BRNA eindopdracht

Winny ’t Hoen
30 augustus 2020
Studentnummer: s1101573

INHOUDELIJKE BESPREKING

Transcriptomen

Het transcriptoom bestaat uit alle RNA-moleculen die te maken hebben met het tot expressie brengen van de genen in een bepaalde cel. Bij het bestuderen van transcriptomen wordt gekeken naar welke genen deze expressie bepalen en onder welke omstandigheden. Het besturen van transcriptomen wordt transcriptomics genoemd. Het DNA (genoom) in een cel wordt omgezet naar RNA (transcriptoom), waarna RNA wordt omgezet naar eiwitten (proteoom). De eiwitten zorgen voor een fenotype. Echter, wordt maar een klein deel van al het aanwezige RNA omgezet naar eiwitten. RNA kan worden opgedeeld in twee groepen: het coderende en het niet-coderende RNA. Alleen het coderende RNA, messenger-RNA (mRNA), is functioneel en zorgt voor translatie naar eiwitten. mRNA bevat één tot vier procent van het totale RNA in een cel. Het overige RNA is niet coderend en kan worden verdeeld in:

Transfer RNA (tRNA) (15%)
Ribosomaal RNA (rRNA) (80-85%)
Small nuclear (snRNA)
Small nucleolar RNA (snoRNA)
MicroRNA
Small interfering RNA (siRNA)

De eerste twee genoemde niet-coderende RNAs, tRNA en rRNA, spelen beide een belangrijke rol in de translatie van mRNA naar eiwit. rRNA is onderdeel van het ribosoom. Ribosomen en tRNA spelen een grote rol in de eiwitsynthese. Het ribosoom zet een aminozuur op elke codon van het mRNA. Elke aminozuur wordt door tRNA naar de juiste plek getransporteerd. Er zijn 20 verschillende soorten aminozuren, met voor elke aminozuur een andere tRNA variant.
snRNA is een small nuclear RNA wat zich bevindt in de nucleus. snRNA zorgt ervoor dat RNA splicing kan plaatsvinden. snoRNA is een small nucleolar RNA die zich bevind in de nucleolus van de celkern. In de nucleolus worden ribosomen in elkaar gezet. Tijdens het in elkaar zetten van de ribosomen zijn snoRNA functioneel.
MicroRNA is een krachtig klein stukje RNA wat een stukje mRNA kan ‘uitzetten’, waardoor een stuk gen niet tot expressie kan komen. Small interfering RNA (siRNA) lijkt erg op microRNA en kan de expressie van genen beïnvloeden. Het verschil tussen microRNA en siRNA is dat siRNA een stukje mRNA na transcriptie kan afbreken, waardoor translatie wordt voorkomen.
Een belangrijk aspect van transcriptomics, is het bekijken welke mRNA isoform te maken heeft met de expressie van genomen. Om dit te meten moet RNA worden omgezet naar coderend DNA (cDNA). Hiervoor moet het transcriptoom geïsoleerd worden, wat bestaat uit alle verschillende stukken RNA. Het isoleren van mRNA wordt gedaan door een poly-A staart selectie te maken wat gebeurd door het toevoegen van probes die zich binden aan de poly-A staart. In combinatie met RT, DNA Pol, een buffer en dNTPs kan mRNA worden omgezet in cDNA door middel van Quantitative Reverse Transcription Polymerase Chain Reaction (qRT-PCR). Vervolgens kan de gen expressie gemeten worden.

Procedure micro-array

Een manier om mRNA levels te identificeren in een biologische sample is Microarray. Een Microarray is een glasplaat bestaande uit een grote hoeveelheid spots. In deze spots bevinden zich allemaal kleine stukjes enkelstrengs DNA, dit wordt een probe genoemd. Deze probes worden specifiek ontwikkeld voor elk onderzoek. Om probes te ontwikkelen is kennis over het sample vereist. Hiervoor worden enkelstrengs DNA-stukjes gebruikt van verschillende genen waarvan bekend is dat die in de sample aanwezig zijn.

Wanneer het RNA wat uit een cel is gehaald op een microarray plaat wordt gezet, kan het gaan binden met complementaire stukjes enkelstrengs DNA. RNA is van zichzelf erg instabiel, daarom wordt het omgezet naar cDNA voordat het op de microarray plaat wordt gezet. Om goed te kunnen terugzien waar het cDNA bindt, wordt aan het RNA een fluorescerende kleur gehangen per sample. Dit wordt gedaan door het amplificeren van cDNA naar aRNA. Bij het maken van een scan van de microarray, zijn alle stipjes zichtbaar. Zo kan precies worden teruggevonden welke stukken DNA aanwezig zijn in het sample. Aan de hand van de intensiteit van de kleur, kan worden gemeten welk gen verhoogd tot expressie komt. Doordat meerdere kleuren aan een kunnen probe binden kan de intensiteit per gen worden bepaald. Deze intensiteit kan afgelezen worden van de scan. Echter, kan veel DNA gaan plakken aan de achtergrond, waardoor de intensiteit hoog is maar dit niet terecht is. Hier moet voor worden gecompenseerd, wat gebeurd door de achtergrond intensiteit te meten.

Verwerken micro-arraydata

Aan het einde van een Microarray uitvoering worden de resultaten omgezet in getallen. Deze getallen geven de intensiteit aan. Deze getallen worden niet zo maar aangenomen, maar dit wordt eerst gecontroleerd. Het uiteindelijk het doel is om te weten of de expressie van een gen is toe- of afgenomen. Omdat deze getallen ver uit elkaar kunnen liggen, doordat de expressie van een gen bijvoorbeeld veel is toegenomen door de aanwezigheid van een ziekte. Dit wordt recht getrokken door het uitvoeren van een LOG-transformatie. Omdat niet ieder gen evenveel tot expressie komt, wordt de LOG gebruikt om aan te geven welke genen uit het sample allemaal interessant zijn. Anders kan een gen, waarvan het verschil niet groot is, maar de foldchange (veelvoudige verschil) wel, worden vergeten. De schaal wordt aangepast op de data, maar de data zelf blijft onveranderd. Wanneer dit wordt weergegeven in een grafiek, zie je dat de data niet breed uiteen loopt maar in een rechtere lijn blijft. Hierdoor zijn ook de echte fouten beter zichtbaar.

Hiernaast wordt een visuele inspectie op de data uitgevoerd, dit wordt gedaan door de verschillende genen en hun intensiteit tegen elkaar op te zetten. Verder worden de ratio's (M: log2 ratio voor de intensiteit van de genen) opgezet tegen de gemiddelde intensiteit (A) voor een bepaald gen. Dit wordt gedaan in een MA-plot. Dit geeft inzicht of in een bepaald sample de intensiteit afwijkt van het gemiddelde. Hierbij worden de samples met elkaar vergeleken. Van deze uitslagen kan een tabel gemaakt worden waar goed te zien is of de gen expressie is toegenomen (up gereguleerd), afgenomen (down gereguleerd) of constant is. Bij het bestuderen van deze visuele plotten en tabellen, worden twee aannames aangehouden. De eerste: er zijn evenveel up- als down gereguleerde genen. De tweede: de meeste transcripten (>80%) zijn niet gereguleerd. Dit betekend dat de M waarde altijd rond de nul ligt.

Omdat technische effecten ook invloed hebben op het resultaat, wordt hiervoor ook gecorrigeerd. Dit wordt normaliseren genoemd. Hier zijn verschillende manieren voor. Drie van deze manieren zijn gebaseerd op de fundamentele aannames, en twee zijn niet gebaseerd op de fundamentele aannames.

De eerste manier is het corrigeren van de totale intensiteit. Hierbij wordt gekeken naar hoeveel intensiteit per gen wordt opgepikt. Een gelijk getal wordt dan verwacht (aanname één). Mocht dit niet zo zijn, dan worden de resultaten vermenigvuldigd met de normalisatie factor. Die factor wordt berekend door het delen tussen de resultaten. Deze manier wordt alleen gebruikt als de distributie van twee samples hetzelfde is maar het gemiddelde verschillend is. Dit is de gemakkelijkste normalisatie methode, maar niet de beste. Hierbij wordt niet gekeken naar de bias van de kleurintensiteit.

