

Detecting Large-scale Genetic Variation in the Genome

EMBL PreDoc Course 2011
EMBL Heidelberg

Course Materials

Tobias Rausch
November 2011

Contents

1	Introduction	3
1.1	Getting Started	3
1.2	Data Description	3
2	Read Depth	4
2.1	Alignment Statistics	4
2.2	Detecting Large-Scale Aberrations	5
3	Structural Variant Calling	10
3.1	Detecting Structural Variants	10
3.2	Large Deletions	10
3.3	Tandem Duplications	12
4	SNPs and Short Indel Calling	15
4.1	SNPs	15
4.2	SNP Filtering and Annotation	16
4.3	SNP Annotation using Annovar	18

Introduction

1.1 Getting Started

Please log in to your teaching computer using the username and password provided in the course. Note that in linux, lower and uppercase letters are not the same and need to be entered exactly as written on the blackboard. Once you are logged in please copy the course zip file to your local hard disk. The linux command for this will also be written on the blackboard during the course. If every participant works locally and not across the network the programs should run a lot faster.

Please extract now the zip file using:

```
tar -xzf variantCalling.tar.gz
cd ./scratch
```

All programs used during the course are installed in `./bin`. The data files are in `./data`.

1.2 Data Description

During the whole course we will work with 4 samples coming from the same individual. One sample is a germline sample that we are going to use as a control. The remaining 3 samples are tumor samples. The raw data is available in `./data`, where `g1` is the germline sample and `t1`, `t2` and `t3` are the 3 tumor samples. All 4 samples have been sequenced to 30x using whole-genome short-read sequencing technology. To save time, all the sequenced data has been mapped already to the latest human genome reference sequence, termed `hg19` or synonymously `GRCh37`. The reference sequence is available as a fasta file in `./ref`. The alignments are provided in the `./data` folder in BAM format. Since this is a binary format please make sure you have read the brief SAM/BAM tutorial before starting the exercises below. The data for this tutorial is in the `./tutorial` subfolder. All 4 data sets are confidential. You are not allowed to use the data outside of this course.

Read Depth

2.1 Alignment Statistics

Calling somatic genomic alterations on a full 30x genome would take several hours if not days. Because of that, we have limited the data analysis part of this course to chromosome 17 and chromosome 8. This size of chr8 is 146.364.022 bp, the size of chr17 is 81.195.210 bp. Before starting the actual variant calling, let us first compute some basic alignment statistics using the samtools flagstat command (Li et al., 2009) and some simple shell commands.

1. How many reads are mapped to chr17 for each sample? How many reads are properly paired for each sample?

```
samtools flagstat ../data/g1.chr17.bam
samtools flagstat ../data/t1.chr17.bam
samtools flagstat ../data/t2.chr17.bam
samtools flagstat ../data/t3.chr17.bam
```

2. Extract the insert size of all properly mapped pairs (column 9 in SAM format) for one of the samples. Make a histogram of these insert sizes using R. An example of such a plot is shown in Figure 2.1 and Figure 2.2.

```
samtools view ../data/g1.chr17.bam |
awk 'and($2, 0x0002) && and($2, 0x0040)' |
cut -f 9 | sed 's/^-//' > inserts.txt
```

3. Create some simple statistics about the insert size distribution using the R commands mean, median, range, sd and boxplot.

```
x = scan("inserts.txt")
hist(x)
mean(x)
median(x)
range(x)
sd(x)
boxplot(x)
```

4. How can that insert size distribution be used to identify cutoffs for the classification of paired-ends?

- Given the insert size distribution and the number of mapped pairs what is the average sequencing coverage and what is the average spanning coverage assuming that all reads have a length of 101bp?

The number of mapped reads for each sample can be used to normalize for differential read counts when plotting read depth ratios. The insert size distribution is crucial for calling structural variants and we will revisit that issue later on in Chapter 3.

2.2 Detecting Large-Scale Aberrations

A very simple method to highlight large-scale chromosomal aberrations uses a read-depth approach. We first compute the coverage for each sample in a window-by-window fashion. The non-overlapping windows are of size 10kb. This cannot be done anymore using shell commands, so I have written a tool (Rausch, 2010) that calculates the coverage for a given set of genomic intervals or genome-wide in a window-by-window manner called `cov`.

```
cov -g ../ref/chr17.fa ../data/g1.chr17.bam > g1.cov
cov -g ../ref/chr17.fa ../data/t1.chr17.bam > t1.cov
cov -g ../ref/chr17.fa ../data/t2.chr17.bam > t2.cov
cov -g ../ref/chr17.fa ../data/t3.chr17.bam > t3.cov
head g1.cov
```

