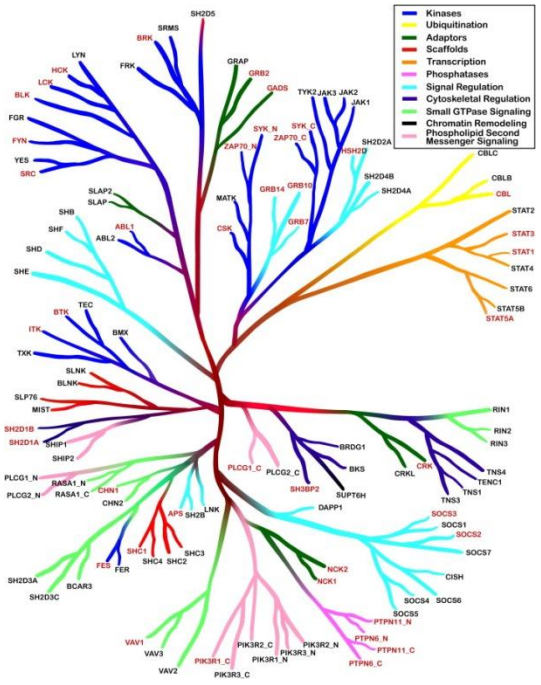
Les 13 – DEG's en clustering (2)

Emile Apol

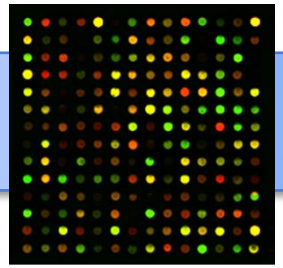
Hanze University Groningen
Applied Sciences
Institute for
Life Science & Technology

LES 13

- Hiërarchisch clusteren: **`hclust`**
  - opties plotten clusters
  - subclusters: **`cutree`**
  - visualisatie: **`matplot`**, **`image`**, **`heatmap`**, **`heatmap.2`**
  - kleurpalet in R



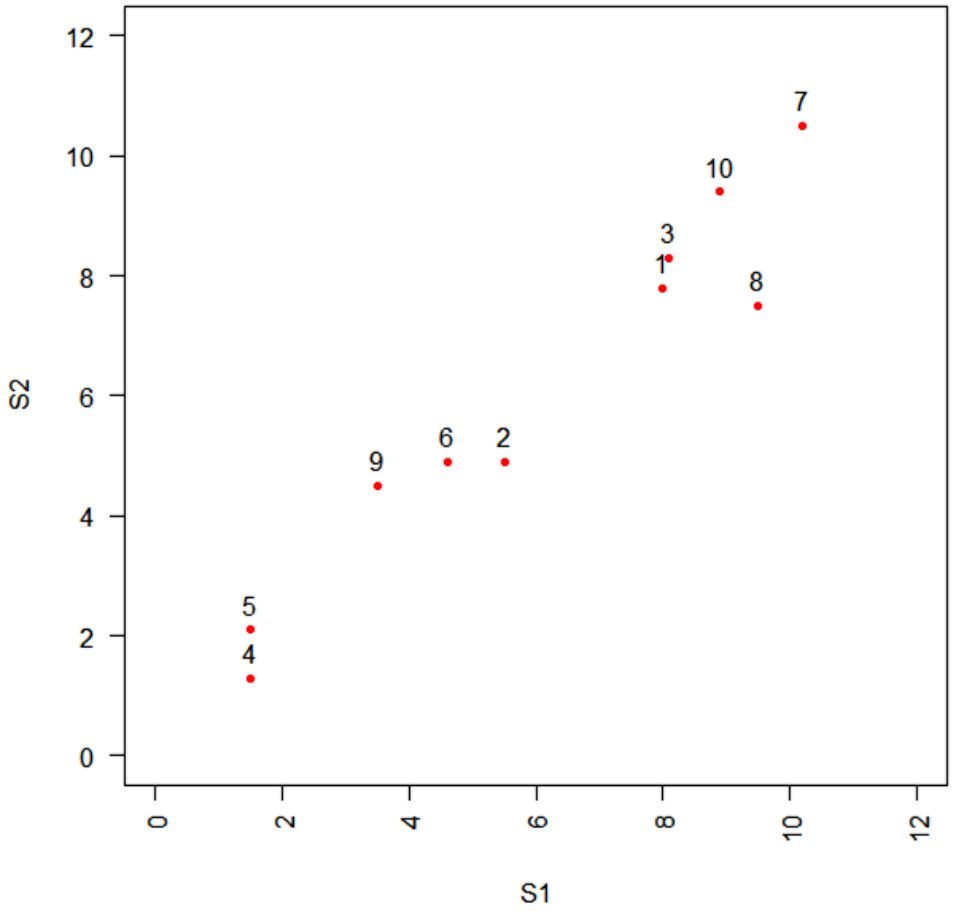
MICROARRAY ANALYSE: STAPPENPLAN



- Background correctie
- Log transformatie
- Normalisatie (bijv. loess)
- Toetsen op DEG's:
  - *t*-toets, 1-way ANOVA, …
  - Wilcoxon's toets, Kruskall-Wallis toets, …
- Aanpassen *p*-waarden voor multiple toetsing
- Clustering van DEG's:
  - Hiërarchisch clusteren
  - *k*-means
  - Principale Componenten Analyse (PCA)
- Grafische weergave: heatmap, vulcano plot, ...

CLUSTERING DEG'S

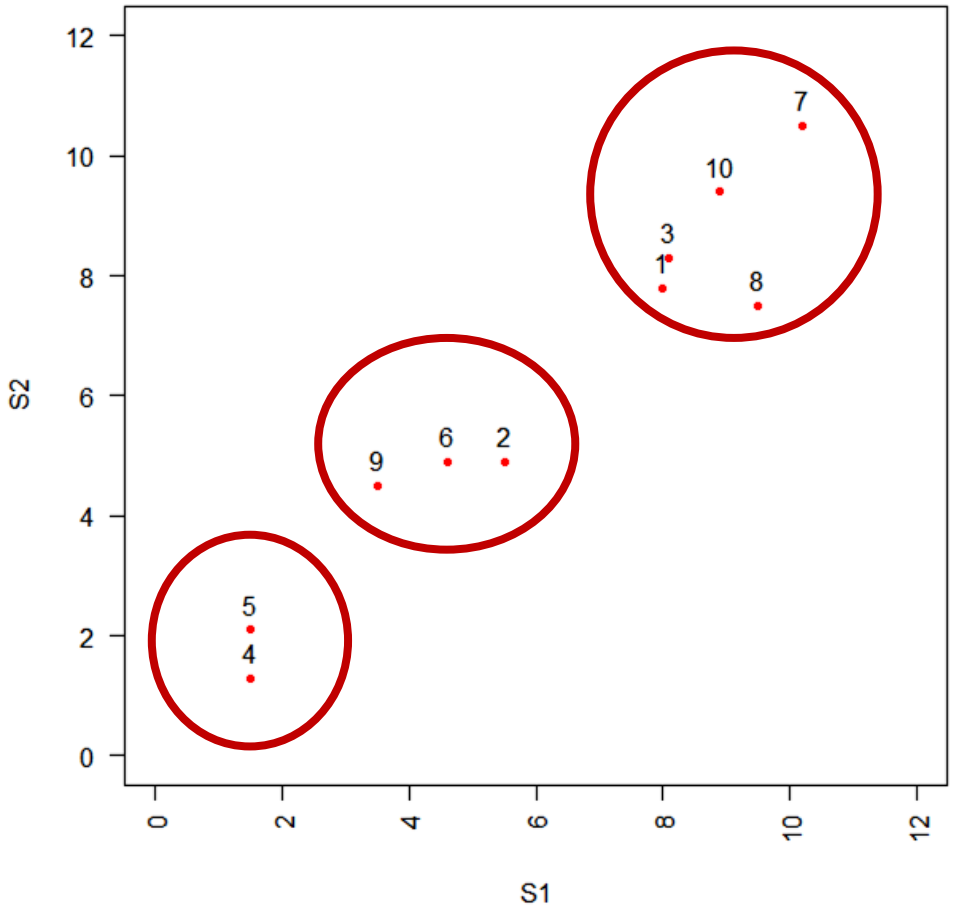
- Voorbeeld: 10 M-waarden van DEG's in 2 situaties S1 en S2 (bijv. tijdstippen, gehalten trigger molecuul X, etc.)



	row.names	S1	S2
1	gene 1	8	7.8
2	gene 2	5.5	4.9
3	gene 3	8.1	8.3
4	gene 4	1.5	1.3
5	gene 5	1.5	2.1
6	gene 6	4.6	4.9
7	gene 7	10.2	10.5
8	gene 8	9.5	7.5
9	gene 9	3.5	4.5
10	gene 10	8.9	9.4

CLUSTERING DEG'S

- Voorbeeld: 10 M-waarden van DEG's in 2 situaties S1 en S2 (bijv. tijdstippen, gehalten trigger molecuul x, etc.)