Mocht deze eerste methode niet geschikt zijn voor jouw samples, dan kan de LOWESS (locally weighted linear regression) normalisatie methode worden toegepast. Hier wordt gekeken naar elke groep aan intensiteit, die even groot moeten zijn (aanname één), en deze worden met een andere normalisatie factor genormaliseerd. Deze groepjes worden bepaald door windows die op de A as worden bepaald. Deze windows bepalen de lokale correcties.

De laatste normalisatie methode, gebaseerd op de aannames, is de quantile normalisatie methode. Hier wordt de data per sample verdeeld. Er wordt gekeken naar laag, hoog en gemiddeld tot expressie komende genen per sample. Dit is stap één en wordt RANKS genoemd. Vervolgens wordt dit gesorteerd waarna voor elke RANK het gemiddelde wordt berekend, dit gemiddelde is de normalisatie factor.

Naast deze drie normalisatie methodes zijn nog twee methodes beschikbaar die niet gebaseerd zijn op de aannames. Een van deze methodes is doormiddel van housekeeping genes. De aanname waar deze methode op gebaseerd is is dat de expressie van de housekeeping genes hetzelfde blijft. Het verschil tussen deze housekeeping genes en de resultaten mag worden rechtgetrokken, omdat wordt verwacht dat deze gelijk blijven.

De andere methode is de spike-in controls. Hier wordt aan elk sample een vaste hoeveelheid gen toegevoegd, die nog niet in het sample aanwezig zijn. Als er bij de resultaten een andere waarde aan zit, mag dat verschil worden gebruikt als normalisatie factor. Dan wordt er vanuit gegaan dat die waarde door een technische fout is gemaakt en dit op alle genen invloed heeft gehad.

Na normalisatie wordt nogmaals een kwaliteitscontrole uitgevoerd. Nu wordt verwacht dat de verschillen tussen de samples zichtbaar zijn, en de opvallende afwijkingen eruit gehaald kunnen worden. Hiervoor wordt vaak hiërarchisch clustering gebruikt. Hier worden de samples met hetzelfde expressie patroon samen gegroepeerd over alle genen. Hierin kunnen vaak al patronen worden gezien. Om de verschillen nog beter te kunnen bestuderen wordt een heatmap gebruikt, hier zijn de gen expressie patronen goed te zien.

Procedure RNA-seq

Een andere manier om mRNA levels te identificeren van een biologische sample is RNA sequencing (RNA-seq). Deze manier wordt vaak gebruikt wanneer er geen voorkennis is van de transcripten. Als eerste wordt het RNA geïsoleerd en wordt cDNA gemaakt. De sequencer gaat vervolgens alles voor je sequencen, door middel van short sequence reads. Hier komen allerlei fragmenten uit. Deze fragmenten

worden gecontroleerd door een Quality control. De slechte stukken worden eruit gefilterd, of er wordt bijvoorbeeld een stuk afgeknipt. Om deze fragmenten weer een geheel te maken, wordt gebruik gemaakt van mapping. Door het gebruik van mapping is een volgorde bekend, waardoor de fragmenten sneller op de juiste plek worden gezet. Zo kan bijvoorbeeld het genoom of transcriptoom worden gebruikt. Er zit een verschil tussen mappen tegen het genoom en het transcriptoom. Bij het isoleren van mRNA uit RNA wordt alternative splicing gebruikt. Dit zorgt voor een andere vorm van mRNA. Alle exonen worden meegenomen, en alle intronen niet. Bij het mappen tegen het transcriptoom gebeurd dit direct op het mRNA. Echter, als je het genoom gebruikt om tegen te mappen wordt dit wat moeilijker. In het genoom zitten namelijk alle intronen, die niet meer in mRNA stukjes aanwezig zijn. Dat betekend dat een stuk aanwezig is tussen elke exon, dit heet een exon-exon junction. Het genoom wordt voornamelijk als referentie gebruikt wanneer er interesse is in de isoformen van het mRNA.

Een andere optie is om de fragmenten te mappen tegen zichzelf, wanneer bijvoorbeeld geen genoom of transcriptoom bekend is. Dit wordt de Novo Assembly genoemd.

Na het mappen wordt er gesummarized. Er wordt gekeken naar hoeveel reads op elke sample gemaakt zijn. Dit wordt gedaan voor alle genen in het genoom. Alles wordt opgeslagen in een count tabel, wat kan tellen per gen, isoform of exon. Dit is het startpunt voor de rest van de analyse.

Normaliseren RNA-seq

Na het mappen en summarizen wordt pas genormaliseerd. Tijdens het normaliseren wordt er gekeken naar de transcript lengte, Library grootte en sequentie specificiteit. Er kan op twee manieren worden genormaliseerd, binnen of tussen de samples. Wanneer wordt gekeken binnen de sample wordt Reads per kilobase Miljoen (RPKM) gebruikt, en tussen de samples wordt Trimmed mean of m-waardes (TMM) gebruikt als normalisatie methode.

RPKM maakt gebruik van een soort formule, die is als volgt; Number of mappen reads *10^9/ length gene (bp)*total Number of mappen reads. De formule wordt toegepast op elke gen in elk sample. Met deze formule wordt de RNA lengte en totale reads per sample genormaliseerd. Door het normaliseren kunnen betere vergelijkingen gemaakt worden. Deze methode is erg handig voor vergelijkingen binnen een sample, maar niet altijd toepasbaar voor vergelijkingen tussen samples. Bij het gebruik van de RPKM methodes voor vergelijkingen tussen samples kan het lijken alsof genen minder tot expressie komen door het verschil aan hoeveel genen.

Daarom wordt er TMM (Trimmed mean of m-waardes) gebruikt. TMM berekend een schalingsfactor voor de library grootte. Om dit te doen worden drie stappen uitgevoerd. Als eerst wordt voor elk gen de fold change bepaald tussen de twee samples. Vervolgens worden alle genen met een hogere fold change dan 30%, en alle genen met een zeer hoge expressie verwijderd. De genen die overblijven zullen niet veel verschillen. Aan de hand van de overgebleven genen wordt de scholingsfactor bepaald. Als de TMM normalisatie factor veranderd is, dus niet meer op de nul ligt, betekend dat de compositie niet meer goed is. Daarom is TMM normalisatie nodig, hiermee wordt genormaliseerd met de schalingsfactor. Daarna kunnen alle uitschieters nog met een andere methode worden verwijderd.

Experimenteel design

Experimenteel design is onderdeel van de eerste drie stappen voor RNA-seq. Voor experimenteel design wordt bijvoorbeeld gekeken naar het RNA, het protocol en de hoeveel samples. Maar ook hoe het onderzoek uitvoeren gaat worden, onder welke omstandigheden. Ook het platform, hoe wilt je het gaan sequencen? Door het maken van een goed experimenteel design worden biases (fouten) voorkomen.

DGE-testing

DGE staat voor differentieel gen expressie. Om dit te testen worden statische toetsen gebruikt. Wanneer gebruik gemaakt is van Microarray (single of dual), wordt er gebruik gemaakt van een t-toets. Als er gebruik gemaakt is van RNA-seq wordt er een Chi-Square toets gebruikt.

De resultaten van Microarray en RNA-seq kunnen niet op dezelfde manier worden vergeleken. Bij Microarray is heel duidelijk te zien of een bepaald gen aanwezig is of niet. Bij het uitvoeren van normalisering en een LOG transformatie kunnen standaard statische toetsen worden gebruikt voor het meten van differentiële gen expressie. Dit is echter niet het geval bij RNA-seq. Bij RNA-seq worden korte reads random gegenereerd, dit betekend dat je niet precies alle aanwezige transcripten gaat sequencen. Hierdoor kan een gen niet wordt gevonden, wat niet betekend dat deze niet aanwezig is in het sample. Om een statische toets uit te voeren op de resultaten wordt een Chi-Square test gebruikt. De Chi-Square test wordt gebruikt met een 2x2 tabel welke vooraf wordt gemaakt. Uit die tabel wordt een p-waarde gehaald. Bij genen die laag tot expressie komen is het differentieel verschil moeilijker te berekenen omdat de p-waarde enorm kan wisselen bij meerdere malen uitvoeren van het experiment. Dit is niet betrouwbaar, daarom worden de transcripten met een lage hoeveelheid counts verwijderd. Dit wordt gebaseerd op counts per million (CPM). Door het bekijken van je Library wordt een grens getrokken, en alles onder die grens wordt verwijderd.

