



LEIDEN UNIVERSITY MEDICAL CENTER

Exome sequencing with GAPSS3

(General annotation Pipeline for Second-generation Sequencers)

Jeroen F. J. Laros

Leiden Genome Technology Center

Department of Human Genetics

Center for Human and Clinical Genetics



In *exome sequencing*, we select genomic regions of interest using a *target-enrichment strategy*.

- PCR.
- On array capture.
- **In-solution capture.**

In *exome sequencing*, we select genomic regions of interest using a *target-enrichment strategy*.

- PCR.
- On array capture.
- **In-solution capture.**

Overview of an in-solution capture.

- Fragmentation.
- Size selection.
- Linker ligation.
- Capture.

In *exome sequencing*, we select genomic regions of interest using a *target-enrichment strategy*.

- PCR.
- On array capture.
- **In-solution capture.**

Overview of an in-solution capture.

- Fragmentation.
- Size selection.
- Linker ligation.
- Capture.

These regions are then *sequenced*.

- Illumina Genome Analyzer II (GAII).
- Illumina HiSeq 2000.

Introduction



Figure 1: GAII.



Figure 2: HiSeq 2000.

Paired end, high throughput, cheap.

Exome sequencing pipelines can roughly be divided in five steps.

Exome sequencing pipelines can roughly be divided in five steps.

1. Pre-alignment.
 - Data cleaning.

Exome sequencing pipelines can roughly be divided in five steps.

1. Pre-alignment.
 - Data cleaning.
2. Alignment.

Exome sequencing pipelines can roughly be divided in five steps.

1. Pre-alignment.
 - Data cleaning.
2. Alignment.
3. Variant calling.
 - Description of difference with the reference.

Exome sequencing pipelines can roughly be divided in five steps.

1. Pre-alignment.
 - Data cleaning.
2. Alignment.
3. Variant calling.
 - Description of difference with the reference.
4. Filtering.
 - Variant calling has a high false positive rate.

Exome sequencing pipelines can roughly be divided in five steps.

1. Pre-alignment.
 - Data cleaning.
2. Alignment.
3. Variant calling.
 - Description of difference with the reference.
4. Filtering.
 - Variant calling has a high false positive rate.
5. Annotation.

We use the Trimmomatic / FASTX toolkit for data cleaning.

- Remove linker sequences.
- Clip low quality reads at the end of the read.
- Judge the read that is left over.

We use the Trimmomatic / FASTX toolkit for data cleaning.

- Remove linker sequences.
- Clip low quality reads at the end of the read.
- Judge the read that is left over.

The FASTQC toolkit is used for quality control (both before and after the data cleaning step).

- GC content.
- GC distribution.
- Quality scores distribution.
- ...

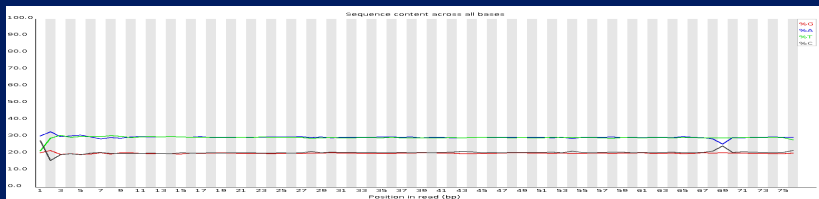


Figure 3: Per base sequence content.

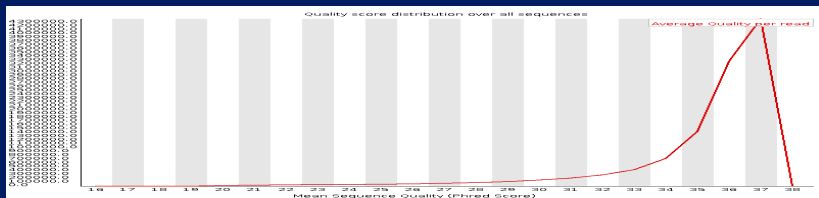


Figure 4: Per sequence quality.

Stampy: A statistical algorithm for sensitive and fast mapping of Illumina sequence reads.

Stampy: A statistical algorithm for sensitive and fast mapping of Illumina sequence reads.

Some features:

- Base quality recalibration.
 - First map 1% of the input.
 - Recalibrate the Fastq quality scores.
 - Redo the alignment with the recalibrated scores.

Stampy: A statistical algorithm for sensitive and fast mapping of Illumina sequence reads.

Some features:

- Base quality recalibration.
 - First map 1% of the input.
 - Recalibrate the Fastq quality scores.
 - Redo the alignment with the recalibrated scores.
- Uses BWA for the hard work.
 - Switches to its accurate built in aligner when BWA fails.

