QUEEN'S UNIVERSITY BELFAST
SCHOOL OF ELECTRONICS, ELECTRICAL ENGNIEERING AND COMPUTER SCIENCE

Data Analysis & Visualisation

CSC3062

BEng (CS & SE), MEng (CS & SE), BIT & CIT

Dr Reza Rafiee

Semester 1 2019

QUEEN'S UNIVERSITY BELFAST | SCHOOL OF ELECTRONICS, ELECTRICAL ENGNIEERING AND COMPUTER SCIENCE

Principal component analysis (PCA)

PCA analysis using *prcomp()* package

	England	N Ireland	Scotland	Wales
Alcoholic drinks	375	135	458	475
Beverages	57	47	53	73
Carcase meat	245	267	242	227
Cereals	1472	1494	1462	1582
Cheese	105	66	103	103
Confectionery	54	41	62	64
Fats and oils	193	209	184	235
Fish	147	93	122	160
Fresh fruit	1102	674	957	1137
Fresh potatoes	720	1033	566	874
Fresh Veg	253	143	171	265
Other meat	685	586	750	803
Other Veg	488	355	418	570
Processed potatoes	198	187	220	203
Processed Veg	360	334	337	365
Soft drinks	1374	1506	1572	1256
Sugars	156	139	147	175

Input_dataset

PCA →

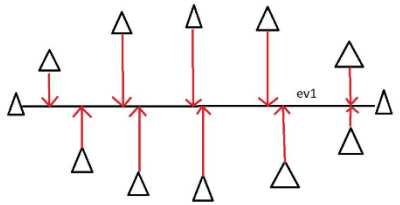
	England	N Ireland	Scotland	Wales
PC1	-144.993	477.3916	-91.8693	-240.529
PC2	2.532999	58.90186	-286.082	224.6469
PC3	105.7689	-4.8779	-44.4155	-56.4756

Reduced dataset

Summarises of features

PC | eigenvector and eigenvalue

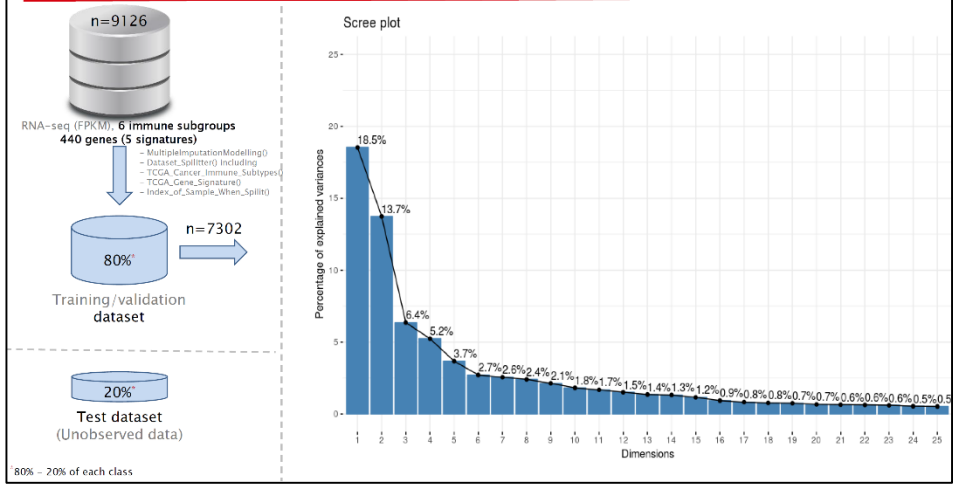
The horizontal line is therefore the principal component in this example.



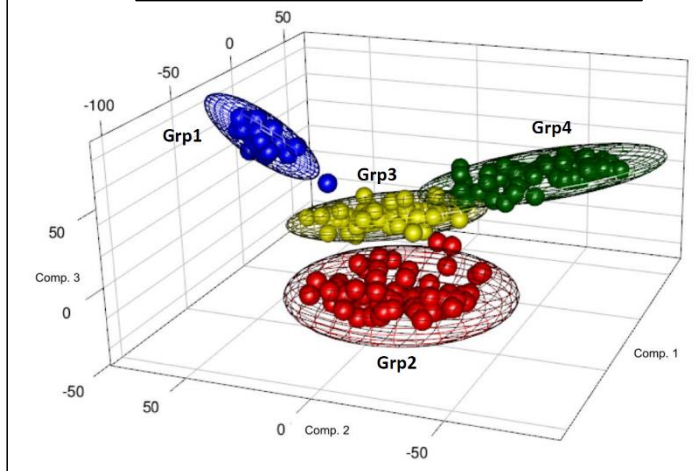
The direction of this line is called **eigenvector**.

An **eigenvalue** is a number telling us how spread out the data is on the line.