Na het verwijderen van de lage counts, kan worden gekeken naar het verschil in gen expressie. Bij elke keer als het experiment uitvoert wordt krijg je verschil in resultaat. Dit is technische variatie waarmee rekening gehouden moet worden bij de statistische toets. Naast de technische variatie is vaak meer variatie aanwezig dan wordt verwacht. Dit kan bijvoorbeeld door biologische variatie komen die aanwezig is in het sample. Voor deze variatie moet worden gecorrigeerd in de statistische toets. Biologische variatie heeft grote invloed op het overschatten van gen expressie, dit wordt gecorrigeerd.

Multiple testing

Uit de statische toets toegepast op de resultaten komt een p-waarde. Deze p-waarde wordt echter niet gebruikt, deze p-waarde moet eerst worden gecorrigeerd. Dit is vanwege de vals positieve waarde, bij een transcriptoom experiment worden veel statistische toetsen uitgevoerd. Bij elke statistische toets wordt de kans op vals positieve resultaten groter. Er zijn twee methodes om de p-waarde te corrigeren, de Bonferroni methode en de Benjamini-Hochberg methode.

Bonferroni corrigeer de p-waarde door het vermenigvuldigen met de hoeveelheid testen die worden uitgevoerd. Deze berekening levert een strenge nieuwe p-waarde. Een andere optie is om de Benjamini-Hochberg methode te gebruiken, deze wordt ook wel false discovery rate (FDR) genoemd. FDR rekent van te

voren het percentage vals positieve resultaten, dit percentage wordt afgehaald van je significante genen. Deze methode is minder streng en wordt meer gebruikt.

GSEA & pathwaylevel-analyse

Het uitvoeren van multiple testing resulteert in een lijst met differentieel tot expressie komende genen, met daarbij een p-waarde en een foldchange (FC) per gen. Vervolgens kan deze lijst worden geanalyseerd om de biologische betekenis te achterhalen. Hierbij wordt gekeken naar annotatie en pathway van de genen. Dit wordt een functionele analyse genoemd van je experimentele data. Benodigde informatie voor een functionele Analyse is genen ontologie, informatie over de biologische processen/pathways en Gene Set Enrichment analyse (GSEA) om de veranderingen te achterhalen.

Genen ontologie houd in dat alle standaard eigenschappen van de gene die voorkomen worden weergegeven, zoals moleculaire functie/biologisch proces en cellulaire componenten. Naast deze informatie wordt er ook gekeken naar de interacties tussen de genen. Op deze eigenschappen wordt geclassificeerd in groepen. Dit geeft een hiërarchische structuur. Deze clustering kan helpen bij het zien van verschillen bij groepen genen. Hier wordt dus geen rekening gehouden met de gevonden p-waarde of FC.

Een pathway is een netwerk bestaande uit allerlei moleculaire componenten. In de pathway zit allerlei informatie over interactie tussen de moleculaire componenten. Informatie over de pathway zelf is te vinden in allerlei databases. Om de informatie van een database aan de uitkomsten van het experiment te koppelen, kunnen verschillende methodes worden gebruikt. Een van deze methodes is Gene Set Enrichment analyse (GSEA). GSEA kijkt voornamelijk naar of de verdeling van een functionele groep met genen in het geheel significant verschillend is dan wordt verwacht. Daar kan uit opgemaakt worden of er een bepaalde pathway uit of aan staat. Om dit te analyseren wordt er gerangschikt op de p-waarde. De analyse wordt uitgevoerd met behulp van een running sum, die per gen bepaald of die in de pathway hoort. De running sum wordt hoog bij veel voorkomende genen die dicht bij elkaar liggen, en lager als er een gen voorkomt die verder van de andere genen liggen. De score die hier uit. Komt wordt de verrijkingsscore (ES) genoemd, en dit wordt voor elk pathway berekend. Vervolgens wordt de significantie niveau geschat aan de hand van hoeveel pathways er wordt getest. Het resultaat kan wordt weergegeven in een Enrichment plot, waar goed te zien is welke genen in welke pathway verhoogd tot expressie komt in elke sample.

RAPPORTAGE ANALYSE

Filteren op lage expressie

Genen met lage resultaten leveren weinig bewijs voor differentieel expressie, en kunnen de resultaten slecht beïnvloeden. Vandaar dat deze genen eruit worden gefilterd. We filteren met een count-per-million (CPM) van boven de 0.5 in minstens twee samples. Eerst zetten we de countdata om in CPM, door de functie cpm(). Vervolgens zetten we de thresh vast.

```
>    myCPM <- cpm(countdata)
```

```
>    thresh <- myCPM > 0.5
```

Vervolgens komen de resultaten in FALSE en TRUE voormaat te staan. Dit is makkelijk te filteren door te tellen waar minstens twee TRUE's voorkomen. De countdata file wordt aangepast om alle data te houden die goed genoeg zijn om verder mee te onderzoeken.

```
>    keep <- rowSums(thresh) >= 2
```

```
>    counts.keep <- countdata[keep,]
```

```
> head(countdata)
                SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516 SRR1039517 SRR1039520 SRR1039521
ENSG00000000003        679        448        873        408       1138       1047        770        572
ENSG00000000005          0          0          0          0          0          0          0          0
ENSG00000000419        467        515        621        365        587        799        417        508
ENSG00000000457        260        211        263        164        245        331        233        229
ENSG00000000460         60         55         40         35         78         63         76         60
ENSG00000000938          0          0          2          0          1          0          0          0
> head(counts.keep)
                SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516 SRR1039517 SRR1039520 SRR1039521
ENSG00000000003        679        448        873        408       1138       1047        770        572
ENSG00000000419        467        515        621        365        587        799        417        508
ENSG00000000457        260        211        263        164        245        331        233        229
ENSG00000000460         60         55         40         35         78         63         76         60
ENSG00000000971       3251       3679       6177       4252       6721      11027       5176       7995
ENSG00000001036       1433       1062       1733        881       1424       1439       1359       1109
>
```

Tabel 1. In het bovenste tabel is de gene counts te zien voor het verwijderen van de lage counts. Daaronder zijn de counts per gen te zien na het filteren.

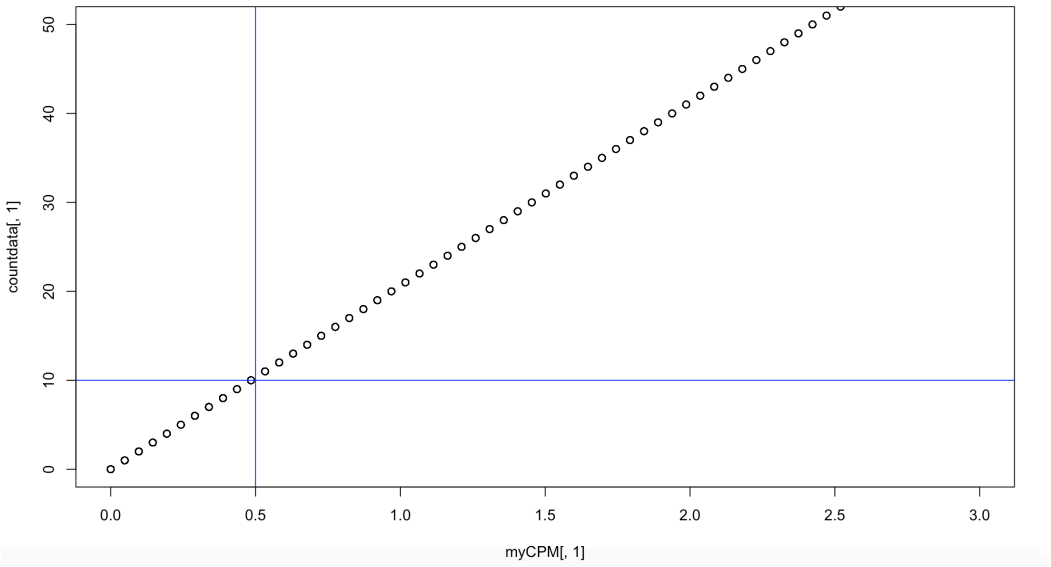
Het resultaat van filteren is te zien in tabel 1. De lage waardes te zien in head(countdata) bovenin tabel 1, zijn in head(counts.keep) niet meer te zien. Deze waardes liggen hoger.

