UNIVERSITÄT LEIPZIG

ECSEQ BIOINFORMATICS

A guide to NGS alignments

A Practical Introduction to NGS Data Analysis

Juriquilla
June 13th 2018

Speaker:
Verónica Jiménez Jacinto
Slides:
Dr. David Langenberger
Dr. Mario Fasold

Index

1. Introduction to High Throughput Sequencing.
    - A short history of High Throughput Sequencing
    - current Platforms
    - Fragmentation problem
    - task: Mapping
        - Problem: repeat region
        - Problem: sequencing error
            - Library Preparation
    - Sequencing applications

    …

Index

- Sequencing applications
  - DNA sequencing
    - De-novo genome assembly
    - Genome re-sequencing
    - SNP discovery
    - Genome rearrangements
    - Exome sequencing / Targeted Sequencing
  - RNA sequencing
    - RNA-seq library preparation
    - De-novo transcriptome assembly
    - Gene prediction
    - Splice site detection
    - Isoform differentiation
    - differential expression.

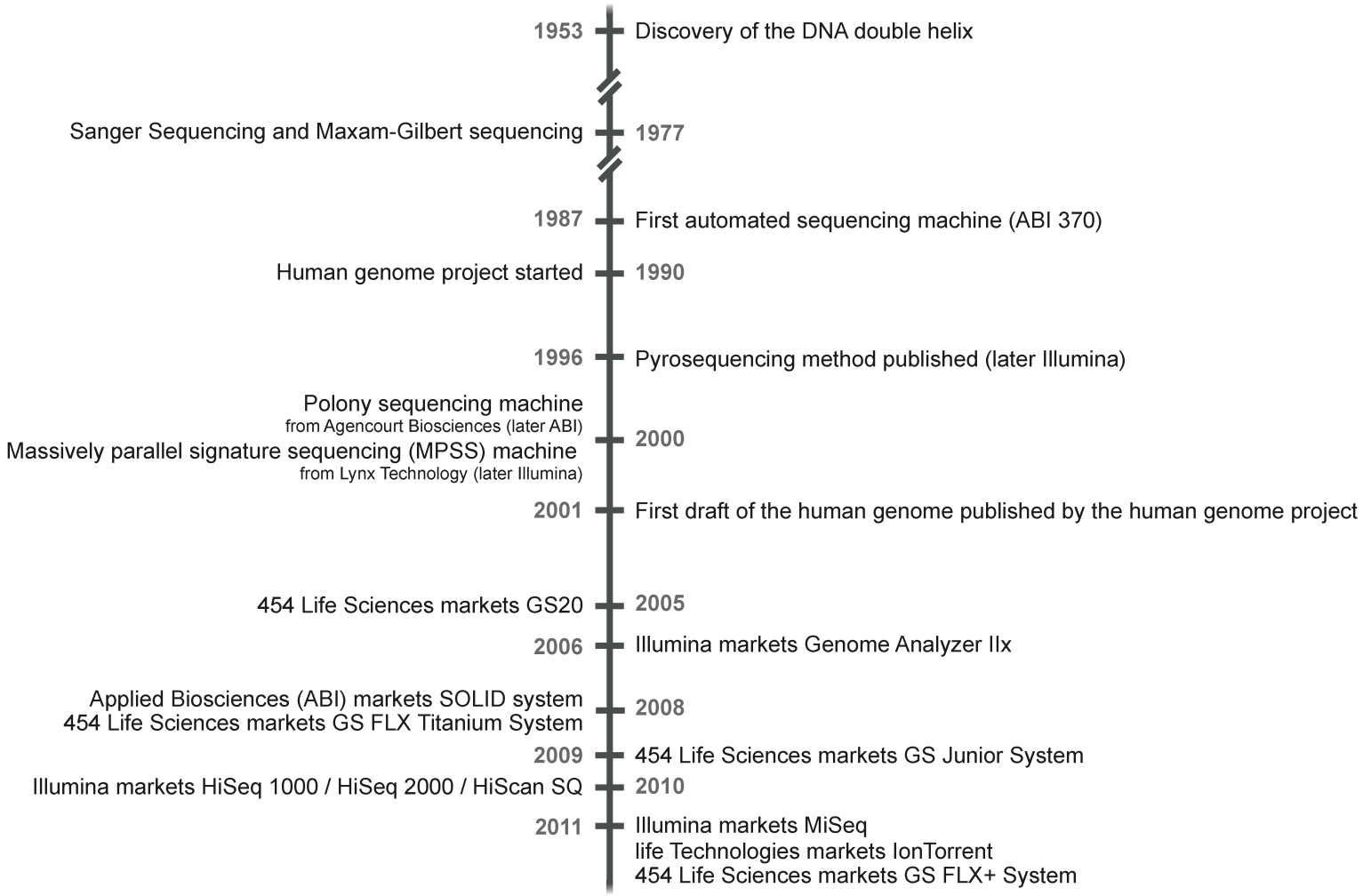
Index

Index

Introduction to High Throughput Sequencing

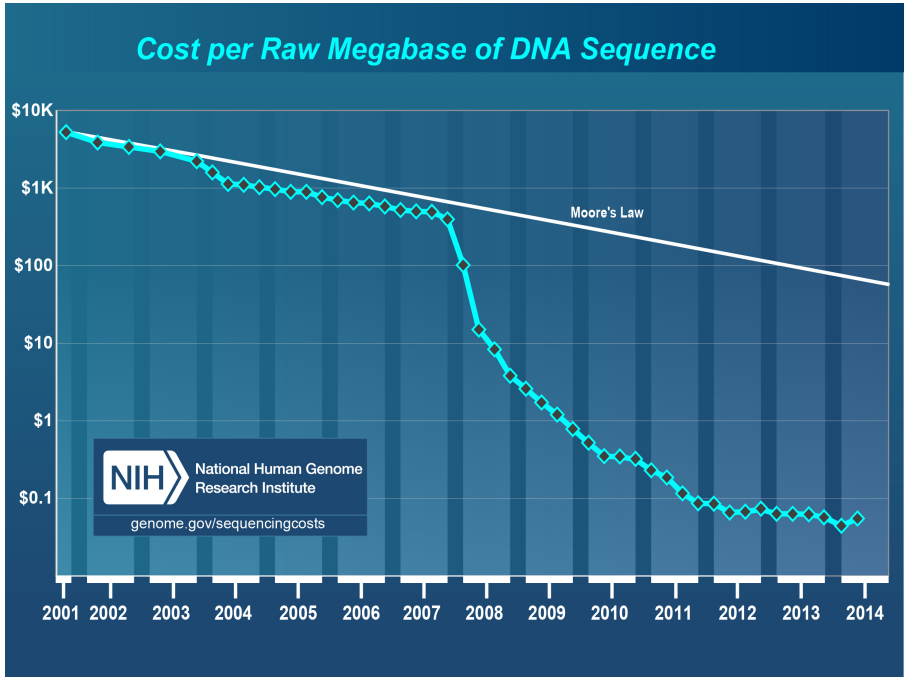
A short history of High Throughput Sequencing

History of DNA sequencing

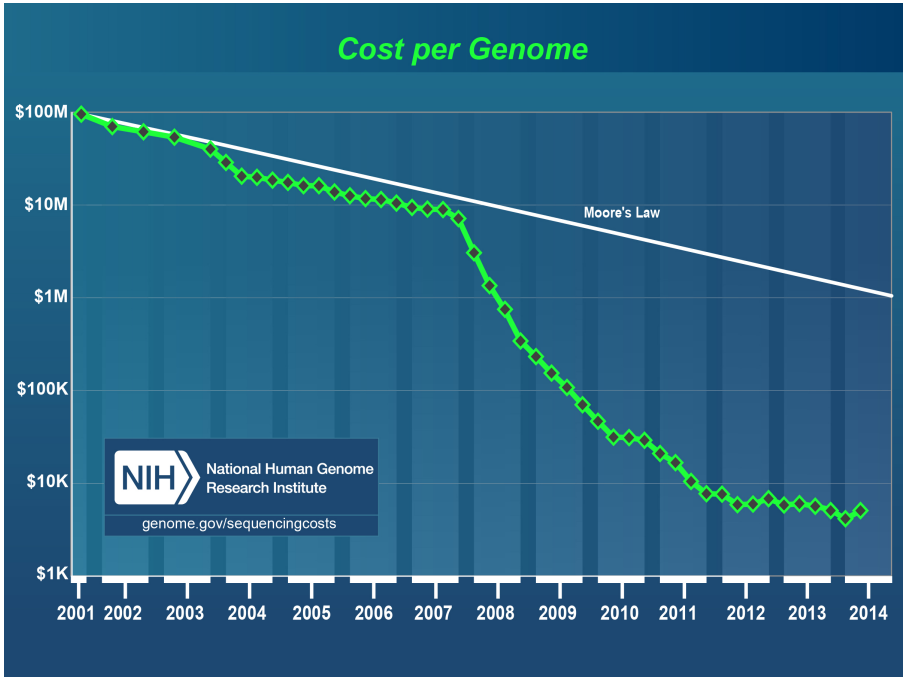


History of DNA sequencing

Cost per Raw Megabase of DNA sequence



Cost per Genome



Current Platforms

Illumina Sequencing Platforms

miSeq

NextSeq 500

HiSeq 2500

HiSeq X Ten

Sequencing workflow

Illumina HiSeq 2500

Machine cost: $740k

Run time: 6 days

Run cost per Gb: $29

Read Length: 2x125 bp

 # single reads: 4B

Throughput: 1 Tb/run

Error Rate: <0.1%

Illumina HiSeq X Ten

Machine cost: $1M*

Run time: 3 days

Run cost per Gb: $7

Read Length: 2x150 bp

 # single reads: 6B

Throughput: 1.8 Tb/run

Error Rate: <0.1%

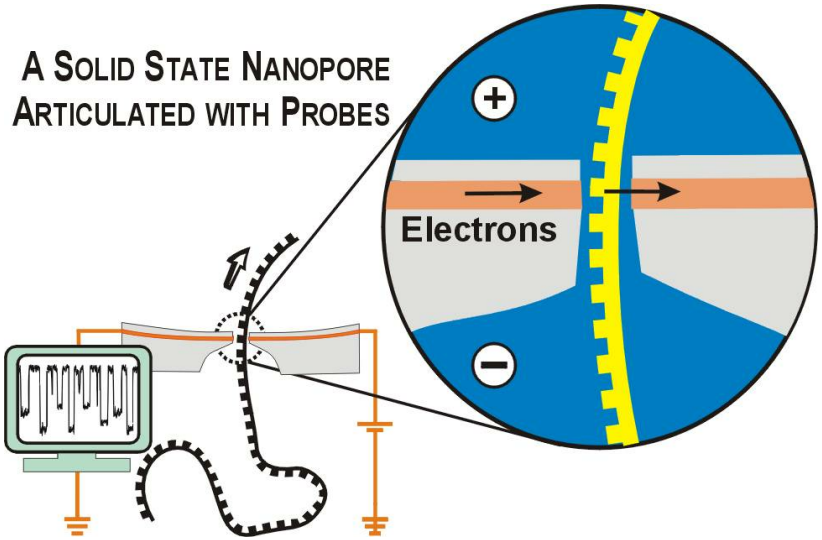
* Minimum purchase of 10 machines

Illumina: Overview

	Run time	Read length	Throughput		Cost	
	(hrs)	(bp)	# reads	bases/run	machine	per Gb
miSeq	65	2 x 300	25M	15Gb	$125k	$93
NextSeq 500	29	2 x 150	400M	129Gb	$250k	$33
HiSeq 2500	144	2 x 125	4B	1Tb	$740k	$29
HiSeq X Ten	72	2 x 150	6B	1.8Tb	$1M	$7

Outlook

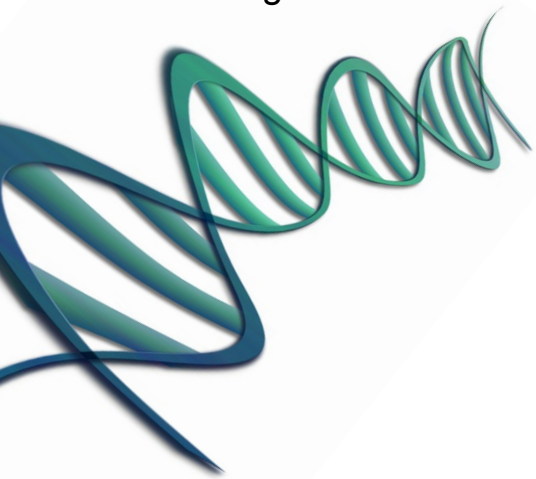

MinION


A Solid State Nanopore
Articulated with Probes
+
Electrons
−

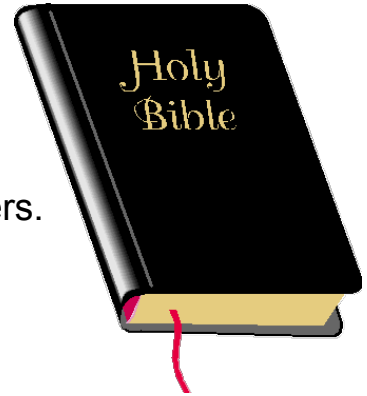
Fragmentation problem

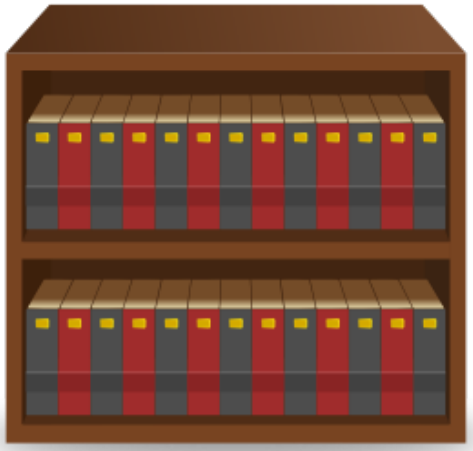
Fragmentation problem

The human genome consists of ~3.2 billion letters.



In comparison, the bible consists of ~4.4 million letters.





That means that one would need ~226 bibles
to store the human genome.

Fragmentation problem

To parallelize and thus speed up the sequencing process, the genome has to be fragmented into smaller pieces (~100-300nt long).

1. In the third year of the reign of Jehoiakim king of Judah came Nebuchadnezzar king of Babylon unto Jerusalem, and besieged it.
2. And the Lord gave Jehoiakim king of Judah into his hand, with part of the vessels of the house of his god: which he carried into the land of Shinar to the house of his god; and he brought the vessels into the treasure house of his god.
3. And the king spake unto Ashpenaz the master of his eunuchs, that he should bring *certain* of the children of Israel, and of the king's seed, and of the princes;
4. Children in whom *was* no blemish, but well favoured, and skilful in all wisdom, and cunning in knowledge, and understanding science, and such as *had* ability in them to stand in the king's palace, and whom they might teach the learning and the tongue of the Chaldeans.
5. And the king appointed them a daily provision of the king's meat, and of the wine which he drank: so nourishing them three years, that at the end thereof they might stand before the king.

3.2 billion letters

Fragmentation problem

To parallelize and thus speed up the sequencing process, the genome has to be fragmented into smaller pieces (~100-300nt long).

1. In the third year of the reign of Jehoiakim king of Judah came Nebuchadnezzar king of Babylon unto Jerusalem, and besieged it.
2. And the Lord gave Jehoiakim king of Judah into his hand, with part of the vessels of the house of his god: which he carried into the land of Shinar to the house of his god; and he brought the vessels into the treasure house of his god.
3. And the king spake unto Ashpenaz the master of his eunuchs, that he should bring *certain* of the children of Israel, and of the king's seed, and of the princes;
4. Children in whom *was* no blemish, but well favoured, and skilful in all wisdom, and cunning in knowledge, and understanding science, and such as *had* ability in them to stand in the king's palace, and whom they might teach the learning and the tongue of the Chaldeans.
5. And the king appointed them a daily provision of the king's meat, and of the wine which he drank: so nourishing them three years, that at the end thereof they might stand before the king.

provision of the king's meat, and

them a daily provision of the king's meat,

and of the princes

the reign of Jehoiakim king of Judah came Nebuchadnezzar

was no blemish, but well favoured, and skilful

In the third year of the reign of Jehoiakim

carried into the land of Shinar to the house of his god

~6 billion pieces

Task: Mapping

One main task in NGS bioinformatics is to find the origin of the pieces in a long reference sequence.

1. In the third year of the reign of Jehoiakim king of Judah came Nebuchadnezzar king of Babylon unto Jerusalem, and besieged it.
2. And the Lord gave Jehoiakim king of Judah into his hand, with part of the vessels of the house of his god: which he carried into the land of Shinar to the house of his god; and he brought the vessels into the treasure house of his god.
3. And the king spake unto Ashpenaz the master of his eunuchs, that he should bring *certain* of the children of Israel, and of the king's seed, and of the princes;
4. Children in whom *was* no blemish, but well favoured, and skilful in all wisdom, and cunning in knowledge, and understanding science, and such as *had* ability in them to stand in the king's palace, and whom they might teach the learning and the tongue of the Chaldeans.
5. And the king appointed them a daily provision of the king's meat, and of the wine which he drank: so nourishing them three years, that at the end thereof they might stand before the king.

provision of the king's meat, and

them a daily provision of the king's meat,

and of the princes

the reign of Jehoiakim king of Judah came Nebuchadnezzar

was no blemish, but well favoured, and skilful

In the third year of the reign of Jehoiakim

carried into the land of Shinar to the house of his god

Problem: Repeated regions

Some regions in the genome are repeated and thus it is hard to find the correct origin.

1. In the third year of the reign of Jehoiakim king of Judah came Nebuchadnezzar king of Babylon unto Jerusalem, and besieged it.
2. And the Lord gave Jehoiakim king of Judah into his hand, with part of the vessels of the **house of his god**: which he carried into the land of Shinar to the **house of his god**; and he brought the vessels into the treasure **house of his god**.
3. And the king spake unto Ashpenaz the master of his eunuchs, that he should bring *certain* of the children of Israel, and of the king's seed, and of the princes;
4. Children in whom *was* no blemish, but well favoured, and skilful in all wisdom, and cunning in knowledge, and understanding science, and such as *had* ability in them to stand in the king's palace, and whom they might teach the learning and the tongue of the Chaldeans.
5. And the king appointed them a daily provision of the king's meat, and of the wine which he drank: so nourishing them three years, that at the end thereof they might stand before the king.

Problem: Sequencing errors

The sequencing of the pieces creates errors (typos), which makes it impossible to find the error-free positions of origin in the reference.

1. In the third year of the reign of Jehoiakim king of Judah came Nebuchadnezzar king of Babylon unto Jerusalem, and besieged it.
2. And the Lord gave Jehoiakim king of Judah into his hand, with part of the vessels of the house of his god: which he carried into the land of Shinar to the house of his god; and he brought the vessels into the treasure house of his god.
3. And the king spake unto Ashpenaz the master of his eunuchs, that he should bring *certain* of the children of Israel, and of the king's seed, and of the princes;
4. Children in whom *was* no blemish, but well favoured, and skilful in all wisdom, and cunning in knowledge, and understanding science, and such as *had* ability in them to stand in the king's palace, and whom they might teach the learning and the tongue of the Chaldeans.
5. And the king appointed them a daily provision of the king's meat, and of the wine which he drank: so nourishing them three years, that at the end thereof they might stand before the king.

provision of the king's meat, and

them a daily provision of the king's meat,

and of the princes

the reign of Jehoiakim king of Jud**e**h came Nebuchadnezzar

was no blemish, but well fav**u**ured, and skilful

In the third year of the re**iiii**ign of Jehoiakim

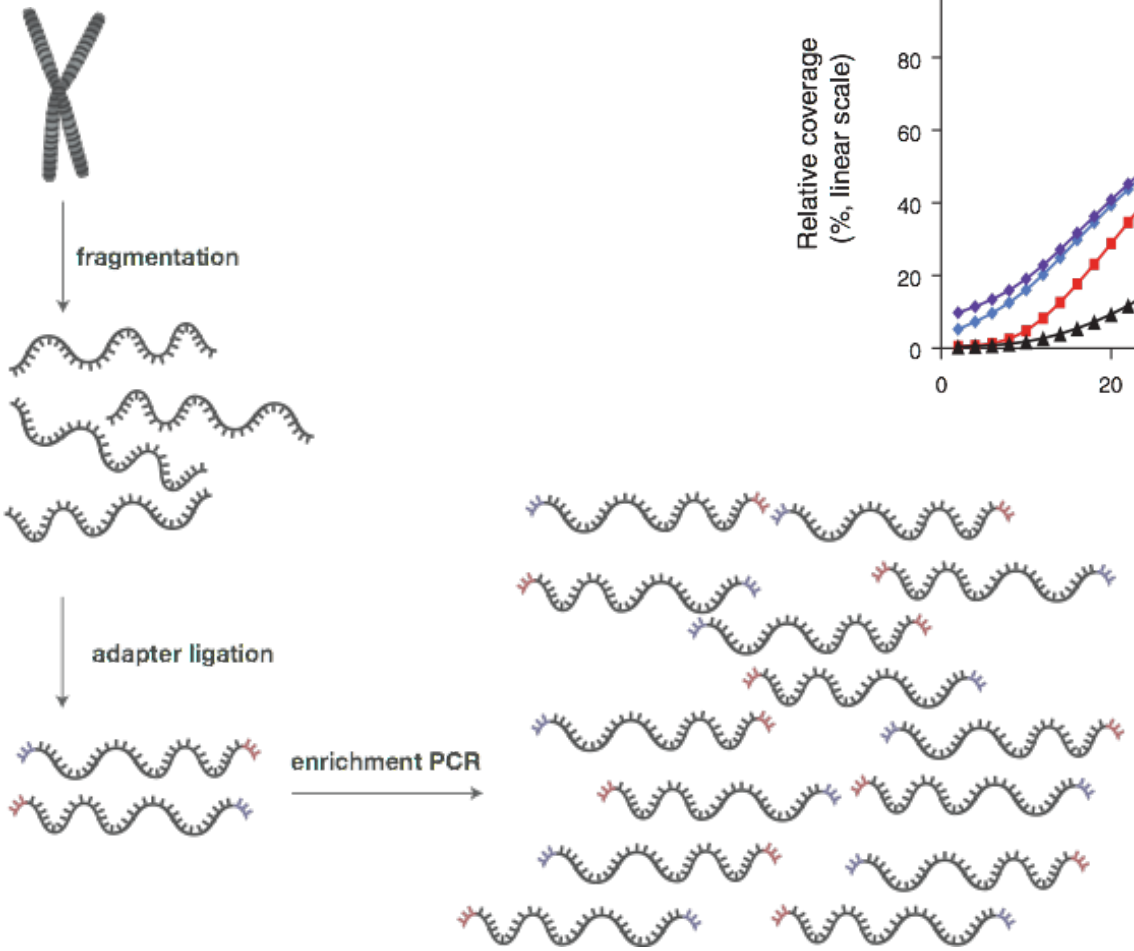
carried into the land of Sh**w**nar to the house of his god

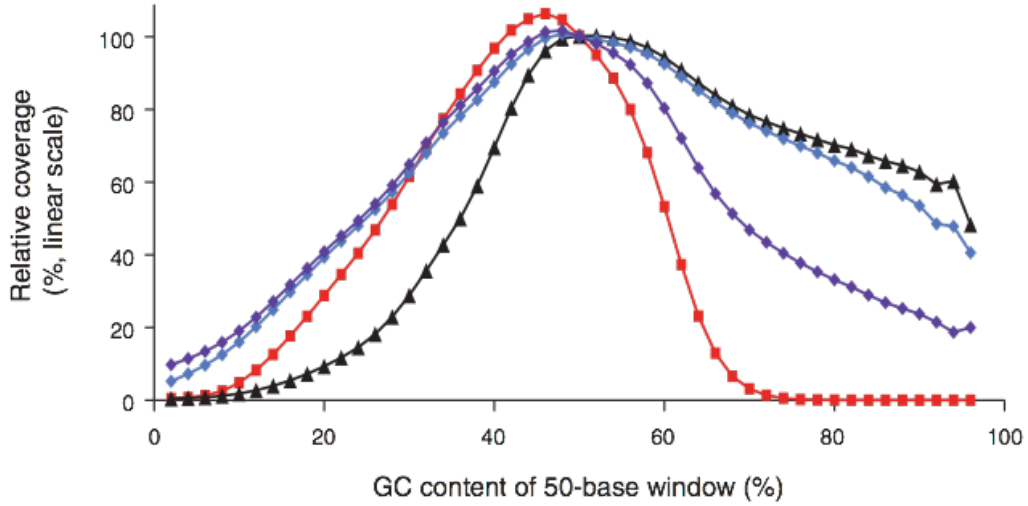
Possible sequencing errors

Possible sequencing errors

Platform	Substitutions	GC bias	AT bias	InDels
Illumina	+++	+	+	+
454	+	+	+	+++
IonTorrent	+++	+	+	+++
PacBio	+++			+++

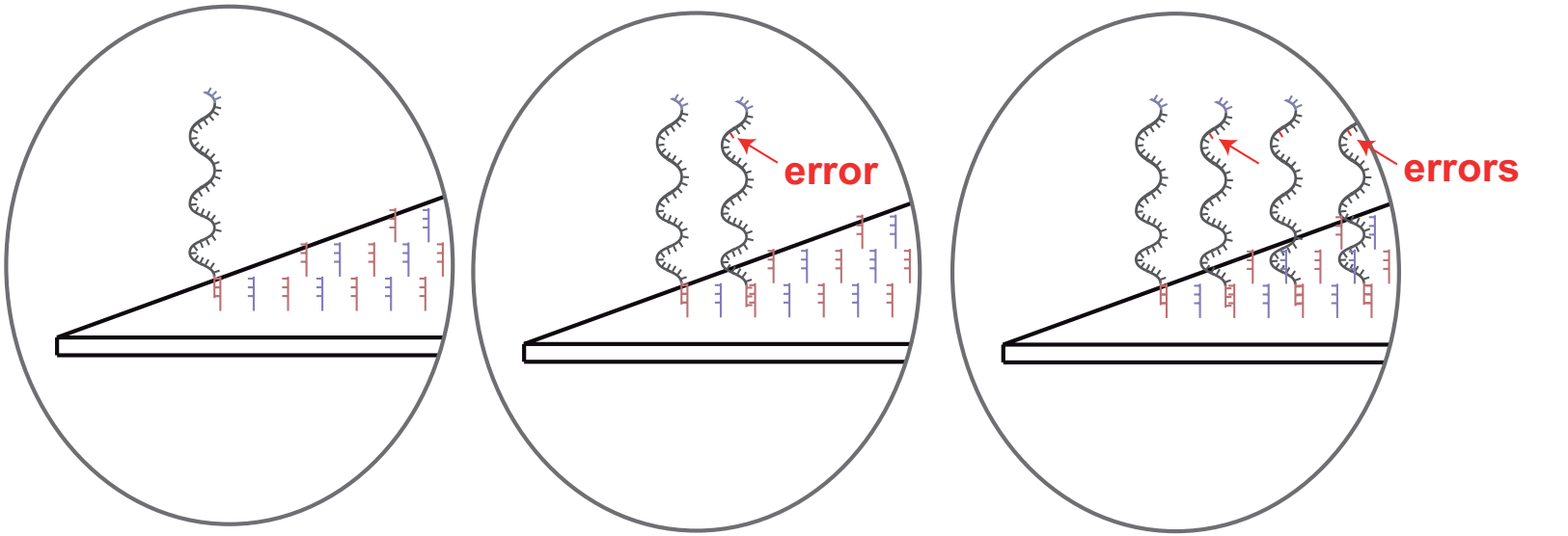
G/C bias





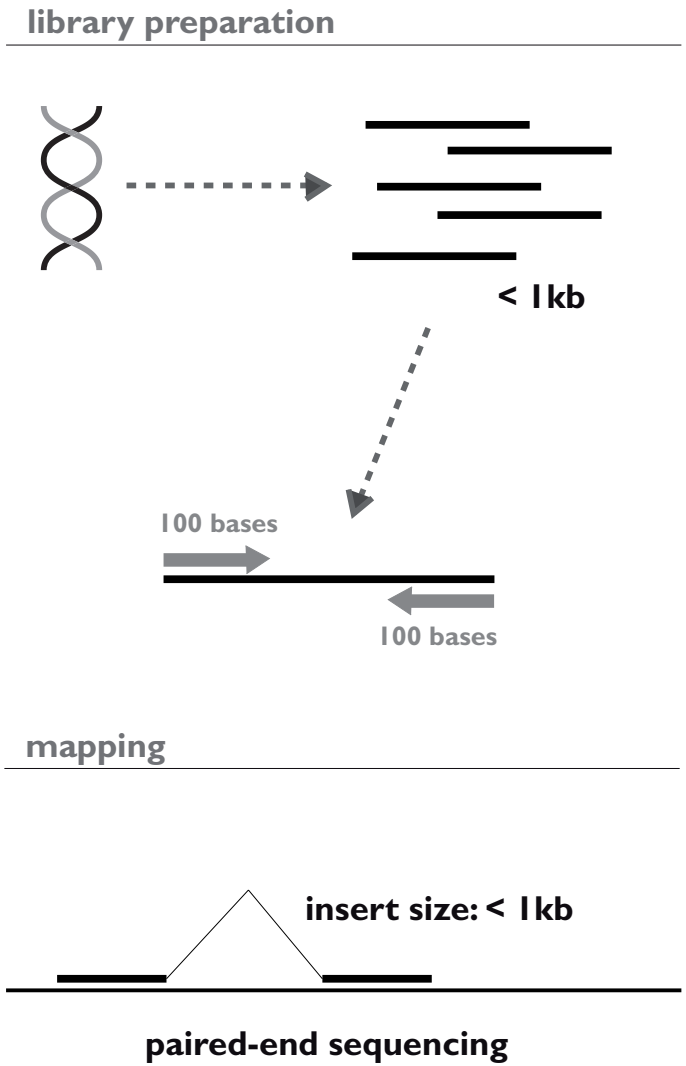
Aird et al., Genome Biology (2011)