Once the program is finished each 10kb window of chr17 has been annotated with the number of reads falling into that window. If we sum up the last column we should get the total number of reads for each sample minus the filtered redundant reads.

```
awk '{SUM+=$4} END {print SUM;}' g1.cov
awk '{SUM+=$4} END {print SUM;}' t1.cov
awk '{SUM+=$4} END {print SUM;}' t2.cov
awk '{SUM+=$4} END {print SUM;}' t3.cov
```

You can compare this output with the previously computed sequencing statistics using `samtools flagstat`. We are now using R to plot the \log_2 ratio for each non-overlapping window comparing one of the tumor samples with one of the normal samples.

```
R
x = read.table("g1.cov")
y = read.table("t1.cov")
plot(log2(y[,4] / x[,4]))
plot(x[,2], log2(y[,4] / x[,4]))
```

So in total we are plotting t1 against g1, t2 against g1 and t3 against g1. The plots are shown in Figure 2.3, Figure 2.4 and Figure 2.5.

- What kind of chromosomal aberrations can you see?

01006_SN169_0174_B80223ABXX/Data/Intensities/BaseCalls/GERALD_16-1
Pairs in plot: 19535450 out of 19545799 (0.9995)
Overall median: 249

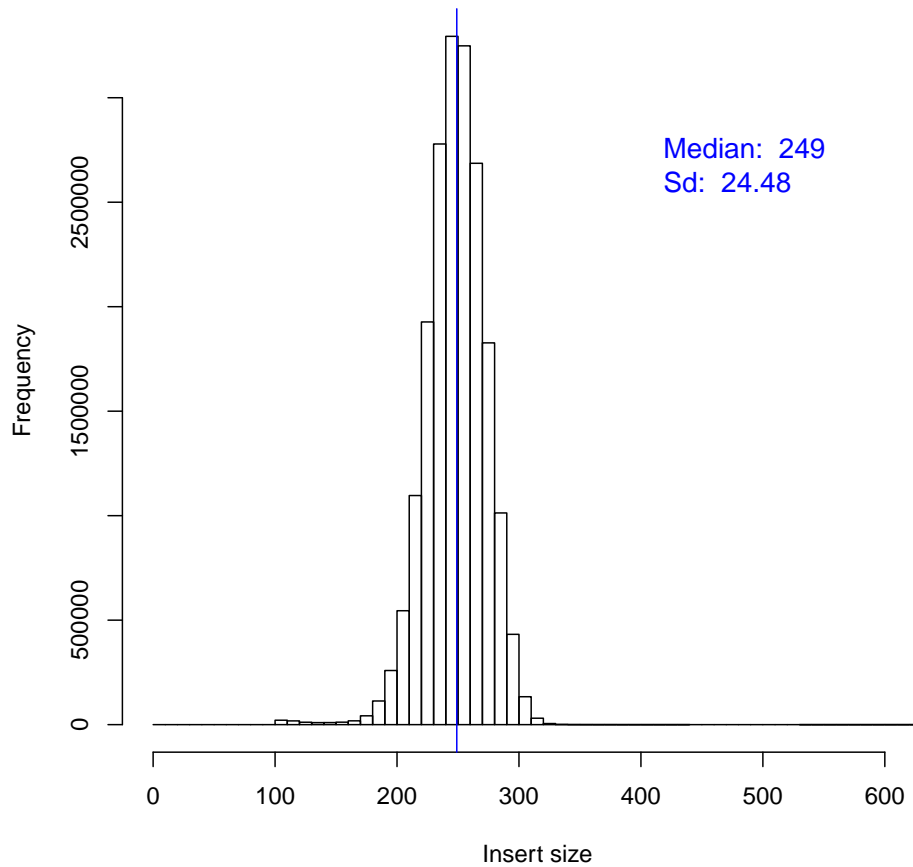


Figure 2.1: Insert size distribution for a paired-end library.

