CS-E5875 High-Throughput Bioinformatics
High-throughput sequencing and sequence alignment

Harri Lähdesmäki

Department of Computer Science
Aalto University

October 27, 2023

Contents

- High-throughput sequencing technologies
- Quality control
- Sequencing data alignment

Human genome

- Human genome is located in the nucleus of each cell and contains the genetic information of an individual
- Each individual has two copies of the genome, inherited from both parents
- Genome consists of 23 chromosome pairs
- Genetic information is contained in the chemical compound called deoxyribonucleic acid (DNA)

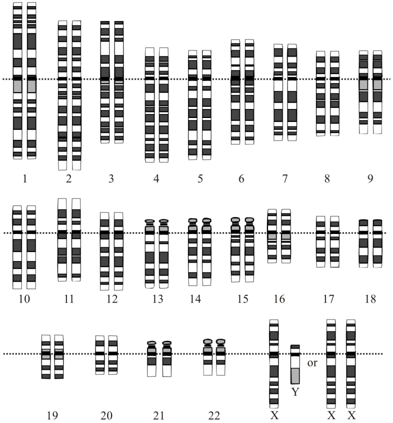


Figure from Wikipedia

Human genome

- DNA is a double-stranded molecule with each strand being a linear sequence of nucleotides
- A nucleotide consists of a phosphate group, sugar, and nucleoside
- A nucleoside is a nitrogenous base connected to a deoxyribose sugar
- There are four different nucleotides depending on the nucleoside: adenine (A), cytosine (C), guanine (G), thymine (T)
- The nucleotides have a specific base pairing in double-stranded DNA:
  - Adenine pairs with thymine
  - Cytosine pairs with guanine
- Total length: about 3 billion nucleotides

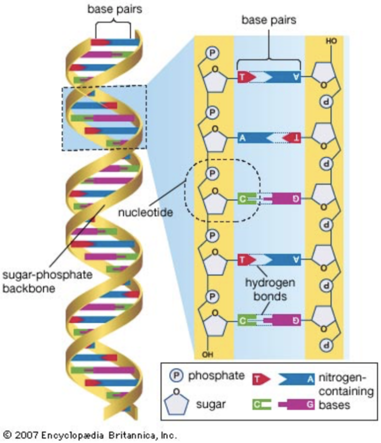


Figure from Wikipedia

High-throughput sequencing technologies

Terminology: the current best sequencing technologies are called interchangeably as

- High-throughput sequencing (HTS)
- Next generation sequencing (NGS)

What it does: time- and cost-efficient sequence determination (DNA, RNA)

- Input: biological sample
  - Biological sample contains a (large) collection of cells
- Output: short nucleotide sequences (also called “reads”)
  - DNA (or RNA) content of the input sample in digital format
  - Not the original long DNA/RNA sequences, but lots of short DNA/RNA sequences (long DNA/RNA sequences are randomly fragmented) from randomly selected cells in the biological input sample
  - This is typically called “bulk” DNA or RNA-sequencing: one is not able to determine that from which cell any of the measured DNA/RNA fragment comes from
  - We will discuss single cell technologies later

NGS technologies

- Sanger sequencing
  - The grand old technology
- Illumina
  - Illumina is currently by far the most commonly used

  → We will focus on Illumina sequencing technology
- Applied Life sciences: SOLiD
- Pacific Biosciences: PacBio
- Ion Torrent sequencing
- Nanopore
  - A technology to measure an entire (long) DNA molecule

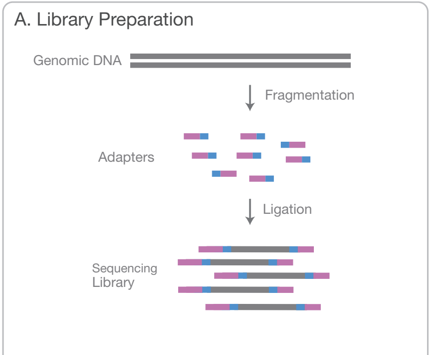
Illumina: basics of NGS chemistry

- Three main steps
  1. Library preparation
  2. Cluster amplification (bridge amplification)
  3. Sequencing by synthesis
- Next: cartoon guide to NGS chemistry

Illumina: basics of NGS chemistry

Step 1: Library preparation

- Isolate DNA (or RNA) from cells in a biological sample
- Sequencing library is prepared by random fragmentation of the DNA (or cDNA sample), followed by 5' and 3' adapter ligation
- Single stranded molecules
- Adapter-ligated fragments are then PCR amplified and gel purified



NGS library is prepared by fragmenting a gDNA sample and ligating specialized adapters to both fragment ends.

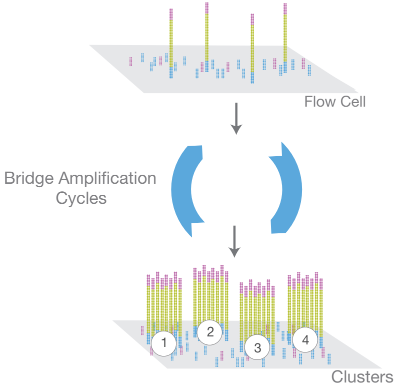
Figure from `http://www.illumina.com/content/dam/illumina-marketing/documents/products/illumina_sequencing_introduction.pdf`

Illumina: basics of NGS chemistry

Step 2: Cluster amplification

- The library is loaded into a flow cell where fragments are captured on a lawn of surface-bound oligos that are complementary to the library adapters
- Each fragment is then amplified into distinct, clonal clusters through bridge amplification
- When cluster generation is complete, the templates (i.e., clonal clusters) are ready for sequencing
- A flow cell contains millions of clonal cluster

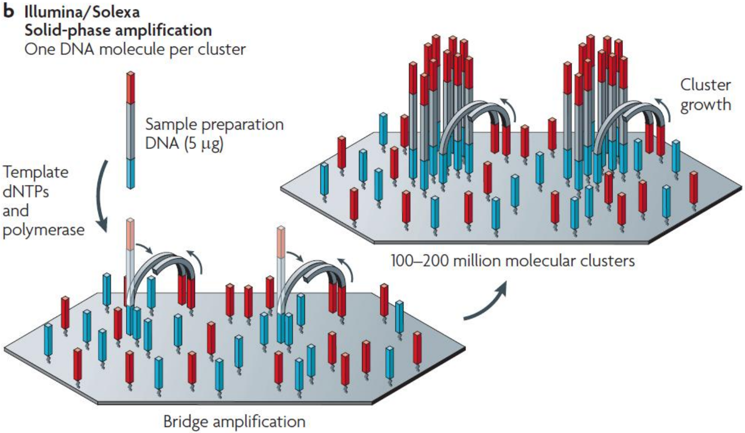
B. Cluster Amplification



Library is loaded into a flow cell and the fragments hybridize to the flow cell surface. Each bound fragment is amplified into a clonal cluster through bridge amplification.

