1 Computer Practicum: Exome sequencing

1.1 General

Biolab website

All documents required for these exercises can be found on the following web-site:
https://bioinformatics.amc.nl/education/gs-bioinformatics-sequence-analysis/

Linux and R

In these exercises we are going to use the operating system *Linux* and the statistical program *R*. I assume that you don't have (much) experience with Linux and R although the introductory lectures about Linux and R (in your first year) should have provided you with a sufficient basis. Nevertheless, all commands required for the exercises will be provided to you. If you want to know more about Linux or R you can make the following tutorials:

- Linux: http://wiki.bioinformaticslaboratory.nl/foswiki/bin/view/BioLab/EducationIntroductionUnix
- R: http://wiki.bioinformaticslaboratory.nl/foswiki/bin/view/BioLab/ComputinginR

Important note!!! In the exercises below we will use Linux and R commands. A common mistake is to give R commands from the Unix prompt or, vice versa, Unix commands from the R prompt. This will result in an error. If you follow the instructions precisely, then this won't be a problem. If you obtain an error then check if you have provided your command in the right environment (R or Unix)

Using SurfSara HPC Cloud

Since we will be handling a large dataset and because the alignment step requires sufficient compute power, we will make use of the SurfSara Cloud (SurfSara http://www.surfsara.nl/). The Cloud provides the computer hardware to do these exercises, and is accessible via your own computer. All data and software has already been installed in the Cloud. Follow the separate instructions to login and get started.

1.2 Introduction to the exercises

1.2.1 Nicolaides Baraitser Syndrome

Nicolaides–Baraitser syndrome (NBS, OMIM 601358) is a rare disorder that was first described in 1993 [2], but was only recently well delineated [3]. The main characteristics are sparse hair, typical facial morphology, short stature, microcephaly, brachydactyly, interphalangeal joint swellings, epilepsy and intellectual disability with marked language impairment. This syndrome occurs in individuals with various ethnic backgrounds without significant differences between sexes. No familial cases are known with the exception of one pair of concordant monozygotic twins. This suggests that NBS is caused by dominant de novo mutations.

To identify the gene(s) and mutation(s) that cause Nicolaides-Baraitser, exome sequencing [4] was initially performed for four patients. After inclusion of additional patients the disease causing gene was identified [1]

In these computer exercises you will use data from the initial cohort of 4 patients. You will analyse this data and try to identify the gene responsible for Nicolaides Baraitser.

References

[1] Van Houdt JK, Nowakowska BA, Sousa SB, van Schaik BD, Seuntjens E, Avonce N, Sifrim A, Abdul-Rahman OA, van den Boogaard MJ, Bottani A, Castori M, Cormier-Daire V, Deardorff MA, Filges I, Fryer A, Fryns JP, Gana S, Garavelli L, Gillessen-Kaesbach G, Hall BD, Horn D, Huylebroeck D, Klapecki J, Krajewska-Walasek M, Kuechler A, Lines MA, Maas S, Macdermot KD, McKee S, Magee A, de Man SA, Moreau Y, Morice-Picard F, Obersztyn E, Pilch J, Rosser E, Shannon N, Stolte-Dijkstra I, Van Dijck P, Vilain C, Vogels A, Wakeling E, Wieczorek D, Wilson L, Zuffardi O, van Kampen AH, Devriendt K, Hennekam R, Vermeesch JR. (2012) Heterozygous missense mutations in SMARCA2 cause Nicolaides-Baraitser syndrome. Nature Genetics, 44(4), 445-9.

[2] Nicolaides, P. & Baraitser, M. An unusual syndrome with mental retardation and sparse hair. Clin Dysmorphol. 1993, 2, 232-6.

[3] Sousa, S.B. et al. Nicolaides-Baraitser syndrome: Delineation of the phenotype. Am J Med Genet A 2009, 149A, 1628-40.

[4] Ng SB, Buckingham KJ, Lee C, Bigham AW, Tabor HK, Dent KM, Huff CD, Shannon PT, Jabs EW, Nickerson DA, Shendure J, Bamshad MJ. Exome sequencing identifies the cause of a mendelian disorder. Nat Genet. 2010, 42(1), 30-5

2 Getting prepared

2.1 Read the instructions carefully

Read the instructions carefully

2.2 Login to the Cloud

Follow the instructions in the separate document on Canvas to login to the Cloud. If you follow these instructions you will have opened a **Unix Terminal** and have given the command **cd Exome** to enter the exome directory. You should a Unix prompt similar to *student28@surfsara-Desktop:~/Exome$*

2.3 Using Linux commands

Common mistake that students make with the exercises below:

- Execute specific command (e.g., BWA) while not being in the right directory (= Linux name for folder or map). Consequently, the program cannot find the data or you get other error messages. Thus take care of all the Unix '**cd**' commands in this tutorial!

