

# CLustering Mouse embryo E8.5

Selina Dreesman

2024-08-13

## Step 1: Load packages

Script laadt de volgende packages in met de library functie om ze toe te kunnen passen voor analyse. - (dplyr) - (ggplot2) - (pheatmap) - (tidyverse) - (RColorBrewer) - (ggrepel) - (Seurat) - (Matrix) - (here)

## Step 2: Load Data & create object

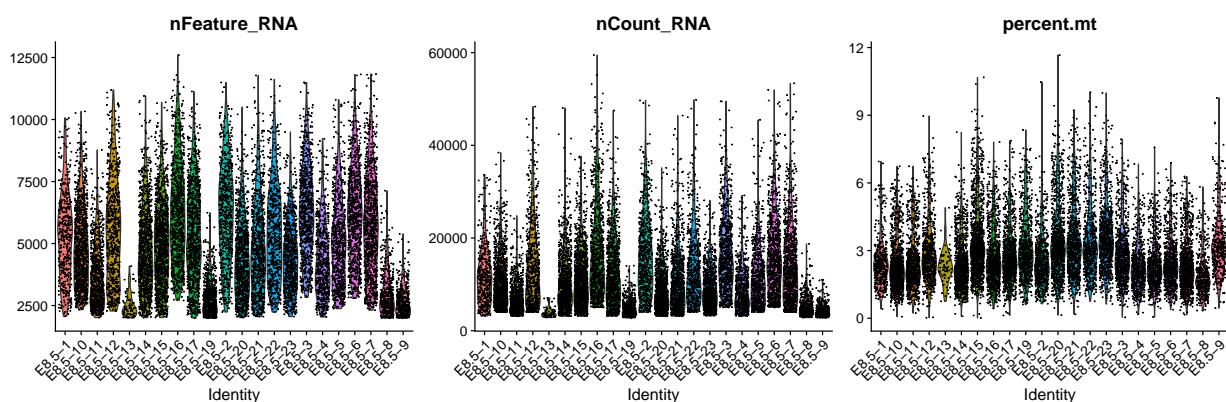
Data wordt als csv bestand ingeladen tot object en bewaard in het werkgeheugen. Vervolgens worden de data als object genaamd “counts” ingeladen samen met een matrix tabel. Als laatste wordt dit met behulp van de Seurat packages een object gemaakt als preprocessing voor de data analyse, afkapwaarde is vanaf minimaal 2000 features (genen) .

Antwoord op de deelvraag: in welke format kan mijn data ingeladen worden is dat het in beide formaten kan, zowel als CSV als MTX bestand kan het met de functie “read” ingeladen worden.

## Step 3: data inspection & quality control

Patroon -mt wordt opgezocht in het object “seurat”. Vervolgens wordt het uitgeplot in een violinplot, net als de functies nFeature\_RNA en nCount\_RNA. Dit als kwaliteitscheck hoe de data verdeeld is.

```
## Warning: Default search for "data" layer in "RNA" assay yielded no results;
## utilizing "counts" layer instead.
```



