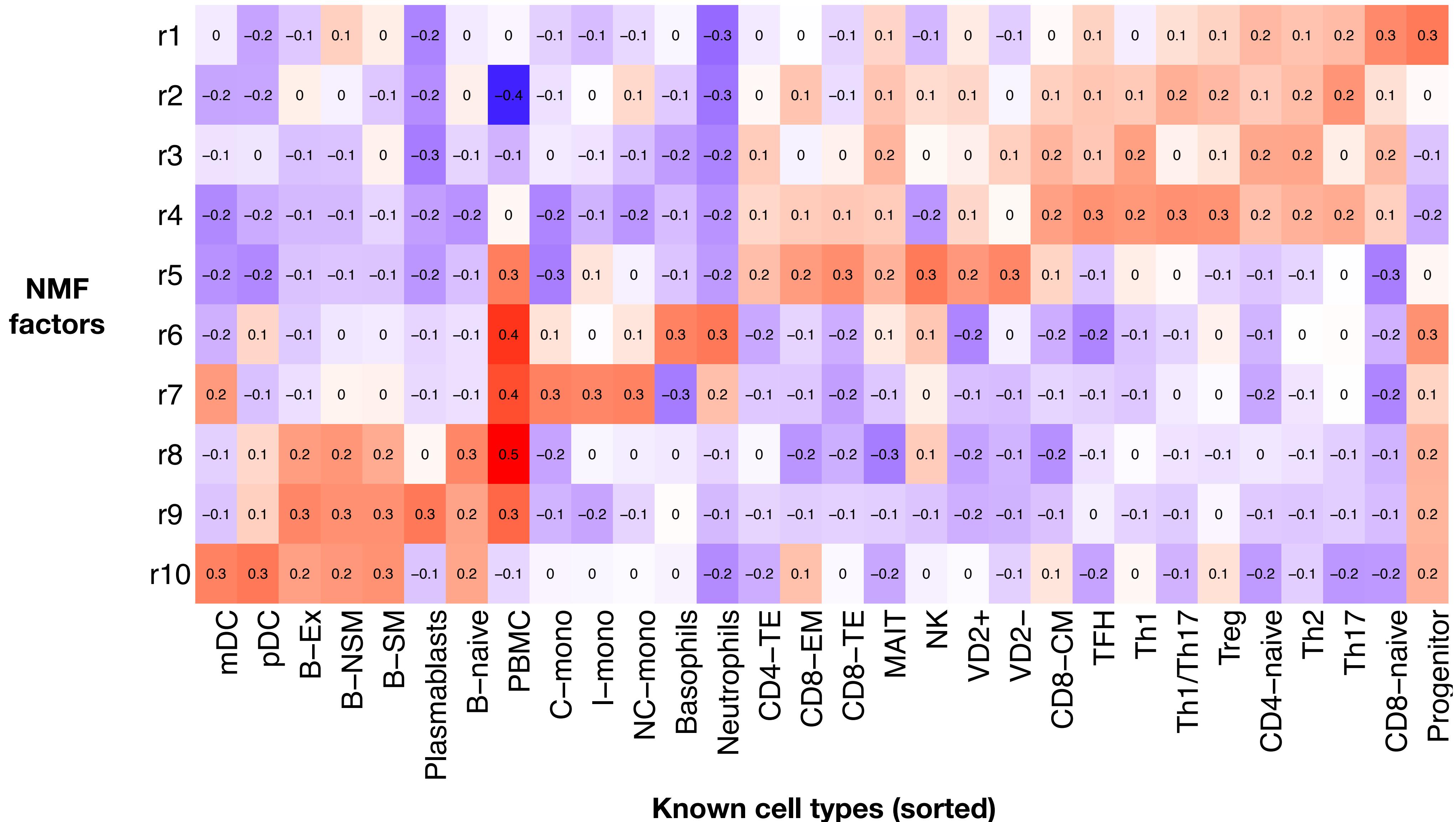


V

Some NMF factors are highly correlated with cell types

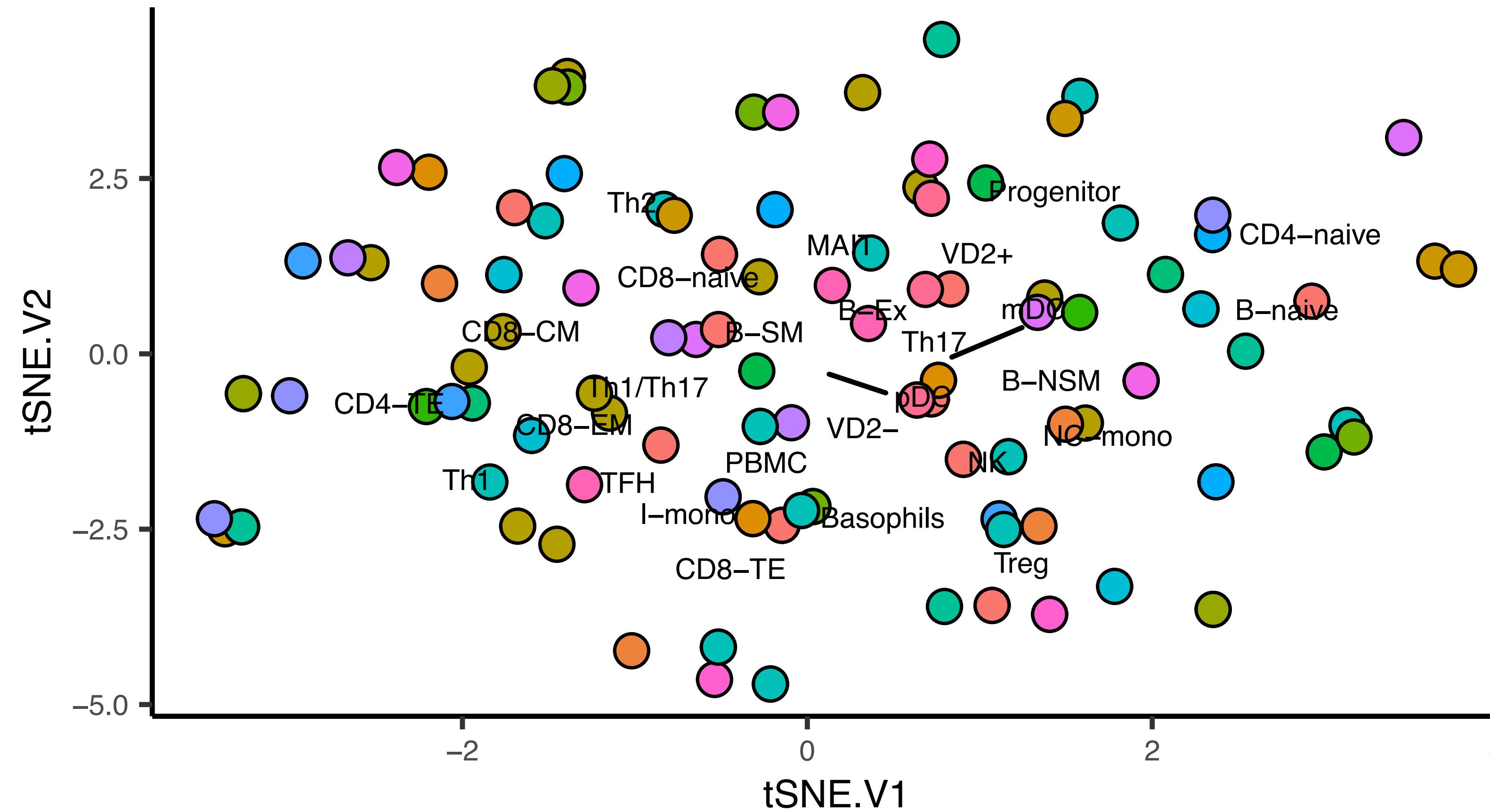