- A Unix command (e.g., **ls**) may take one or more arguments (e.g., **-al**). Make sure to put a space between the Unix command and all of its arguments (e.g., **ls –al**)

- If you get an error message then in 99.99% of the cases you made a typo in the command (e.g., you have forgotten a space, or put a space too many; you typed a small cap letter where a capital letter was expected.

- Input files are missing or empty (0 bytes). In this case one or more of your previous commands failed. You can check for empty files by using the Unix command ls –al. You will see the files with 0 bytes. This may give you a clue what might went wrong.

2.4 The dataset: the fastq files

The full dataset is too large for these computer exercises. Therefore, you will use part of the sequence data, which comprises a random selection of sequences corresponding to chromosome 9 from 1 of the 4 patients included in the study. These sequences will be used to demonstrate how variants are detected in sequence data. For the final selection of the disease causing gene/variants you are provided with the full list of mutations of all 4 patients.

You can find this NicBar.fastq in the ./Exome/fastq directory in the cloud environment to which you will login. You should see the fastq directory if you type **ls** from the Unix prompt.

3 Exome sequencing exercises

Question 1

a. What is a fastq file?
b. What is the format of a fastq file?

3.1 Step 1. Aligning the sequence reads against a reference Human genome.

The first step of the data analysis is to align the sequences (reads) against the Human reference sequence (build hg19), which we downloaded for you from the UCSC database. The reference sequence for chromosome 9 is provided to you in the Exome/reference directory (chr9.fa). The reference sequence is the full length DNA sequence of this chromosome. You will do the alignment with the program *BWA*. Align the fastq files against the reference human genome sequence according to the instructions below.

Question 2

a. What does the set of fastq sequences represent?
b. What is a reference genome?
c. What is alignment?
d. Why do you need to align the fastq sequences against the reference sequence?
e. What is the purpose of BWA?

- Go to the fastq directory: from the prompt ("$") type
    - **`cd fastq`**
- With the linux command '**pwd**' you can show the current directory path. With the command '**ls**' or '**ls –al**' you can show the content of the current directory. See Figure 1.

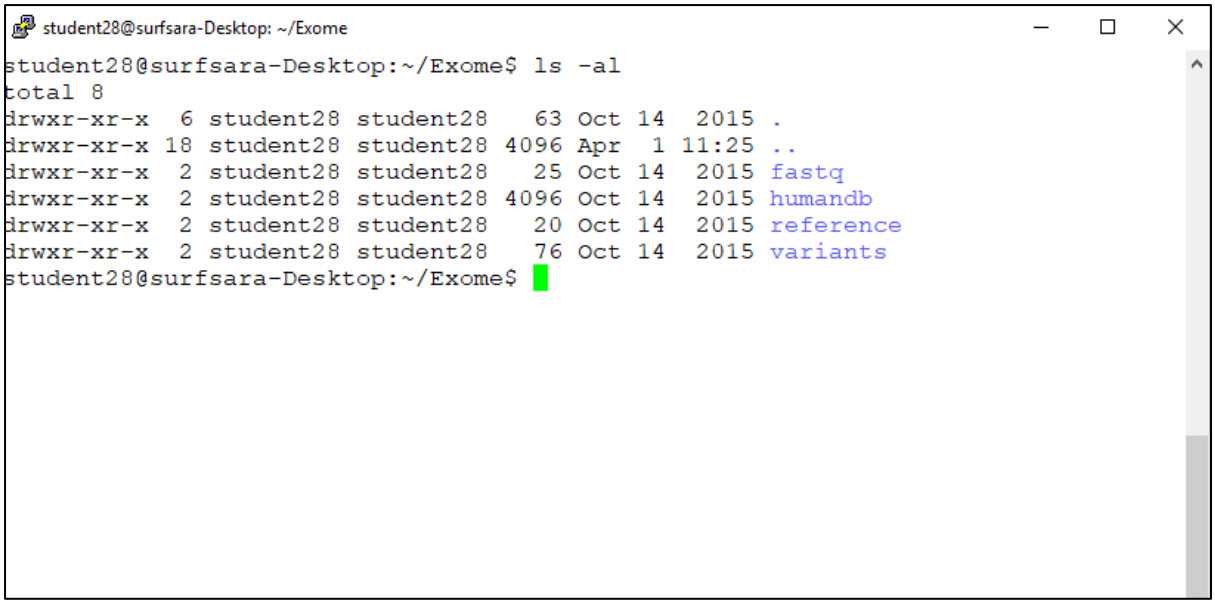


Figure 1. The white or black window is the Unix Terminal, which allows execution of (Linux) commands such as **cd**, **pwd** and **ls**. If you followed instructions correctly, then you will be in the Exome directory as shown. If you type ls –al then you will see 4 sub-directories. The directory 'fastq' contains the raw sequence data.

