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 →

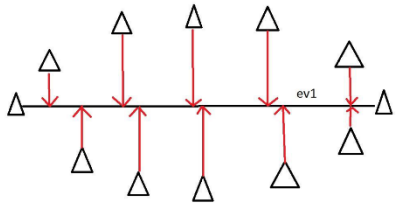
	England	N Ireland	Scotland	Wales
PC1	-144.993	477.3916	-91.8693	-240.529
PC2	2.532999	58.90186	-286.082	224.6469
PC3	105.7689	-4.8779	-44.4155	-56.4756

Reduced dataset

Summarises of features

PC | eigenvector and eigenvalue

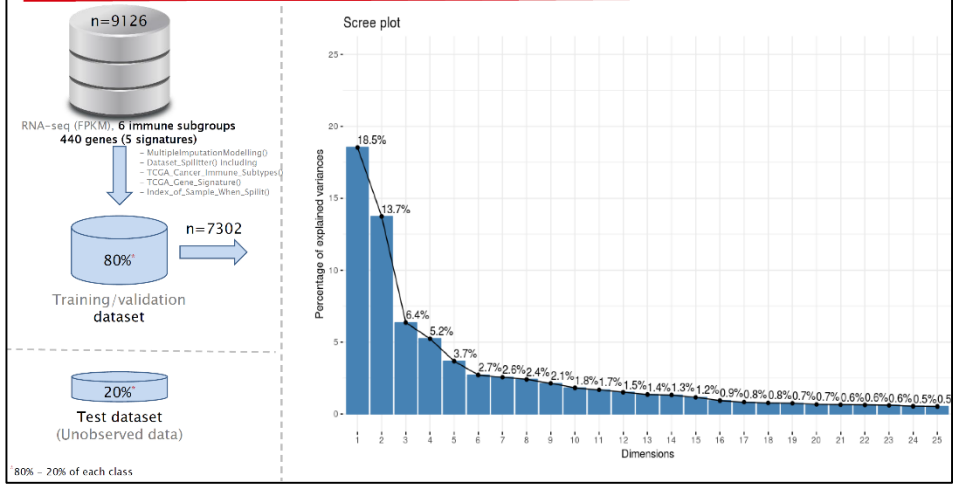
The horizontal line is therefore the principal component in this example.



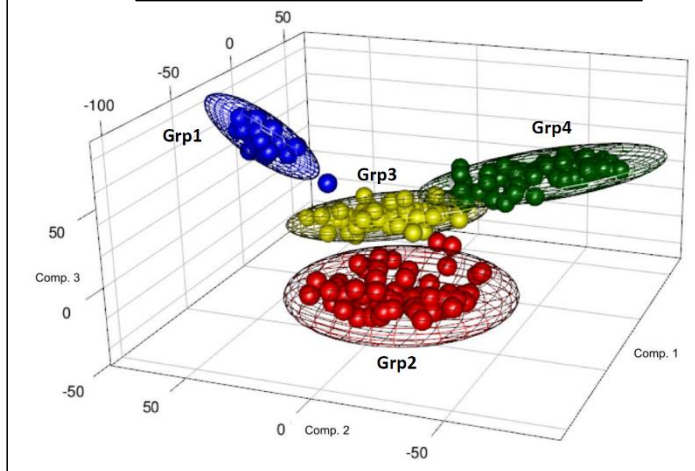
The direction of this line is called **eigenvector**.

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

Consider 9126 samples with 440 features



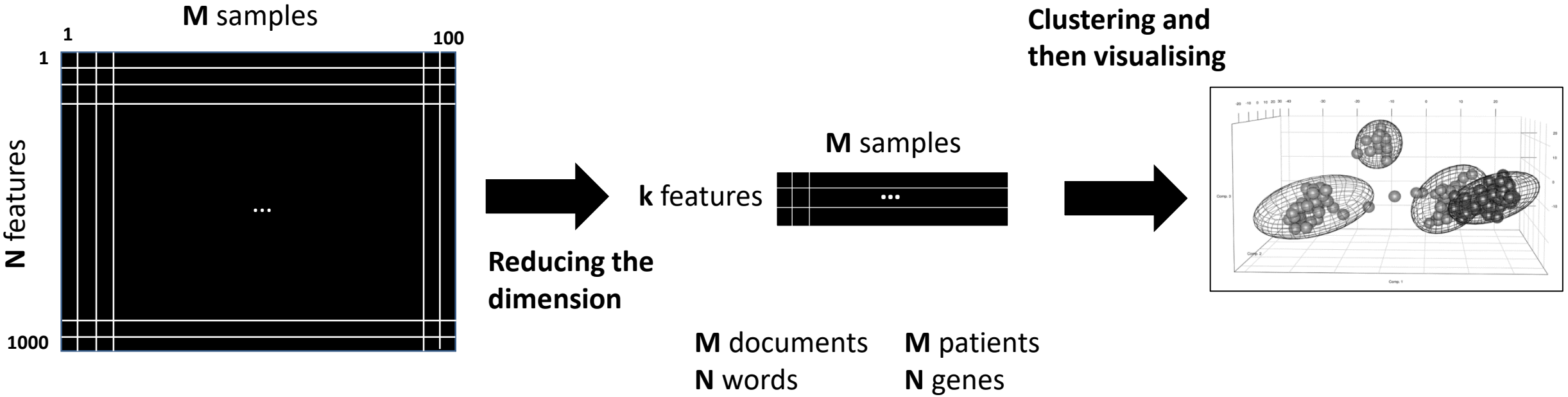
220 samples with 17 features



PCA visualisation of groups identified using a consensus NMF clustering

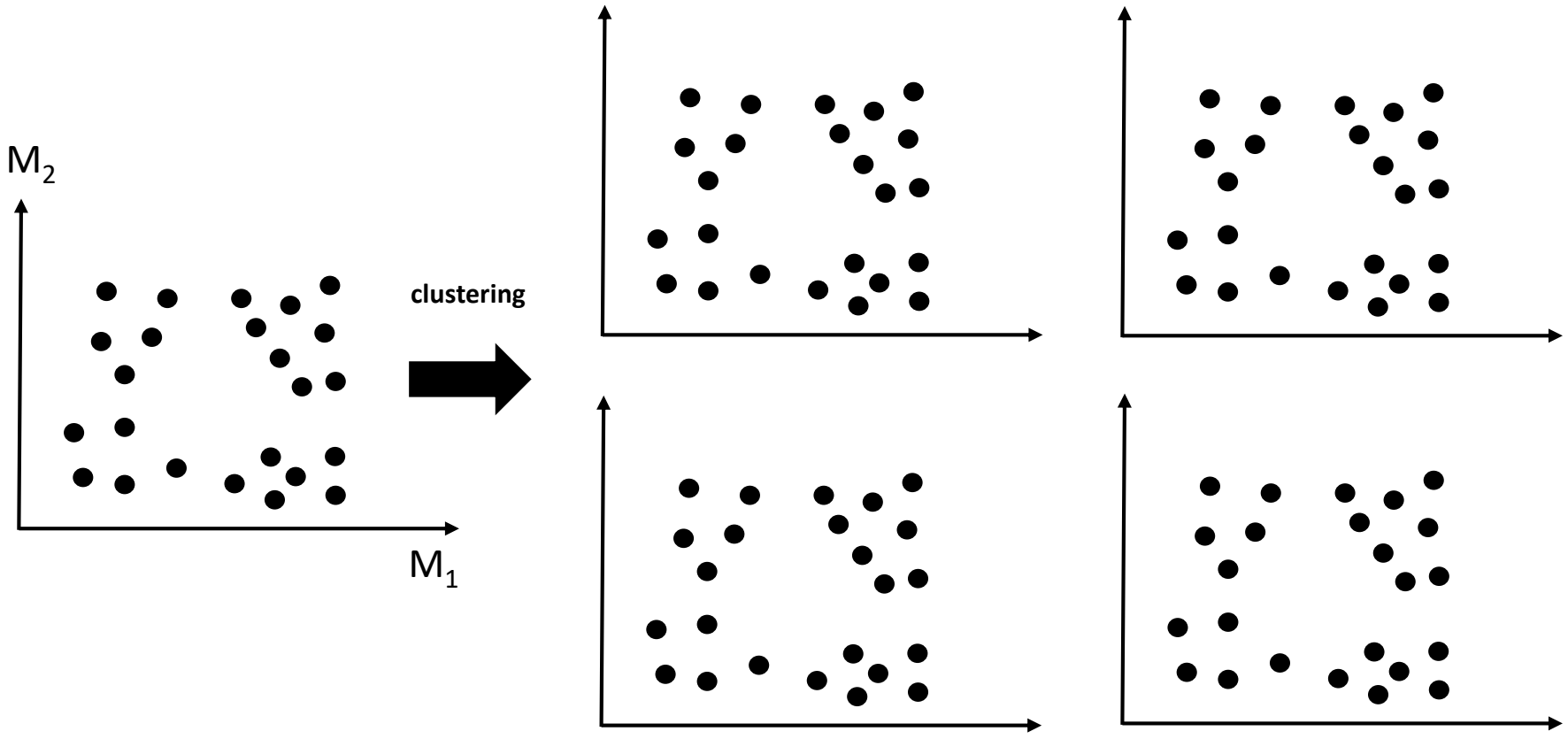
Question: which technique/method?

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

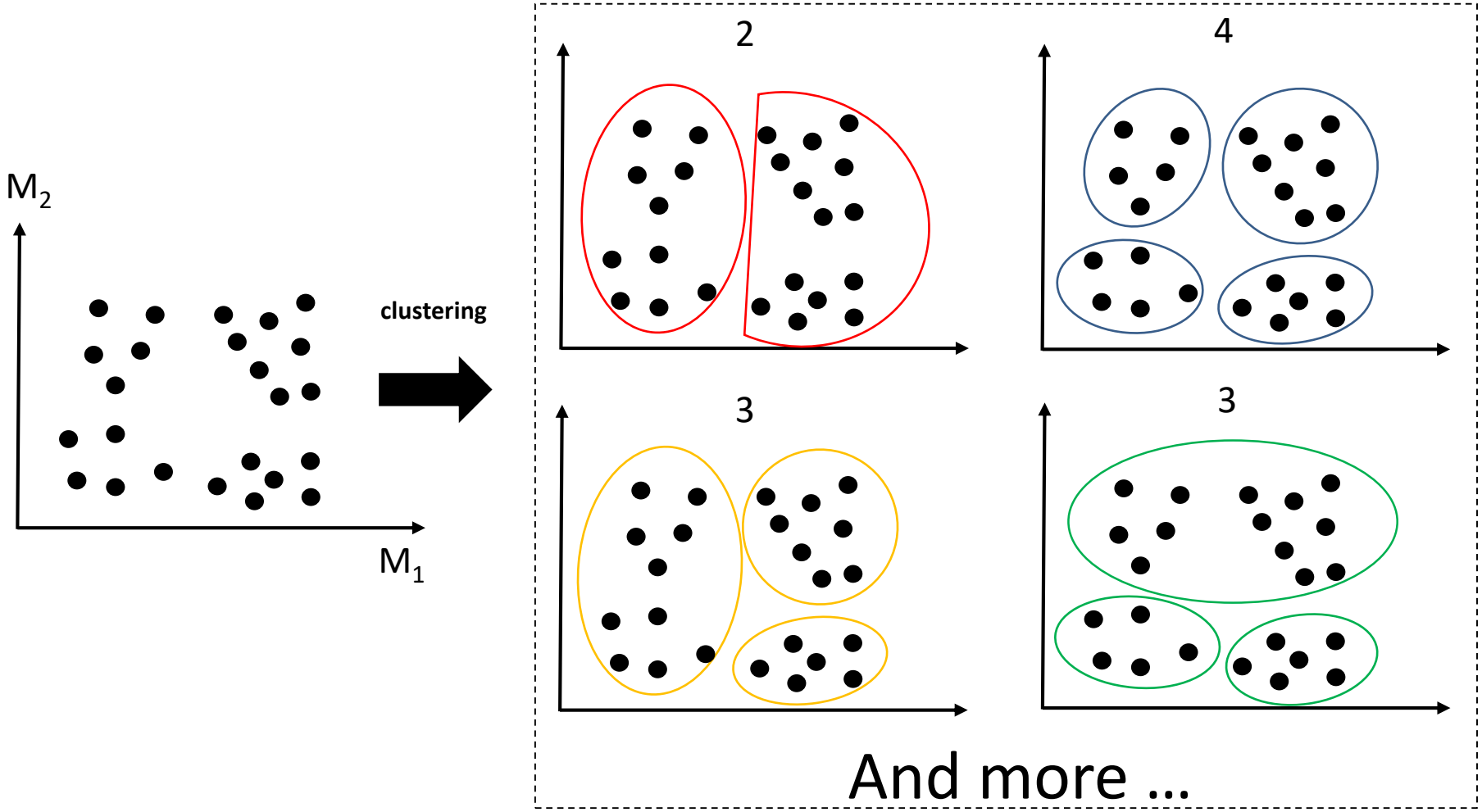


And more …

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

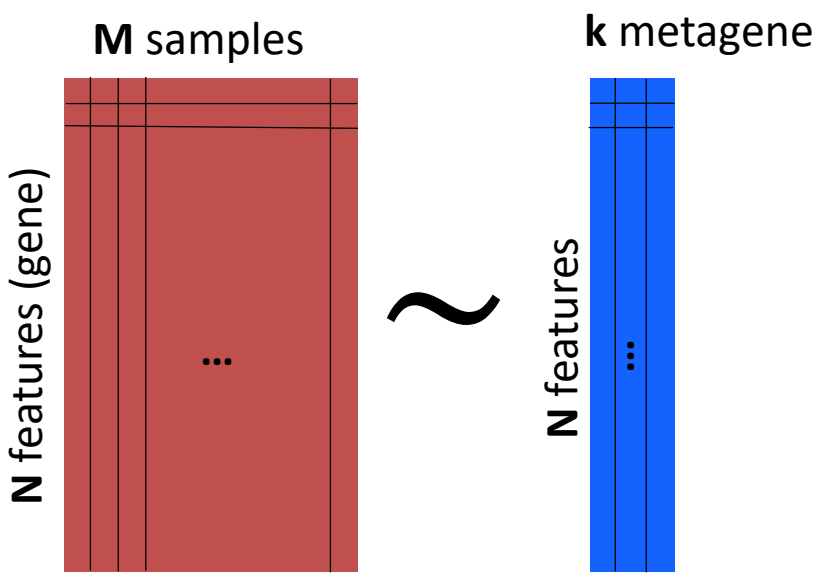
An iterative algorithm aiming at factorising an input matrix **A** into two matrices with positive entries.

$$A \sim WH$$

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

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

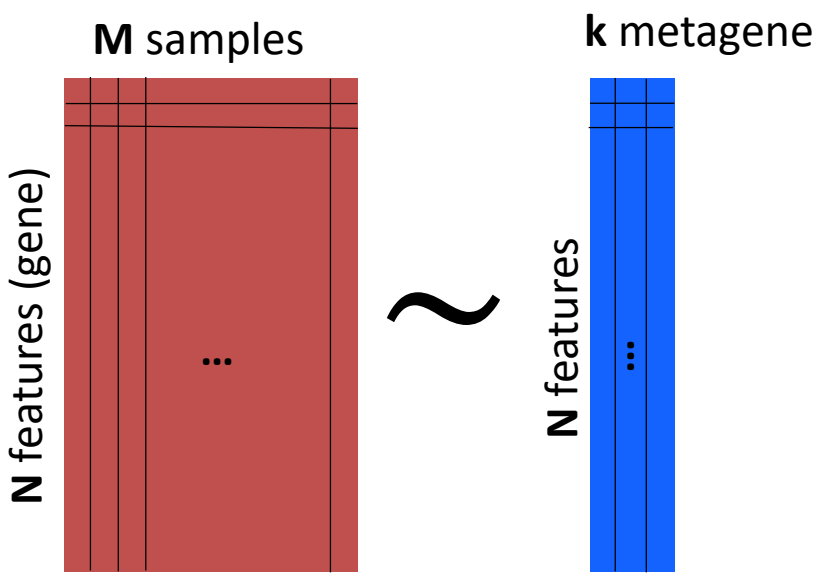
Factorising matrix A into two matrices with positive entries.