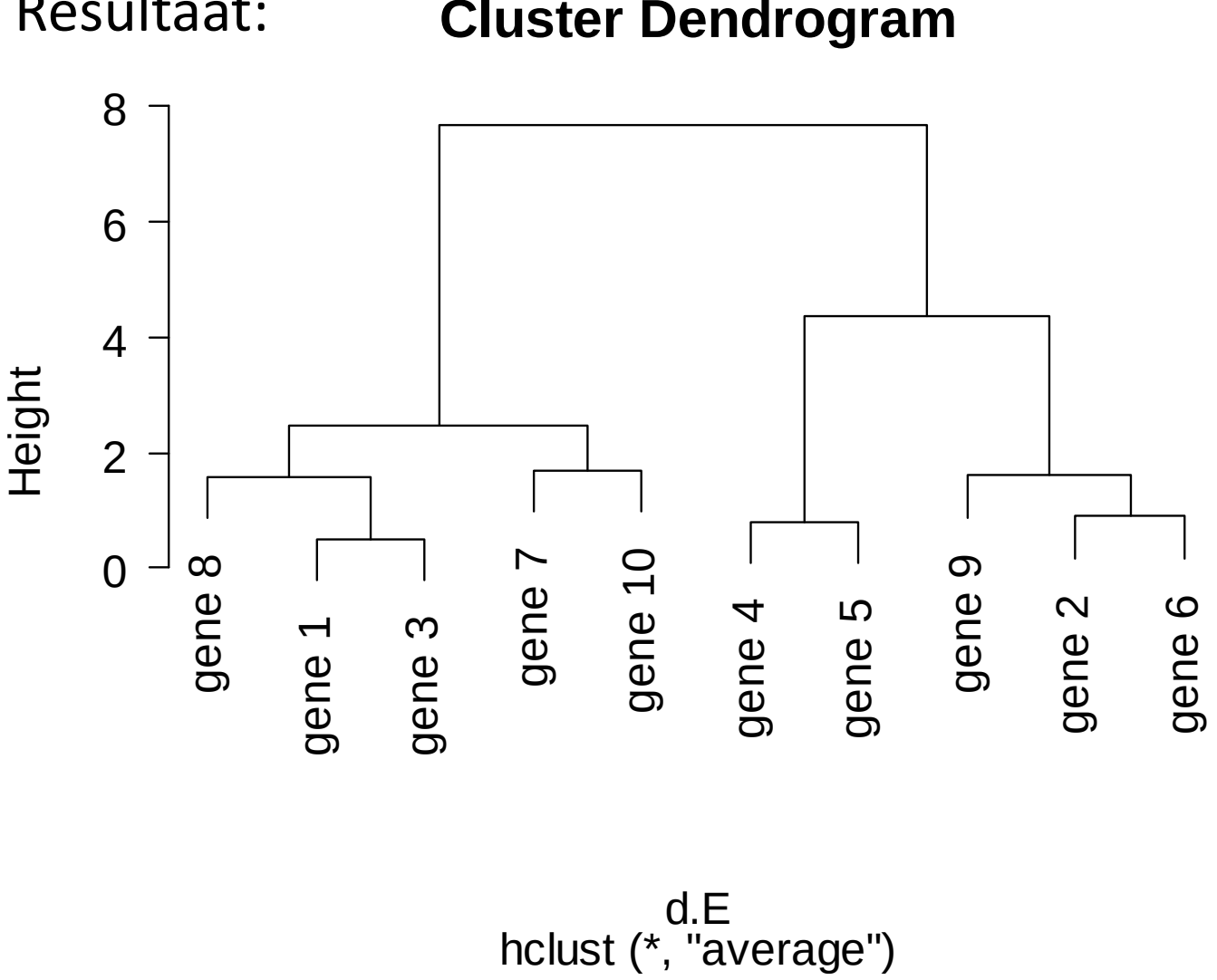
	row.names	S1	S2
1	gene 1	8	7.8
2	gene 2	5.5	4.9
3	gene 3	8.1	8.3
4	gene 4	1.5	1.3
5	gene 5	1.5	2.1
6	gene 6	4.6	4.9
7	gene 7	10.2	10.5
8	gene 8	9.5	7.5
9	gene 9	3.5	4.5
10	gene 10	8.9	9.4

CLUSTERING DEG'S

- Clustering van dataframe/matrix **M** met 2 samples per row, en 10 genen (=rows):
- Berekenen van afstandsmatrix:
  - `dMat <- dist(M, method="euclidean")`
- Bereken van clustering:
  - `clust <- hclust(dMat, method="average")`
- Plotten van dendrogram:
  - `plot(clust)`

