# Technische documentatie

Het doel van dit onderzoek is, door middel van een eQTL analyse een verschil aan te tonen in genexpressie tussen kanker en niet-kanker patiënten.​ Om dit te kunnen bepalen is er gebruikt gemaakt van een aantal Python scripts en Bash commando’s. Deze scripts en commando’s zijn vervolgens aan elkaar vastgemaakt met behulp van een Snakemake. Een duidelijke weergave van de stappen die zijn gevolgd is te zien in figuur 1. Ook staan een aantal gegevens in de volgende GitHub account: <https://github.com/moshtachismail/final_blok10.git>, maar dit zijn niet alle bestanden. Doordat veel bestanden erg groot zijn en niet op de Git gezet konden worden, dus de andere bestanden staan op de server van de Han, Guacamole.

## GitHub

Ook al staan de meeste bestanden op de server staan sommige bestanden ook op de GitHub, onder andere dit bestand (technische documentatie), maar ook het volgende:

* Inputs\_ (Map):

Binnen deze mapje staat het bestand test\_deelchr10.txt en chr3\_10\_protein.txt. Dit is het bestand dat gebruikt is als testbestand in de SNP\_data.py. Dit bestand bestaat uit een set protein ID’s dat gebruikt is om de SNP’s op te halen die in chromosoom 10 gevonden kunnen worden. Er is voor gekozen om een set te pakken, omdat het proces om de SNP’s op te halen lang kan duren en om er zeker van te zijn dat het proces ook echt werkt is er nu voor gekozen om een set protein ID’s te pakken.

Het bestand chr3\_10\_protein.txt, bevat alle protein ids van chr10.

* Output\_ (Map)
  + Regression
    - Hierin staan output bestanden van de regressie analyse.
  + GATK
    - In het mapje gatk\_outputs bij de output staat het eindelijke bestand, wat gebruikt kan worden voor de SNP-selectie (Snps\_sel.py), het bestand heet final\_output\_genotypegvcfs.vcf.gz. GATK werd gebruikt om SNP’s en indels te identificeren.
  + STAR
    - In het mapje STAR staat een bestand genaamd normalized\_values.txt. In dit bestand staan de counts per gen deze waarden zijn genormaliseerd met behulp van de Python-script normalizing.py.
  + SNP
    - In het mapje SNP staat een bestand genaamd snps.tsv. Dit is de output van het Python script SNP\_data.py, dus hierin zitten alle snps die gevonden zijn voor chromosoom 10.
* Src
  + Normalizing.py
    - Dit Python script normaliseert de rauwe counts die worden gegenereerd door de STAR en zet de genormaliseerd waarden in een nieuw bestand die te vinden is in Output/STAR binnen deze Git.
  + SNP\_data.py
    - Hier worden alle SNPs opgehaald van chromosoom 10.
  + Regressie\_analyse\_final.py
    - Dit python-script maakt de plotjes die de waarbij het fenotype tegen het genotype wordt uitgezet.
  + Requirements.txt
    - Dit is het bestand met alle gebruikte packages. Deze is niet erg van belang als je alles gaat runnen op de server, omdat alles al geinstalleerd in de conda environment staat. Dus je kan Snakemake meteen aanroepen en runnen!
  + Snps\_sel.py
    - Hier worden de SNPs van chromosoom 10 (snps.tsv) vergeleken met de SNPs die gevonden zijn onder de patiënten (output gatk) .
  + Version.py
    - Er is een python script geschreven om de versie nummers uit de gennamen te halen die in het genormaliseerde bestand staan, zodat deze output gebruikt kon worden voor de regressieanalyse. Dit is ook het uiteindelijke bestand dat op de Git staat (normalized\_values.txt).
* Doc
  + Hier is de technische documentatie te vinden.
* Snakefile, losse bestand met de Snakefile.