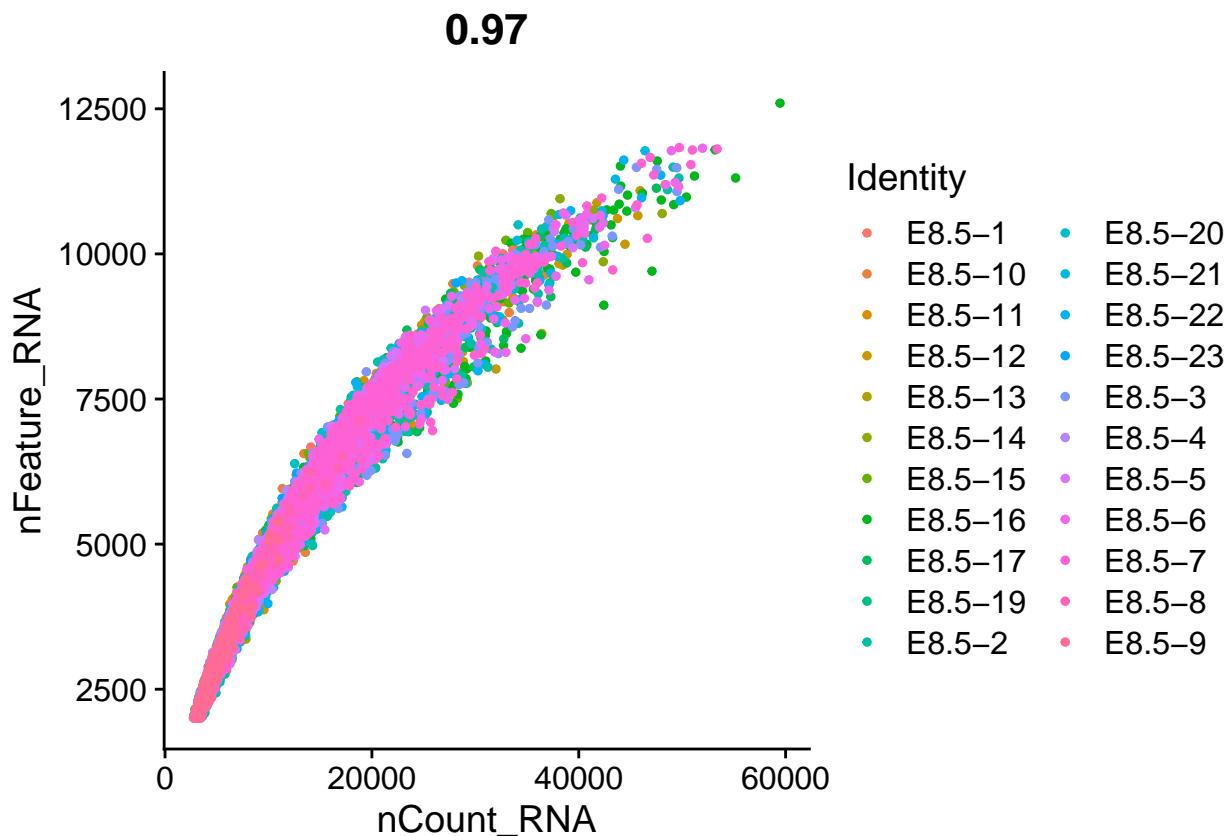
Violinplot die de verhouding weergeeft van de data aan de hand van de volgende features x-as= datalabel, y-as= aantal : - nFeature\_RNA; het aantal genen in de dataset - nCount\_RNA; het aantal moleculen in de dataset - percent.mt; het percentage mitochondriaal genexpressie

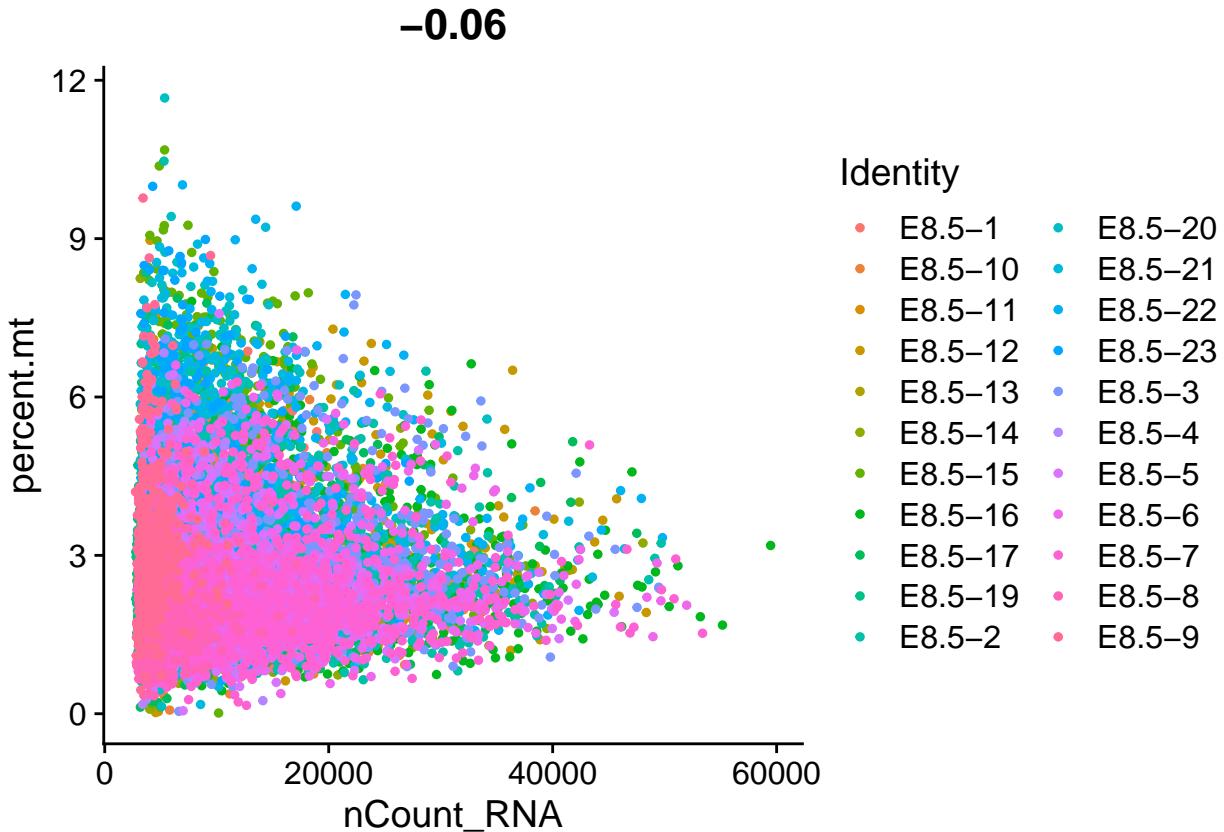
De data toont een verdeling aan volgens het vioolprincipe, waarbij het begint vanaf 2000 features. Optimaal zou zoveel mogelijk features meenemen beter zijn echter vanwege maximale servercapaciteiten gekozen voor 2000 features om de server te ontlichten en crashes te voorkomen.