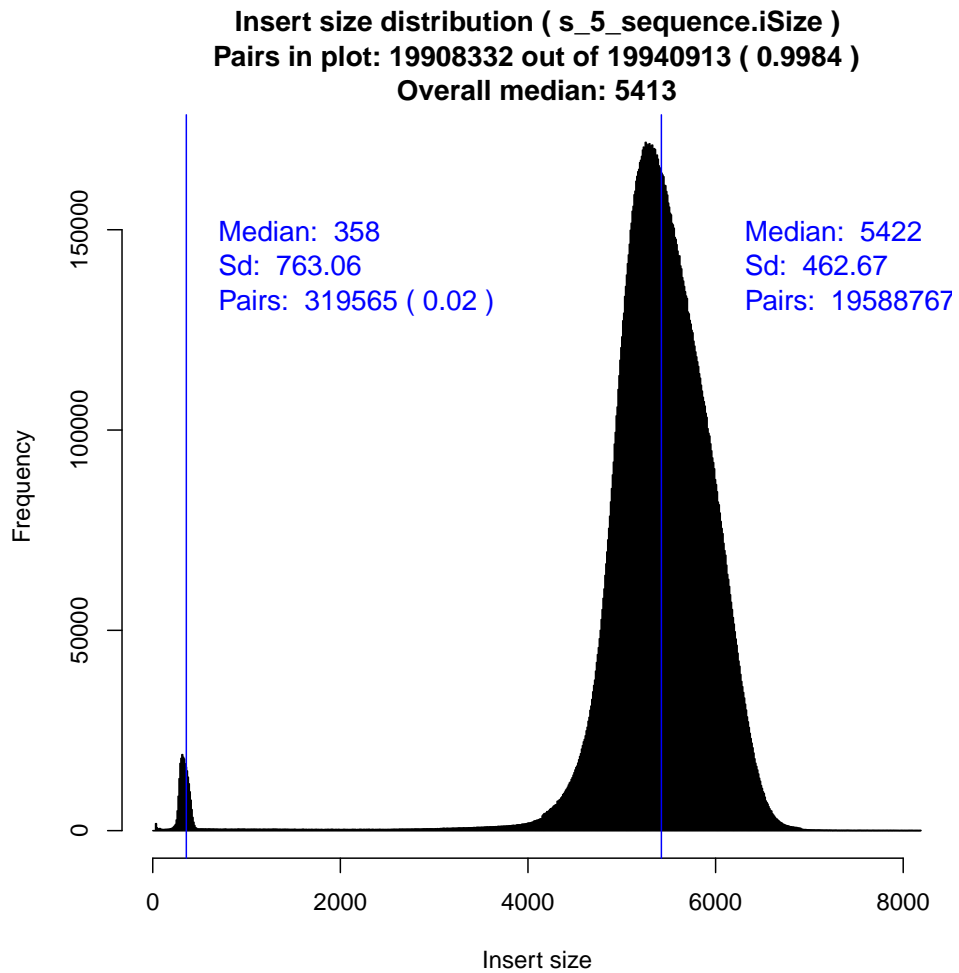


Figure 2.2: Insert size distribution for a mate-pair library.

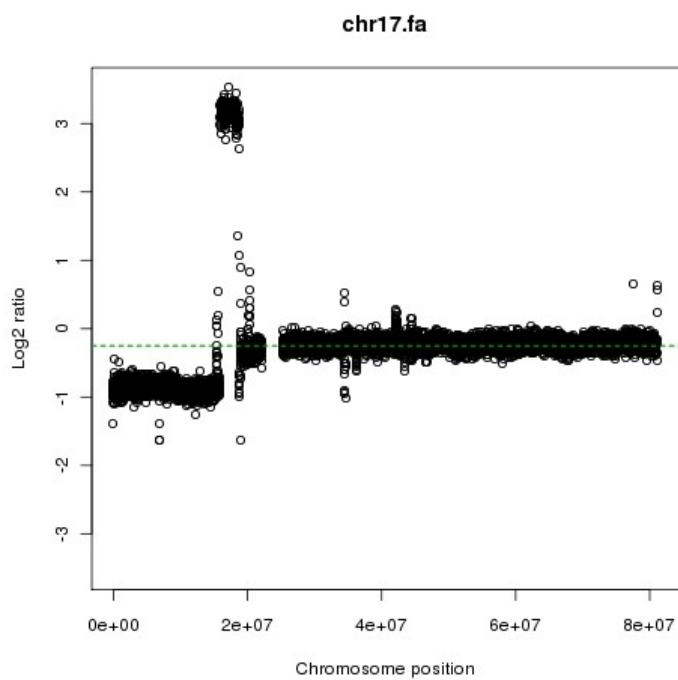


Figure 2.3: Read-depth plot of t1 against g1.