In order to use BWA you must first create an index of the reference sequence which facilitates fast searching of sequences. The reference sequence of chromosome 9 is in the file chr9.fa, which you can view with the command 'more'

- Now go to the 'reference' directory, which contains the chromosome 9 reference sequence
    - **`cd ..`** (this will take you one directory up)
    - **`cd reference`** (this will take you to the 'reference' directory)
    - **`more chr9.fa`** (use 'space'-key to scroll further, use 'q' to quit and return to the command prompt)

Before we can perform a sequence alignment we need to create a so-called index that will accelerate the alignment. This is done by executing the following command:

- **`bwa index -a bwtsw ./chr9.fa`** #build index for BWA

Explanation of command:

'bwa' is a software application for alignment that is installed as an Unix program. The other items (index, -a, bwtsw, and ./chr9.fa) are arguments to the bwa application that instruct bwa to perform a certain action (in this case, the indexing of the sequence of chromosome 9). The argument ./chr9.fa instructs bwa to read the sequence of chromosome 9 which is found in the current directory (indicated by the dot).

Notes:

- *There should be NO SPACES in ./chr9.fa*
- *All information behind the # are comments and not part of the Unix command. Thus, you do not have to type* #build index for BWA

Next we can align the sequences obtained for one of the NicBar patients against our reference sequence (i.e., chromosome 2). We use the 'BWA' program for doing this.

- **`cd ..`** #go back to the Exome directory
- **`bwa aln -q 20 ./reference/chr9.fa ./fastq/NicBar.fastq > ./fastq/NicBar.sai`** # align sequences and place output in NicBar.sai file.

Explanation of command:
Again the application bwa is used but now with different parameters. The files chr9.fa and NicBar.fastq are used as input. These are found in the directories 'reference' and 'fastq' which can be accessed from the current directory (indicated by the dot). The output is written to the file NiCBar.sai in the directory fastq. This is indicated by the 'redirect' symbol '>' which redirects the output of bwa to this output file instead of the screen. With this information, you should be able to understand the other Unix commando's below also. Don't worry about the exact meaning of all the arguments.
Note:

- *This bwa command should be typed as a single line. Thus do not type 'Enter' after the '>'. Thue command read:*
  - `bwa aln -q 20 ./reference/chr9.fa ./fastq/NicBar.fastq > ./fastq/NicBar.sai`

3.2 Step 2. Visualize the sequence alignments.

To visualize the alignment of the reads you can use the Integrative Genome Viewer (IGV). This will give you a first impression of the data.

You will need to convert the output of BWA (.sai files) to *sorted* BAM files. You also have to make an index of the sorted BAM file. Use samtools to do this file conversion and indexing, and then visualize the alignments in the Integrative Genome Viewerer (IGV). The index should be in the same directory as the BAM file and the names should be the same except for the extension which is BAI for the index. You will also need the BAM files for the next step.

Question 3

a. What is samtools?
b. What is a sam file?
c. What is a bam file?

Use the following commands to visualize the alignment and variants.

- **`bwa samse ./reference/chr9.fa ./fastq/NicBar.sai ./fastq/NicBar.fastq > ./fastq/NicBar.sam`**
- **`samtools view -bhS ./fastq/NicBar.sam > ./fastq/NicBar.bam`**

- **`samtools view -bhq1 ./fastq/NicBar.bam > ./fastq/NicBar.unique.bam`**
- **`samtools sort ./fastq/NicBar.unique.bam ./fastq/NicBar.sorted`**

- # You will need to index the BAM file for viewing the reads in the IGV viewer:
- **`samtools index ./fastq/NicBar.sorted.bam ./fastq/NicBar.sorted.bai`**

- **`igv.sh &`** # starts the IGV viewer, the '&' means the process is started in the background, so you can still type things at the command-prompt.)
  - Select reference genome "hg19" (drop-down menu in top-left corner)
  - Load bam file /home/studentXX/Exome/fastq/NicBar.sorted.bam
  - Select chromosome 9 (chr9 in drop-down menu)
  - Zoom in till you see the aligned reads
  - You can directly go to a location or gene if you type this in the box (locations as 'chrX:<start_position>-<end-position>, for example 'chr1:1000345-1000500'. Or type a gene symbol, for example 'TTN')

Question 4

a. Which variations can you identify in the gene "DMAC1" on chromosome 9?
b. By how many reads (=sequences) are these variants covered?

3.3 Step 3. Quality control for the target capture experiments

In this exercise we will use the statistical package 'R'.