## Server

Op de server staan alle bestanden die gebruikt zijn voor de snakemake

* Inputs
  + GATK\_inputs
    - A549\_0\_2Aligned.sortedByCoord.out.bam, dit is een van de bestanden die worden aangemaakt in de rule alignments. Hiermee start de workflow om de GATK uit te voeren.
    - Clinvar bestand die nodig is om de BaseRecalibrator stap binnen de GATK uit te voeren
  + SNPS\_input
    - Hier staat het bestand test\_deelchr10.txt. Ook te vinden op de Git.
    - En het bestand chr3\_10\_protein.txt, dit zijn alle protein id’s van chr10.
  + STAR\_input
    - Hier staan de bestanden van de referentiegenoom en de twee bestanden van de patienten. De patienten data zijn de bestanden beginnend metA549\_25
* Output
  + Regression2analysis
    - Binnen het mapje staan mogelijke plotjes van de regressie analyse
  + SNP\_output
    - Binnen het Mapje get\_snps, staat het bestand snps.tsv, ook te vinden op de git.
  + STAR\_output
    - Normalized\_values.txt en
      * Dit is het output bestand van STAR, maar hier zijn de versienummers van de gennamen weggehaald. Zodat deze gebruikt kon worden voor de regressieanalyse.
    - Normalised\_counts\_A549\_matrix.txt
      * Dit is het output bestand van star met alle genormaliseerde counts. (incl versienummers)
    - Indexes
      * Hier staan 16 bestanden die automatisch als output worden gegenereerd na het runnen van de rule indexes.
    - Alignments
      * Ook hier staan zes bestanden die automatisch als output worden gegenereerd zodra de rule alignments wordt gerund.

* + gatk\_outputs
    - Ook worden hier een aantal bestanden gemaakt door de rules uit te voeren die benodigd zijn voor de GATK. Het uiteindelijke output bestand is, final\_output\_genotypegvcfs.vcf.gz. Dit bestand wordt ook gebruikt als input om de SNP\_selectie uit te voeren
* Snakefile, dit is het Snakemake bestand. Wat ook op de Git staat.
* Src, hierin staan dezelfde bestanden als op de Git
  + - Ook staan er een aantal mapjes die zijn geïnstalleerd om tools uit te kunnen voeren, namelijk de mapjes: gatk, miniconda3, bcftools en R

Ook is het goed om te weten dat de Snakefile wordt gerund binnen een conda environment, genaamd gatk. Dit maakt het mogelijk om verschillende tools en programmeertalen te gebruiken binnen de Snakefile.

## Processen

### Zoeken naar de eiwitten van de genen uit chromosoom 10

In dbSNP (van NCBI) zijn alle pathogene SNP’s van de eiwitten van alle genen van chromosoom 10 gezocht en opgehaald. Hiermee worden alle SNP’s opgehaald die voorkomen in deze eiwitten, het script slaat alle pathogene SNP’s op in een lijst en schrijft die later weg naar een output TSV-bestand (snps.tsv). Het TSV-bestand bevat voor elke SNP het SNP ID, de clinical\_significance, de gen naam, de mutatie, de soort variatie en de positie.

### Mapping to reference (STAR)

Om te kunnen mappen naar het referentiegenoom is als input gebruik gemaakt van de data van een patiënt en het referentiegenoom. Met de input data is er een bestand gemaakt met STAR-index. Vervolgens zijn de reads gealigned en zijn de reads geteld. Als output kwam er een BAM-bestand met daarin de patiënt reads gemapt tegen het referentie genoom.

#### Normaliseren van counts

Nadat de reads geteld (counts) waren zijn deze waardes genormaliseerd met behulp van de Python-script normalizing.py. Dit Python script geeft een CSV-bestand met daarin de genormaliseerde counts.

### GATK

Voor GATK zijn er 6 commando’s toegepast. Het eerste commando (AddOrReplaceReadGroups) zorgt ervoor dat de reads in het opgegeven bestand toe worden gevoegd aan één nieuwe read-group. Het tweede commando (SplitNCigarReads) zorgt ervoor dat een read wordt gesplit op exonen en de intronen wordt weg gefilterd. De actie wordt gebaseerd op de CIGAR-string van de read. Het derde commando (BaseRecalibrator) wat is uitgevoerd, genereerd een recalibration tabel, hierin staan de read groups beschreven, de quality score, machine cycle and nucleotide context. Het vierde commando (ApplyBQSR) doet een herijking van de bases. Dit zijn de bases van de reads die in de recalibration table zijn meegeven door het 3e commando.

Nu komen de 2 commando's die zijn gebruikt voor de bepaling van de SNP's. Het eerste van deze commando is de HaplotypeCaller, deze zorgt ervoor dat de SNP’s en indels worden bepaald in de reads. Het allerlaatste commando (GenotypeGVCFs) zorgt ervoor dat het genotype bepaald voor HaplotypeCaller wordt weggeschreven in een VCF. De output van het eerste commando wordt gebruikt voor de twee commando en de output van commando twee wordt weer gebruikt voor commando 3 enzovoort.