Identificatie van inhoud - Weergave het aantal per datalabel - Weergaven van genen en de count die eraan verbonden is. Weergave is suboptimaal, is alleen ter illustratie om aan te tonen dat het zich in de dataset bevindt. Aantal rijen en kolommen te groot voor correcte weergave.

```
## 
##   E8.5-1  E8.5-10 E8.5-11 E8.5-12 E8.5-13 E8.5-14 E8.5-15 E8.5-16 E8.5-17 E8.5-19
##   411     1098    778     652     73     1004    1154     865     934     654
##   E8.5-2  E8.5-20 E8.5-21 E8.5-22 E8.5-23 E8.5-3   E8.5-4   E8.5-5   E8.5-6   E8.5-7
##   671     938     835     596     798     772     586     696     814     962
##   E8.5-8  E8.5-9
##   437     398
```

Feature Scatter: Inladen van twee nieuwe object, waarvan de pearson correlatie geanalyseerd wordt. Dit om aan te tonen of de data in de dataset wel overeenkomt met de verwachtingen, geen data verloren of vervuild is met andere data. De andere genaamde cnt\_mt is om aan te tonen dat de set niet vervuild is met mitochondriaal materiaal.



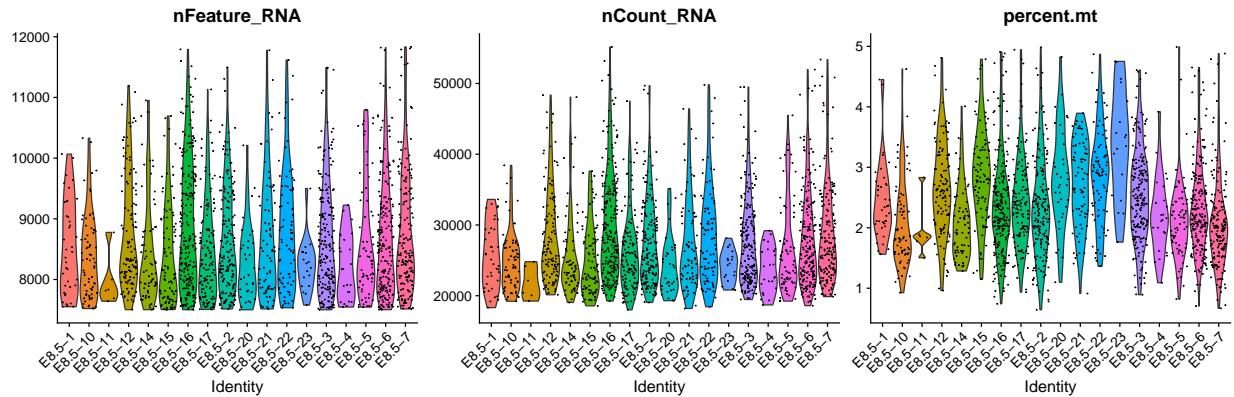


cnt\_ftr heeft een correlatie van 0.97 wat aantoont dat het een sterke correlatie heeft, aannemelijk is dat de data zuiver is en niet ontbreekt of vervuild is met andere data. cnt\_mt heeft een correlatie van -0.06 wat aantoont dat het een negatief lage correlatie heeft, we kunnen aannemen dat de dataset weinig vervuild is met mitochondriaal materiaal.