Illumina: clonally amplified templates

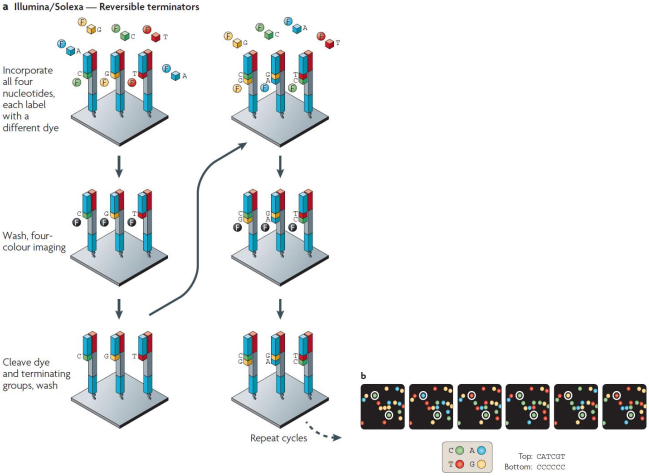
- Step 2: Cluster amplification


b Illumina/Solexa
Solid-phase amplification
One DNA molecule per cluster
Sample preparation DNA (5 µg)
Template dNTPs and polymerase
Bridge amplification
100–200 million molecular clusters
Cluster growth

Illumina: basics of NGS chemistry

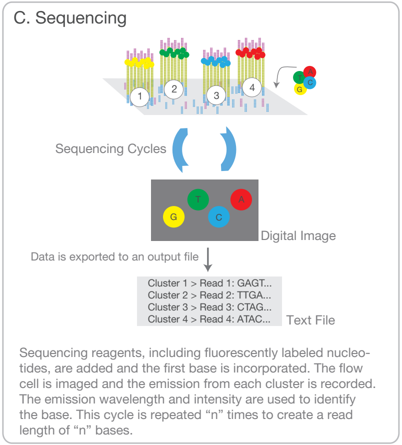
Step 3: Sequencing

- Illumina uses a terminator-based method that detects single bases as they are incorporated into DNA template strands
- Each nucleotide (dNTP) has a different fluorescent dye
- All 4 reversible terminator-bound dNTPs are present during each sequencing cycle
- Natural competition minimizes incorporation bias and greatly reduces raw error rates compared to other technologies
- Laser excitation and imaging of the emitted flourescence


a Illumina/Solexa — Reversible terminators
Incorporate all four nucleotides, each label with a different dye
Wash, four-colour imaging
Cleave dye and terminating groups, wash
Repeat cycles
b
C A
T G
Top: CATCGT
Bottom: CCCCCC

Illumina: basics of NGS chemistry

- Step 3: Sequencing



C. Sequencing

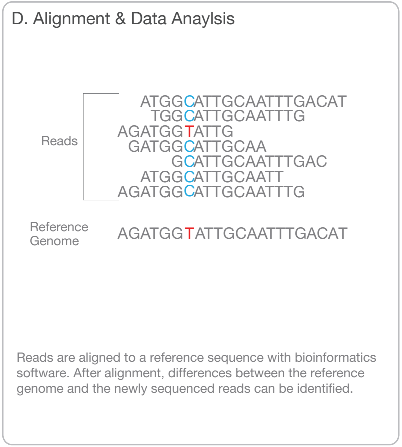
Cluster 1 > Read 1: GAGT...
Cluster 2 > Read 2: TTGA...
Cluster 3 > Read 3: CTAG...
Cluster 4 > Read 4: ATAC...

Sequencing reagents, including fluorescently labeled nucleotides, are added and the first base is incorporated. The flow cell is imaged and the emission from each cluster is recorded. The emission wavelength and intensity are used to identify the base. This cycle is repeated "n" times to create a read length of "n" bases.

Figure from http://www.illumina.com/content/dam/illumina-marketing/documents/products/illumina_sequencing_introduction.pdf

Illumina: basics of NGS chemistry

- Step 4: Data analysis



D. Alignment & Data Anaylsis

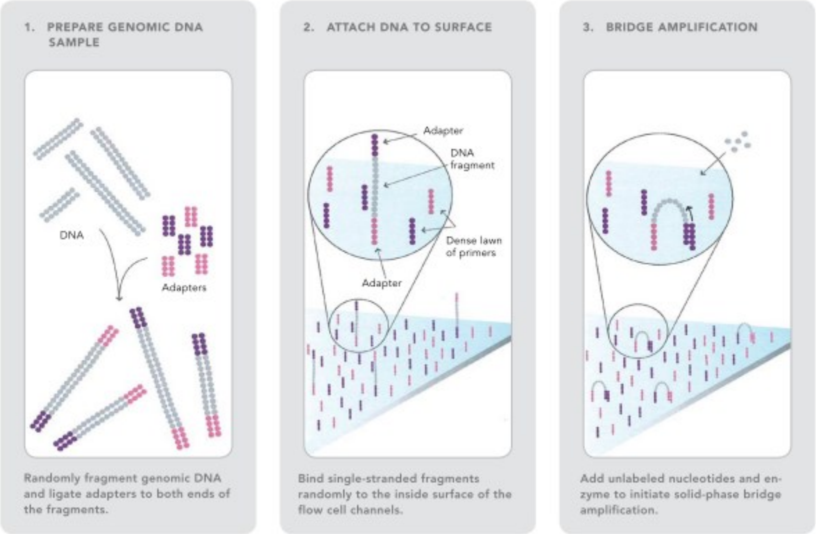
```
  ATGGCATTGCAATTTGACAT
   TGGCATTGCAATTTG
AGATGGTATTG
 GATGGCATTGCAA
     GCATTGCAATTTGAC
  ATGGCATTGCAATT
AGATGGCATTGCAATTTG

AGATGGTATTGCAATTTGACAT
```

Reads are aligned to a reference sequence with bioinformatics software. After alignment, differences between the reference genome and the newly sequenced reads can be identified.

