

# Genome Capture and Next-Generation Sequencing

Practical Session  
EMBL Heidelberg

Course Materials

*Tobias Rausch*  
*June 2010*

---

# Contents

---

<b>1</b>	<b>Genome Capture</b>	<b>3</b>
1.1	Baits and Target Regions . . . . .	3
1.2	Size Distribution . . . . .	3
1.3	On-target / Off-target . . . . .	4
1.4	GC-Content . . . . .	8
1.5	ChrX and chrY analysis . . . . .	9
1.6	SNP Calling . . . . .	10
1.6.1	1000 Genomes Project . . . . .	11
1.6.2	dbSNP . . . . .	12

---

# Genome Capture

---

## 1.1 Baits and Target Regions

Agilent kindly provided two bed files, one that contains all the bait coordinates and one that contains all the target region coordinates. Let us have a look at these two files first.

```
cd /tmp
mkdir targets
cp /g/solexa/RunVol10/scratch/targets/*.bed /tmp/targets/
cd /tmp/targets
ls
head baits.bed
head targets.bed
```

We now use the UCSC genome browser ([genome.ucsc.edu](http://genome.ucsc.edu)) to visualize the bait and target regions for an arbitrary gene. I picked the ADCK4 gene on chr19 and extracted all baits and targets close to it. Using simple Linux command you can extract the regions for any gene you like. For instance, for the ADCK4 gene, I used:

```
awk '$1=="chr19" && $2>=45882982 && $3<=45920732' baits.bed
awk '$1=="chr19" && $2>=45882982 && $3<=45920732' targets.bed
```

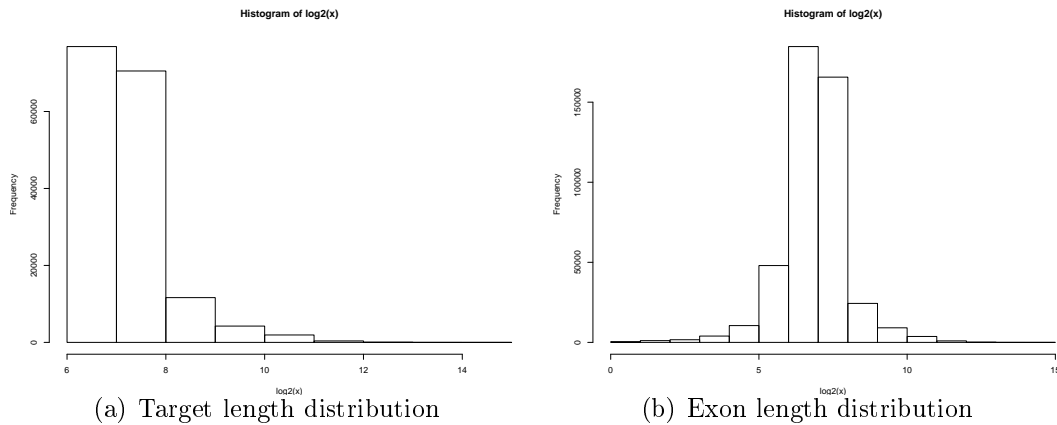
When you open up the UCSC genome browser pick the hg18 version and press submit. Then click manage custom tracks, add the two provided bed files for the ADCK4 gene and go back to the genome browser. Zoom in for one of the exons to get an impression how a single exon is covered by baits and target regions.

## 1.2 Size Distribution

Let us try to get a rough impression how much of the genome we tried to capture. To do this we simply have to sum up the lengths of all target regions. With awk that is a piece of cake.

```
wc -l targets.bed
awk '{SUM+=$3-$2+1} END {print SUM;}' targets.bed
```

1. What is the total number of bases included in all target regions?
2. What is the size of that region compared to the total length of the human genome?