Bij een CPM van 0.5 ligt de grens bij de teller van 10 á 15. Deze grens kunnen we goed bekijken bij het plotten van de data en de myCPM waardes.

```
>    plot(myCPM[,1],countdata[,1],ylim=c(0,50),xlim=c(0,3))
```

```
>    abline(v = 0.5, col = "blue")
```

```
>    abline(h = 10, col = “blue")
```



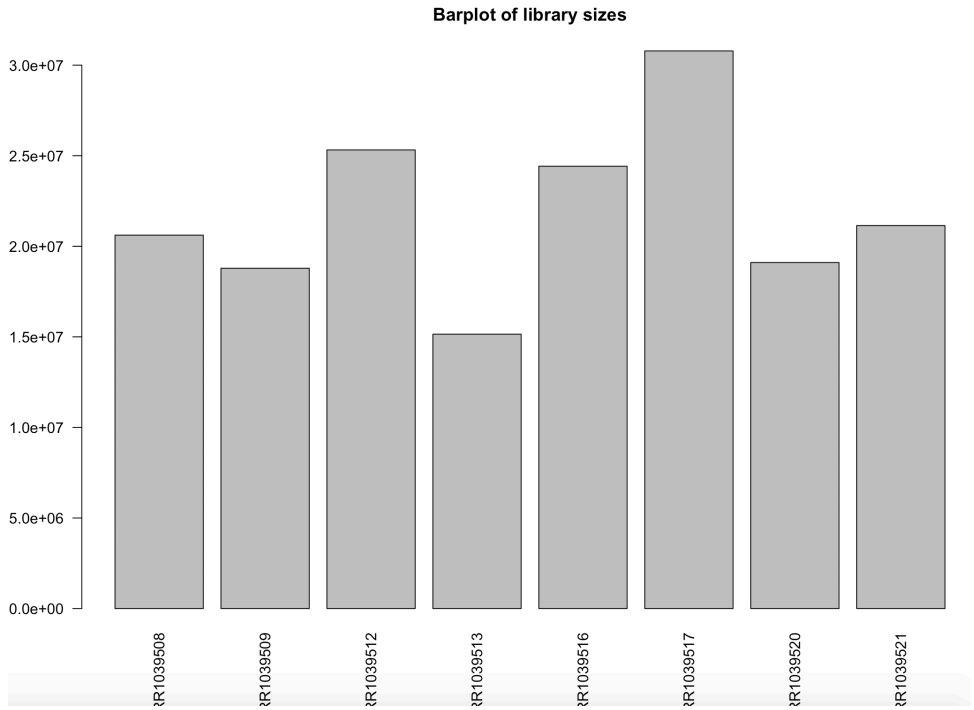
Figuur 1. In dit figuur is de eerste sample te zien, met op de x-as CPM en op de y-as counts van sample 1. De punten die zich onder de blauwe lijnen bevindt, worden eruit gefilterd.

Dit zijn de resultaten van de eerste sample. De lijnen geven weer welke data wordt meegenomen in counts.keep. De verticale lijn geeft de threshold weer, alle puntjes links van de blauwe lijn hebben een te lage threshold en zijn dus niet genoeg aanwezig. De horizontale lijn geeft de telling weer, puntjes met een telling lager dan 10 worden niet verder meegenomen in het onderzoek. In dit geval is goed te zien dat het vierkantje in de linkerhoek, waar de twee lijnen elkaar snijden, niet wordt meegenomen in verder onderzoek.

Quality Control

Vervolgens wordt er een DGEList gemaakt. DGE staat voor differentieel gen expressie. Het voordeel aan een DGEList is dat het meer informatie kan opslaan onder andere kopjes. Bij het aanmaken van een DGEList wordt het kopje counts aangemaakt, met de data counts.keep en een kopje samples, met informatie over de samples zoals library size. Met deze DGEList kunnen er plots gemaakt worden, of de differentieel gen expressie te bekijken. Eerst wordt er gekeken naar de library size van elke sample, om te kijken of hier al een groot verschil te zien is.

```
> barplot(y$samples$lib.size, names=colnames(y), las=2)
> title("Barplot of library sizes")
```



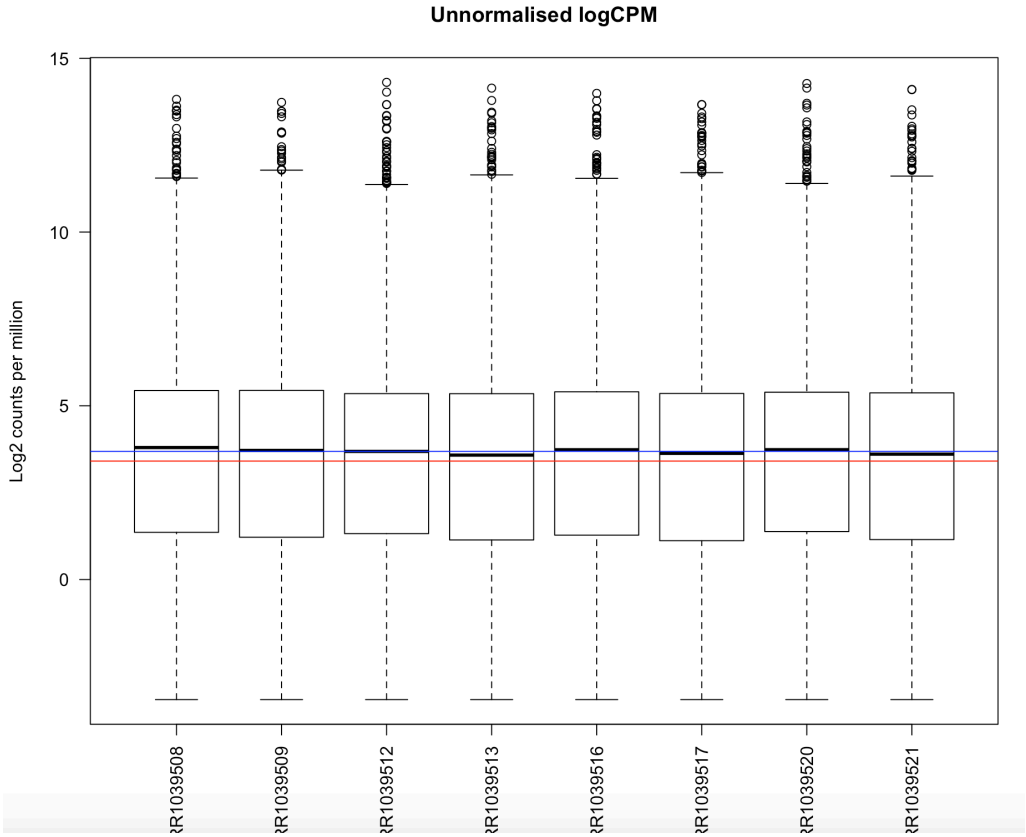
Figuur 2. In deze barplot is de Library size van alle samples weergegeven.

Library size geeft de totale count weer van de sample. Je kan een groot verschil zien in library size tussen sample SRR1039513 en sample SRR1039517, waarbij sample SRR1039517 bijna twee keer zo groot is als sample SRR1039513. Vervolgens wordt er gekeken naar de verdeling van de samples door het maken van een boxplot. Omdat de data niet normaal verdeeld is, wordt log gebruikt om de data weer te geven.

```
> logcounts <- cpm(y, log = TRUE)
> boxplot(logcounts, xlab="", ylab="Log2 counts per million", las=2)
> abline(h=median(logcounts), col=“blue”)
> abline(h=mean(logcounts), col="red", pch=18)
```

> title("Unnormalised logCPM")

De blauwe lijn is het mediaan van alle samples. De rode lijn is de mean, dus het gemiddelde van alle counts. Ideaal zou worden gezien als alle individuele medianen van de samples overeenkomen met de mediaan van alle counts.