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

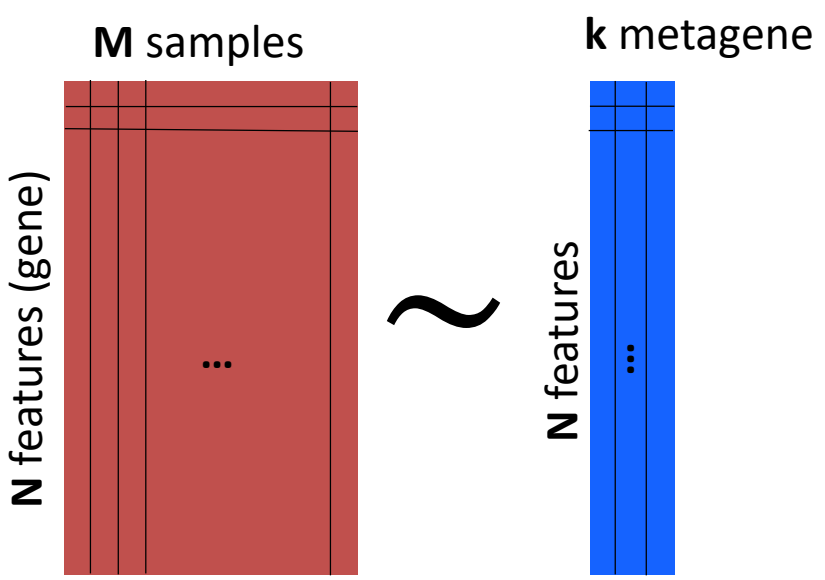
Factorising matrix A into two matrices with positive entries.

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

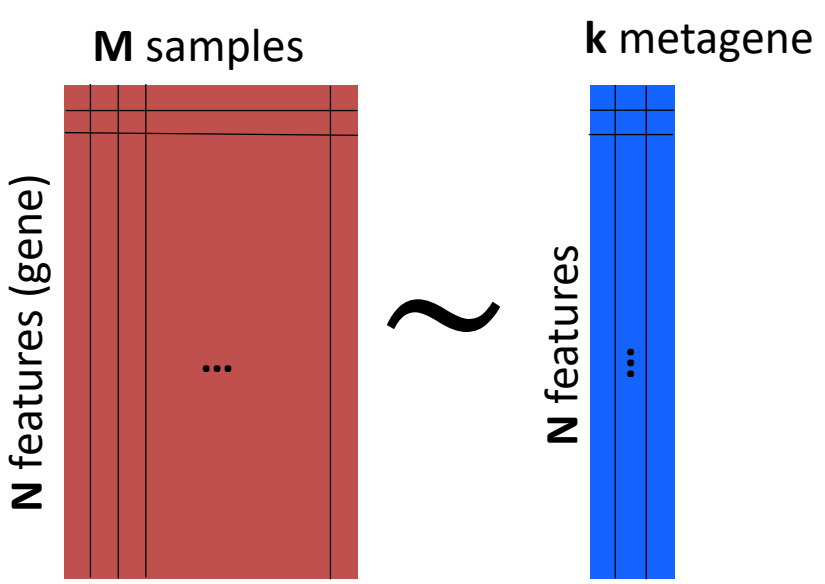
Factorising matrix A into two matrices with positive entries.

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



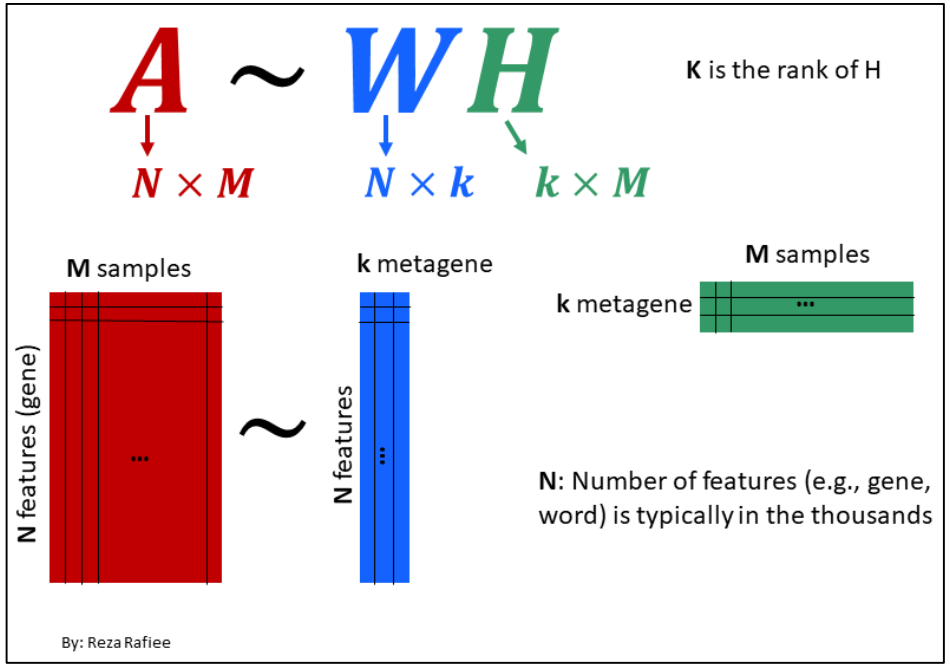
N: Number of features (e.g., gene, word) is typically in the thousands

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.

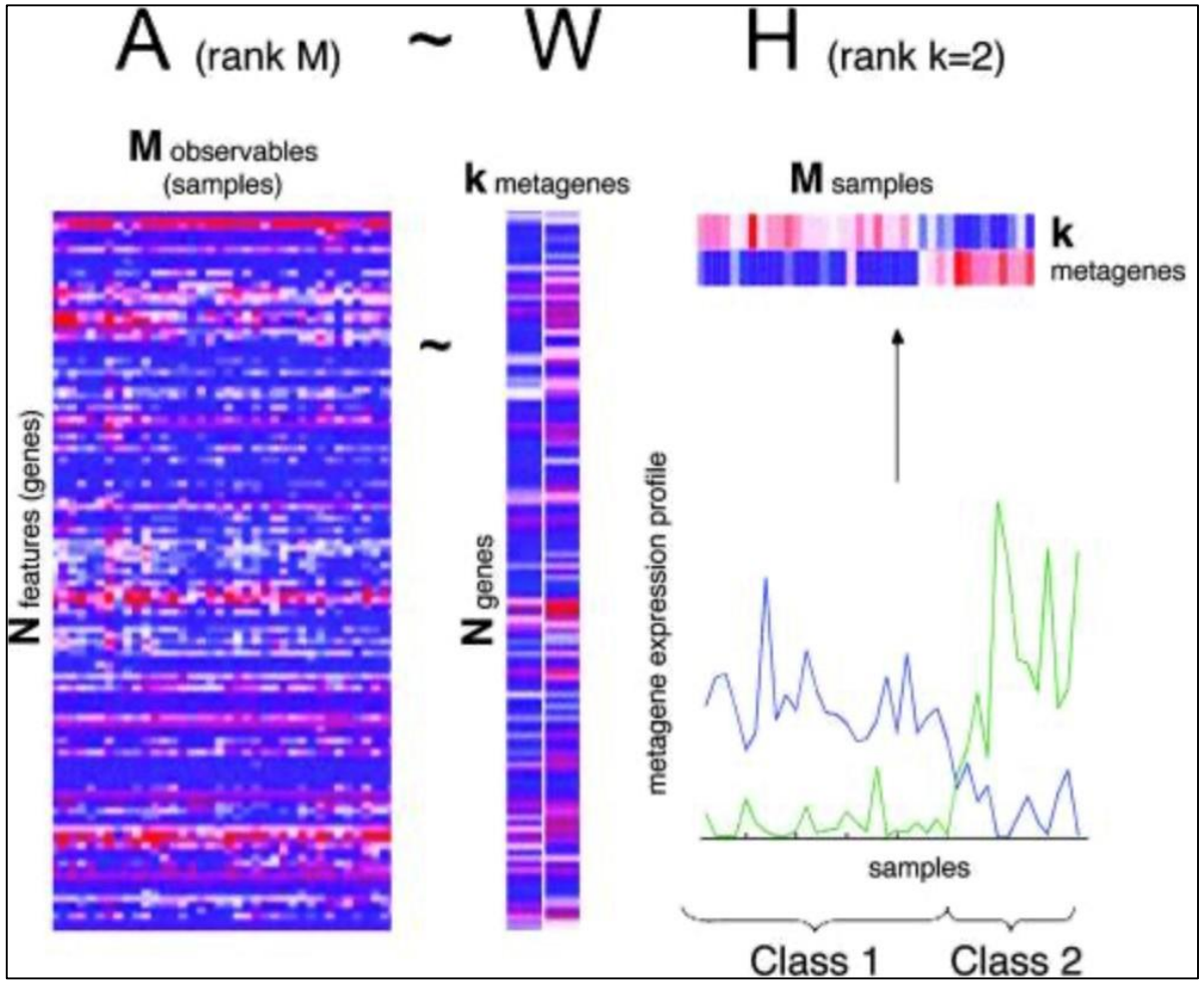


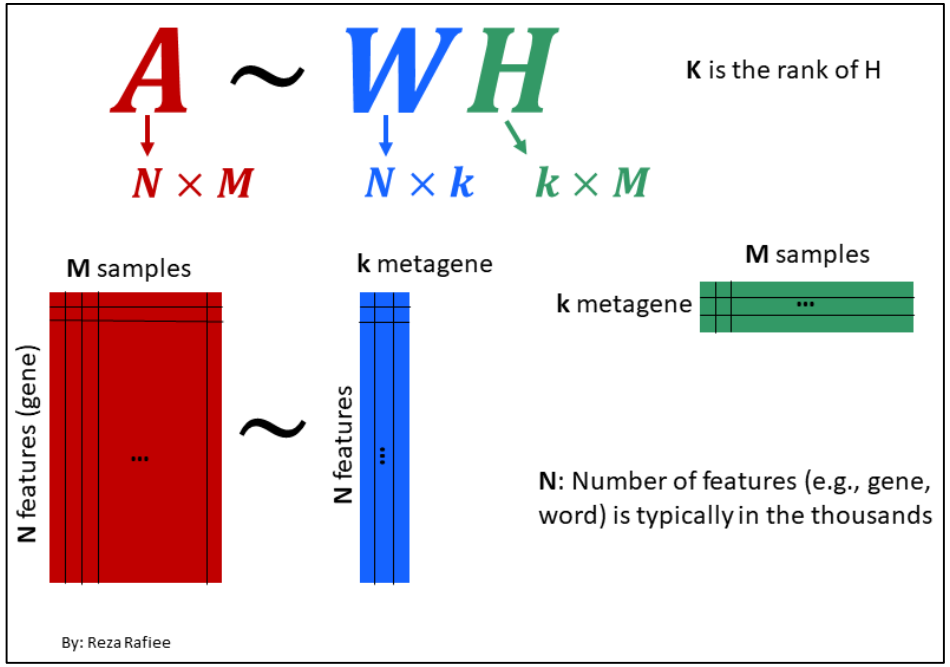
[+]Metagenes and molecular pattern discovery using matrix factorization, J. Brunet et al., PNAS, 2004

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

Assume a dataset with 17×220 size

Factorising matrix A into two matrices with positive entries

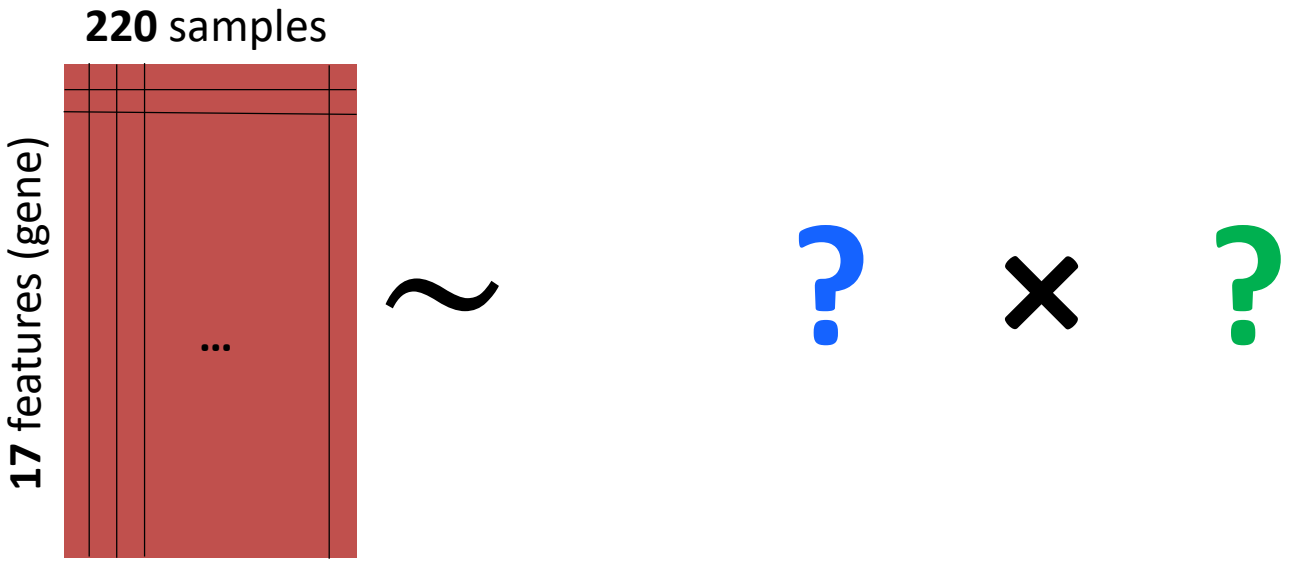
What is the size (dimension) of matrix W and H?

$$A \sim WH$$

17×220 $\quad ?\times? \quad$ $k \times ?$

N=17: Number of features
M=220: Number of samples

k ?



