

# Seurat tutorial

Anne Brussaard

2025-06-12

## 1 Seurat tutorial

Voor dit project is het belangrijk om kennis te maken met de packages Seurat zodat op de juiste manier gebruik gemaakt kan worden van Seurat. Hieruit is de volgende deelvraag opgesteld.

### 1.0.1 Deelvraag: Kan ik door het volgen van een tutorial met Seurat data preprocessing en visualisatie uitvoeren op de uitgereikte data?

Om deze deelvraag te beantwoorden zal het volgende flowschema aangehouden worden. 1. De data wordt geladen 2. Er wordt een Seurat object gemaakt 3. De filterstappen worden uitgevoerd 4. Clusters worden visueel gemaakt

De commands worden uitgevoerd in het environment project\_brie2. project\_brie2 geeft de mogelijkheid om te werken met seurat-4.4.0. Dit script wordt vervolgens gemaakt via de terminal waarin R wordt geopend met het command `rmarkdown::render()`

### 1.1 Tutorial

Deze tutorial is afkomstig van de github van de maker van Seurat. ([https://satijalab.org/seurat/articles/pbmc3k\\_tutorial.html](https://satijalab.org/seurat/articles/pbmc3k_tutorial.html)) In deze tutorial zal een data set van PBMC cellen die gesequenced zijn geanalyseerd worden met Seurat.

Stap 1.1: Voor we kunnen beginnen wordt gecontroleerd of de packages die we nodig hebben zijn geïnstalleerd.

Stap 1.2: Eerst worden de packages geladen. Seurat voor analyse, dplyr voor filteren en selecteren en patchwork voor maken van plots.

Stap 1.3: De data wordt ingeladen de data is gedownload van `vignettes/pbmc3k_tutorial`. Het is een data set van PBMC cellen waarbij 2700 losse cellen zijn gesequenced met Illumina Next Seq 500. Deze data wordt ingeladen.

Stap 2.1: Er wordt een Seurat object gemaakt. De data wordt vervolgens met behulp van de packages seurat opgeslagen als object `e` zodat dit gebruikt kan worden voor preprocessing en data analyse.

Stap 2.2: De 0 values worden weg gefilterd en een compactere versie van het object wordt opgeslagen. Deze worden namelijk niet gebruikt en nemen wel veel geheugen in beslag.

Stap 3.1: Kwaliteits check uitvoeren en cellen selecteren voor analyse. Om de betrouwbaarheid van de analyse te waarborgen wordt eerst gekeken naar welke cellen meegenomen worden voor analyse. Daarvoor wordt eerst het percentage mitochondriale RNA toegevoegd aan het seurat object die los is aangeleverd vanuit de tutorial.

Stap 3.2: Om de kwaliteit van de data te beoordelen wordt deze gevisualiseerd in een violin plot van `nFeature_RNA` (unieke genen per cel), `nCount_RNA` (totaal aantal moleculen), en `percent.mt` (mitochondriale

expressie). Een laag aantal unieke genen of moleculen kan namelijk wijzen op lege of slechte kwaliteit van de cellen. En te hoog kan wijzen op dubbele metingen. Het mitochondriaal percentage wordt gebruikt om te kijken naar dode of slechte kwaliteit van cellen, een te hoog percentage wijst hier namelijk op. In deze data ligt de mitochondriale expressie voor de meeste cellen bij 5%. De cellen die daar boven liggen worden dus als minder betrouwbaar gezien. In deze violin plot wordt dit weergegeven en is te zien hoe de data verdeeld is. op de Y-as wordt het aantal RNA, moleculen of percentage MT weergegeven, op de X-as staat het data set. Op het breedste punt zijn de meeste cellen te zien maar ook hoger in het figuur zijn cellen te zien. Voor de kwaliteit van het data set is het belangrijk om te bepalen of dit afwijkende waarden zijn of dat deze moeten worden uitgesloten van verdere analyse. Optimaal is om zo veel mogelijk van deze data mee te nemen.