Some NMF factors are highly correlated with cell types



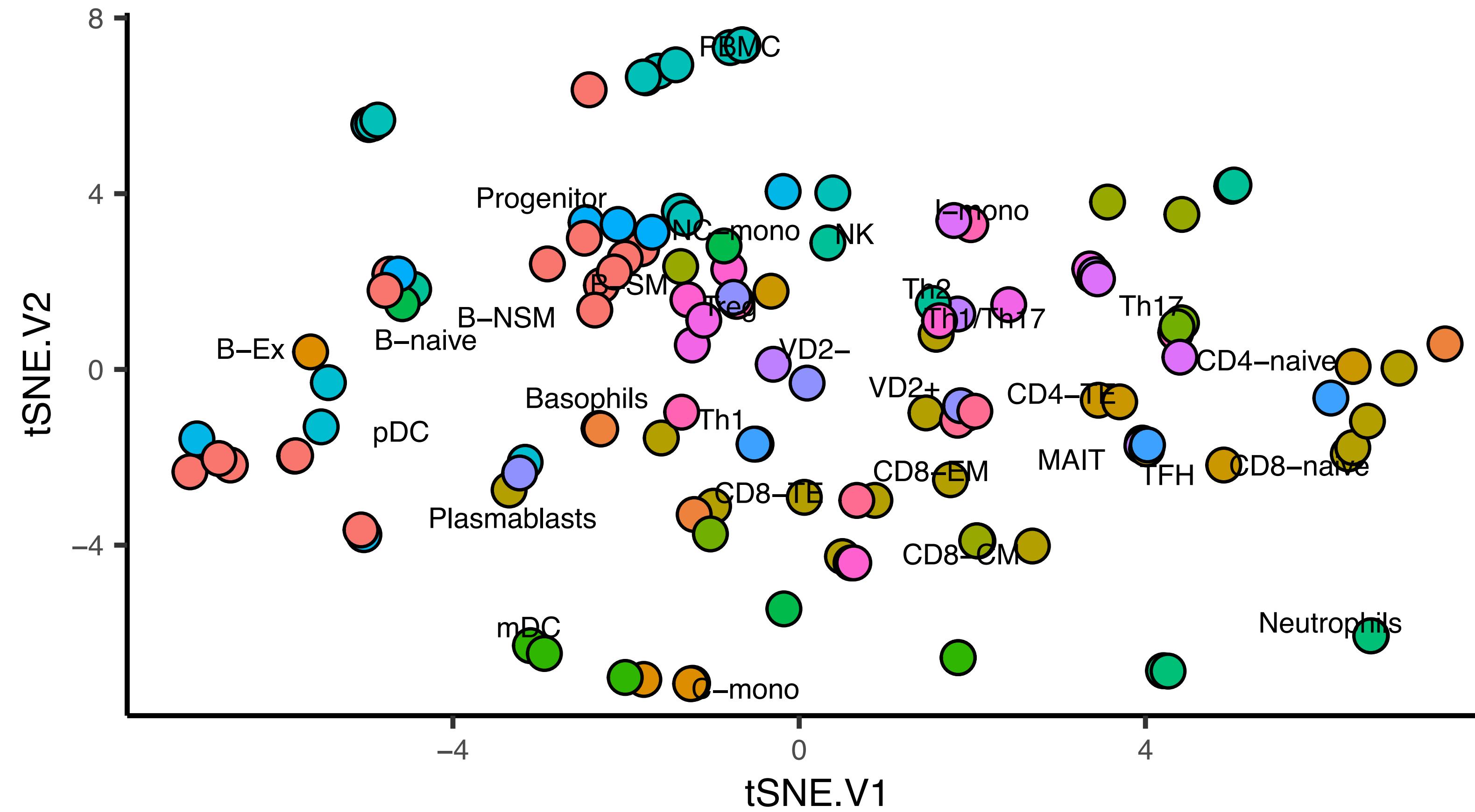
Sample-specific factor loading matrix can be an input to other methods

```
.tsne <- Rtsne::Rtsne(apply(.svd$v, 2, scale))
```