Amplification



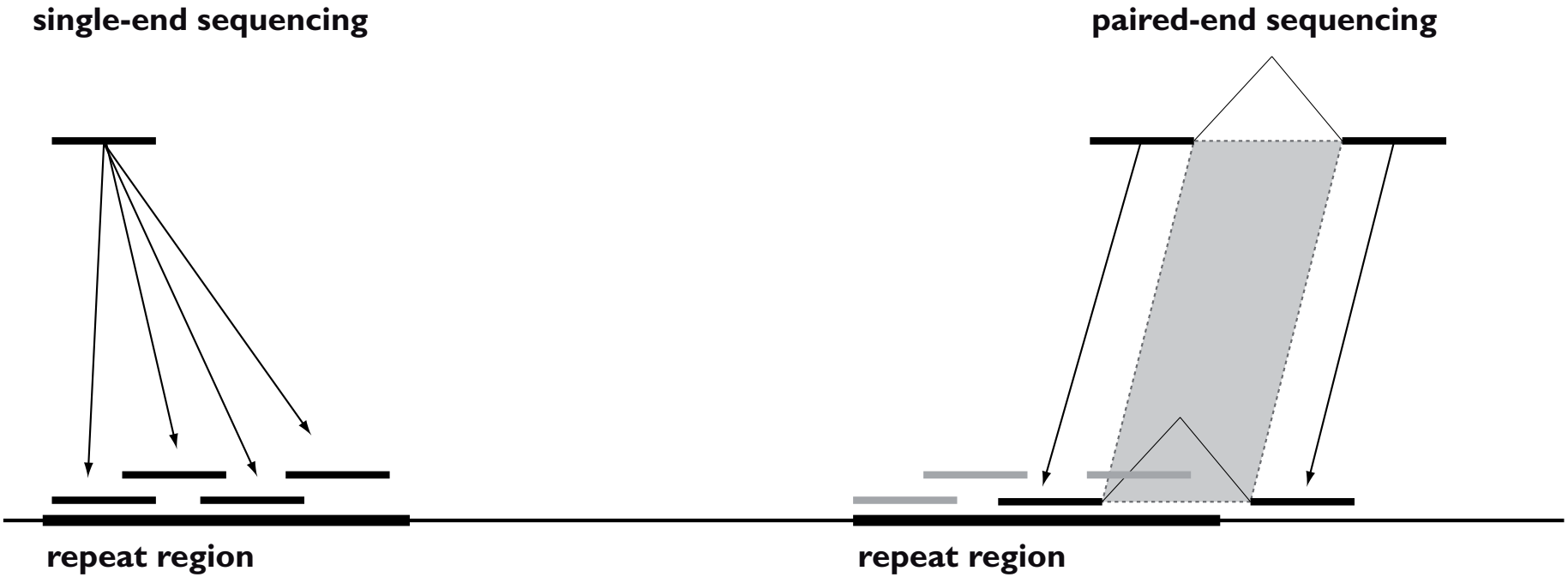
Library Preparation

Paired-End Sequencing



Paired-End Sequencing

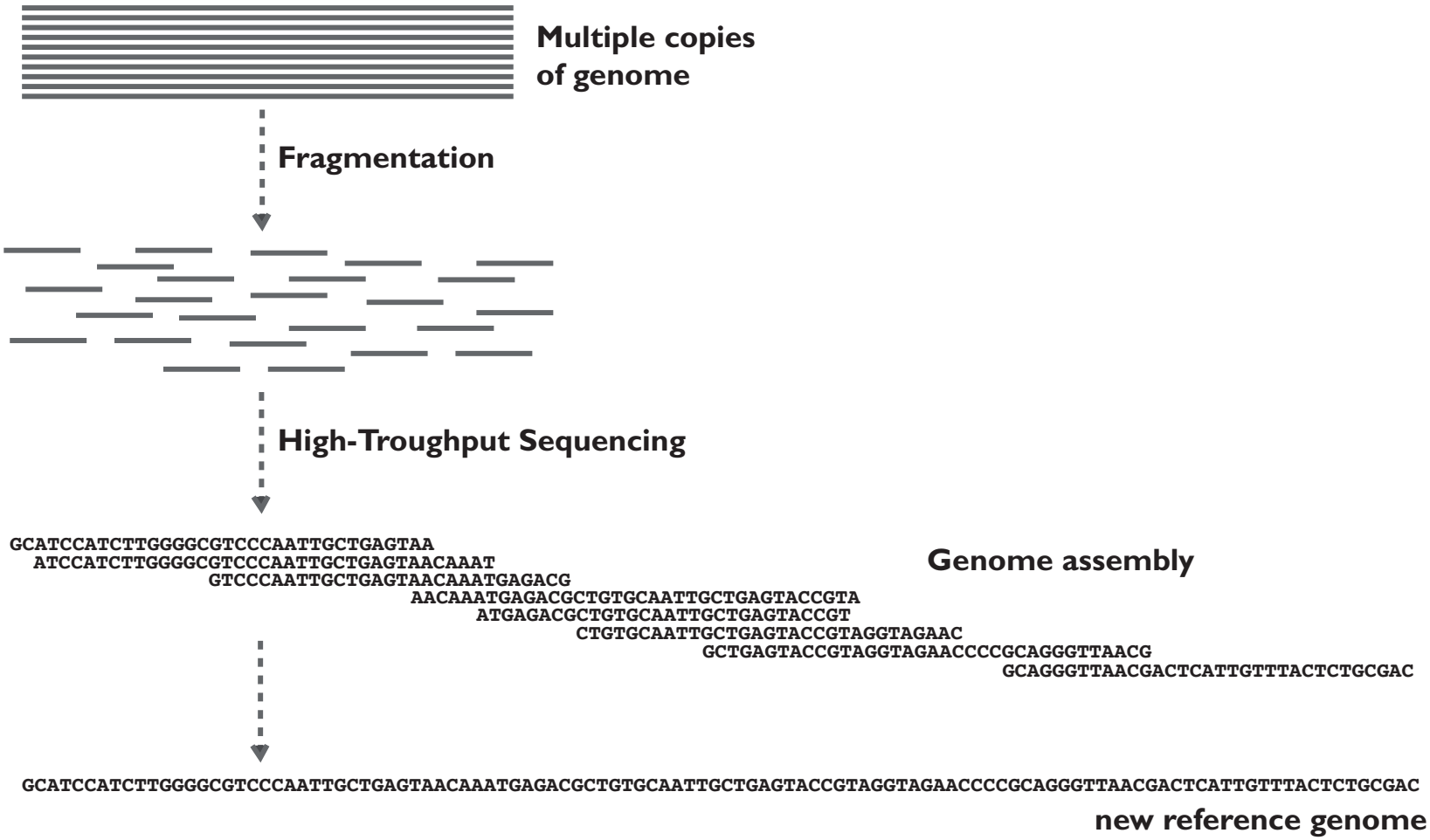
mapping



Introduction to High Throughput Sequencing applications

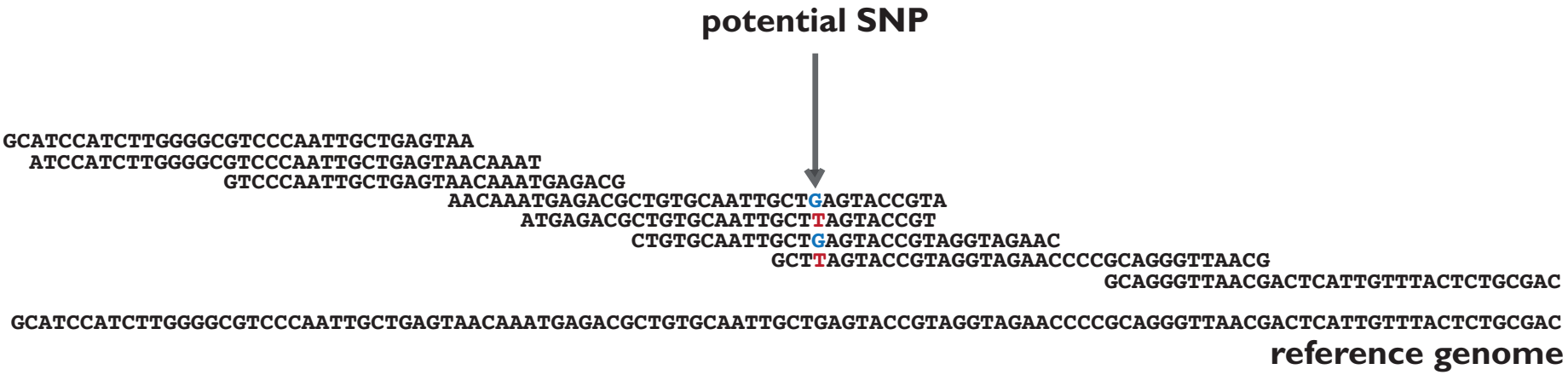
DNA sequencing

De-novo genome assembly

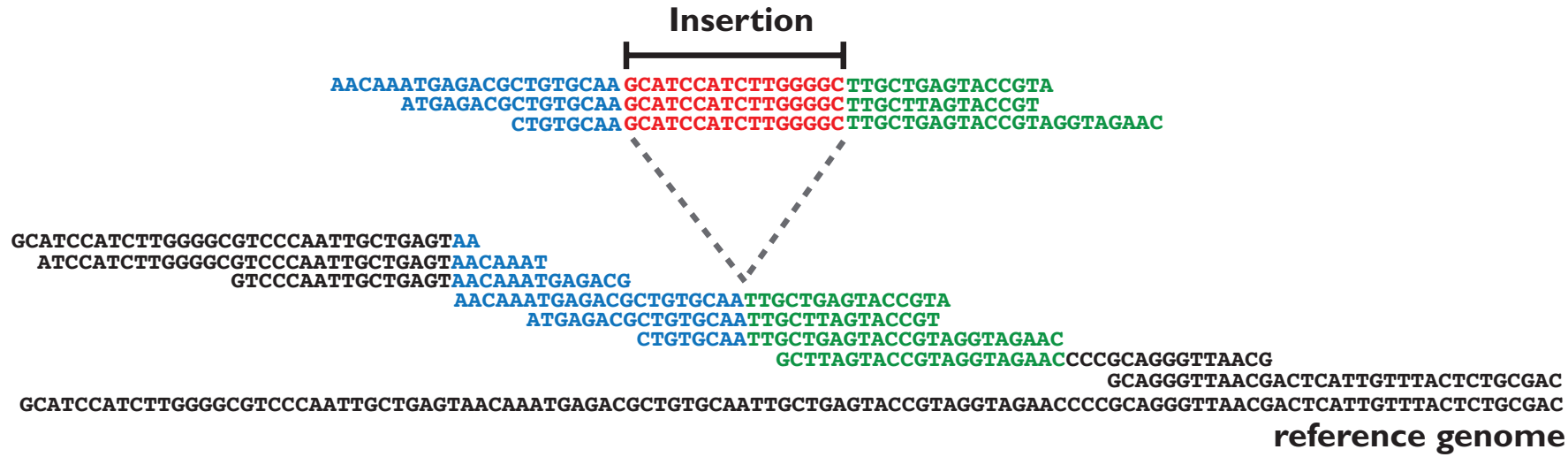


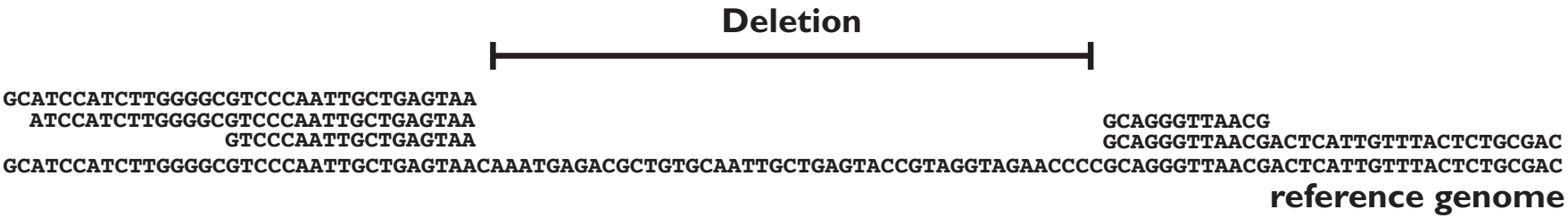
Genome re-sequencing

SNP discovery



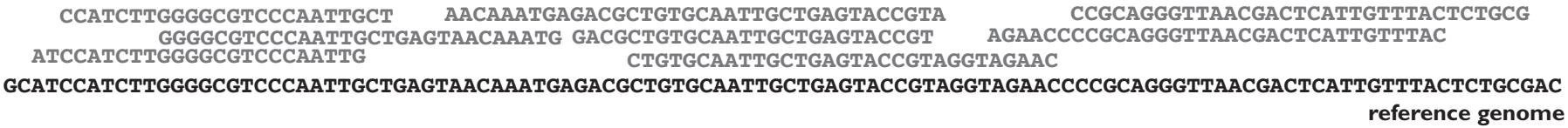
Genome rearrangements



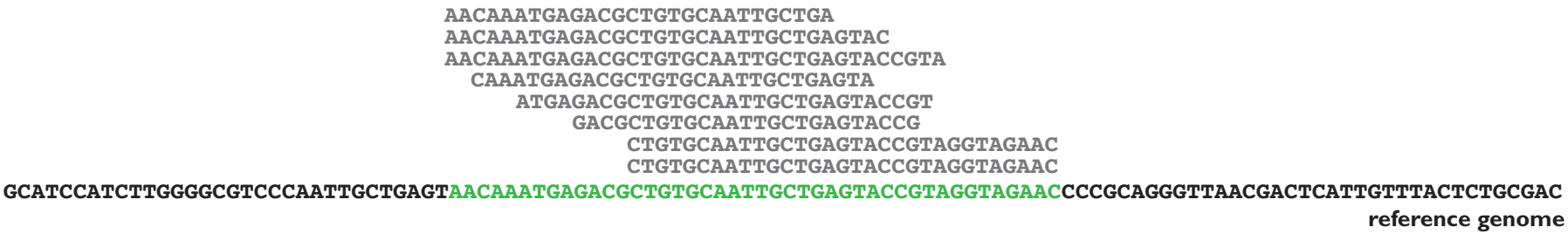


Exome sequencing / Targeted Sequencing

Whole Genome Sequencing

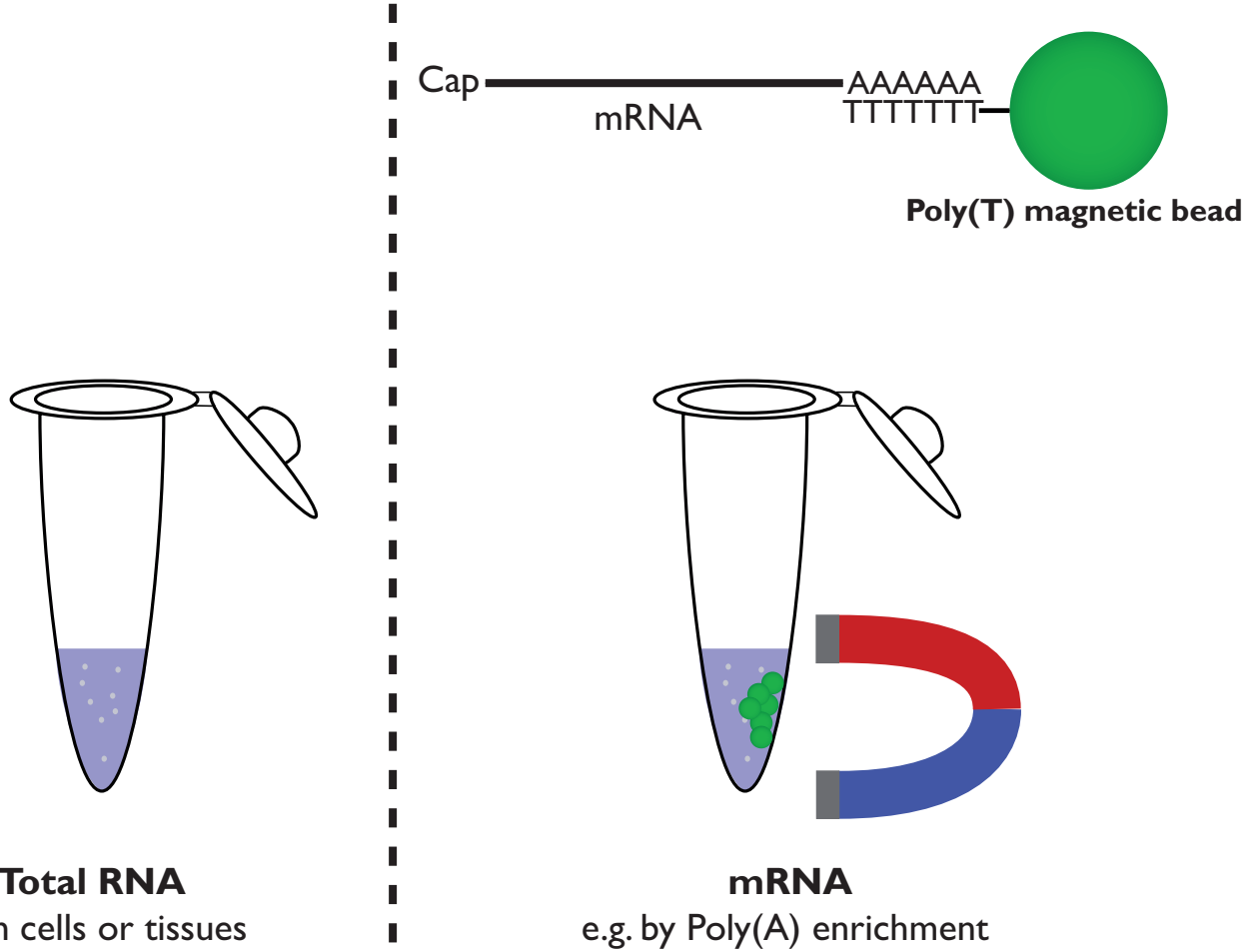


Targeted Sequencing

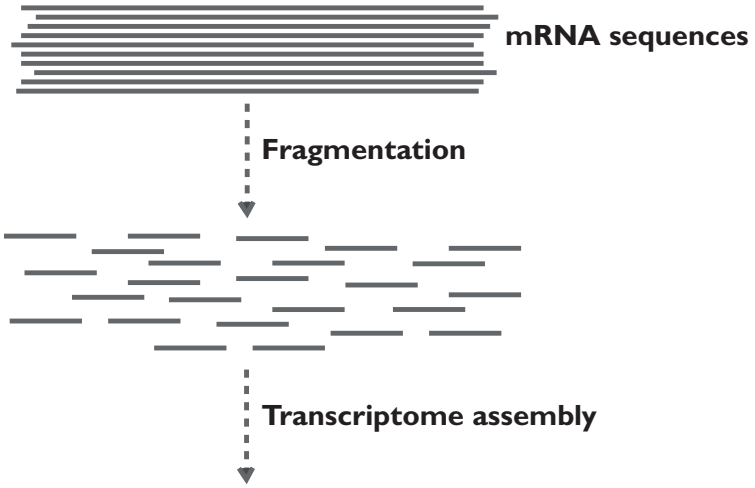


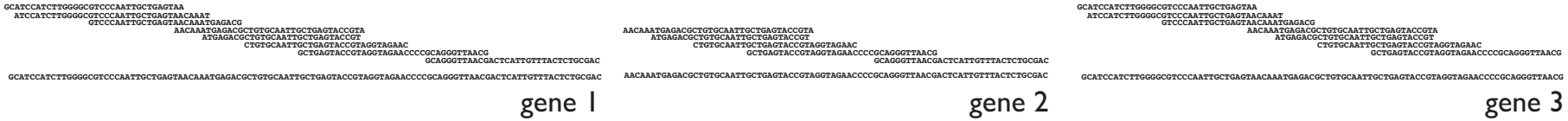
RNA sequencing

RNA-seq library preparation

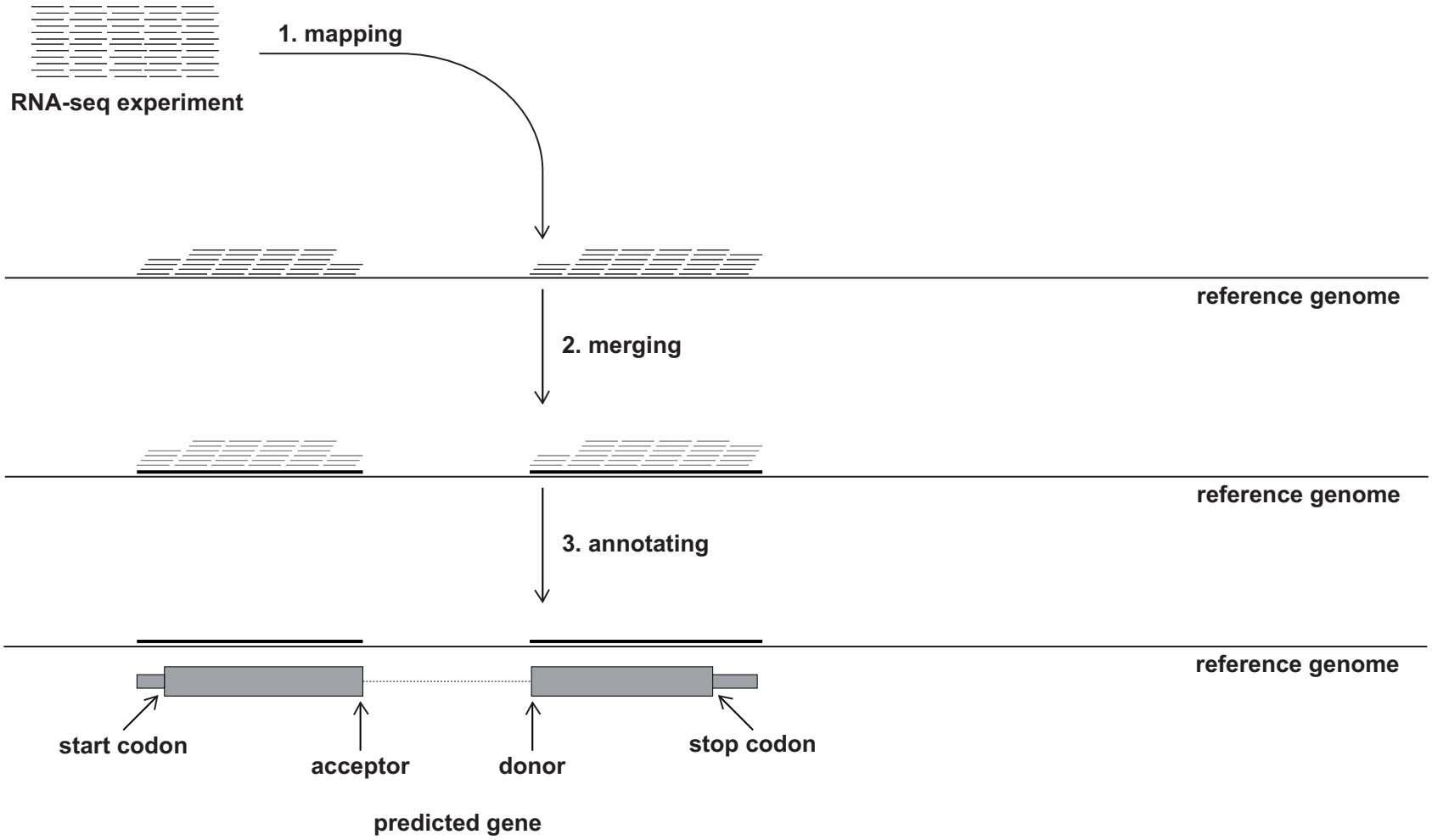


De-novo transcriptome assembly

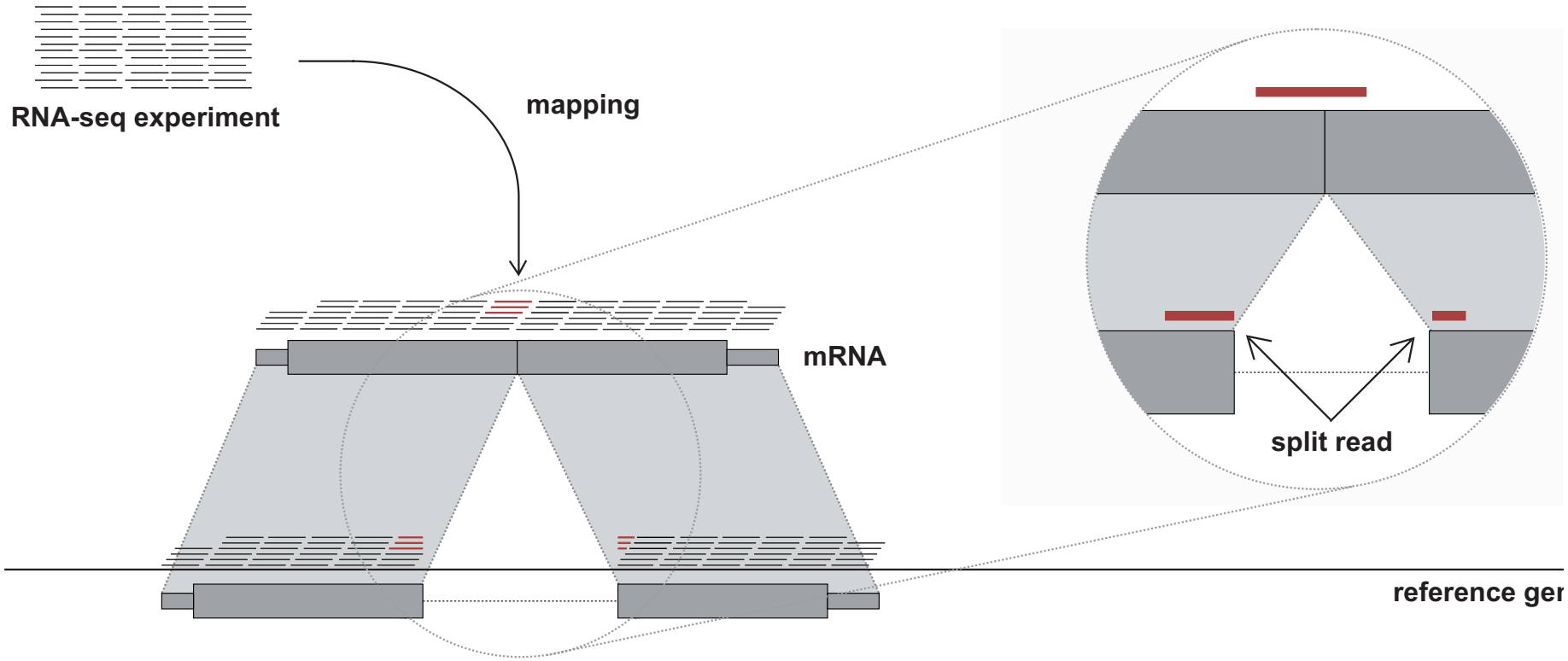




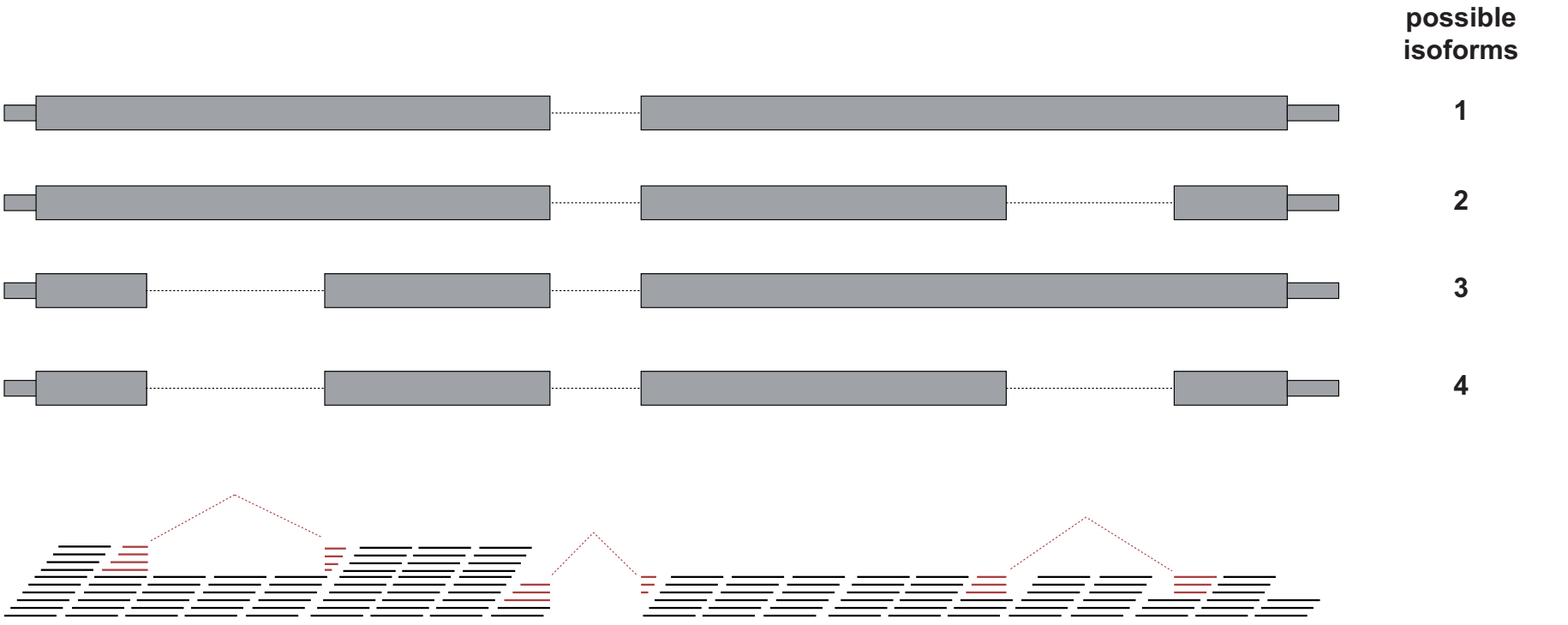
Gene prediction



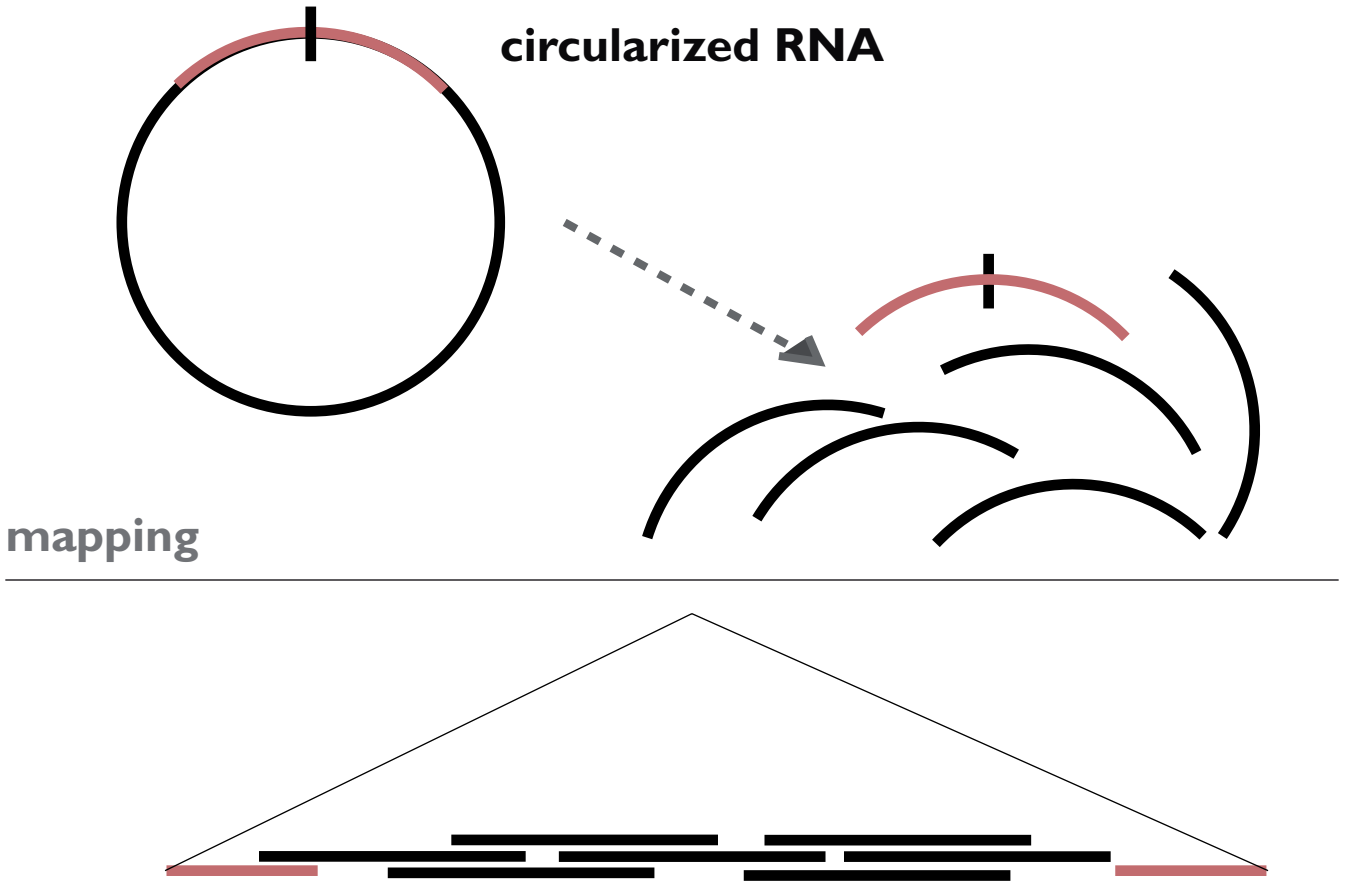
Splice site detection



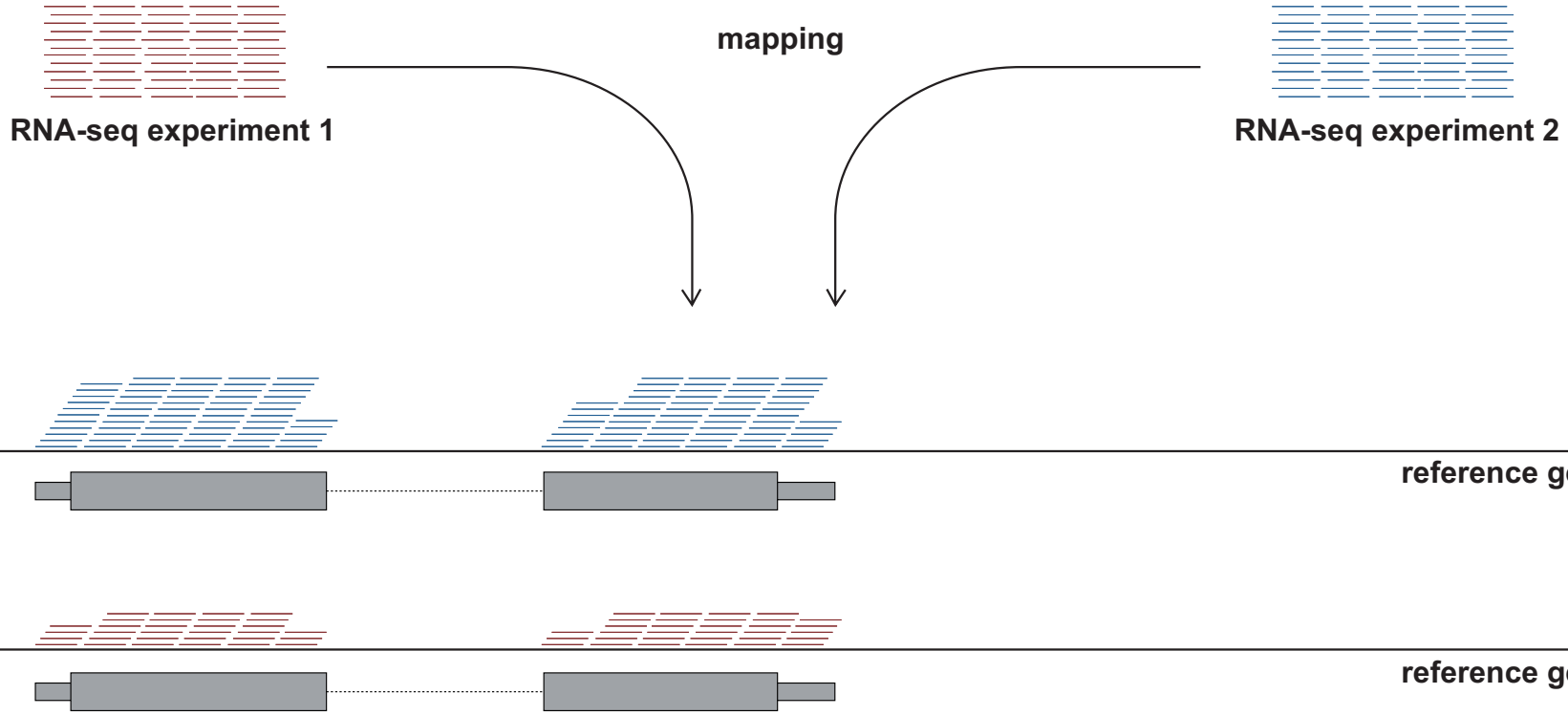
Isoform differentiation



Non-standard isoform prediction



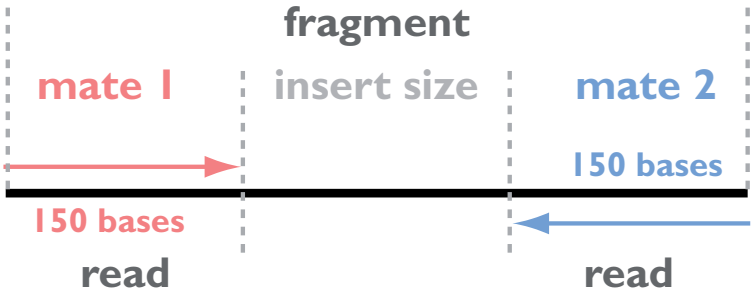
Differential expression



Sequence Reads

Basic Notations

Basic Notations



fragment
molecule to be sequenced

read
One sequenced part of a biological fragment
(mate1 and/or mate2)

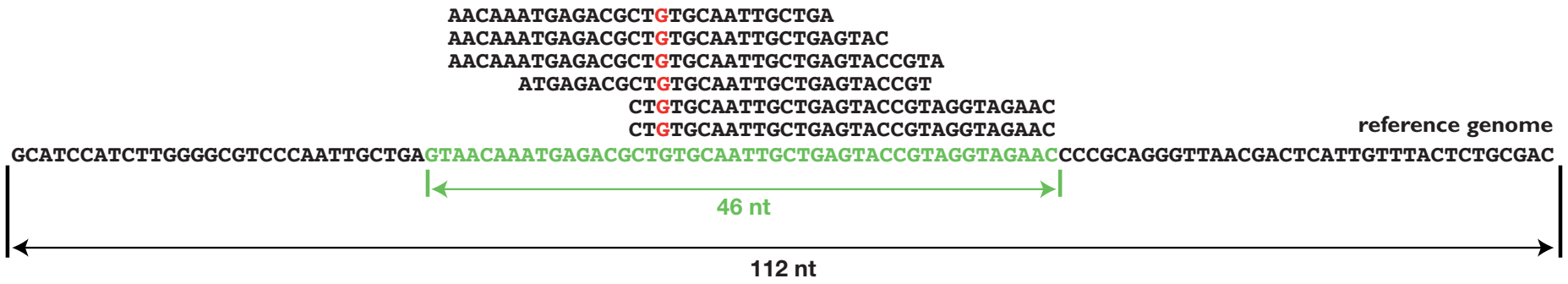
mate 1
sequence of the 5' end of paired-end sequencing

mate 2
sequence of the 3' end of paired-end sequencing

sequencing depth
The total number of all the sequences, reads or bases represented in a single sequencing experiment

coverage
whole genome: (# of sequenced bases) / (size of genome)
one locus: (# of bases mapping to the locus) / (size of locus)
one position: (# of reads overlapping with one position)

Coverage



whole genome: (# of sequenced bases) / (size of genome)

188 / 112 = 1.68 fold

one locus: (# of bases mapping to the locus) / (size of locus)

188 / 46 = 4.09 fold

one position: (# of reads overlapping with one position)

6 fold

Sequence Read Archive (SRA)

Access the SRA

Info

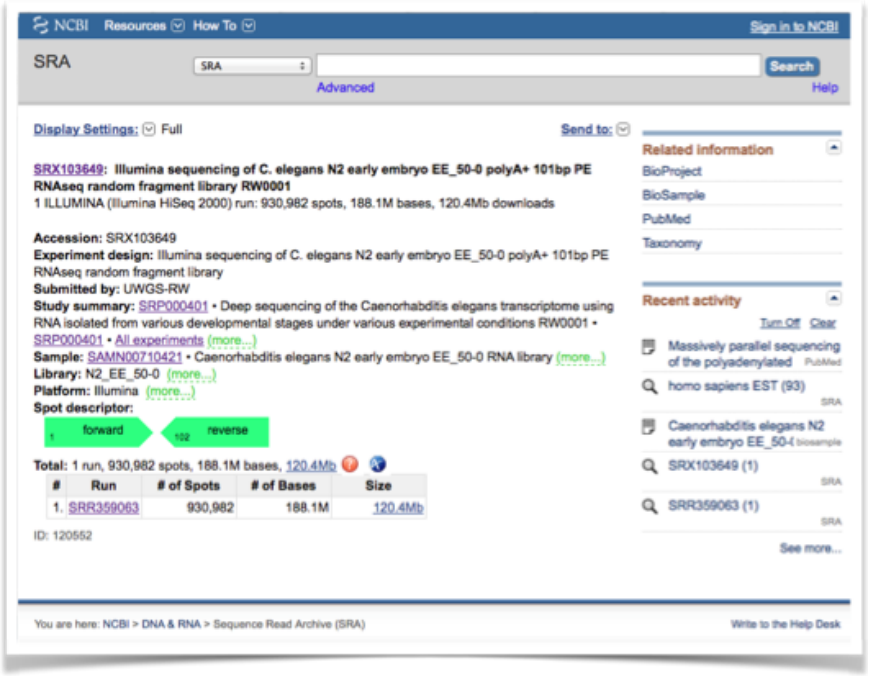
The Sequence Read Archive (SRA) is a bioinformatics database from NCBI (National Center for Biotechnology Information) that provides a public repository for DNA sequencing data, especially the "short reads" generated by High-throughput sequencing.

Tasks

Find the sample with the accession SRX103649 from the Sequence Read Archive (http://www.ncbi.nlm.nih.gov/sra/)

Collect the following information:

- What species?
- Genome or Transcriptome?
- What sequencing machine?
- What read length?
- Single-end or paired-end?



Folder Structure

Task

Create a folder for this course

```
$ mkdir NGS2018
$ cd NGS2018
```

Info

In this course we will create a folder structure that looks like this:

```
- NGS2017
  |- raw
  |- clipped
  |- genome
  |   |- segemehl
  |   |- star
  |   |- bowtie2
  |   |- bwa
  |- mapping
```

Download Samples

Task
Download the SRA file

```
$ mkdir raw
$ cd raw
$ wget ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/
  reads/ByRun/sra/SRR/SRR359/SRR359063/SRR359063.sra
```

Info
wget is a tool to download files from the Web

Preprocess SRA Samples

Task

Convert the SRA format data into FASTQ using the SRA toolkit (http://www.ncbi.nlm.nih.gov/Traces/sra/?view=software)

```
$ fastq-dump --split-files --gzip SRR359063.sra
Written 930982 spots for SRR359063.sra
Written 930982 spots total
$
$ ls -hl
total 329M
-rw-r--r-- 1 david staff  83M Mar 16 11:23 SRR359063_1.fastq.gz