Burrows-Wheeler Aligner (BWA) is a short read aligner that allows small insertions and deletions.

Variant calling

Variant calling is done by Samtools, BCFtools / VCFutils.

The output of most modern aligners is in *Sequence Alignment / Map* (SAM) format.

Variant calling

Variant calling is done by Samtools, BCFtools / VCFutils.

The output of most modern aligners is in *Sequence Alignment / Map* (SAM) format.

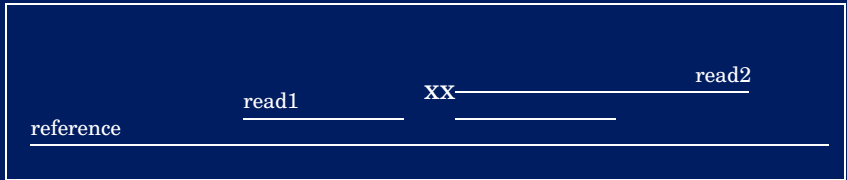
Mainly file format conversions.

- **SAM** → BAM.
- BAM → BAM.sorted.
- BAM.sorted → BAM.sorted.index.
- BAM.sorted → mpileup (**BAQ realignment**).
- BAM.sorted → BCF.
- BCF → **VCF**.

We end up with a list in *Variant Call Format* (VCF).

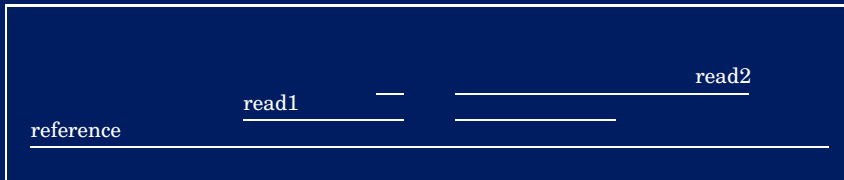
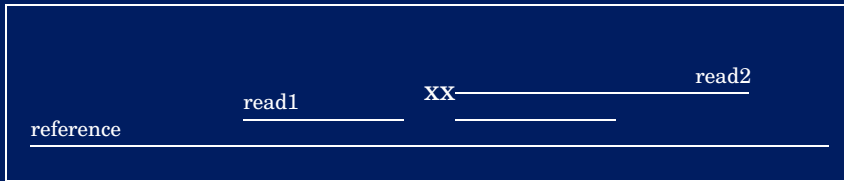
Variant calling

Base Alignment Quality (BAQ) realignment:
 Remove SNPs around indels.



Variant calling

Base Alignment Quality (BAQ) realignment:
Remove SNPs around indels.



We use different annotation sources and an in-house database.

We use different annotation sources and an in-house database.

A selection of SeattleSeq annotation:

- Is the variant known?
- Does it hit a gene?

We use different annotation sources and an in-house database.

A selection of SeattleSeq annotation:

- Is the variant known?
- Does it hit a gene?
 - Is it in an intron?
 - Does it hit a splice site?

We use different annotation sources and an in-house database.

A selection of SeattleSeq annotation:

- Is the variant known?
- Does it hit a gene?
 - Is it in an intron?
 - Does it hit a splice site?
 - Is it in the coding region?
 - Is there a gain/loss of a stop codon?
 - Does the variant result in a frameshift?
 - ...

We use different annotation sources and an in-house database.

A selection of SeattleSeq annotation:

- Is the variant known?
- Does it hit a gene?
 - Is it in an intron?
 - Does it hit a splice site?
 - Is it in the coding region?
 - Is there a gain/loss of a stop codon?
 - Does the variant result in a frameshift?
 - ...
 - Is it in the 5'/3' UTR of a gene?
 - ...

We use different annotation sources and an in-house database.

A selection of SeattleSeq annotation:

- Is the variant known?
- Does it hit a gene?
 - Is it in an intron?
 - Does it hit a splice site?
 - Is it in the coding region?
 - Is there a gain/loss of a stop codon?
 - Does the variant result in a frameshift?
 - ...
 - Is it in the 5'/3' UTR of a gene?
 - ...
- Is it in a regulatory region?
- ...

Combining all these tools in a pipeline:

```
1 bwa aln -t 8 $reference $i > $i.sai  
2 bwa samse $reference $i.sai $i > $i.sam  
3 samtools view -bt $reference -o $i.bam $i.sam
```

Listing 1: Shell script

Combining all these tools in a pipeline:

```

1  bwa aln -t 8 $reference $i > $i.sai
2  bwa samse $reference $i.sai $i > $i.sam
3  samtools view -bt $reference -o $i.bam $i.sam

```

Listing 1: Shell script

```

1  %.sai: %.fq
2      $(BWA) aln -t $(THREADS) $(call MKREF, $@) $< > $@
3
4  %.sam: %.sai %.fq
5      $(BWA) samse $(call MKREF, $@) $^ > $@
6
7  %.bam: %.sam
8      $(SAMTOOLS) view -bt $(call MKREF, $@) -o $@ $<

```

Listing 2: Makefile

Name	Manual	Options
bwa	490	±40
samtools	616	±70
fastx	?	±30

Table 1: Selection of used tools

The number of parameters is impressive, but the number of combinations is incredible.