Consider 9126 samples with 440 features



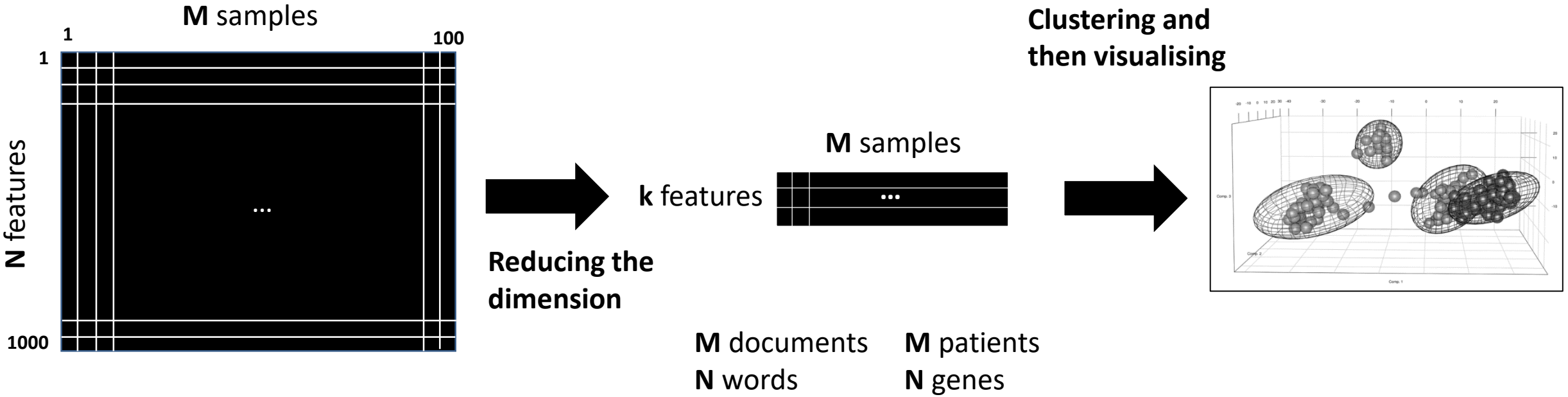
220 samples with 17 features



PCA visualisation of groups identified using a consensus NMF clustering

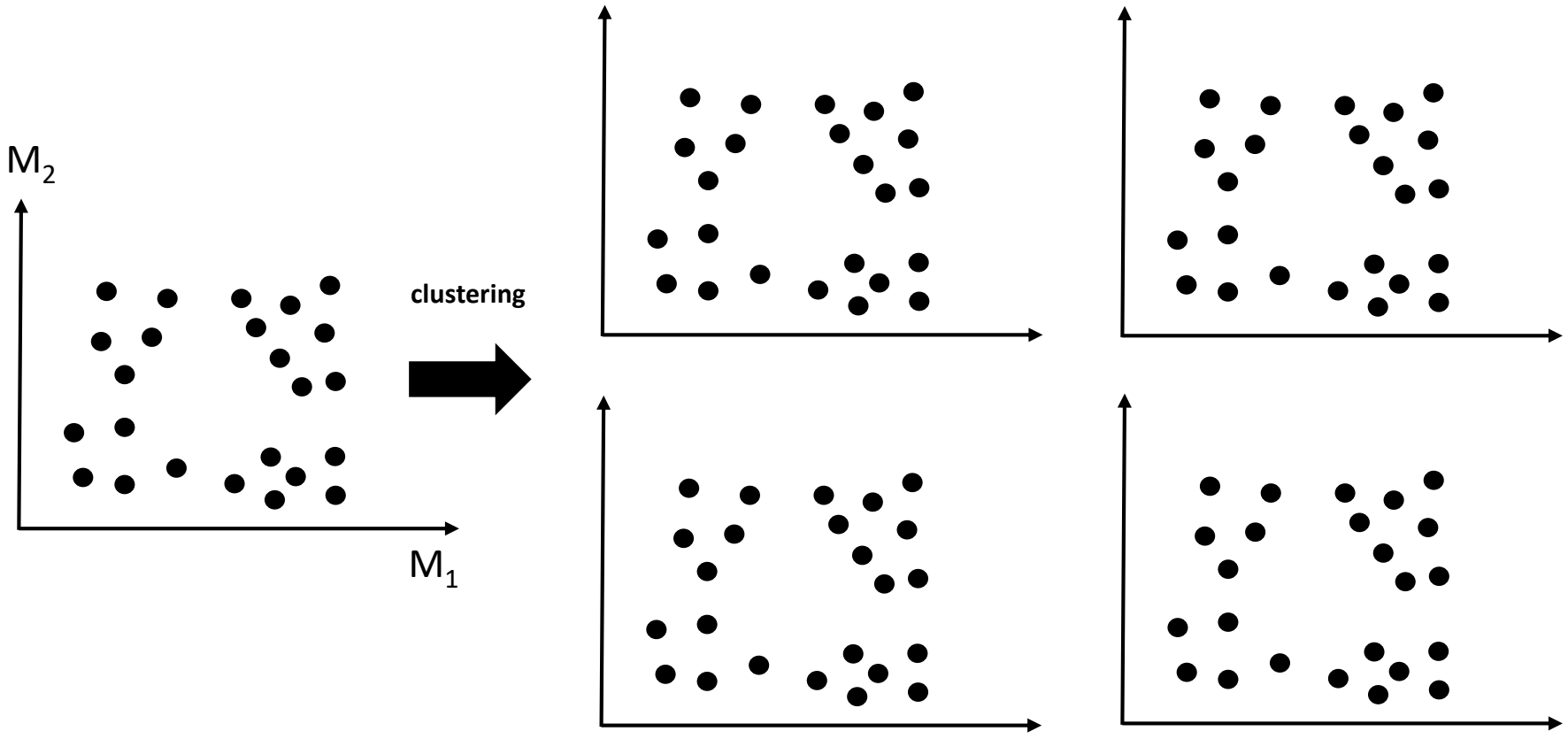
Question: which technique/method?