-rw-r--r-- 1 david staff  85M Mar 16 11:23 SRR359063_2.fastq.gz
-rw-r--r-- 1 david staff 121M Mar 16 00:34 SRR359063.sra
```

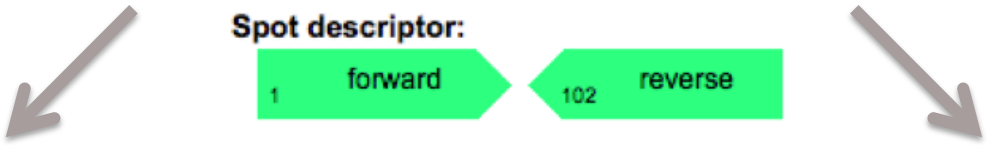
Preprocess SRA Samples

```
$ fastq-dump --split-files --gzip SRR359063.sra
```

Info

--split-files is used because of the special format the paired-end reads are stored in the SRA file

```
@SRR359063.1 D042KACXX:3:1101:2690:2160 length=202
NCATCGTCCGGTATGTAGAACAGGGGAACCGGACGTTTTCCAAGGCGTAGCCATGTTAGACAAGGCGCAGATATA
GGTGATGCTGATGCAGAAAAACGATTGGCGTCTCCATCGAAATCATTGATGCCCGTGGACTATCCGAACATCCAA
AAATCAATCGTTTTTCTGCATCAGCATCACCTATATCTGCGCCTTGTCTAAC
```




```
@SRR359063.1 D042KACXX:
3:1101:2690:2160 length=101
NCATCGTCCGGTATGTAGAACAGGGGAACCGGACG
TTTTCCAAGGCGTAGCCATGTTAGACAAGGCGCAG
ATATAGGTGATGCTGATGCAGAAAAACGATT
```

```
@SRR359063.1 D042KACXX:
3:1101:2690:2160 length=101
GGCGTCTCCATCGAAATCATTGATGCCCGTGGACT
ATCCGAACATCCAAAAATCAATCGTTTTTCTGCAT
CAGCATCACCTATATCTGCGCCTTGTCTAAC
```

Preprocess SRA Samples

```
$ fastq-dump --split-files --gzip SRR359063.sra
```

Info

--gzip is used to minimize the size of the two FASTQ files.

```
-rw-r--r-- 1 david staff  83M Mar 16 11:23 SRR359063_1.fastq.gz
-rw-r--r-- 1 david staff 284M Mar 16 11:31 SRR359063_1.fastq

-rw-r--r-- 1 david staff  86M Mar 16 11:23 SRR359063_2.fastq.gz
-rw-r--r-- 1 david staff 284M Mar 16 11:31 SRR359063_2.fastq
```

FASTQ format

FASTQ format

Task

Take a look at the FASTQ file of mate 1

```
$ zcat SRR359063_1.fastq.gz | head
@SRR359063.1 D042KACXX:3:1101:2690:2160 length=101
NCATCGTCCGGTATGTAGAACAGGGGAACCGGACGTTTTCCAAGGCGTAGCCATGTTAGACAAGGCGCAGATATA
GGTGATGCTGATGCAGAAAAACGATT
+SRR359063.1 D042KACXX:3:1101:2690:2160 length=101
#4=DBDDDHFHFFHIGHIIIJJJJJJJJJJJJBHDAGHJGGGHIJHFFFFDDEDCCDCCCCDDDDDBDBD>CDEE
>C@CDDDDDDCACAACCDCDBDBB<1
@SRR359063.2 D042KACXX:3:1101:5202:2193 length=101
CTCTGGTACAGAACACGTGGATTATAAGAGTTGCCGCTTCGCACAGAAGTCGGAGTTCTCTCACCACTTTTGAGC
TCTTCCTCGGCTTCTTCTTCCTCTTT
```

Info

cat displays the contents of a file

zcat is the same as cat, but for gziped files

'|' forwards the output of the call before (left) to a new tool (right); that's called a 'pipe'

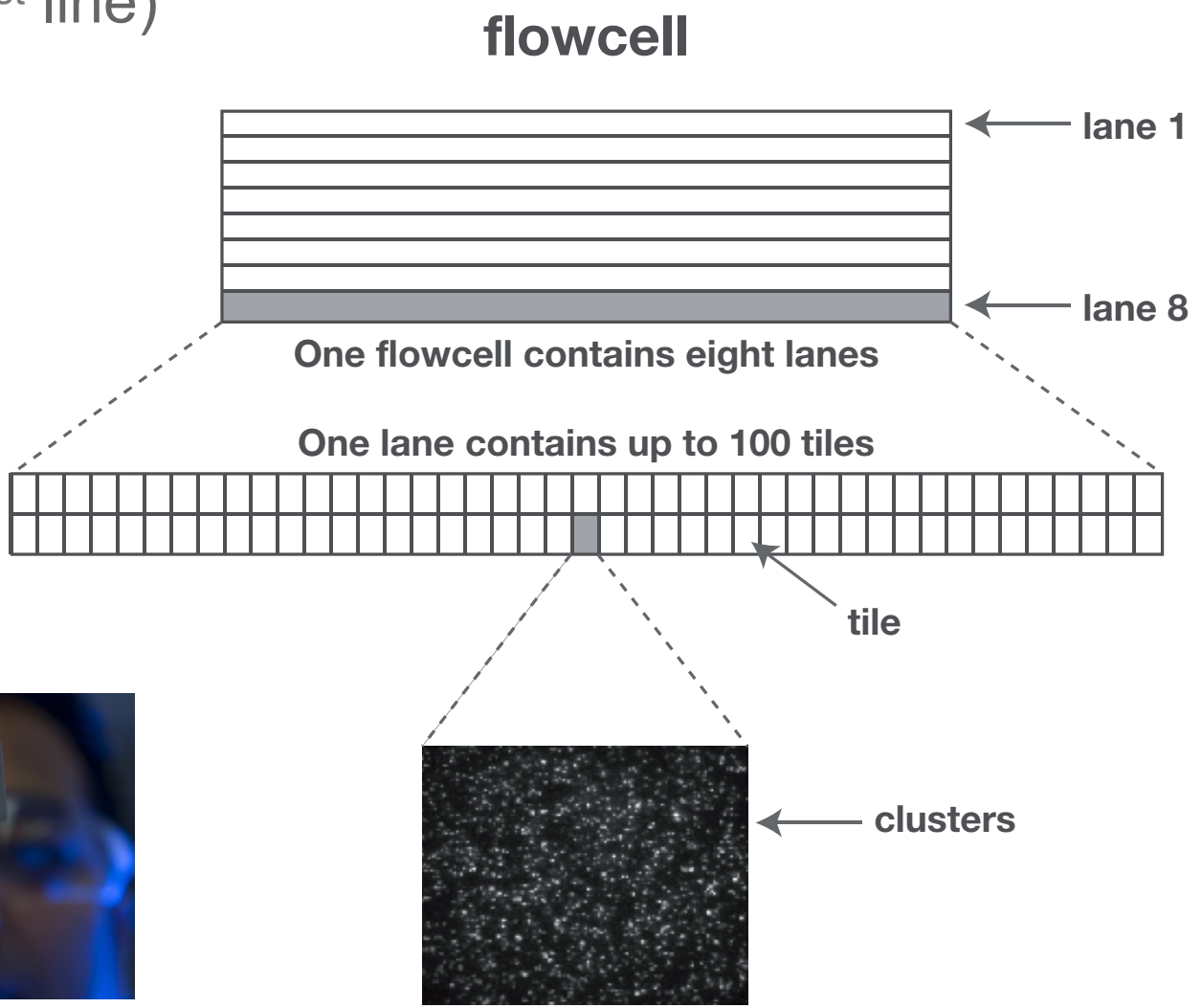
head returns just the first rows and not the complete file

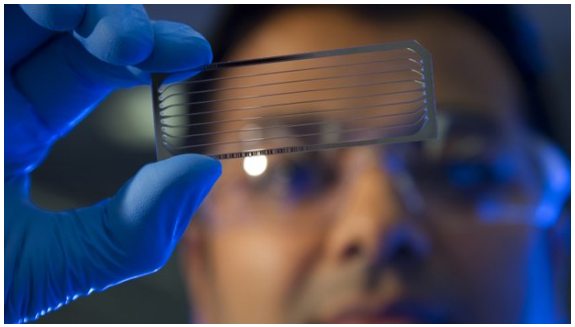
FASTQ format (1st line)

```
$ zcat SRR359063_1.fastq.gz | head
@SRR359063.1 D042KACXX:3:1101:2690:2160 length=101
NCATCGTCCGGTATGTAGAACAGGGGAACCGGACGTTTTCCAAGGCGTAGCCATGTTAGACAAGGCGCAGATATA
GGTGATGCTGATGCAGAAAAACGATT
+SRR359063.1 D042KACXX:3:1101:2690:2160 length=101
#4=DBDDDHFHFFHIGHIIIJJJJJJJJJJJJBHDAGHJGGGHIJHFFFFDDEDCCDCCCCDDDDDBDBD>CDEE
>C@CDDDDDDCACAACCDCDBDBB<1
@SRR359063.2 D042KACXX:3:1101:5202:2193 length=101
CTCTGGTACAGAACACGTGGATTATAAGAGTTGCCGCTTCGCACAGAAGTCGGAGTTCTCTCACCACTTTTGAGC
TCTTCCTCGGCTTCTTCTTCCTCTTT
```

SRR359063.1	run ID
D042KACXX	flowcell ID
3	flowcell lane
1101	tile number within the flowcell lane
2690	'x'-coordinate of the cluster within the tile
2160	'y'-coordinate of the cluster within the tile

FASTQ format (1st line)





Source: utsandiego.com

FASTQ format (2nd line)

```
$ zcat SRR359063_1.fastq.gz | head
@SRR359063.1 D042KACXX:3:1101:2690:2160 length=101
NCATCGTCCGGTATGTAGAACAGGGGAACCGGACGTTTTCCAAGGCGTAGCCATGTTAGACAAGGCGCAGATATA
GGTGATGCTGATGCAGAAAAACGATT
+SRR359063.1 D042KACXX:3:1101:2690:2160 length=101
#4=DBDDDHFHFFHIGHIIIJJJJJJJJJJJJBHDAGHJGGGHIJHFFFFDDEDCCDCCCCDDDDDBDBD>CDEE
>C@CDDDDDDCACAACCDCDBDBB<1
@SRR359063.2 D042KACXX:3:1101:5202:2193 length=101
CTCTGGTACAGAACACGTGGATTATAAGAGTTGCCGCTTCGCACAGAAGTCGGAGTTCTCTCACCACTTTTGAGC
TCTTCCTCGGCTTCTTCTTCCTCTTT
```

Info
The raw sequence letters.