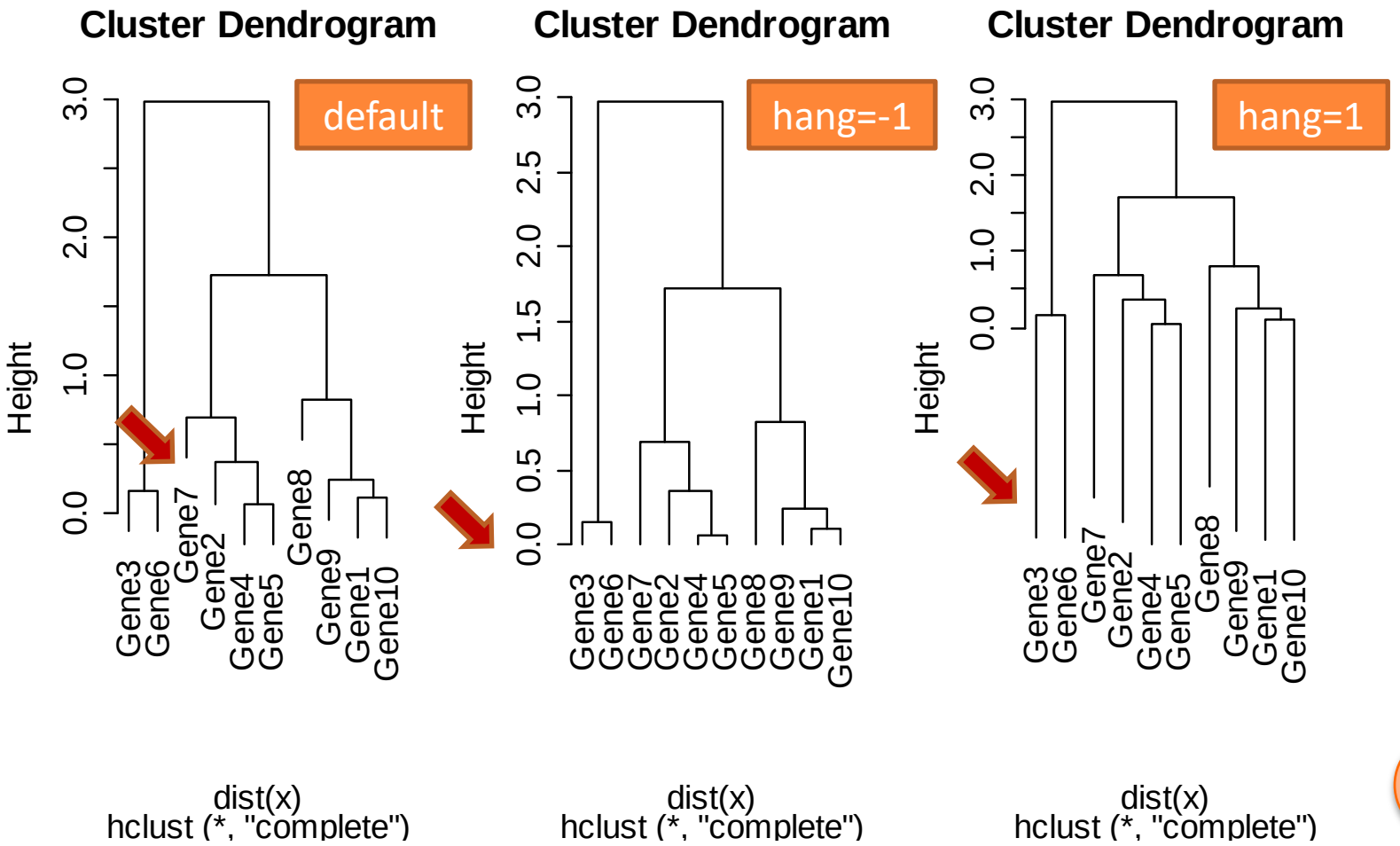
CLUSTERING DEG'S

- Resultaat:



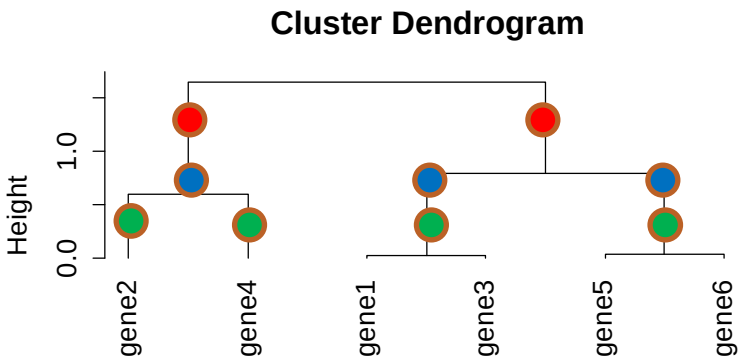
PLOTTEN VAN CLUSTERS

- `plot(clust)` = `plot(clust, hang=0.1)`:

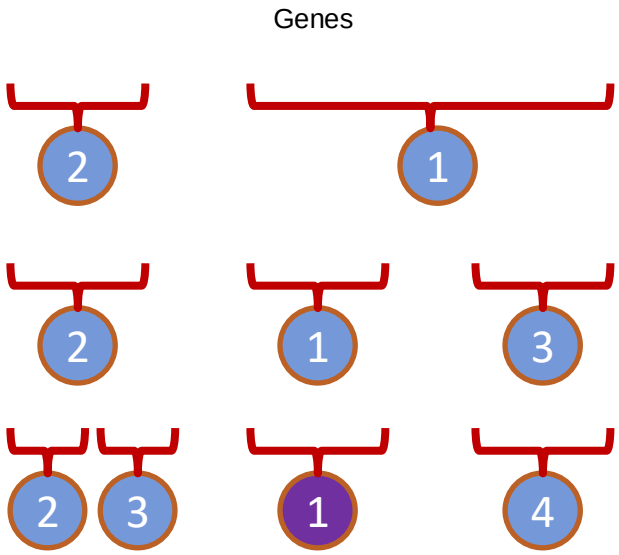


SUBCLUSTERS: CUTREE( )

- Opsplitsen van clustering **`clust`** in subclusters
- Gegeven aantal subclusters k:
  - **`sC.2<-cutree(clust, k=2)`**
  - **`sC.3<-cutree(clust, k=3)`**
  - **`sC.4<-cutree(clust, k=4)`**
- Resultaat: subcluster nummer per gen




```
> ( sC.2 <- cutree(clust, k=2) )
gene1 gene2 gene3 gene4 gene5 gene6
    1     2     1     2     1     1
> ( sC.3 <- cutree(clust, k=3) )
gene1 gene2 gene3 gene4 gene5 gene6
    1     2     1     2     3     3
> ( sC.4 <- cutree(clust, k=4) )
gene1 gene2 gene3 gene4 gene5 gene6
    1     2     1     3     4     4
```



SUBCLUSTERS: CUTREE( )

- Dataframe **M**:

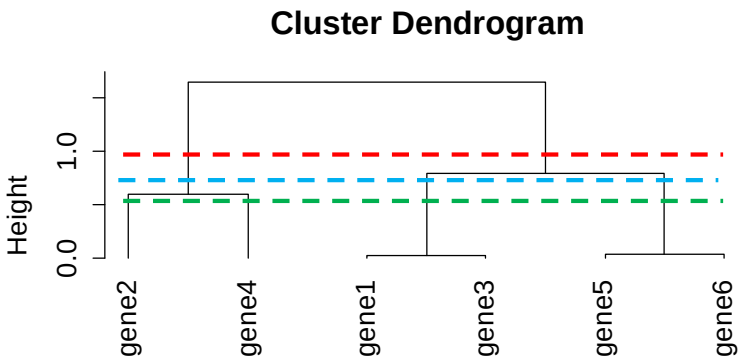
	M1	M2	M3	M4	M5
gene1	5.1	5.6	6.2	5.7	5.5
gene2	4.9	4.5	4.3	4.4	4.7
gene3	10.1	10.7	11.9	11.1	10.5
gene4	8.9	8.5	7.8	7.5	7.1
gene5	2.3	2.5	2.7	3.4	3.9
gene6	2.4	4.1	5.8	7.7	8.9

- Data uit dataframe **M** selecteren voor 1e subcluster van de 4:
  - `M[sC.4 == 1, ]`

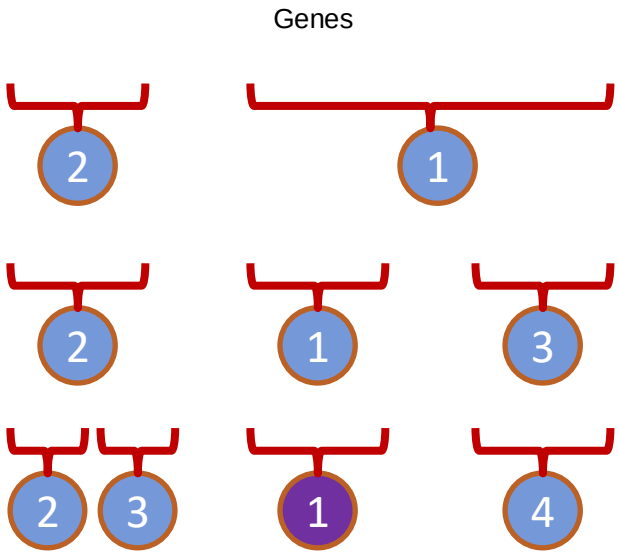
	M1	M2	M3	M4	M5
gene1	5.1	5.6	6.2	5.7	5.5
gene3	10.1	10.7	11.9	11.1	10.5

SUBCLUSTERS: CUTREE( )

- Opsplitsen van clustering **`clust`** in subclusters
- Gegeven hoogte *h* in dendrogram:
  - **`sC<-cutree(clust, h=1.0)`**
  - **`sC<-cutree(clust, h=.75)`**
  - **`sC<-cutree(clust, h=.50)`**
- Resultaat: subcluster nummer per gen