**Figure 1.1:** Comparison of length distribution of exons and targets

We can also plot the lengths of all target regions with R to get an impression of the distribution of these target lengths (see Figure 1.1(a)).

```
awk '{print $3-$2+1;}' targets.bed > targetlen.txt  
R
```

Now we create a simple histogram of the lengths with R.

```
x = scan("targetlen.txt")  
hist(log2(x), breaks=10)
```

Obviously, each target region is at least as long as a single bait. This is not so telling by itself but we can do the same for a list of all exons in the human genome (see Figure 1.1(b)).

1. Make a histogram of the lengths of all exons in the file exon.bed.
2. Compare the exon histogram with the target region histogram. What should you be aware of when you analyze the on-target reads?

If you sum up the lengths of all exons you notice that this sum is much larger than the total length of all target regions. This is due to alternative splicing. Hence, the same exon or overlapping exons appear multiple times in the file. In order to determine the on-exon statistic we need a file with the unique exonic coordinates: basically a map for each base stating if that base is part of an exon or not. This is a bit more tricky and cannot be done easily with simple shell commands (So finally there is a need for a mediocre computer scientist like me). The file exon\_uniq.bed has the non-overlapping exonic regions. If you sum up all regions present in this file you should get a slightly smaller value than the one for the target regions. The reason for the difference are targeted miRNAs and exons shorter than the target region containing it.

```
awk '{SUM+=$3-$2+1} END {print SUM;}' exon_uniq.bed
```

## 1.3 On-target / Off-target

It is time to get hold of some real data. From now on you can choose to work with the SOLiD or Illumina data. Please copy the files to your local tmp directory. This may

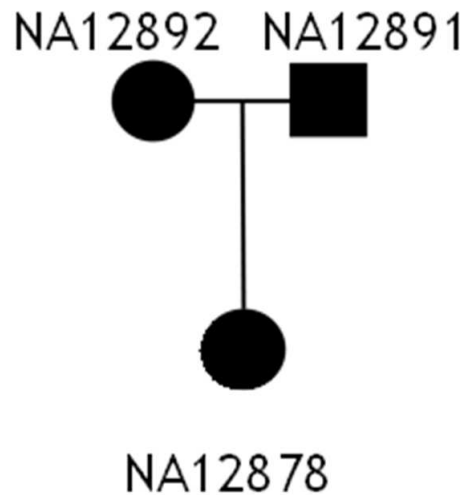


Figure 1.2: HapMap Trio.

take a while because the files are very large. However, the subsequent processing will be quicker since the files are on your local disk then. For SOLiD use:

```
cd /tmp
mkdir solid
cp /g/solexa/RunVol10/scratch/solid/*.bam /tmp/solid/
cd /tmp/solid
ls
```

For Illumina use:

```
cd /tmp
mkdir illumina
cp /g/solexa/RunVol10/scratch/illumina/*.bam /tmp/illumina/
cd /tmp/illumina
ls
```

There are two BAM files, one for the father and one for the daughter from the HapMap trio (see Figure 1.2). We can easily get all reads in a certain window using awk.

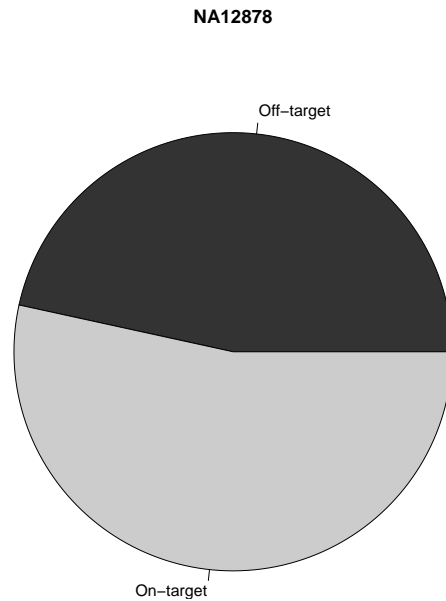
```
samtools view NA12878.bam | awk ' $3=="chr1" && $4>=58932 && $4
<=59892'
```