Figure from http://www.illumina.com/content/dam/illumina-marketing/documents/products/illumina_sequencing_introduction.pdf

Illumina Genome Analyzer: bridge amplification



1. PREPARE GENOMIC DNA SAMPLE

Randomly fragment genomic DNA and ligate adapters to both ends of the fragments.

2. ATTACH DNA TO SURFACE

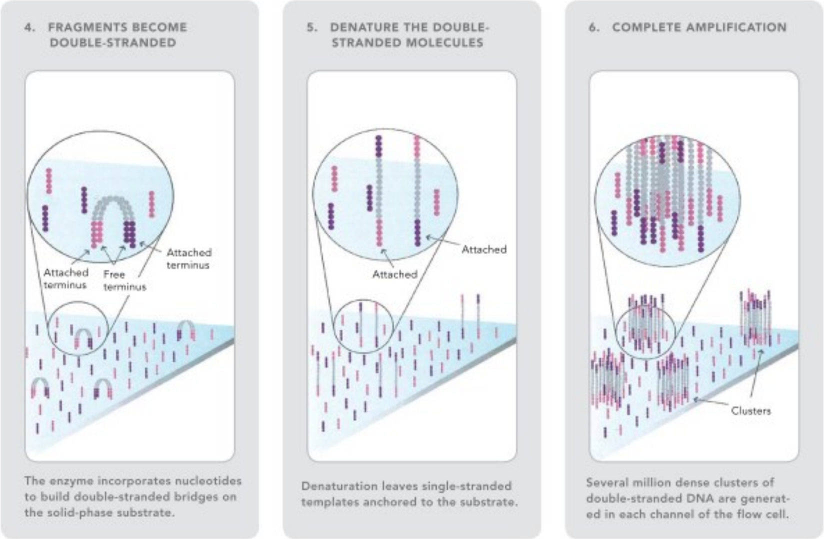
Bind single-stranded fragments randomly to the inside surface of the flow cell channels.

3. BRIDGE AMPLIFICATION

Add unlabeled nucleotides and enzyme to initiate solid-phase bridge amplification.

Figure from http://www.illumina.com

Illumina Genome Analyzer: bridge amplification



4. FRAGMENTS BECOME DOUBLE-STRANDED

The enzyme incorporates nucleotides to build double-stranded bridges on the solid-phase substrate.

5. DENATURE THE DOUBLE-STRANDED MOLECULES

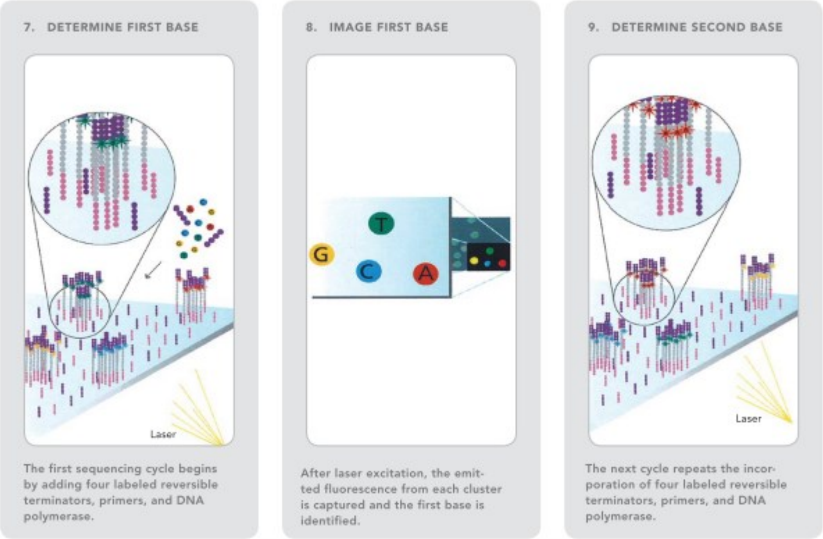
Denaturation leaves single-stranded templates anchored to the substrate.

6. COMPLETE AMPLIFICATION

Several million dense clusters of double-stranded DNA are generated in each channel of the flow cell.

Figure from http://www.illumina.com

Illumina Genome Analyzer: sequencing by synthesis



7. DETERMINE FIRST BASE

The first sequencing cycle begins by adding four labeled reversible terminators, primers, and DNA polymerase.

8. IMAGE FIRST BASE

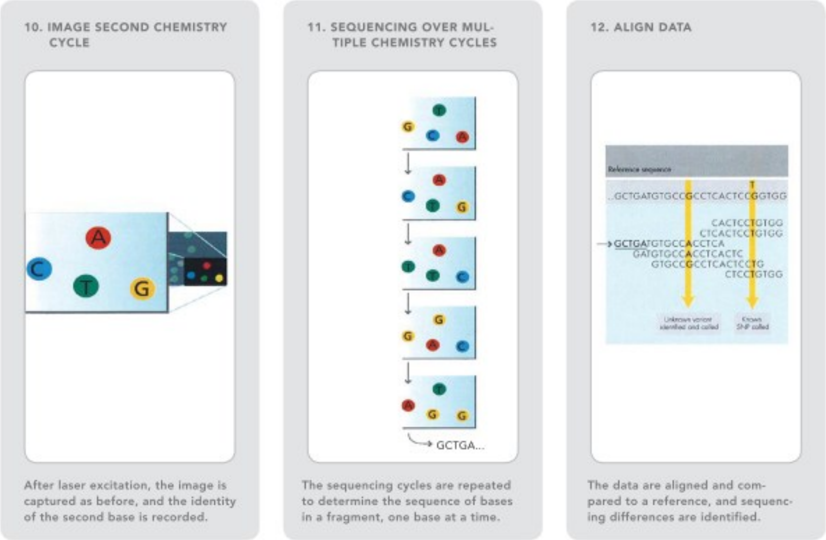
After laser excitation, the emitted fluorescence from each cluster is captured and the first base is identified.

9. DETERMINE SECOND BASE

The next cycle repeats the incorporation of four labeled reversible terminators, primers, and DNA polymerase.

Figure from http://www.illumina.com

Illumina Genome Analyzer: sequencing by synthesis



10. IMAGE SECOND CHEMISTRY CYCLE

After laser excitation, the image is captured as before, and the identity of the second base is recorded.