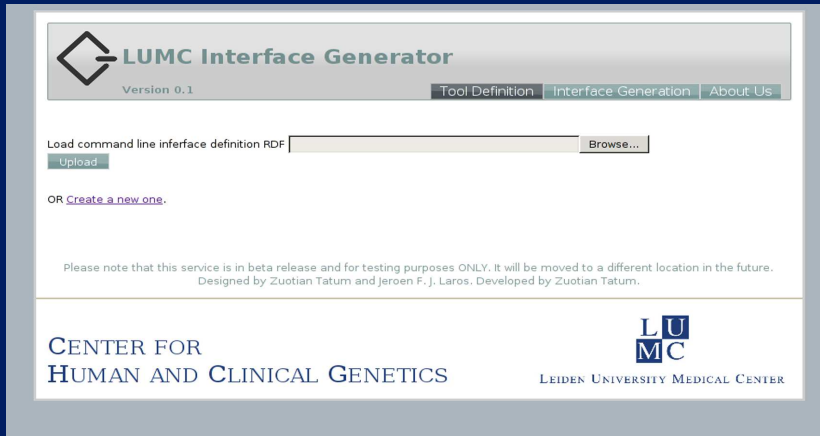
Name	Manual	Options
bwa	490	± 40
samtools	616	± 70
fastx	?	± 30

Table 1: Selection of used tools

The number of parameters is impressive, but the number of combinations is incredible.

Sometimes, people want to tweak these parameters, which is not very practical without a graphical interface.

Graphical interfaces



The screenshot shows the LUMC Interface Generator web application. At the top left is the LUMC logo. The main header contains the text "LUMC Interface Generator" and "Version 0.1". To the right of the header are three navigation tabs: "Tool Definition", "Interface Generation", and "About Us". Below the header, there is a section for loading a command line interface definition. It includes a text input field, a "Browse..." button, and an "Upload" button. Below this, there is a link that says "OR [Create a new one.](#)". A disclaimer text states: "Please note that this service is in beta release and for testing purposes ONLY. It will be moved to a different location in the future. Designed by Zuotian Tatum and Jeroen F. J. Laros. Developed by Zuotian Tatum." At the bottom, the footer contains the text "CENTER FOR HUMAN AND CLINICAL GENETICS" on the left and the LUMC logo with "LEIDEN UNIVERSITY MEDICAL CENTER" on the right.

Figure 5: Interface generator

Graphical interfaces

Parameters

i Parameter name cannot contain **white space** and following characters: - . () , ; \$

▶ Reference

▼ Genotype

Type	<input type="text" value="None"/>	Default Value (use "," to separate values for select type)
Name	<input type="text" value="Genotype"/>	<input type="text" value="True"/>
Argument	<input type="text" value="-g"/>	Label (short description)
Display	<input type="text" value="show"/>	<input type="text" value="Compute genotype likelihoods"/>
Repeatable	<input type="text" value="False"/>	Help (hint)
		<input type="text" value="Compute genotype likelihoods and output them in the binary call for"/>

▶ Uncompressed

▶ Input

▶ Output

▶ Add Parameter

Figure 6: Using the interface generator

Mpileup

Compute genotype likelihoods:
 ▼
 Compute genotype likelihoods and output them in the binary call format (BCF).

Output uncompressed BCF:
 ▼
 Similar to the Genotype parameter, except that the output is uncompressed BCF, which is preferred for piping.

Input :
 ▼

Generated By:
 LUMC Interface Generator (0.1)
 2011-09-03T14:29:36.793452Z

Based On:
 RDF Definition of "Mpileup"
 2011-09-02T16:17:29.010890Z

Generate BCF or pileup for one or multiple BAM files. Alignment records are grouped by sample identifiers in @RG header lines. If sample identifiers are absent, each input file is regarded as one sample.

Figure 8: User friendly interface with Galaxy

NGS course Genomic Resequencing

15/16

Thursday, 5 April 2012

Acknowledgements

Michiel van Galen
Martijn Vermaat
Zuotian Tatum
Yu-Ching Lai
Michel Villerius
Jaap van der Heijden
Bradley ten Broeke
Johan den Dunnen

<https://www.mutalyzer.nl/projects/GAPSS3/>
<http://www.lgtc.nl/>