For all windows this is a bit more time-consuming and I have prepared a coverage file showing the read count for each target region and each exonic region.

```
cp /g/solexa/RunVol10/scratch/illumina/*.cov /tmp/illumina/
cp /g/solexa/RunVol10/scratch/solid/*.cov /tmp/solid/
head NA12878.targets.cov
```

Each region has been annotated by two additional values in the last two columns. The first value is the number of mapped bases and the second value is the number of reads falling in that region. We can quickly sum up the last column to get an idea of how many reads are on target.

```
awk '{SUM+=$6} END {print SUM;}' NA12878.targets.cov
awk '{SUM+=$6} END {print SUM;}' NA12891.targets.cov
```



**Figure 1.3:** Pie chart of on-target reads for NA12878.

We need to put that number into relation to the total number of reads. Luckily, the samtools provide this kind of sequencing statistics using the flagstat command.

```
samtools flagstat NA12878.bam
samtools flagstat NA12891.bam
```

You should see a row stating the number of mapped reads. Use that number and the total number of reads in the target regions to estimate the percentage of reads on-target. You can easily render this information as a pie chart using R (see Figure 1.3 for NA12878).

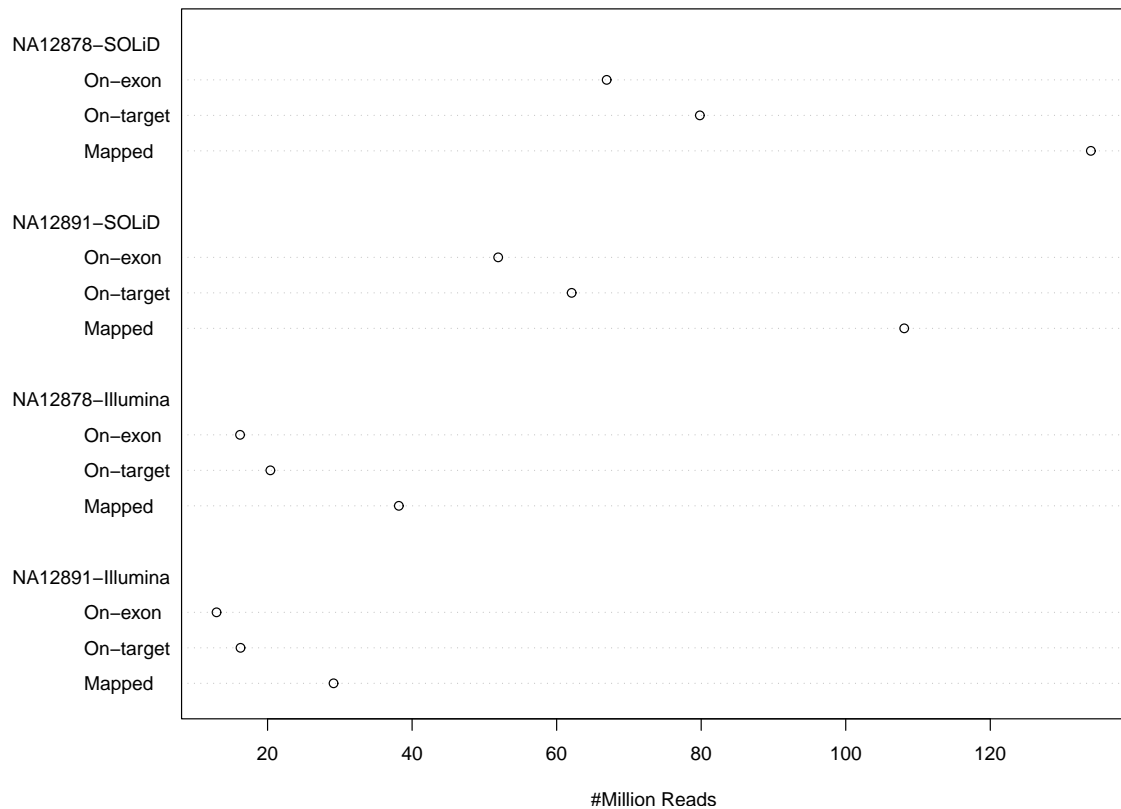
```
mapped=38163843
target=20389403
pie(c(mapped-target, target), c("Off-target", "On-target"), main="
  NA12878", col=gray(c(0.2, 0.8)))
```

Note that for the SOLiD data we only have the mapped reads due to the transformation from color space to sequence space. You can do the same now for the on-exon and off-exon analysis.