As you should know now, whole genome sequencing it is still rather expensive to achieve sufficient read coverage, for example for the detection of genomic variants. Furthermore, in some cases one might be interested only in some fraction (e.g., the exome) rather than the whole genome. Target capture (target enrichment, targeted sequencing) experiments are a suitable strategy in these situations. The genomic regions of interest are selected and enriched prior to next-generation sequencing. A frequently used approach for the enrichment of the target sequences is based on hybridization with pre-designed probes, either on microarrays or in solution. The hybridized molecules are then captured (eluted from the microarrays or pulled-down from the solution, respectively), amplified and sequenced.

Besides quality control of the sequencing data, it is therefore also crucial to assess whether the capture has been successful, i.e. to determine if

a) most of the sequenced reads actually fall on the target (=exome)
b) the targeted bases reach sufficient coverage

We will use the R/Bioconductor package TEQC (which contains functions for quality control) to assess and visualize the quality of the target enrichment process, like specificity and sensitivity of the capture and per-target read coverage.

The TEQC package is already installed and you can load the package by starting R and then typing

- `cd ~/Exome` # "~" is the equivalent of /home/studentXX
- `R` #start R
- `library(TEQC)` #note; this and the subsequent commands are not given from the Linux prompt but from the R prompt (">")

The design and annotation file (target.bed) of Nimblegen's SeqCap EZ Exome Library v2.0 used for the target capture are located in Exome/variants. You can read in the bed file that describes which regions of the genome are targeted by the Nimblegen probes:

- `targets = get.targets(targetsfile="./variants/target.bed", chrcol=1, startcol=2, endcol=3, skip=0)`

You can have a quick look at (part of) the data with the commands

- `head(targets)`
- `tail(targets)`

Question 5
What does the information in the target bed (and 'targets' variable) represent?

Use the following commands to find out what fraction of the genome is targeted:

- `ft = fraction.target(targets, genome="hg19")`
- `ft`

Question 6
Does the coverage correspond to what you expect? Why or why not?

Next, we load the genomic positions of the aligned reads. For this purpose, you can use the bam file that you produced in an earlier step. The bam file contains the alignment. Read in the bam file using the following command:

- `reads = get.reads("./fastq/NicBar.bam",filetype="bam")`
- `reads`

You can ignore the warning messages and have a quick look at (part of) the data with the commands head(reads) and tail(reads).

Question 7

- How many reads does the bam file contain?
- What information do you see when you look at the data with head(reads)?

One important component of quality control in target capture experiments is to check whether most of the sequenced reads actually fall on target regions. A barplot showing the fraction of reads and targets aligning to each chromosome can give a first impression on that.

- `chrom.barplot(reads, targets)`

Question 8

- Can you give an interpretation of the bar plot? Which chromosome contains most targets and which chromosome contains most reads?

A measure for the capture specificity is the fraction of aligned reads that align with any target region. It can be calculated by function fraction.reads.target:

- `fr1 = fraction.reads.target(reads, targets)`

Since usually the captured DNA molecules are longer than what is actually sequenced, it is expected to have many reads that do not overlap, but are close to the target. Therefore, you can specify an offset that will enlarge every target on each side by the specified number of bases.

- `fr2 = fraction.reads.target(reads, targets, Offset=100)`

Question 9

- What percentage of the reads does overlap with the target?
- What is the percentage of reads that overlaps with the extended target?
- Make a figure to explain fr1 and fr2.

Question 10

With the output from fraction.target (ft) and fraction.reads.target (fr2) the 'enrichment of the target capture experiment can be calculated as the 'fraction of on-target reads / fraction of target within genome'. What is the value of this enrichment? How do you interpret this enrichment?

Besides capture specificity, it is important to check the read coverage within target regions, since it is crucial for follow-up analyses. The function *coverage.target* calculates read coverage for each base that is sequenced and/or located in a target region:

- `Coverage = coverage.target(reads, targets, perTarget=TRUE, perBase=TRUE)`
- `Coverage["avgTargetCoverage"]` #average coverage of all target bases
- `Coverage["targetCoverageSD"]` #standard deviation over all target bases

Note: this command might take some time a perfect moment to get some coffee! Or you can open a second Unix terminal and continue with Step 4 (below)

Since the bam file only contains reads that align to human chromosome 9, it is useful to write the coverage statistics to file for chromosome 9 only:

- `targets2 = Coverage$targetCoverages`
- `targets2 = readsPerTarget(reads, targets2)` # count reads

Look at the results by selecting chromosome 9:

- `head(targets2[targets2$space=="chr9",])`

Question 11
What information is represented by targets2? What do you notice? How does this affect the identification of gene variants?