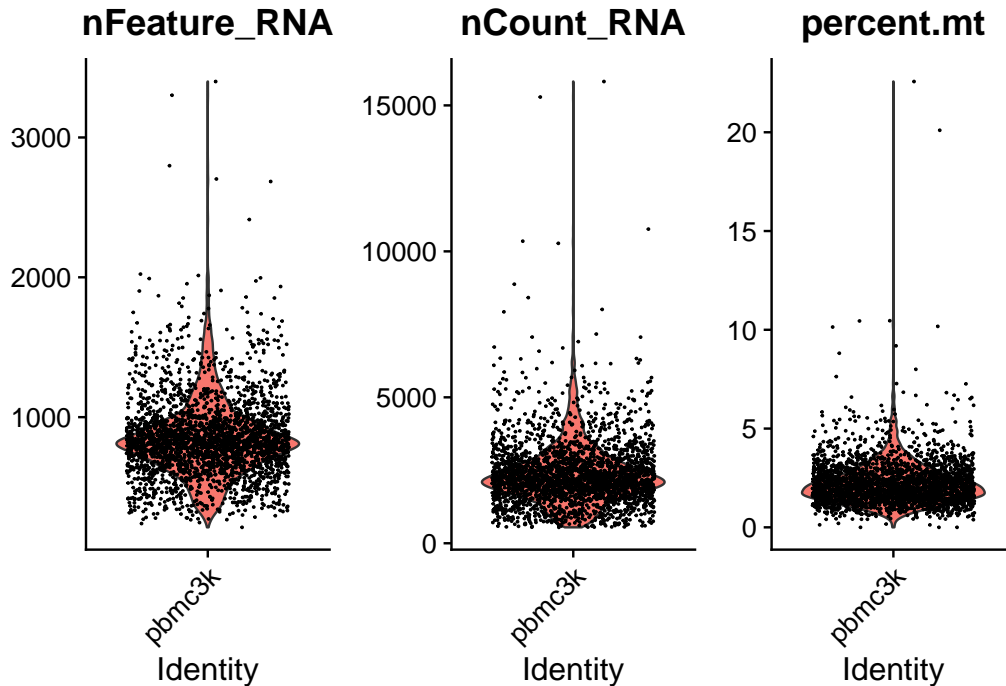


Figure 1: totale dataset in violin plot

Stap 3.3: Visualisatie in Feature Scatter. Om de relatie tussen de nCount en percent.mt of nFeature aan te geven wordt de pearson correlatie geanalyseerd. deze is voor nCount en nFeature 0.95 wat aangeeft dat het een sterke correlatie heeft. Voor nCount en percent.mt is deze -0.13 wat wijst op geen directe correlatie. De correlatie tussen nFeature en percentage.mt wordt hierin niet meegenomen omdat een hoog percentage MT kan samengaan met een zowel hoog of laag nFeature terwijl slechte kwaliteit vaak een hoge MT en een lage nCount heeft. Daarom is de correlatie van nFeature met percentage MT minder informatief over de kwaliteit. In deze afbeelding zien we op de Y-as het percentage MT en nFeature tegen nCount op de X-as.

Stap 3.4: Filtering Voor de filtering is bij de tutorial is gekozen voor >200 gene expression (lage of lege droplets hebben vaak weinig genen), <2500 cellen met veel genen (dubbel getelde droplets hebben vaak hoge genen), <5 mitochondriale expressie (hogere MT expressie komt vaak door lage kwaliteit van de cel)

Stap 3.5: Normalisatie van data volgens standaard normalisatie. Er is een veel gebruikte schaling methode gebruikt waarbij de feature expressie van iedere cel genormaliseerd wordt voor de normale expressie. Zodat bij analyse gerekend wordt met het zelfde aantal RNA moleculen voor iedere cel. Dit wordt gedaan door daarna te schalen met factor 10000 en log te transformeren. De log transformatie is nodig om te zorgen dat genen met hoge counts vergelijkbaar is met genen met lage counts. Ook helpt het bij het normaal verdelen van de data voor PCA analyse.