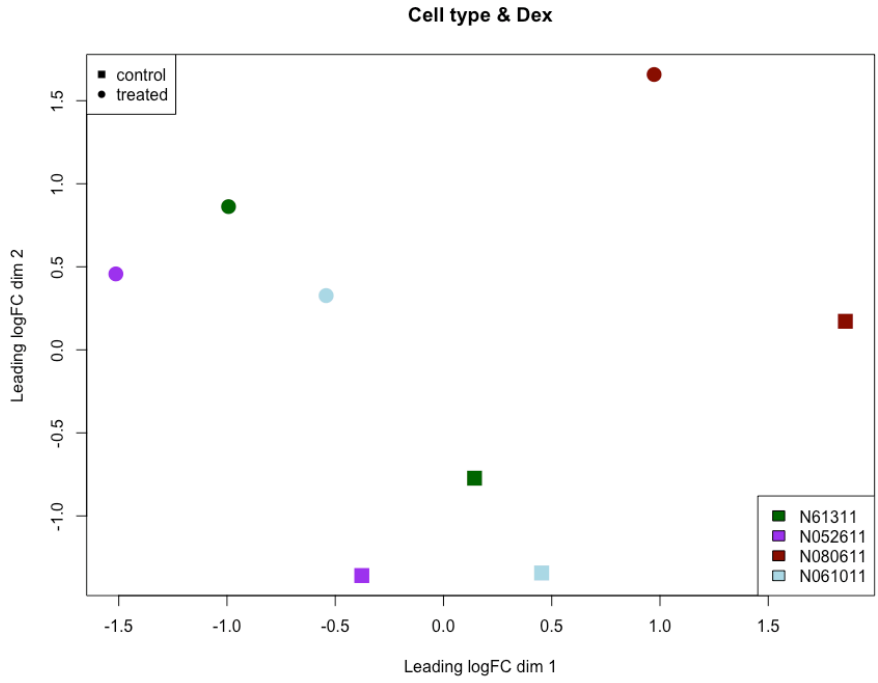
Figuur 3. In deze boxplot zijn de CPM met een LOG transformatie van elk sample weergegeven. De rode lijn geeft het gemiddelde weer van de counts, de blauwe lijn geeft de mediaan weer. De zwarte lijn geeft de mediaan weer van elke sample.

In figuur 3 is goed te zien dat alle samples goed overeenkomen tot de mediaan, het verschil per sample is niet groot. De mean ligt onder de mediaan, dit geeft aan dat de data een grotere hoeveelheid lage scores bevat. Dit is ook te zien aan de mediaan die iets dichterbij de bovenkant van de boxen liggen, dan aan de onderkant. Echter is het verschil niet erg groot, en komen de medianen van de samples goed overeen met de mediaan van alle counts dus wordt er geen verandering gemaakt in de CPM.

Ongepaarde vs gepaarde analyse

Het maken van een MDS plot is van groot belang bij de kwaliteitscontrole. In een MDSplot worden belangrijke componenten visueel weergegeven. De variatie tussen de samples, behandeld of onbehandeld, worden weergegeven in de MDS plot.

```
> col.cell <- c("dark green", "purple", "dark red", "light blue")[sampleinfo$celltype]
> shps <- c(15,16)[sampleinfo$dex]
> plotMDS(y, dim=c(3,4), col=col.cell, pch = shps, cex = 2)
> legend("topright", fill = c("dark green", "purple", "dark red", "light blue"),
  legend = unique(sampleinfo$celltype))
> legend("topleft", legend = unique(sampleinfo$dex), pch = shps)
> title("Cell type & Dex")
```



Figuur 4. In dit figuur wordt de MDS plot van de control en behandelde samples.

In figuur 4 is de MDS plot weergegeven. De samples laten zien dat ze erg verschillend zijn, niet alleen de behandelde en onbehandelde celtypes liggen ver uit elkaar. Deze informatie kan verder worden meegenomen in het experiment.

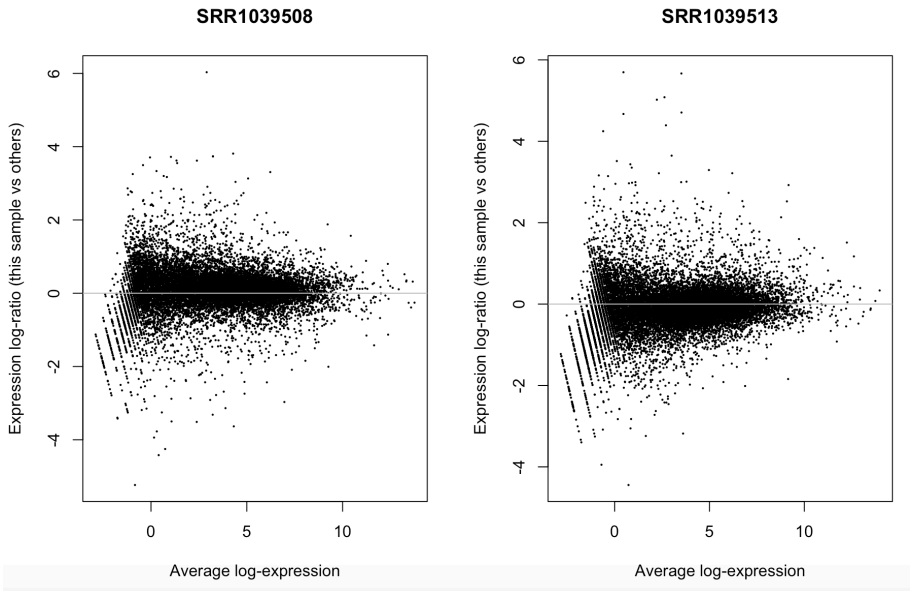
Normalisatie

De data wordt genormaliseerd. Dit wordt gedaan door middel van de TMM normalisatie methode. TMM wordt gebruikt omdat er wordt genormaliseerd tussen de samples. De data wordt genormaliseerd aan de hand van hun library size. De norm.factors waardes worden veranderd in de DGEList.

```
>       y <- calcNormFactors(y)
>       y$samples
+                group  lib.size        norm.factors
+       SRR1039508    1 20615750   1.0558949
+       SRR1039509    1 18789410   1.0203974
+       SRR1039512    1 25320612   0.9915741
+       SRR1039513    1 15149527   0.9489665
+       SRR1039516    1 24420766   1.0313658
+       SRR1039517    1 30786651   0.9767417
+       SRR1039520    1 19104845   1.0281090
+       SRR1039521    1 21144737   0.9523625
```

Zoals te zien liggen de nieuwe norm.factors allemaal rond het getal 1 (default). Als een norm.factor onder de een is, betekend dat dat er te veel van die sample is. Dit is goed te zien in het volgende figuur 5, waar we een sample nemen met de hoogste waarde; SRR1039508, en een sample met de laagste waarde; SRR1039513, naast elkaar in een plot weergeven. Deze plots worden zonder normalisatie waardes zodat het verschil goed te zien is.

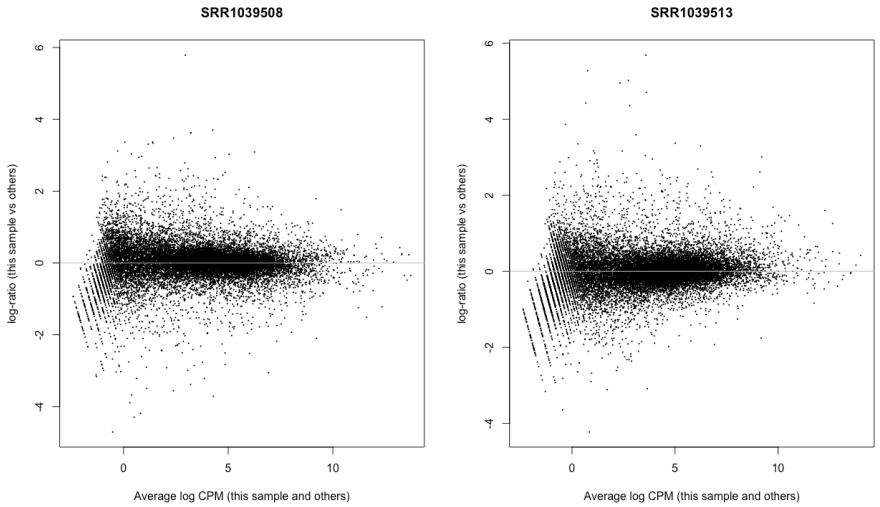
```
>       par(mfrow=c(1,2))
>       plotMD(logcounts, column = 1)
>       abline(h=0, col="grey")
>       plotMD(logcounts, column = 4)
>       abline(h=0, col="grey")
```



