Seurat Tutorial - Verwerken van single-cell data als voorbereiding op experimenteel ontwerp

Jalisa van der Zeeuw

2025-06-01

Inleiding

In dit document leg ik mijn leerproces vast rondom het gebruik van Seurat voor het verwerken van scRNA-sequencing data. Dit is gebaseerd op een online tutorial en dient als voorbereiding op het toepassen van Seurat binnen mijn eigen project.

Doel van de tutorial

Het doel van deze tutorial is om basiskennis en ervaring op te doen met de bioinformatica-tool Seurat, die wordt gebruikt voor de analyse van single-cell RNA-sequencing data.

Aangezien ik voorafgaand aan dit project geen ervaring had met Seurat, volg ik deze tutorial stap voor stap om inzicht te krijgen in de volledige workflow: van het inladen en voorbewerken van data tot aan de visualisatie en clustering.

Deze opgedane kennis en vaardigheden wil ik uiteindelijk toepassen in mijn eigen project.

Stappen van de workflow

Hier worden de belangrijkste stappen uit de Seurat-tutorial beschreven. Per stap wordt kort beschreven wat het doel is, welke commando's worden gebruikt, en wat de resultaten betekenen.

Inladen van data

In deze stap wordt een single-cell dataset van *Peripheral Blood Mononuclear Cells (PBMC)*, afkomstig van **10X Genomics** ingeladen. De data wordt ingelezen met de `Read10X()` functie. Dit levert een UMI-count matrix op met het aantal getelde RNA-moleculen per gen per cel. Daarna maak ik een Seurat-object aan met `CreateSeuratObject()`. Dit object slaat zowel de ruwe data als de resultaten van nog te komen analyses op. Dit is belangrijk om de ruwe data te kunnen gebruiken voor kwaliteitscontrole, normalisatie en clustering.

Bekijk samenvatting van het object pbmc

Standaard pre-processing workflow

Kwaliteitscontrole (QC) en selectie van cellen voor verdere analyse

In deze stap voeren we kwaliteitscontrole uit om slechte of afwijkende cellen te identificeren en uit te sluiten.

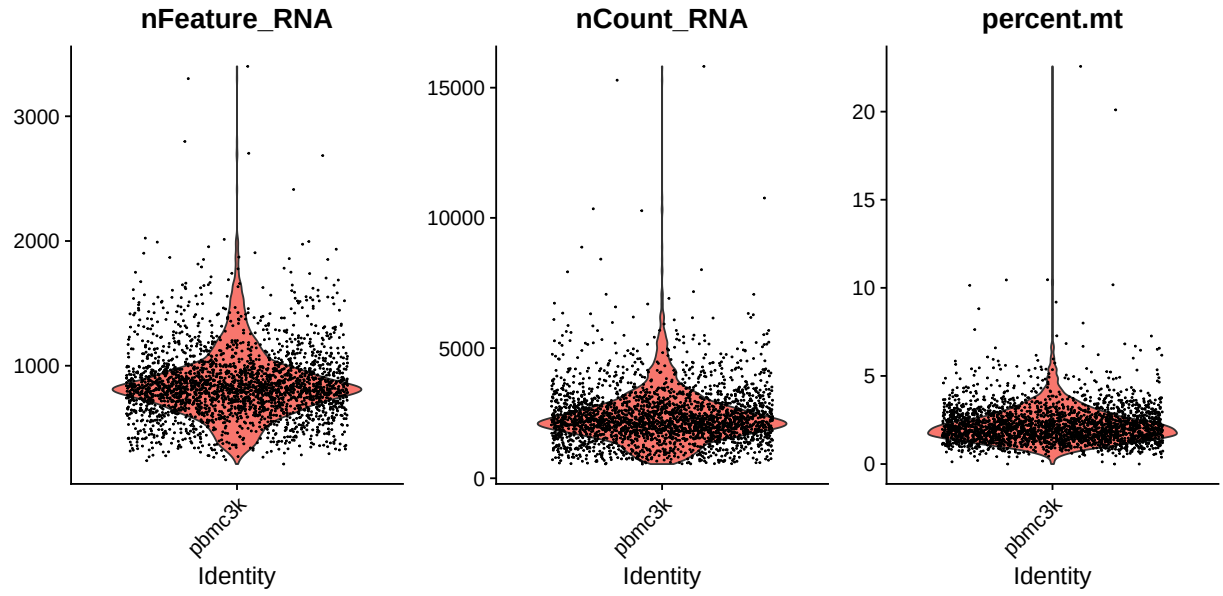
- Het aantal unieke genen per cel: cellen met te weinig genen kunnen slechte kwaliteit zijn, terwijl cellen met extreem veel genen mogelijk doublets zijn.
- Het totaal aantal moleculen per cel, wat sterk correleert met unieke genen
- Het percentage mitochondriale genen, wat een indicatie kan zijn van dode of stervende cellen