#### Step 4: Normalize and scale data

Data wordt geoptimaliseerd voor betere processing, afkapwaarde zijn vanaf >2000 en maximaal <12000 features, percentaal mitochondriaal is < 5 % Dit ter optimalisatie van de dataset, preprocessing voor de volgende stappen, mindere belasting voor de server en een zo groot mogelijk bereik mee te nemen. Data wordt genormaliseerd met de LogNormalize functie met een factor 10000, hiervoor gekozen omdat dit standaard wordt aangegeven in de Seurat analyse.

```
## Normalizing layer: counts
```



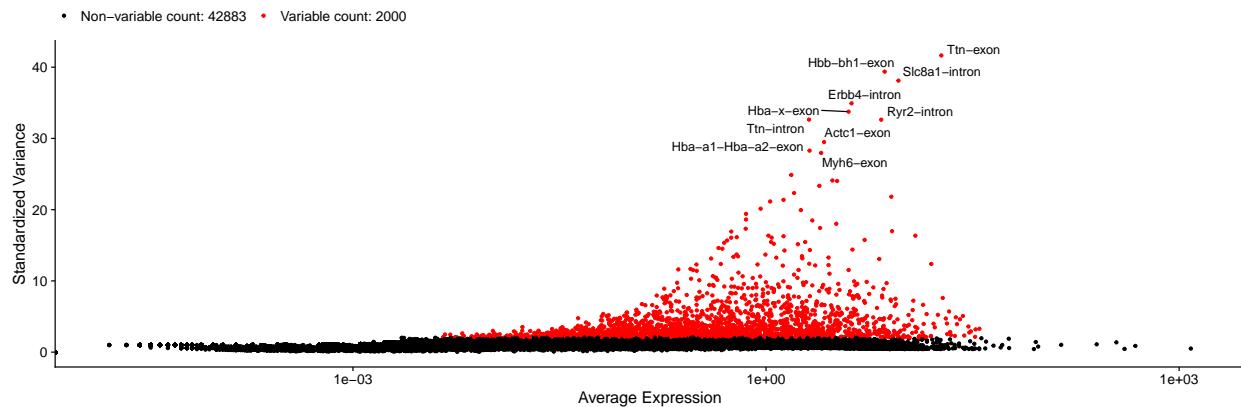
Violinplot die de verhouding weergeeft van de data aan de hand van de volgende features x-as= datalabel, y-as= aantal : - nFeature\_RNA; het aantal genen in de dataset - nCount\_RNA; het aantal moleculen in de dataset - percent.mt; het percentage mitochondriaal genexpressie

Opnieuw gegeneerd een violingplot dit ter aantoning dat de preprocessing goed verlopen is.

### Step 5: identifying top genes

Het identificeren van genen uit de dataset en een selectie meenemen naar de volgende analyse stap. Selectiemethode is “vst” dat de data op basis van logaritmische variatie standaardiseert en hiermee een verband schept tussen de data en gen met de hoogste expressie. Gekozen voor een aantal van 2000 hoogste variabelen ter ontlasting van de server en na overleg met opdrachtgever dat dit een standaard keuze is om mee te nemen. Echter betekent dat er wel 42883 geexcluseerd worden, dit vanwege het bereik van de dataset.

```
## Finding variable features for layer counts
## Warning in scale_x_log10(): log-10 transformation introduced infinite values.
```



Plot de variabele waarden van de genexpressie. X-as = average expression, y-as is standaard variatie. - De zwarte stippen staan voor de variabelen die niet variabel zijn en dus niet mee zullen genomen naar de volgende processing. Voordeel ontlasting van de server, nadeel minder data mee. Echter vallen deze in de lage expressie radius en willen we vooral kijken naar de hoge expressie. - De rode stippen zijn de variabelen van de 2000 hoogste expressie. Daarvan zijn de 10 met de hoogste expressie weergeven in de plot.