Consider a dataset with about N=1000 features and M=100 samples (all have positive numeric values). In a data analytics task, we are asked to cluster this samples into different groups based on firstly reducing the dimension of feature space and then a clustering method.



What is clustering?

Clustering concept

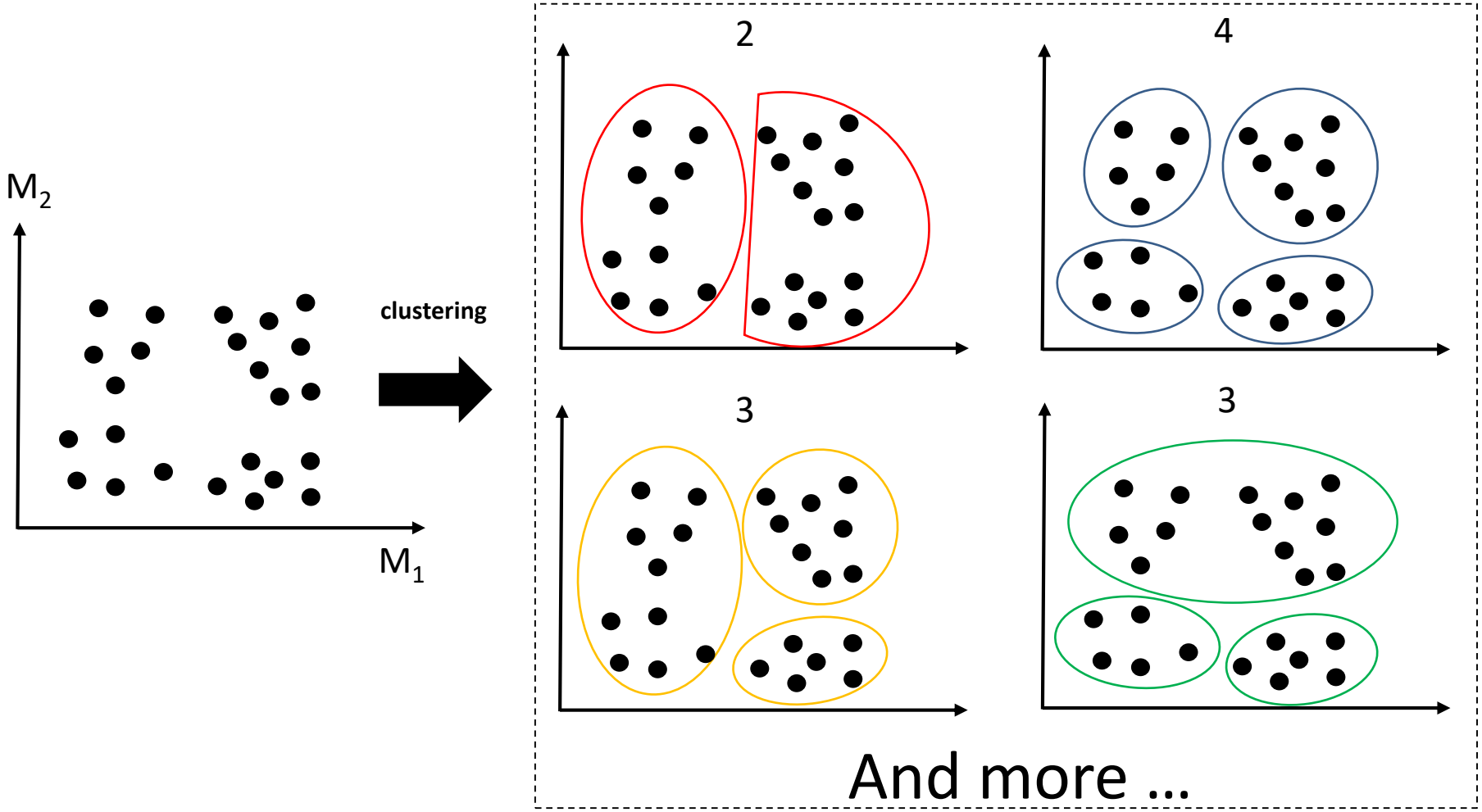


And more …

In case of applying an appropriate clustering method (with well-adjusted parameters/initialisation, bootstrapping and cross-validation techniques), we could have distinct groups (with possibly different number of clusters) but they might be meaningless!