What is the best NMF rank?

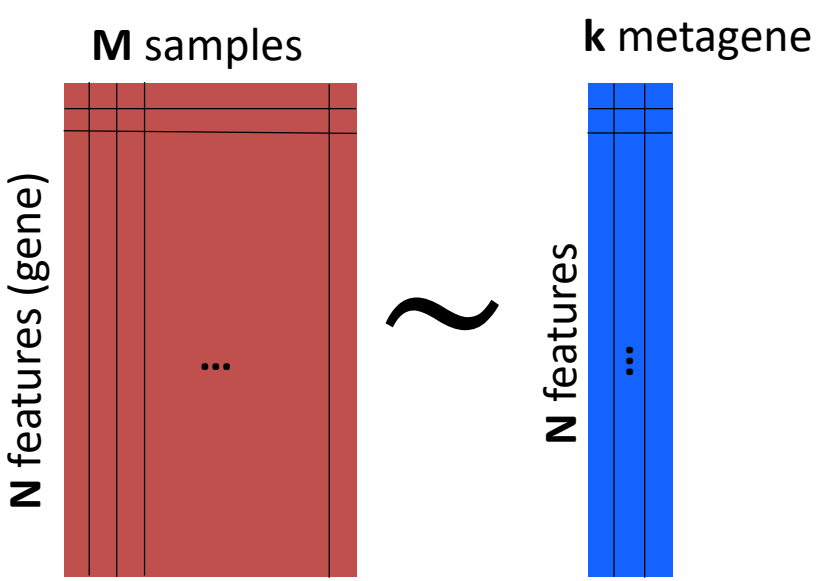
Factorising matrix A into two matrices with positive entries

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

$$A \sim WH$$

$$17 \times 220 \qquad 17 \times k \qquad k \times 220$$

k is the rank of H

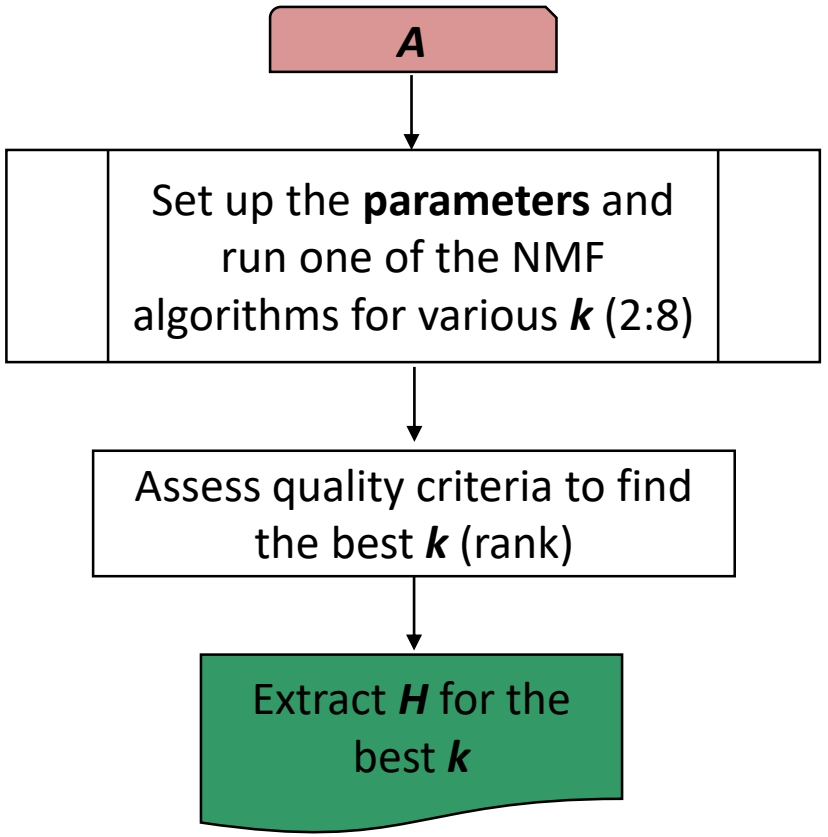


N=17: Number of features
M=220: Number of samples

What is the best NMF rank?

Factorising matrix A into two matrices with positive entries

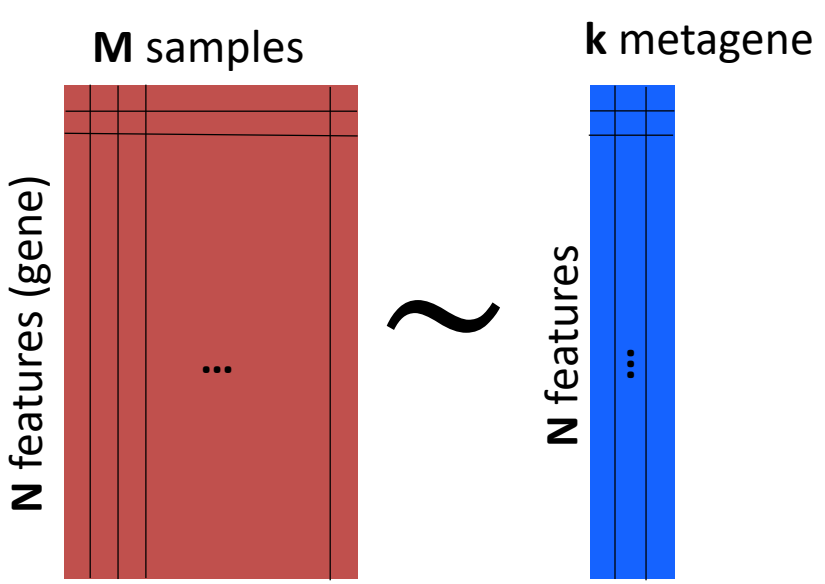
For **any rank k**, the NMF algorithm **groups** the samples into clusters.



$$A \sim WH$$

17×220 $\quad 17 \times k \quad k \times 220$

k is the rank of H



N=17: Number of features
M=220: Number of samples

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**).

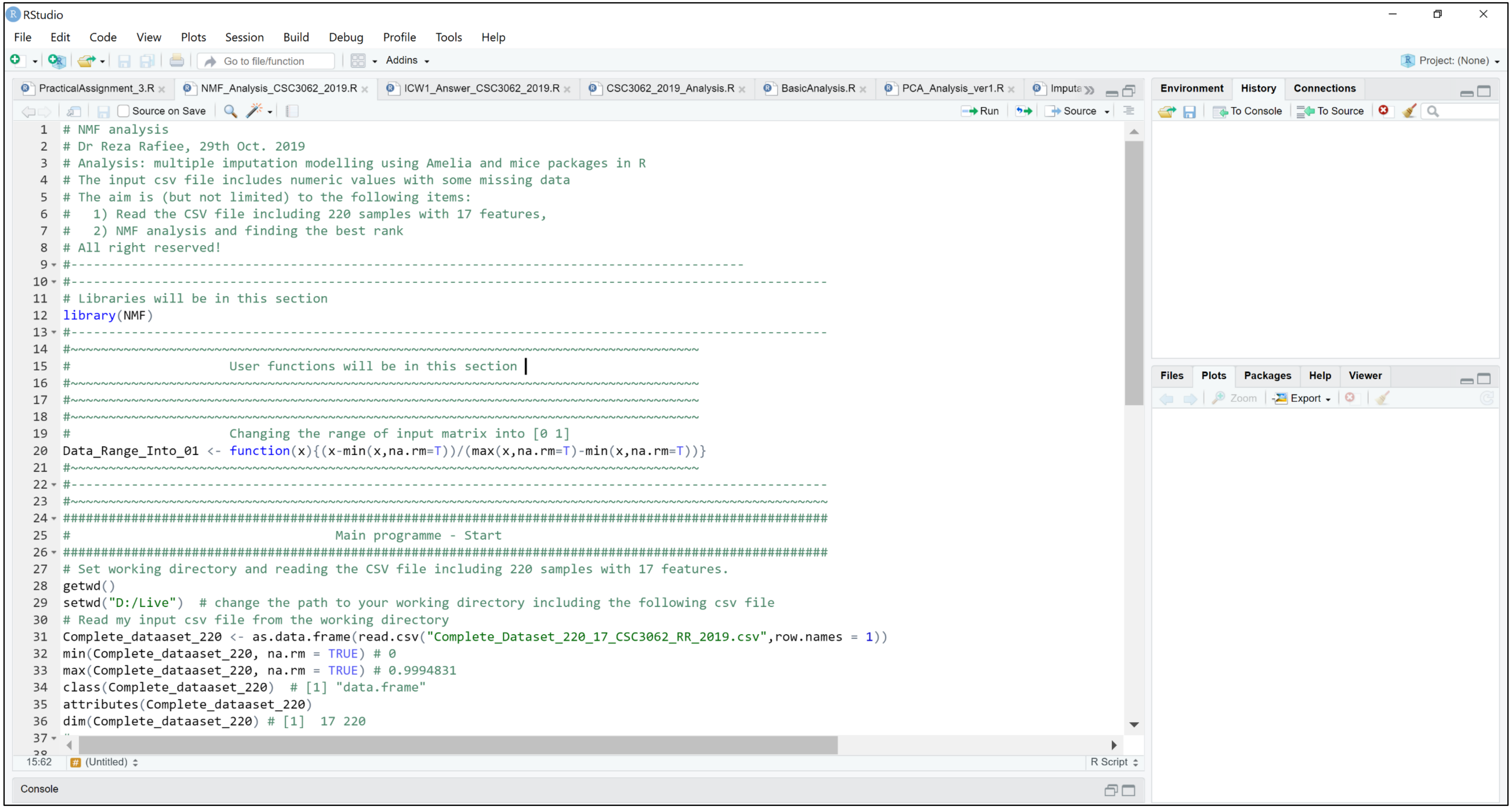
`seed` is the seeding method used to compute the starting point. The default method is given by the package specific option `'default.seed'`, which defaults to `'random'` on initialization (see method `?rnmf` for details on its implementation).

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

QUEEN'S UNIVERSITY BELFAST

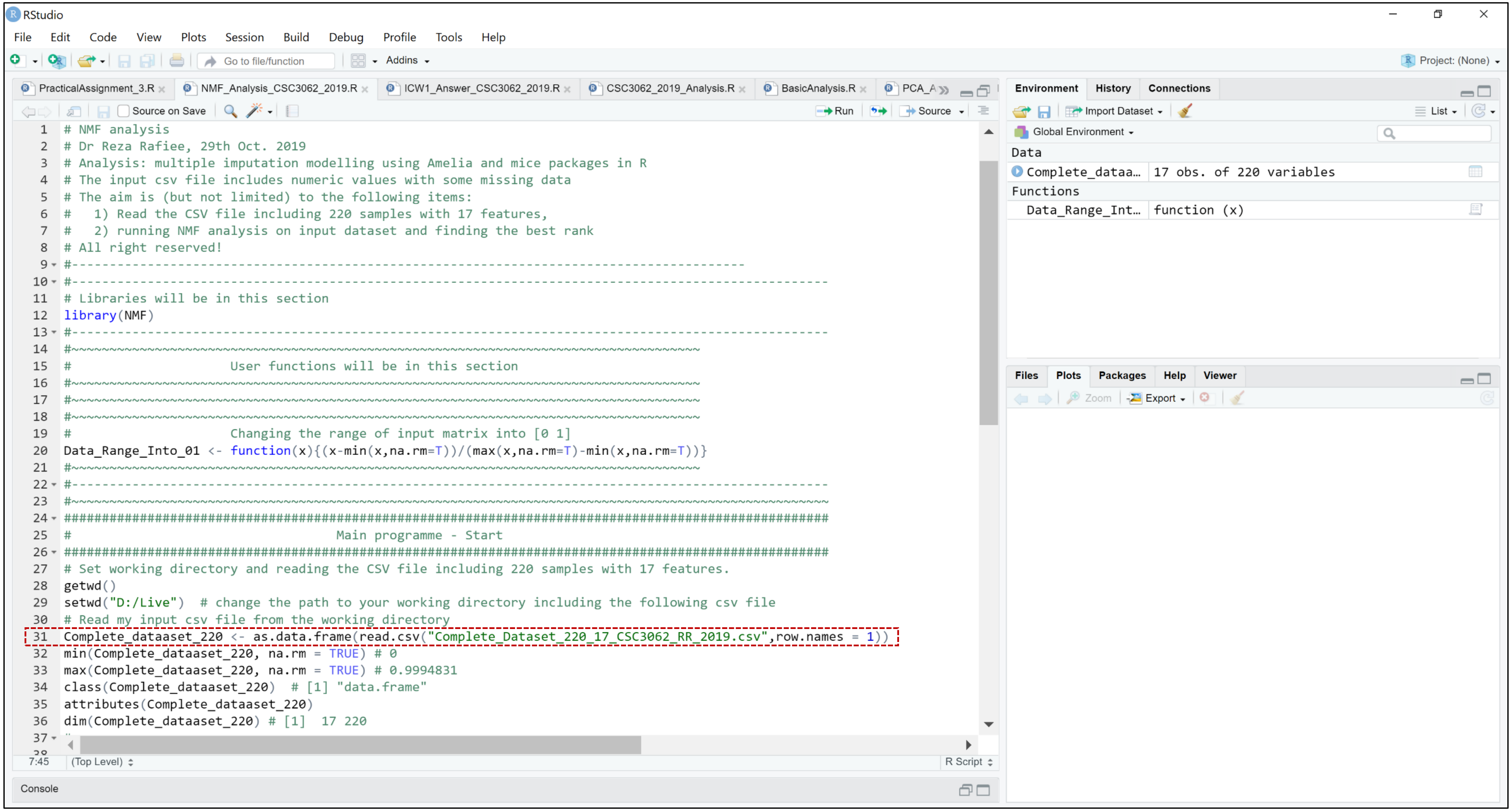
SCHOOL OF ELECTRONICS, ELECTRICAL ENGNIEERING AND COMPUTER SCIENCE

How to run NMF package in R




```
# NMF analysis
# Dr Reza Rafiee, 29th Oct. 2019
# Analysis: multiple imputation modelling using Amelia and mice packages in R
# The input csv file includes numeric values with some missing data
# The aim is (but not limited) to the following items:
#   1) Read the CSV file including 220 samples with 17 features,
#   2) NMF analysis and finding the best rank
# All right reserved!
#-----------------------------------------------------------------------------------------
#----------------------------------------------------------------------------------------------------
# Libraries will be in this section
library(NMF)
#----------------------------------------------------------------------------------------------------
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#                      User functions will be in this section
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#                      Changing the range of input matrix into [0 1]
Data_Range_Into_01 <- function(x){(x-min(x,na.rm=T))/(max(x,na.rm=T)-min(x,na.rm=T))}
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#----------------------------------------------------------------------------------------------------
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
####################################################################################################
#                                    Main programme - Start
####################################################################################################
# Set working directory and reading the CSV file including 220 samples with 17 features.
getwd()
setwd("D:/Live")  # change the path to your working directory including the following csv file
# Read my input csv file from the working directory
Complete_dataaset_220 <- as.data.frame(read.csv("Complete_Dataset_220_17_CSC3062_RR_2019.csv",row.names = 1))
min(Complete_dataaset_220, na.rm = TRUE) # 0
max(Complete_dataaset_220, na.rm = TRUE) # 0.9994831
class(Complete_dataaset_220)  # [1] "data.frame"
attributes(Complete_dataaset_220)
dim(Complete_dataaset_220) # [1]  17 220
```