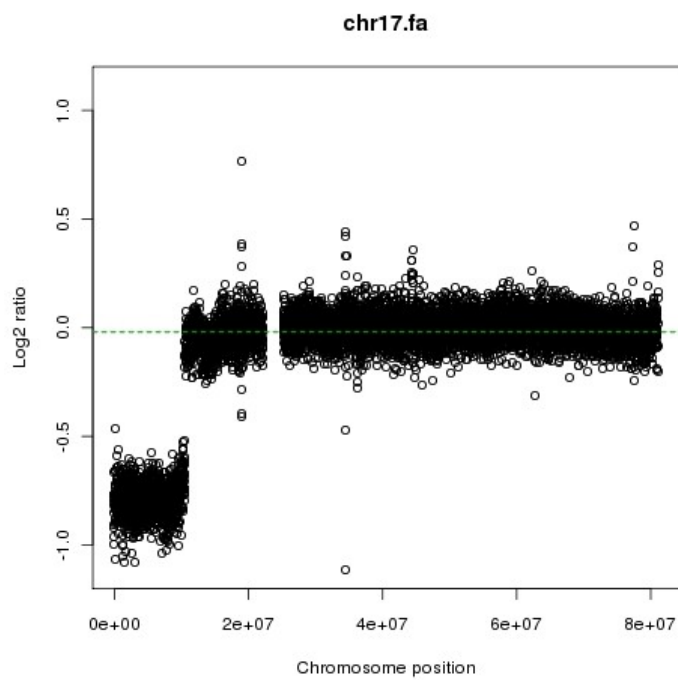


Figure 2.4: Read-depth plot of t2 against g1.

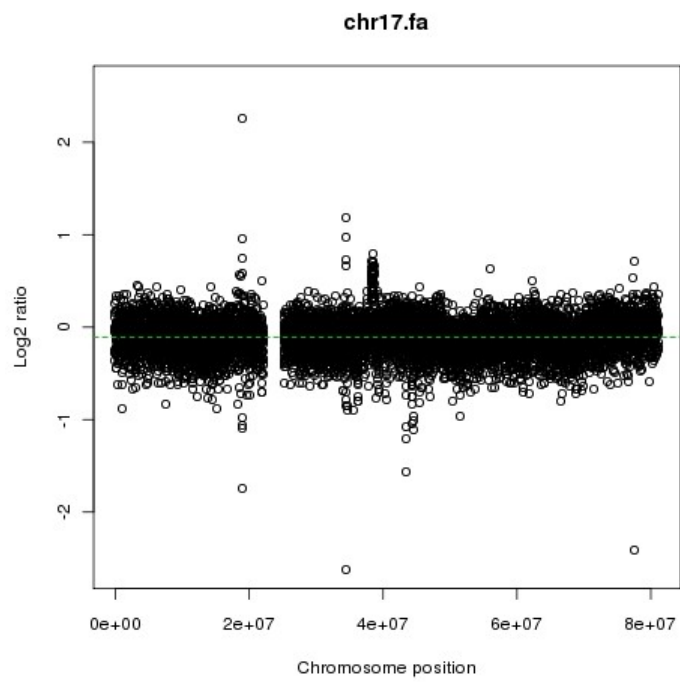


Figure 2.5: Read-depth plot of t3 against g1.

Structural Variant Calling

3.1 Detecting Structural Variants

The most prominent method to detect large structural variants is paired-end mapping (Korbel et al., 2009). The basic idea of this method is to extract all reads having an abnormal mapping distance or an abnormal orientation. Subsequently, these reads are clustered based on regional coherence. For structural variants introducing a copy number change such as deletions or tandem duplications, one also seeks read-depth support. In addition, the next-generation sequencing technologies are constantly improving on the number of bases per read. The longer the reads the more fruitful is also the detection of split-reads to further support a paired-end structural variant call. This has the additional benefit of mapping structural variants at breakpoint resolution. Recently, our lab focused on combining paired-end, read-depth and split-read alignment methods to increase sensitivity and specificity of structural variant calls.

3.2 Large Deletions

We recently developed a paired-end structural variant detection tool called delly. For each raw paired-end call we then try to identify split-reads.

1. Please run delly on chr8 for all samples.

```
delly -pi g1 -g ../ref/chr8.fa -o g1.pe -b g1.br g1.chr8.bam
delly -pi t1 -g ../ref/chr8.fa -o t1.pe -b t1.br t1.chr8.bam
delly -pi t2 -g ../ref/chr8.fa -o t2.pe -b t2.br t2.chr8.bam
delly -pi t3 -g ../ref/chr8.fa -o t3.pe -b t3.br t3.chr8.bam
emacs -nw g1.br
emacs -nw g1.pe
```

The raw paired-end deletion calls are written to g1.pe, t1.pe, t2.pe and t3.pe, respectively. For a subset of these calls split-reads could be found. These split-read supported paired-end calls are in the files g1.br, t1.br, t2.br and t3.br, respectively.