We gebruiken de functie `PercentageFeatureSet()` om het precentage mitochondriale expressie te berekenen, door te zoeken naar genen die beginnen met "MT-".

```
##                 orig.ident nCount_RNA nFeature_RNA percent.mt
## AAACATACAACCAC-1     pbmc3k       2419          779  3.0177759
## AAACATTGAGCTAC-1     pbmc3k       4903         1352  3.7935958
## AAACATTGATCAGC-1     pbmc3k       3147         1129  0.8897363
## AAACCGTGCTTCCG-1     pbmc3k       2639          960  1.7430845
## AAACCGTGTATGCG-1     pbmc3k        980          521  1.2244898
## AAACGCACTGGTAC-1     pbmc3k       2163          781  1.6643551
```

Visualisatie van QC-metrics Om te beoordelen welke cellen geschikt zijn voor verdere analyse, visualiseren we de distributie van de belangrijke kwaliteitscontroles:

- nFeature_RNA (aantal unieke genen per cel)
- nCount_RNA (totaal aantal moleculen per cel)
- percent.mt (percentage mitochondriale genen)

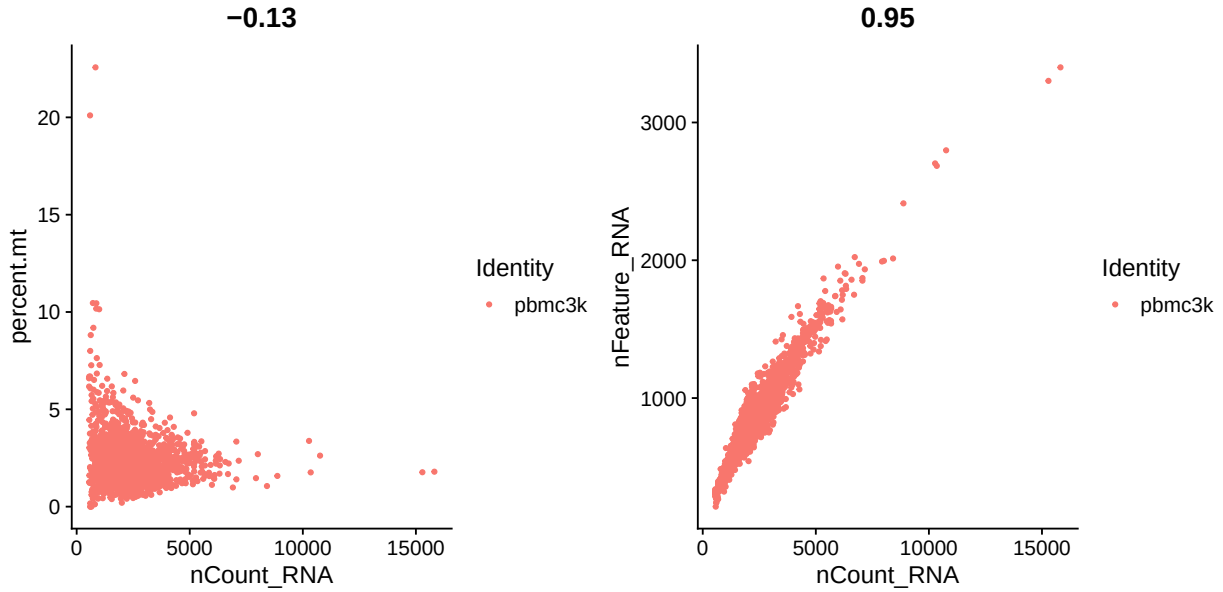


Filteren van cellen op basis van kwaliteitscriteria Nu visualiseren we de relatie tussen de QC-statistieken om outliers beter te kunnen identificeren. Daarna filteren we op basis van de volgende drempelwaardes:

- Cellen met minder dan 200 of meer dan 2500 unieke genen worden verwijderd.

- Cellen met meer dan 5% mitochondriale genen worden uitgesloten.