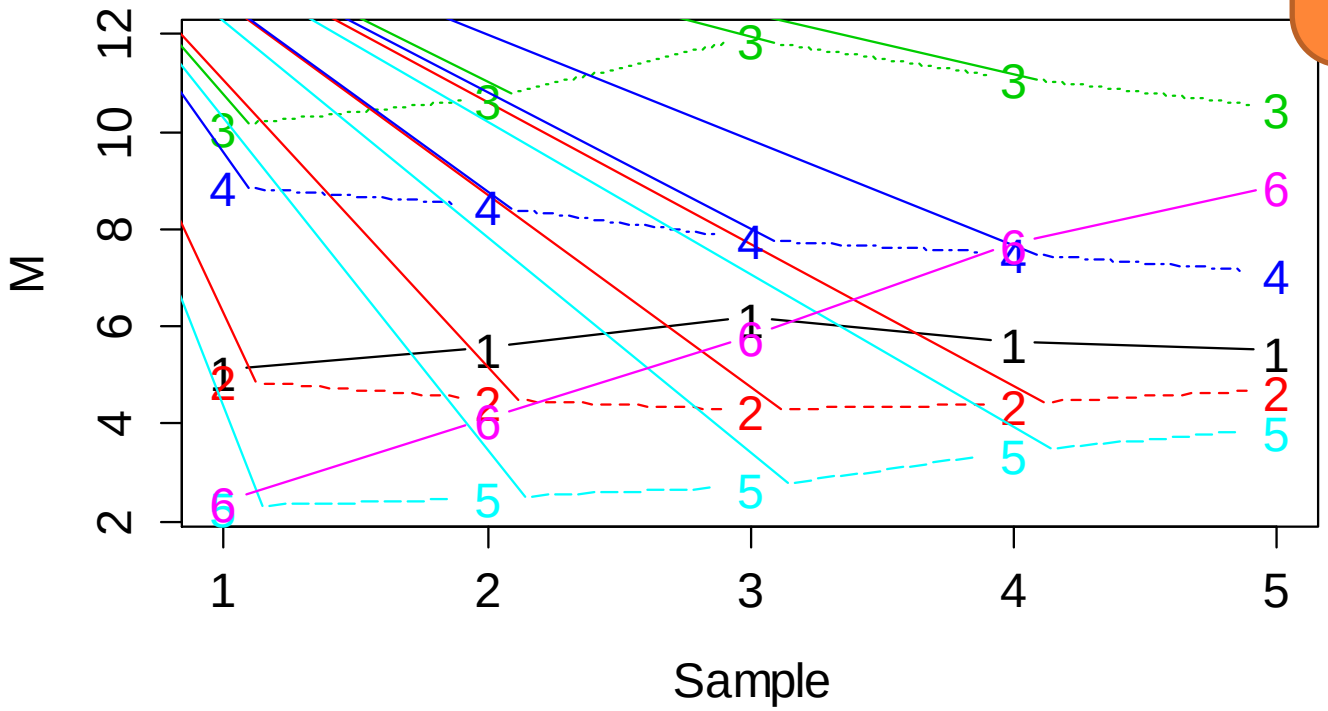

```
> ( sC <- cutree(clust, h=1.0))
gene1 gene2 gene3 gene4 gene5 gene6
    1     2     1     2     1     1
> ( sC <- cutree(clust, h=0.75))
gene1 gene2 gene3 gene4 gene5 gene6
    1     2     1     2     3     3
> ( sC <- cutree(clust, h=0.5))
gene1 gene2 gene3 gene4 gene5 gene6
    1     2     1     3     4     4
```



MATPLOT( )

- Plotten van een dataframe **M** met G rijen (genes) en n samples
- **matplot(t(MA), type="b", xlab="Sample", ylab="M")**

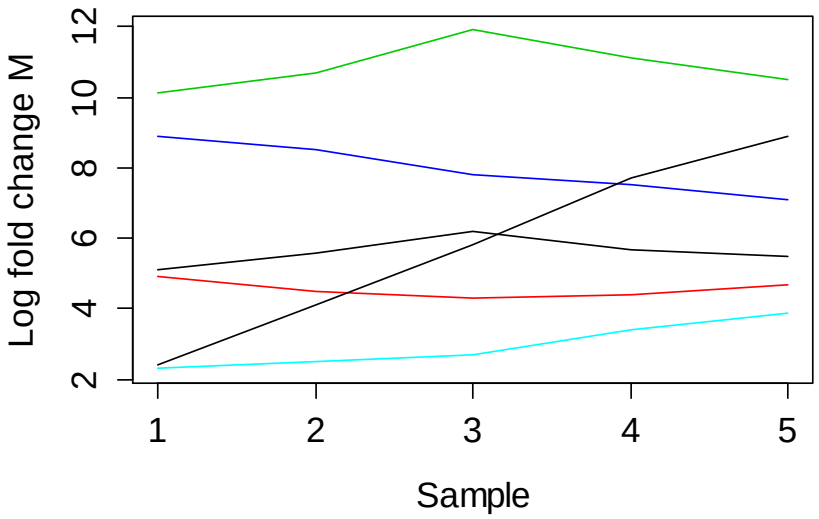
matplot plot *kolommen* van matrix als losse datasets



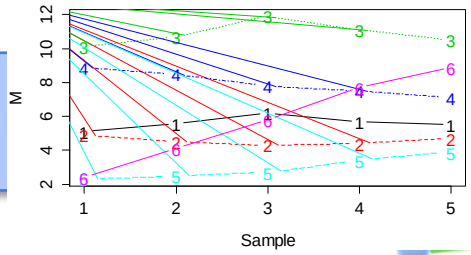
MATPLOT( )

- Naast **`matplot( )`**, die een nieuwe plot aanmaakt, zijn er ook:
  - **`matpoints( )`** # tekent punten in bestaande plot
  - **`matlines( )`** # tekent lijnen in bestaande plot
- Bijvoorbeeld:

```
matplot(t(MA), type="n", xlab="Sample", ylab="Log fold change M")
matlines(t(MA), col=1:ncol(MA), lty=1)
```



IMAGE( )