## Step 6: scaling genes

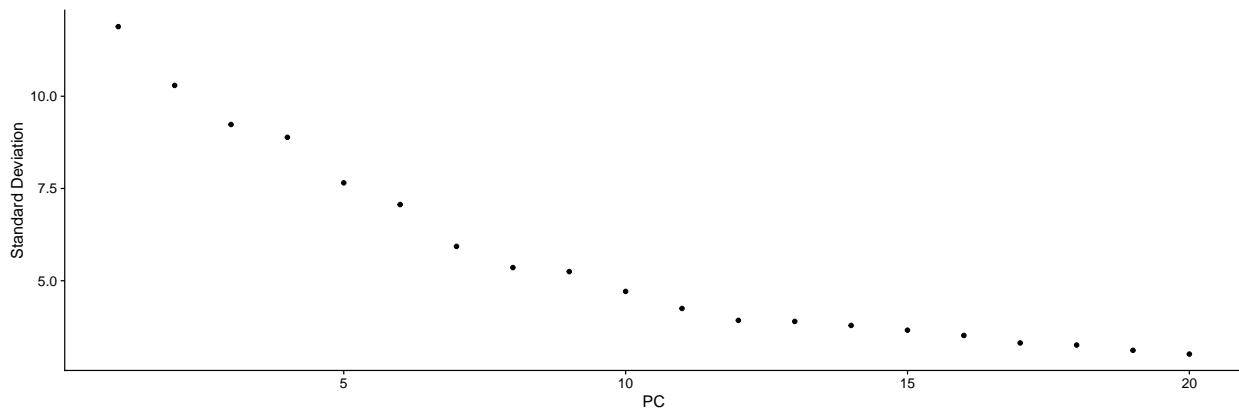
Genen worden op basis van waarden met de geanalyseerde data gescaled. Dit betekent dat de genen die voorkomen in de set aan de data gelinkt worden.

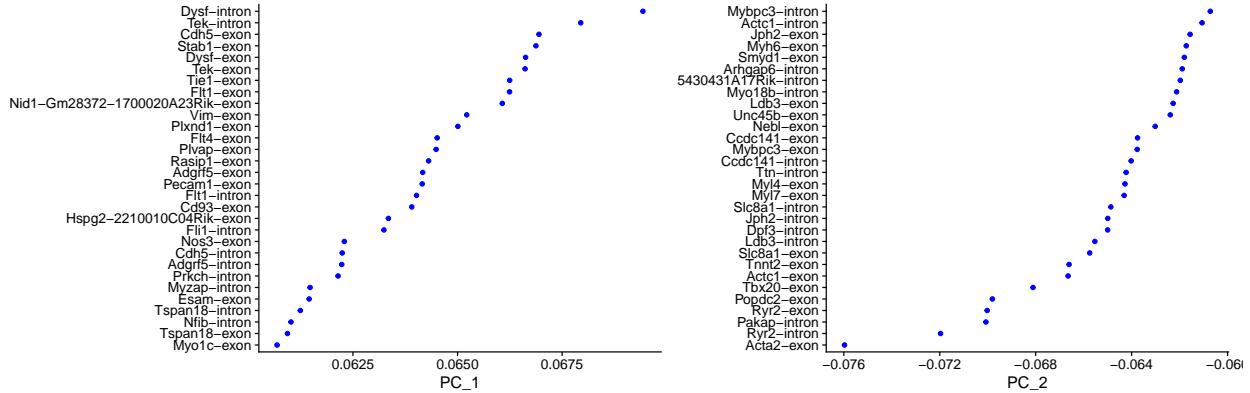
```
## Centering and scaling data matrix
```