How to run NMF package in R



QUEEN'S UNIVERSITY BELFAST

SCHOOL OF ELECTRONICS, ELECTRICAL ENGNIEERING AND COMPUTER SCIENCE

How to run NMF package in R

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Project: (None)

PracticalAssignment_3.R | NMF_Analysis_CSC3062_2019.R | ICW1_Answer_CSC3062_2019.R | CSC3062_2019_Analysis.R | BasicAnalysis.R | PCA_A

Source on Save | Run | Source

```
36 dim(Complete_dataaset_220) # [1]  17 220
37 #-------------------------------------------------------------------------------------------------
38 #                    First analysis: running NMF on complete dataset
39 #-------------------------------------------------------------------------------------------------
40 seq1 <- 2    # initial rank
41 seq2 <- 8    # final rank, this parameter should be carefully seleceted
42 noofrun <- 20  # the number of run
43 initseed <- 123456
44 usedmethod <- "ns"
45 Res_MultiRank <- nmf(Complete_dataaset_220, seq(seq1,seq2),
46                      method=usedmethod, nrun=noofrun,
47                      seed=initseed, .options = "t") # nsNMF is the best algorithm in terms of minimum residual errors
48 plot(Res_MultiRank)
49
50 # which rank is the best? check the NMF rank survey and assess the cophenetic score for different ranks
51 # The proper factorization rank should be selected where
52 # the magnitude of the cophenetic correlation coefficient begins to fall.
53 length(Res_MultiRank$measures$rank) # number of rank used in nmf function
54 cophen_max <- max(Res_MultiRank$measures$cophenetic)
55 for (i in 1:length(Res_MultiRank$measures$rank))
56 {
57   if (Res_MultiRank$measures$cophenetic[i] == cophen_max)
58   {
59     idx_cophen_max <- i
60   }
61 }
62
63 # assess the silhouette!
64 # Res_MultiRank$measures$silhouette.consensus
65
66 NMFfitClass <- Res_MultiRank$fit[[idx_cophen_max]]
67 H_matrix <- NMFfitClass@fit@H
68 W_matrix <- NMFfitClass@fit@W
69
70 filename_NMF_pdf <- paste("NMF_Rank_",idx_cophen_max+1,"Metagenes",nrow(Complete_dataaset_220),"genes_",ncol(Complete_dat
71 pdf(filename_NMF_pdf)
72
```

63:3 (Untitled) R Script

Console

Environment | History | Connections

Import Dataset | List

Global Environment

Data	
Complete_dataa…	17 obs. of 220 variables
Res_MultiRank	List of 3
Values	
initseed	123456
noofrun	20
seq1	2
seq2	8
usedmethod	"ns"
Functions	
Data_Range_Int…	function (x)

Files | Plots | Packages | Help | Viewer

Zoom | Export

How to run NMF package in R

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

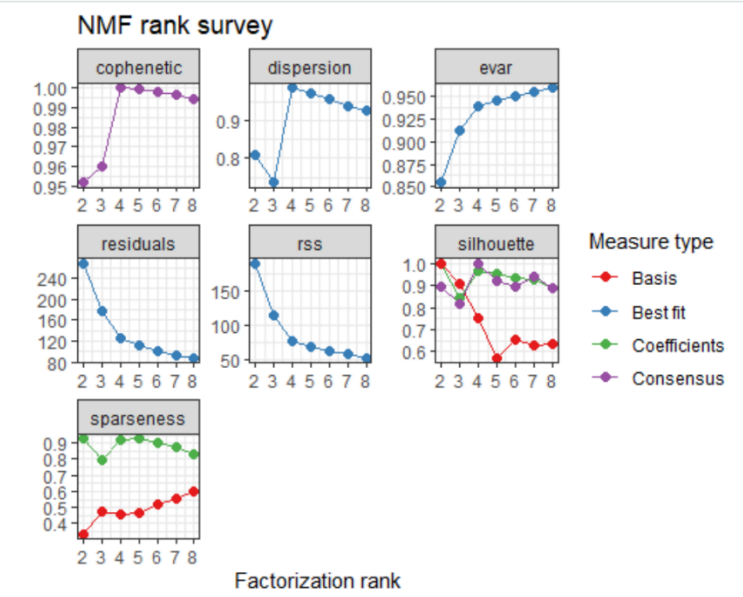
PracticalAssignment_3.R | NMF_Analysis_CSC3062_2019.R | ICW1_Answer_CSC3062_2019.R | CSC3062_2019_Analysis.R | BasicAnalysis.R | PCA_A

```
36 dim(Complete_dataaset_220) # [1]  17 220
37 #-------------------------------------------------------------------------------------------------
38 #                    First analysis: running NMF on complete dataset
39 #-------------------------------------------------------------------------------------------------
40 seq1 <- 2    # initial rank
41 seq2 <- 8    # final rank, this parameter should be carefully seleceted
42 noofrun <- 20  # the number of run
43 initseed <- 123456
44 usedmethod <- "ns"
45 Res_MultiRank <- nmf(Complete_dataaset_220, seq(seq1,seq2),
46                      method=usedmethod, nrun=noofrun,
47                      seed=initseed, .options = "t") # nsNMF is the best algorithm in terms of minimum residual errors
48 plot(Res_MultiRank)
49
50 # which rank is the best? check the NMF rank survey and assess the cophenetic score for different ranks
51 # The proper factorization rank should be selected where
52 # the magnitude of the cophenetic correlation coefficient begins to fall.
53 length(Res_MultiRank$measures$rank) # number of rank used in nmf function
54 cophen_max <- max(Res_MultiRank$measures$cophenetic)
55 for (i in 1:length(Res_MultiRank$measures$rank))
56 {
57   if (Res_MultiRank$measures$cophenetic[i] == cophen_max)
58   {
59     idx_cophen_max <- i
60   }
61 }
62
63 # assess the silhouette!
64 # Res_MultiRank$measures$silhouette.consensus
65
66 NMFfitClass <- Res_MultiRank$fit[[idx_cophen_max]]
67 H_matrix <- NMFfitClass@fit@H
68 W_matrix <- NMFfitClass@fit@W
69
70 filename_NMF_pdf <- paste("NMF_Rank_",idx_cophen_max+1,"Metagenes",nrow(Complete_dataaset_220),"genes_",ncol(Complete_dat
71 pdf(filename_NMF_pdf)
72
```

Environment | History | Connections

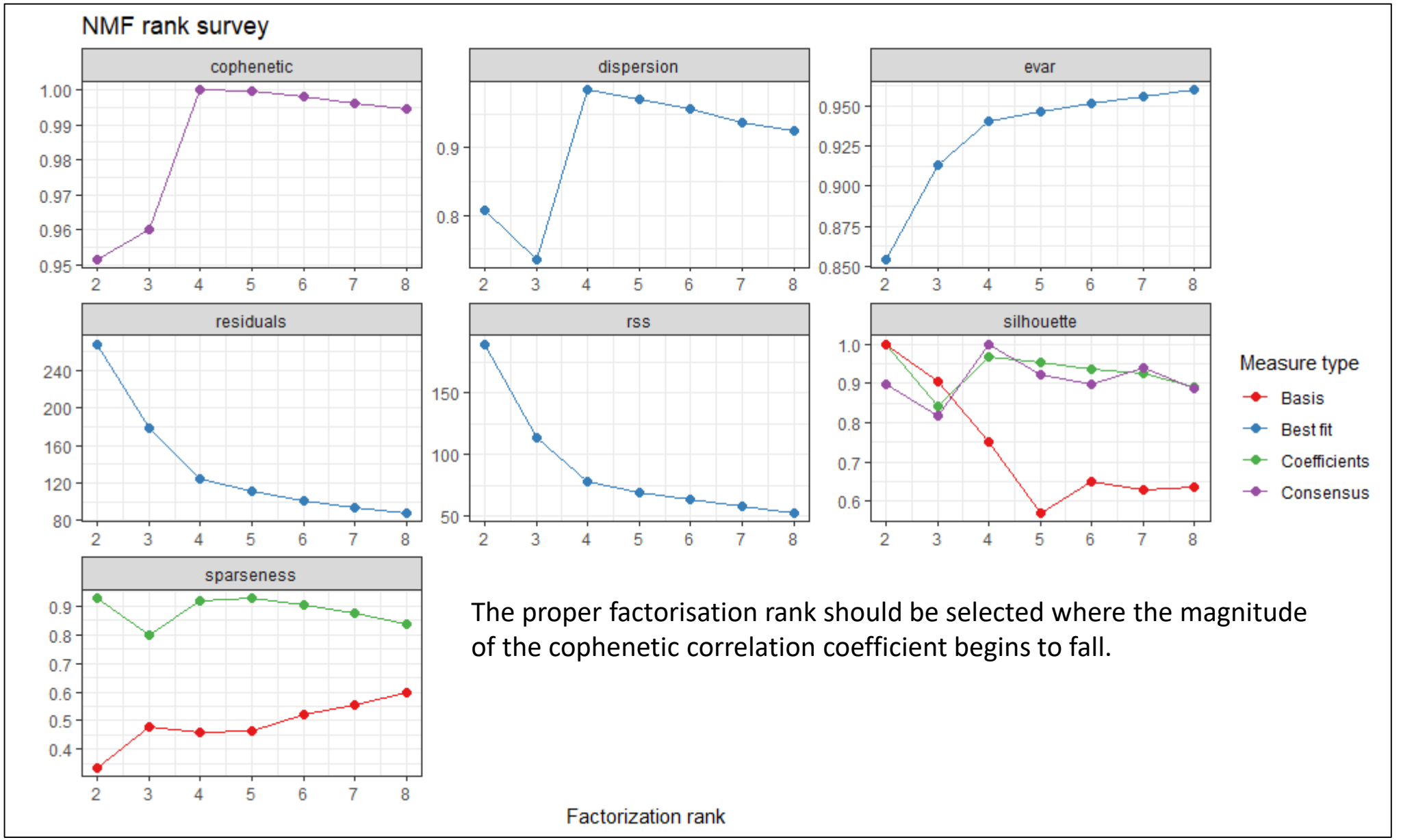
Data	
Complete_dataa…	17 obs. of 220 variables
Res_MultiRank	List of 3
Values	
initseed	123456
noofrun	20
seq1	2
seq2	8
usedmethod	"ns"
Functions	
Data_Range_Int…	function (x)