Stap 3.6: Feature selection. De genen die veel verschil in expressie hebben per cel worden geselecteerd. Uit

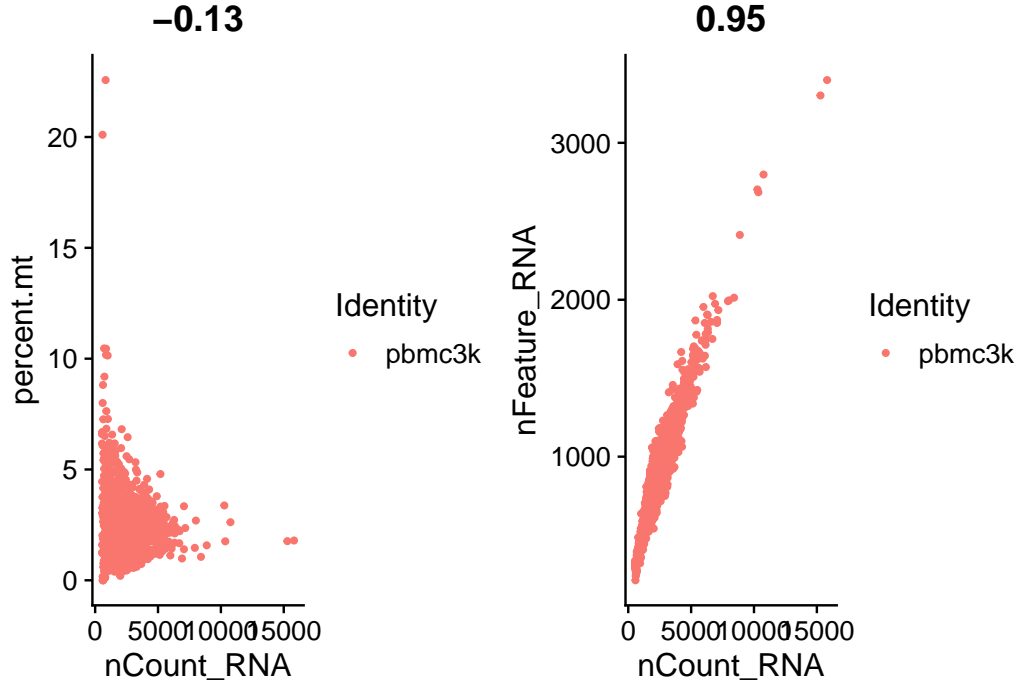


Figure 2: correlatie dataset

eerder onderzoek is namelijk gebleken dat focus op deze genen helpt bij downstream analyse van biologisch signaal in single cell datasets. Er zijn hier 2000 features voor een data set om de analyse werkbaar te houden voor de server.

Stap 3.6. De 10 meest variabele genen worden geselecteerd.

Stap 3.7. De 10 meest variabele genen worden geplot. Hierin worden zijn de zwarte stippen die niet variabel zijn en niet meegenomen worden bij verdere analyse. De rode stippen zijn de variabele genen en de top tien meest variabele genen zijn aangegeven met een label. Op de Y-as wordt de variatie (verschil in expressie) weergegeven, en op de X-as de gemiddelde expressie.