2. Please have a look at the paired-end and split-read call files. Each call starts with a print out of all supporting pairs or split-reads and finally, there comes a summary line. For the paired-end calls, the summary line contains the chromosome as well as the estimated start and end of the deletion. This information is succeeded by the size of the deletion, the number of supporting pairs and a unique deletion id. For the split-reads, the summary line contains the chromosome as well as the exact

breakpoint coordinates of the deletion and the exact size. This information is succeeded by the number of split-reads, the alignment quality of the flanking sequence and the unique deletion id. That deletion id is followed by the sequence of the left and right breakpoint as well as the length of the microinsertion and microhomology if present.

If you use multiple libraries of different insert size (e.g., a paired-end and a mate-pair library) these need to be passed separately to Delly because of the initial insert size cutoff estimation.

1. How many deletions have been identified by paired-end mapping?

```
cat t1.pe | grep ">Deletion" | wc -l
```

2. How many calls could be substantiated with split-reads?

```
cat t1.br | grep ">Deletion" | wc -l
```

3. What are potential problems that could lead to a false positive paired-end call?
4. What are potential problems that could lead to a false positive split-read call?
5. Take one of the split-read supported calls and blast the reads individually? Do you find alternative mappings that would invalidate the split-read alignment?
6. Please use the UCSC genome browser to investigate some of the split-read supported calls. Can you identify any type of mutational events (NHR, NAHR, retrotransposition, Alu, Line) that may have formed these structural variants? Some interesting candidate regions are chr8:16545400-16545800 (t3), chr8:73787800-73793807 (t3), chr8:25066706-25070632 (t2), chr8:135082922-135089025 (t1) and chr8:58116904-58118237 (t3) among many others.

To answer the last question it might be helpful to switch on the following UCSC browser tracks: RefSeq Genes, DGV Struct Var, Repeat-Masker, Simple Repeats and Microsatellites. At least one of the events is several kb in size and encompassing some exons.

1. Can you spot that large deletion already in one of the read-depth plots of chromosome 8 shown in Figure 3.1, Figure 3.2 and Figure 3.3?
2. Given that a deletion affects a gene, what potential functional effect may this have?

Once a somatic alteration of interest has been found, we usually seek further validation of such a structural variant event.

1. How can a deletion be validated using PCR? What is important regarding the primer design?
2. We validated the large deletion using Sanger sequencing. Can you spot the breakpoint in the chromatogram using the traceedit viewer?

```
./traceedit.sh
```

3.3 Tandem Duplications

Using paired-end mapping, we can also detect tandem duplications.

1. What kind of mapping pattern characterizes tandem duplications?
2. Do we expect a read-depth signal?

For tandem duplications one could also try to detect split-reads. How does a split-read identifying a tandem duplication aligns to the reference?

```
duppy -pi g1 -g ../ref/chr8.fa -o g1.dup -b g1.dbr g1.chr8.bam
duppy -pi t1 -g ../ref/chr8.fa -o t1.dup -b t1.dbr t1.chr8.bam
duppy -pi t2 -g ../ref/chr8.fa -o t2.dup -b t2.dbr t2.chr8.bam
duppy -pi t3 -g ../ref/chr8.fa -o t3.dup -b t3.dbr t3.chr8.bam
```

1. How many tandem duplications have been identified by paired-end mapping?
2. How many have been supported by split-reads?
3. Is there any large duplication supported by more than 10 pairs of high mapping quality (>100) that is viewable in the log2 ratio plots of Figure 3.1, Figure 3.2 and Figure 3.3?
4. We haven't discussed so far inversions and translocations. Can these be identified with paired-ends? Do you expect a read-depth signal?

There are two final questions regarding the calling of large structural variants.

- What are the individual strength and weakness of paired-end mapping, split-read alignment and read-depth to detect structural variants?
- Given the numerous on-going sequencing projects do you have any ideas how we can leverage the sequencing data of multiple samples to detect common structural variant sites more confidently in a population?