Files | Plots | Packages | Help | Viewer



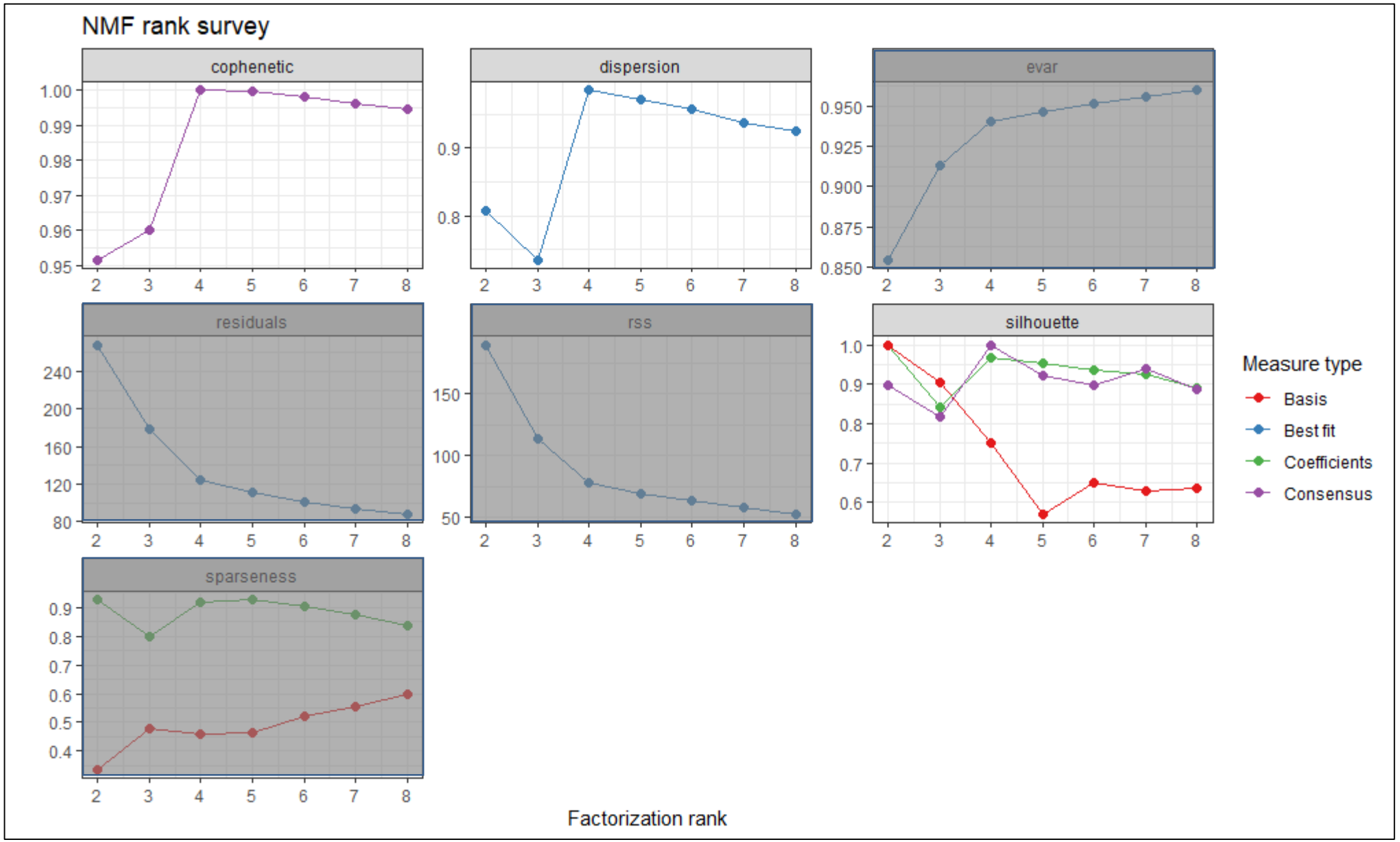
Console

NMF rank

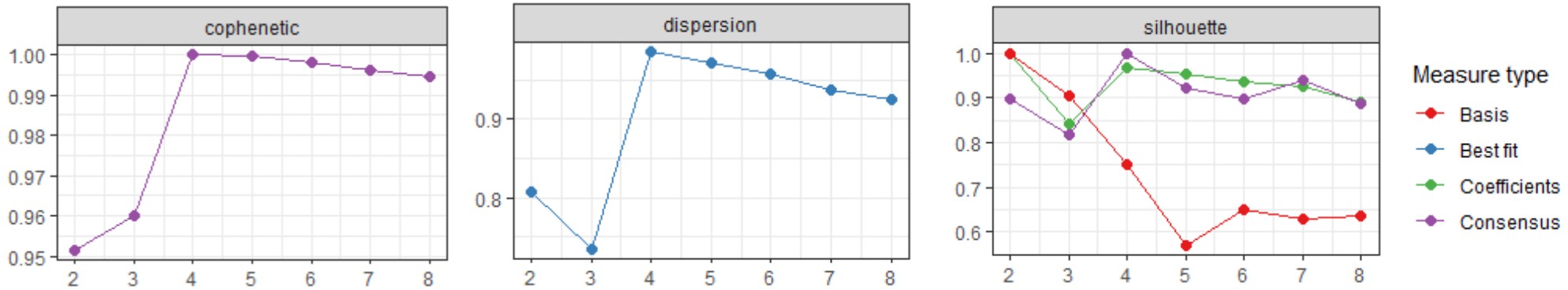


The proper factorisation rank should be selected where the magnitude of the cophenetic correlation coefficient begins to fall.

NMF rank

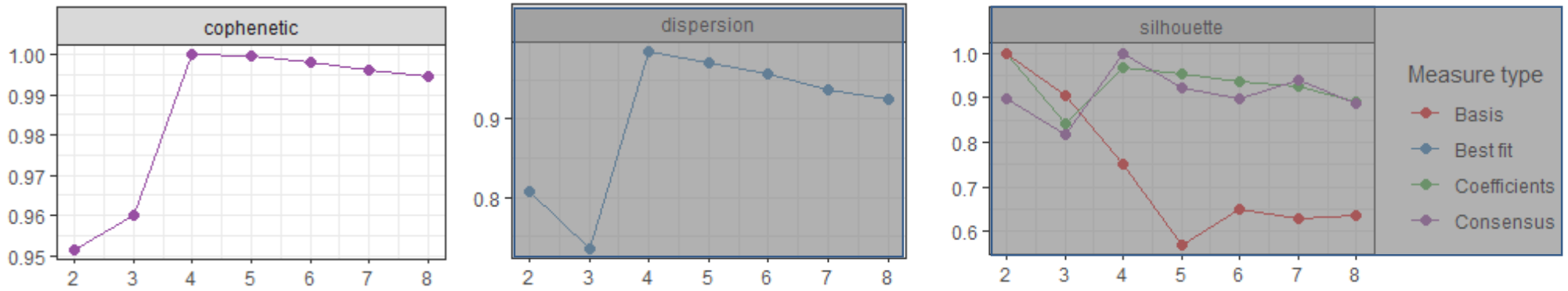

NMF rank survey
cophenetic
dispersion
evar
residuals
rss
silhouette
sparseness
Measure type
Basis
Best fit
Coefficients
Consensus
Factorization rank

How to find the best NMF rank?



A critical parameter in NMF is the factorisation **rank k**.
It defines the number of metagenes used to approximate the target matrix.
Given a NMF method and the target matrix, a common way of deciding on **k** is to try different values, compute some quality measure of the results, and choose the best value according to this quality criteria

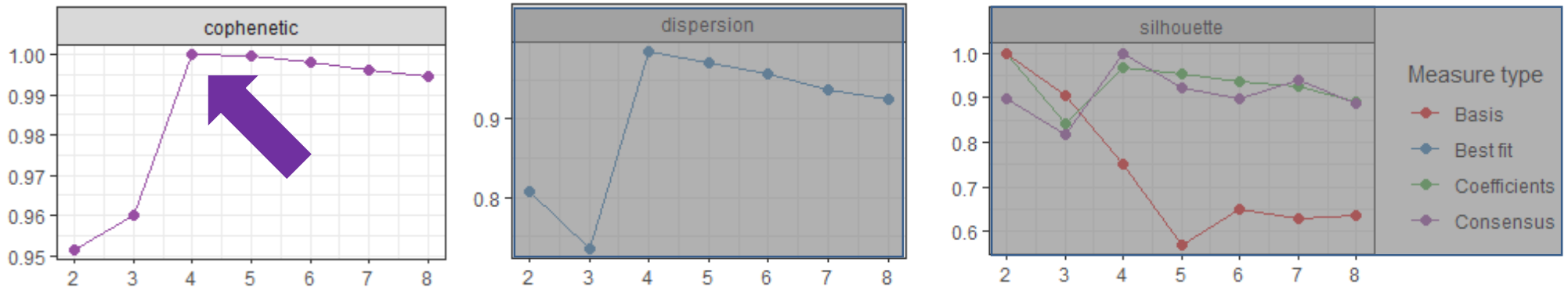
How to find the best NMF rank?



A critical parameter in NMF is the factorisation **rank k**.
It defines the number of metagenes used to approximate the target matrix.
Given a NMF method and the target matrix, a common way of deciding on **k** is to try different values, compute some quality measure of the results, and choose the best value according to this quality criteria

Several approaches have then been proposed to choose the optimal value of r. For example, (**Brunet2004**) proposed to take the first value of **k** for which **the cophenetic coefficient** starts decreasing

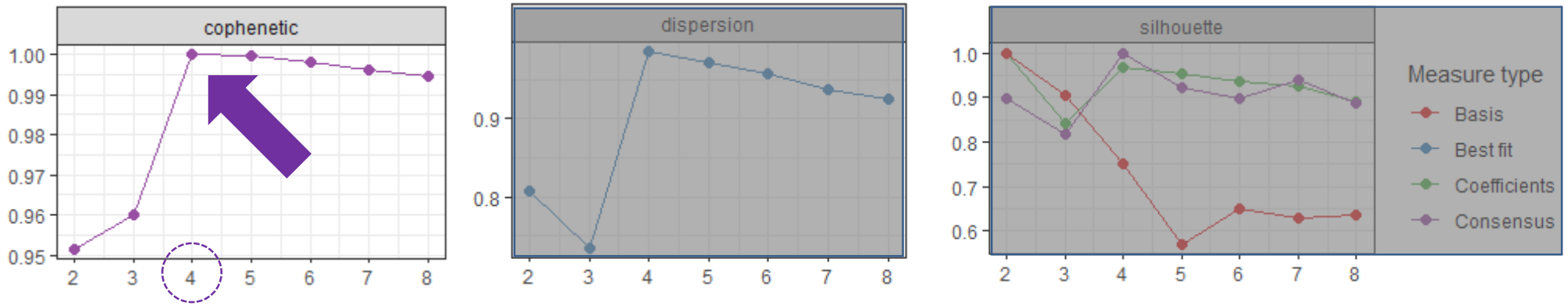
How to find the best NMF rank?



A critical parameter in NMF is the factorisation **rank k**.
It defines the number of metagenes used to approximate the target matrix.
Given a NMF method and the target matrix, a common way of deciding on k is to try different values, compute some quality measure of the results, and choose the best value according to this quality criteria

Several approaches have then been proposed to choose the optimal value of r. For example, (**Brunet2004**) proposed to take the first value of **k** for which **the cophenetic coefficient** starts decreasing

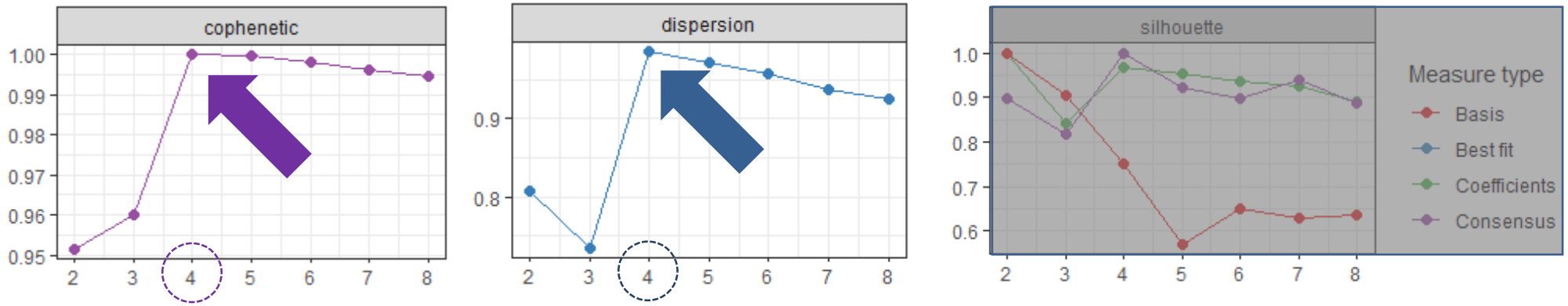
How to find the best NMF rank?



A critical parameter in NMF is the factorisation **rank k**.
It defines the number of metagenes used to approximate the target matrix.
Given a NMF method and the target matrix, a common way of deciding on k is to try different values, compute some quality measure of the results, and choose the best value according to this quality criteria

Several approaches have then been proposed to choose the optimal value of r. For example, (**Brunet2004**) proposed to take the first value of **k** for which **the cophenetic coefficient** starts decreasing

How to find the best NMF rank?

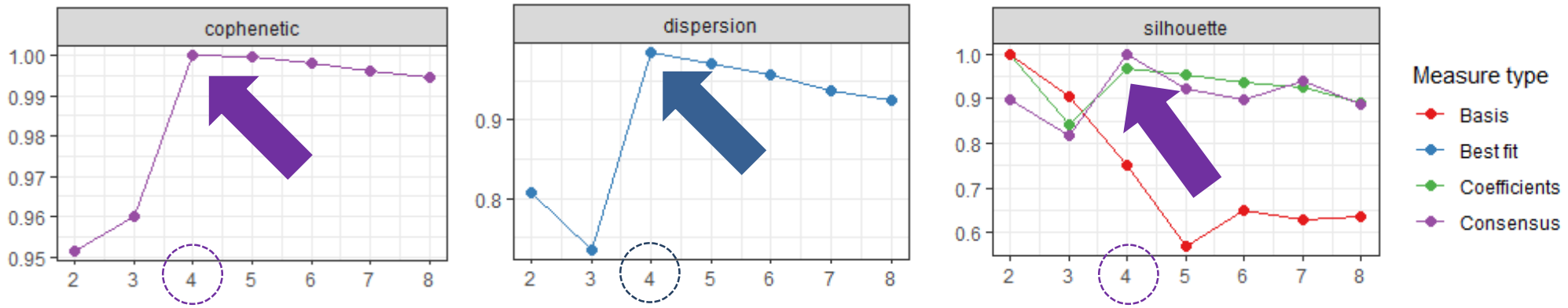


A critical parameter in NMF is the factorisation **rank k**.
It defines the number of metagenes used to approximate the target matrix.
Given a NMF method and the target matrix, a common way of deciding on r is to try different values, compute some quality measure of the results, and choose the best value according to this quality criteria

The dispersion coefficient is based on **the consensus matrix** (i.e. the average of connectivity matrices) and was proposed by Kim et al. (2007) to measure the reproducibility of the clusters obtained from NMF.

ρ: dispersion coefficient, $0 \leq \rho \leq 1$, and $\rho = 1$ only for a perfect consensus matrix, where all entries 0 or 1. A perfect consensus matrix is obtained only when all the connectivity matrices are the same, meaning that the algorithm gave the same clusters at each run. See Kim et al. (2007).

How to find the best NMF rank?



A critical parameter in NMF is the factorisation **rank k**.
It defines the number of metagenes used to approximate the target matrix.
Given a NMF method and the target matrix, a common way of deciding on r is to try different values, compute some quality measure of the results, and choose the best value according to this quality criteria

The silhouette coefficient is a measure of how similar a sample is to its own cluster (cohesion) compared to other clusters (separation). Silhouette is a method for the interpretation and **validation of consistency** within clusters of data.