11. SEQUENCING OVER MULTIPLE CHEMISTRY CYCLES

The sequencing cycles are repeated to determine the sequence of bases in a fragment, one base at a time.

12. ALIGN DATA

The data are aligned and compared to a reference, and sequencing differences are identified.

Figure from http://www.illumina.com

Differences between NGS technologies

- Template preparation
  - Amplified templates from single DNA molecule (454, SOLiD, Illumina)
  - Single DNA-molecule templates (PacBio, Nanopore)

Differences between NGS technologies

- Template preparation
  - Amplified templates from single DNA molecule (454, SOLiD, Illumina)
  - Single DNA-molecule templates (PacBio, Nanopore)
- Sequencing
  - Cyclic reversible termination (Illumina)
  - Single-nucleotide addition (454)
  - Single DNA-molecule templates (PacBio)
  - Real-time sequencing (PacBio)
  - Sequencing by ligation (SOLiD)
  - Physical properties (Nanopore)

Differences between NGS technologies

- Template preparation
  - Amplified templates from single DNA molecule (454, SOLiD, Illumina)
  - Single DNA-molecule templates (PacBio, Nanopore)
- Sequencing
  - Cyclic reversible termination (Illumina)
  - Single-nucleotide addition (454)
  - Single DNA-molecule templates (PacBio)
  - Real-time sequencing (PacBio)
  - Sequencing by ligation (SOLiD)
  - Physical properties (Nanopore)
- Imaging
  - Four-color imaging (Illumina, SOLiD, PacBio)
  - Bioluminescense (454)
- Data analysis

Differences between NGS technologies

- Differences
  - Sequencing read length
  - Raw read accuracy:
    - Substitutions, insertions, deletions
    - Error models (quality scores)
  - Speed: nucleotide bases/second
  - Costs:
    - Nucleotide bases/€
    - Equipment cost/maintanence

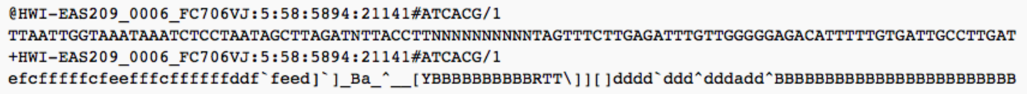
- Illumina, SOLiD
  - De novo assembly
  - Resequencing
  - RNA-seq
  - Non coding RNAs
  - ChIP-seq
  - etc.
- Sanger, PacBio, Nanopore
  - De novo assembly
  - Detection of large structural variants

Contents

- High-throughput sequencing technologies
- Quality control
- Sequencing data alignment

NGS sequencing data formats: FASTQ

- FASTQ format is the most commonly used text-based format for storing both a nucleotide sequence and its corresponding quality scores
  - Line 1: sequence identifier containing: instrument name, flowcell lane, tile number, coordinates in a tile, etc.
  - Line 2: The raw sequence letters: A, C, G, T, or N
  - Line 3: +[sequence identifier]
  - Line 4: The quality values for the sequence in Line 2
- An example



```
@HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1
TTAATTGGTAAATAAATCTCCTAATAGCTTAGATNTTACCTTNNNNNNNNNNTAGTTTCTTGAGATTTGTTGGGGAGACATTTTTGTGATTGCCTTGAT
+HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1
efcfffffcfeefffcffffffddf`feed]`]_Ba_^__[YBBBBBBBBBBRTT\]][]dddd`ddd^dddadd^BBBBBBBBBBBBBBBBBBBBBBBB
```

Fastq quality scores

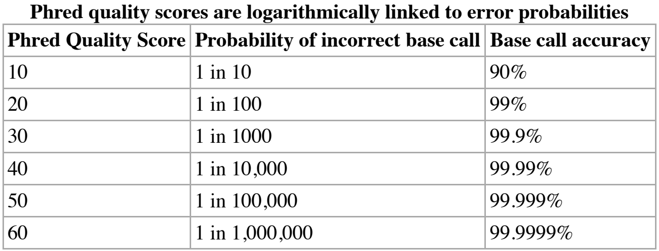
- Each nucleotide that has been sequenced is associated with a quality score Q
- Q is a function of the probability P that the corresponding base call is incorrect

$$Q = -10 \log_{10} P$$

- This probability is estimated during the sequencing run (affected by e.g. the quality of imaging, etc.)
- The score Q is mapped to an integer to reduce space
  - E.g. using ASCII from 33 to 126 (encoding varies between versions)
- This is called the Phred quality score

Fastq quality scores

- The Phred quality score table



Phred quality scores are logarithmically linked to error probabilities

Phred Quality Score	Probability of incorrect base call	Base call accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9%
40	1 in 10,000	99.99%
50	1 in 100,000	99.999%
60	1 in 1,000,000	99.9999%

Figure from `https://en.wikipedia.org/wiki/FASTQ_format`

Quality control (QC) for sequencing data

- NGS experiments can generate hundreds of millions or billions of sequences in a single run
- Before analysing the data and drawing any biological conclusions, one should perform some quality control checks to ensure that
  - the raw data looks good overall, and
  - there are no problems or biases in the data
- Most sequencing facilities will generate a quality control report as part of their analysis pipeline
  - These are typically limited to identifying and reporting overall problems that may have happened during the laboratory work or sequencing experiment

Quality control (QC) for sequencing data

- NGS experiments can generate hundreds of millions or billions of sequences in a single run
- Before analysing the data and drawing any biological conclusions, one should perform some quality control checks to ensure that
  - the raw data looks good overall, and
  - there are no problems or biases in the data
- Most sequencing facilities will generate a quality control report as part of their analysis pipeline
  - These are typically limited to identifying and reporting overall problems that may have happened during the laboratory work or sequencing experiment
- Many tools are available for more comprehensive sequence-level QC analysis that aim to identify problems originating either in the sequencer or in the starting library material
- We will look at FastQC tool (Andrews, 2010)
- Note: quality control analysis depends on the particular NGS protocol: genome sequencing (assumed here), RNA-seq, methylation sequencing, etc.