What is clustering?

Clustering concept



In case of applying an appropriate clustering method (with well-adjusted parameters/initialisation, bootstrapping and cross-validation techniques), we could have distinct groups (with possibly different number of clusters) but they might be meaningless!

Non-negative matrix factorization

NMF

Non-negative matrix factorization: NMF

- A dimensionality reduction technique
  - based on decomposition by parts
- An efficient method for identification of distinct patterns (e.g., class discovery and clustering)

NMF

Applications NMF in

- Text mining
- Astronomy
- Spectral data analysis
- Speech processing (denoising)
- Image processing (object detection)
- Bioinformatics (& biological data analysis)

NMF

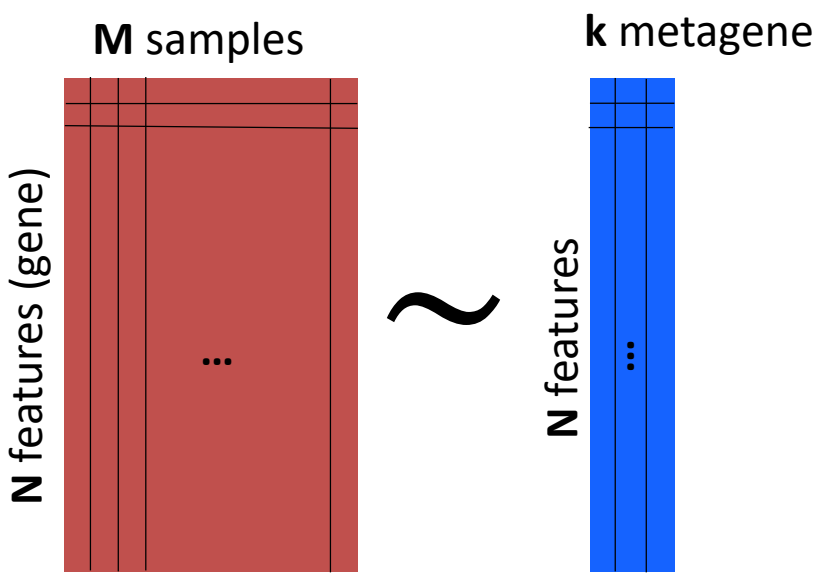
An iterative algorithm aiming at factorising an input matrix **A** into two matrices with positive entries.

$$A \sim WH$$

$$N \times M \qquad N \times k \qquad k \times M$$

k is the rank of H

$\mathbf{k} \ll \mathbf{N}$



N: Number of features (e.g., gene, word) is typically in the thousands

NMF

An iterative algorithm aiming at factorising an input matrix **A** into two matrices with positive entries.

$$\underset{N \times M}{A} \sim \underset{N \times k}{W}\,\underset{k \times M}{H}$$

What is a matrix rank?

NMF – matrix rank

The rank of a matrix definition:
The maximum number of **linearly independent** columns (non-zero) in the matrix

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 6 & 8 & 20 \end{bmatrix}_{2 \times 3}$$

$$B = \begin{bmatrix} 1 & 0 & 1 \\ -2 & -3 & 1 \\ 3 & 3 & 0 \end{bmatrix}_{3 \times 3}$$

What is a matrix rank?

NMF – matrix rank

The rank of a matrix definition:
the maximum number of **linearly independent** columns (non-zero) in the matrix

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 6 & 8 & 20 \end{bmatrix}_{2 \times 3}$$

Rank(A) = 3

$$B = \begin{bmatrix} 1 & 0 & 1 \\ -2 & -3 & 1 \\ 3 & 3 & 0 \end{bmatrix}_{3 \times 3}$$

Rank(B) = ?

What is a matrix rank?

NMF – matrix rank

The rank of a matrix definition:
the maximum number of **linearly independent** columns (non-zero) in the matrix

$$\begin{bmatrix}1\\-2\\3\end{bmatrix} - \begin{bmatrix}0\\-3\\3\end{bmatrix} = \begin{bmatrix}1\\1\\0\end{bmatrix}$$

$$B = \begin{bmatrix}1 & 0 & 1\\-2 & -3 & 1\\3 & 3 & 0\end{bmatrix}_{3\times 3}$$

The third column is a linear combination of the first two columns (the second subtracted from the first), the three columns are **linearly dependent** so the rank must be less than 3

Rank(B) = 2

What is a matrix rank?

NMF – matrix rank

The rank of a matrix definition:
the maximum number of **linearly independent** columns (non-zero) in the matrix

$$C = \begin{bmatrix} 1 & 1 & 0 & 2 \\ -1 & -1 & 0 & -2 \end{bmatrix}_{2 \times 4}$$

$$C^T = \begin{bmatrix} 1 & -1 \\ 1 & -1 \\ 0 & 0 \\ 2 & -2 \end{bmatrix}_{4 \times 2}$$

Rank(C) = ?