You can visualize per-base coverages along chromosomal positions using the function *coverage.plot*. First look at the first 9 targets of chromosome 9

- `head(targets2[targets2$space=="chr9",],25)`

We can plot these coverages by using:

- `coverage.plot(Coverage$coverageAll, targets, Offset=100, chr="chr9", Start=116801, End=123289)`

Question 12
- How good is the correspondence between target regions and mapped reads? Can you think of possible reasons for this outcome?

3.4 Step 4. Determine the gene variants.

We have now aligned (and visualized) the sequences obtained for the patient against the chromosome 9. Now we can compare these sequence reads to the reference genome to establish all the differences. Any difference that is identified is a possible single nucleotide polymorphism (SNP) and short insertions and deletions (indels). Use the *VarScan* program to identify all variants for chromosome 9.

Question 13
- What is the purpose of mpileup?
- What is the purpose of VarScan?
- Are all the observed differences true SNPs
- What is a SNP?
- What is an indel?

The next commands are executed from the Unix prompt. Since you are still working in R, first terminate R before continuing:

- q()

You will get a question: 'Save workspace'. Answer 'y'.

Next, first run mpileup to determine all differences between the NicBar sequences and the reference

- `samtools mpileup -f ./reference/chr9.fa ./fastq/NicBar.sorted.bam > ./variants/NicBar.pileup`

Now run VarScan with its default values:

```
# default values:
# Warning: No p-value threshold provided, so p-values will not be calculated
# Min coverage: 8
# Min reads2: 2
# Min var freq: 0.01
# Min avg qual: 15
```

- **`java -jar /data/Programs/VarScan.jar pileup2snp ./variants/NicBar.pileup --coverage 8 > ./variants/NicBar.snp`**
  - **note the double dash "--"**

- **`java -jar /data/Programs/VarScan.jar pileup2indel ./variants/NicBar.pileup --coverage 8 > ./variants/NicBar.indel`**
  - note the double dash "--"

- **Note:** we did not specify all parameters (only coverage). Therefore, by default, the default values as given above are used.

All SNPs and indels are now written to NicBar.snp and NicBar.indel. In the next step we will annotate these variants.

Question 14
What is the meaning of the parameters specified above (minimum coverage, minimum reads2, variant frequency, and minimum average quality)?

3.5 Step 5. Annotate the variants.

Once we have identified the SNPs and indels we need to annotate (describe) these variants with additional information such as the corresponding gene, the location on the chromosome, the type of variation, etc. This will allow use to further analyse the list of variants to identify the variant that causes the disorder. The annotation of variants is done by the program *ANNOVAR*.

Run the following programs to provide part of the snp annotation.

- **`/data/programs/varscan2annovar.pl ./variants/NicBar.snp > ./variants/NicBar.annovar`**

- **`/data/programs/annotate_variation.pl -buildver hg19 -geneanno -dbtype knowngene ./variants/NicBar.annovar ./humandb`**

Note that the output of ANNOVAR is written to *./variants/NicBar.annovar.variant_function*

Question 15

Have a quick look at this file. You now can probably understand part of the information in this file. What type of information do you see in the NicBar.annovar.variant_function file?

3.6 Step 6. The ultimate challenge: identifying the gene(s) that cause Nicolaides Baraitser.

All the variants/genes that were detected for the four Nicolaides Baraitser patients are given in the file NicBar-Variants.txt (in the directory ./variants). However, we will not do the filtering steps in Unix but on your own laptop.

- Download the NicBar-Varients files (txt, xlsx) from Canvas to your laptop.

These files contain the same information. The xlsx file is an Excel file. The txt file is a tab-delimited file for use in R.

You can try to do the filtering in Excel (using the xlsx file) or you can try to do it in R if you want to be challenged (although filtering in Excel has its own issues). Thus, continue with 4.6.1 or with 4.6.2.

Have a careful look at the information that is present in the Excel file (the same information is presented in the text file). The columns in the excel file and text file are described in Appendix A. You will use this information to define your own filtering strategy!

3.6.1 Filtering in Excel

Question 20

The exome experiments for four patients resulted in 307,333 variants in 19,624 genes. All these variants and genes were annotated (NicBar-Variants.xlsx). It is now up to you to identify the gene(s) that are responsible for Nicolaides-Baraitser. Ideally, you should define a sequence of filtering steps that lead to a single candidate gene. Or, if this is not possible, to a small selection of genes that should be validated in the lab. Define a filtering strategy that results in a small list of genes (or a single gene) that occur in all 4 patients. Motivate every filtering step in your strategy. Hint: the disease causing gene is located on chromosome 9 **but do not use this as a filter** since in practice you don't know!!