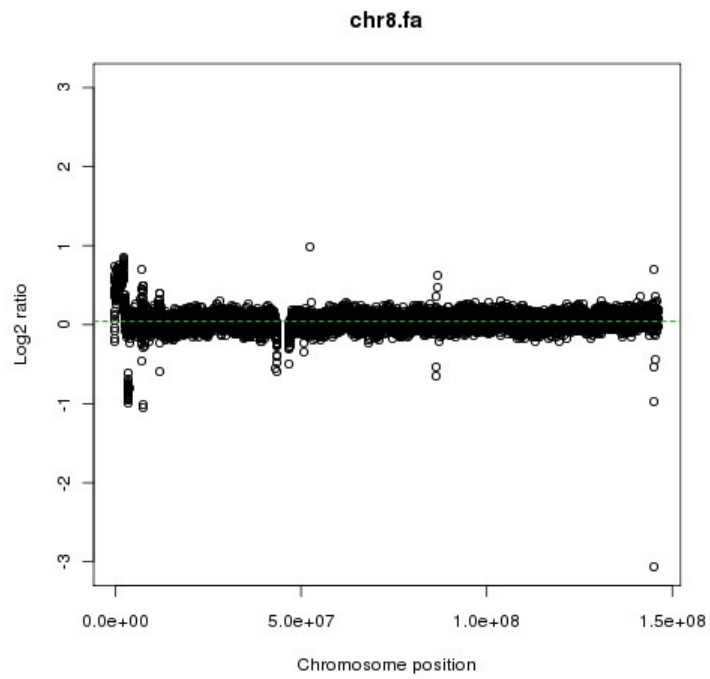


Figure 3.1: Read-depth plot of t1 against g1.

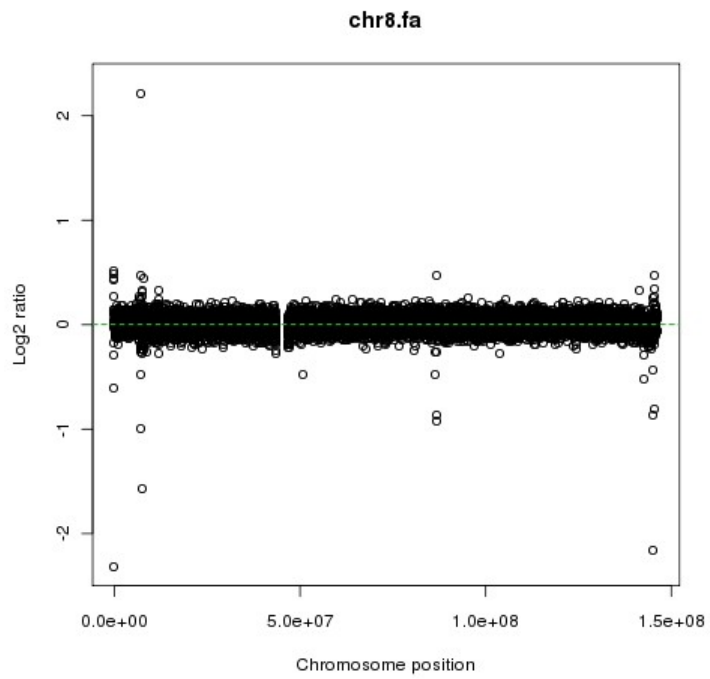


Figure 3.2: Read-depth plot of t2 against g1.

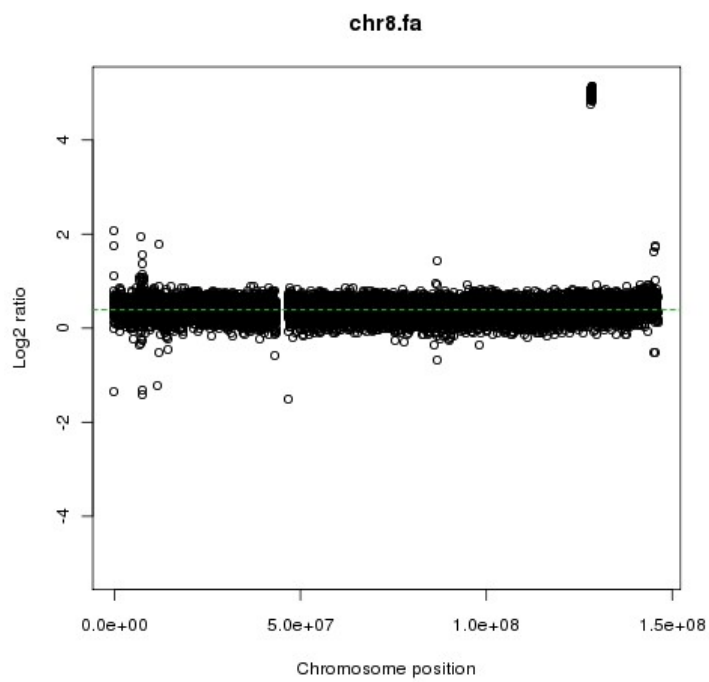


Figure 3.3: Read-depth plot of t3 against g1.