Any pair of columns is linearly dependent (ignore non-zero column)

What is a matrix rank?

NMF – matrix rank

The rank of a matrix definition:
the maximum number of **linearly independent** columns (non-zero) in the matrix

$$C = \begin{bmatrix} 1 & 1 & 0 & 2 \\ -1 & -1 & 0 & -2 \end{bmatrix}_{2 \times 4}$$

$$C^T = \begin{bmatrix} 1 & -1 \\ 1 & -1 \\ 0 & 0 \\ 2 & -2 \end{bmatrix}_{4 \times 2}$$

Rank(C) = 1

Any pair of columns is linearly dependent (ignore non-zero column).

What is a matrix rank?

NMF

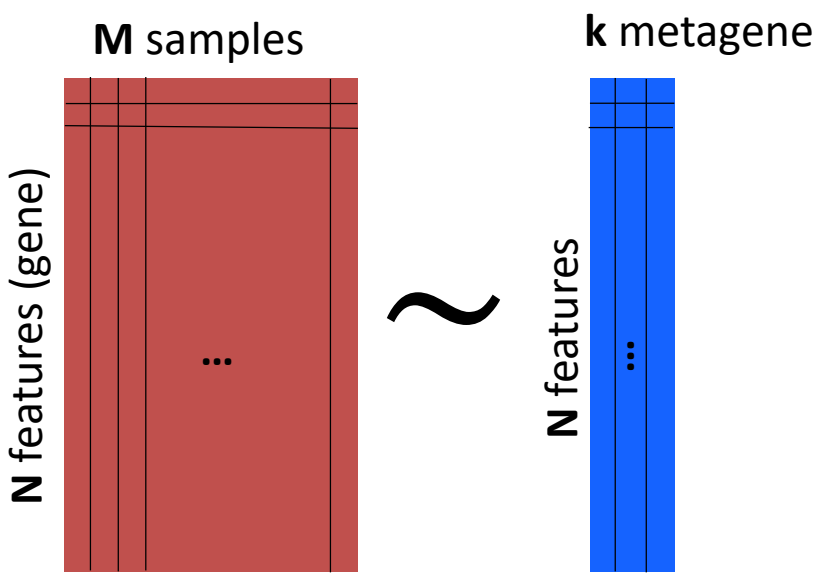
Factorising matrix A into two matrices with positive entries.

Matrix W has size N×k, with each of the k columns defining a metagene.

$$A \sim WH$$

$$N \times M \qquad N \times k \qquad k \times M$$

K is the rank of H



N: Number of features (e.g., gene, word) is typically in the thousands

NMF

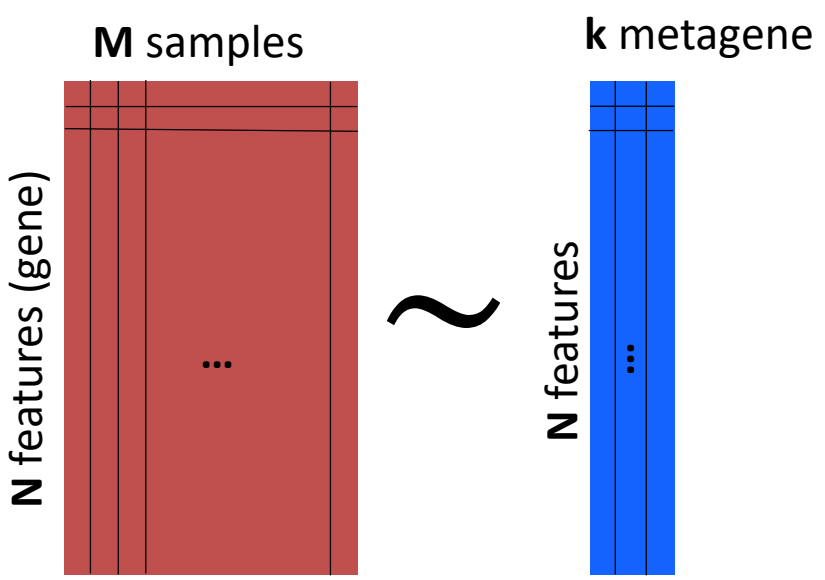
Factorising matrix A into two matrices with positive entries.