FASTQ format (3rd line)

```
$ zcat SRR359063_1.fastq.gz | head
@SRR359063.1 D042KACXX:3:1101:2690:2160 length=101
NCATCGTCCGGTATGTAGAACAGGGGAACCGGACGTTTTCCAAGGCGTAGCCATGTTAGACAAGGCGCAGATATA
GGTGATGCTGATGCAGAAAAACGATT
+SRR359063.1 D042KACXX:3:1101:2690:2160 length=101
#4=DBDDDHFHFFHIGHIIIJJJJJJJJJJJJBHDAGHJGGGHIJHFFFFDDEDCCDCCCCDDDDDBDBD>CDEE
>C@CDDDDDDCACAACCDCDBDBB<1
@SRR359063.2 D042KACXX:3:1101:5202:2193 length=101
CTCTGGTACAGAACACGTGGATTATAAGAGTTGCCGCTTCGCACAGAAGTCGGAGTTCTCTCACCACTTTTGAGC
TCTTCCTCGGCTTCTTCTTCCTCTTT
```

Info

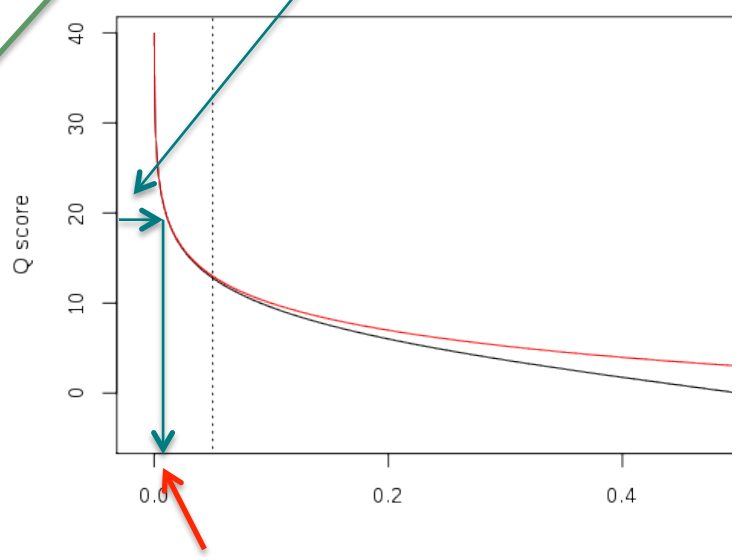
Begins with a '+' character and is optionally followed by the same sequence identifier (and any description) again.

FASTQ format - Quality Values

```
@4=DBDDDHFHFFHIGHIIIJJJJJJJJJJJJJJBHDAGHJGGGHIJHFFFFDDEDCCDCCCCDDDDDBDBD>CDEE
>C@CDDDDDDCACAACCDCDBDBB<1
```

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	De
0	0	000	NUL (null)	32	20	040		Space	6
1	1	001	SOH (start of heading)	33	21	041	!	!	6
2	2	002	STX (start of text)	34	22	042	"	"	6
3	3	003	ETX (end of text)	35	23	043	#	#	6
4	4	004	EOT (end of transmission)	36	24	044	$	$	6
5	5	005	ENQ (enquiry)	37	25	045	%	%	6
6	6	006	ACK (acknowledge)	38	26	046	&	&	7
7	7	007	BEL (bell)	39	27	047	'	'	7
8	8	010	BS (backspace)	40	28	050	(	(	7
9	9	011	TAB (horizontal tab)	41	29	051	)	)	7
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	7
11	B	013	VT (vertical tab)	43	2B	053	+	+	7
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	7
13	D	015	CR (carriage return)	45	2D	055	-	-	7
14	E	016	SO (shift out)	46	2E	056	.	.	7
15	F	017	SI (shift in)	47	2F	057	/	/	7
16	10	020	DLE (data link escape)	48	30	060	0	0	8
17	11	021	DC1 (device control 1)	49	31	061	1	1	8
18	12	022	DC2 (device control 2)	50	32	062	2	2	8
19	13	023	DC3 (device control 3)	51	33	063	3	3	8
20	14	024	DC4 (device control 4)	52	34	064	4	4	8
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	8
22	16	026	SYN (synchronous idle)	54	36	066	6	6	8
23	17	027	ETB (end of trans. block)	55	37	067	7	7	8
24	18	030	CAN (cancel)	56	38	070	8	8	8

52 – 33 = 19 = 'Phred Score'



the probability that the corresponding base call is incorrect

Quality Control

FastQC

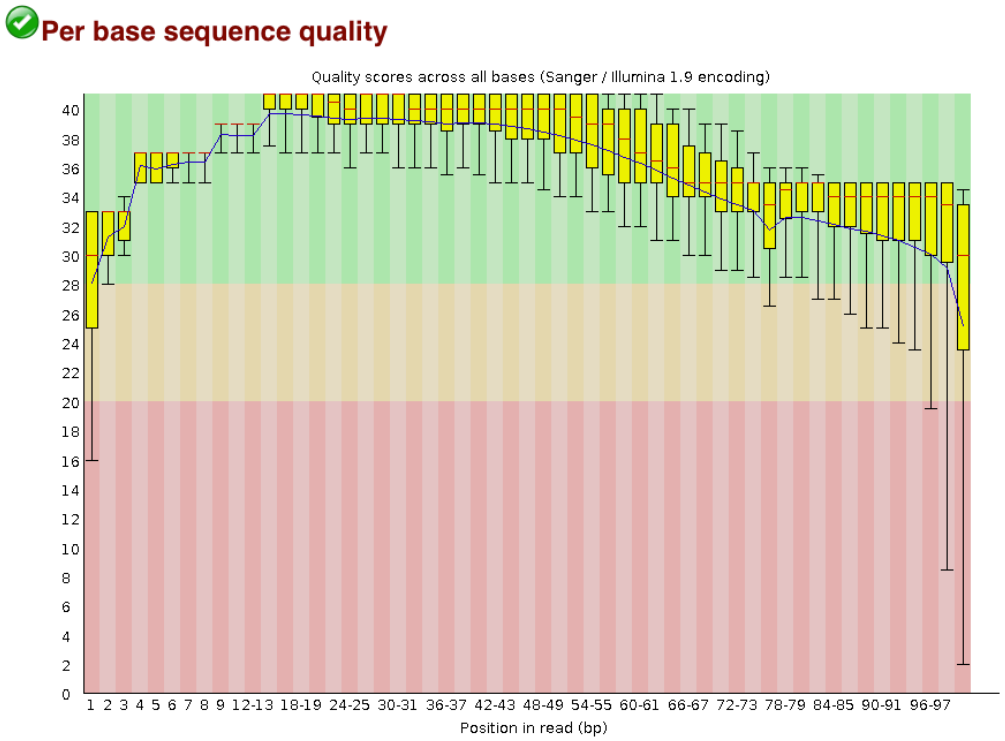
Task
Run the FastQC tool for mate 1.
(http://www.bioinformatics.babraham.ac.uk/projects/fastqc/)

```
$ fastqc SRR359063_1.fastq.gz
```

Task
Have a look at the result page.

```
$ firefox SRR359063_1_fastqc.html &
```

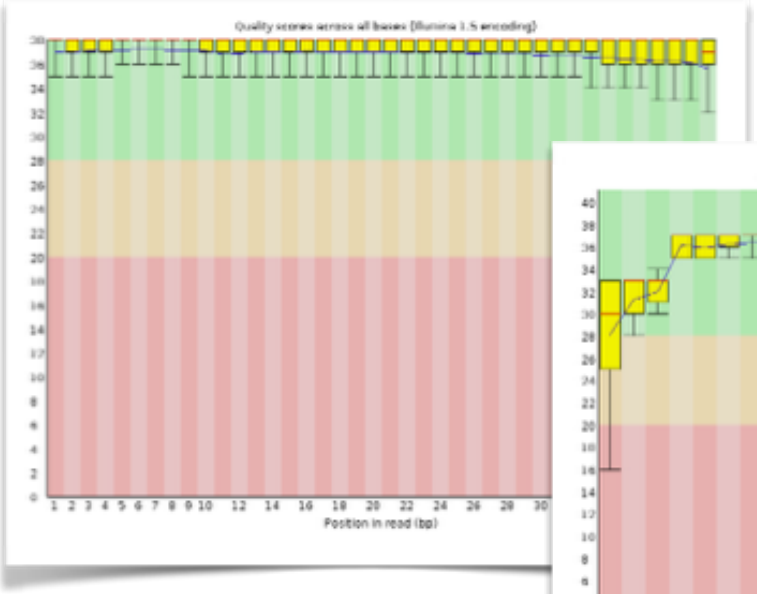
Per base sequence quality



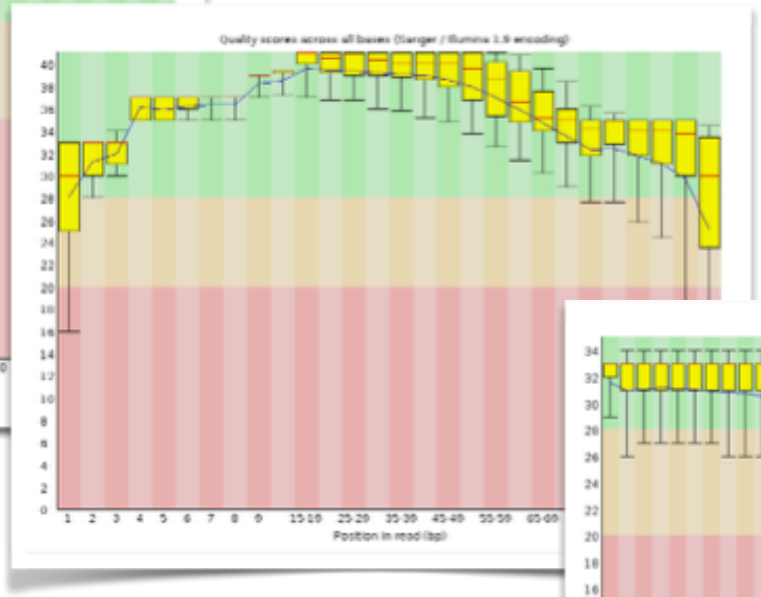
Info

- The central red line is the median value
- The yellow box represents the inter-quartile range (25-75%)
- The upper and lower whiskers represent the 10% and 90% points
- The blue line represents the mean quality

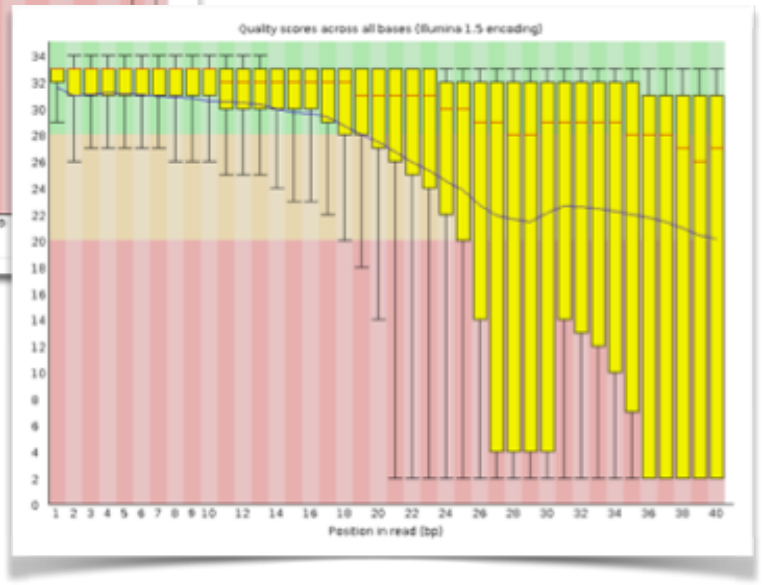
Per base sequence quality



Very good dataset



Our dataset



Poor dataset

Per base sequence quality

Important:

The quality value only refers to the sequencing itself !!!

It does **NOT** tell you anything about problems happened before that step:

- Library preparation
- Contamination in the sample
- PCR clones

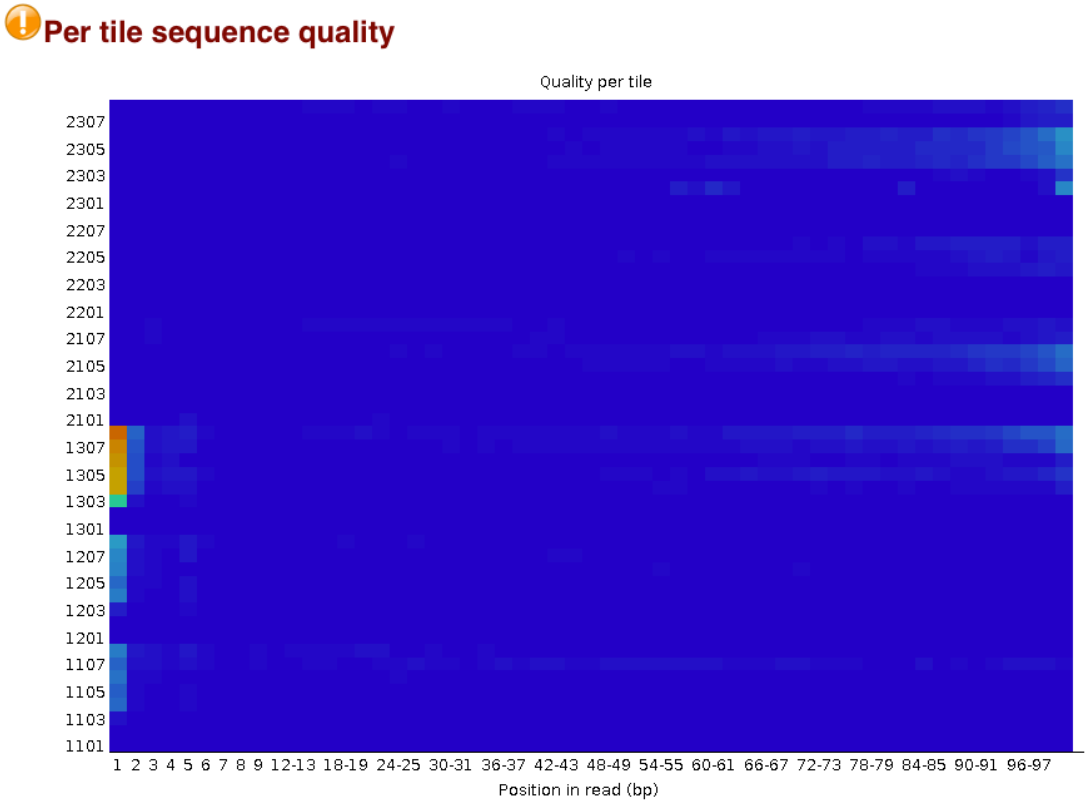
nor afterwards:

- Alignment errors
- Reference errors
- SNP calling biases

There are a lot of error sources that do not change the quality values.

=> Having a 'good' read with high quality values does not automatically imply it is correct!

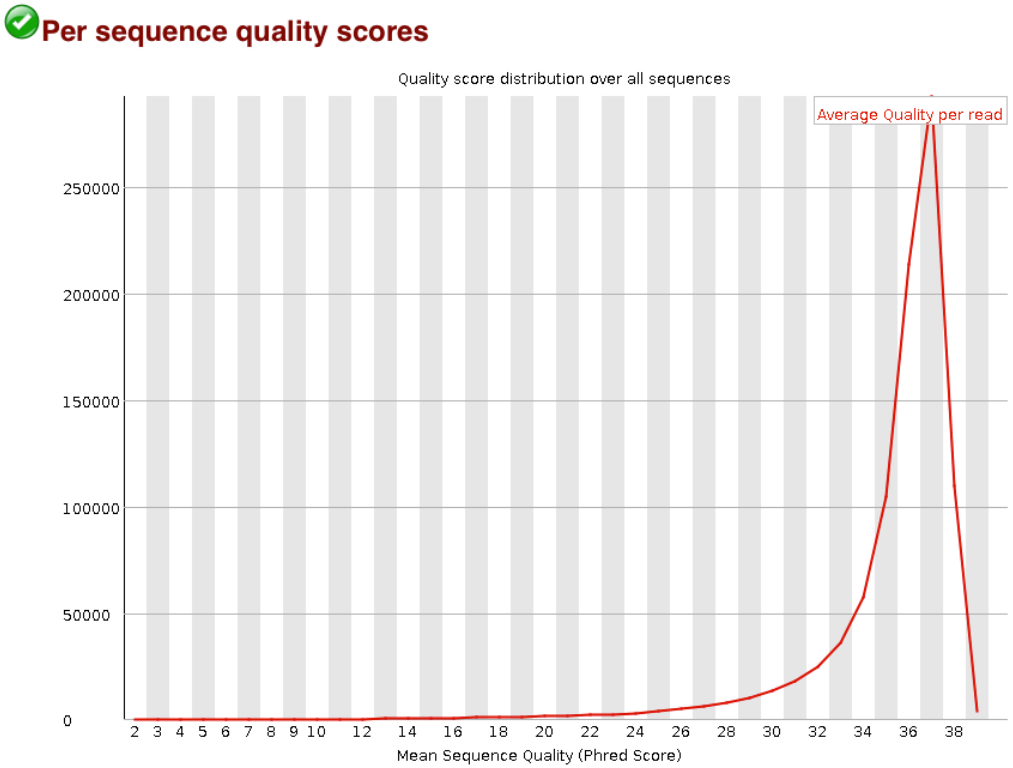
Per tile sequence quality



Info

Reasons for seeing warnings or errors on this plot could be transient problems such as bubbles going through the flowcell, or they could be more permanent problems such as smudges on the flowcell or debris inside the flowcell lane.

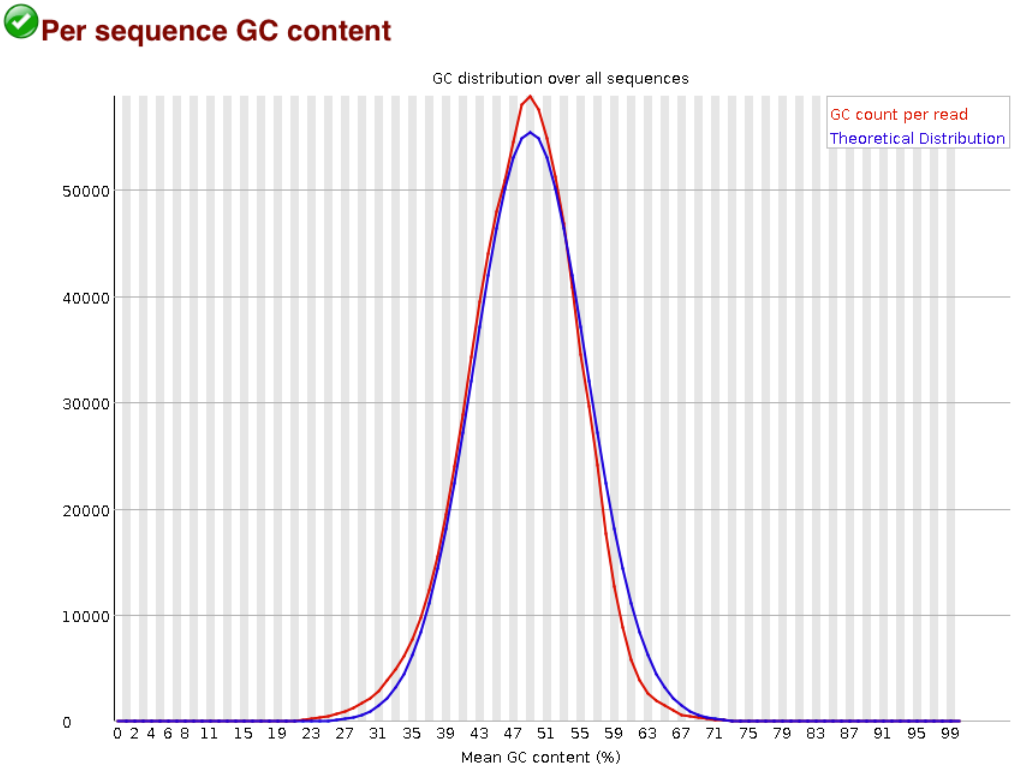
Per sequence quality scores



Info

If a significant proportion of the sequences in a run have overall low quality then this could indicate some kind of systematic problem.

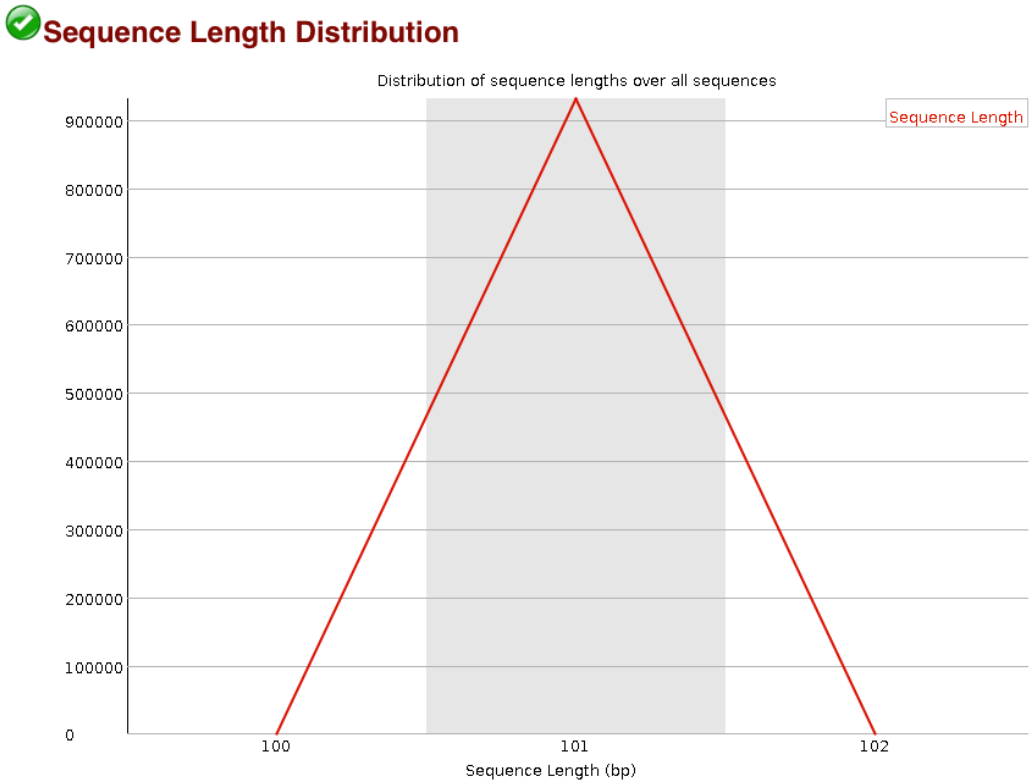
Per sequence GC content



Info

In a normal random library you would expect to see a roughly normal distribution of GC content where the central peak corresponds to the overall GC content of the underlying genome.

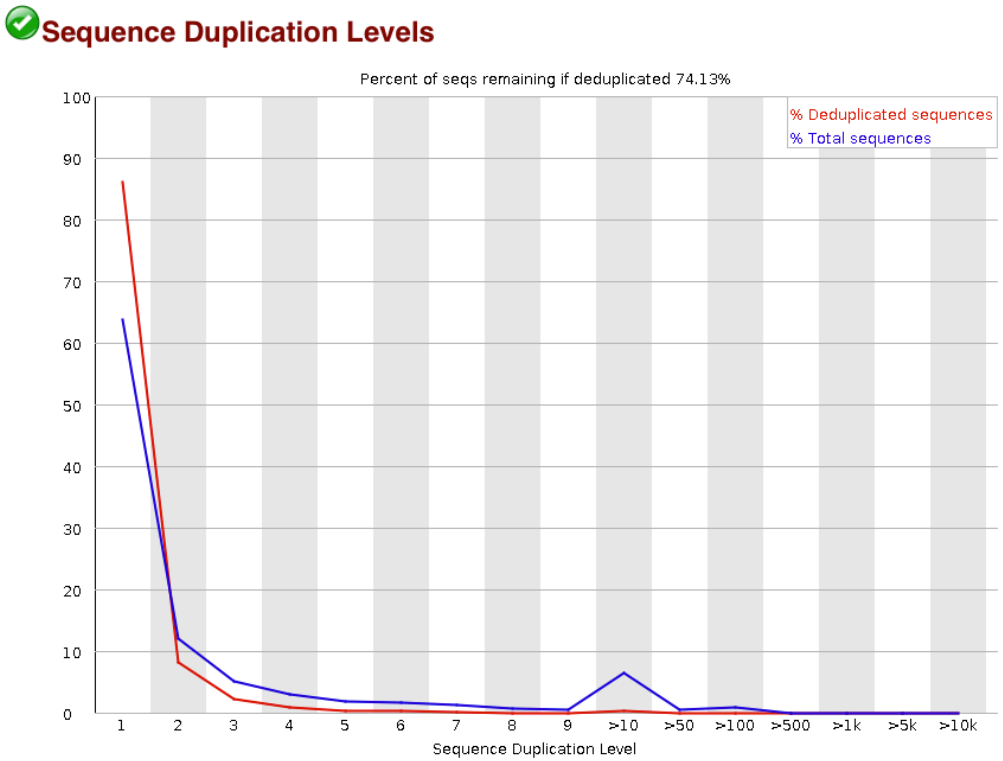
Sequence Length Distribution



Info

Some high throughput sequencers generate sequence fragments of uniform length, but others can contain reads of wildly varying lengths. In Illumina experiments, all raw reads have the same length!