1. Compute the number of reads in exonic regions using both \*.exon.cov files.
2. What is the percentage of reads in exonic regions compared to all mapped reads?

I did this on-target / off-target and on-exon / off-exon analysis for all Illumina and SOLiD data sets. The results are summarized in Figure 1.4. If you want to create that plot yourself just create a matrix in R with 3 rows for mapped, on-target and on-exon read counts. Out of this matrix you can easily create the dot chart shown in Figure 1.4.

```
x = c(133915955, 79815980, 66922127, 108092260, 62078729, 51909349,
      38163843, 20389403, 16196544, 29133027, 16259993, 12939470)
mat = matrix(x, nrow = 3, byrow = FALSE)
colnames(mat) = c("NA12878-SOLiD", "NA12891-SOLiD", "NA12878-Illumina",
                 "NA12891-Illumina")
rownames(mat) = c("Mapped", "On-target", "On-exon")
dotchart(mat/1000000, xlab="#Million Reads")
```



**Figure 1.4:** Dot chart of all mapped, on-target and on-exon reads.

Well, there are, of course, some more advanced research questions.

1. How many targets have not a single mapped base?
2. How many of these unmapped targets are equal among all individuals? What might be the reason that these targets are unmapped in all individuals?
3. How uniform have the targets been captured? Calculate the average coverage for each target and make a histogram out of all these values.

Feel free to try your own approach to answer these questions. Here is what I did. The targets without any mapped bases can be easily counted using `wc` and `awk`.

```
awk '$5==0' NA12878.targets.cov > NA12878.unmapped
awk '$5==0' NA12891.targets.cov > NA12891.unmapped
cat NA12891.unmapped | wc -l
```

To compare how many targets have not been mapped in NA12878 and NA12891 we can use `sort` and `uniq`.

```
sort *.unmapped | uniq -d | wc -l
```

The histogram of the coverage values is a bit more involved. We first calculate the average coverage using the number of mapped bases divided by the target region size.

```
awk '{ print $5 / ($3 - $2 + 1); }' NA12878.targets.cov > avg.cov
```

Then we start R and plot the histogram (see Figure 1.5).

```
x=scan("avg.cov")
hist(log2(x))
```