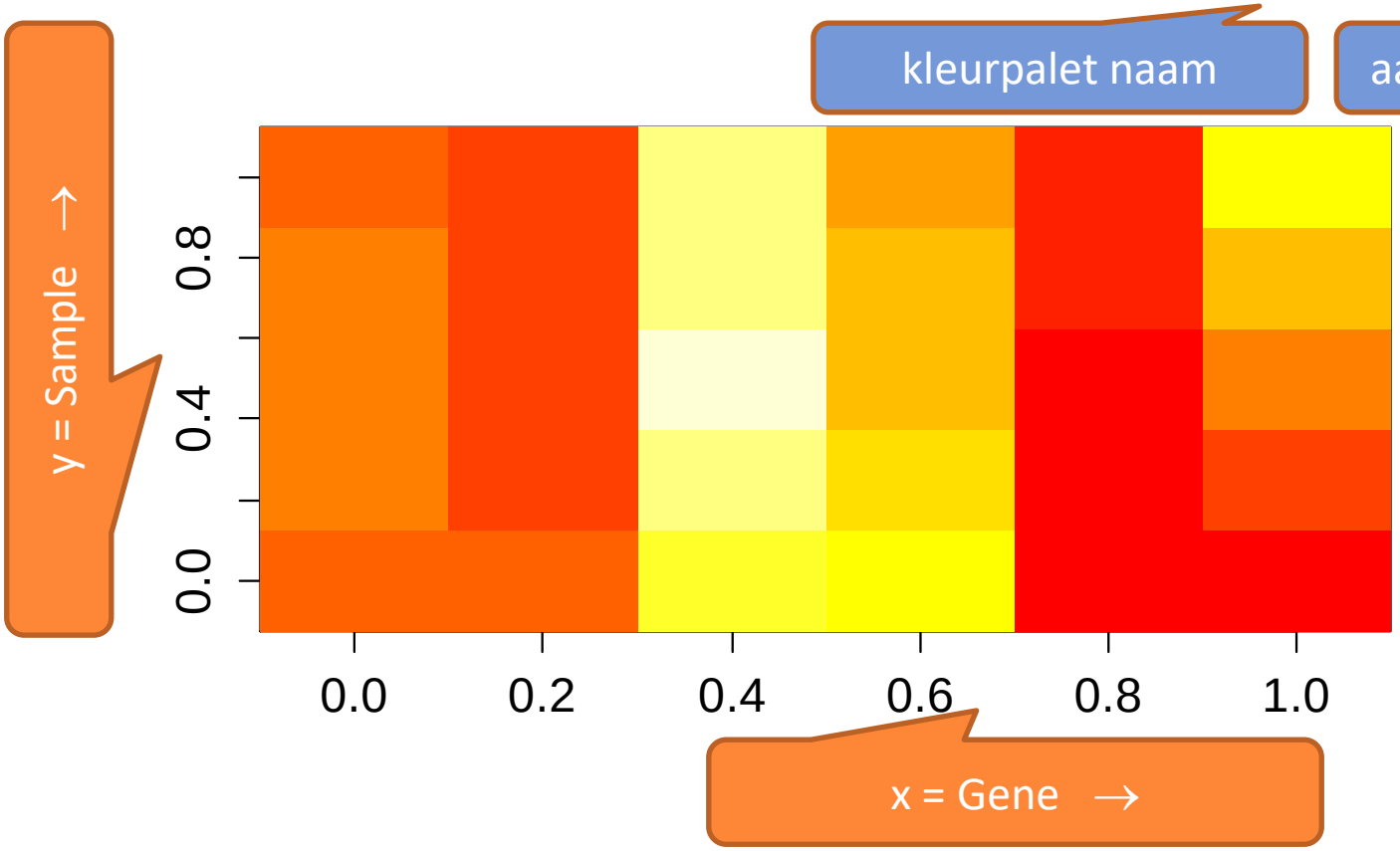


- Maken van een kleurenmap van dataframe **M**:
  - `X <- as.matrix(M)`
  - `image(X) = image(X, col=heat.colors(12))`

image wil matrix, geen dataframe!

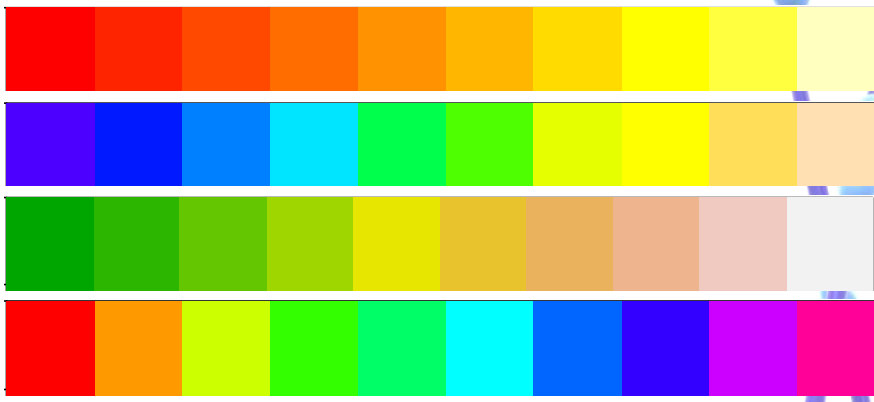
kleurpalet naam

aantal versch. kleuren



KLEURPALET IN R

- Voorgedefinieerde kleurpaletten (**n** kleuren):
  - **heat.colors(n=)**
  - **topo.colors(n=)**
  - **terrain.colors(n=)**
  - **rainbow(n=)**



- Eigen palet maken: RGB kleuren (Red/Green/Blue)
  - **rood**: `rgb( c(255,0,0), maxColorValue=255)`
  - **groen**: `rgb( c(0,255,0), maxColorValue=255)`
  - **blauw**: `rgb( c(0,0,255), maxColorValue=255)`
  - WIT: `rgb( c(255,255,255), maxColorValue=255)`
  - **zwart**: `rgb( c(0,0,0), maxColorValue=255)`

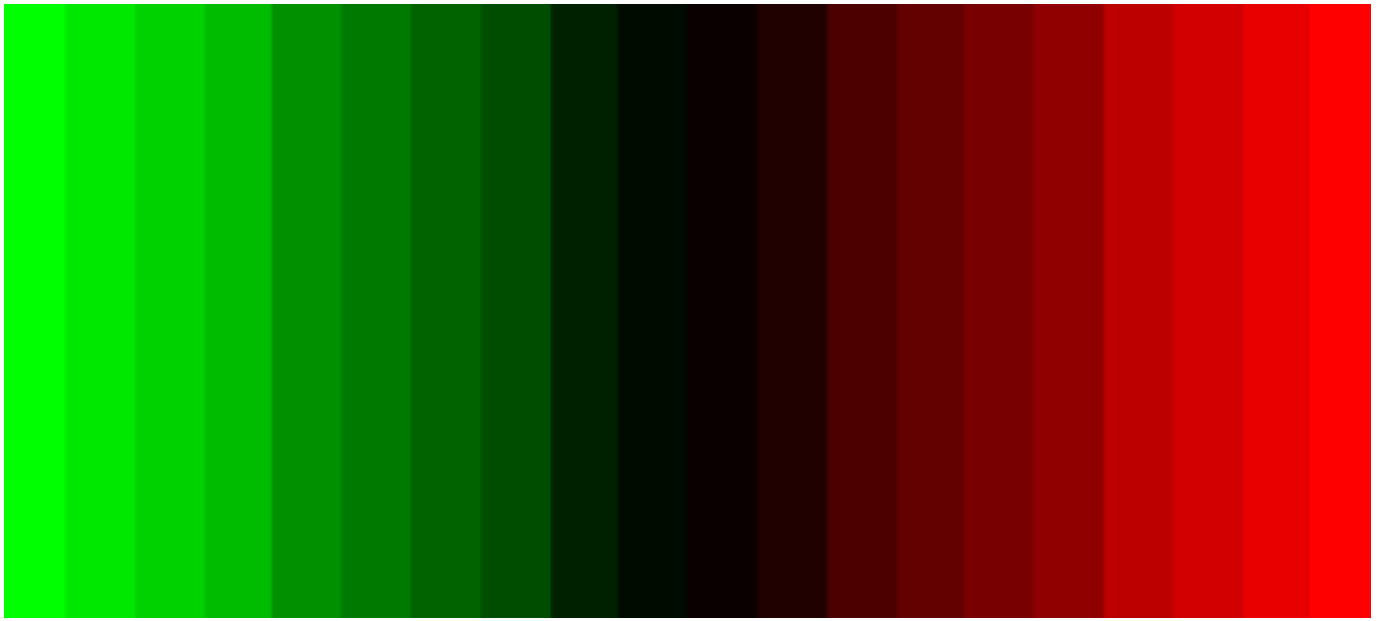
KLEURPALET VOOR MA'S: LET OP

- Breng de lezer niet in verwarring:
  - NIET: rainbow (onlogische kleuren!)
  - WEL: 1 of 2 kleuren
- Denk aan lezers met kleurenblindheid:
  - NIET: rood+groen, groen+bruin, groen+blauw, blauw+grijs, blauw+paars, groen+grijs, groen+zwart, licht groen+geel
  - WEL: blauw+oranje, blauw+rood, blauw+bruin

MA.COLORS( )

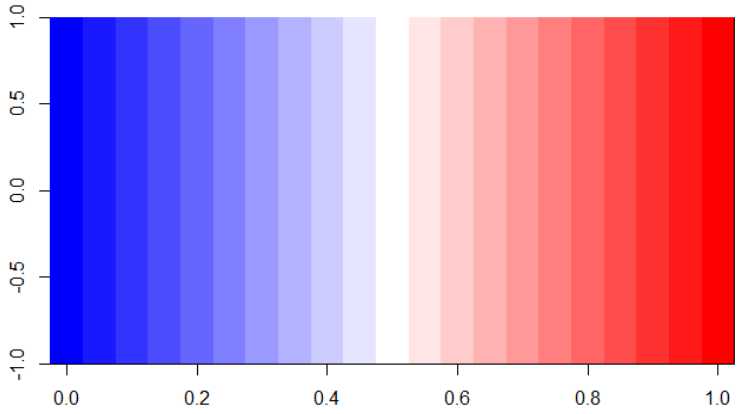
- Eigen definitie van kleurpalet (met n kleuren):