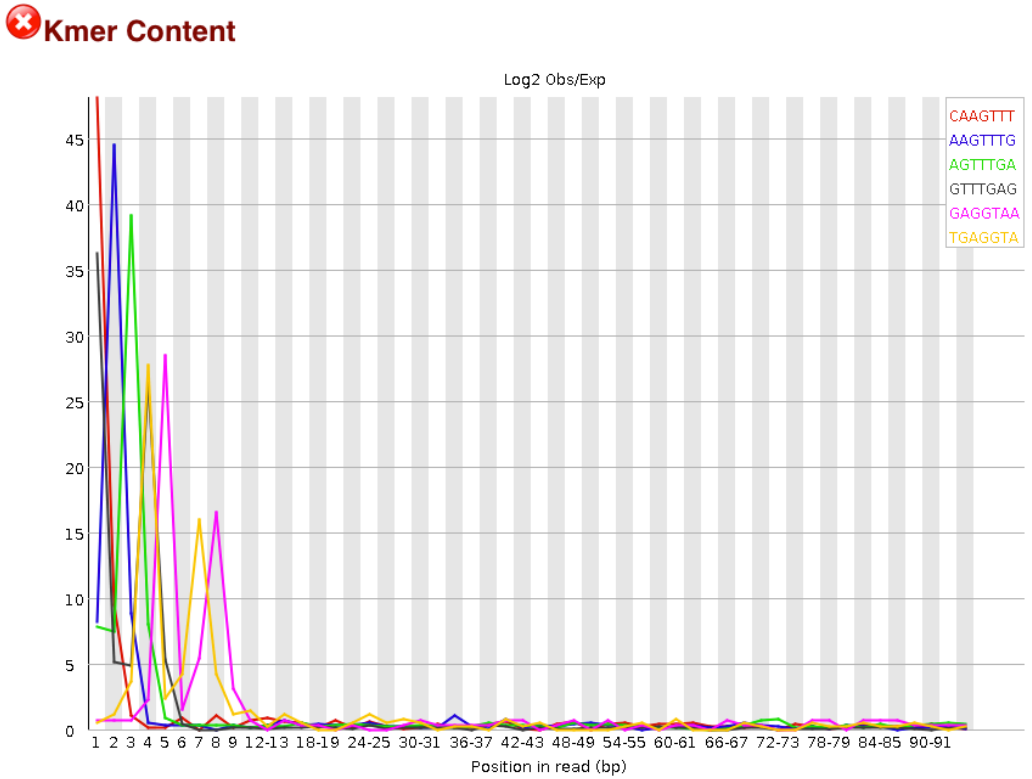
Sequence Duplication Level



Info

In a diverse library most sequences will occur only once in the final set. A low level of duplication may indicate a very high level of coverage of the target sequence, but a high level of duplication is more likely to indicate some kind of enrichment bias (e.g. PCR over amplification).

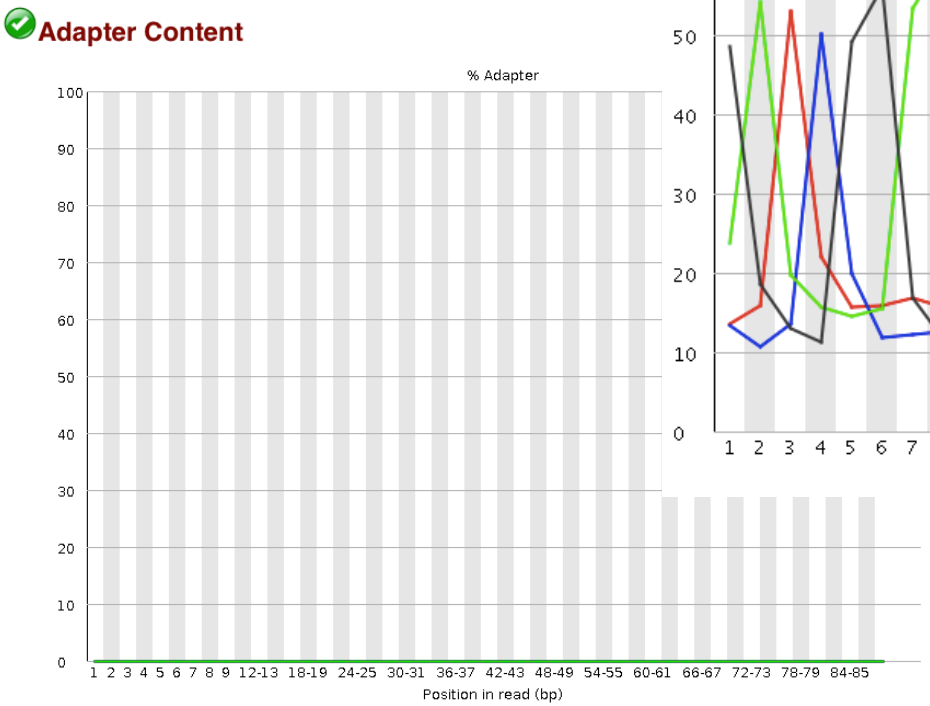
Kmer Content

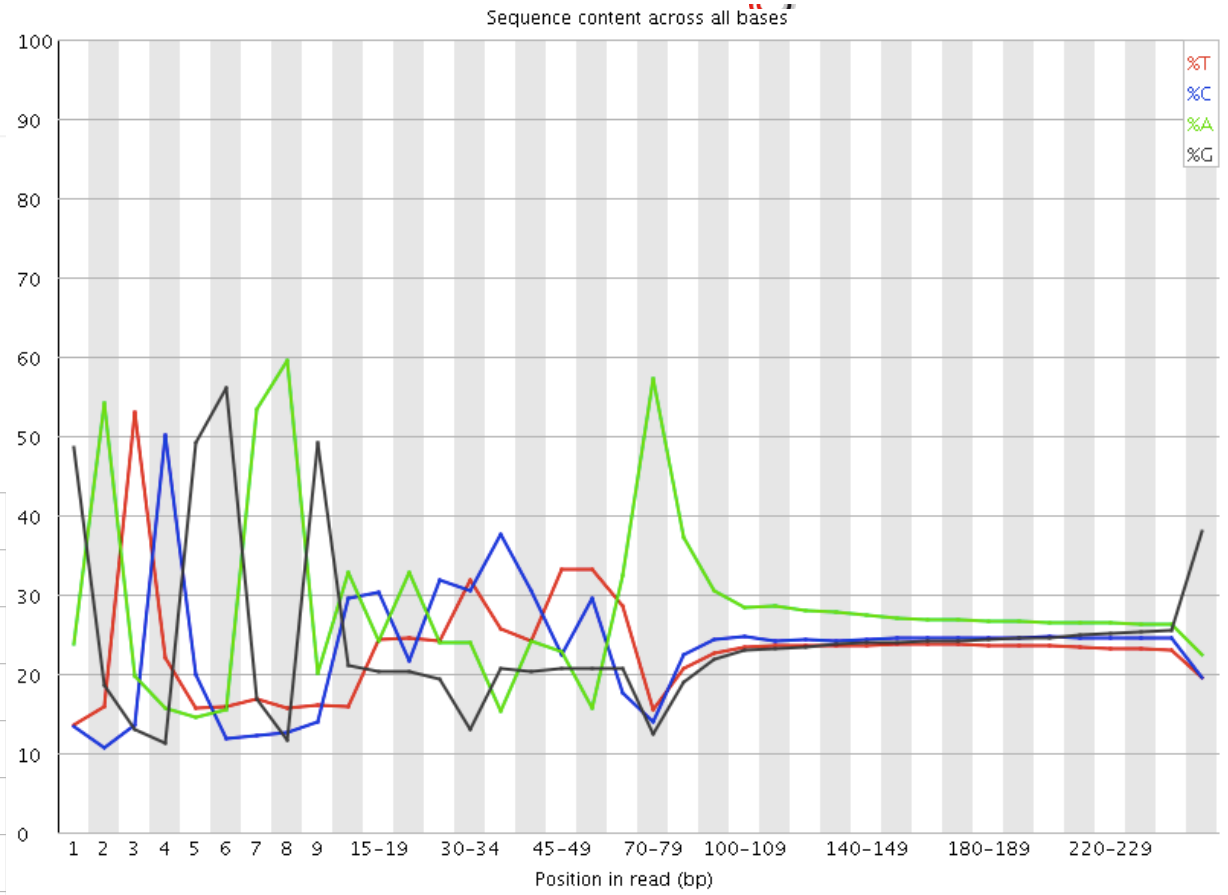


Info

The analysis of overrepresented sequences will spot an increase in any exactly duplicated sequences.

Adapter Content

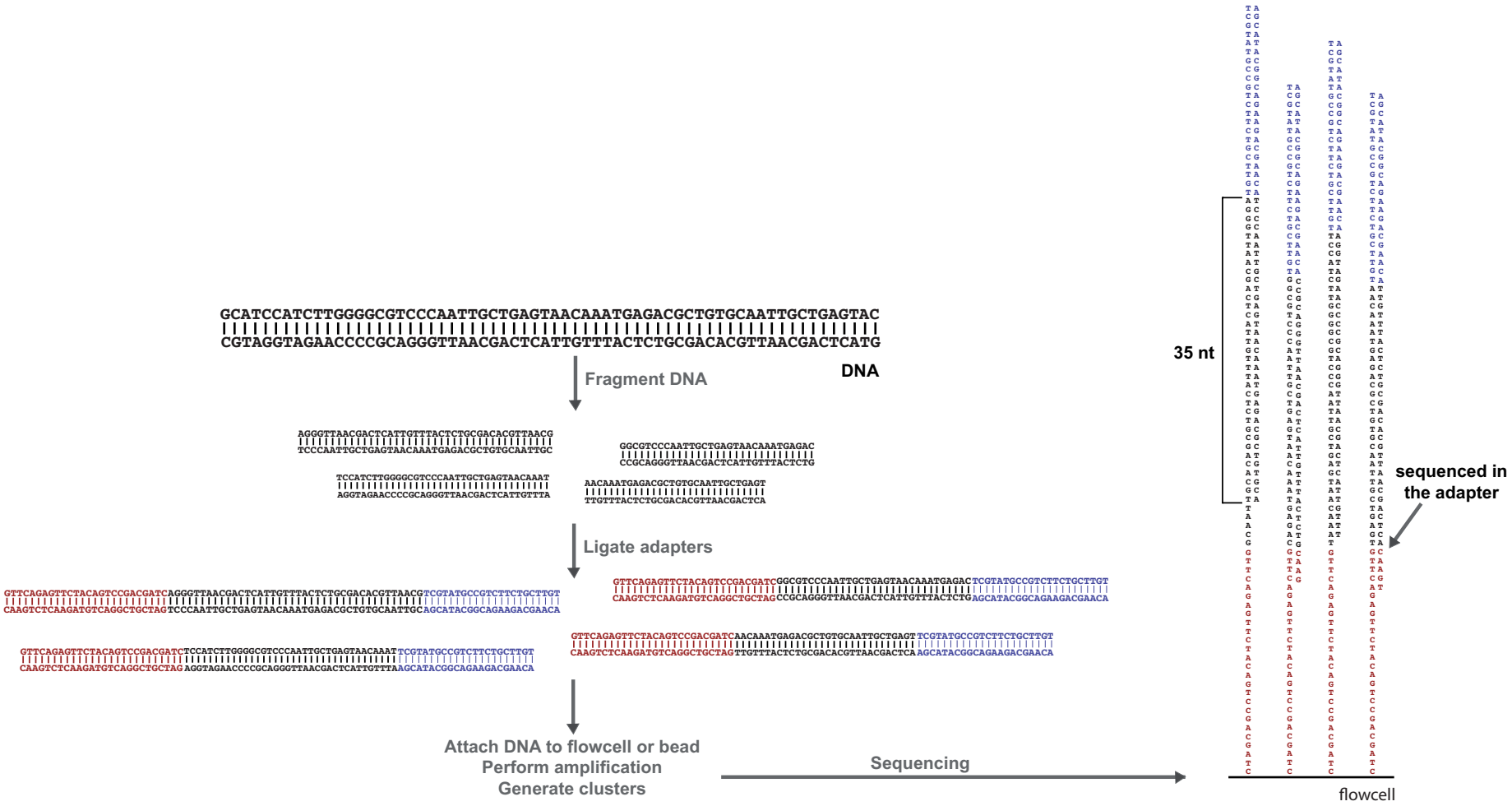




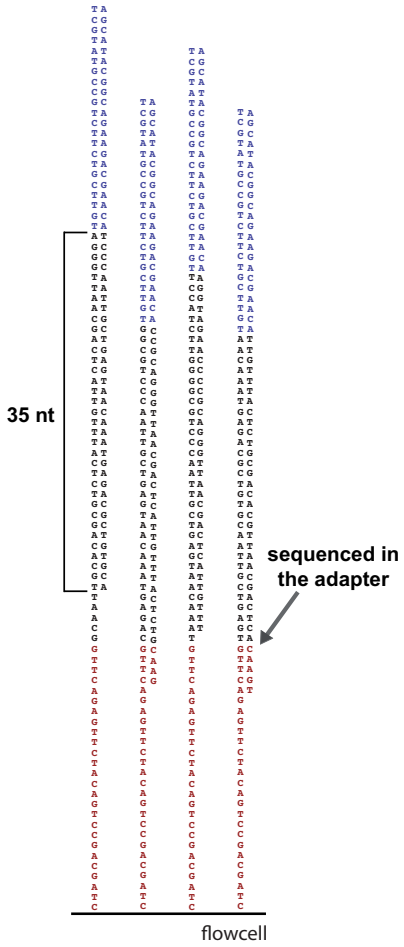
Info

If it was frequently sequenced into the adapter, the adapter sequence would pop up here. It is sequenced into the adapter, if the length of the sequenced molecule is smaller than the number of cycles the Illumina machine performed.

Background


GCATCCATCTTGGGGCGTCCCAATTGCTGAGTAACAAATGAGACGCTGTGCAATTGCTGAGTAC
CGTAGGTAGAACCCCGCAGGGTTAACGACTCATTGTTTACTCTGCGACACGTTAACGACTCATG
DNA
Fragment DNA
AGGGTTAACGACTCATTGTTTACTCTGCGACACGTTAACG
TCCCAATTGCTGAGTAACAAATGAGACGCTGTGCAATTGC
GGCGTCCCAATTGCTGAGTAACAAATGAGAC
CCGCAGGGTTAACGACTCATTGTTTACTCTG
TCCATCTTGGGGCGTCCCAATTGCTGAGTAACAAAT
AGGTAGAACCCCGCAGGGTTAACGACTCATTGTTTA
AACAAATGAGACGCTGTGCAATTGCTGAGT
TTGTTTACTCTGCGACACGTTAACGACTCA
Ligate adapters
GTTCAGAGTTCTACAGTCCGACGATCAGGGTTAACGACTCATTGTTTACTCTGCGACACGTTAACGTCGTATGCCGTCTTCTGCTTGT
CAAGTCTCAAGATGTCAGGCTGCTAGTCCCAATTGCTGAGTAACAAATGAGACGCTGTGCAATTGCAGCATACGGCAGAAGACGAACA
GTTCAGAGTTCTACAGTCCGACGATCGGCGTCCCAATTGCTGAGTAACAAATGAGACTCGTATGCCGTCTTCTGCTTGT
CAAGTCTCAAGATGTCAGGCTGCTAGCCGCAGGGTTAACGACTCATTGTTTACTCTGAGCATACGGCAGAAGACGAACA
GTTCAGAGTTCTACAGTCCGACGATCTCCATCTTGGGGCGTCCCAATTGCTGAGTAACAAATTCGTATGCCGTCTTCTGCTTGT
CAAGTCTCAAGATGTCAGGCTGCTAGAGGTAGAACCCCGCAGGGTTAACGACTCATTGTTTAAGCATACGGCAGAAGACGAACA
GTTCAGAGTTCTACAGTCCGACGATCAACAAATGAGACGCTGTGCAATTGCTGAGTTCGTATGCCGTCTTCTGCTTGT
CAAGTCTCAAGATGTCAGGCTGCTAGTTGTTTACTCTGCGACACGTTAACGACTCAAGCATACGGCAGAAGACGAACA
Attach DNA to flowcell or bead
Perform amplification
Generate clusters
Sequencing
35 nt
sequenced in the adapter
flowcell

Background



sequenced reads

```
TCCCAATTGCTGAGTAACAAATGAGACGCTGTGCA
CCGCAGGGTTAACGACTCATTGTTTACTCTGCAAG
AGGTAGAACCCCGCAGGGTTAACGACTCATTGTTT
TTGTTTACTCTGCGACACGTTAACGACTCACAAGT
```

before clipping

2 mapping errors

```
                CCGCAGGGTTAACGACTCATTGTTTACTCTGCAAG
                |||||||||||||||||||||||||||||||| |
CGTAGGTAGAACCCCGCAGGGTTAACGACTCATTGTTTACTCTGCGACACGTTAACGACTCATG
```

DNA

after clipping

0 mapping errors

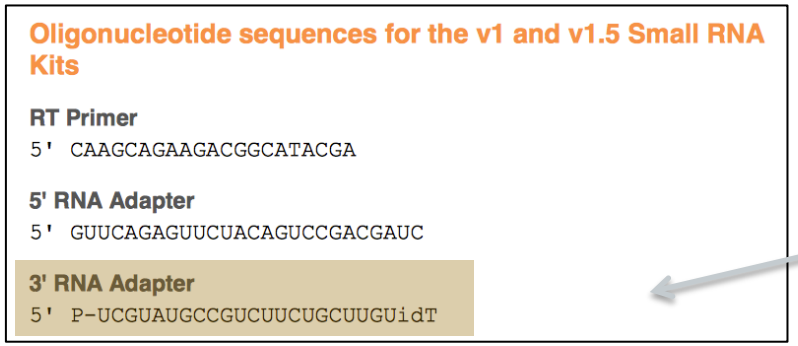
```
                CCGCAGGGTTAACGACTCATTGTTTACTCTG
                |||||||||||||||||||||||||||||||
CGTAGGTAGAACCCCGCAGGGTTAACGACTCATTGTTTACTCTGCGACACGTTAACGACTCATG
```

DNA

Method

```
@
TAAGTGGGAGGCCCTCGTATGCCGTCTTCTGCTTGTAAAAAAAAAAAATA
+
BCA6<>>BBAB?AC@AABACBAB?'>A:A>?@A@################
@
```

1. Get the correct adapter sequence from the Illumina Customer Sequence Letter'
(Link:https://support.illumina.com/content/dam/illumina-support/documents/documentation/chemistry_documentation/experiment-design/illumina-adapter-sequences-1000000002694-06.pdf)



This is not our adapter!
This adapter sequence is for small RNA-Seq

Method

```
@
TAAGTGGGAGGCCCTCGTATGCCGTCTTCTGCTTGTAAAAAAAAAAAATA
+
BCA6<>>BBAB?AC@AABACBAB?'>A:A>?@A@################
@
```

1. Get the correct adapter sequence from the Illumina Customer Sequence Letter'

   UCGUAUGCCGUCUUCUGCUUGU

2. Align the adapter sequence to each read

```
                UCGUAUGCCGUCUUCUGCUUGU
@               |||||||||| ||||||||||
TAAGTGGGAGGCCCTCGTATGCCGACTTCTGCTTGTAAAAAAAAAAAATA
+
BCA6<>>BBAB?AC@AABACBAB?'>A:A>?@A@################
@
```

Alignment parameters:

- Minimum length of detected adapter sequence (10 nt)
- Number of allowed mismatches

Method

```
@
TAAGTGGGAGGCCCTCGTATGCCGTCTTCTGCTTGTAAAAAAAAAAAATA
+
BCA6<>>BBAB?AC@AABACBAB?'>A:A>?@A@################
@
```

1. Get the correct adapter sequence from the Illumina Customer Sequence Letter’

   UCGUAUGCCGUCUUCUGCUUGU

2. Align the adapter sequence to each read

   ```
   @
   TAAGTGGGAGGCCCTCGTATGCCGTCTTCTGCTTGTAAAAAAAAAAAATA
   +
   BCA6<>>BBAB?AC@AABACBAB?'>A:A>?@A@################
   @
   ```

3. Clip adapter

   ```
   @
   TAAGTGGGAGGCCC
   +
   BCA6<>>BBAB?AC
   @
   ```

Method

```
@
TAAGTGGGAGGCCCTCGTATGCCGTCTTCTGCTTGTAAAAAAAAAAAATA
+
BCA6<>>BBAB?AC@AABACBAB?'>A:A>?@A@################
@
TGAGGTAGTAGATTGTATAGTTTCGTATGCCGTCTTCTGCTTGATTATGT
+
BC@2ABCC?BBA?=BABB??ABB@B@=A@?@B?AB;##############
@
TCGTATGCCGTCTTCTGCTTGAAAAAAAAAAAATAATTTTTTTTTTTTTT
+
B@;>?BBB?3?BBBB@9@A?0<AA##########################
@
TTCAAGTAATCCAGGATAGGCAGTAATCGCCGTCTTCTGCTTTAATTTTT
+
>CBCBBA?@@CBB@?BA@7:?A7=>8/:7<>=29################
@
TAATACTGCCTGGTAATGATGACTCGTATGCCGTCTTCTGCTTGTTGTGG
+
BCCCCCBCCCCCC?>ACCCBCCBAB>>@@BBAB=;;?=B@##########
@
TGAGGTAGTAGATTGTATAGTTTCGTATGCCGTCTTCTGCTTGATTTTTT
+
BCAAA?BC:6<AABA>@:98=B:AA@A9>>@;??################
@
TAGCTTATCAGACTGATGTTGACTCGTATGCCGTCTTCTGCTTGTGTGTT
+
BBBB<BBBABCABBABB@@B>@?=B@7:@6A?=8>B>7?###########
```

Method

```
@
TAAGTGGGAGGCCCTCGTATGCCGTCTTCTGCTTGTAAAAAAAAAAAATA
+
BCA6<>>BBAB?AC@AABACBAB?'>A:A>?@A@################
@
TGAGGTAGTAGATTGTATAGTTTCGTATGCCGTCTTCTGCTTGATTATGT
+
BC@2ABCC?BBA?=BABB??ABB@B@=A@?@B?AB;##############
@
TCGTATGCCGTCTTCTGCTTGAAAAAAAAAAAAATAATTTTTTTTTTTTT
+
B@;>?BBB?3?BBBB@9@A?0<AA##########################
@
TTCAAGTAATCCAGGATAGGCAGTAATCGCCGTCTTCTGCTTTAATTTTT
+
>CBCBBA?@@CBB@?BA@7:?A7=>8/:7<>=29################
@
TAATACTGCCTGGTAATGATGACTCGTATGCCGTCTTCTGCTTGTTGTGG
+
BCCCCCBCCCCCC?>ACCCBCCBAB>>@@BBAB=;;?=B@##########
@
TGAGGTAGTAGATTGTATAGTTTCGTATGCCGTCTTCTGCTTGATTTTTT
+
BCAAA?BC:6<AABA>@:98=B:AA@A9>>@;??################
@
TAGCTTATCAGACTGATGTTGACTCGTATGCCGTCTTCTGCTTGTGTGTT
+
BBBB<BBBABCABBABB@@B>@?=B@7:@6A?=8>B>7?###########
```

Method

```
@
TAAGTGGGAGGCCC                                                    14 nt
+
BCA6<>>BBAB?AC
@
TGAGGTAGTAGATTGTATAGTT                                            22 nt
+
BC@2ABCC?BBA?=BABB??AB
@
                                                                  0 nt
+

@
TTCAAGTAATCCAGGATAGGCAGTAATCGCCGTCTTCTGCTTTAATTTTT                not trimmed
+
>CBCBBA?@@CBB@?BA@7:?A7=>8/:7<>=29################
@
TAATACTGCCTGGTAATGATGAC                                           22 nt
+
BCCCCCBCCCCCC?>ACCCBCCB
@
TGAGGTAGTAGATTGTATAGTT                                            22 nt
+
BCAAA?BC:6<AABA>@:98=B
@
TAGCTTATCAGACTGATGTTGAC                                           23 nt
+
BBBB<BBBABCABBABB@@B>@?
```

Clip Adapter

Paired End DNA oligonucleotide sequences

PE Adapters

5' P-GATCGGAAGAGCGGTTCAGCAGGAATGCCGAG
5' ACACTCTTTCCCTACACGACGCTCTTCCGATCT

Task
Try to find the adapter sequences for our paired-end experiments in the ‘Illumina Customer Sequence Letter’.

Clip Adapter

Task
Use cutadapt to trim reads with bad quality at their 3' ends (https://code.google.com/p/cutadapt/)

```
$ cd ..
$ mkdir clipped
$ cd clipped
```

Clip Adapter

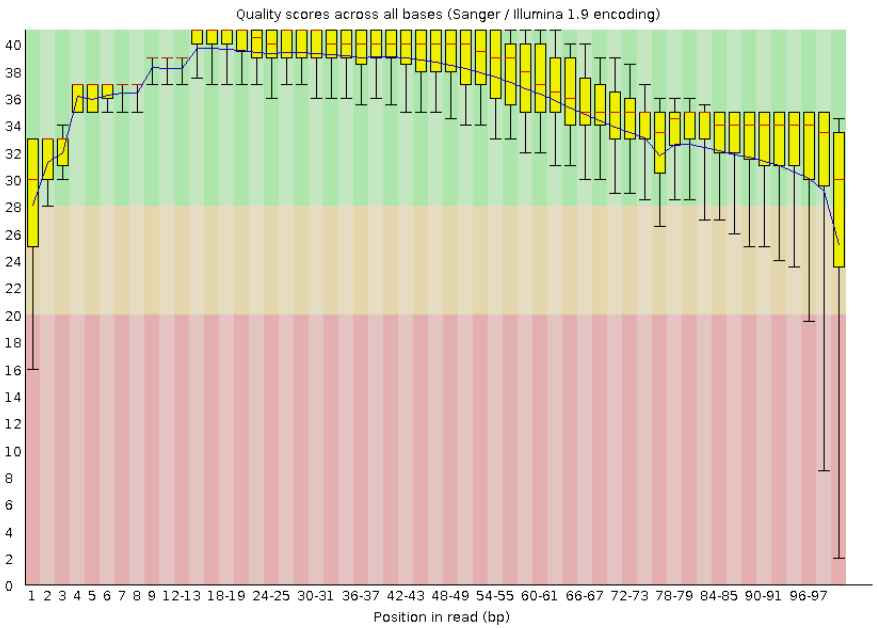
Task
Use cutadapt to trim reads with bad quality at their 3' ends (https://code.google.com/p/cutadapt/)

```
$ cutadapt -a GATCGGAAGAGCGGTTCAGCAGGAATGCCGAG
  -A ACACTCTTTCCCTACACGACGCTCTTCCGATCT -q 20 -O 10 -m 25
  -o SRR359063_1.trimmed.fastq.gz -p SRR359063_2.trimmed.fastq.gz
  ../raw/SRR359063_1.fastq.gz ../raw/SRR359063_2.fastq.gz
```

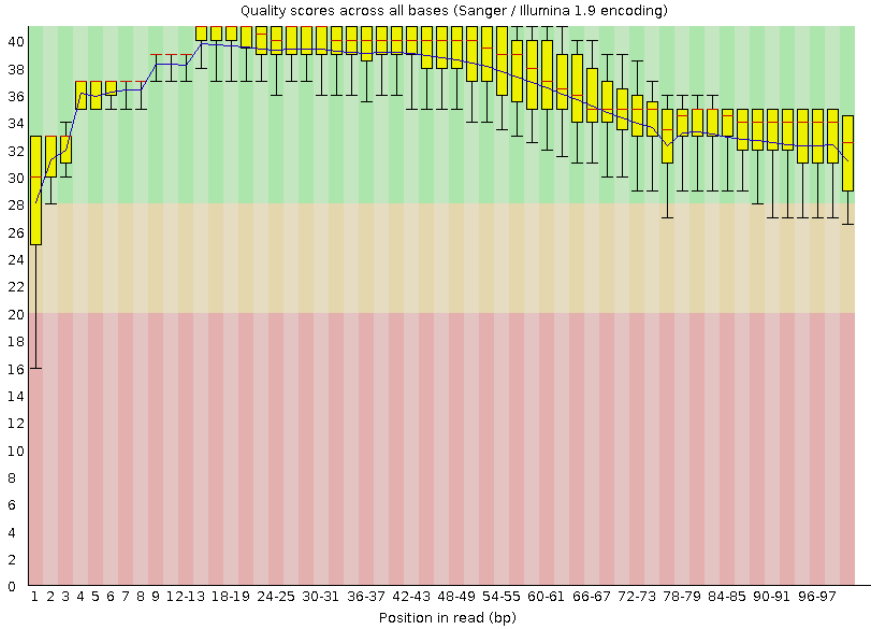
Clip Adapter

Task
Check your clipped reads using fastqc

```
$ fastqc SRR359063_1.trimmed.fastq.gz
```



before



after

Read Mapping

Pairwise Alignment

A formal alignment definition

We assume we have two sequences u, v over the alphabet A of length m and n, respectively.