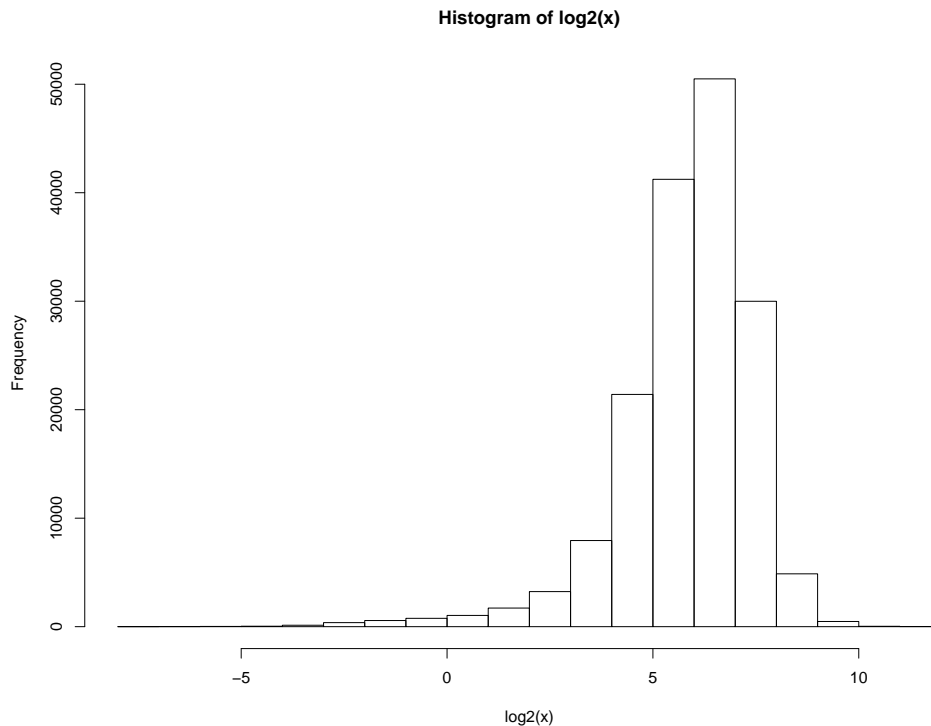


Figure 1.5: Histogram of the average coverage of each target.

## 1.4 GC-Content

As you have noticed the coverage is not uniform across all targets. One contributing factor to this non-uniform coverage is the gc-content. In Figure 1.6 I have plotted the gc-content distribution of all target regions. I have created a file `targets.gc` containing the gc-content of each target region. We are now going to plot the gc-content of a target against its average coverage. Please copy the following files.

```
cd /tmp
mkdir gc
cp /g/solexa/RunVol10/scratch/gc/* /tmp/gc/
cd /tmp/gc
ls
```

Now we create two files, one for the gc-values and one for the corresponding coverage values. Afterwards, we use R to plot the average coverage against the gc-content.

```
cut -f 5 targets.gc > gc
awk '{print $5/($3-$2+1)}' NA12878_illumina.cov > ill
awk '{print $5/($3-$2+1)}' NA12878_solid.cov > sol
```

R

```
x=scan("gc")
y=scan("ill")
plot(x,log2(y))
y=scan("sol")
plot(x,log2(y))
```