We gebruiken functie `FeatureScatter()` om de verbanden te visualiseren



Vervolgens filteren we de dataset met `subset()`. Het oorspronkelijke object (pbmc) wordt behouden indien nodig om te vergelijken. De gefilterde data wordt opgeslagen in pbmc.filtered en gebruiken we voor verdere analyse.

Normaliseren van de data

Na het verwijderen van ongewenste cellen moet de dataset worden genormaliseerd. Dit is belangrijk omdat de ruwe tellingen verschillen tussen cellen door bijvoorbeeld sequencing diepte. Normalisatie maakt de data beter vergelijkbaar.

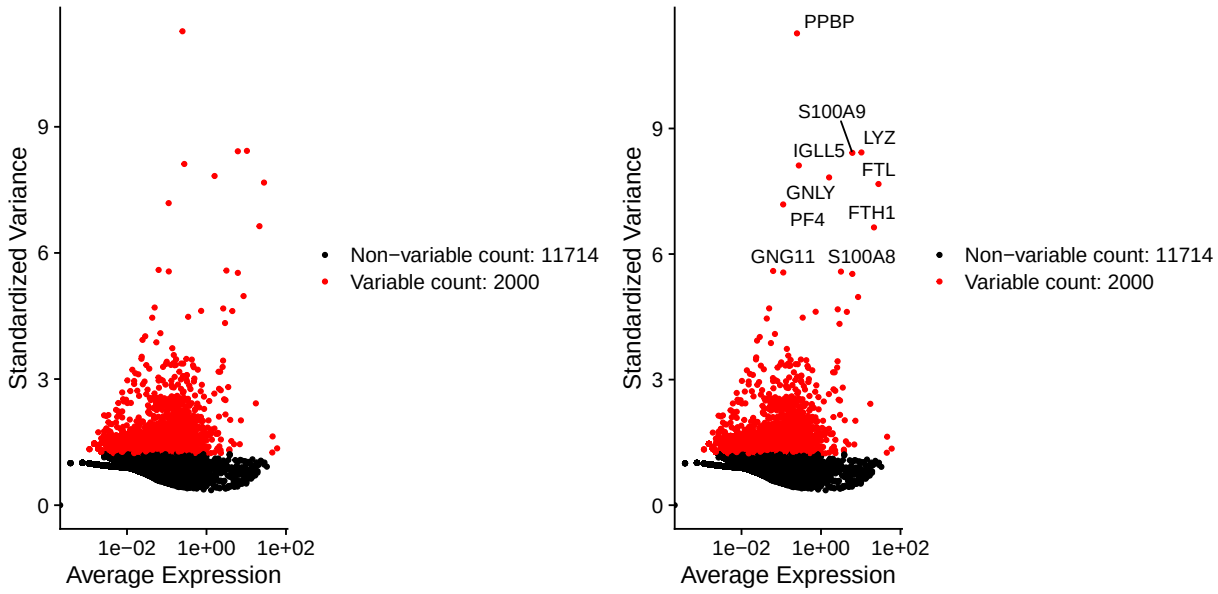
Standaard wordt er in Seurat gebruik gemaakt van de methode LogNormalize. Hierbij wordt voor elke cel het aantal getelde RNA-moleculen gedeeld door het totaal aantal moleculen in die cel, daarna wordt dit vermenigvuldigd met een schaalfactor (standaard 10.000) en vervolgens wordt de log-transformatie toegepast. Deze genormaliseerde waarden worden opgeslagen in pbmc[["RNA"]]$data.

Deze globale scaling aanname houdt in dat elke cel oorspronkelijk evenveel RNA moleculen bevat, wat niet altijd klopt. Rr bestaan alternatieve normalisatiemethoden zoals SCTransform() die deze aanname niet maken en die soms betere resultaten geven. Zie uitgebreide tutorial en paper van Seurat over deze methode. **Hier wil ik later nog op terugkomen**

Selecteren van hoog-variabele genen

In deze stap identificeren we genen die een grote variatie in expressie laten zien tussen verschillende cellen. Deze genen geven vaak het sterkste biologische signaal (bijvoorbeeld celtypeverschillen) en zijn daarom het meest informatief voor de downstream analyses zoals dimensional reduction en clustering.

De functie `FindVariableFeatures()` geeft de relatie tussen gemiddelde expressie en variatie, en selecteert standaard de top 2.000 meest variabele genen. Vervolgens visualieren we deze genen, inclusief de 10 meest variabele genen met `VariableFeaturePlot()` en `LabelPoints()`.