$\mathbf{u = u_1 u_2 \ldots u_m}$

$\mathbf{v = v_1 v_2 \ldots v_n}$

Note that these sequences can be of different length.

A formal alignment definition

First we need to model ‘biological’ or ‘chemical’ events.

Typically we restrict ourselves to these three events:

edit operations:

($u_i \rightarrow v_j$) substitution
($u_i \rightarrow \varepsilon$) deletion
($\varepsilon \rightarrow v_j$) insertion

We call a substitution a match if $u_i = v_j$ or mismatch if $u_i \neq v_j$.

A formal alignment definition

An example:

```
u = ACTCG-CCACGCG
    ||  ||  ||||  ||
v = ACCCGGCCAC-CG
```

(A→A)(C →C)(T →C)(C →C)(G →G)(ε→G)…

A formal alignment definition

An example:

```
u = ACTCG-CCACGCG
    || || |||| ||
v = ACCCGGCCAC-CG
```

This is also an alignment of u and v:

```
u = ACTCGCCACGCG
    || || |    ||
v = ACCCGGCCACCG
```

What is the difference?

Dynamic Programming

The DP is explained using a global alignment. It can easily be modified to a semi-global alignment.

The unit edit distance (edist) is the number of mismatches, insertions and deletions in an optimal sequence alignment.

- Minimize the edist.
- Partial solutions are tabulated in a (m + 1) x (n + 1) matrix.

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 & \text{insertion} \\ E(i,j-1)+1 & \text{deletion} \\ E(i-1,j-1)+(u[i] \neq v[j]) & \text{substitution} \end{cases}$$

Dynamic Programming

The DP is explained using a global alignment. It can easily be modified to a semi-global alignment.

The unit edit distance (edist) is the number of mismatches, insertions and deletions in an optimal sequence alignment.

- Minimize the edist.
- Partial solutions are tabulated in a (m + 1) x (n + 1) matrix.

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 & \text{insertion} \\ E(i,j-1)+1 & \text{deletion} \\ E(i-1,j-1)+(u[i]\neq v[j]) & \text{substitution} \end{cases}$$

Note that the penalties can easily be replaced by some other scoring function

Dynamic Programming

Initialization of the alignment matrix

	i →	T	G	A	T	A	T
j ↓	0	1	2	3	4	5	6
G	1						
C	2						
A	3						
C	4						
T	5						

← deletion E(i,0) = j

↑ insertion E(0,j) = i

Dynamic Programming

	i →	T	G	A	T	A	T
j ↓	0	1	2	3	4	5	6
G	1	?					
C	2						
A	3						
C	4						
T	5						

Dynamic Programming

	i →	T	G	A	T	A	T
j ↓	0	1	2	3	4	5	6
G	1	?					
C	2						
A	3						
C	4						
T	5						

0	1
1	2

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 & \text{insertion} \\ E(i,j-1)+1 & \text{deletion} \\ E(i-1,j-1)+(u[i] \neq v[j]) & \text{substitution} \end{cases}$$

2

Dynamic Programming

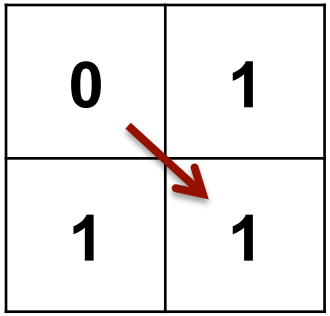
	i →	T	G	A	T	A	T
j ↓	0	1	2	3	4	5	6
G	1	?					
C	2						
A	3						
C	4						
T	5						

0	1
1	→ 2

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 & \text{insertion} \quad 2 \\ \mathbf{E(i,j-1)+1} & \mathbf{\text{deletion} \quad 2} \\ E(i-1,j-1)+(u[i]\neq v[j]) & \text{substitution} \end{cases}$$

Dynamic Programming

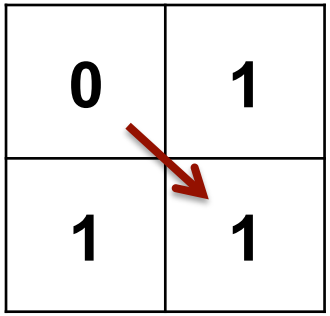
	i →	T	G	A	T	A	T
j ↓	0	1	2	3	4	5	6
G	1	?					
C	2						
A	3						
C	4						
T	5						



$$E(i,j) = \min \begin{cases} E(i-1,j)+1 & \text{insertion} \quad 2 \\ E(i,j-1)+1 & \text{deletion} \quad 2 \\ E(i-1,j-1)+(u[i] \neq v[j]) & \text{substitution} \quad 1 \end{cases}$$

Dynamic Programming

	i →	T	G	A	T	A	T
j ↓	0	1	2	3	4	5	6
G	1	**1**					
C	2						
A	3						
C	4						
T	5						



$$E(i,j) = \min \begin{cases} E(i-1,j)+1 & \text{insertion} \quad 2 \\ E(i,j-1)+1 & \text{deletion} \quad 2 \\ E(i-1,j-1)+(u[i] \neq v[j]) & \text{substitution} \quad 1 \end{cases}$$

Dynamic Programming

	i →	T	G	A	T	A	T
j ↓	0	1	2	3	4	5	6
G	1	1	1	2	3	4	5
C	2	2	2	2	3	4	5
A	3	3	3	2	3	4	5
C	4	4	4	3	3	4	4
T	5	4	5	4	3	4	4

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 & \text{insertion} \\ E(i,j-1)+1 & \text{deletion} \\ E(i-1,j-1)+(u[i] \neq v[j]) & \text{substitution} \end{cases}$$

Dynamic Programming

	i →	T	G	A	T	A	T
j ↓	0	1	2	3	4	5	6
G	1	1	1	2	3	4	5
C	2	2	2	2	3	4	5
A	3	3	3	2	3	4	5
C	4	4	4	3	3	4	4
T	5	4	5	4	3	4	**4**

← edist of u and v

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 & \text{insertion} \\ E(i,j-1)+1 & \text{deletion} \\ E(i-1,j-1)+(u[i] \neq v[j]) & \text{substitution} \end{cases}$$

Dynamic Programming

Backtracking

		T	G	A	T	A	T
	0	1	2	3	4	5	6
G	1	1	1	2	3	4	5
C	2	2	2	2	3	4	5
A	3	3	3	2	3	4	5
C	4	4	4	3	3	4	4
T	5	4	5	4	3	4	4

```
TGATAT
  |   |
-GCACT
```

```
TGATAT
  |   |
GCAC-T
```

```
TG-A-TAT
  | | |
-GCACT--
```

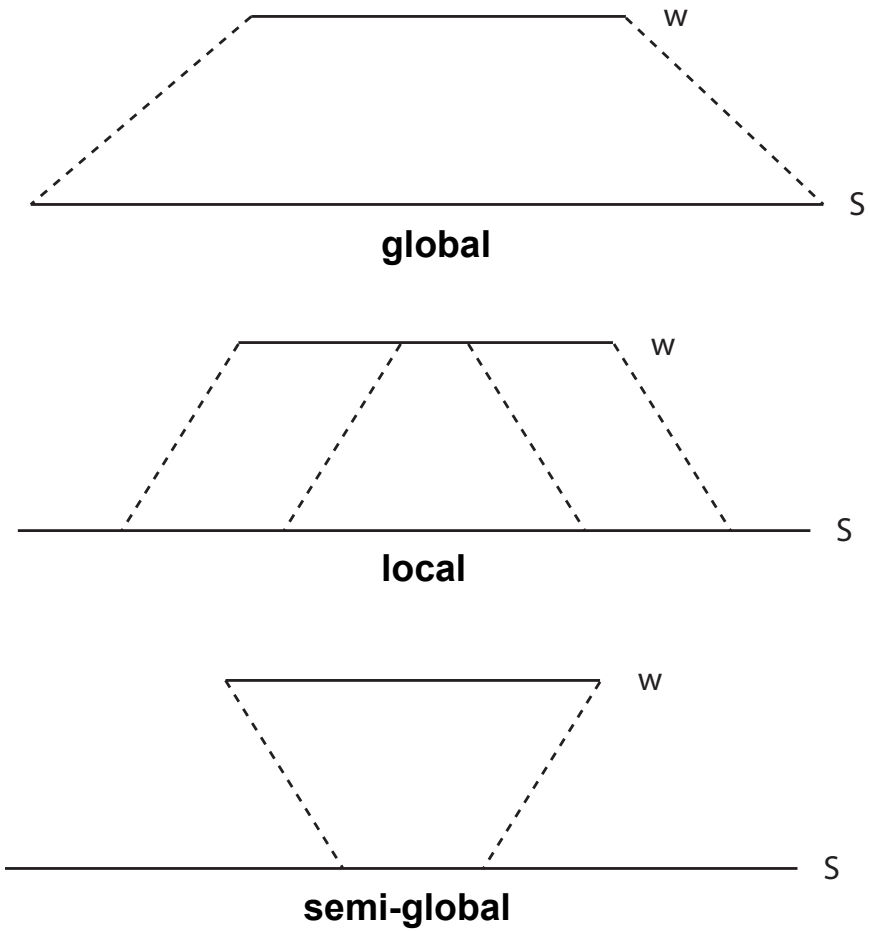
Optimal Sequence Alignment

The previous example was a global alignment. For aligning comparably short reads to large reference genomes other alignments are more useful.

For example:

1. local alignment
   - Smith-Waterman algorithm
   - Gotoh algorithm
2. semi-global alignment
   - Modified Needleman-Wunsch algorithm
   - Myers' bit-vector algorithm (edist-bounded)

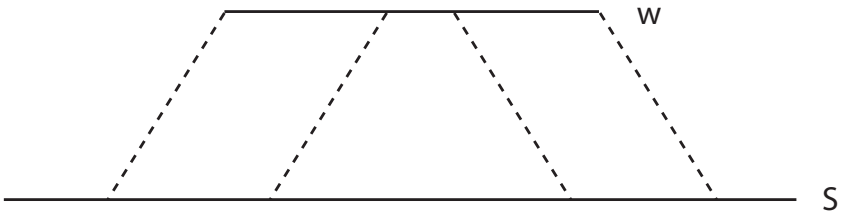
Optimal Sequence Alignment



Optimal Sequence Alignment

global

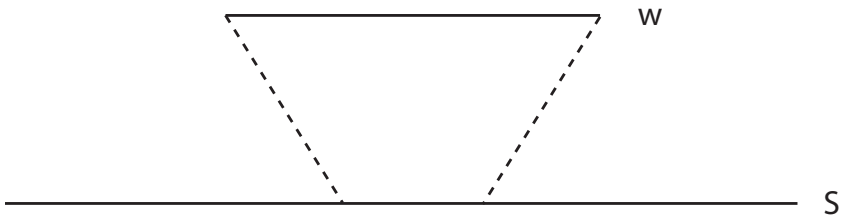
```
--GAAC-GTGGATTC--------------------
  ||||  |||  |   |
ACGAACAGTG-A--CACGATCACAAGGGAACGATTC
```



local

```
   GAAC-GTGGATTC
   ||||  |||
ACGAACAGTGACACGATCACAAGGGAACGATTC

                              GAACGTGGATTC
                              |||||
ACGAACAGTGACACGATCACAAGGGAACGATTC
```



semi-global

```
  GAAC-GTGGATTC
  ||||  |||  |   |
ACGAACAGTG-A--CACGATCACAAGGGAACGATTC
```

Optimal Sequence Alignment

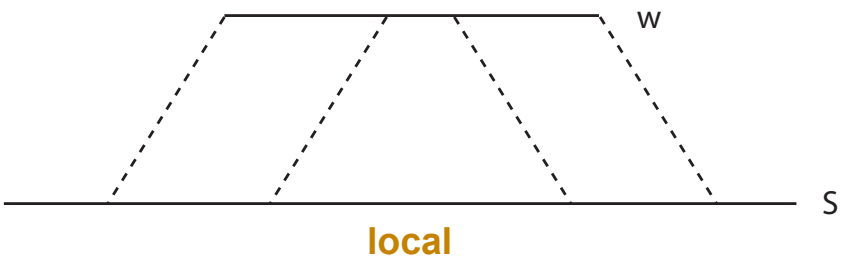
global

deletion E(i,0) = j

		T	G	A	T	A	T
	0	1	2	3	4	5	6
G	1						
C	2						
A	3						
C	4						
T	5						

insertion E(0,j) = i

Optimal Sequence Alignment



deletion E(i,0) = 0

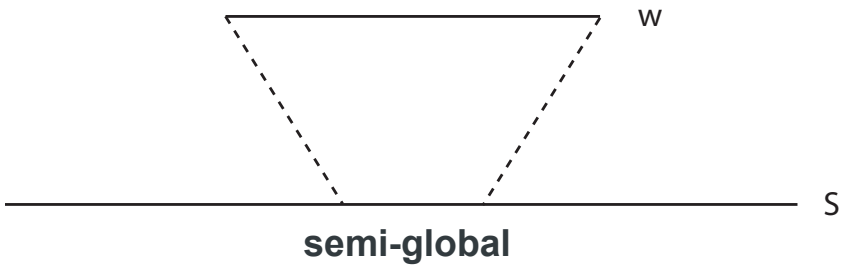
		T	G	A	T	A	T
	0	0	0	0	0	0	0
G	0						
C	0						
A	0						
C	0						
T	0						

insertion E(0,j) = 0

Optimal Sequence Alignment

deletion E(i,0) = 0

		T	G	A	T	A	T
	0	0	0	0	0	0	0
G	1						
C	2						
A	3						
C	4						
T	5						



insertion E(0,j) = i

Optimal Sequence Alignment

Semi-Global or Local?

- No difference for exact matches
- local alignments
  - aligns substrings (risk: false positive)
  - helps to skip adapters, poly-A, mismatch-runs in head or tail
- semi-global alignment
  - aligns whole read (risk: false negative)
  - more specific

Optimal Sequence Alignment

The scoring/distance function

The optimal alignments are the alignments with the maximum score or minimum distance according to a scoring or a distance function.

Optimal Sequence Alignment

Complexity

With dynamic programming local and semi-global alignments require O(nm) in time.

One NGS-read vs. Human genome
3.2G · 100 = 320, 000, 000, 000, 000 calculations!!

Mapping Problem

The search space

Search problem

Find the correct locus of origin of a **100 bp** long read within a **3.0 x 10^9 bp** long reference genome.

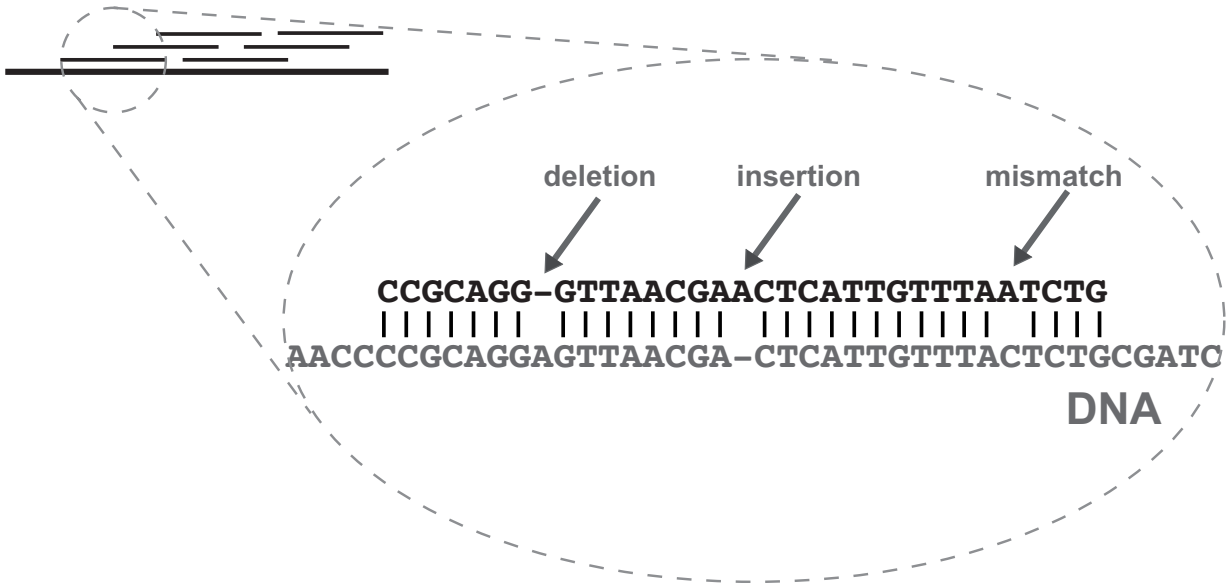
Example
If the mapping of **one read** takes **0.5 seconds**, you would need to wait around **130 days**, until **all reads** are mapped (on 30 cores ~4 days).

BUT: It's even more difficult!

The reads

The reads have errors:

- Errors from PCR amplification and sequencing machine
  - Wrong nucleotides (mismatches).
  - Missing nucleotides (deletions).
  - Inserted nucleotides (insertions).
- Adapter sequences not correctly clipped
- Contaminated sample



The reference genome

The reference genome is not perfect:

- Missing region (Ns)
- Repeats / Low complexity regions
- SNVs
- Genome rearrangements

Example - The Human Genome

Despite enormous financial and scientific efforts to generate a high quality reference recent studies report:

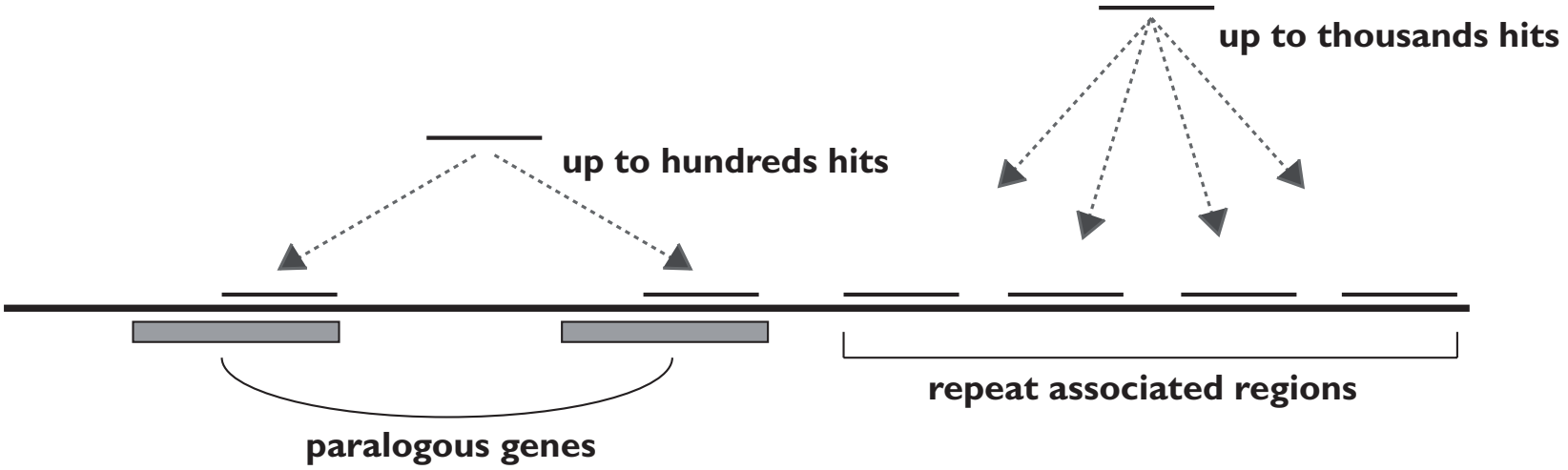
- 23–29 Mb are absent from the latest build [Wang et al.]
- 2,363 novel insertion sequences (720 loci) [Kidd et al.]
- 5 Mb novel sequences and 104 unalignable RefSeq genes [Chen et al.]

In these cases the retrieval of the origin of a read must fail.

Multiple mapping loci

The genome is non-randomly distributed and has repetitive regions:

- paralogous genes
- Transposable elements (e.g. Alu repeats)



The expectation of a sequence

Assume an alphabet of 4 characters {A,C,G,T} and a text of length n.

Question

When is a sequence uniquely occurring in the text?

The expectation of a sequence

Assumptions

- read of length m, reference of length n
- both uniformly randomly composed
- each character occurs with p = 0.25

$$E = p^m \cdot n \quad (1)$$

1. Solving for m:

$$m = \log_p(E/n) \quad (2)$$

2. Setting E = 1:

$$m = \log_p(1/n) = -\log(n)/\log(p) \quad (3)$$

The expectation of a sequence

Assumptions

- length n is 3.2G nucleotides

$$-\log(3.2G)/\log(0.25) \approx 16$$

BUT:

- non-randomly distributed
- repetitive regions

➔ **more than 16 nucleotides necessary to map uniquely**

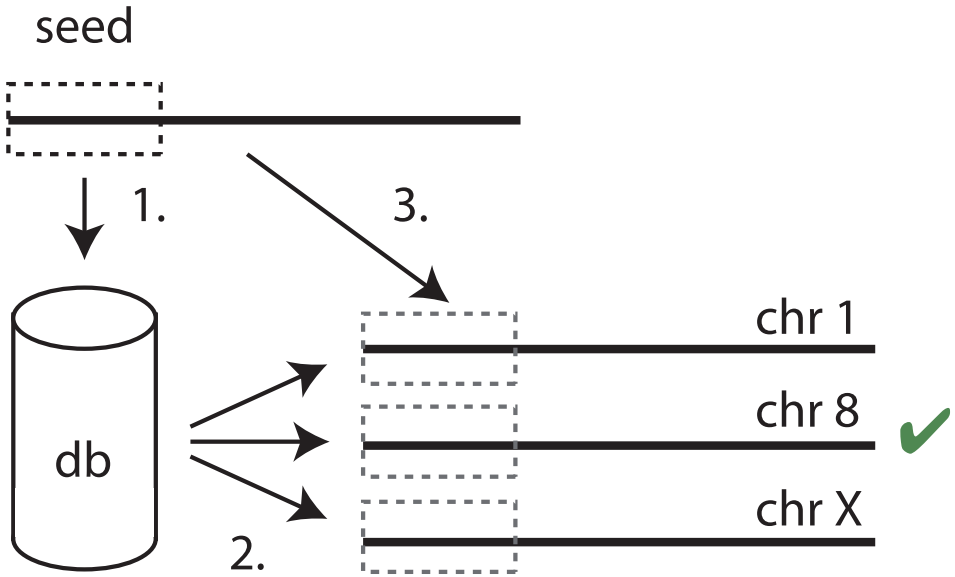
NGS Alignment

Seed Search

Alignment Heuristics

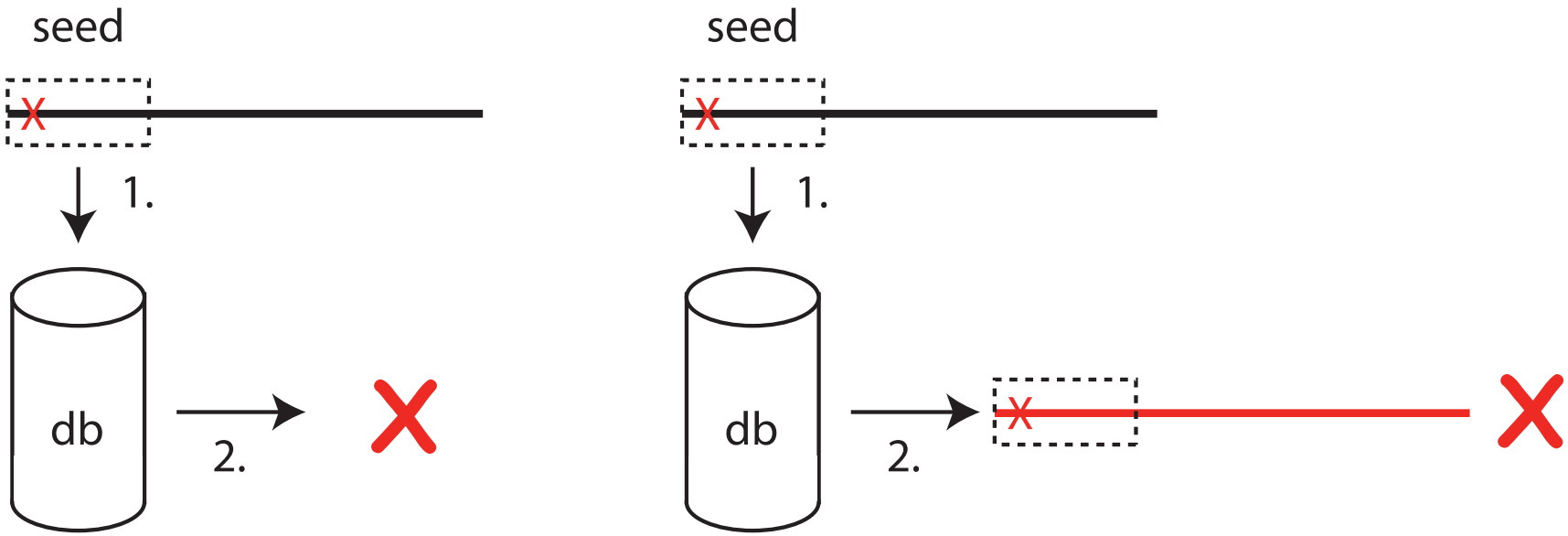
Typically, alignment heuristics aim to find the optimal alignment in large sequences in three steps:

1. fast (near-)exact search of read substrings/seeds
2. elongate regions with seed matches
3. perform optimal alignments



Seed Search

Seed search can fail - handling of errors in seed regions is crucial



The seed is not in the db

The wrong seed (with the error) is available and the db entry leads to a wrong locus

Seed Search

k-mer indices

Method

- fragment genome in pieces of length k
- store position(s) of each fragment in a hash (forward and reverse)
  - key = sequence
  - value = position(s) in the genome

big memory footprint, but fast

AGCGATGACGAGTCAT	chr8:304
GCGATGACGAGTCATC	chr8:305
CGATGACGAGTCATCA	chr3:59438;chr8:306
GATGACGAGTCATCAT	chr8:307
ATGACGAGTCATCATA	chr8:308;chrX:97364

Seed Search

k-mer indices

Problem:

- too short seed length results in too many false positives
  - shorter seed results in more positions and longer running time
  - in most cases it will find the correct mapping position
- too long seed length might result in no hit at all

AGCGATGACGAGTCAT	chr8:304
GCGATGACGAGTCATC	chr8:305
CGATGACGAGTCATCA	chr3:59438;chr8:306
GATGACGAGTCATCAT	chr8:307
ATGACGAGTCATCATA	chr8:308;chrX:97364