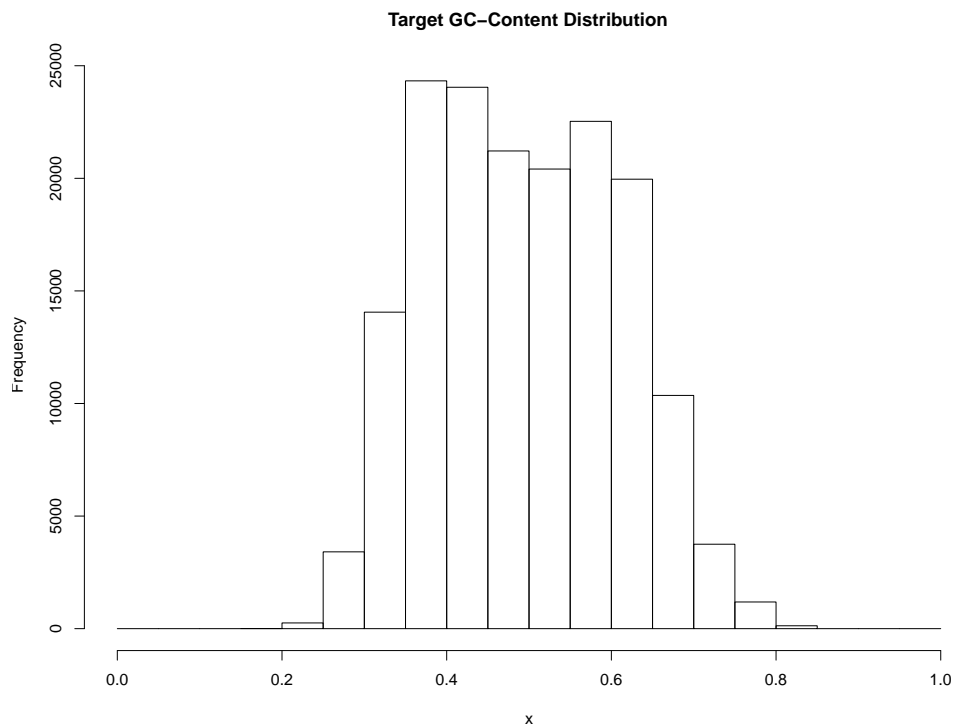
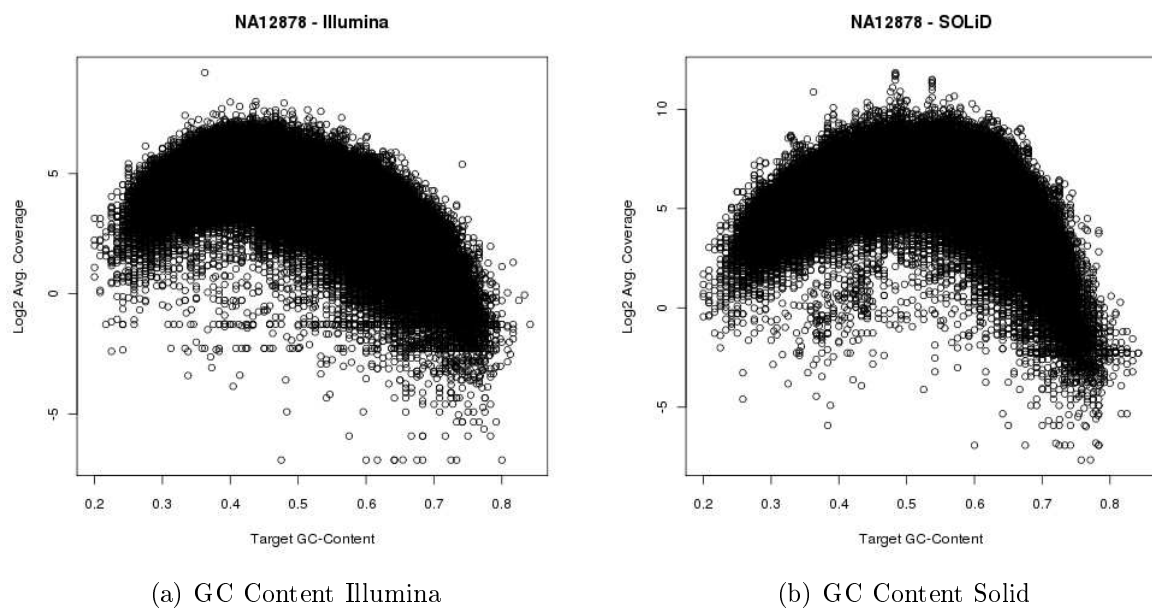


Figure 1.6: Histogram of Target GC-Content



(a) GC Content Illumina

(b) GC Content Solid

Figure 1.7: Avg. Coverage against Target GC-Content

In Figure 1.7(a) and Figure 1.7(b) I created that plot for the Illumina and SOLiD data.

## 1.5 ChrX and chrY analysis

Since the daughter of the HapMap trio has two X chromosomes it should have approximately twice as many reads as the father of the trio. Similarly, we should see only very few reads on chrY for the daughter but much more for the father. Let us quickly calculate