## Step 7: PCA analysis & UMAP

De data verdelen in componenten op basis van de grootste variatie middels PCA analyse. De data visualiseren middels UMAP en DIMplot. Deelvraag: hoe PCA analyse te controleren, dit kan middels een SCREE plot en een dimplot om aan te tonen dat data daadwerkelijk in componenten gedeeld is en genen geanalyseerd zijn. (Waarde geven)

```
## PC_ 1
## Positive: Dysf-intron, Tek-intron, Cdh5-exon, Stab1-exon, Dysf-exon
## Negative: Nr6a1-intron, Cadm1-intron, Tenm3-Nxt1-intron, Tenm4-intron, Mllt3-intron
## PC_ 2
## Positive: Arap3-exon, Ldb2-intron, Rapgef5-intron, Tmem164-intron, Cdh5-exon
## Negative: Acta2-exon, Ryr2-intron, Pakap-intron, Ryr2-exon, Popdc2-exon
## PC_ 3
## Positive: Zfp423-intron, Zeb1-intron, Auts2-intron, Cdh2-intron, Maml3-intron
## Negative: Apoal1-exon, Rbp4-exon, Amn-exon, Apom-exon, Apoc2-Gm44805-exon
## PC_ 4
## Positive: Bnc2-intron, Dlk1-exon, Col1a2-exon, Pmp22-exon, Svep1-exon
## Negative: Mybpc3-exon, Myh6-exon, Ttn-intron, Myl4-exon, Myo18b-intron
## PC_ 5
## Positive: Efna5-intron, Col18a1-exon, Zfp423-intron, Tcf7l2-intron, Nhs11-intron
## Negative: Hbb-bh1-exon, Spta1-exon, Alas2-exon, Hba-a1-Hba-a2-exon, Ank1-intron
```



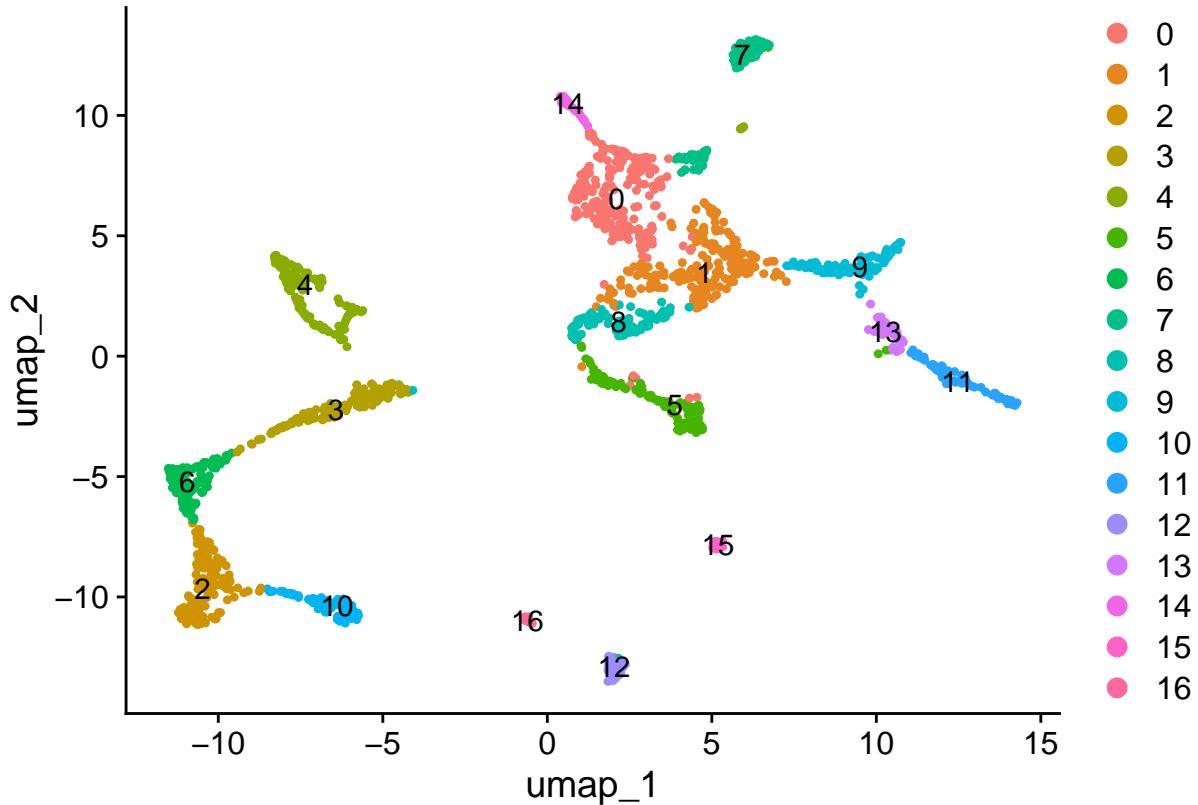


Elbowplot: Data opgedeeld in 20 PCA componenten: x-as = componenten aantal, y-as= standaard deviatie (grootste variatie) Print Seurat: Van 5 PCA componentende hoogste positieve waarden en hoogste negatieve waarden weergeven. VizDimloading: x-as= negatieve waarden & PC\_component, y-as = naam gen/extron/intron