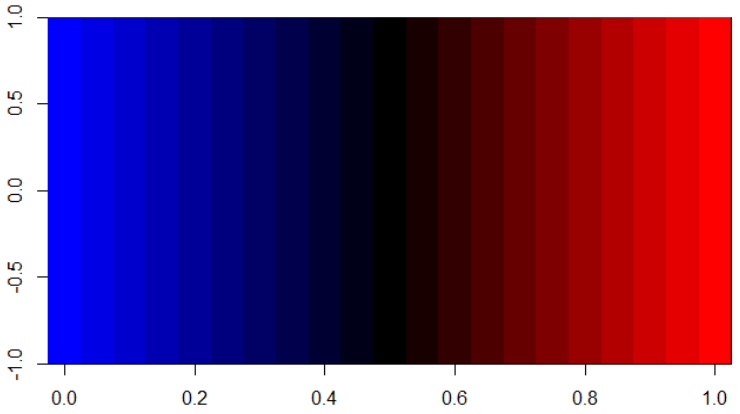
```
MA.colors <- function(n=12){
  colorRampPalette(c("green", "black", "red"), space="rgb")(n)
}
```



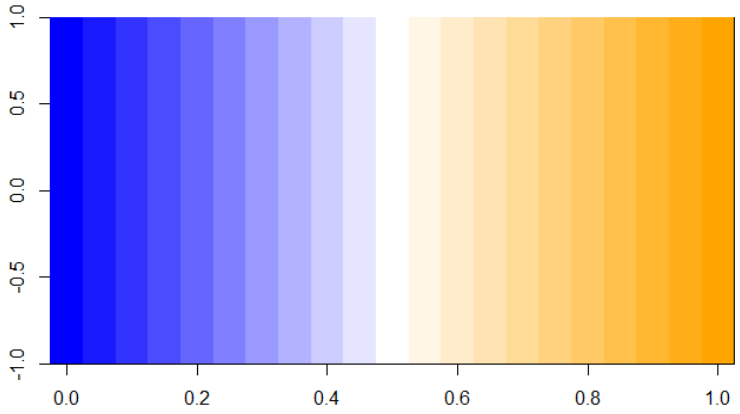
HANDIGE KLEURPALETTEN

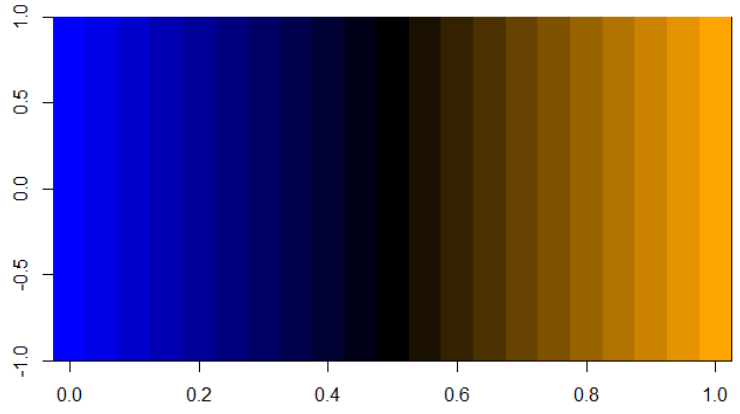
Blauw + rood:





Blauw + oranje:

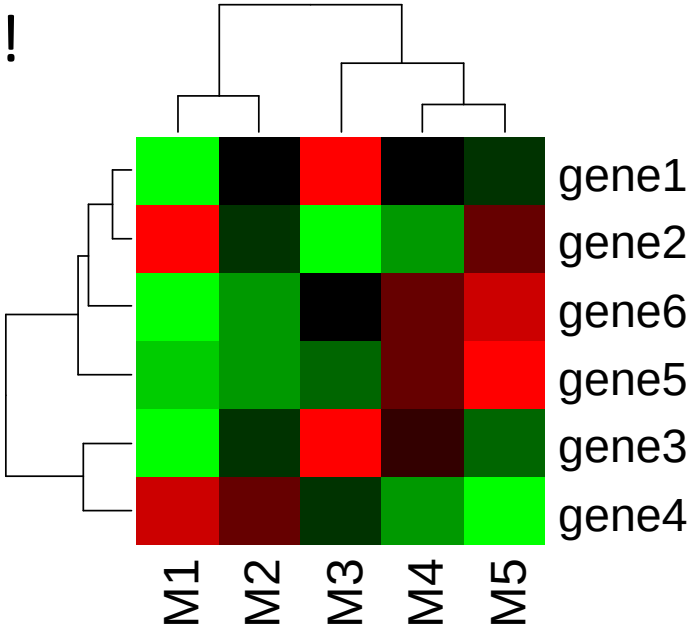




HEATMAPS

- Image van dataframe met enkel- (gen) of tweevoudige (gen & sample) clustering van dataframe **`M`**:
  - **`X <- as.matrix(M)`**
  - **`heatmap(X, col=MA.colors() )`**
- **`heatmap`** gebruikt de *default* methoden voor **`dist( )`** en **`hclust( )`** functies!

heatmap voer ook default een *scaling normalisatie* uit per gen!



DEFAULTS VAN FUNCTIES DIST( ) EN HCLUST( )

- Dataframe/matrix **M** met G rijen (genes) en n samples
- Distances:
  - **d <- dist(M)**
  - default: **method="euclidean"**
- Distance object **d**
- Hiërarchisch clusteren:
  - **clust <- hclust(d)**
  - default: **method="complete"**

EIGEN DIST EN HCLUST FUNCTIES

- **`dist`** functies, werkend op matrix **`x`** (rij = gen):
  - Euclidisch:
  - `myDist.E <- function(x) { dist(x, method="euclidean") }`
  - Manhattan:
  - `myDist.M <- function(x) { dist(x, method="manhattan") }`
  - Pearson:
  - `myDist.P <- function(x) { as.dist( 1 - cor(t(x)) ) }`
  - "Absolute" Pearson:
  - `myDist.AP <- function(x) { as.dist( 1 - abs( cor(t(x)) ) ) }`