### SNP-selectie

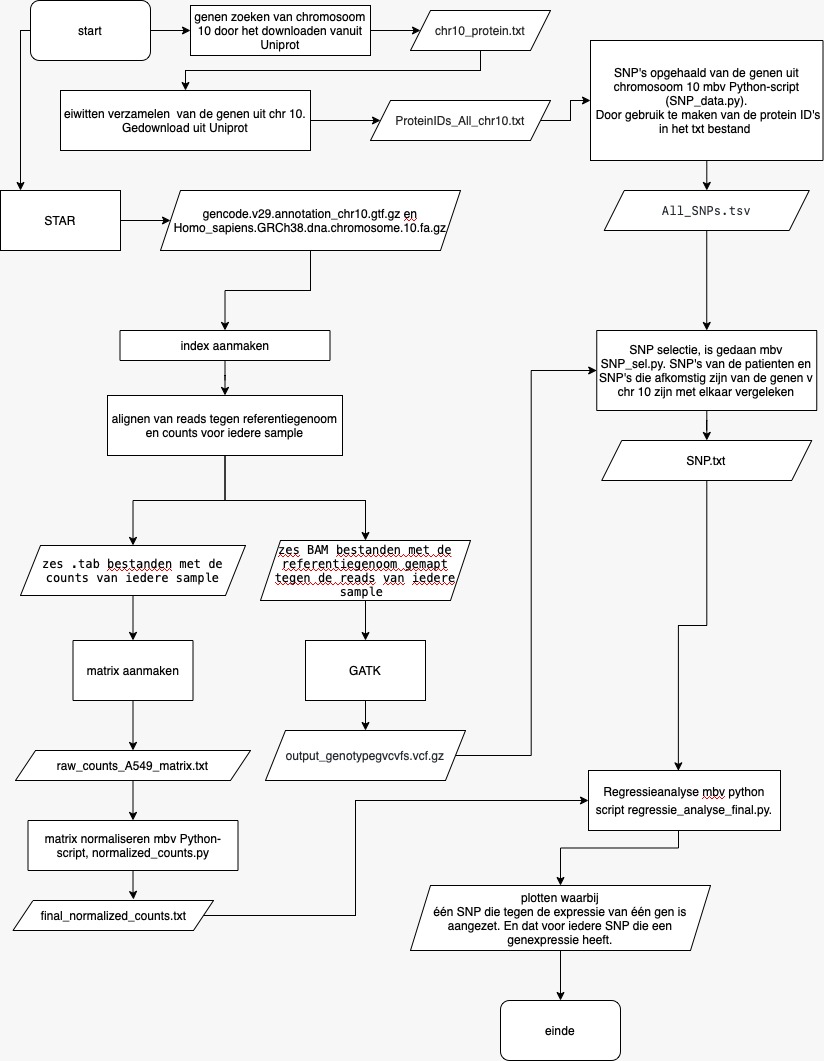
Nadat de SNP’s van de patiënten verkregen zijn en dat SNP’s die afkomstig zijn van de genen van chromosoom 10, konden deze vergeleken worden met elkaar. Door middel van het python script snps\_sel.py is dit uitgevoerd. Dit script leest een gezipt vcf bestand in die de SNP’s bevat, die aangetoond zijn bij de patiënten. Ook is er een tsv bestand ingelezen die de SNP’s bevat van chromosoom 10 Deze SNP’s worden ook wel de SNP’s van interesse genoemd. Vervolgens zijn alle SNP-locaties van de patiënten die ook voorkomen in de lijst van SNP’s van interesse opgeslagen in een nieuw bestand. In dit bestand is voor iedere SNP-locatie het genotype voor iedere patiënt opgeslagen.

### EQTL-analyse

De regressieanalyse die is uitgevoerd is een eQTL analyse. Dit is uitgevoerd door middel van het python script regressie\_analyse\_final.py. Er zijn hier 2 input bestanden voor gebruikt, namelijk de genormaliseerde counts en het bestand waar de genotypes in staan. Als output komt er van één SNP die tegen de expressie van één gen is aangezet. En dat voor iedere SNP die een genexpressie heeft.

## Vervolgonderzoek

Het doel van dit onderzoek, was om door middel van eQTL analyse een verschil aan te tonen in genexpressie tussen kanker en niet-kanker patienten. Voor een vervolgonderzoek zou er gekeken kunnen worden naar alle chromosomen van de mens ipv alleen chromosoom 10. Als dit gedaan wordt zou er ook rekening gehouden worden met de hoeveelheid data de server kan. Ook is het beter om meerdere patiënten te bekijken in plaats van een. Hierdoor krijg je meerdere outputs en kan je een betere conclusie maken.



Figuur : Flowchart van de Snakemake. De eerste vier stappen staan niet in de Snakemake, omdat deze stappen zijn uitgevoerd om uiteindelijk een goed input bestand te maken benodigd voor de Snakemake. Het staat in deze flowchart voor de volledigheid, aangezien de input bestand niet is verkregen en we deze zelf hebben gegenereerd.