SNPs and Short Indel Calling

4.1 SNPs

Calling SNPs on a full 30x human genome sample takes several hours. Because of that, we restrict the calling in this course to chr17. For SNP calling, the samtools package offers the so-called mpileup command. Please use mpileup on all 4 samples, g1, t1, t2 and t3. Each command should take about 45 minutes for alignments restricted to chr17. For the sake of time you do not need to recompute these results. I have placed the results into the data (the *.vcf files are from samtools mpileup, the *.snps files are from GATK) folder but if you want to try it here are the commands:

```
samtools mpileup -ugf ../ref/chr17.fa ../data/g1.chr17.bam |
  bcftools view -bcvg - > g1.bcf
bcftools view g1.bcf |
  perl vcfutils.pl varFilter -D 100000000 > g1.vcf
```

The columns of that file are briefly explained in Table 4.1. The samtools package also has a simple alignment viewer called samtools tvview. In that viewer, a dot or a comma in the read stands for a match to the reference, either on the forward (dot) or reverse (comma) strand. Any other DNA nucleotide that appears in the read stands for a mismatch. Upper case letters are mismatches on the forward strand whereas lower case letters are mismatches on the reverse strand.

1. Pick one of the called SNPs and compare the mpileup output with the graphical view of the alignment using samtools tvview. A screenshot of tvview for one of the SNPs is shown in Figure 4.1.

```
head -n 12020 g1.chr17.snps | tail -n 40
samtools tvview ../data/g1.chr17.bam ../ref/chr17.fa
```

2. Do the same thing for one of the called indels.
3. How many variants have been called in total for each sample?
4. Please have a look at one of the coding SNPs on chr17 at position 7577539. Is that a silent mutation? If not, what kind of amino acid change does this SNP introduce in the TP53 gene?

Index	Field Name	Description
1	CHROM	Chromosome identifier
2	POS	1-based position. For an indel, this is the position preceding the indel.
3	ID	Variant identifier. Usually the dbSNP rsID.
4	REF	Reference sequence at POS involved in the variant. For a SNP, it is a single base.
5	ALT	Comma delimited list of alternative sequence(s).
6	QUAL	Variant quality.
7	FILTER	Semicolon delimited list of filters that the variant fails to pass.
8	INFO	Semicolon delimited list of variant information.
9	FORMAT	Colon delimited list of the format of individual genotypes in the following fields.
10+	Sample(s)	Individual genotype information defined by FORMAT.

Table 4.1: Brief summary of the Mpileup vcf format.



Figure 4.1: SNP displayed in samtools tview.

4.2 SNP Filtering and Annotation

Obviously, a significant amount of variant sites are false positives. For instance, multiple SNPs occurring right next to each other are almost always due to misalignments. Likewise huge pileups of reads usually signal a collapsed repeat where inter-repeat differences show up as a putative SNPs. Given the large amount of called SNPs, we also lack a proper SNP annotation that allows us to quickly identify coding missense or nonsense mutations. Doing such a manual check-up like you just did for the TP53 mutation is a cumbersome

task. Because of all these limitations many sophisticated SNP callers have been developed in the past. In our lab, we are currently using Annovar (Wang et al., 2010) and the Genome Analysis Toolkit (McKenna et al., 2010) (GATK) to annotate and filter SNPs. Since the annotation using GATK is very time-consuming, I have precomputed this SNP annotation before the course and placed the SNP calls of the Genome Analysis Toolkit of chr17 into the data folder.

1. How many SNPs have been called by the GATK?

```
cat ../data/t1.chr17.snps | cut -f 1-5 | sort | uniq | wc -l
```

2. How many SNPs are novel, how many are present in dbSNP?

```
cat ../data/t1.chr17.snps | awk '$3=="."' | cut -f 1-5 |
sort | uniq | wc -l
```

3. How many SNPs are intronic, how many are coding?

```
cat ../data/t1.chr17.snps | grep "intron" | cut -f 1-5 |
sort | uniq | wc -l
```

4. How many SNPs are missense or nonsense mutations?

```
cat ../data/t1.chr17.snps | grep "CDS" | cut -f 1-5 |
sort | uniq | wc -l
```

Let's focus for the time being on the novel missense and nonsense mutations.

```
awk '$3=="."' ../data/g1.chr17.snps | egrep "missense|nonsense" > g1s
awk '$3=="."' ../data/t1.chr17.snps | egrep "missense|nonsense" > t1s
awk '$3=="."' ../data/t2.chr17.snps | egrep "missense|nonsense" > t2s
awk '$3=="."' ../data/t3.chr17.snps | egrep "missense|nonsense" > t3s
```

Given these novel missense or nonsense mutations let's run some simple cross comparisons.