3.6.2 Filtering in R

We will start with a quick introduction to some R constructs that you will need for the filtering

- Start RStudio and load the text file into the data frame 'annoVar'.

- Create a new script by clicking on the '+' icon in the top-left corner of R Studio. All R commands can be entered and executed from this script window.

First create a so-called data frame which holds all the annotated variant and genes for the four patients. First export the Excel file to a tab-delimited file with name 'NicBar-Variants.txt'. Next read this file into R (make sure R is pointing to the correct folder):

- `annoVar = read.delim("./NicBar-Variants.txt",fill=T)`

You can have a quick look at (part of) the data with the commands

- `head(annoVar)`
- `tail(annoVar)`

Note: by using the cursor keys, you can scroll through previous R commands.

The column names of this data frame (table) can be obtained by the next R command. These columns are explained in Appendix A.

- `dimnames(annoVar)[2]`

Question 16

How many annotated variants and how many columns does the data frame contain (use the R function 'dim')?

For the four patients we have very many variants. One of the main challenges is to find the variant and corresponding gene that causes Nicolaides Baraitser. This is a difficult task! We will use the provided annotation in the variant table to perform this task.

In 'R' you can easily sort, filter, select and combine data from one or more tables. The next commands show a few possibilities. Try the next commands to see how they work!

- `annoVar$Qual1` #extract the Qual1 column from the data frame
- `annoVar[,"Qual1"]` #idem
- `head(annoVar$Qual1,25)` #show the first 25 entries of the Qual1 column
- `sort(annoVar$Qual1)` #sort the Qual1 column
- `sort(annoVar$Qual1,decreasing=TRUE)` #sort the Qual1 column by decreasing order
- `unique(annoVar$Qual1)` #show the unique values in the Qual1 column
- `head(sort(unique(annoVar$Qual1,decreasing=TRUE)),25)` #show the first 25 sorted and unique values in the Qual1 column.
- `length(unique(annoVar$Qual1))`
- `head(annoVar[,c("Gene","Qual1","Qual2")],10)` #show multiple columns

Question 17

- How many unique gene names are included in the data frame?

With the R function 'subset' you can easily select parts of the data frame. For example, filtering on the average base quality of reference-supporting read bases can be done using:

- `sub1 = subset(annoVar,annoVar$Qual1>40)`
- `dim(sub1)`

- `sub1 = subset(annoVar,annoVar$Qual1>40 | annoVar$Qual1==0)` #note the double '=' sign
- `dim(sub1)`

- `sub2 = subset(annoVar,annoVar$dbsnp.1000g=='')` #select entries that don't occur in dbsnp

Note: '|' indicates a logical OR. A '&' indicates a logical 'AND' (see example below)

Question 18
Why is it necessary to include annoVar$Qual1==0 in the subset if we want to select for high qualities (Q>40)?

We can also combine different criteria or perform subsequent filtering steps. For example:

- `sub1 = subset(annoVar,(annoVar$Qual1>40 |annoVar$Qual1==0) &annoVar$Qual2>40)`
- `dim(sub1)`
- `head(sub1,10)`

- `sub2 = subset(sub1,sub1$Func=='exonic')` #note, we now use sub1 in subset!
- `dim(sub2)`
- `head(sub2,2)`

Question 19
What did we accomplish in these filtering steps, i.e., what is not contained in the subset sub2?

You now know how you can make sub selections from the data frame. The last thing you must know is how you can check in how many patients a variant/gene occurs. This is little bit more complex.

First we determine the genes that are contained in the data frame. We use the R command 'factor' because then we can construct a 'table' in the next step

- `f = factor(annoVar$Gene)#NOT REQUIRED`
- `levels(f)` #shows all unique genes in the table

Next we can construct a table for the patients (Sample)

- `t=table(annoVar$Sample,factor(annoVar$Gene))`
- `dim(t)`

- **`t[,1:5]`** #Shows for the first 5 genes how many variants were found in the four patients

Thus, if we want to know which genes occur in 4 samples then we can use the following construct:

- **`selection = t[,which(apply(t,2,min)!=0)]`**
- **`selection`**

This constructs first determines all columns that do not contain a minimum value of 0 (which indicates the absence of variants in a gene in a particular patient) and then returns the indexes (via 'which') of these entries and finally the actual values. If you don't understand this construct, don't worry but just use it in your filtering.

Putting it all together: suppose that we perform a very simple filtering strategy: we only consider variants with a very high quality (Q>59) and want to know which corresponding gene affect all 4 patients. We would proceed as follows:

- **`sub1 = subset(annoVar,annoVar$Qual2>59)`** #make the sub-selection of variants
- **`t=table(sub1$Sample,factor(sub1$Gene))`** #construct the table that shows the occurrence in all patients
- **`selection=t[,which(apply(t,2,min)!=0)]`** #select the affected genes that occur in all 4 patients
- **`dim(selection)`** #show the number of selected genes in 4 patients
- **`selection[,1:5]`** #show the first 5 selected genes.