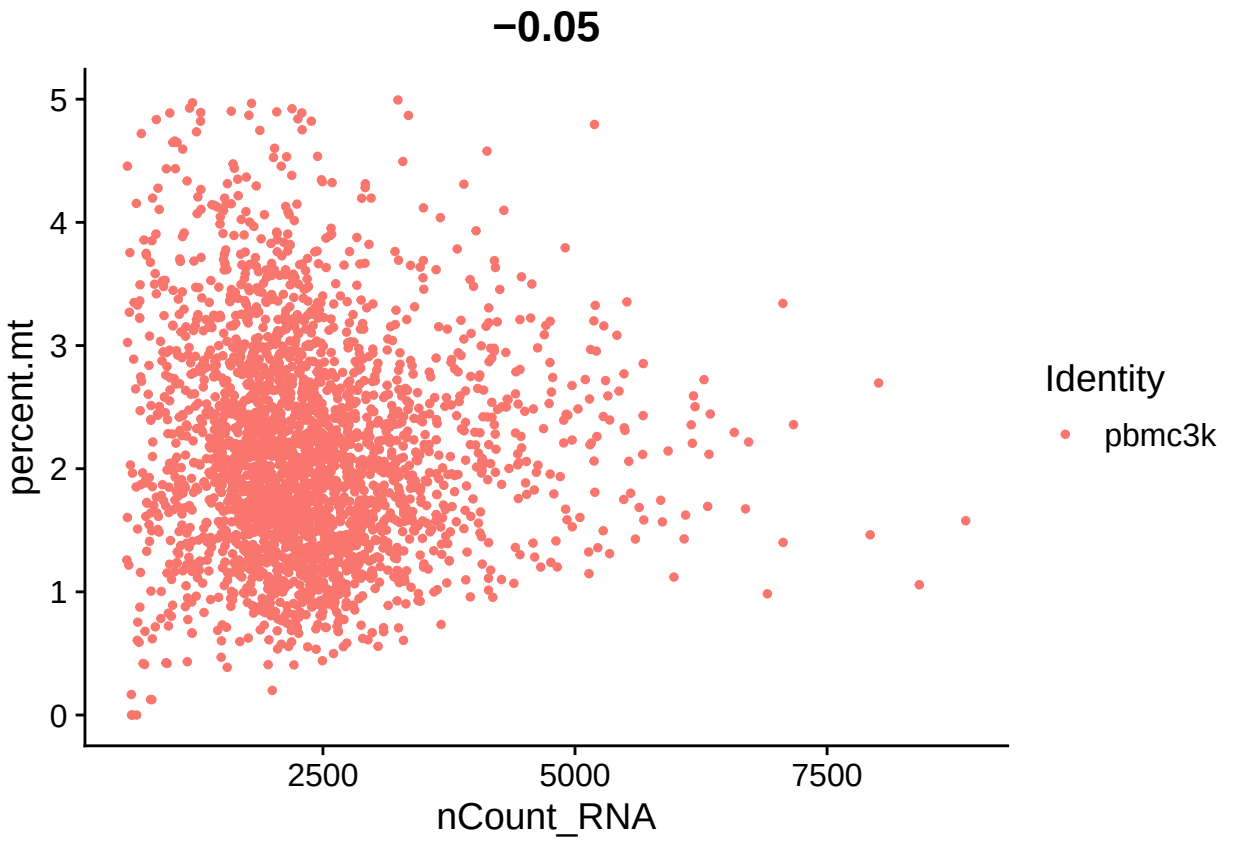


In deze scatterplot zie je op de x-as de gemiddelde expressie van genen, en op de y-as de genormaliseerde variantie. Genen die rechtsboven liggen in de plot zijn zowel hoog uitgedrukt als sterk variabel tussen cellen, en dus biologisch interessant.

Data schaling (normalisatie over genen)

In deze stap passen we lineaire schaaltransformatie toe, zodat elk gen een gemiddelde van 0 en een variantie van 1 heeft. Dit zorgt ervoor dat de downstream analyse zoals PCA niet wordt gedomineerd door genen die van nature sterk tot expressie komen.