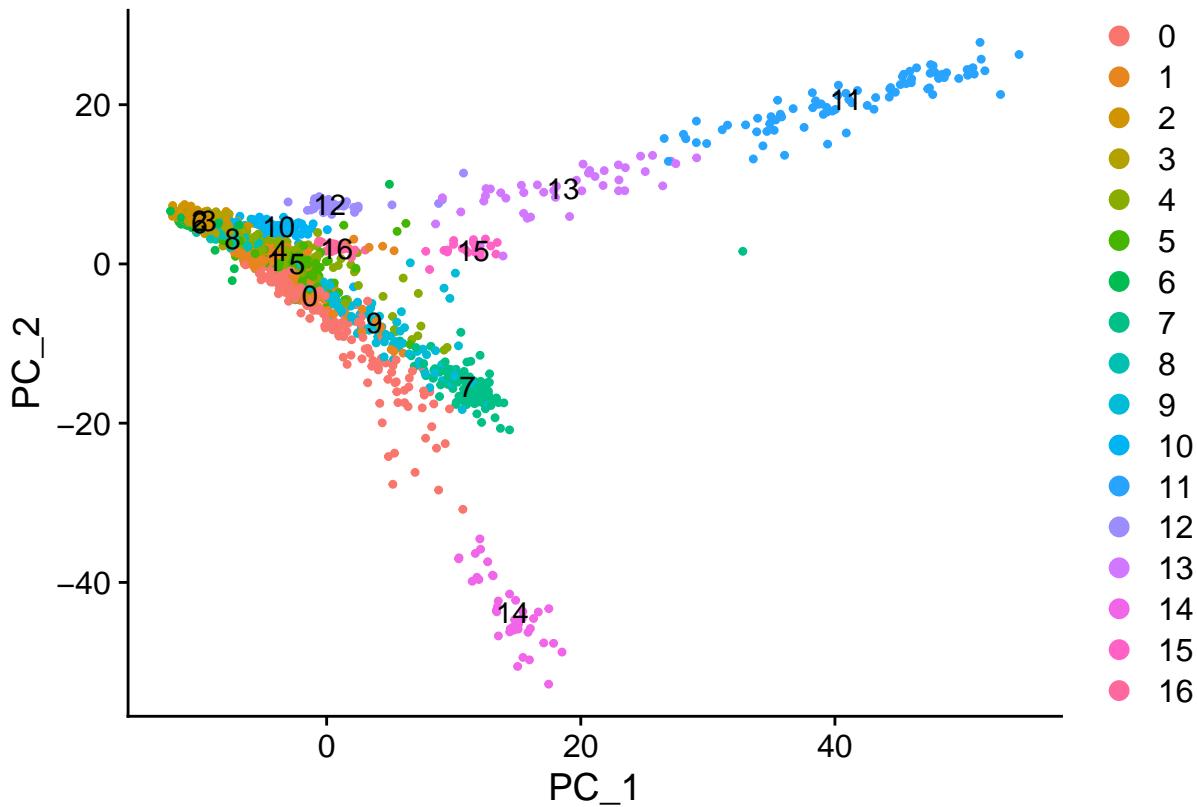
Dit ter controle van de PCA analyse en of het overeenkomt/geen overlappingen zijn.

Visualiseren van de PCA middels UMAP methode, dit om aan te tonen dat data geclusterd is.

```
## Warning: The default method for RunUMAP has changed from calling Python UMAP via reticulate to the R
## To use Python UMAP via reticulate, set umap.method to 'umap-learn' and metric to 'correlation'
## This message will be shown once per session
```



Visualisatie van de PCA analyse met UMAP methode, nieuwe berekening laat clusters zien. Bedoeld voor snelle weergave van de verdeling. Legenda van 26 clusters x-as= umap\_1 y-as= umap\_2; heeft geen definitieve betekenis, is alleen om een indicatie te geven hoe de clusters zich verhouden. Geen definitieve waarden.



Dimplot waarvan x-as= PC\_1 en y-as= PC\_2. Twee PC componenten uitgeplot tegenover elkaar om overlapping/verschillen aan te tonen.