Figuur 5. In dit figuur worden niet genormaliseerd gen expressie resultaten

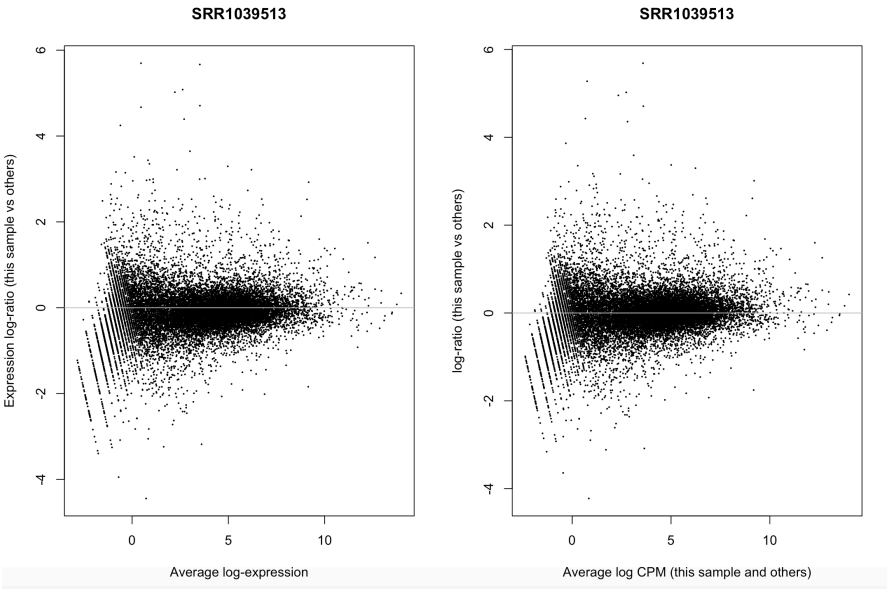
Je kan goed zien dat er bij SRR1039513 veel datapunten zich onder de 0-lijn bevinden, en dat dat bij SRR1039508 juist tegenovergesteld is. Vervolgend worden dezelfde plots gemaakt, maar dit keer met de normalisatie waardes die zich nu in de DGEList bevinden.

```
>       par(mfrow=c(1,2))
>       plotMD(y, column = 1)
>       abline(h=0, col="grey")
>       plotMD(y, column = 4)
>       abline(h=0, col="grey")
```



Figuur 6. In dit figuur worden genormaliseerd gen expressie resultaten weergegeven

Bij deze plots is duidelijk verschil te zien. De 0-lijn lijkt zich precies middenin de datapunten te bevinden, er is geen duidelijk overschot aan datapunten meer aan een van de kanten van de lijn. Om het effect duidelijker in beeld te brengen plotten we SRR1039513 voor en na het normaliseren naast elkaar.

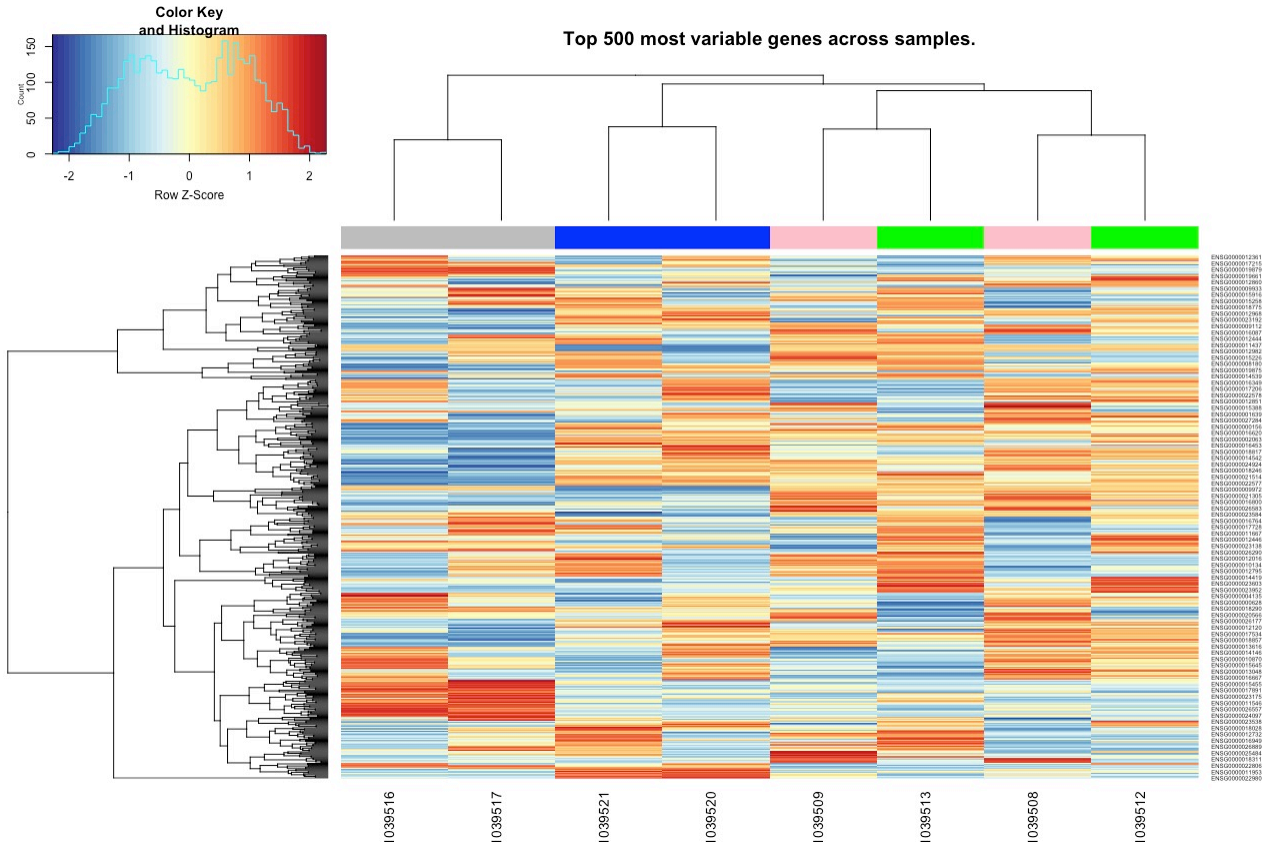


Figuur 7. In dit figuur wordt links de niet genormiseerde gen expressie resultaten weergegeven, en recht de genormaliseerde gen expressie resultaten van sample

Clustering

Clustering vindt plaats voor het visueel weergeven van gen expressie. Clustering wordt visueel weergegeven door een heatmap. De kleurintensiteit wordt bepaald door de Z-score, legenda is zichbaar links bovenin de heatmap. Z-score laat zien of een gen, omhoog/omlaag of gelijk tot expressie komt.

```
> var_genes <- apply(logcounts, 1, var)
> select_var <- names(sort(var_genes, decreasing = TRUE))[1:500]
> highly_variable_lcpm <- logcounts[select_var,]
> mypallet <- brewer.pal(11, "RdYlBu")
> morecols <- colorRampPalette(mypallet)
> col.cell <- c("green", "blue", "grey", "pink")[sampleinfo$celltype]
> heatmap.2(highly_variable_lcpm, col = rev(morecols(50)), trace = "none", main = "Top 500 most variable genes across samples.", ColSideColors = col.cell, scale = "row")
```



Figuur 8. Heatmap van clustering samples.