Mapping Example

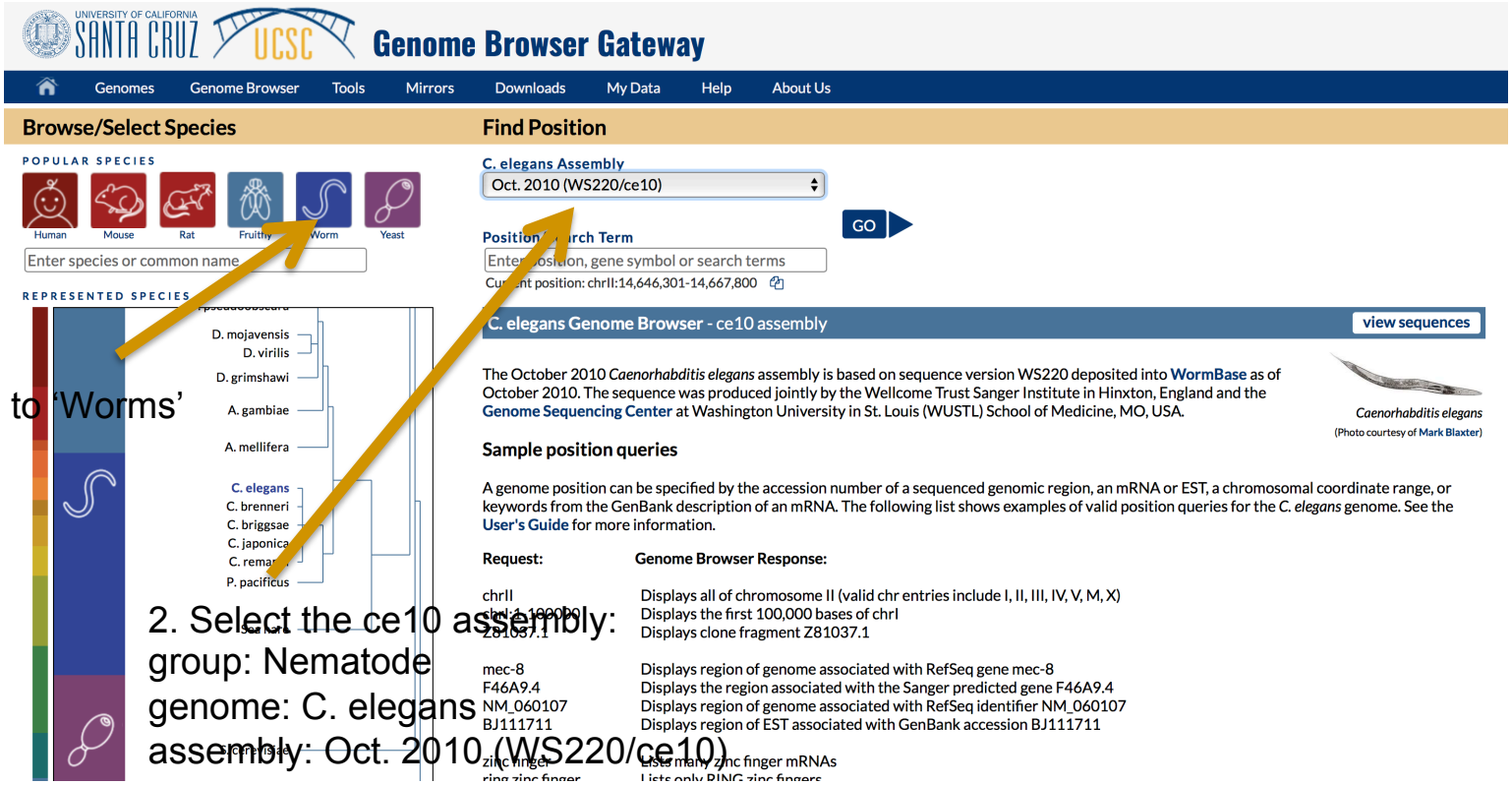
Typical Proceeding:

1. find all seeds
2. go to every locus and elongate the seed match on both sides
3. compute a semi-global or local alignment
4. discard all alignments with more than the maximum number of allowed errors
5. compare all alignments and return these with the minimum number of errors

Read Mapping

Get reference genome

Task
Download the C. elegans genome from the UCSC Genome Browser (http://genome.ucsc.edu/cgi-bin/hgGateway)

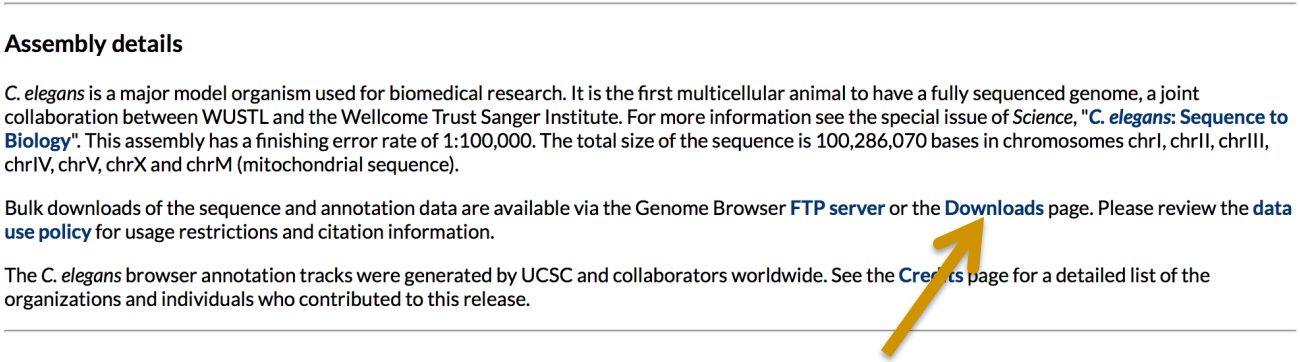


1. go to 'Worms'

2. Select the ce10 assembly: group: Nematode genome: C. elegans assembly: Oct. 2010 (WS220/ce10)

Get reference genome

Task
Download the C. elegans genome from the UCSC Genome Browser (http://genome.**ucsc**.edu/cgi-bin/hgGateway**)**



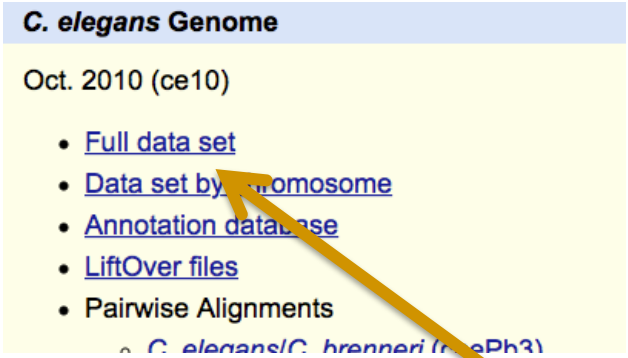
Assembly details

C. elegans is a major model organism used for biomedical research. It is the first multicellular animal to have a fully sequenced genome, a joint collaboration between WUSTL and the Wellcome Trust Sanger Institute. For more information see the special issue of *Science*, "***C. elegans*: Sequence to Biology**". This assembly has a finishing error rate of 1:100,000. The total size of the sequence is 100,286,070 bases in chromosomes chrI, chrII, chrIII, chrIV, chrV, chrX and chrM (mitochondrial sequence).

Bulk downloads of the sequence and annotation data are available via the Genome Browser **FTP server** or the **Downloads** page. Please review the **data use policy** for usage restrictions and citation information.

The *C. elegans* browser annotation tracks were generated by UCSC and collaborators worldwide. See the **Credits** page for a detailed list of the organizations and individuals who contributed to this release.

3. Scroll down to ‘Assembly details’ and click on ‘Downloads’



***C. elegans* Genome**

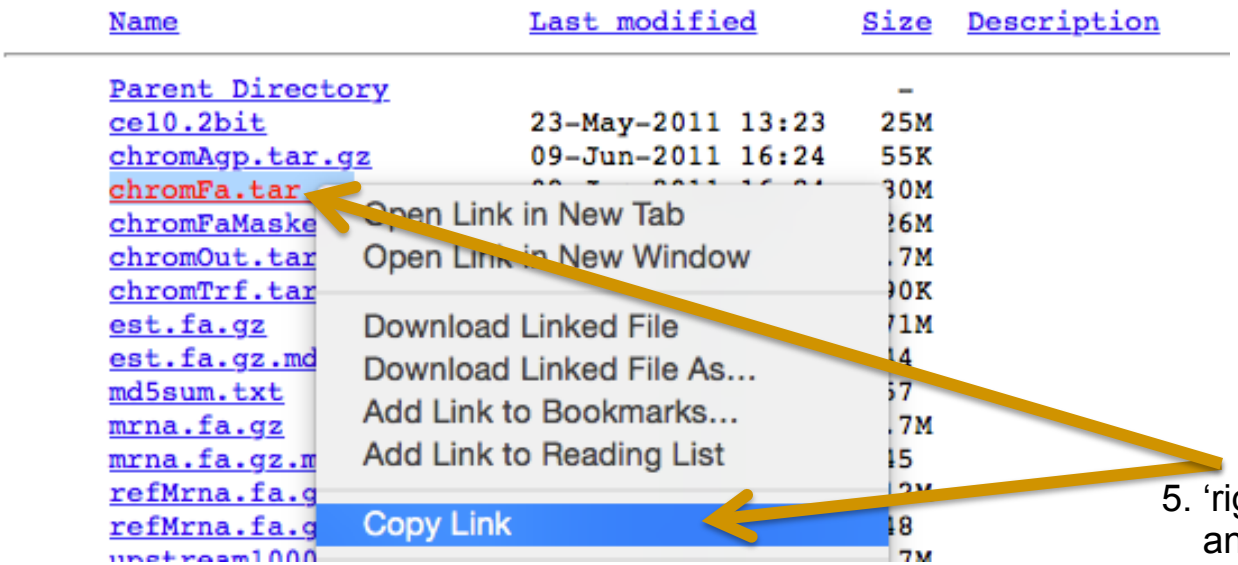
Oct. 2010 (ce10)

- Full data set
- Data set by chromosome
- Annotation database
- LiftOver files
- Pairwise Alignments
  - *C. elegans*/*C. brenneri* (ce4Ph3)

4. Click on ‘Full data set’

Get reference genome

Task
Download the C. elegans genome from the UCSC Genome Browser (http://genome.**ucsc**.edu/cgi-bin/hgGateway**)**



5. ‘right click on ‘chromFA.tar.gz and select ‘Copy Link’

Get reference genome

Task
Download the C. elegans genome from the UCSC Genome Browser (http://genome.**ucsc**.edu/**)**

6. Create a genome folder and download the genome fasta file using wget (paste the copied link after wget)

```
$ cd ..
$ mkdir genome
$ cd genome
$ wget http://hgdownload.cse.ucsc.edu/goldenPath/ce10/bigZips/
  chromFa.tar.gz
$
$ tar xvf chromFA.tar.gz
$ cat chr*.fa > ce10.fa
$ rm chr*
```

7. Unzip the downloaded file and merge the single chromosome files to one final genome file.

Create Index

Task
Create index for the C. elegans genome

segemehl

```
$ mkdir segemehl
$ segemehl.x -d ce10.fa -x segemehl/ce10.idx
```

bowtie2

```
$ mkdir bowtie2
$ bowtie2-build ce10.fa bowtie2/ce10
```

bwa

```
$ mkdir bwa
$ cp ce10.fa bwa/
$ cd bwa
$ bwa index ce10.fa
$ cd ..
```

Create Index

Task
Create index for the C. elegans genome

star

```
$ mkdir star
$ STAR --runMode genomeGenerate --genomeDir star/ --genomeFastaFiles ce10.fa
```

Map Reads

Task
Map the clipped reads against our C. elegans genome

```
$ cd ..
$ mkdir mapping
$ cd mapping
```

segemehl

```
$ segemehl.x -d ../genome/ce10.fa -i ../genome/segemehl/ce10.idx
  -q ../clipped/SRR359063_1.trimmed.fastq.gz
  -p ../clipped/SRR359063_2.trimmed.fastq.gz -t 4 -S
  > SRR359063.segemehl.sam
```

bowtie2

```
$ bowtie2 -x ../genome/bowtie2/ce10
  -1 ../clipped/SRR359063_1.trimmed.fastq.gz
  -2 ../clipped/SRR359063_2.trimmed.fastq.gz > SRR359063.bowtie2.sam
```

Map Reads

Task
Map the clipped reads against our C. elegans genome

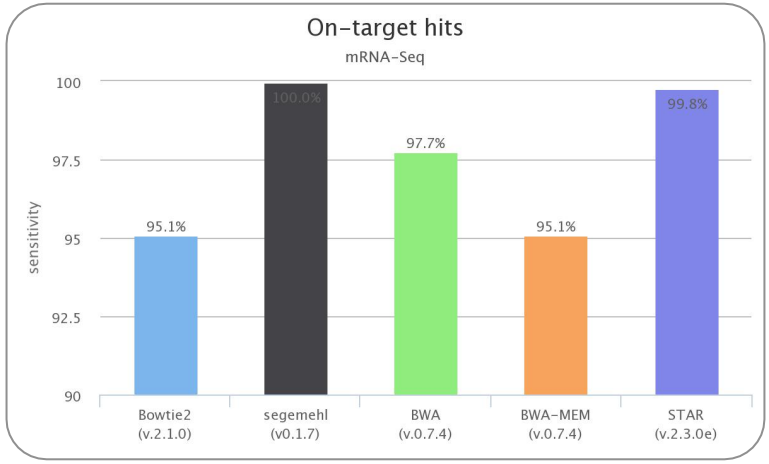
bwa

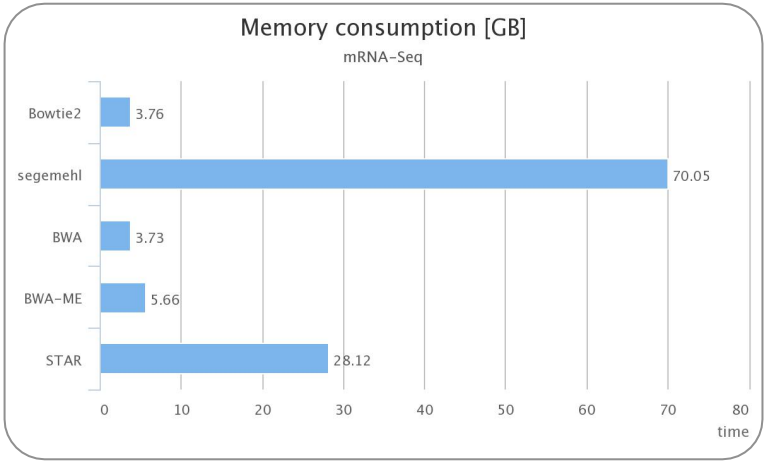
```
$ bwa aln ../genome/bwa/ce10.fa ../clipped/SRR359063_1.trimmed.fastq.gz
  > SRR359063_1.sai
$ bwa aln ../genome/bwa/ce10.fa ../clipped/SRR359063_2.trimmed.fastq.gz
  > SRR359063_2.sai
$ bwa sampe ../genome/bwa/ce10.fa SRR359063_1.sai SRR359063_2.sai
  ../clipped/SRR359063_1.trimmed.fastq.gz
  ../clipped/SRR359063_2.trimmed.fastq.gz > SRR359063.bwa.sam
$ rm *.sai
```

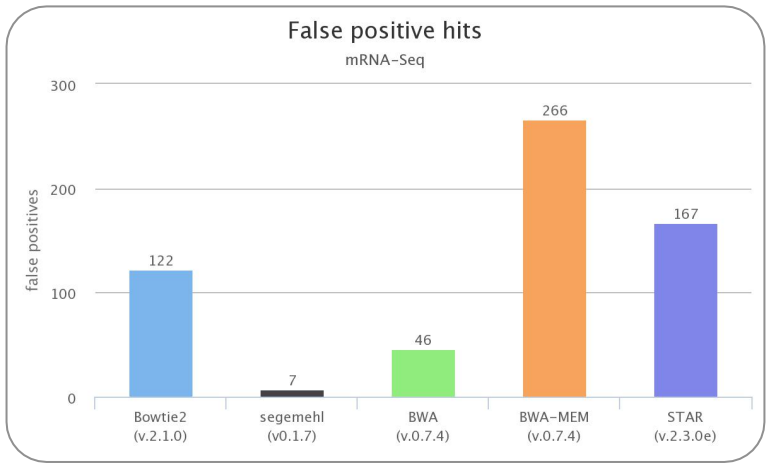
star

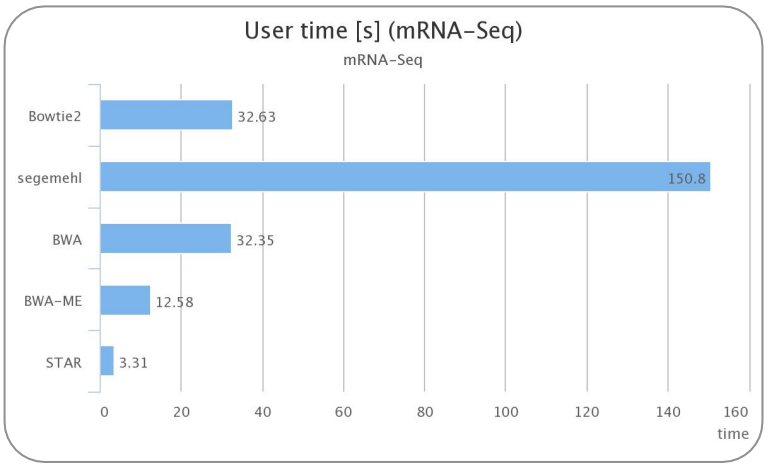
```
$ STAR --genomeDir ../genome/star/ --readFilesIn
  ../clipped/SRR359063_1.trimmed.fastq.gz
  ../clipped/SRR359063_2.trimmed.fastq.gz
  --outFileNamePrefix SRR359063.star. --outSAMattributes All
  --runThreadN 5 --readFilesCommand zcat
```

Popular Mapping Tools









Source: http://www.ecseq.com/support/benchmark.htm

Popular Mapping Tools

tool	Multiple Loci	Mismatches		InDels		Type
		Seed	Alignment	Seed	Alignment	
segemehl	Y	Y	Y	Y	Y	ESA
BWA	y	Y	Y	N	Y	BWT
Bowtie2	y	y	Y	N	Y	BWT
STAR	Y	N	Y	N	Y	SA
MAQ	N	Y	Y	N	Y	HT
SOAP2	y	N	Y	N	N	BWT

Abbreviations :
Y: default, N: not possible y: possible, but not default
SA: suffix array, ESA: enhanced suffix array, BWT: Burrows-Wheeler-transform, HT: hash table

Mapped Reads

SAM/BAM Format

SAM Format

Sequence Alignment/Map Format Specification:
http://samtools.github.io/hts-specs/SAMv1.pdf

Task

Access the SAM files using the view tool of the samtools toolkit (http://samtools.sourceforge.net)

```
$ samtools view -hS SRR359063.segemehl.sam | head
@HD     VN:1.0
@SQ     SN:I    LN:15072434
@SQ     SN:II   LN:15279421
@SQ     SN:III  LN:13783801
@SQ     SN:IV   LN:17493829
@SQ     SN:MtDNA        LN:13794
@SQ     SN:V    LN:20924180
@SQ     SN:X    LN:17718942
@RG       ID:A1     SM:sample1          LB:library1         PU:unit1  PL:illumina
@PG     ID:segemehl     VN:0.1.7-$Rev: 407 $ ($Date: 2014-02-06 10:55:25 +0100 (Thu, 06
Feb 2014) $)    CL:/scr/k41san/tools/segemehl/0_1_7/segemehl/segemehl.x -d ../../genome/
WBcel235.fa -i ../../genome/indices/segemehl/WBcel235.idx -q ../../clipped/
SRR359063_1.clipped.fastq.gz -p ../../clipped/SRR359063_2.clipped.fastq.gz -S -t 20
SRR359063.183484 D042KACXX:3:1202:16105:70173 length=101        163     I       3376
255     101M    =       3426    -51
AATCGACGAAAATCGGTACAAAATCAAATAAAAATAGAAGGAAAATATTCAGCTCGTAAATCCGCAAGTGCGGCACGGCTCCGTGGGC
CGCGCCGCCTCTCG   2B:A2D2AA222:CBEACECBEECCE22<2D22
```

Info

-S tells samtools that the input file is in SAM format

-h returns the complete SAM format (Header + Alignments)

The Header

Sequence Alignment/Map Format Specification:
http://samtools.github.io/hts-specs/SAMv1.pdf

```
$ samtools view -hS SRR359063.segemehl.sam | head
@HD     VN:1.0
@SQ     SN:I    LN:15072434
@SQ     SN:II   LN:15279421
@SQ     SN:III  LN:13783801
@SQ     SN:IV   LN:17493829
@SQ     SN:MtDNA        LN:13794
@SQ     SN:V    LN:20924180
@SQ     SN:X    LN:17718942
@RG       ID:A1     SM:sample1          LB:library1         PU:unit1  PL:illumina
@PG     ID:segemehl     VN:0.1.7-$Rev: 407 $ ($Date: 2014-02-06 10:55:25 +0100 (Thu, 06
Feb 2014) $)    CL:/scr/k41san/tools/segemehl/0_1_7/segemehl/segemehl.x -d ../../genome/
WBcel235.fa -i ../../genome/indices/segemehl/WBcel235.idx -q ../../clipped/
SRR359063_1.clipped.fastq.gz -p ../../clipped/SRR359063_2.clipped.fastq.gz -S -t 20
```

@HD	The header line; VN: Format version; SO: Sorting order of alignments
@SQ	Reference sequence dictionary
@PG	Program; ID: Program record identifier; VN: Program version; CL: Command line
@RG	The read group (important if multiple groups of reads are present)

BAM file

Task
Use samtools to compress your SAM file to a BAM file

```
$ samtools view -bS SRR359063.segemehl.sam >
  SRR359063.segemehl.bam
```

Task
Check the file size to see the difference

```
$ ls -lh
-rw-r--r-- 1 david staff 158M Mar 16 13:52 SRR359063.segemehl.bam
-rw-r--r-- 1 david staff 682M Mar 16 13:59 SRR359063.segemehl.sam
```

BAM file

Task
Access the BAM file using samtools

```
$ samtools view SRR359063.segemehl.bam | head
```

Task
Access the header of the BAM file

```
$ samtools view -H SRR359063.segemehl.bam
```

BAM file

Task
Sort the BAM file

```
$ samtools sort SRR359063.segemehl.bam SRR359063.segemehl.sorted
$ mv SRR359063.segemehl.sorted.bam SRR359063.segemehl.bam
```

Info
Since all information are stored in the sorted BAM file, the unsorted bam file can be overwritten (and thus deleted) with the sorted one

Task
Create an index of the BAM file

```
$ samtools index SRR359063.segemehl.bam
```

SAM Format

Sequence Alignment/Map Format Specification:
http://samtools.github.io/hts-specs/SAMv1.pdf

Task

Access the SAM files using the view tool of the samtools toolkit (http://samtools.sourceforge.net)

```
$ samtools view SRR359063.segemehl.bam | head 1
SRR359063.183484 D042KACXX:3:1202:16105:70173 length=101        163    chrI 3376
         17        18=1X41=1X17=1X1=1X4=    =    3426          151
AATCGACGAAAATCGGTACAAAATCAAATAAAAATAGAAGGAAAATATTCAGCTCGTAAATCCGCAAGTGCGGCACGGCTCCGTG ?
B;4?D?AA???:CBEACEGBFEGGE??<?D??DEH998DEEGIJ9=BBF=@CGH8AAAA)7A>B,=>C@@@?6?B&05008<A?
HI:i:0      NH:i:1   NM:i:4   MD:Z:18T41C17T1T4    RG:Z:A1   YZ:Z:0  XA:Z:P
```

Info

Without –h returns the complete SAM format (without Header, only Alignments)

The Alignment

Sequence Alignment/Map Format Specification:
http://samtools.github.io/hts-specs/SAMv1.pdf

1 2 3

```
SRR359063.183484 D042KACXX:3:1202:16105:70173 length=101      163    chrI
3376    17      18=1X41=1X17=1X1=1X4=    =    3426    151
AATCGACGAAAATCGGTACAAAATCAAATAAAAATAGAAGGAAAATATTCAGCTCGTAAATCCGCAAGTGCGGCA
CGGCTCCGTG ?B;4?D?AA???:CBEACEGBFEGGE??<?D??
DEH998DEEGIJ9=BBF=@CGH8AAAA)7A>B,=>C@@@?6?B&05008<A?
HI:i:0    NH:i:1  NM:i:4  MD:Z:18T41C17T1T4 RG:Z:A1  YZ:Z:0  XA:Z:P
```

1	query template name
2	bitwise flag
3	reference sequence name

The Alignment

Sequence Alignment/Map Format Specification:
http://samtools.github.io/hts-specs/SAMv1.pdf

1 2 3

```
SRR359063.183484 D042KACXX:3:1202:16105:70173 length=101      163    I
3376    255     101M    =       3426    -51
AATCGACGAAAATCGGTACAAAATCAAATAAAAATAGAAGGAAAATATTCAGCTCGTAAATCCGCAAGTGCGGCA
CGGCTCCGTGGGCGGGGCGCCTCTGG    ?B;4?D?AA???:CBEACEGBFEGGE??<?D??
DEH998DEEGIJ9=BBF=@CGH8AAAA)7A>B,=>C@@@?6?B&05008<A?################
NM:i:5  MD:Z:18T41C17T1T13T6    NH:i:1  XI:i:0  XA:Z:P
```

1	query template name
2	bitwise flag
3	reference sequence name

The Alignment

Sequence Alignment/Map Format Specification:
http://samtools.github.io/hts-specs/SAMv1.pdf

```
   4          5          6
SRR359063.183484 D042KACXX:3:1202:16105:70173 length=101        163    chrI
3376     17       18=1X41=1X17=1X1=1X4=     =     3426     151
AATCGACGAAAATCGGTACAAAATCAAATAAAAATAGAAGGAAAATATTCAGCTCGTAAATCCGCAAGTGCGGCA
CGGCTCCGTG ?B;4?D?AA???:CBEACEGBFEGGE??<?D??
DEH998DEEGIJ9=BBF=@CGH8AAAA)7A>B,=>C@@@?6?B&05008<A?
HI:i:0     NH:i:1  NM:i:4  MD:Z:18T41C17T1T4 RG:Z:A1 YZ:Z:0 XA:Z:P
```

4	1-based leftmost mapping position
5	mapping quality
6	CIGAR string (NOTE THE DIFFERENCE BETWEEN '=' and 'M')

The Alignment

Sequence Alignment/Map Format Specification:
http://samtools.github.io/hts-specs/SAMv1.pdf

```
                                                    7       8       9
SRR359063.183484 D042KACXX:3:1202:16105:70173 length=101        163     chrI
3376    17      18=1X41=1X17=1X1=1X4=    =     3426    151
AATCGACGAAAATCGGTACAAAATCAAATAAAAATAGAAGGAAAATATTCAGCTCGTAAATCCGCAAGTGCGGCA
CGGCTCCGTG ?B;4?D?AA???:CBEACEGBFEGGE??<?D??
DEH998DEEGIJ9=BBF=@CGH8AAAA)7A>B,=>C@@@?6?B&05008<A?
HI:i:0    NH:i:1  NM:i:4  MD:Z:18T41C17T1T4 RG:Z:A1 YZ:Z:0 XA:Z:P
```

7	ref. name of the mate / next segment
8	position of the mate / next segment
9	observed template length