1. How many SNPs are left per sample? You should take care of SNPs occurring in multiple transcripts.

```
cut -f 1-5 g1s | sort | uniq > g1.snp.txt
cut -f 1-5 t1s | sort | uniq > t1.snp.txt
cut -f 1-5 t2s | sort | uniq > t2.snp.txt
cut -f 1-5 t3s | sort | uniq > t3.snp.txt
wc -l *.snp.txt
```

2. How many are shared among all samples?
3. How many of these are unique to a sample?
4. Have a look again at the shared TP53 mutation. In which samples has it been called homozygous or heterozygous? Any ideas how this can happen if all samples belong to the same individual?

5. How would you validate a SNP and is that necessary? How do you expect a heterozygous SNP to look like in a chromatogram? We provide one example chromatogram, try to find the heterozygous SNP in the tumor and check that position in the normal tissue?
6. Is there a way to rank the individual SNP calls based upon their putative mutational consequences?

4.3 SNP Annotation using Annovar

Annovar cannot yet use SNP calls in vcf format. Therefore we first convert the vcf file to Annovar's input format.

```
convert2annovar.pl -format vcf4 g1.vcf > g1.annovar  
sed -i 's/\.fa\t\t/' g1.annovar
```

For annotation purposes, Annovar relies on the annotation database of the UCSC browser or alternatively user's can provide their own annotation files in bed or vcf format. We are obviously most interested in classification of coding and non-coding variants so we first download the refGene database.

```
mkdir hg19db  
annotate_variation.pl --buildver hg19 --downdb seq hg19db/hg19_seq  
retrieve_seq_from_fasta.pl hg19db/hg19_refGene.txt -seqdir  
hg19db/hg19_seq -format refGene -outfile hg19db/hg19_refGeneMrna.fa
```

This builds the mRNA sequences from the whole genome sequence files. For hg19, UCSC already supplies mRNA sequences so we can directly download these mRNA sequences.

```
annotate_variation.pl --buildver hg19 --downdb refGene hg19db/
```

Using these mRNA sequences, we can quickly annotate synonymous and nonsynonymous SNP sites.

```
annotate_variation.pl --buildver hg19 --outfile g1  
--hgvs g1.annovar hg19db/
```

This simple annotation pipeline can be used for any species in UCSC by simply replacing hg19 with, for instance, dm3 (D. melanogaster) or mm9 (Mouse).

1. How many variants are intergenic, intronic and exonic?
2. How many synonymous and nonsynonymous variants have been called for each sample?
3. What is the overlap between the GATK SNP Calls and the Mpileup SNP Calls?

For human data, Annovar can also annotate segmental duplications, transcription factor binding sites, dbSNPs and SIFT and Polyphen scores. To annotate, for instance, the SNPs present in dbSNP we can use the following.

```
annotate_variation.pl -buildver hg19 -downdb snp132 hg19db/  
annotate_variation.pl --buildver hg19 --filter --dbtype snp132  
g1.annovar hg19db/
```

Bibliography

- J. Korbelt, A. Abyzov, X. Mu, N. Carriero, P. Cayting, Z. Zhang, M. Snyder, and M. Gerstein. Peme: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. *Genome Biology*, 10(2):R23, 2009.
- H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, and . G. P. D. P. Subgroup. The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, 2009.
- A. McKenna, M. Hanna, E. Banks, A. Sivachenko, K. Cibulskis, A. Kernytsky, K. Garimella, D. Altshuler, S. Gabriel, M. Daly, and M. A. DePristo. The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research*, 20(9):1297–1303, 2010.
- T. Rausch. Sequence analysis tools. 2010. URL www.embl.de/~rausch.
- K. Wang, M. Li, and H. Hakonarson. Annovar: functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Research*, 38(16):e164, 2010.

Index

Alignment Statistics, 4
Annovar, 18

Chromosomal Aberrations, 5

Detecting Structural Variants, 10

Introduction, 3

Large Deletions, 10

Read depth, 4

Sample Data, 3
Samtools
 flagstat, 4
 mpileup, 15
Short Indel Calling, 15
Single Nucleotide Polymorphisms, 15
SNP Annotation, 16
SNP Calling, 15
Structural Variant, 10

Tandem Duplications, 12