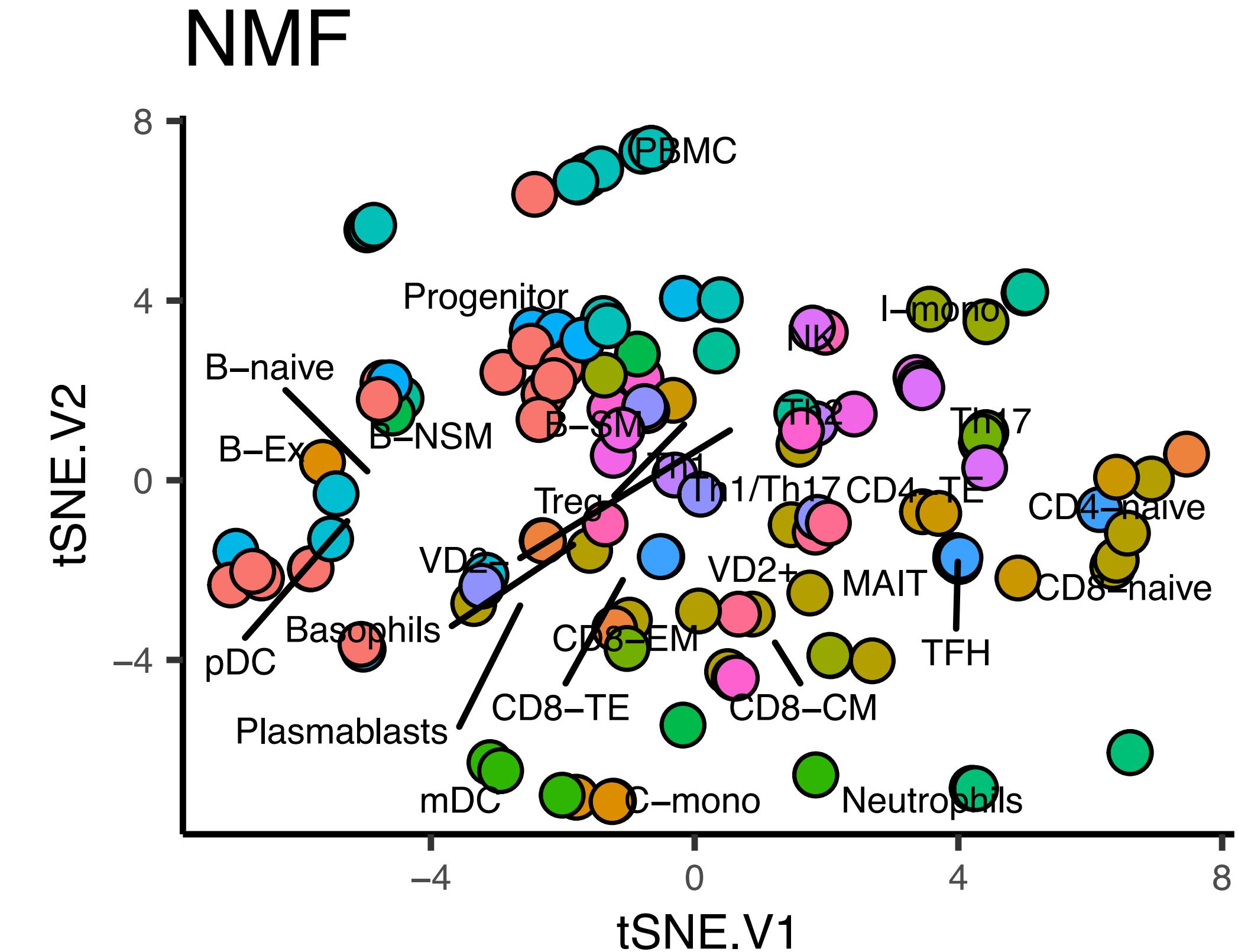
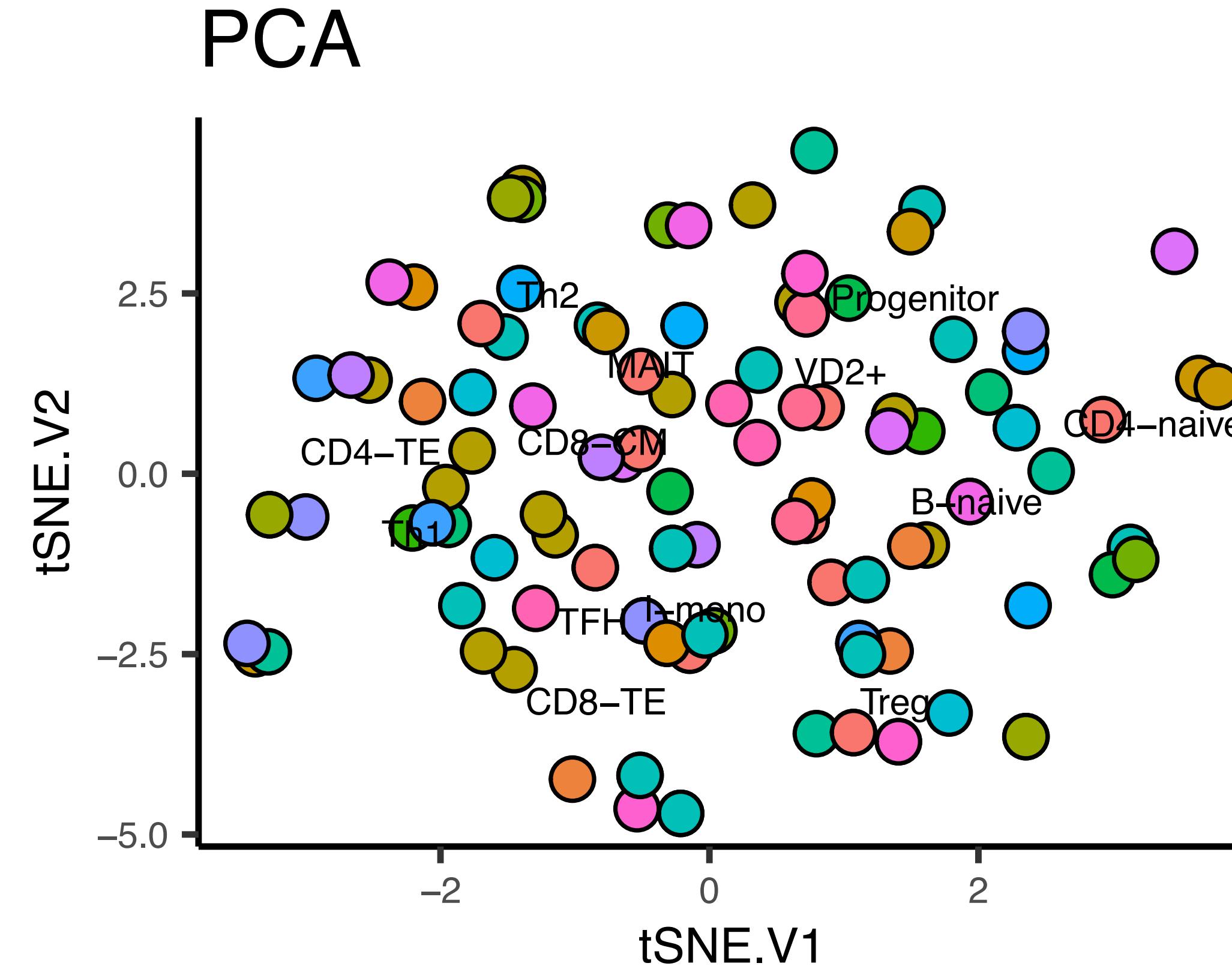


Sample-specific factor loading matrix can be an input to other methods

```
.tsne <- Rtsne::Rtsne(apply(log(V), 2, scale))
```



PCA vs. NMF: Which one looks better?



Today's lecture: Enrichment Analysis

- **Motivations: What's next after genomics analysis?**
 - What have we learned? How do we know that our discovery is meaningful?
- **Gene set enrichment analysis**
 - Set-based approach: Hypergeometric test
 - Rank-based approach: GSEA by KS statistic
- **Can we engineer new gene sets/scores?**
 - Principal Component Analysis
 - Matrix factorization of count data