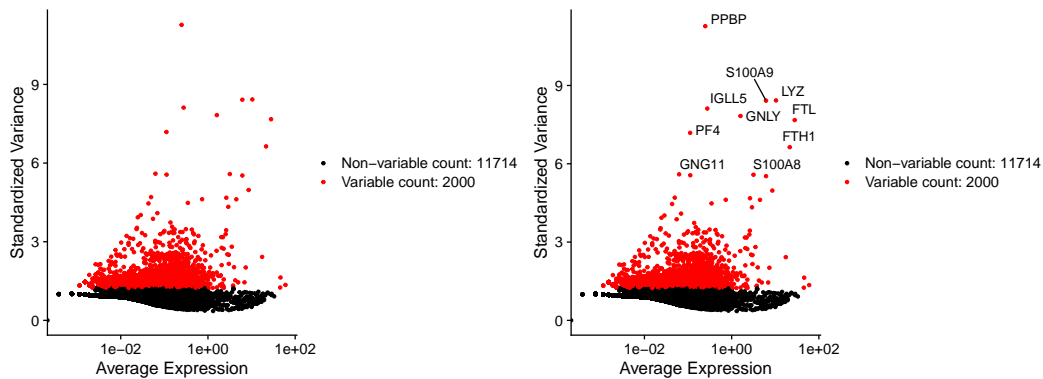


Figure 3: Variable genen

Stap 3.8. De data wordt geschaalt volgens standaard procedures voor PCA analyse. Hierbij wordt de expressie van alle genen gelijk zodat de hoeveelheid expressie geen invloed heeft in de verdere analyse.

Stap 3.9. De PCA analyse wordt uitgevoerd en gevisualiseerd. Dit wordt gedaan om de data te analyseren op genexpressie patronen. Daarna kan dan bepaald worden hoeveel van deze patronen worden meegenomen in verdere analyse.

Deze PCA analyse kan op verschillende manieren worden gevisualiseerd. Hieronder zullen een aantal voorbeelden langskomen. Als eerste een VizDim waarin de genen die verantwoordelijk zijn voor een PC worden weergegeven. Op de Y-as worden de namen van de genen weergegeven en op de X-as wordt loading waarden weergegeven. Dit betekent hoeveel het gen bijdraagt aan een PC. Zo kan je bijvoorbeeld kijken of PC's genen bevatten die biologisch relevant zijn.

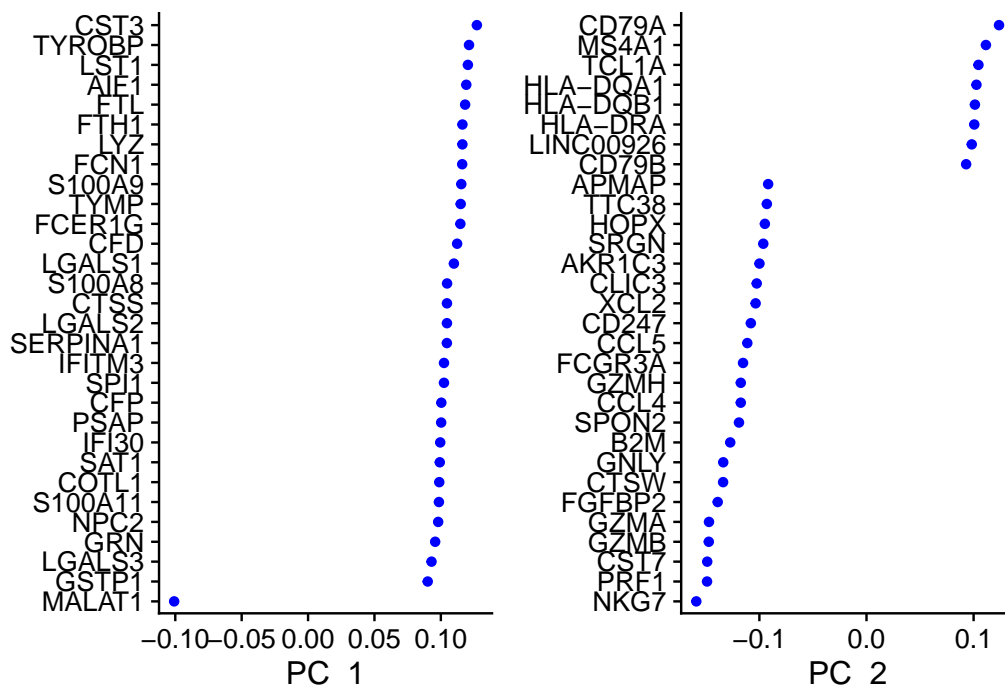


Figure 4: VizDim plot PCA analyse

Dan volgt de Dimplot. Hierin wordt de data structuur weergegeven. Op de Y-as staat tweede hoofdmentie en op de X-as de eerste hoofdmentie. In deze weergaven is iedere stip 1 cel. Hieruit kan gezien worden of de PC goed van elkaar gescheiden zijn of juist overlappen.