DGE-testing

Nu de data is genormaliseerd, kan er worden getest op differentieel expressie. Voordat de differentieel gen expressie wordt berekend, worden er statistische toetsen uitgeoefend op de data. Omdat een RNA sequentie analyse wordt uitgevoerd, kunnen er fouten worden gemaakt of een gen aanwezig is of niet. Hiervoor wordt de data gegroepeerd, en deze groepen worden in een design matrix gezet. De groepen worden nogmaals bekeken.

```
>       group
+       [1] N61311.control  N61311.treated  N052611.control N052611.treated N080611.control
+       [6] N080611.treated N061011.control N061011.treated
+       8 Levels: N052611.control N052611.treated N061011.control ... N61311.treated
```

De design matrix wordt gemaakt, en de kolom namen worden veranderd om het overzichtelijker te maken.

```
>       design <- model.matrix(~sampleinfo$celltype + sampleinfo$dex)
>       colnames(design) <- c("Intercept", "N061011", "N080611", "N61311", "dextreated")
>       design
```

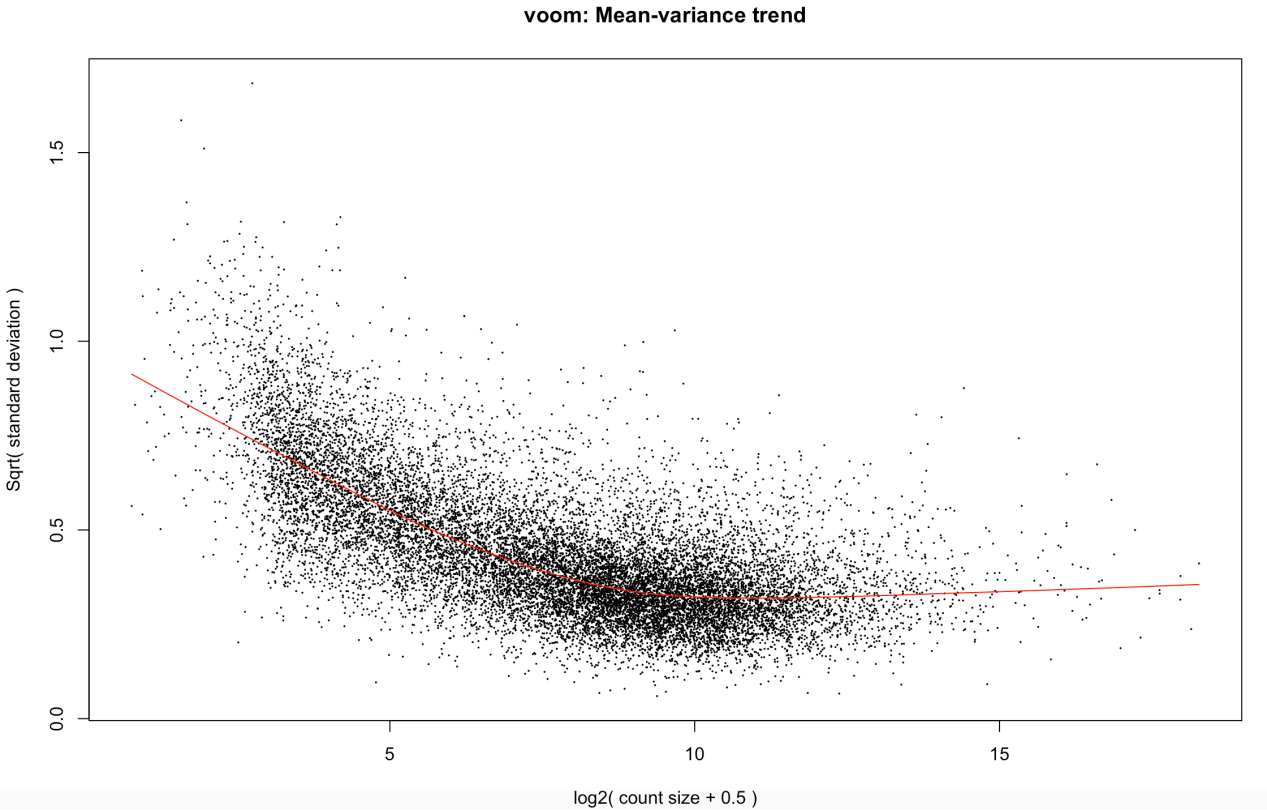
```
  Intercept N061011 N080611 N61311 dextreated
1         1       0       0      1          0
2         1       0       0      1          1
3         1       0       0      0          0
4         1       0       0      0          1
5         1       0       1      0          0
6         1       0       1      0          1
7         1       1       0      0          0
8         1       1       0      0          1
attr(,"assign")
[1] 0 1 1 1 2
attr(,"contrasts")
attr(,"contrasts")$`sampleinfo$celltype`
[1] "contr.treatment"

attr(,"contrasts")$`sampleinfo$dex`
[1] "contr.treatment"
```

Figuur 8. Hier wordt de design matrix weergegeven.

De design matrix is in figuur 9 weergegeven. Voor de rijen onder elke sample betekend ‘0’ niet behandeld, en ‘1’ betekend behandelde sample. Vervolgens kan een voom transformatie worden gedaan. Hierbij worden de counts getransformeerd naar log2 CPM, die zijn gebaseerd op de nieuwe norm factoren na het normaliseren van de counts. Op de x-as wordt de log2 counts aangegeven, en op de y-as de standaarddeviatie.

```
> v <- voom(y, design, plot = TRUE)
```

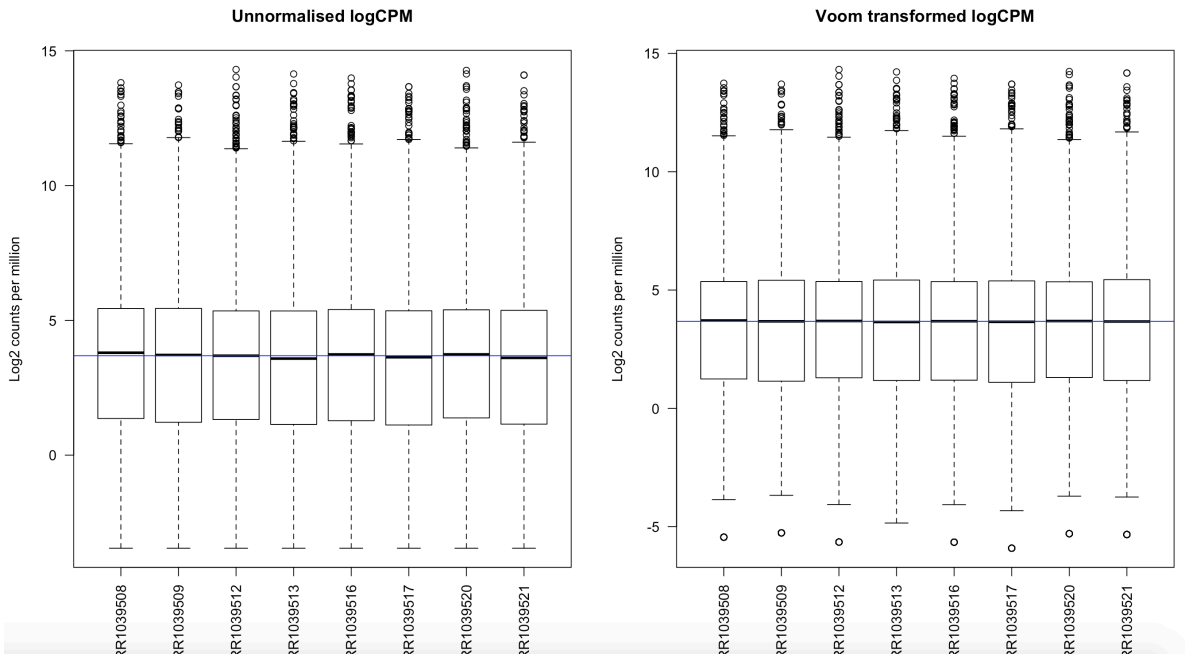
De voom plot wordt op de volgende pagina weergegeven, in figuur 9. Mocht de rode erg afwijken van de vorm die de punten aannemen, was er wellicht meer filtering nodig.



Figuur 9. Voom plot van de design matrix.

Nu de data genormaliseerd is en weer is omgezet in CPM, kunnen we het verschil gaan bekijken in een boxplot. De eerste bosplot is eerder al gemaakt met de nog niet genormaliseerde data, en de tweede is genormaliseerd. De boxplot is weergegeven in figuur 10 op de volgende pagina.

```
> par(mfrow=c(1,2))
> boxplot(logcounts, xlab = "", ylab="Log2 counts per million", las=2, main="Unnormalised logCPM")
> abline(h=median(logcounts), col="blue")
> boxplot(v$E, xlab="", ylab="Log2 counts per million", las=2, main="Voom transformed logCPM")
> abline(h=median(v$E), col="blue")
```



Figuur 10. Links wordt de niet genormaliseerde CPM met LOG transformatie weergegeven van alle samples. Rechts wordt de genormaliseerde CPM met LOG transformatie weergegeven.