The best NMF rank is k=4 in this case

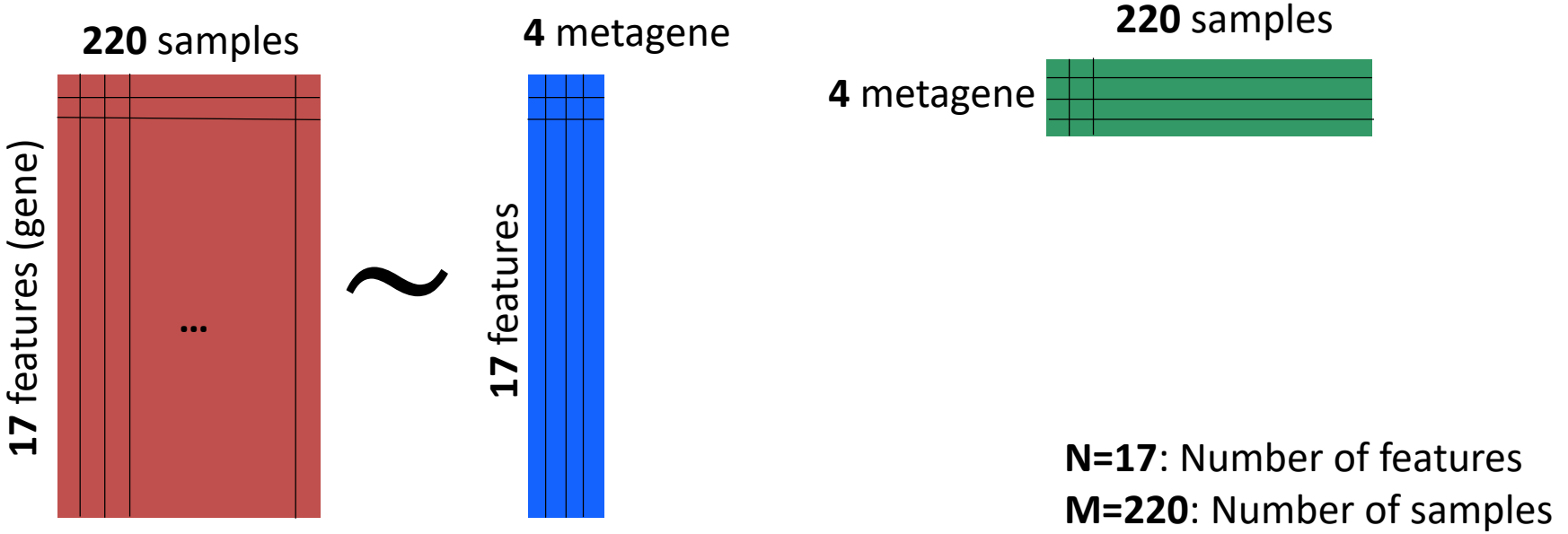
Factorising matrix A into two matrices with positive entries

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

$$A \sim WH$$

$$17 \times 220 \quad 17 \times 4 \quad 4 \times 220$$

4 is the rank of H



N=17: Number of features
M=220: Number of samples

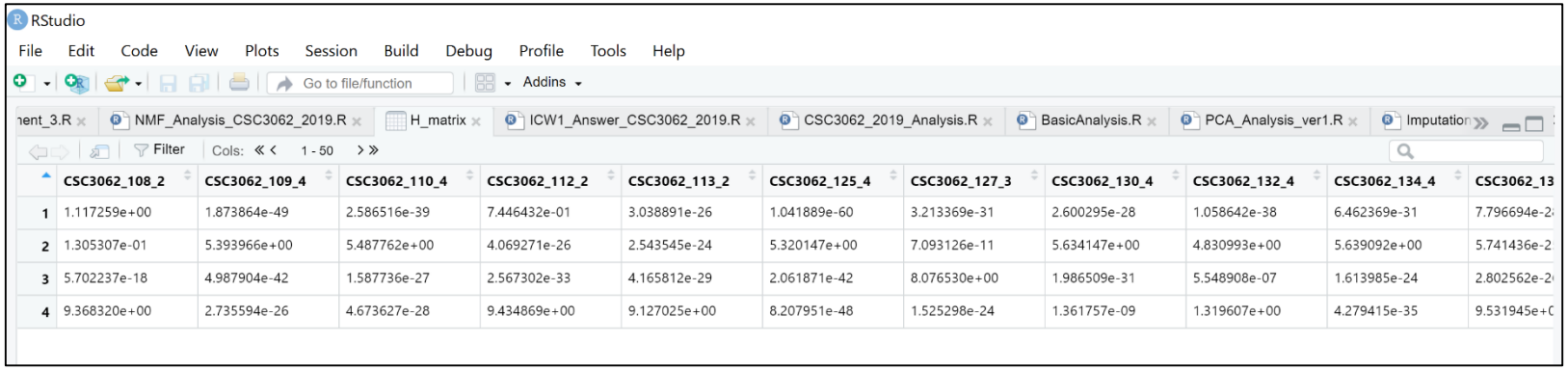
The best NMF rank is k=4 in this case

Factorising matrix A into two matrices with positive entries

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

Filter

	V1	V2	V3	V4
feature_1	4.215313e-04	2.053741e-01	2.371792e-02	2.542289e-33
feature_2	1.979353e-01	7.501003e-45	1.365094e-01	2.848400e-02
feature_3	2.509317e-15	2.094534e-01	1.129478e-01	6.675662e-02
feature_4	3.511767e-29	2.464217e-54	3.616047e-49	1.385990e-01
feature_5	5.701924e-45	1.960103e-75	1.762793e-49	1.421257e-01
feature_6	1.018935e-40	1.337246e-52	5.039218e-14	1.558415e-01
feature_7	1.088720e-01	1.359781e-14	2.143801e-62	1.140709e-01
feature_8	1.328337e-01	1.501848e-01	9.930686e-02	4.683295e-67
feature_9	1.942779e-03	2.187951e-01	1.365283e-01	9.296294e-02
feature_10	1.222782e-01	4.306796e-88	1.178640e-85	1.500447e-73
feature_11	1.676070e-01	7.097493e-53	8.175700e-02	6.885259e-02
feature_12	4.119409e-24	2.161926e-01	1.355757e-01	8.639518e-02
feature_13	5.064658e-02	2.097838e-50	1.711604e-01	1.809169e-27
feature_14	4.494635e-79	8.226998e-88	1.024966e-01	3.040853e-124
feature_15	8.504862e-02	5.409959e-226	6.944700e-236	5.458798e-255
feature_16	9.129506e-02	1.561589e-30	8.071210e-57	1.059116e-01
feature_17	4.111910e-02	4.041225e-156	4.876097e-164	5.695017e-234



	CSC3062_108_2	CSC3062_109_4	CSC3062_110_4	CSC3062_112_2	CSC3062_113_2	CSC3062_125_4	CSC3062_127_3	CSC3062_130_4	CSC3062_132_4	CSC3062_134_4	CSC3062_13
1	1.117259e+00	1.873864e-49	2.586516e-39	7.446432e-01	3.038891e-26	1.041889e-60	3.213369e-31	2.600295e-28	1.058642e-38	6.462369e-31	7.796694e-2
2	1.305307e-01	5.393966e+00	5.487762e+00	4.069271e-26	2.543545e-24	5.320147e+00	7.093126e-11	5.634147e+00	4.830993e+00	5.639092e+00	5.741436e-2
3	5.702237e-18	4.987904e-42	1.587736e-27	2.567302e-33	4.165812e-29	2.061871e-42	8.076530e+00	1.986509e-31	5.548908e-07	1.613985e-24	2.802562e-2
4	9.368320e+00	2.735594e-26	4.673627e-28	9.434869e+00	9.127025e+00	8.207951e-48	1.525298e-24	1.361757e-09	1.319607e+00	4.279415e-35	9.531945e+0

The best NMF rank is k=4 in this case

Factorising matrix A into two matrices with positive entries

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

W **(basis)**

Filter

	V1	V2	V3	V4
feature_1	4.215313e-04	2.053741e-01	2.371792e-02	2.542289e-33
feature_2	1.979353e-01	7.501003e-45	1.365094e-01	2.848400e-02
feature_3	2.509317e-15	2.094534e-01	1.129478e-01	6.675662e-02
feature_4	3.511767e-29	2.464217e-54	3.616047e-49	1.385990e-01
feature_5	5.701924e-45	1.960103e-75	1.762793e-49	1.421257e-01
feature_6	1.018935e-40	1.337246e-52	5.039218e-14	1.558415e-01
feature_7	1.088720e-01	1.359781e-14	2.143801e-62	1.140709e-01
feature_8	1.328337e-01	1.501848e-01	9.930686e-02	4.683295e-67
feature_9	1.942779e-03	2.187951e-01	1.365283e-01	9.296294e-02
feature_10	1.222782e-01	4.306796e-88	1.178640e-85	1.500447e-73
feature_11	1.676070e-01	7.097493e-53	8.175700e-02	6.885259e-02
feature_12	4.119409e-24	2.161926e-01	1.355757e-01	8.639518e-02
feature_13	5.064658e-02	2.097838e-50	1.711604e-01	1.809169e-27
feature_14	4.494635e-79	8.226998e-88	1.024966e-01	3.040853e-124
feature_15	8.504862e-02	5.409959e-226	6.944700e-236	5.458798e-255
feature_16	9.129506e-02	1.561589e-30	8.071210e-57	1.059116e-01
feature_17	4.111910e-02	4.041225e-156	4.876097e-164	5.695017e-234

H **(coef)**

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

ient_3.R | NMF_Analysis_CSC3062_2019.R | H_matrix | ICW1_Answer_CSC3062_2019.R | CSC3062_2019_Analysis.R | BasicAnalysis.R | PCA_Analysis_ver1.R | Imputation

Filter | Cols: « < 1 - 50 > »

	CSC3062_108_2	CSC3062_109_4	CSC3062_110_4	CSC3062_112_2	CSC3062_113_2	CSC3062_125_4	CSC3062_127_3	CSC3062_130_4	CSC3062_132_4	CSC3062_134_4	CSC3062_13
1	1.117259e+00	1.873864e-49	2.586516e-39	7.446432e-01	3.038891e-26	1.041889e-60	3.213369e-31	2.600295e-28	1.058642e-38	6.462369e-31	7.796694e-2
2	1.305307e-01	5.393966e+00	5.487762e+00	4.069271e-26	2.543545e-24	5.320147e+00	7.093126e-11	5.634147e+00	4.830993e+00	5.639092e+00	5.741436e-2
3	5.702237e-18	4.987904e-42	1.587736e-27	2.567302e-33	4.165812e-29	2.061871e-42	8.076530e+00	1.986509e-31	5.548908e-07	1.613985e-24	2.802562e-2
4	9.368320e+00	2.735594e-26	4.673627e-28	9.434869e+00	9.127025e+00	8.207951e-48	1.525298e-24	1.361757e-09	1.319607e+00	4.279415e-35	9.531945e+0

Consensus matrix

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

PracticalAssignment_3.R | NMF_Analysis_CSC3062_2019.R | ICW1_Answer_CSC3062_2019.R | CSC3062_2019_Analysis.R | BasicAnalysis.R | PCA_A

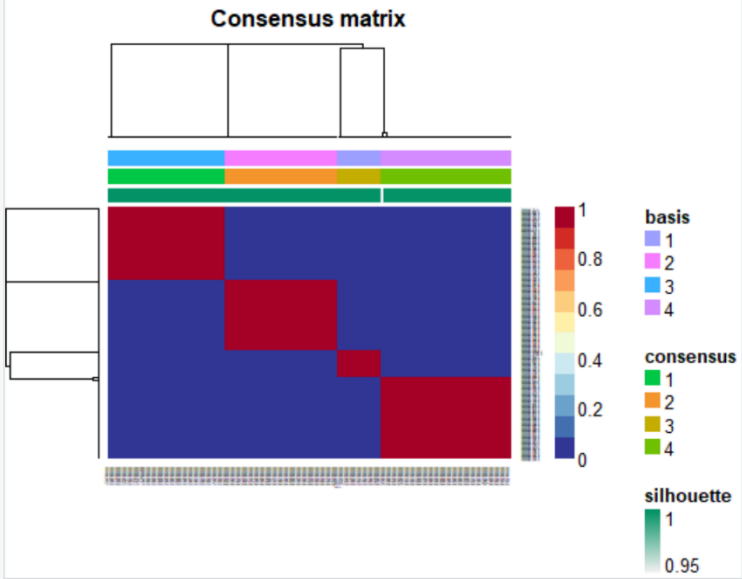
```
43 # to get a robust estimate of the factorisation rank (Brunet2004; Hutchins2008).
44 initseed <- 123456
45 usedmethod <- "ns"
46 Res_MultiRank <- nmf(Complete_dataaset_220, seq(seq1,seq2),
47                     method=usedmethod, nrun=noofrun,
48                     seed=initseed, .options = "t") # nsNMF is the best algorithm in terms of minimum residual errors
49 plot(Res_MultiRank)
50
51 # which rank is the best? check the NMF rank survey and assess the cophenetic score for different ranks
52 # The proper factorization rank should be selected where
53 # the magnitude of the cophenetic correlation coefficient begins to fall.
54 length(Res_MultiRank$measures$rank) # number of rank used in nmf function
55 cophen_max <- max(Res_MultiRank$measures$cophenetic)
56 for (i in 1:length(Res_MultiRank$measures$rank))
57 {
58   if (Res_MultiRank$measures$cophenetic[i] == cophen_max)
59   {
60     idx_cophen_max <- i
61   }
62 }
63
64 # assess the silhouette!
65 # Res_MultiRank$measures$silhouette.consensus
66
67 NMFfitClass <- Res_MultiRank$fit[[idx_cophen_max]]
68 H_matrix <- NMFfitClass@fit@H
69 W_matrix <- NMFfitClass@fit@W
70
71 consensusmap(Res_MultiRank$fit[[idx_cophen_max]])
72 filename_NMF_pdf <- paste("NMF_Rank_",idx_cophen_max+1,"Metagenes",nrow(Complete_dataaset_220),"genes_",ncol(Complete_dat
73 pdf(filename_NMF_pdf)
74 dev.off()
75 #par(mfrow=c(2,2))
76
77 plot(Res_MultiRank)
78 filename2_NMF_pdf <- paste("NMF_Consensus_",idx_cophen_max+1,"Metagenes",nrow(Complete_dataaset_220),"genes_",ncol(Comple
79 pdf(filename2_NMF_pdf)
80
```