Als laatste wordt de Dim Heatmap weergegeven. Deze visualisatie geeft een overzicht van welke genen belangrijk zijn voor welk PC en hoe deze tot expressie komen in alle genen. De rijen zijn de genen met hoge loading (bijdragen PC) en de kolommen zijn de cellen. De kleur gele kleur laat de expressie zien. Deze Heatmap kan laten zien welke genen invloed hebben op de PC's, kijken of er bepaalde clusters of celtypes een verschillend patroon hebben of welke PC je wilt gebruiken voor verdere analyse.

Stap 3.10. De dimentie van het dataset wordt bepaald. Hierin worden de PC's weergegeven. Vanuit deze afbeelding wordt bepaald welke soorten expressie patronen er zijn binnen het data set. De elbowplot wordt vervolgens gebruikt om te bepalen welke PC's er worden meegenomen in verdere analyse. Op basis van deze ElbowPlot zijn de eerste 10 PC's geselecteerd omdat de "elbow" stopt rond PC9-10, wat wijst op een signaal in de eerste 10 PC's.

Stap 3.11. De cellen worden geclusterd op basis van de eerste 10 PC's.

Stap 4.1 De UMAP maken. De gevormde clusters worden weergegeven aan de hand van 10 PC's. In deze UMAP wordt weergegeven welke clusters er zijn en hoe deze zijn verdeeld.