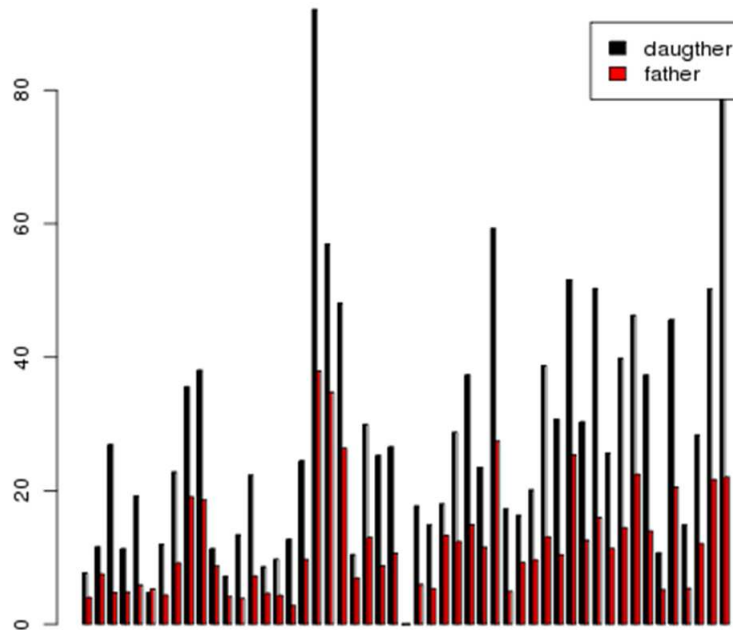


Figure 1.8: Read counts for 50 random targets on chrX.

the number of reads on chrX and chrY.

```
grep "chrX" NA12878.targets.cov | awk '{SUM+=$6} END {print SUM;}'
grep "chrY" NA12878.targets.cov | awk '{SUM+=$6} END {print SUM;}'
grep "chrX" NA12891.targets.cov | awk '{SUM+=$6} END {print SUM;}'
grep "chrY" NA12891.targets.cov | awk '{SUM+=$6} END {print SUM;}'
```

Do not forget to normalize the values by the total number of mapped reads because this differs between the individuals.

1. What relationship did you observe between the number of mapped reads on chrX among NA12878 and NA12891?

In Figure 1.8 I picked 50 random targets on chrX and plotted on the y-axis the number of reads mapping to that target.

## 1.6 SNP Calling

We already introduced the basics of SNP calling. To save some time I already called SNPs on all alignment files using the pileup command from the samtools. Please copy these SNP files to a new directory.

```
cd /tmp
mkdir snp
cp /g/solexa/RunVol10/scratch/illumina/*.snp /tmp/snp/
cd /tmp/snp
ls
```

For the SOLiD files:

```
cd /tmp
mkdir snp
```

```
cp /g/solexa/RunVol10/scratch/solid/*.snp /tmp/snp/
cd /tmp/snp
ls
```

Using some simple Linux commands we can compute some basic statistics.

1. How many SNPs have been called for each individual?

```
wc -l NA12878.snp
```

2. How many SNPs per chromosome?

```
cut -f 1 NA12878.snp | sort | uniq -c
```

3. How many homozygous / heterozygous SNPs have been called?

```
cut -f 4 NA12878.snp | sort | uniq -c
```

Since we sequenced two individuals from the HapMap Trio that were also sequenced in the 1000 genomes project we now have the opportunity to compare our Genome Capture SNP calls with the 1000 genomes data.

### 1.6.1 1000 Genomes Project

The 1000 Genomes project ([www.1000genomes.org](http://www.1000genomes.org)) aims at producing a catalog of human genetic variation such as SNPs and genomic rearrangements. Genomic rearrangements or structural variants include small insertions and deletions as well as large-scale events such as long indels, translocations or inversions. Ultimately, the project will sequence about 2000 people from many different populations around the world. Hence, the project aims at cataloging the *naturally* occurring variation and thus, it provides a great resource of background information for any disease-oriented study. Although the project is still in its pilot phase it is very useful for this course since the same HapMap Trio was used in the 1000 Genomes project to assess coverage issues and different sequencing platforms and centers in the 1000 Genomes project. All of the 3 individuals have been sequenced at 20x to 60x (whole-genome sequencing). At the end of March the 1000 Genomes project released the first set of SNPs for the pilot studies in VCF format ([vcftools.sourceforge.net](http://vcftools.sourceforge.net)). There is a set of tools, called the vcftools, that allow you to process and use these SNP calls.

```
cd /tmp
mkdir gpro
cp /g/solexa/RunVol10/scratch/gpro/* /tmp/gpro/
cd /tmp/gpro
ls
```