Quality control (QC) for sequencing data

- Quality control criteria include e.g.
  1. Per base quality
  2. Per sequence quality
  3. Per base content
  4. Sequence duplication levels
  5. Overrepresented sequences

1. Per base quality

- The accuracy of reading the nucleotides tends to decrease towards the end of the sequences
- It is useful to quantify if the quality scores decrease more than expected or to decide where to trim / cut the sequences
- Sequencing machine can also output N read-out to indicate that the sequencing machine was not able to make decision of the base
- Boxplots of quality scores Q for each base position across all reads
- Example of good quality data

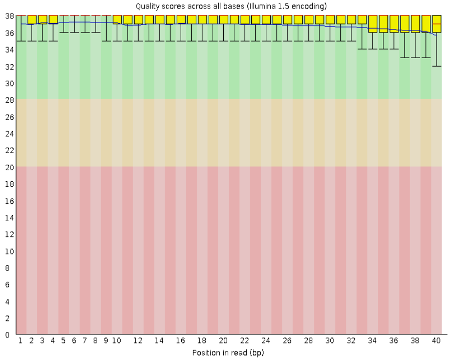


Figure from http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

1. Per base quality

- An example of low quality data
- A solution: keep only the first k nucleotides for each sequence and cut out nucleotides at positions $[k+1, \ldots, n]$, where n is the sequencing read length

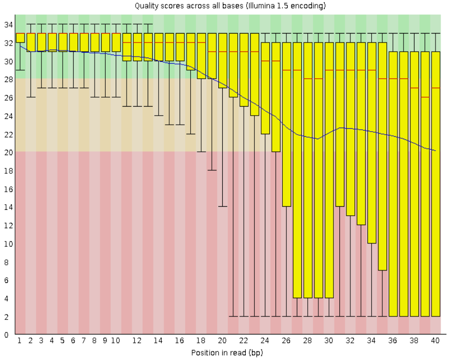


Figure from `http://www.bioinformatics.babraham.ac.uk/projects/fastqc/`

2. Per sequence quality

- Quantify if a sequence has overall lower quality using the mean quality score
- Low quality sequences should represent only a small percentage of the total sequencing data
- Example of good quality data

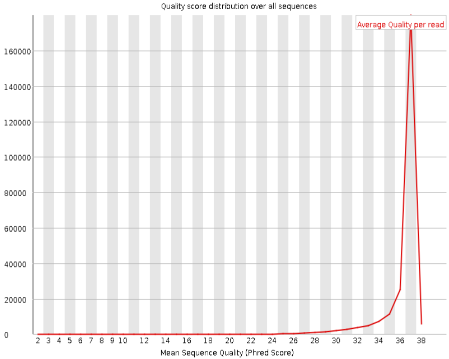


Figure from `http://www.bioinformatics.babraham.ac.uk/projects/fastqc/`

2. Per sequence quality

- An example of a sample where a subset of the sequences have lower quality
- A solution: remove sequencing reads that have mean quality score below a threshold and continue the analysis with the good quality sequences

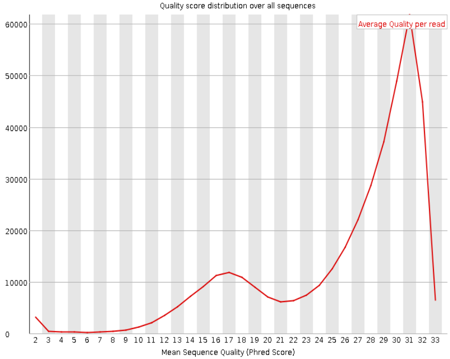


Figure from `http://www.bioinformatics.babraham.ac.uk/projects/fastqc/`

3. Per base content

- In a good quality random DNA library you would expect to see little or no variation in sequence/GC/N content across the bases
- E.g.: the relative amount of each base should reflect the overall amount of these bases in a genome
  - Deviations from genome-wide averages indicates lower quality e.g. due to biases in sample preparation
  - Strong biases at selected bases can indicate e.g. a contamination of overrepresented sequences, power outage, etc.
- Some variation at the first nucleotides is not so uncommon though
- An example of a good quality library

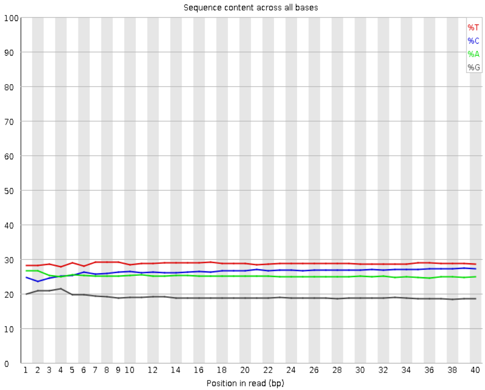


Figure from http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

3. Per base content

- An example of a good quality library based on GC content
- Solutions for a low quality library depends on the application: in the worst case ignore the data altogether

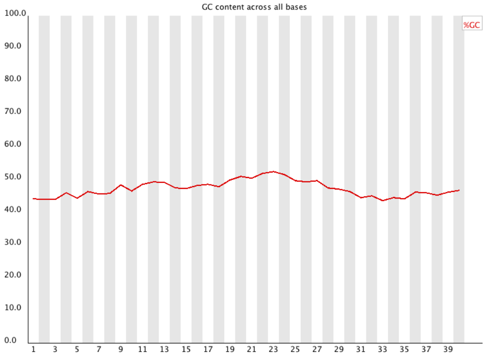


Figure from `http://www.bioinformatics.babraham.ac.uk/projects/fastqc/`

4. Sequence duplication levels

- In a complex (complex=good) DNA library most sequences will occur only once
  - This applies e.g. to DNA genome sequencing
  - Think about randomly fragmenting the 3 billion long human genome into fragments and sequencing those DNA fragments
  - It is realistic to assume that such fragments from different cells / chromosome copies will be mostly different

  → Unlikely to measure the same DNA fragment more than once
  - Note: there are also sequencing applications where this assumption does not hold
- A high level of sequence duplication may indicate some kind of enrichment bias (e.g. PCR over-amplification during library preparation).
- Plot the relative number of sequences with different degrees of duplication

4. Sequence duplication levels

- An example of a library with suspicious duplication levels
- Note: this looks at the sequence similarity only, not the location where they come from
  - For example: two reads may have nucleotide read errors and thus are non-identical, but map to the same location in a genome

  → Detection of duplicate reads can also be done after read alignment
- A solution: keep only one copy of each duplicated read