Matrix H has size k×M, with each of the M columns representing the metagene values of the corresponding sample.

$$A \sim WH$$

$N \times M$ $\quad N \times k$ $\quad k \times M$

K is the rank of H



N: Number of features (e.g., gene, word) is typically in the thousands

NMF

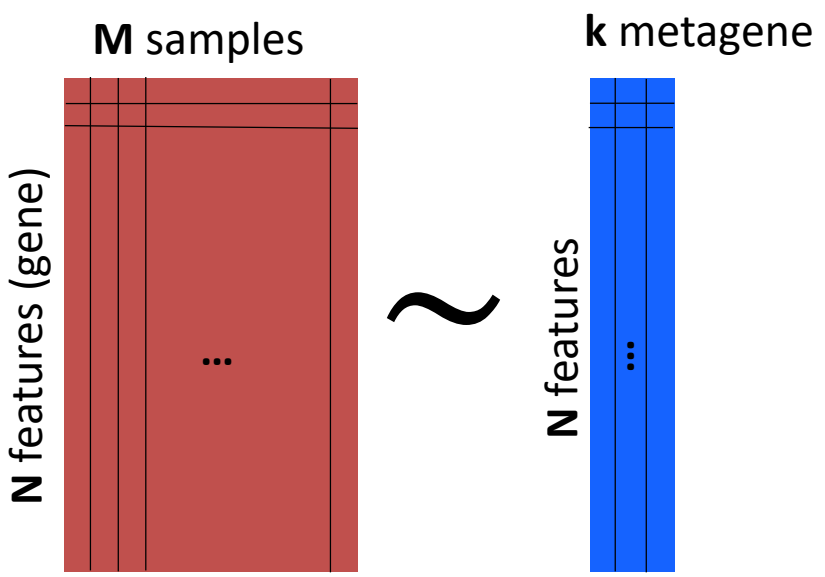
Factorising matrix A into two matrices with positive entries.

For **any rank k**, the NMF algorithm **groups** the samples into clusters.

$$A \sim WH$$

A: $N \times M$ — W: $N \times k$ — H: $k \times M$

K is the rank of H



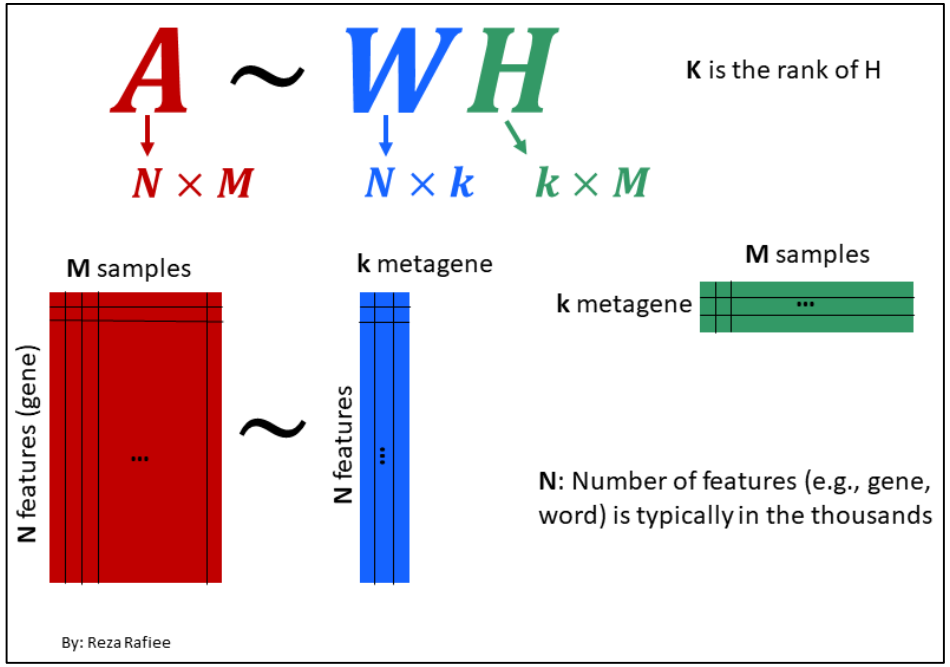
N: Number of features (e.g., gene, word) is typically in the thousands

NMF

Factorising matrix A into two matrices with positive entries.

For **any rank k**, the NMF algorithm **groups** the samples into clusters.

[+]The important question that we need to address is whether a given rank k could decompose the samples into "**meaningful**" clusters or not.