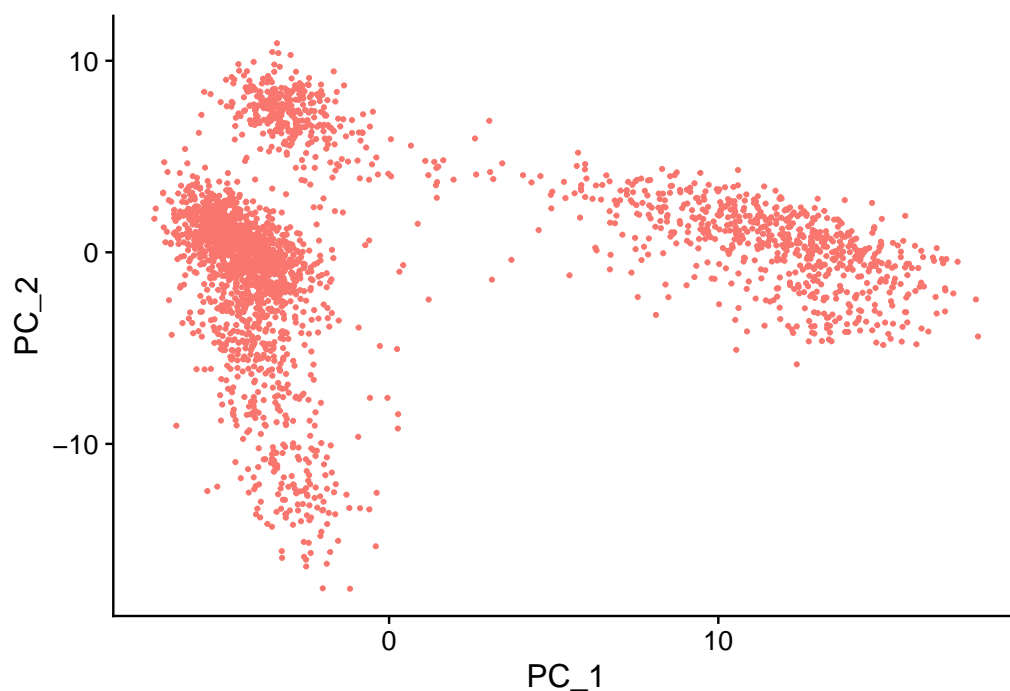


Figure 5: Dimplot PCA analyse

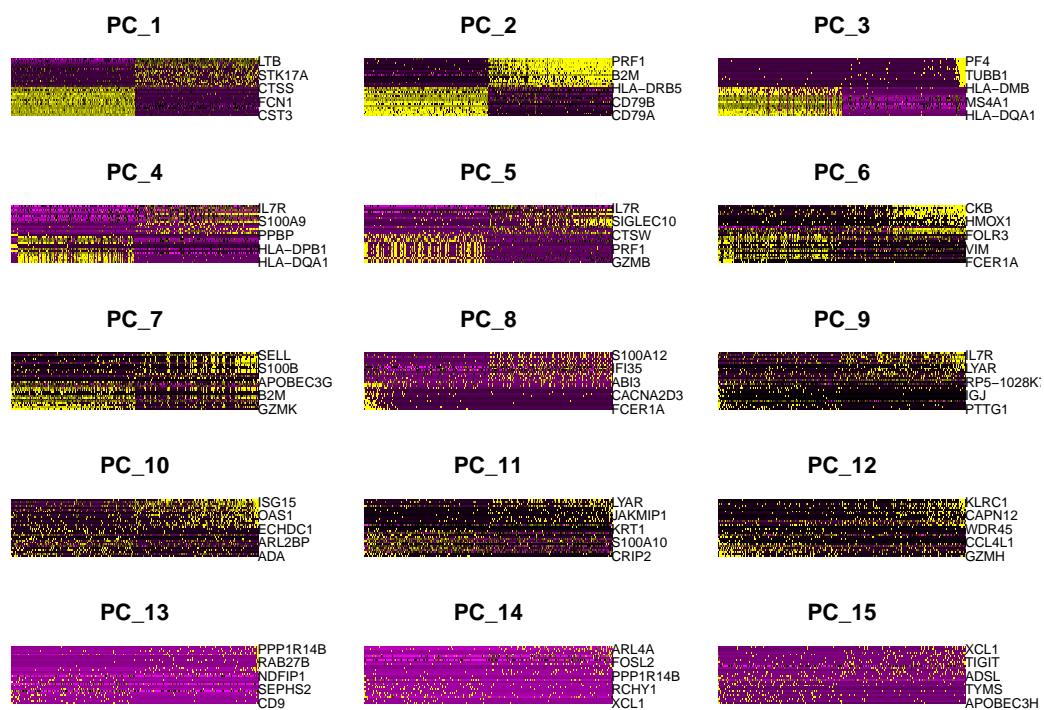


Figure 6: Heatmap PCA analyse