Controle op technische variatie en beslissing over regressie Om de aanwezigheid van ongewenste technische variatie te beoordelen, is er onderzocht of het percentage mitchodonriale genexpressie (percent.mt) correleert met het aantal getelde moleculen per cel (nCount_RNA). Een sterke correlatie kan wijzen op technische ruis die de biologische signalen kan verstoren. Met behulp van een scatterplot wordt de relatie gevisualiseerd en de correlatiecoëfficiënt wordt berekend:




```
##
##  Pearson's product-moment correlation
##
## data:  pbmc.filtered$nCount_RNA and pbmc.filtered$percent.mt
## t = -2.3202, df = 2636, p-value = 0.0204
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.083165522 -0.006994103
## sample estimates:
##         cor
## -0.04514543
```

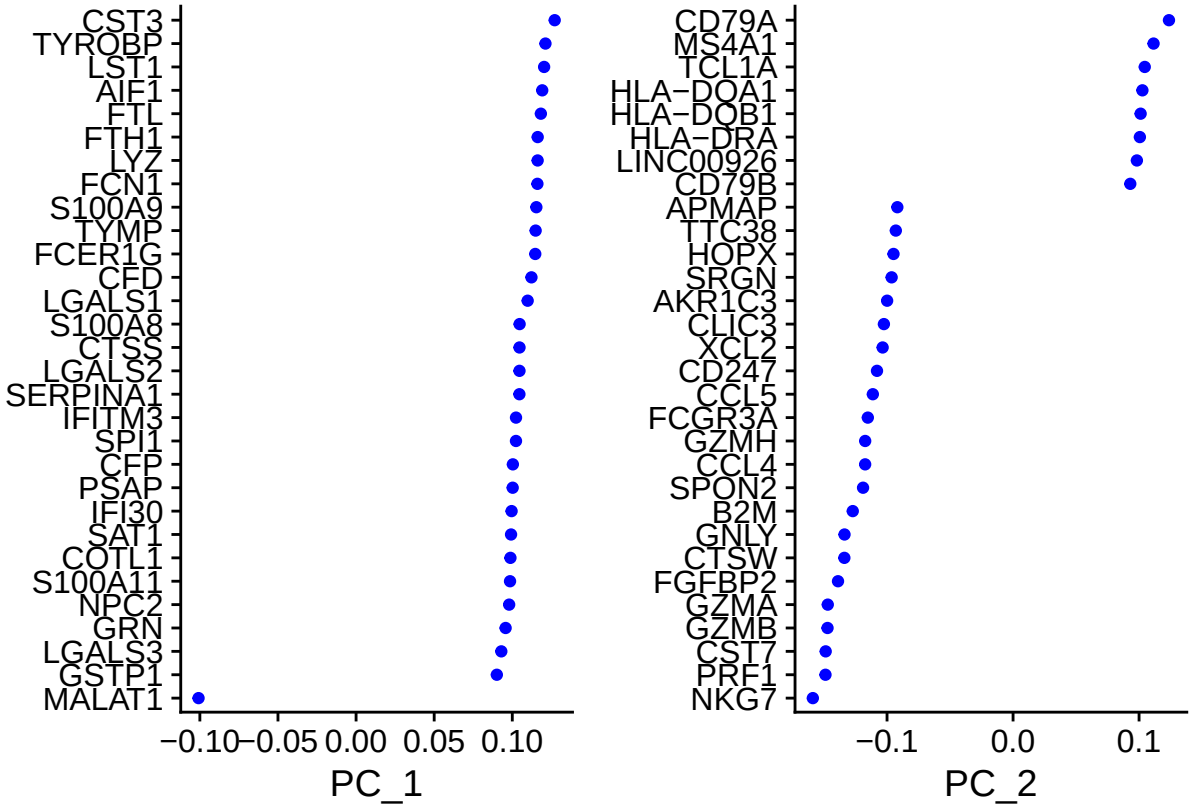
Uit de plot blijkt dat er geen duidelijk patroon zichtbaar is, en de correlatiecoefficient is zeer laag (r=-0.045), war wijst op afwezigheid van samenhang.