Environment | History | Connections

Global Environment

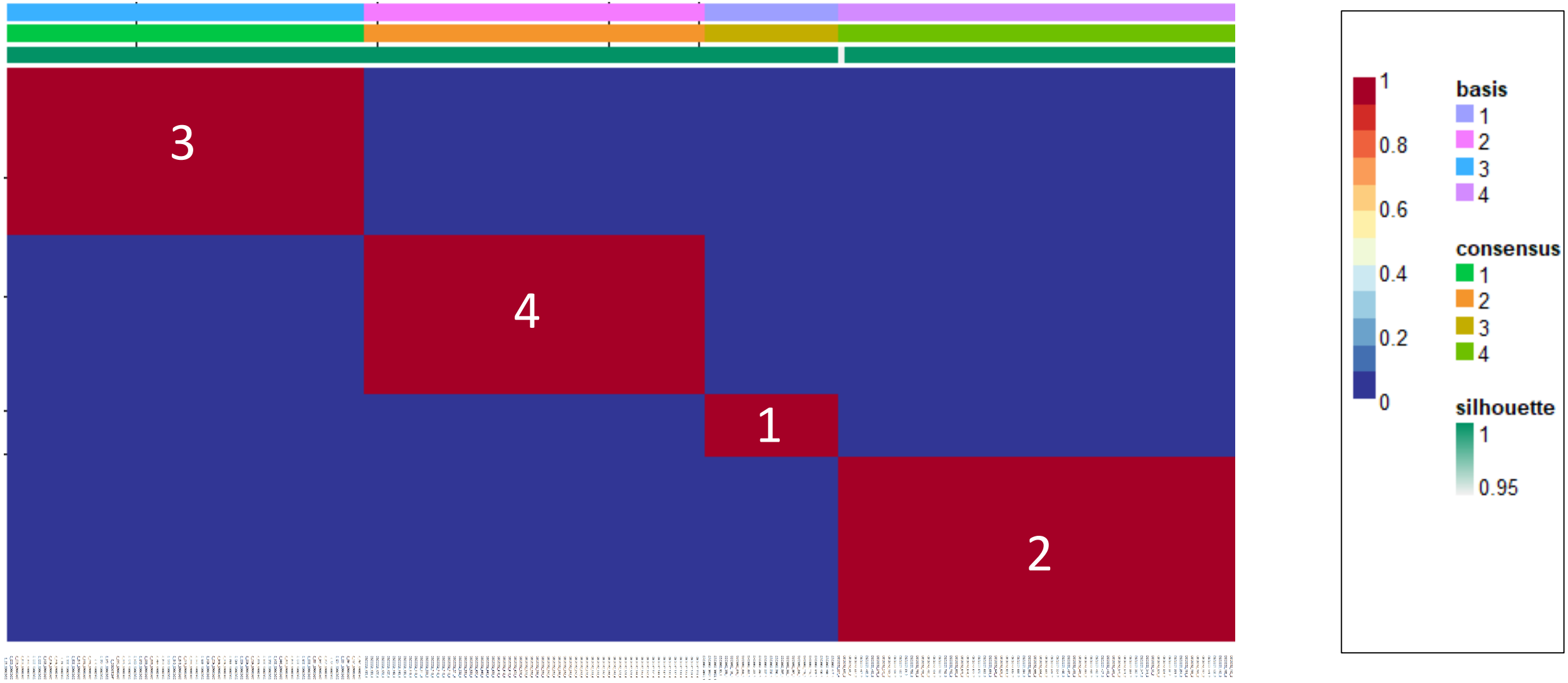
Data	
Complete_dataa…	17 obs. of 220 variables
H_matrix	num [1:4, 1:220] 1.12 1.31e-01 5.70e-18 …
NMFfitClass	Formal class NMFfitX1
Res_MultiRank	List of 3
W_matrix	num [1:17, 1:4] 4.22e-04 1.98e-01 2.51e-…
Values	
cophen_max	0.999977872267844
filename_NMF_p…	"NMF_Rank_4Metagenes17genes_220samples_NMF…
i	7L
idx_cophen_max	3L

Files | Plots | Packages | Help | Viewer



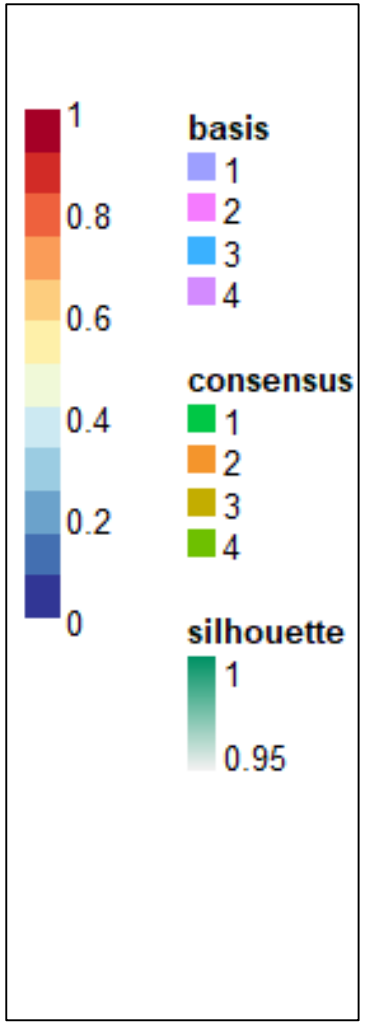
Console

Heatmap of consensus matrix

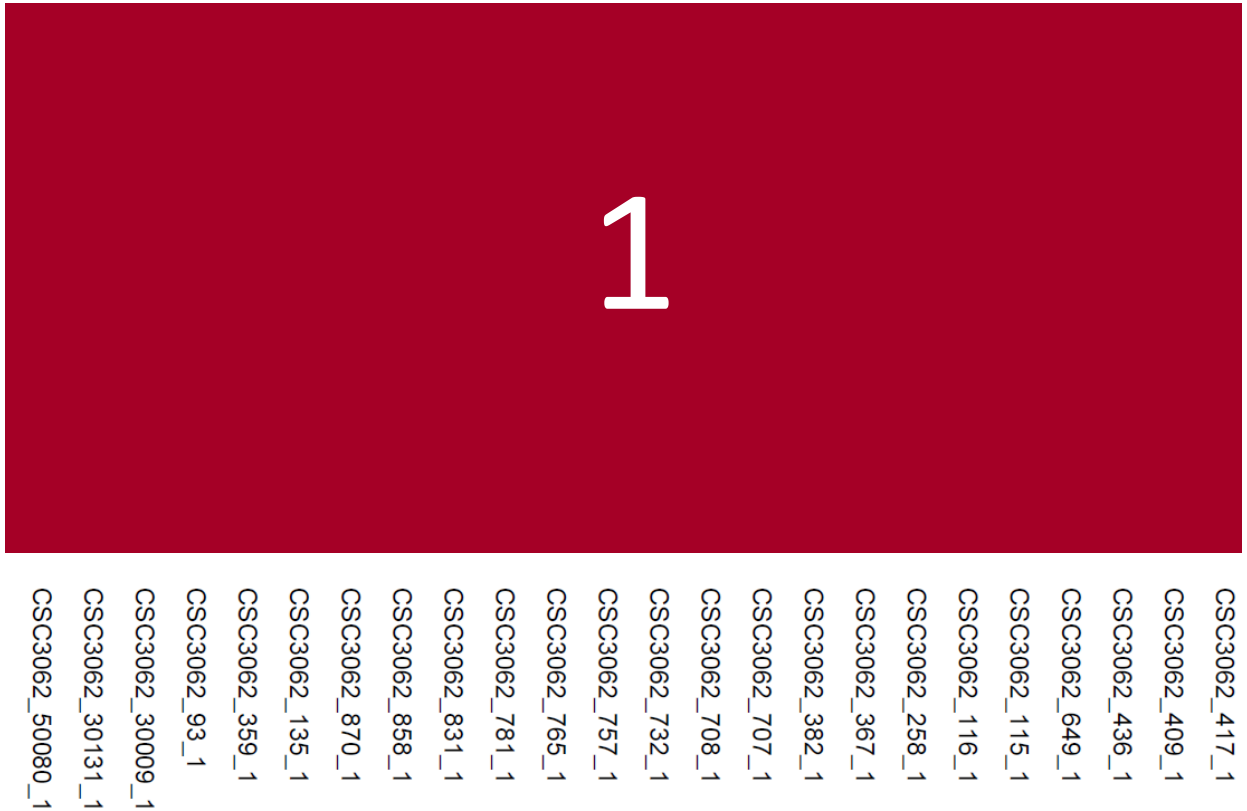


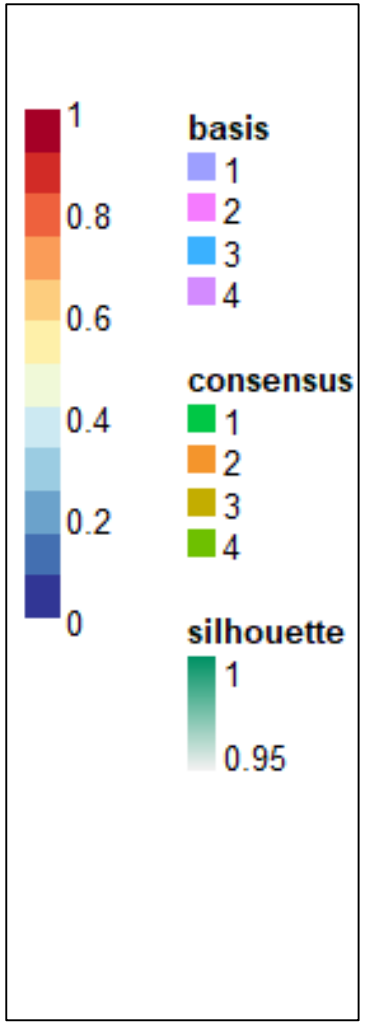
In the class discovery context, it would useful to check if the clusters obtained correspond to known classes.