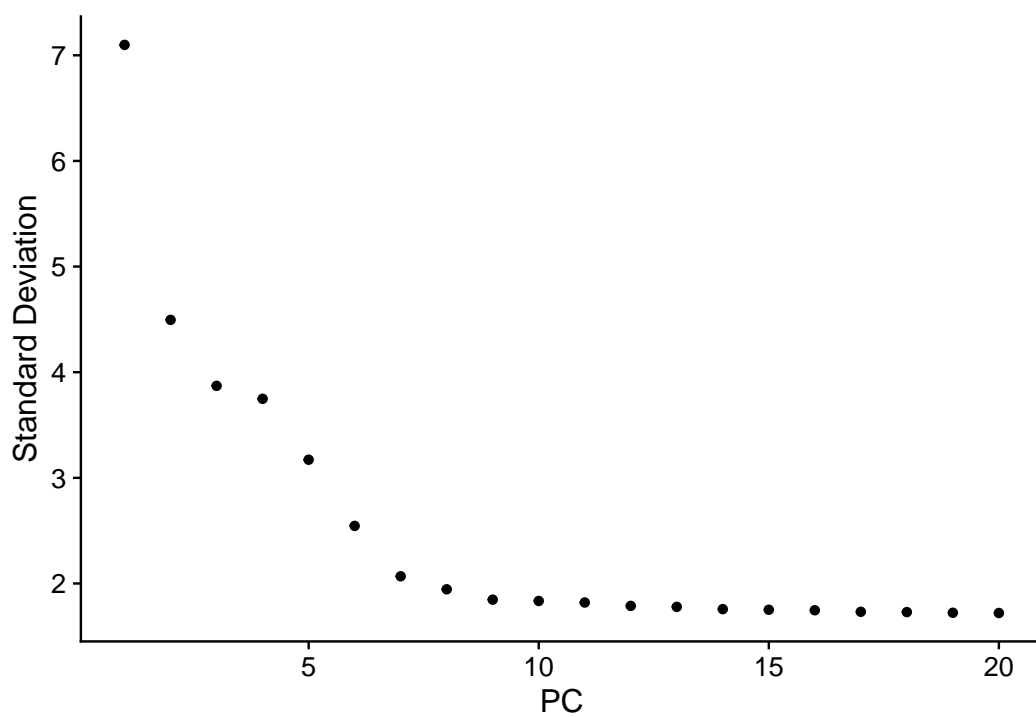


Figure 7: Elbowplot PC's

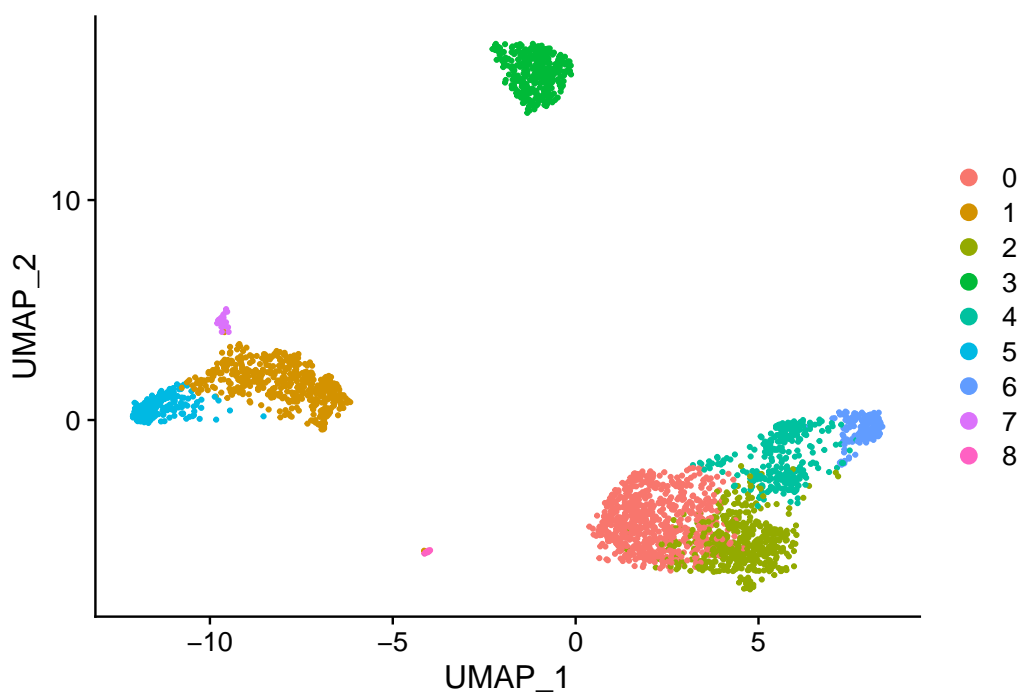
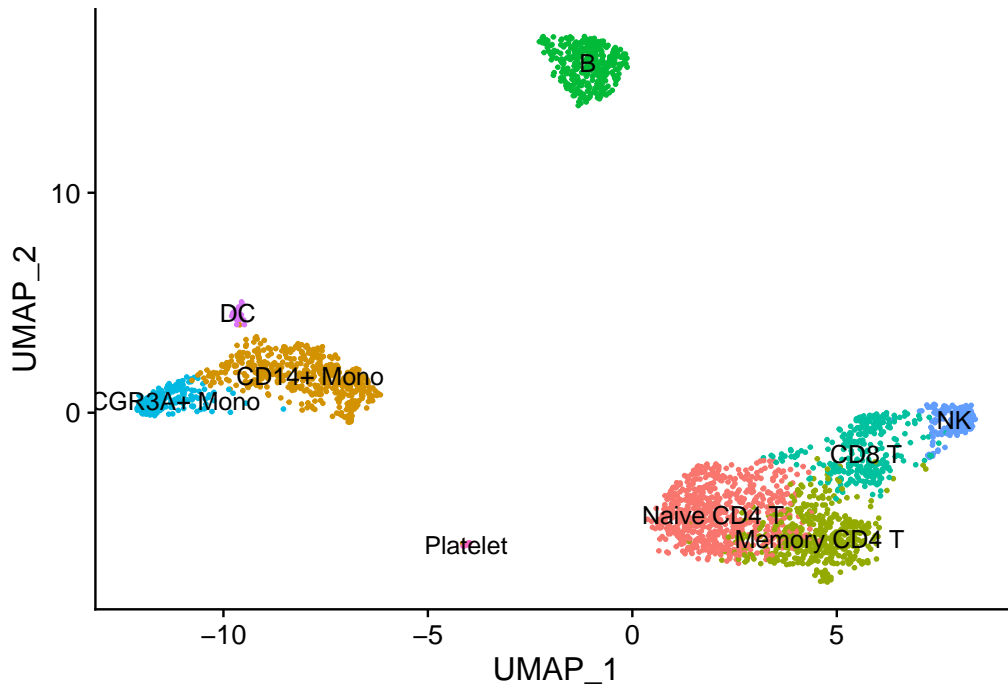