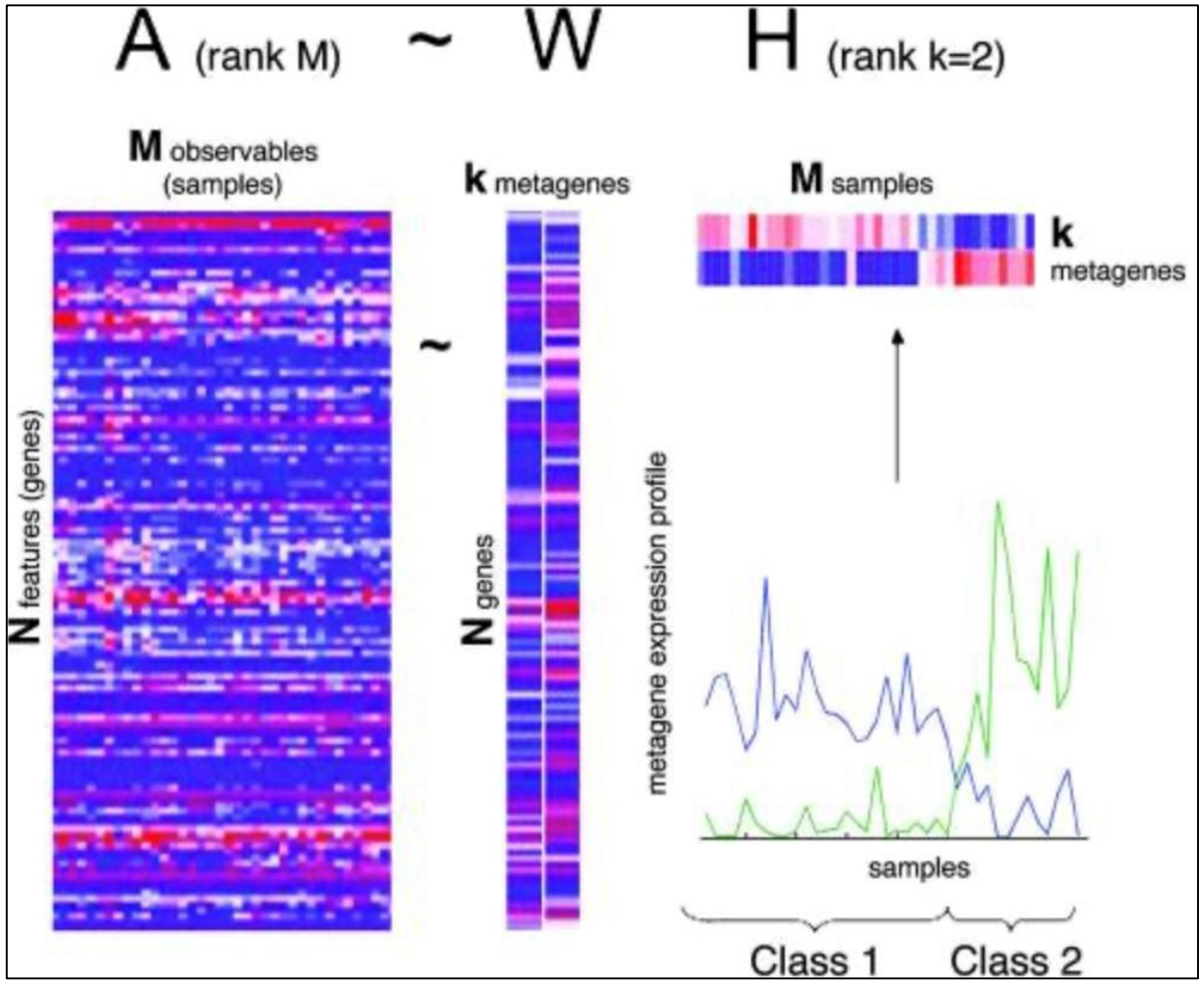


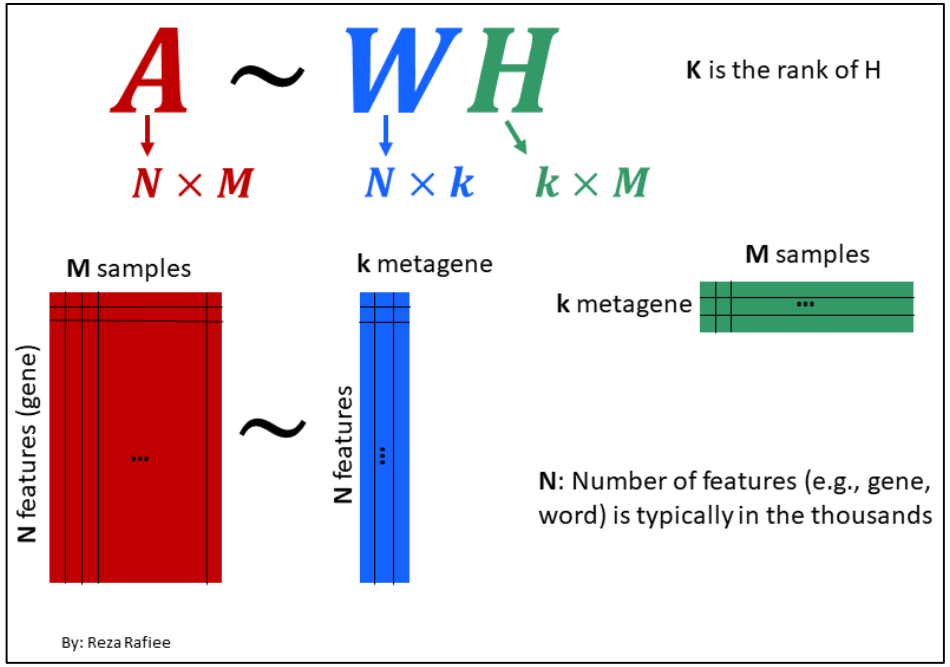
[+]Metagenes and molecular pattern discovery using matrix factorization, J. Brunet et al., PNAS, 2004

NMF

Factorising matrix A into two matrices with positive entries.