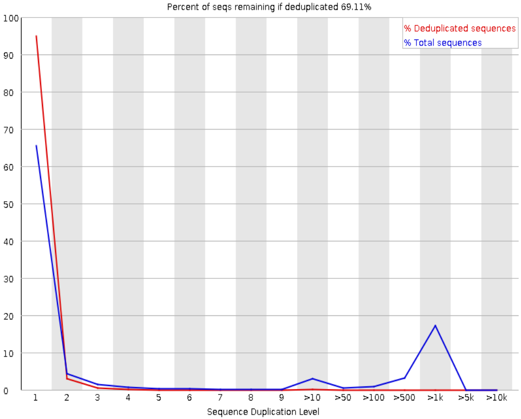
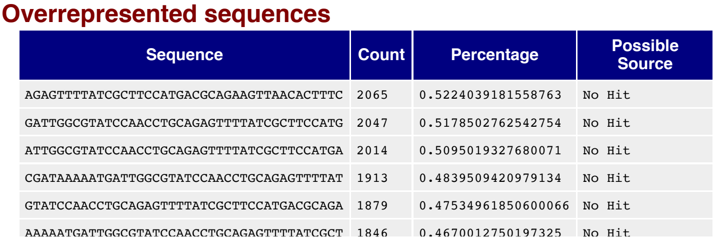


Figure from http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

5. Overrepresented sequences

- A good quality, complex library will contain a diverse set of sequences
- No individual sequence should be overrepresented
- Finding a few overrepresented sequences may indicate that the library is contaminated
- These overrepresented sequences may also correspond to adapter sequences, in which case there is typically less reason to worry
- An example of a library that contains overrepresented sequences (>0.1%)



Overrepresented sequences

Sequence	Count	Percentage	Possible Source
AGAGTTTTATCGCTTCCATGACGCAGAAGTTAACACTTTC	2065	0.5224039181558763	No Hit
GATTGGCGTATCCAACCTGCAGAGTTTTATCGCTTCCATG	2047	0.5178502762542754	No Hit
ATTGGCGTATCCAACCTGCAGAGTTTTATCGCTTCCATGA	2014	0.5095019327680071	No Hit
CGATAAAAATGATTGGCGTATCCAACCTGCAGAGTTTTAT	1913	0.4839509420979134	No Hit
GTATCCAACCTGCAGAGTTTTATCGCTTCCATGACGCAGA	1879	0.47534961850600066	No Hit
AAAAATGATTGGCGTATCCAACCTGCAGAGTTTTATCGCT	1846	0.4670012750197325	No Hit

Figure from http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Contents

- High-throughput sequencing technologies
- Quality control
- Sequencing data alignment

Sequencing data

- Current sequencing technologies produce huge amounts of short sequencing read data
  - The obtained sequences are typically short ∼25-250bp
  - A sequencing run produces batches of, say, 10M to 1B sequences
- Challenge: make sense of all these short reads
- Some obvious questions:
  - From which genomic positions / chromosome in a given reference genome the reads originate?
  - Can you construct the original "full length" genome from the measured short sequencing reads?
    - De novo assembly of a genome
- For a primer in short read data alignment, see (Trapnell & Salzberg, 2009)

Short read alignment

Many of the HTS sequencing applications involve a task of finding the origin of a short sequence in a large reference sequence

- E.g. identify the location of a 50bp read in a $\sim$100,000,000bp human chromosome or in any of the chromosomes with total size of $\sim$3,000,000,000bp
- Dynamic programming based local sequence alignment (Smith-Waterman algorithm) would provide an optimal sequence alignment (assuming substitution and gap-penalties)
- Asymptotic time complexity for aligning a single short sequence is $O(nm)$ for sequences of length n (reference genome) and m (a short read)

Short read alignment

Many of the HTS sequencing applications involve a task of finding the origin of a short sequence in a large reference sequence

- E.g. identify the location of a 50bp read in a ∼100,000,000bp human chromosome or in any of the chromosomes with total size of ∼3,000,000,000bp
- Dynamic programming based local sequence alignment (Smith-Waterman algorithm) would provide an optimal sequence alignment (assuming substitution and gap-penalties)
- Asymptotic time complexity for aligning a single short sequence is $O(nm)$ for sequences of length n (reference genome) and m (a short read)

Challenges in short read alignment

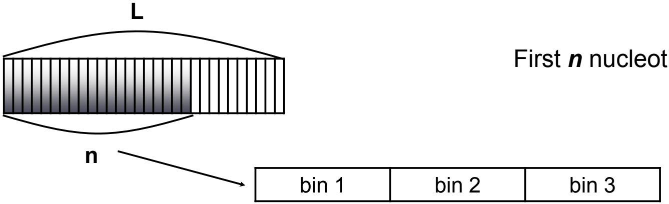
- The volume of data / length of the reference sequence is often so large that performance is a real concern
- Although many of the currently available sequencing machines are relatively accurate, short reads contains errors, i.e., some bases are read incorrectly (technical noise)
- Similarly, variation between individuals cause a mismatch between the obtained reads and a single reference genome (biological "noise")
- Parts of e.g. the human genome is repetitive which causes mapping ambiguity
  - Reads are short; if they fall in repetitive areas, it's hard to know where they truly map

Short read alignment

- Classical sequence alignment methods, such as Smith-Waterman and BLAST algorithms, are too slow and faster alignment methods have been developed
- The most important ingredient is an index: a look-up structure to rapidly find short sub-sequences
  - Hash table based methods
  - Burrows-Wheeler transform
- An index is constructed
  - For the reads
  - For the reference genome(s)
  - Or for both

Hash table aligners

- Extensions of the idea of BLAST: seed-and-extend
- Two steps: indexing and alignment
- E.g. Maq tool (Li et al, 2008)
- Indexing for sequencing reads: divide short sequencing reads of length L into bins based on their first n nucleotides

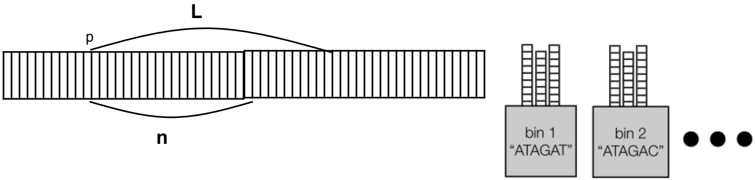


- In practice, n can be e.g. $20 \rightarrow 4^{20} = 1.1 \times 10^{12}$ bins

Hash table aligners

Alignment:

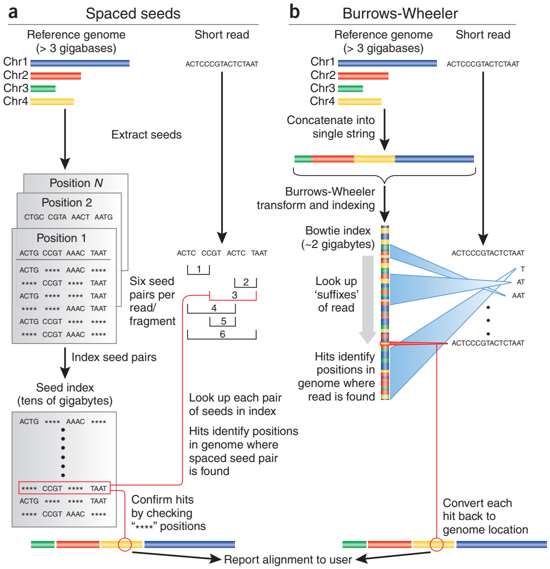
- For each position *p* in the reference genome
  - Consider the next L nucleotides in the reference sequence
  - Seed: Find the appropriate bin using the first n nucleotides (out of L)
  - Extend: Match the remaining reference sequence to reads in that bin
  - Some methods allow gaps, others don't



Hash table aligners

- Spaced seed: allow a number of mismatches in an alignment (e.g. at most 2) using several seeds / keys in parallel
- Example: Maq method divides each read into 4 seeds of equal length
  - Reasoning: Two mismatches will fall into at most two seed segments, leaving the other two to match perfectly
  - Procedure:
    - Find candidate locations by looking up all 6 possible pairs of seeds (for each read) in index
    - Then check remaining segments for candidates and remove candidates with too many mismatches

Hash table aligners



Hash table aligners

- By choosing proper seed lengths hash table index aligners can be very sensitive, but that may decrease the performance
  - Short seeds → false positives that slow down the (later) mapping process
  - Longer seeds → more seeds needed → more memory
- Comprehensive hash tables take lots of memory, which degrades performance in practical implementations
- Depending on the application, we may want to sacrifice sensitivity for performance, or limit mismatches

Burrows-Wheeler transform

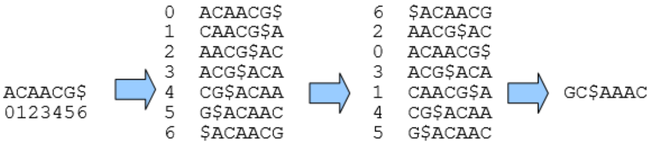
- Creates a transformation of the reference sequence that
  - Contains the same information as the original sequence
  - Can be compressed more efficiently
  - Allows fast lookup of substrings
  - Can be back-transformed
- Burrows-Wheeler Transform
  - Suffix array
  - Memory-efficient ($\sim$1GB for human genome)
  - Used by e.g. Bowtie (Langmead et al., 2009) and BWA

Suffix and prefix

- Definitions:
  - A prefix of a string S is a substring of S that occurs at the beginning of S
  - A suffix of a string S is a substring that occurs at the end of S
- By using certain representations of suffix/prefix tries the alignment of a sequencing read to multiple identical copies of a substring in the reference is done only once
  - All identical copies collapse on a single path in the trie
  - When using a typical hash table index, the alignment must be performed for each individual genome position
- Many of the current aligners use the so-called Burrows-Wheeler Transform (BWT)

Burrows-Wheeler transform: example

- Sequence: $X = \text{ACAACG}$
- Add the dollar symbol (or some other symbol) to mark the end of the string
  - \$ is considered to be lexicographically smaller than all the other symbols
- Create all cyclic permutations of the sequence X\$ = ACAACG\$ and then sort them in lexicographic order
- BWT of the sequence corresponds to the concatenation of the last character of each line in sorted list, here: BWT(ACAACG\$)=GC\$AAAC



```
ACAACG$
0123456
```

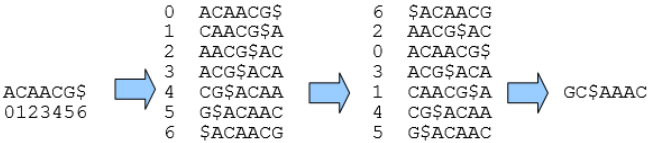
0	ACAACG$
1	CAACG$A
2	AACG$AC
3	ACG$ACA
4	CG$ACAA
5	G$ACAAC
6	$ACAACG

6	$ACAACG
2	AACG$AC
0	ACAACG$
3	ACG$ACA
1	CAACG$A
4	CG$ACAA
5	G$ACAAC

```
GC$AAAC
```

Burrows-Wheeler transform: example

- Suffix array: list of the original row numbers $A = (6, 2, 0, 3, 1, 4, 5)$
  - The suffix array A of a string X is an array of integers providing the starting positions of suffixes of S in lexicographical order
  - $A[i]$ = the starting position of the ith smallest suffix in S



ACAACG$
0123456

0	ACAACG$
1	CAACG$A
2	AACG$AC
3	ACG$ACA
4	CG$ACAA
5	G$ACAAC
6	$ACAACG

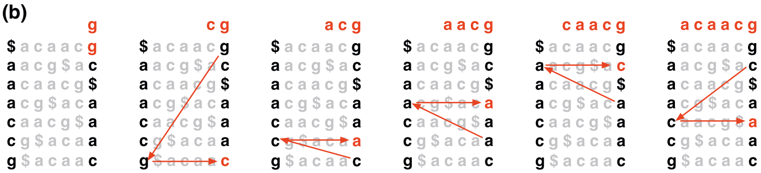
6	$ACAACG
2	AACG$AC
0	ACAACG$
3	ACG$ACA
1	CAACG$A
4	CG$ACAA
5	G$ACAAC

GC$AAAC

Reversing the BWT

- The BW matrix has the property of "last first (LF) mapping"
  - The *i*th occurrence of character C in the last column corresponds to the same character in the original string X as the *i*th occurrence of C in the first column
- We can use this property to reverse the transformation
  - Example: X = acaacg\$, BWT(X) = gc\$aaac