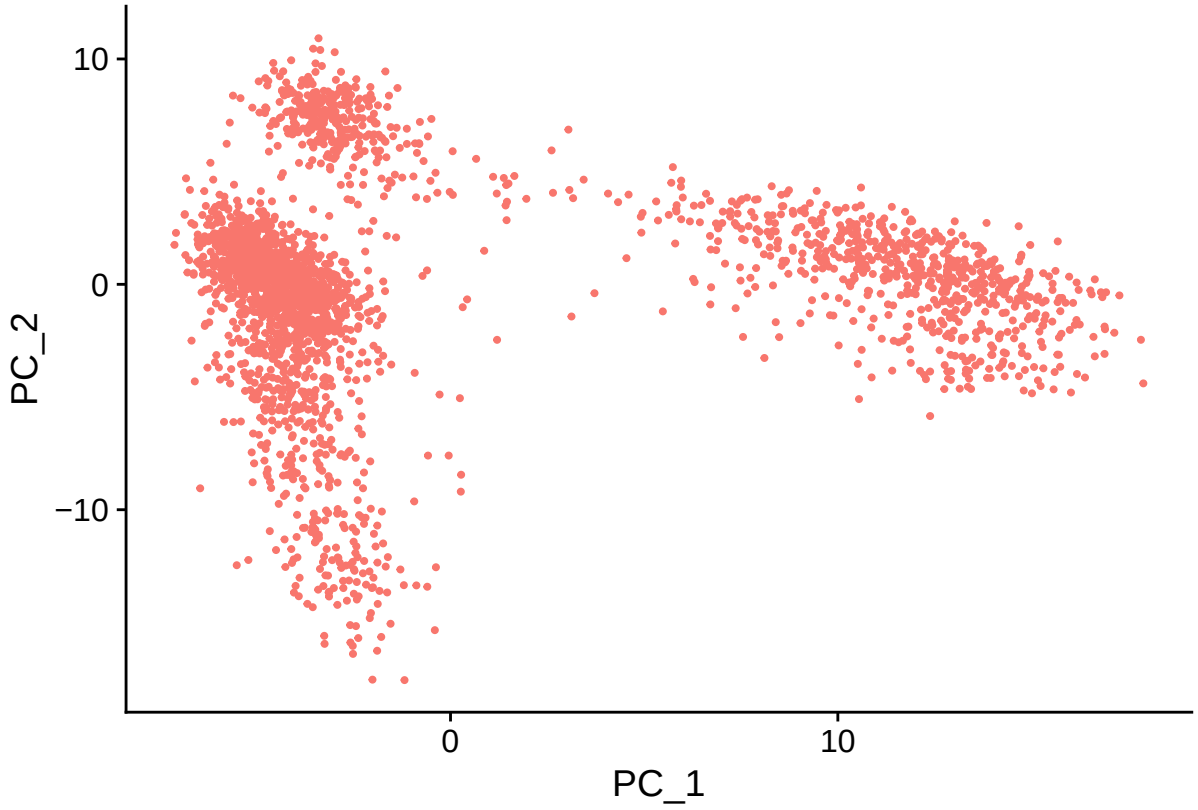
Conclusie: Aangezien er geen sterke correlatie is tussen percent.mt en nCount_RNA, concluderen we dat het niet nodig is om deze variabele te corrigeren door middel van regressie in de ScaleData() stap.

Lineaire dimensiereductie met PCA

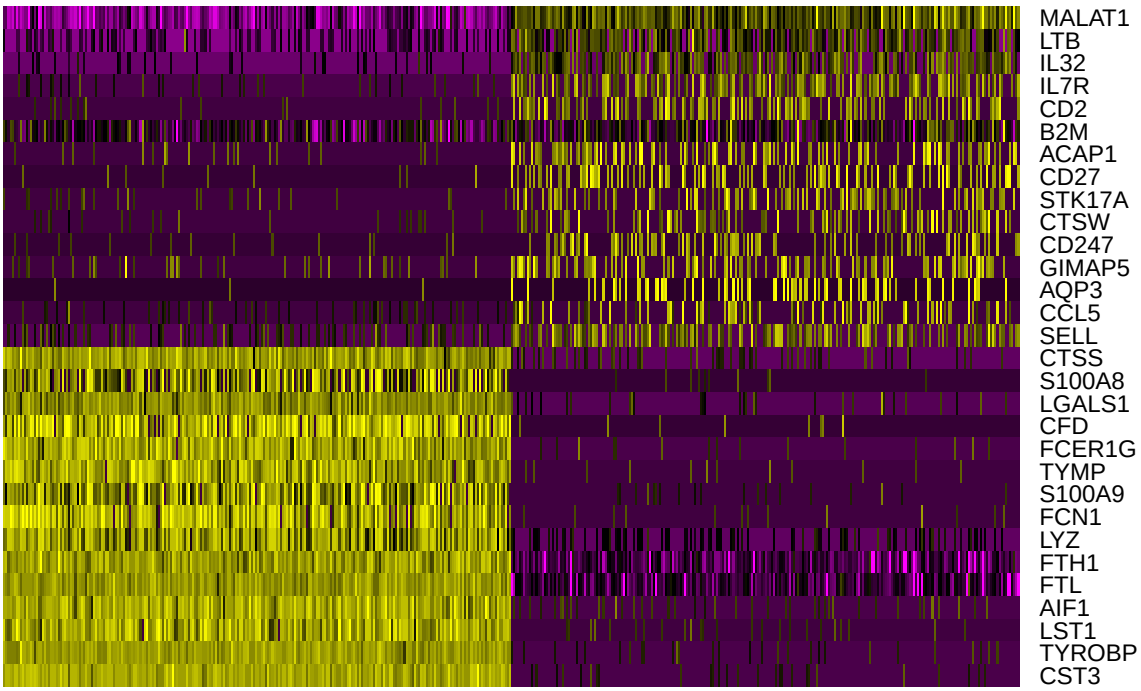
Na het schalen van de data voeren we Principal Component Analysis (PCA) uit. PCA reduceert de complexiteit van de dataset door een aantal samenhangende variabelen (principal compontents) te maken die het grootste deel van de variantie in de data verklaren. Dit helpt bij het identificeren van patronen en groepen in cellen.