With all the constructs and examples given above you should now have sufficient skills to do your own filtering.

Question 20

The exome experiments for four patients resulted in 307,333 variants in 19,624 genes. All these variants and genes were annotated (NicBar-Variants.txt). It is now up to you to identify the gene(s) that are responsible for Nicolaides-Baraitser. Ideally, you should define a sequence of filtering steps that lead to a single candidate gene. Or, if this is not possible, to a small selection of genes that should be validated in the lab. Define a filtering strategy that results in a small list of genes (or a single gene) that occur in all 4 patients. Motivate every filtering step in your strategy. Hint: the disease causing gene is located on chromosome 9 **but do not use this as a filter** since in practice you don't know!!

As a starting point you can use the following script:

```
#print column names
print(dimnames(annoVar)[2])

#DEFINE YOUR FILTERING STRATEGY HERE
sub = subset(annoVar,(annoVar$Qual1>=55|annoVar$Qual1==0))
```

```
sub = subset(sub,Qual2>55)
sub = subset(sub, Chr==9)
# END OF FILTERING STRATEGY

#check which of the selected genes occur in 4 patients
t = table(sub$Sample,factor(sub$Gene))
selection = t[,which(apply(t,2,min)!=0)]

#print results
cat("This filtering leads to: ",dim(selection)[2]," genes.\n")
cat("The first 10 genes are: \n")
print(selection[,1:10])
print(dimnames(selection)[2])
```

4 Appendix A

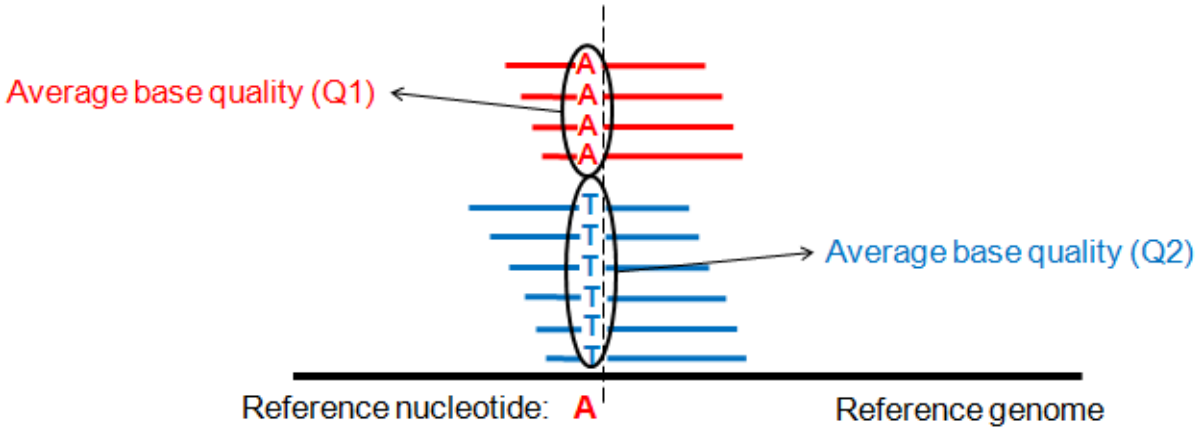
The columns in annotation Excel file and tab-delimited file are described below.

Columns in excel file

1. Sample: Patient 1, 2, 3 or 4
2. Func: The region in which mutation is present, e.g., exonic, splicing site
3. Gene: Gene name
4. ExonicFunc: Type of mutation e.g., synonymous, frameshift
5. AAChange: The change of amino acid due to the mutation. Look for Genetic Code tables to see how this works.
6. Conserved: Determines whether the mutation is in a conserved region of the gene (values>0 indicate conserved areas. The value reflects the amount of conservation). Conservation refers to conservation between organisms and is likely to be of functional importance. Therefore, a mutation in such region has a larger probability of having an effect on the protein sequence.
7. SegDup: Some parts of the genome are duplicated but not exactly the same. This may have consequences for sequence alignment with bwa (e.g., a read may map to multiple regions on the genome) and thus mutation detection. A value larger>0 indicates a duplication.
8. dbSNP: Known SNPs as found in the dpSNP database
9. SIFT: The SIFT score is a statistical p-value that indicates if the variant is likely to have an effect on the protein. Thus lower values reflect variants with more impact.
10. Chr: Chromosome number
11. Ref: Nucleotide found in reference sequence
12. Obs: Nucleotide found in sequence read (obtained from patient)
13. Reads1: Number of reads with the Ref nucleotide (i.e., corresponding to reference sequence)
14. Reads2: Number of reads corresponding with the Obs nucleotide (i.e., corresponding to the read obtained for a patient)
15. VarFreq: Percentage of reads supporting the Obs nucleotide (thus reads with mutation)
16. Strands1: Whether the reference base is found on 1 or 2 strands
17. Strands2
18. Qual1: Average base quality of the nucleotides corresponding to Ref
19. Qual2: Average base qualities corresponding to Obs