For **any rank k**, the NMF algorithm **groups** the samples into clusters.





The above image originated from the paper: Metagenes and molecular pattern discovery using matrix factorization, J. Brunet et al., PNAS, 2004

Description of the implemented NMF algorithms

Key	Description
`brunet`	Standard NMF. Based on Kullback-Leibler divergence, it uses simple multiplicative updates from (**Lee2001**), enhanced to avoid numerical underflow. $$H_{kj} \leftarrow H_{kj} \frac{\left(\sum_l \frac{W_{lk} V_{lj}}{(WH)_{lj}}\right)}{\sum_l W_{lk}} \quad (3)$$ $$W_{ik} \leftarrow W_{ik} \frac{\sum_l [H_{kl} A_{il}/(WH)_{il}]}{\sum_l H_{kl}} \quad (4)$$ **Reference:** (**Brunet2004**)
`lee`	Standard NMF. Based on euclidean distance, it uses simple multiplicative updates $$H_{kj} \leftarrow H_{kj} \frac{(W^T V)_{kj}}{(W^T WH)_{kj}} \quad (5)$$ $$W_{ik} \leftarrow W_{ik} \frac{(VH^T)_{ik}}{(WHH^T)_{ik}} \quad (6)$$ **Reference:** (**Lee2001**)
`nsNMF`	Non-smooth NMF. Uses a modified version of Lee and Seung's multiplicative updates for Kullback-Leibler divergence to fit a extension of the standard NMF model. It is meant to give sparser results. **Reference:** (**Pascual-Montano2006**)
`offset`	Uses a modified version of Lee and Seung's multiplicative updates for euclidean distance, to fit a NMF model that includes an intercept. **Reference:** (**Badea2008**)
`pe-nmf`	Pattern-Expression NMF. Uses multiplicative updates to minimize an objective function based on the Euclidean distance and regularized for effective expression of patterns with basis vectors. **Reference:** (**Zhang2008**)
`snmf/r`, `snmf/l`	Alternating Least Square (ALS) approach. It is meant to be very fast compared to other approaches. **Reference:** (**KimH2007**)

```r
# Install
install.packages('NMF')
# Load
library(NMF)
```

Description of the implemented NMF algorithms

```
# list all available algorithms
nmfAlgorithm()

##  [1] "brunet"    "KL"        "lee"       "Frobenius" "offset"
##  [6] "nsNMF"     "ls-nmf"    "pe-nmf"    "siNMF"     "snmf/r"
## [11] "snmf/l"

# retrieve a specific algorithm: 'brunet'
nmfAlgorithm('brunet')

## <object of class: NMFStrategyIterative>
##  name: brunet [NMF]
##  objective: 'KL'
##  model: NMFstd
##  <Iterative schema>
##   onInit: none
##   Update: function (i, v, x, copy = FALSE, eps = .Machine$double.eps, ...)
##   Stop: 'connectivity'
##   onReturn: none
```

Initialisation methods

Key	Description
`ica`	Uses the result of an Independent Component Analysis (ICA) (from the *fastICA* package[5] (**Rpackage:fastICA**)). Only the positive part of the result are used to initialize the factors.
`nnsvd`	Nonnegative Double Singular Value Decomposition. The basic algorithm contains no randomization and is based on two SVD processes, one approximating the data matrix, the other approximating positive sections of the resulting partial SVD factors utilizing an algebraic property of unit rank matrices. It is well suited to initialize NMF algorithms with sparse factors. Simple practical variants of the algorithm allows to generate dense factors. **Reference:** (**Boutsidis2008**)
`none`	Fix seed. This method allows the user to manually provide initial values for both matrix factors.
`random`	The entries of each factors are drawn from a uniform distribution over $[0, max(V)]$, where V is the target matrix.

Table 2: Description of the implemented seeding methods to initialize NMF algorithms. The first column gives the key to use in the call to the `nmf` function.

```
nmfSeed('nndsvd')

## <object of class:  NMFSeed >
## name:  nndsvd
## method:  <function>
```

How to run NMF package in R

Method `nmf` provides a single interface to run NMF algorithms. It can directly perform NMF on object of class `matrix` or `data.frame` and `ExpressionSet` – if the *Biobase* package[6] (**Rpackage:Biobase**) is installed. The interface has four main parameters:

```
nmf(x, rank, method, seed, ...)
```

`x` is the target `matrix`, `data.frame` or `ExpressionSet` [7]

`rank` is the factorization rank, i.e. the number of columns in matrix W.

`method` is the algorithm used to estimate the factorization. The default algorithm is given by the package specific option `'default.algorithm'`, which defaults to `'brunet'` on installation (**Brunet2004**).

`seed` is the seeding method used to compute the starting point. The default method is given by the package specific option `'default.seed'`, which defaults to `'random'` on initialization (see method `?rnmf` for details on its implementation).

See also `?nmf` for details on the interface and extra parameters.

Any Questions?