You have just copied the file `ceu.geno.vcf` that I downloaded from the 1000 Genomes project web site. It contains the SNP calls with genotype information for the HapMap trio. Let us extract the SNP list for our two sequenced individuals using the vcftools.

```
vcftools --vcf ceu.geno.vcf --indv NA12878 --counts --out NA12878
vcftools --vcf ceu.geno.vcf --indv NA12891 --counts --out NA12891
```

## 1. Genome Capture

---

These commands created two count tables, called NA12878.frq.count and NA12891.frq.count. Let us have a look at these files.

```
head NA12878.frq.count
head NA12891.frq.count
```

Both files have 6 columns. The first column is the chromosome, the second is the position, the third and the fourth columns are the number of alleles and the number of chromosomes respectively (always equal to 2 for a single individual) and the last two columns are the observed allele counts. To compare such a file with our SNP calls I wrote a simple converter that translates the 1000 Genomes SNP file to a pileup file and limits all SNP calls to the exonic regions. The two final SNP files we are going to use in our comparison are NA12878.snp and NA12891.snp.

```
head NA12878.snp
head NA12891.snp
```

The obvious question is to identify the overlap of our genome capture SNP calls with the 1000 Genomes SNP calls. A simple way of doing this comparison is computing all common SNP calls in both files.

```
cut -f 1-4 ../illumina/NA12878.snp > our.snp
wc -l our.snp
sort our.snp NA12878.snp | uniq -d | wc -l
cut -f 1-4 ../illumina/NA12891.snp > our.snp
wc -l our.snp
sort our.snp NA12891.snp | uniq -d | wc -l
```

For the SOLiD data you can use:

```
cut -f 1-4 ../solid/NA12878.snp > our.snp
wc -l our.snp
sort our.snp NA12878.snp | uniq -d | wc -l
cut -f 1-4 ../solid/NA12891.snp > our.snp
wc -l our.snp
sort our.snp NA12891.snp | uniq -d | wc -l
```

What is the percentage of SNPs present in the 1000 Genomes data? Since we have SOLiD and Illumina data we can also first intersect the calls from Illumina and SOLiD and then do the comparison.

```
cut -f 1-4 ../solid/NA12878.snp ../illumina/NA12878.snp | sort | uniq
-d > common.snp
wc -l common.snp
sort common.snp NA12878.snp | uniq -d | wc -l
```

1. What is the number of common, called SNPs among the SOLiD and Illumina data?
2. What is the percentage of common SNPs present in the 1000 Genomes data?

### 1.6.2 dbSNP

The Single Nucleotide Polymorphism Database is a free public archive for genetic variation. dbSNP is a bit of a misnomer since the database also contains, for instance, short

indels. Since this data was accumulated from several different studies we merely use it now to compare our SNP locations with the data in dbSNP. Hence, we do not compare anymore the actually called alleles but simply the SNP locations.

```
cut -f 1-3 ../illumina/NA12878.snp > our.snp
wc -l our.snp
sort our.snp dbSNP.snp | uniq -d | wc -l
```

1. What is the number of SNPs present in dbSNP for the Illumina and SOLiD data?
2. If you only take into account the common SNPs among Illumina and SOLiD. How many of these are present in dbSNP?

---

# Index

---

1000 Genomes Project, 11  
Bait, 3  
BAM, 5  
Chromosomal Analysis, 9  
Coverage, 6  
dbSNP, 12  
Dotchart, 7  
Exons  
    Size Distribution, 3  
    Total Length, 4  
GC-Content, 8  
Genome Capture, 3  
HapMap, 5  
Heterozygous SNPs, 11  
Homozygous SNPs, 11  
On-target Analysis, 4  
Pie chart, 7  
Pileup, 10  
SAM, 5  
SAM Tools, 10  
samtools, 5  
Sequencing Statistics, 6  
Single Nucleotide Polymorphism, 12  
SNP Calling, 10  
Target Region, 3  
Targets  
    Size Distribution, 3  
    Total Length, 4  
    Unmapped targets, 7  
UCSC Browser, 3  
VCF Format, 11  
VCF Tools, 11