EIGEN DIST EN HCLUST FUNCTIES

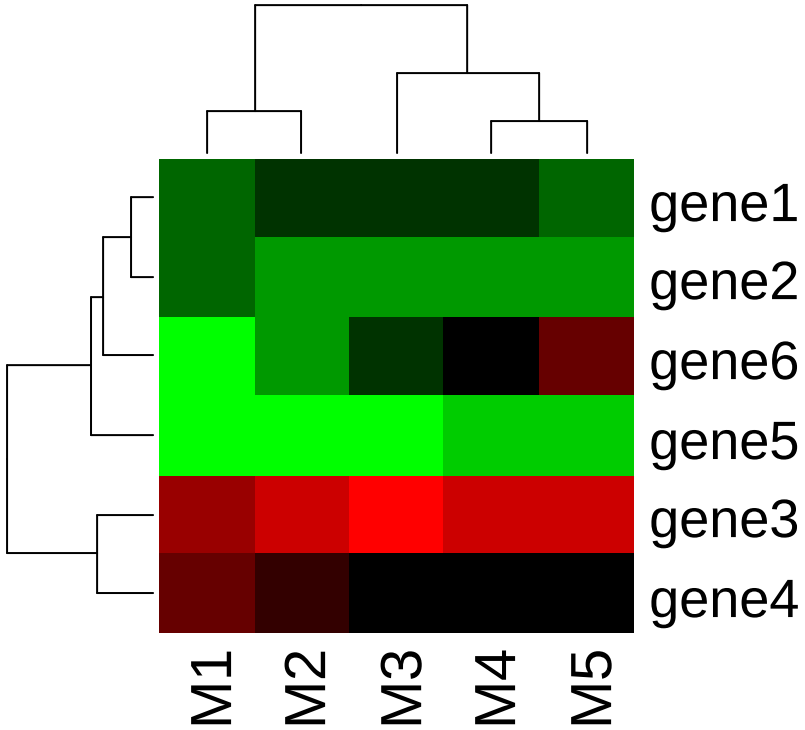
- **`hclust`** functies, werkend op dist object **`x`**:
  - Single linkage:
  - ```
    myHclust.S <- function(x) {
          hclust(x, method="single") }
    ```
  - Complete linkage:
  - ```
    myHclust.C <- function(x) {
          hclust(x, method="complete") }
    ```
  - Average linkage:
  - ```
    myHclust.A <- function(x) {
          hclust(x, method="average") }
    ```
    ```

HEATMAPS MET WILLEKEURIGE CLUSTER METHODEN

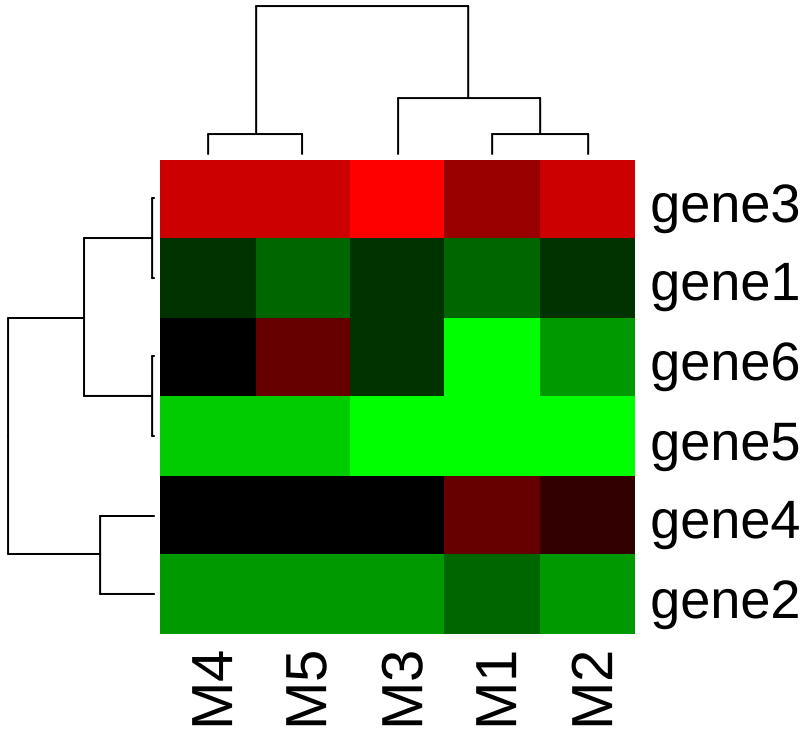
- Definieer eerst eigen dist en hclust functies, bijv.
  - `myDist.P <- function(x) { as.dist( 1 - cor(t(x)) ) }`
  - `myHclust.A <- function(x) { hclust(x, method="average") }`
- Roep **heatmap** aan met eigen functies, *zonder scaling*:
  - `X <- as.matrix(M)`
- `heatmap(X, col=MA.colors(), scale="none", hclustfun=myHClust.A, distfun=myDist.P)`

EIGEN HEATMAPS

- Default heatmap (Euclidisch/Complete) vs. Pearson/Average, allen met optie **`scale="none"`**:



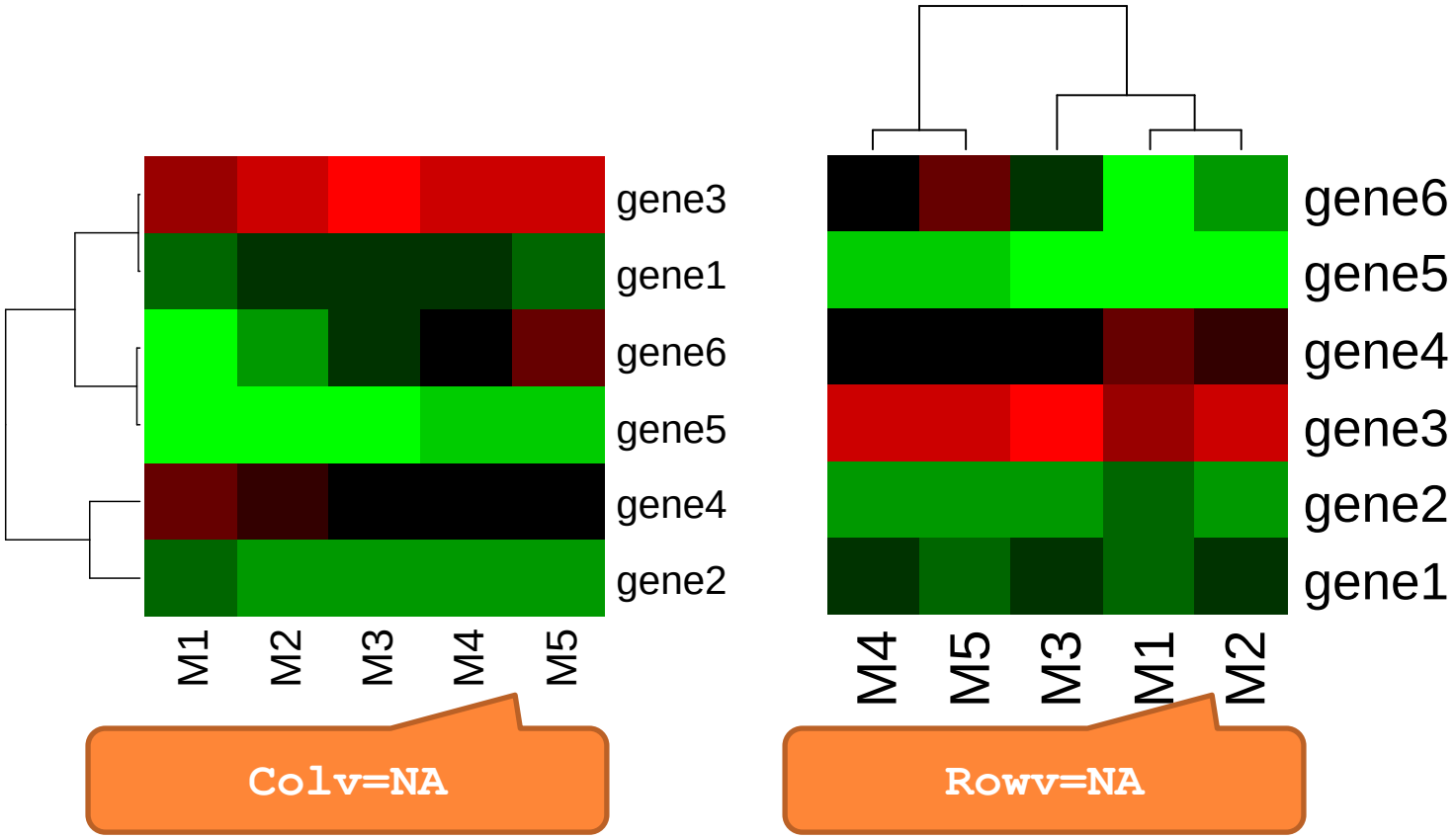