Figure 8: UMAP PBMC tutorial





Stap 5.4. Er wordt een UMAP gemaakt waarin wordt aangegeven welk cluster welk celtype is.

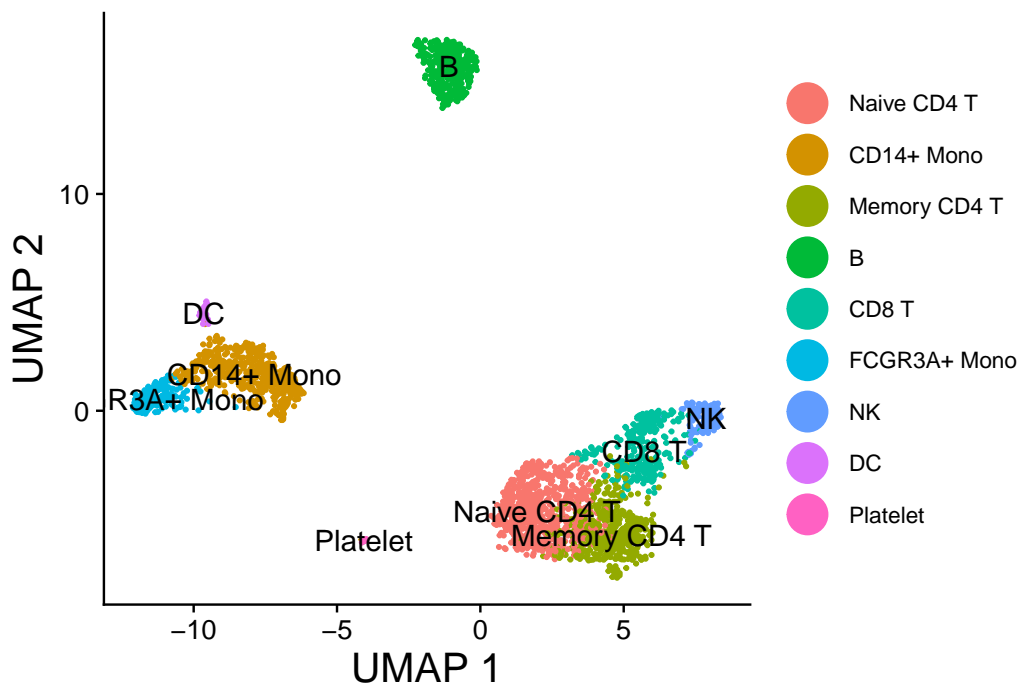


Figure 10: UMAP met celtype

Conclusie: Het is gelukt om door het volgen van de tutorial de preprocessing en visualisatie van de aangereikte data uit te voeren. Op deze manier is kennis opgedaan welke manieren van analyseren en visualiseren



mogelijk zijn in Seurat. Dit kan verder gebruikt worden voor het analyseren van de eigen data.