Heatmap of consensus matrix

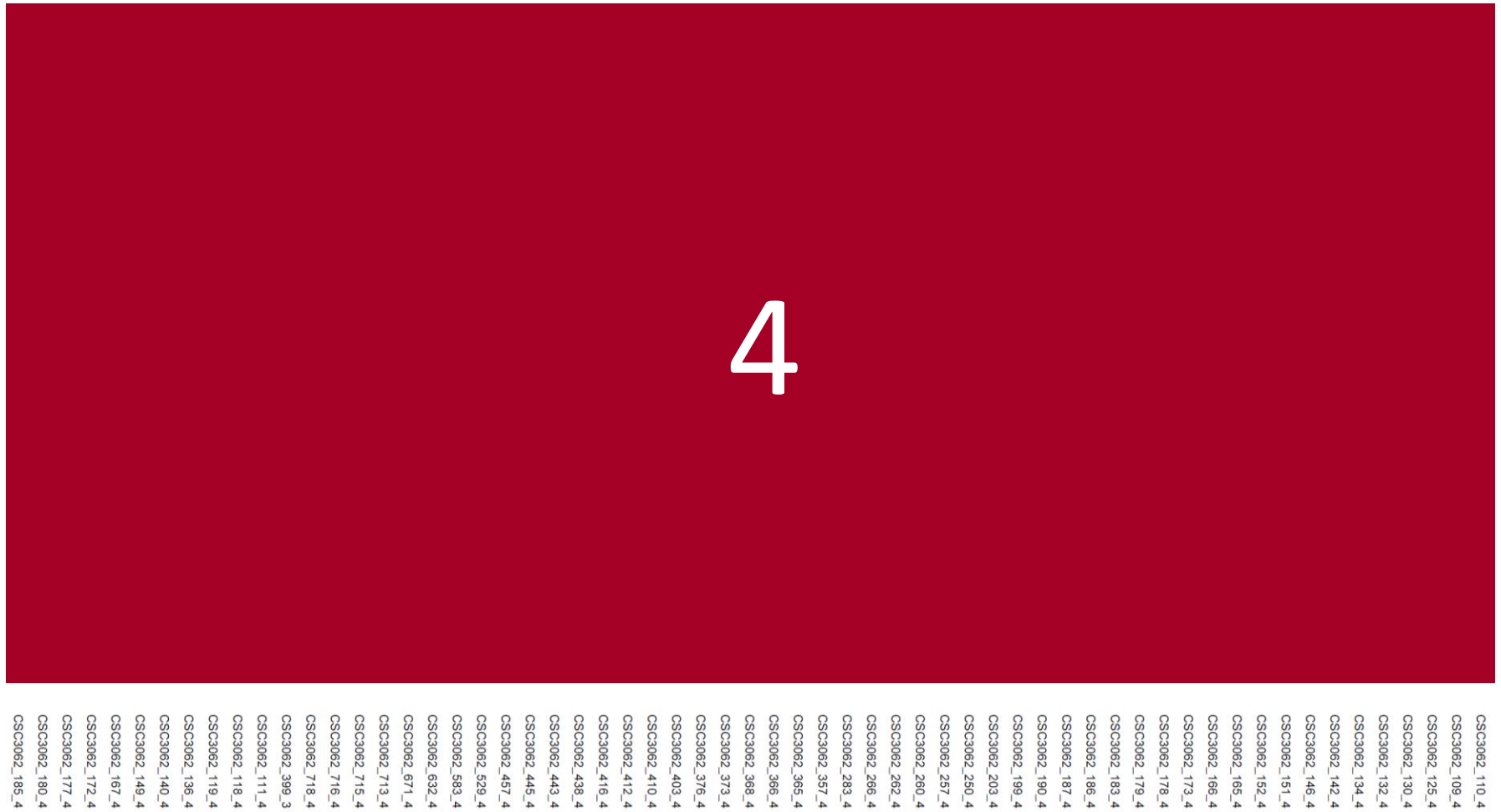


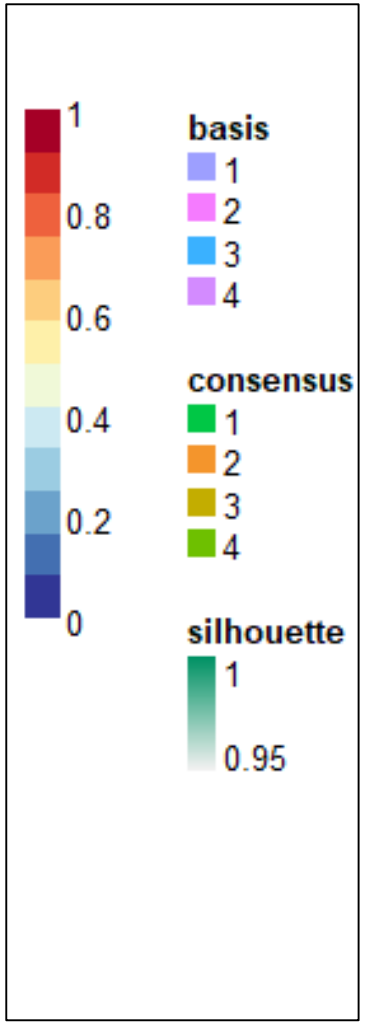
Heatmap of consensus matrix



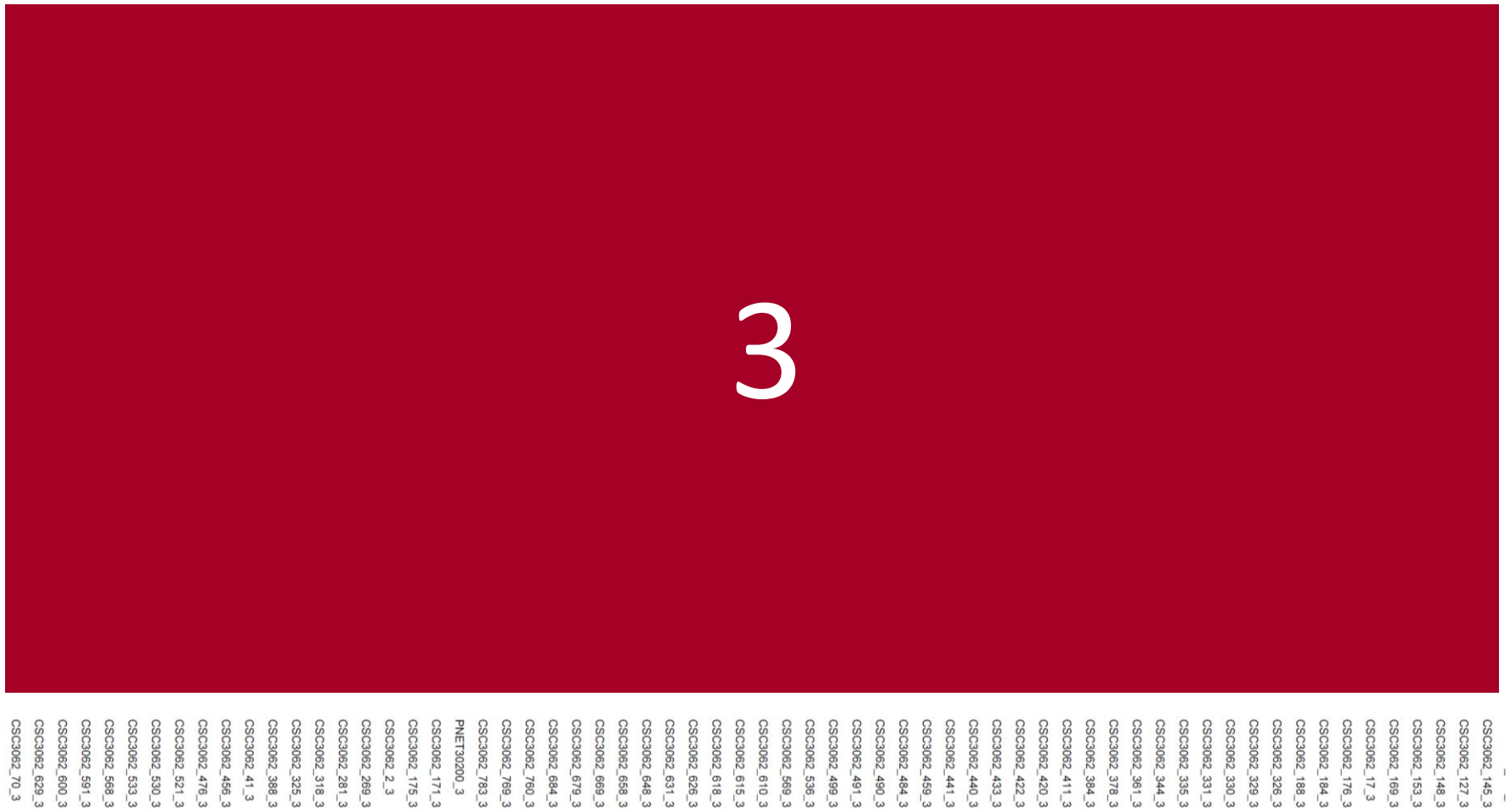


Heatmap of consensus matrix

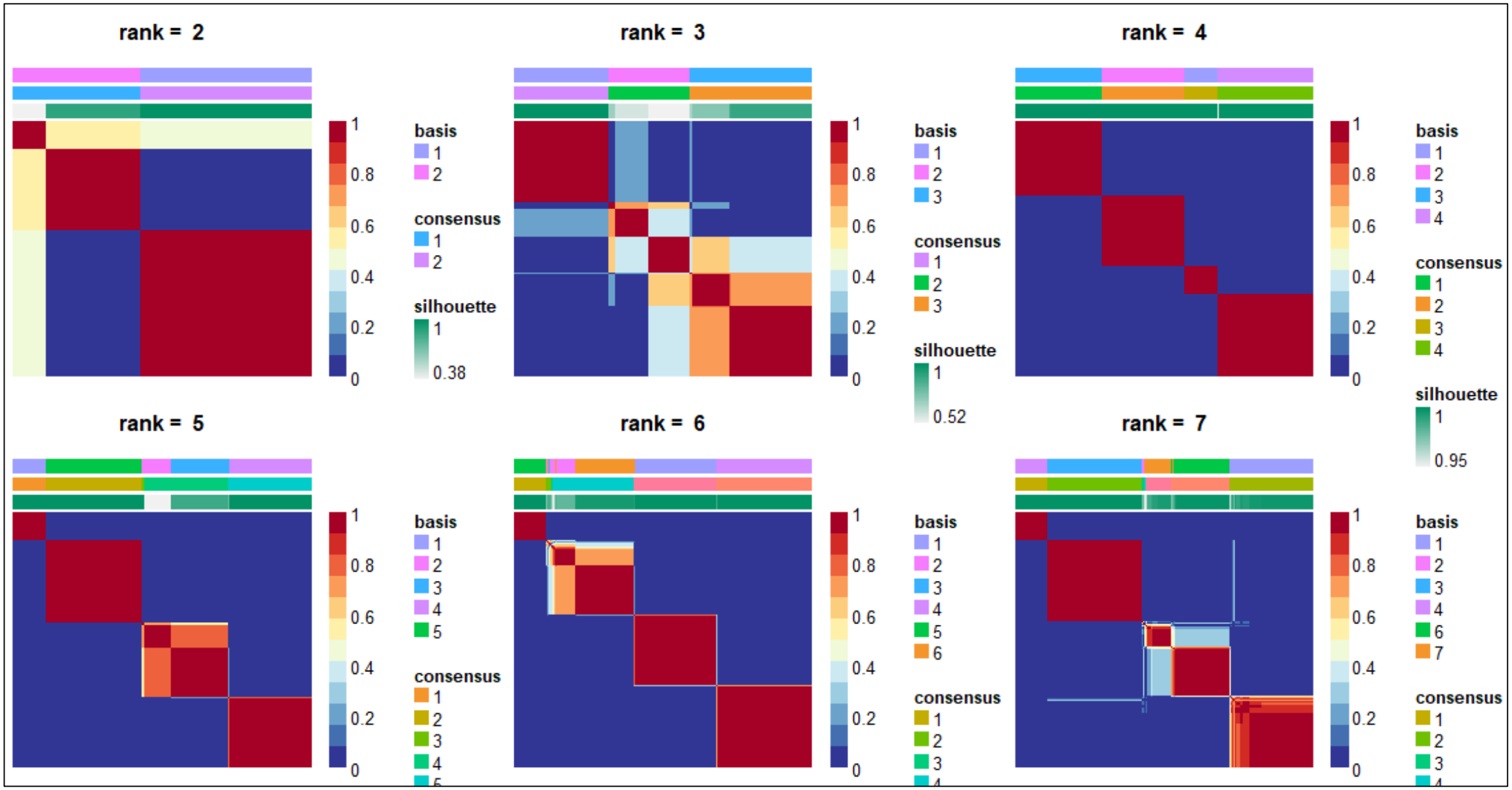




Heatmap of consensus matrix

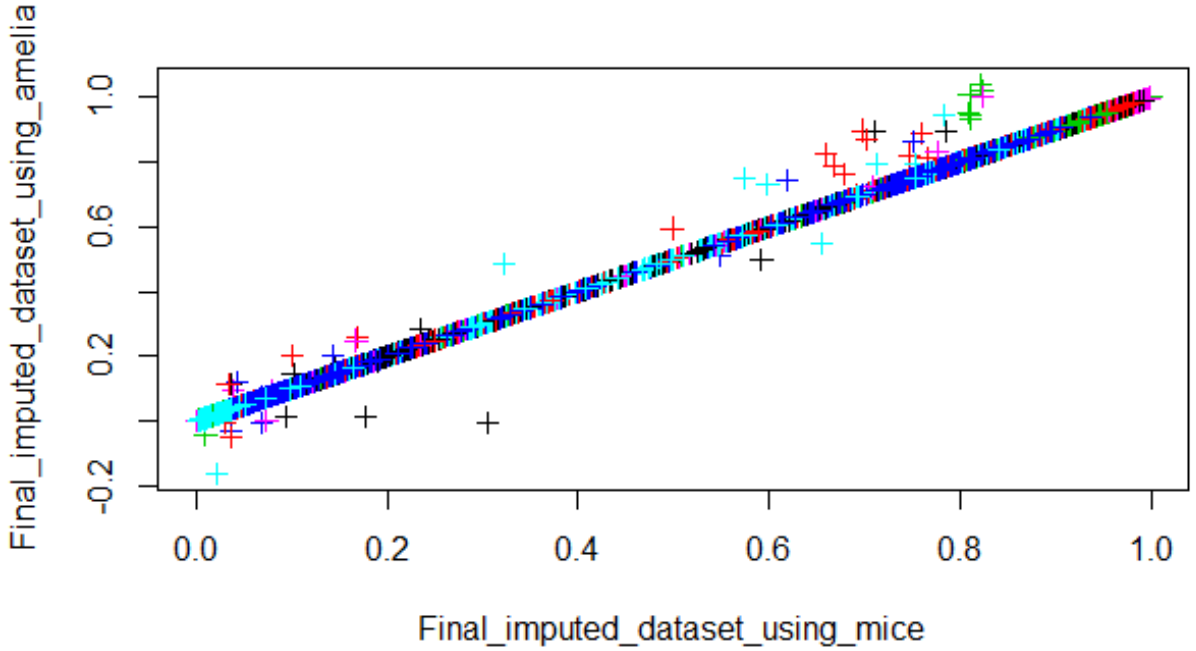


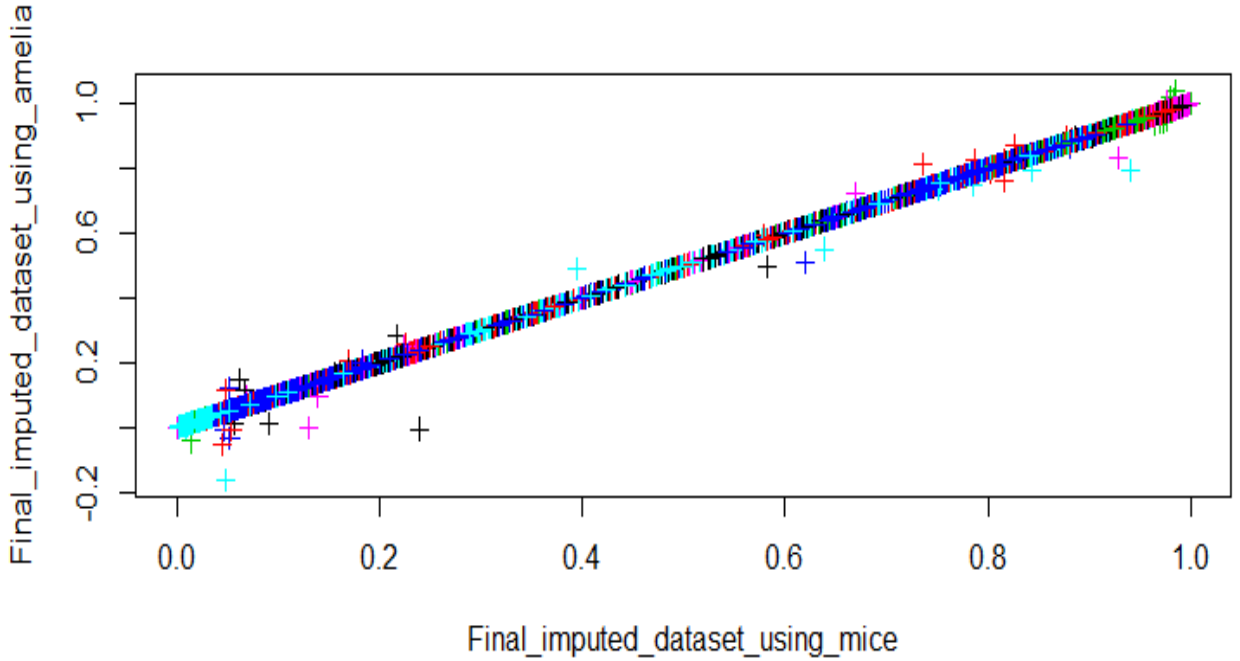
Heatmaps of consensus matrix for different ranks



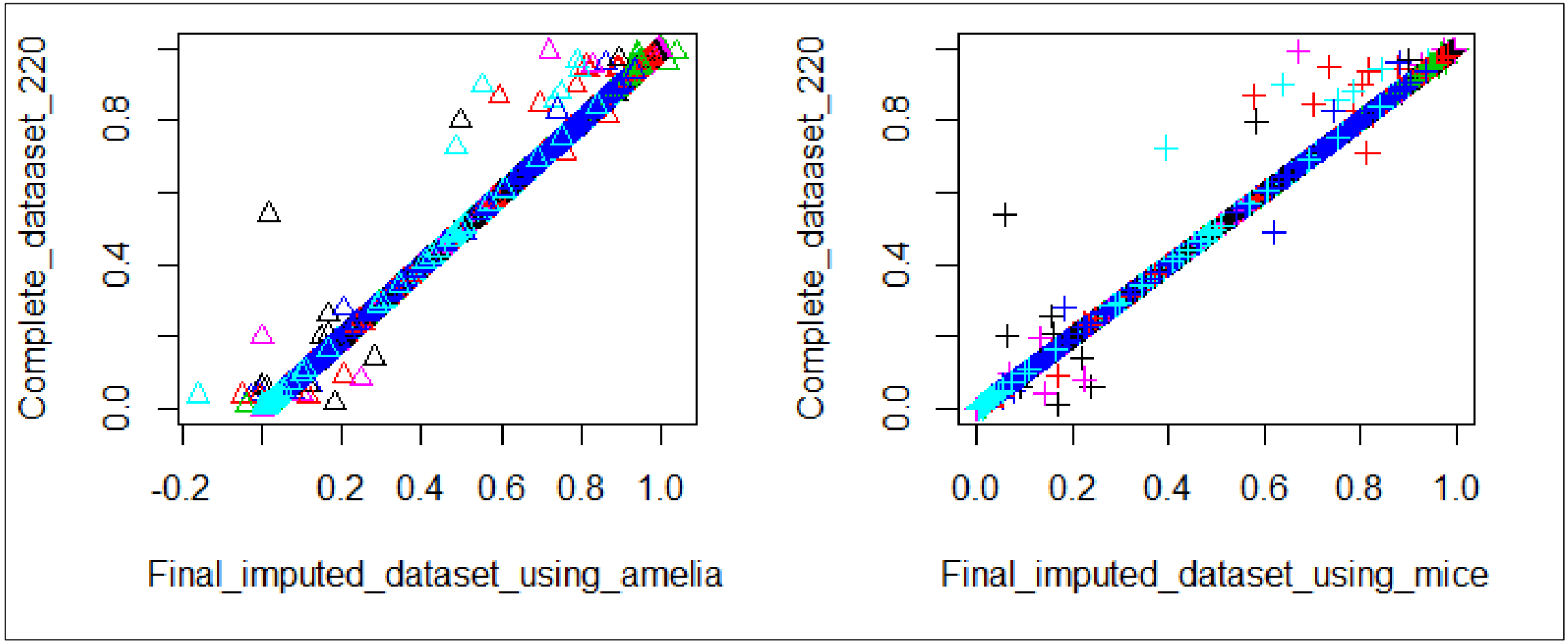
Any Questions?

Question: interpret two imputation results





Question: interpret two imputation results



Pattern of missing