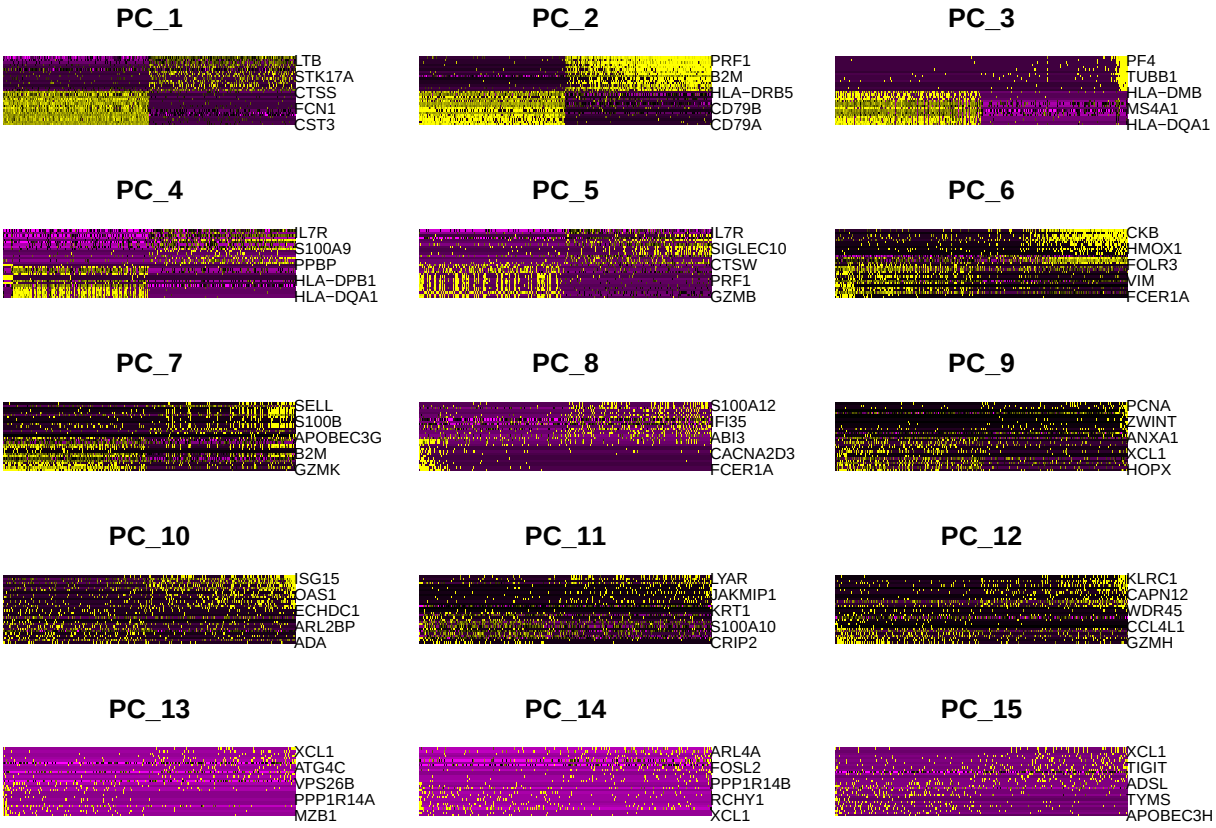
```
## PC_ 1
## Positive:  CST3, TYROBP, LST1, AIF1, FTL
## Negative:  MALAT1, LTB, IL32, IL7R, CD2
## PC_ 2
## Positive:  CD79A, MS4A1, TCL1A, HLA-DQA1, HLA-DQB1
## Negative:  NKG7, PRF1, CST7, GZMB, GZMA
## PC_ 3
## Positive:  HLA-DQA1, CD79A, CD79B, HLA-DQB1, HLA-DPB1
## Negative:  PPBP, PF4, SDPR, SPARC, GNG11
## PC_ 4
## Positive:  HLA-DQA1, CD79B, CD79A, MS4A1, HLA-DQB1
## Negative:  VIM, IL7R, S100A6, IL32, S100A8
## PC_ 5
## Positive:  GZMB, NKG7, S100A8, FGFBP2, GNLY
## Negative:  LTB, IL7R, CKB, VIM, MS4A7
```