Note: if at a certain position we only find variants (thus Reads1=0) then Qual1 is also zero! However, this does then not indicate a low quality but absence of a read (thus a homozygous case). Thus we should not remove entries with Qual1=0 per se.

Some of the columns are further explained in the figure below.



In this figure we see an example where 10 reads (from 1 patient) were aligned to a certain genomic region. At a certain position (indicated with the dashed line) we see an 'A' nucleotide in the reference genome. We see that 4 reads also have a 'A' at that particular position in the alignment, while 6 other reads have a variant ('T'). In the excel file the average base qualities are reported for these two set of reads. This example represents a heterozygous situation.

- Average base quality (Q1) = Qual1 from excel sheet
- Average base quality (Q2) = Qual2 from excel sheet
- Reference nucleotide ('A') = Ref from excel sheet
- Obs from excel sheet = 'T'
- Frequency of reads supporting variant = VarFreq from excel sheet = 6/10 = 60%

5 Appendix B

Vraag: wat is mijn home directory en hoe kom ik daar? Hoe kom ik in andere directories?

Antwoord:

Een directory is equivalent aan een folder of map zoals je dat van Windows of je Mac kent.

Je home directory is de directory waarin al jouw bestanden staan. In de home directory kunnen ook subdirectories (sub folders/mappen) staan.

Zodra je inlogt (bv via Putty of een ander terminal programma) op de Cloud computer dan kom je automatisch terecht in je home directory.

Als je wilt weten in welke directory je precies zit dan geef je het commando **pwd**. Je krijgt dan zoiets te zien als /data/home/student28

In Unix is / de hoogste directory. Dus als je het commando **cd /** geeft kom je in deze directory. Dit kun je weer controleren met **pwd**. Als je vervolgens het commando **cd /data/home/student28** geeft dan kom je weer in je home directory terecht. Controleer met **pwd**. Je had hetzelfde kunnen bereiken met **cd data** en daarna **cd home** en daarna **cd student28.** Dus de / geeft eigenlijk de directory structuur weer.

Je kunt ook altijd terug naar je home directory door simpelweg het commando **cd** te geven of **cd ~**

Als je in je home directory staat geef dan het command **ls -al** Je ziet nu de directory structuur (dus de bestanden en/of subdirectories die in jouw home directory staan. Daar staat ook een subdirectory Exome. Ga nu naar deze directory met het commando **cd Exome.** Controleer met **pwd** in welke directory je zit. Geef het commando **ls -al** om te zien welke andere bestanden en/of subdirectories hierin staan.

Je ziet dan in Exome de volgende subdirectories aanwezig zijn: fastq, humandb, variants, en reference. De totale directory structuur is dus:

```
/data/home/student28/Exome/fastq
                          /humandb
                          /variants
                          /reference
```

Je kunt nu naar de reference directory door middel van het commando **cd reference.** Controleer weer met **pwd**en kijk wat er in deze map staat met **ls -al**. Als het goed is snap je nu waarom cd Exome/reference een foutmelding geeft: dit is omdat in Exome geen subdirectory Exome/reference staat. Als je in je home directory had gestaan dan had dit wel gelukt.

Je kunt dit proberen: ga eerst naar je home directory (bv met **cd ../..** als je in reference staat, of dmv **cd ~**, of simpelweg **cd**). De 2 puntjes geven aan dat je weer een directory omhoog gaat (in dit geval ga je dus 2x een directory omhoog). Controleer weer met **pwd**.

Als je in home directory staat kun je wel **cd Exome/reference** doen. Controleer met **pwd** en geef het commando **ls -al**

Als laatste moet je weten dat de . (punt) je huidige directory aangeeft. Dit kun je proberen. Geef je command **pwd** en kijk in welke directory je staat. Geef nu het command **cd .** en controleer weer in welke directory je staat met **pwd**. Je ziet dat er niets is gebeurd.

Ga nu naar je home directory met **cd**

Je kunt nu naar je Exome directory met cd Exome zoals we eerder hebben gezien. Echter, wat ook werkt is **cd ./Exome**. Dit betekent dus: ga vanuit je huidige directory (.) naar Exome.