The Alignment

Sequence Alignment/Map Format Specification:
http://samtools.github.io/hts-specs/SAMv1.pdf

10 11

```
SRR359063.183484 D042KACXX:3:1202:16105:70173 length=101        163    chrI
3376    17      18=1X41=1X17=1X1=1X4=    =    3426    151
AATCGACGAAAATCGGTACAAAATCAAATAAAAATAGAAGGAAAATATTCAGCTCGTAAATCCGCAAGTGCGGCA
CGGCTCCGTG ?B;4?D?AA???:CBEACEGBFEGGE??<?D??
DEH998DEEGIJ9=BBF=@CGH8AAAA)7A>B,=>C@@@?6?B&05008<A?
HI:i:0     NH:i:1  NM:i:4  MD:Z:18T41C17T1T4 RG:Z:A1  YZ:Z:0 XA:Z:P XA:Z:P
```

10	segment sequence
11	ASCII of phred-scaled base quality+33

The bitwise flag

Sequence Alignment/Map Format Specification:
http://samtools.github.io/hts-specs/SAMv1.pdf

```
SRR359063.183484 D042KACXX:3:1202:16105:70173 length=101      163    chrI
3376     17       18=1X41=1X17=1X1=1X4=     =     3426     151
AATCGACGAAAATCGGTACAAAATCAAATAAAAATAGAAGGAAAATATTCAGCTCGTAAATCCGCAAGTGCGGCA
CGGCTCCGTG ?B;4?D?AA???:CBEACEGBFEGGE??<?D??
DEH998DEEGIJ9=BBF=@CGH8AAAA)7A>B,=>C@@@?6?B&05008<A?
HI:i:0     NH:i:1  NM:i:4  MD:Z:18T41C17T1T4 RG:Z:A1 YZ:Z:0 XA:Z:P XA:Z:P
```

Bit	Description
0x1	read paired
0x2	read mapped in proper pair
0x4	read unmapped
0x8	mate unmapped
0x10	read reverse strand
0x20	mate reverse strand
0x40	first in pair
0x80	second in pair
0x100	not primary alignment
0x200	read fails platform/vendor quality checks
0x400	read is PCR or optical duplicate
0x800	supplementary alignment

Tip:
https://broadinstitute.github.io/picard/explain-flags.html

The bitwise flag

Sequence Alignment/Map Format Specification:
http://samtools.github.io/hts-specs/SAMv1.pdf

```
$ samtools view

Usage:   samtools view [options] <in.bam>|<in.sam> [region1 [...]]

Options: -b       output BAM
         -h       print header for the SAM output
         -H       print header only (no alignments)
         -S       input is SAM
         -u       uncompressed BAM output (force -b)
         -1       fast compression (force -b)
         -x       output FLAG in HEX (samtools-C specific)
         -X       output FLAG in string (samtools-C specific)
         -c       print only the count of matching records
         -L FILE  output alignments overlapping the input BED FILE [null]
         -t FILE  list of reference names and lengths (force -S) [null]
         -T FILE  reference sequence file (force -S) [null]
         -o FILE  output file name [stdout]
         -R FILE  list of read groups to be outputted [null]
         -f INT   required flag, 0 for unset [0]
         -F INT   filtering flag, 0 for unset [0]
         -q INT   minimum mapping quality [0]
         -l STR   only output reads in library STR [null]
         -r STR   only output reads in read group STR [null]
         -s FLOAT fraction of templates to subsample; integer part as seed [-1]
         -?       longer help
```

The bitwise flag

Sequence Alignment/Map Format Specification:
http://samtools.github.io/hts-specs/SAMv1.pdf

Task
Count all entries mapping to the negative strand, using the bitwise flag

```
$ samtools view -S –f 0x10 SRR359063.segemehl.bam | wc -l
1029380
```

Info
samtools:
-f required flag
-F filtering flag

wc stand for 'word count' and prints a count of newlines, words, and bytes for each input file
-l just returns the number of lines

Bit	Description
0x1	read paired
0x2	read mapped in proper pair
0x4	read unmapped
0x8	mate unmapped
0x10	read reverse strand
0x20	mate reverse strand
0x40	first in pair
0x80	second in pair
0x100	not primary alignment
0x200	read fails platform/vendor quality checks
0x400	read is PCR or optical duplicate
0x800	supplementary alignment

The bitwise flag

Sequence Alignment/Map Format Specification:
http://samtools.github.io/hts-specs/SAMv1.pdf

Task
Count all entries mapping to the positive strand, using the bitwise flag

```
$ samtools view -S –F 0x10 SRR359063.segemehl.bam | wc -l
1047914
```

Info
samtools:
-f required flag
-F filtering flag

wc stand for 'word count' and prints a count of newlines, words, and bytes for each input file
-l just returns the number of lines

Bit	Description
0x1	read paired
0x2	read mapped in proper pair
0x4	read unmapped
0x8	mate unmapped
0x10	read reverse strand
0x20	mate reverse strand
0x40	first in pair
0x80	second in pair
0x100	not primary alignment
0x200	read fails platform/vendor quality checks
0x400	read is PCR or optical duplicate
0x800	supplementary alignment

The extended CIGAR string

Sequence Alignment/Map Format Specification:
http://samtools.github.io/hts-specs/SAMv1.pdf

```
SRR359063.183484 D042KACXX:3:1202:16105:70173 length=101        163    chrI
3376    17      18=1X41=1X17=1X1=1X4=    =    3426    151
AATCGACGAAAATCGGTACAAAATCAAATAAAAATAGAAGGAAAATATTCAGCTCGTAAATCCGCAAGTGCGGCA
CGGCTCCGTG ?B;4?D?AA???:CBEACEGBFEGGE??<?D??
DEH998DEEGIJ9=BBF=@CGH8AAAA)7A>B,=>C@@@?6?B&05008<A?
HI:i:0    NH:i:1  NM:i:4  MD:Z:18T41C17T1T4 RG:Z:A1 YZ:Z:0 XA:Z:P XA:Z:P
```

```
Explanation:

CIGAR string: 28=1X1=1X70=
28M: matches
1X: one mismatch
1=: one match
1X: one mismatch
70=: seventy matches
```

Info

In the CIGAR string the alignment information is stored, i.e. which nucleotides of the read is aligned and where insertions and deletions occur.

The CIGAR string

Sequence Alignment/Map Format Specification:
http://samtools.github.io/hts-specs/SAMv1.pdf

```
$ grep SRR359063.183484 SRR359063.bowtie2.bam | grep 163
SRR359063.183484 163     chrI    3376    7       85M     =       3426
151
AATCGACGAAAATCGGTACAAAATCAAATAAAAATAGAAGGAAAATATTCAGCTCGTAAATCCGCAAGTGCGGCA
CGGCTCCGTG
?B;4?D?AA???:CBEACEGBFEGGE??<?D??DEH998DEEGIJ9=BBF=@CGH8AAAA)7A>B,=>C@@@?6?
B&05008<A?
AS:i:-14 XS:i:-18 XN:i:0   XM:i:4   XO:i:0   XG:i:0   NM:i:4 MD:Z:18T41C17T1T4
YS:i:-10 YT:Z:CP
```

Example:

CIGAR string: 1M1I28M1D3M1I2M
1M: one match or mismatch
1I: one insertion
28M: 28 matches and/or mismatches
1D: one deletion
3M: three matches and/or mismatches
1I: one insertion
2M: two matches and/or mismatches

Info

In the CIGAR string the alignment information is stored, i.e. which nucleotides of the read is aligned and where insertions and deletions occur.

The CIGAR string

Sequence Alignment/Map Format Specification:
http://samtools.github.io/hts-specs/SAMv1.pdf

```
AGTGATGGGA------GGATGTCTCGTCTGTGAGTTACAGCA
||  |  |||  |      ||||||||||||||||||||||||||  MD:Z:3C3T1^GCTCAG26
AG-GCTGGTAGCTCAGGGATGTCTCGTCTGTGAGTTACAGCA


AGTGATGGGA------GGATGTCTCGTCTGTGAGTTACAGCA
||  |  |||  |      ||||||||||||||||||||||||||  MD:Z:3T3C1^GCTTAG26
AG-GTTGGCAGCTTAGGGATGTCTCGTCTGTGAGTTACAGCA
```

Example:

CIGAR string: 2M1I7M6D26M
2M: two matches
1I: one insertion
7M: 7 matches
6D: six deletion
26M: 26 matches

Important

The CIGAR string does not distinguish between a match and a mismatch! To calculate the number of errors for an alignment, the CIGAR string AND the MD-TAG are mandatory.

Important TAGs

Sequence Alignment/Map Format Specification:
http://samtools.github.io/hts-specs/SAMv1.pdf

```
SRR359063.183484 D042KACXX:3:1202:16105:70173 length=101        163    chrI
3376    17      18=1X41=1X17=1X1=1X4=    =    3426    151
AATCGACGAAAATCGGTACAAAATCAAATAAAAATAGAAGGAAAATATTCAGCTCGTAAATCCGCAAGTGCGGCA
CGGCTCCGTG ?B;4?D?AA???:CBEACEGBFEGGE??<?D??
DEH998DEEGIJ9=BBF=@CGH8AAAA)7A>B,=>C@@@?6?B&05008<A?
HI:i:0    NH:i:1  NM:i:4  MD:Z:18T41C17T1T4 RG:Z:A1 YZ:Z:0 XA:Z:P XA:Z:P
```

TAG	description
MD	string for mismatching positions
NH	number of reported alignments (multiple hits)
NM	edit distance to the reference

Important TAGs

Sequence Alignment/Map Format Specification:
http://samtools.github.io/hts-specs/SAMv1.pdf

Task
How many entries show an alignment with no error?

```
$ samtools view -S SRR359063.segemehl.bam | grep "NH:i:\d+" | wc -l
1775001
```

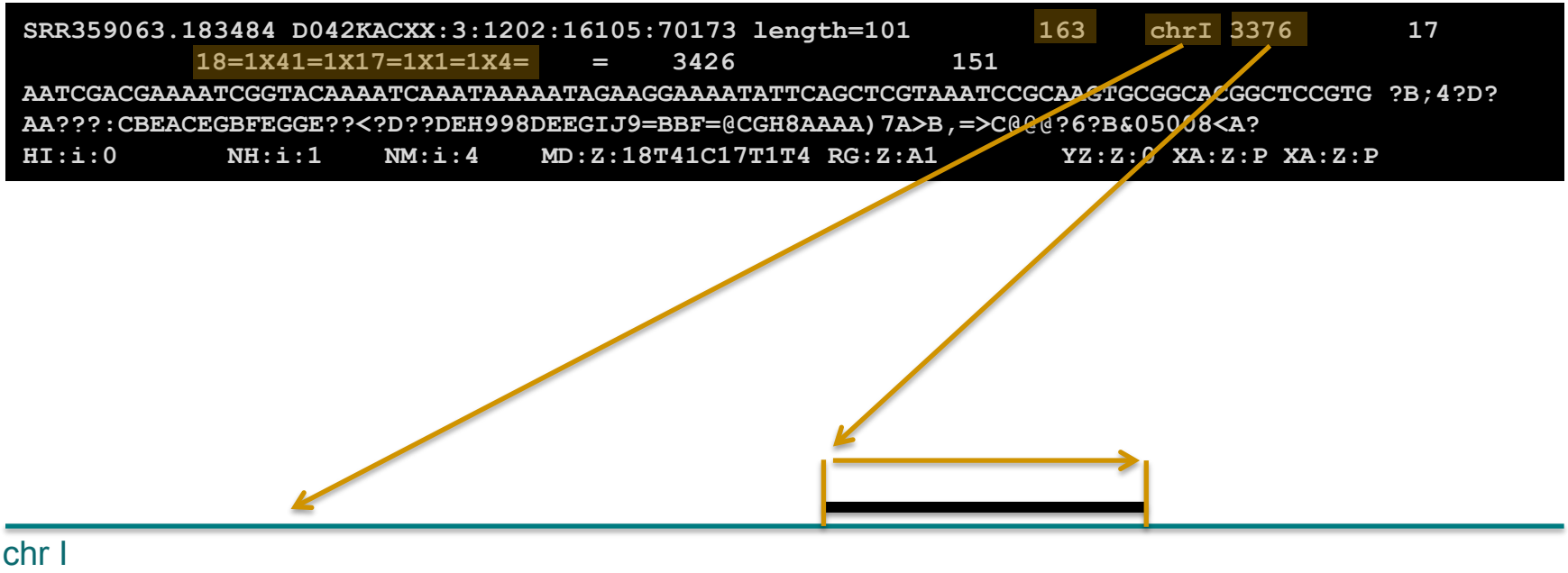
Task
How many reads map to only one single position (no multiple hits)?

```
$ samtools view -S SRR359063.segemehl.bam | grep -oP "NH:i:\d+" |
  cut -d":" -f3 | sort -n | uniq -c | head -n1
1820949  1
```

```
$ samtools view -S SRR359063.segemehl.bam | grep  "NH:i:1" | wc -l
1820949  1
```

The Alignments - How to extract the genomic position

```
SRR359063.183484 D042KACXX:3:1202:16105:70173 length=101      163     chrI 3376       17
          18=1X41=1X17=1X1=1X4=     =     3426              151
AATCGACGAAAATCGGTACAAAATCAAATAAAAATAGAAGGAAAATATTCAGCTCGTAAATCCGCAAGTGCGGCACGGCTCCGTG ?B;4?D?
AA???:CBEACEGBFEGGE??<?D??DEH998DEEGIJ9=BBF=@CGH8AAAA)7A>B,=>C@@@?6?B&05008<A?
HI:i:0        NH:i:1     NM:i:4     MD:Z:18T41C17T1T4 RG:Z:A1         YZ:Z:0 XA:Z:P XA:Z:P
```



chromosome = column 3
start position = column 4
end position = start position + length from CIGAR string (column 6)
strand = stored in bitwise flag (column 2)

BAM file

Task
Sort the BAM file

```
$ samtools sort SRR359063.segemehl.bam SRR359063.segemehl.sorted
$ mv SRR359063.segemehl.sorted.bam SRR359063.segemehl.bam
```

Info
Since all information are stored in the sorted BAM file, the unsorted bam file can be overwritten (and thus deleted) with the sorted one

Task
Create an index of the BAM file

```
$ samtools index SRR359063.segemehl.bam
```

Mapping Statistics

Mappable reads

Task
Find out how many mates were mappable.
There are 930,982 fragments in the FASTQ files (1,861,964 mates)

```
$ samtools view -F 0x4 SRR359063.segemehl.bam | wc -l
2058923
```

Question
Does the result makes sense?
>110% of the mates were mappable?
Where is the problem?

Mappable reads

Explanation
Mates with multiple hits were counted several times!

Task
Count the mates in a correct way.
There are 930,982 fragments in the FASTQ files (1,861,964 mates)

```
$ samtools view -F 0x4 -f 0x40 SRR359063.segemehl.bam | cut -f1 |
sort | uniq | wc -l
917644
$ samtools view -F 0x4 -f 0x80 SRR359063.segemehl.bam | cut -f1 |
sort | uniq |wc -l
916987
```

Info
-f 0x40 and **-f 0x80** return mate 1 and mate 2 respectively
cut -f1 returns the first columns of all entries (the read Name/ID)
sort sorts the names
uniq merges all identical entries in a block to a single entry

Mappable reads

Explanation
Mates with multiple hits were counted several times!

Task
Count the mates in a correct way.
There are 930,982 fragments in the FASTQ files (1,861,964 mates)

```
$ samtools view -F 0x4 -f 0x40 SRR359063.segemehl.sorted.bam | cut
-f1 | sort | uniq | wc -l
917644
$ samtools view -F 0x4 -f 0x80 SRR359063.segemehl.sorted.bam | cut
-f1 | sort | uniq |wc -l
916987
```

>98 % of the mates were mappable

Info
If less than 70% of the mates are mappable, there might be a problem with the sequencing experiment!

Mappable reads

Task
Another way to get alignment statistics is using bamtools

```
$ bamtools stats –in SRR359063.segemehl.bam > SRR359063.segemehl.stats
```

Question
What is the output of bamtools stats?

Mapping statistics

Further Statistics

Try to find out:

- How many reads were mappable?
- How many mates1 reads were mappable?
- How many mates2 reads were mappable?
- How many reads/mates did map with no error?
- How many reads did map uniquely and how many multiple times?
- What is the read with the highest edit distance?

Results

	mappable reads			mappable fragments	
tool	**# mate1**	**# mate2**	**% mates**	**# fragments**	**% fragments**
segemehl	908,901	908,122	97.59 %	613,892	65.94 %
STAR	912,175	912,078	97.79 %	912,040	97.96 %
Bowtie2	746,513	747,793	80.25 %	612,332	65.77 %
BWA	747,485	746,118	80.22 %	707,385	75.89 %

	# mates that map			highest edit distance
tool	**with no errors**	**unique**	**multiple**	
segemehl	76.68 %	94.36 %	5.64 %	11
STAR	80.64 %	93.54 %	6.46 %	10
Bowtie2	58.30 %	100.00 %	0.00 %	23
BWA	67.91 %	100.00 %	0.00 %	58

Visualization

Fasta index of the genome

Task
Create an index on the C. elegans genome

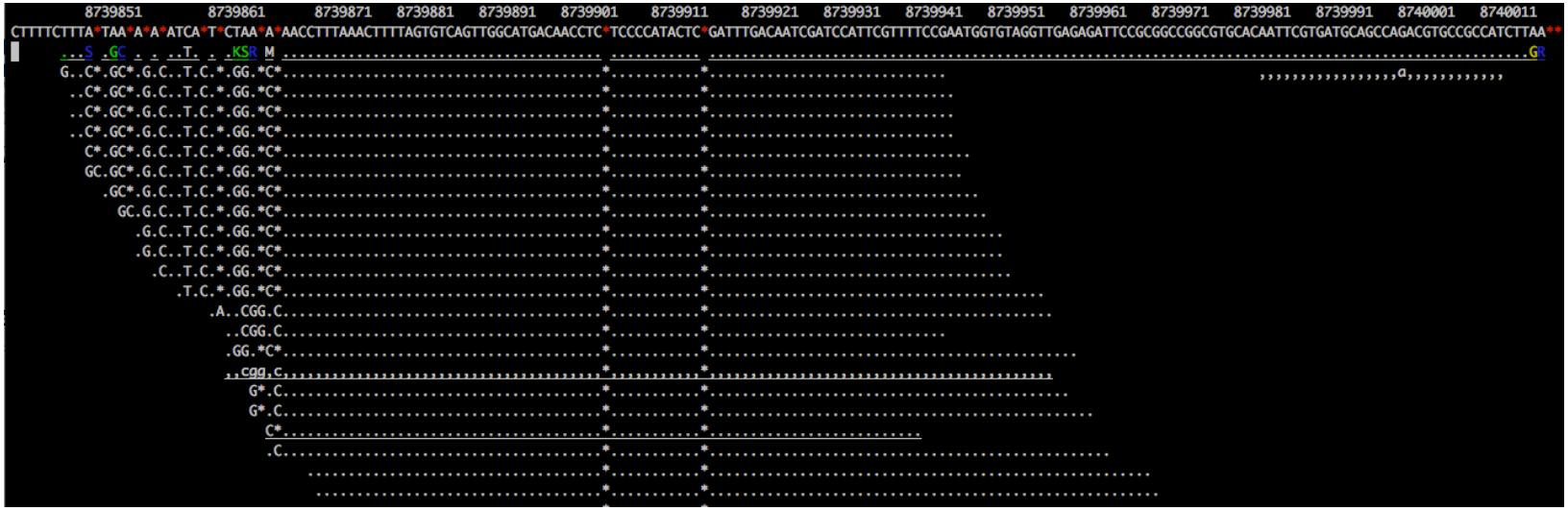
```
$ samtools faidx ../genome/ce10.fa
```

Samtools tview

Task
Take a first look at the BAM file using the samtools tviewer.

```
$ samtools tview SRR359063.segemehl.bam ../genome/ce10.fa
```

Press ‘g’ to open the GoTo-window and enter e.g. chrIV:8739842
With the left and right arrow you can move around.



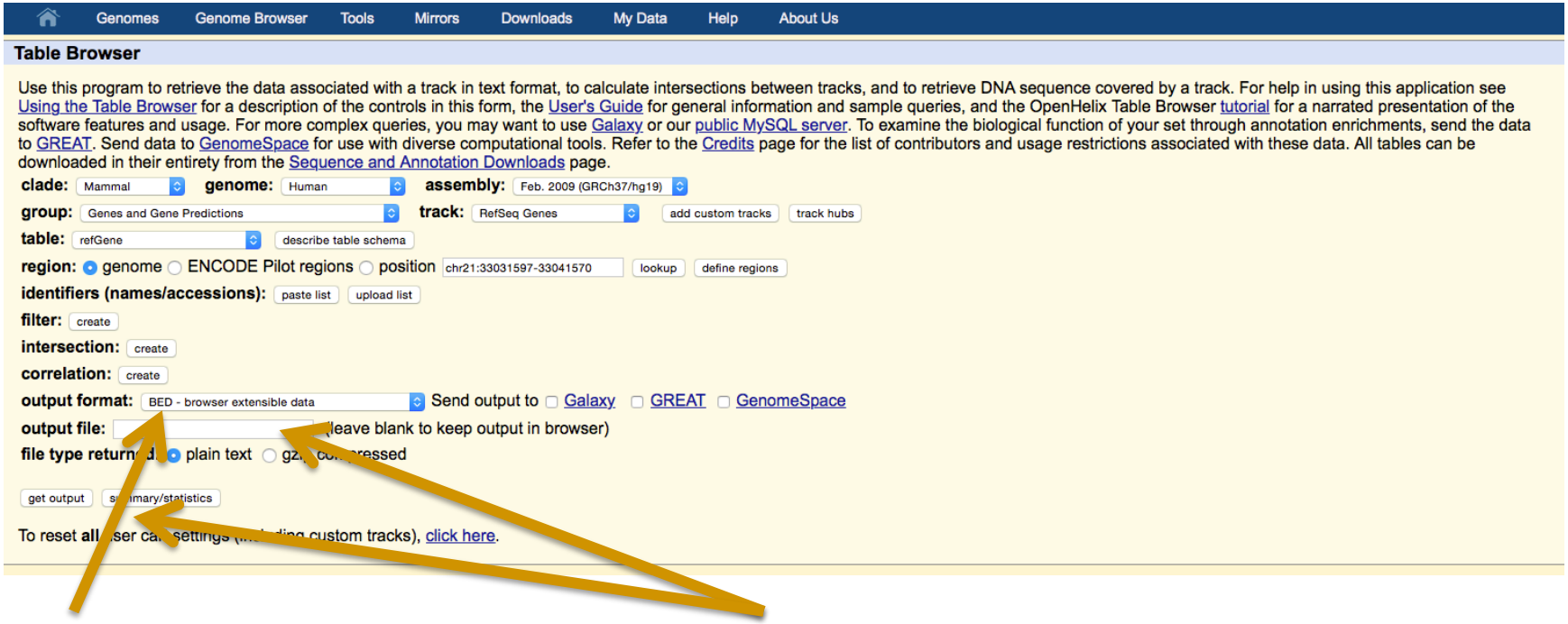
UCSC

Task
Download the ce10 RefSeq gene annotations from UCSC using the 'Table Browser'

2. Select the ce10 assembly:
group: Nematode
genome: C. elegans
assembly: Oct. 2010 (WS220/ce10)

UCSC

Task
Download the ce10 RefSeq gene annotations from UCSC using the ‘Table Browser’



3. Select as output format ‘GTF format’

4. Name your output file ‘RefSeq_genes_ce10.gtf’
5. Click on ‘get output’

UCSC Genome Browser

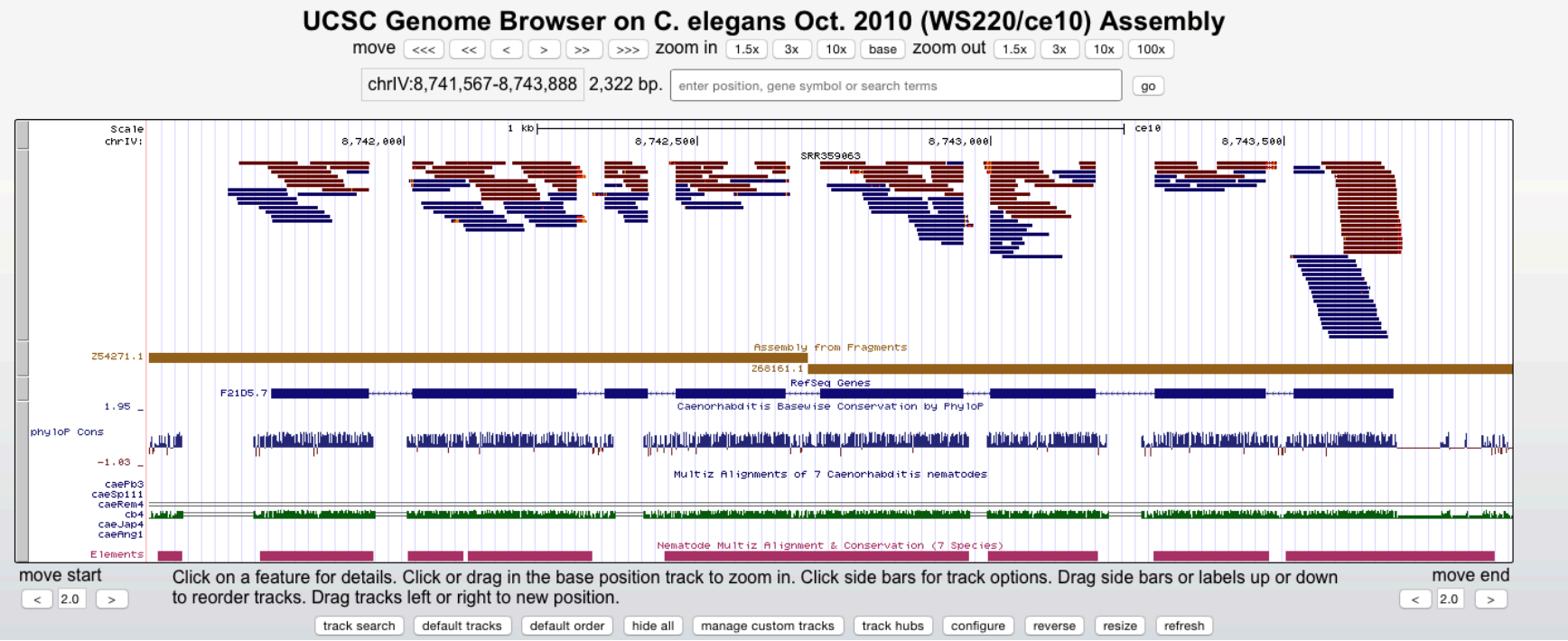
Task
Visualize BAM file at UCSC Genome Browser without uploading the whole file

1. sort and index your BAM file using SAMtools
2. move the BAM file together with its index file (.bam.bai) to a publicly accessible http, https, or ftp location on your server
3. at UCSC Genome Browser select ‚add custom tracks' and construct a track line containing the link to your BAM file together with its name:
   ‚track type=bam name="SRR359063" bigDataUrl=http://www.ecseq.com/ngs2014/SRR359063.bam'
4. push submit and take a look at your locus of interest using the UCSC Genome Browser

see also: http://genome.ucsc.edu/goldenPath/help/bam.html

UCSC Genome Browser

Task
Visualize BAM file at UCSC Genome Browser without uploading the whole file



Thank you very much for your attention!!

UUSMB
UNIDAD UNIVERSITARIA DE SECUENCIACIÓN MASIVA Y BIOINFORMÁTICA