10
0
−10
PC_2
0
10
PC_1

PC_1


MALAT1
LTB
IL32
IL7R
CD2
B2M
ACAP1
CD27
STK17A
CTSW
CD247
GIMAP5
AQP3
CCL5
SELL
CTSS
S100A8
LGALS1
CFD
FCER1G
TYMP
S100A9
FCN1
LYZ
FTH1
FTL
AIF1
LST1
TYROBP
CST3



Conclusie en Discussie

Het doel van deze oefening was om kennis op te doen over de Seurat workflow voor scRNA-sequencing data-analyse. Door gebruik te maken van de standaardwaarden en een turotial dataset, is het gelukt om een goed overzicht te krijgen van de verschillende stappen binnen Seurat.

Tijdens het proces heb ik gebruik gemaakt van de `LogNoralize` methode voor normalisatie. Ondanks dat ben ik geinteresseerd geraakt in de alternatieve `SCTranform` methode, die recentelijk steeds meer wordt toegepast vanwege de verbeterde normalisatie en variance stabilisatie. Hier wil ik mij verder in verdiepen zodat ik deze eventueel kan toepassen wanneer ik de VASA-sew data ga analyseren met Seurat.

Verder heb ik stilgestaan bij de verschillende parameters, hoe deze de uitkomsten van de analysestappen kunnen beinvloeden. Ook heeft deze kennismaking mij inzicht gegeven over de standaardwaarden die Seurat gebruikt. Het is belangrijk om hier goed bij stil te staan en na te gaan of deze standaarden wel realistisch zijn voor de dataset die je gebruikt. Zo is het bijvoorbeeld verstandig om de standaardgrens van percent.mt bij spiercellen verhogen. Spiercellen hebben een hoge energiebehoefte en bevatten daarom veel meer mitochondrien dan de meeste andere celtypen. Wanneer je deze grens niet aanpast, dan zul je hoogstwaarschijnlijk onbedoeld goede, gezonde spiercellen wegfilteren. Dit heeft directe gevolgen voor je analyse.

Een belangrijk onderdeel van deze workflow was het werken met de verschillende soorten grafieken die Seurat genereert, zoals VlnPlot, FeaturePlot en Heatmaps. Deze visualisaties waren in het begin nieuw voor mij, en ik merkte dat het aflezen ervan niet altijd vanzelfsprekend is. Daarom heb ik tijdens de analyse ook actief opgezocht hoe deze grafieken geinterpreteerd moeten worden. Dit hielp mij om beter te begrijpen wat de grafieken precies laten zien. Het leren lezen van deze plots is heel belangrijk om op een juiste manier te kunnen interpreteren en er betekeniscolle conclusies uit te trekken.

Wat ik vooral wil meenemen uit deze oefening - en wat ik ook als tip van mijn tutor kreeg - is het belang

van het spelen met parameters. het experimenteren met verschillende instellingen, zoals filterdrempels of normalisatiemethoden, helpt om betere keuzes te maken die passen bij de aard van de data en het biologische systeem.

Samengevat heeft deze oefening mij niet alleen technische vaardigheden met Seurat verbeterd, maar ook mijn inzicht verdiept in hoe belangrijk het is om kritisch om te gaan met standaard workflows.