Als er gedetailleerd naar de blauwe mediaan lijn wordt gekeken, liggen deze dichterbij alle mediaan lijnen van de sampels. In de bosplot van de voom getransformeerde data zijn ook uitschieters te zien onder het minimum. De data kan nu worden gebruikt voor het testen op differentiele expressie door limma te gebruiken. Limma is handig om snel en makkelijk een samenvatting te maken van de contrasten tussen de genen. Eerst wordt er een linear model van gemaakt.

```
>	fit <-lmFit(v)
>	cont.matrix <- makeContrasts(N08 = N080611, N6 = N61311, N06 = N061011, levels = design)
>	fit2 <- contrasts.fit(fit, cont.matrix)
>	fit2 <- eBayes(fit2)
>	summa.fit <- decideTests(fit2)
>	vennDiagram(summa.fit)
>	summary(summa.fit)
```

De summary verteld hoeveel van de genen gen expressie omhoog (Up), omlaag (Down) of constant (NotSIg) is gebleven.

	N08	N6	N06
Down	1138	898	281
NotSig	14821	15117	16139
Up	771	715	310

Met de topTable command kunnen de top 10 genen, gesorteerd op p-waarde, worden weergegeven per sample. Te zien in figuren 12 tot en met 14.

	logFC	AveExpr	t	P.Value	adj.P.Val	B
ENSG00000123243	-4.644894	8.514869	-43.85260	8.587626e-12	1.436710e-07	17.00615
ENSG00000262902	-6.564767	1.302968	-29.77699	2.722671e-10	2.118932e-06	11.76338
ENSG00000018625	-3.534526	4.235066	-28.01358	4.687416e-10	2.118932e-06	13.56407
ENSG00000007933	-3.263642	4.755868	-27.18492	6.121775e-10	2.118932e-06	13.43868
ENSG00000131831	-3.645075	4.842939	-27.08149	6.332733e-10	2.118932e-06	13.15303
ENSG00000118849	-4.128919	6.386230	-24.62052	1.475379e-09	4.012920e-06	12.69231
ENSG00000198759	-3.875078	5.710288	-24.26401	1.679046e-09	4.012920e-06	12.55414
ENSG00000165092	-4.104112	4.050267	-23.57529	2.167008e-09	4.531756e-06	11.80550
ENSG00000164308	3.123221	4.795209	23.21103	2.487367e-09	4.623738e-06	11.86460
ENSG00000106565	-5.486747	3.864918	-22.89482	2.808435e-09	4.698512e-06	10.84770

Figuur 11. Top tien genen van sample N080611.

	logFC	AveExpr	t	P.Value	adj.P.Val	B
ENSG00000204941	-4.611251	4.988710	-30.59384	2.139509e-10	3.579399e-06	14.148673
ENSG00000262902	-6.717357	1.302968	-26.25966	8.327141e-10	5.728598e-06	10.263024
ENSG00000164308	3.453735	4.795209	25.64602	1.027244e-09	5.728598e-06	12.452048
ENSG00000018625	-2.811030	4.235066	-23.35026	2.359099e-09	9.866930e-06	12.156865
ENSG00000145681	-3.742407	5.058709	-19.11760	1.377823e-08	4.610197e-05	10.443842
ENSG00000243137	4.050767	0.324571	18.38382	1.943019e-08	5.417785e-05	9.860330
ENSG00000227081	1.746604	6.290538	17.62094	2.817259e-08	6.265336e-05	9.834711
ENSG00000114948	-3.157775	4.207529	-17.43979	3.083994e-08	6.265336e-05	9.669271
ENSG00000232111	-9.062655	-3.186011	-16.94429	3.968112e-08	6.265336e-05	5.086545
ENSG00000180914	-3.773474	3.723185	-16.86739	4.128992e-08	6.265336e-05	9.372821

Figuur 12. Top tien genen van sample N61311.

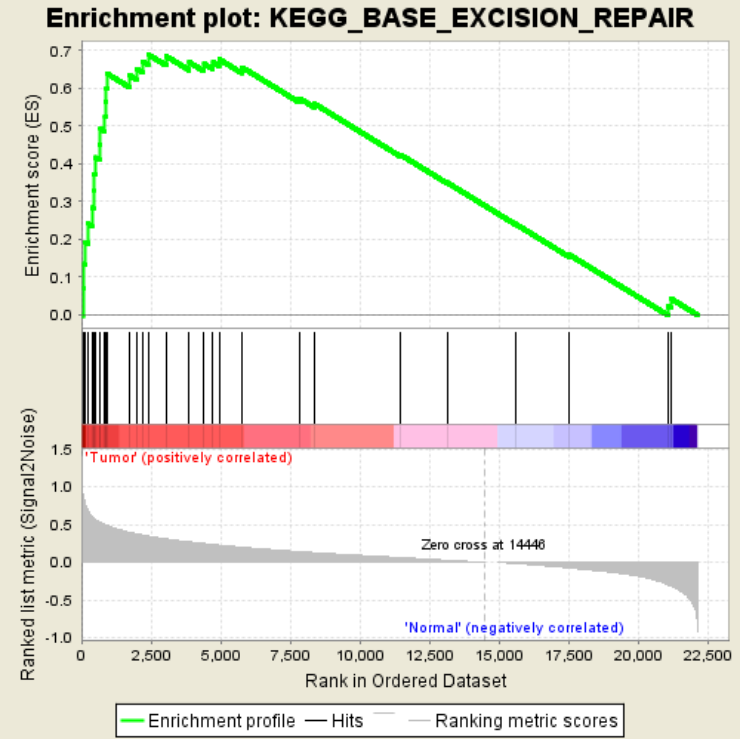
	logFC	AveExpr	t	P.Value	adj.P.Val	B
ENSG00000204941	-5.236667	4.988710	-31.26791	1.761894e-10	2.947649e-06	14.530520
ENSG00000164308	3.771385	4.795209	28.06427	4.612640e-10	3.858474e-06	13.531367
ENSG00000262902	-6.778051	1.302968	-25.90362	9.400284e-10	5.242225e-06	11.208934
ENSG00000007933	-2.790490	4.755868	-23.30463	2.400308e-09	1.003929e-05	12.282131
ENSG00000018625	-2.170299	4.235066	-19.31094	1.261162e-08	4.219849e-05	10.626369
ENSG00000114374	-10.502928	3.353311	-17.94676	2.399485e-08	4.486374e-05	6.590602
ENSG00000197728	-1.623092	6.771462	-17.90950	2.443589e-08	4.486374e-05	9.912361
ENSG00000129824	-11.654353	4.676577	-17.79096	2.589993e-08	4.486374e-05	6.564003
ENSG00000131002	-10.275107	2.496363	-17.74744	2.646168e-08	4.486374e-05	6.549308
ENSG00000183878	-8.989680	1.488137	-17.57573	2.881346e-08	4.486374e-05	6.493287

Figuur 13. Top tien genen van sample N061011.

GSEA en Pathwaylevel-analyse

Aan de hand van de p-waarde en FC waarde uitkomstig van de multiple testing kan de functionele analyse beginnen. Er wordt gekeken naar alle genen in elk sample, hoevaak zij voorkomen in elke pathway. Hieruit kunnen allerlei biologische conclusies worden getrokken. Het resultaat kan worden weergegeven in een Enrichment plot, waar goed te zien is welke genen in welke pathway verhoogd tot expressie komt per sample. Een voorbeeld hiervan is weergegeven in figuur 15.

De groene lijn geeft de running sum getal weer, dit getal geeft aan of de gen tot de pathway behoord. Hoe hoger de lijn, hoe meer dit gen voorkomt in je pathway. Dit is niet alleen afhankelijk van het gen zelf, maar ook van de positie van het gen en hoeveel genen er naast elkaar liggen.



Figuur 15. Voorbeeld van een Enrichment plot.