Euclidisch/Complete



Pearson/Average

HEATMAPS MET ENKELE CLUSTERING

- Je kunt clustering van genen en/of samples uitzetten:
  - geen clustering samples: **`Colv=NA`**
  - geen clustering genen: **`Rowv=NA`**



HEATMAP.2( )

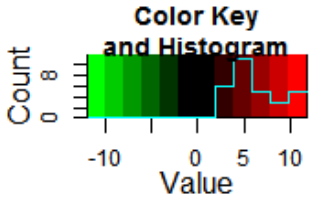
- In package **gplots**
  - `package(gplots)`
  - `library(gplots)`
  - `X <- as.matrix(M)`
  - `heatmap.2(X, col=MA.colors(), scale="none", hclustfun=myHClust.A, distfun=myDist.P, symbreaks=T, trace="none")`

Zorgt ervoor dat het midden van het kleurpalet overeenkomt met de waarde X = 0

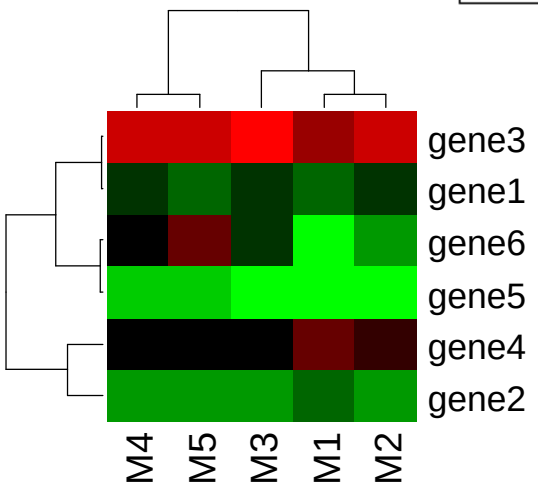
HEATMAP.2( )

histogram van kleuren

- Resultaat:



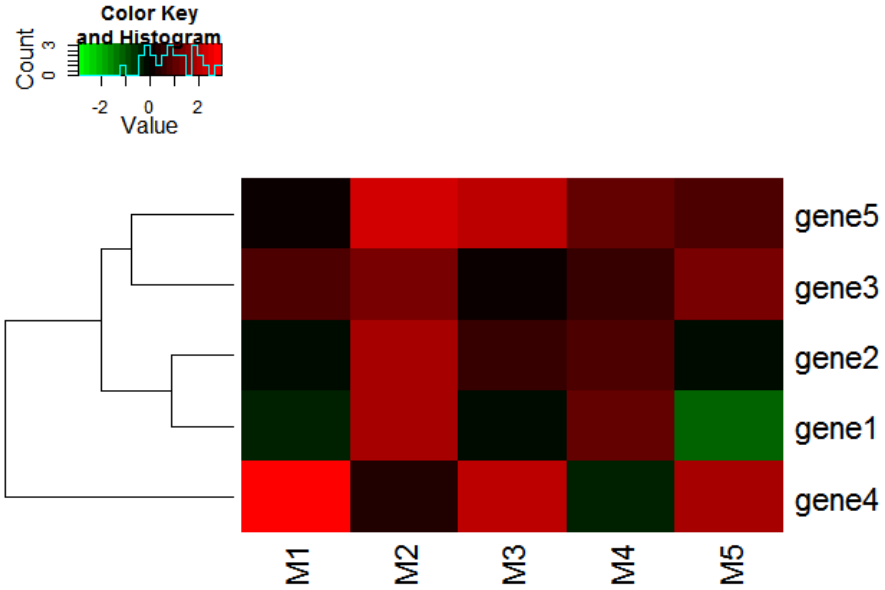
legenda!



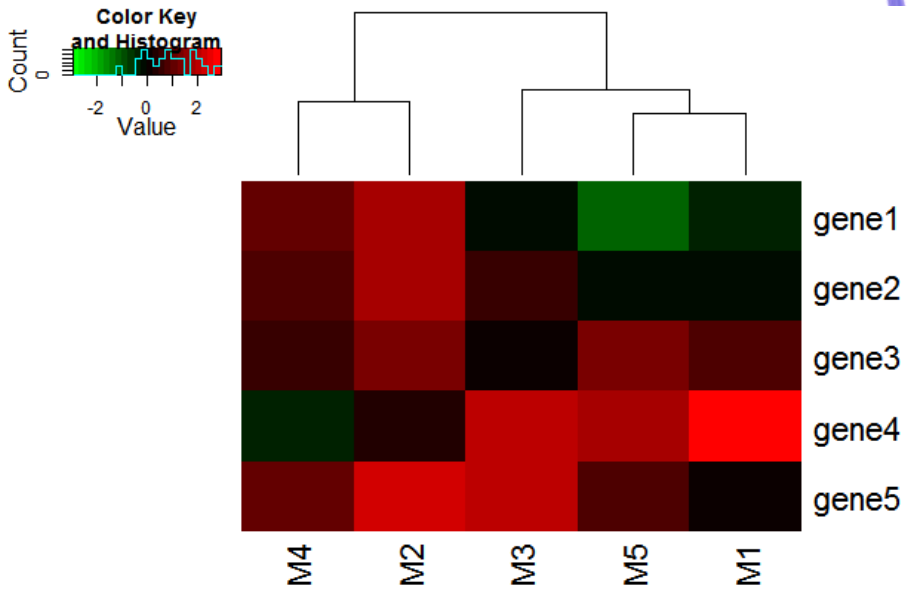
heatmap( )

HEATMAP.2( )

- Ook hier alleen clustering op genen of samples, via **`Colv=NA`** of **`Rowv=NA`**:

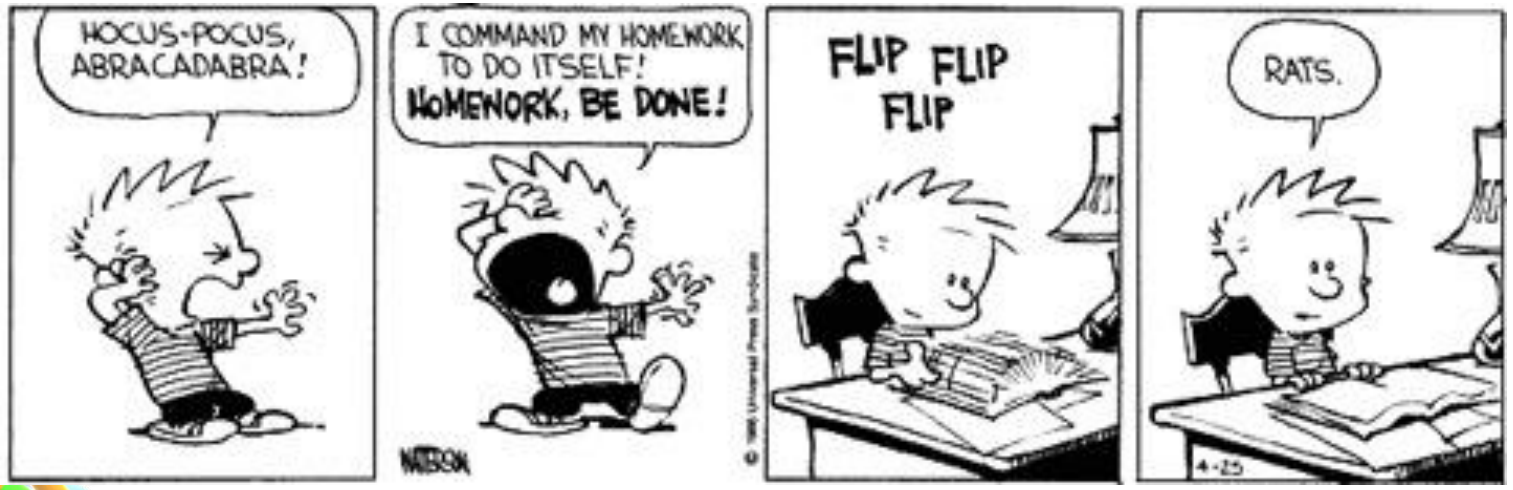


`Colv=NA`



`Rowv=NA`

Jullie kunnen nu de opdrachten van les 13 maken



Hanze University Groningen
Applied Sciences
Institute for
Life Science & Technology