(b)



g	cg	acg	aacg	caacg	acaacg
\$acaacg	\$acaacg	\$acaacg	\$acaacg	\$acaacg	\$acaacg
aacg\$ac	aacg\$ac	aacg\$ac	aacg\$ac	aacg\$ac	aacg\$ac
acaacg\$	acaacg\$	acaacg\$	acaacg\$	acaacg\$	acaacg\$
acg\$aca	acg\$aca	acg\$aca	acg\$aca	acg\$aca	acg\$aca
caacg\$a	caacg\$a	caacg\$a	caacg\$a	caacg\$a	caacg\$a
cg\$acaa	cg\$acaa	cg\$acaa	cg\$acaa	cg\$acaa	cg\$acaa
g\$acaac	g\$acaac	g\$acaac	g\$acaac	g\$acaac	g\$acaac

Figure from (Langmead et al., 2008)

Reversing the BWT

- The BW matrix has the property of "last first (LF) mapping"
  - The *i*th occurrence of character C in the last column corresponds to the same character in the original string X as the *i*th occurrence of C in the first column
- We can use this property to reverse the transformation
  - Example: X = acaacg\$, BWT(X) = gc\$aaac

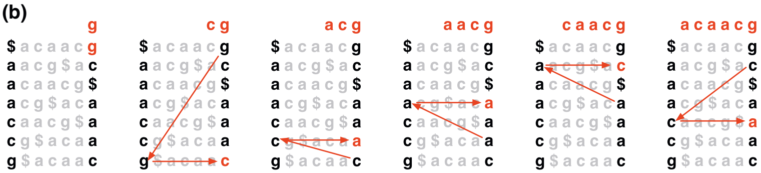


Figure from (Langmead et al., 2008)

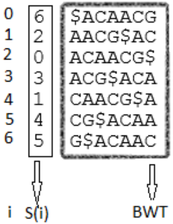
- Thus, $\mathrm{BWT}^{-1}(\mathrm{BWT}(X))$ = acaacg\$ = X

Suffix array interval

- Suffix array values store the positions of the occurrences of suffixes in the string in *lexicographical order*

→ All suffixes that have a given substring W as prefix appear on consecutive rows

- All occurrences of a substring W in X appear in an interval (consecutive indices) in suffix array
- Examples: W = AC → [2,3], W = A → [1,3]

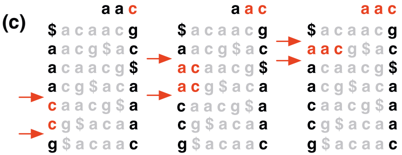


i	S(i)	
0	6	$ACAACG
1	2	AACG$AC
2	0	ACAACG$
3	3	ACG$ACA
4	1	CAACG$A
5	4	CG$ACAA
6	5	G$ACAAC

Short read aligment using BWT

The LF and suffix array intervals are used in exact matching of substrings/short reads

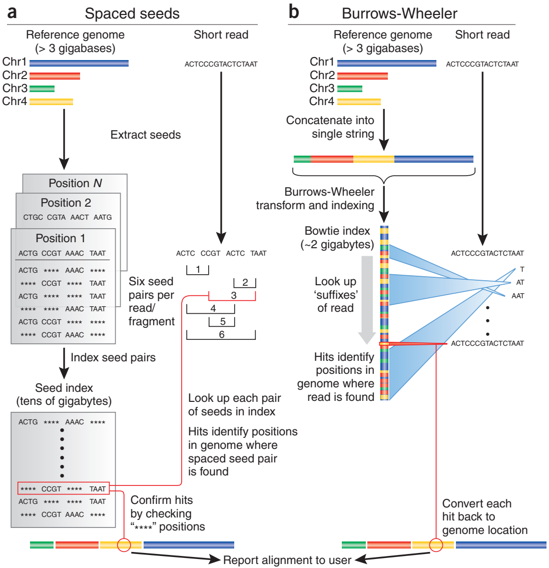
- The range of matrix rows beginning with successively longer suffixes of the query/read are back-tracked
- At each step, the size of the range either decreases or remains the same
- When the algorithm completes, rows beginning with the entire query correspond to exact occurrences of the query in the text
- If the range is empty, the text does not contain the query
- Example query: AAC



```
        a a c          a a c          a a c
  $ a c a a c g  $ a c a a c g  $ a c a a c g
  a a c g $ a c  a a c g $ a c  a a c g $ a c
  a c a a c g $  a c a a c g $  a c a a c g $
  a c g $ a c a  a c g $ a c a  a c g $ a c a
  c a a c g $ a  c a a c g $ a  c a a c g $ a
  c g $ a c a a  c g $ a c a a  c g $ a c a a
  g $ a c a a c  g $ a c a a c  g $ a c a a c
```

Figure from (Langmead et al., 2008)

BWT aligners



a Spaced seeds

Reference genome (> 3 gigabases)

Short read

Extract seeds

Six seed pairs per read/ fragment

Index seed pairs

Seed index (tens of gigabytes)

Look up each pair of seeds in index

Hits identify positions in genome where spaced seed pair is found

Confirm hits by checking "****" positions

b Burrows-Wheeler

Reference genome (> 3 gigabases)

Short read

Concatenate into single string

Burrows-Wheeler transform and indexing

Bowtie index (~2 gigabytes)

Look up 'suffixes' of read

Hits identify positions in genome where read is found

Convert each hit back to genome location

Report alignment to user

Inexact alignment with BWT

- The search proceeds similarly to exact alignment
- If the range becomes empty (a suffix does not occur in the text), then the algorithm may select an already-matched query position and substitute a different base there, introducing a mismatch into the alignment
- The exact alignment search resumes from just after the substituted position
- The algorithm selects only those substitutions which yield a modified suffix that occurs at least once in the text (and that are consistent with the alignment policy)
- If there are multiple candidate substitution positions, then the algorithm greedily selects a position with a minimal quality value
- More recent methods, such as Bowtie2, also allow indels, e.g.

References

- Illumina sequencing technology http://www.illumina.com
- Andrews S. (2010) http://www.bioinformatics.babraham.ac.uk/projects/fastqc/
- Trapnell C & Salzberg SL, How to map billions of short reads onto genomes, *Nature Biotechnology*, 27, 455–457, 2009.
- Li H, et al., “Mapping short DNA sequencing reads and calling variants using mapping quality scores.” *Genome Res.* 18:1851–8, 2008.
- Langmead B et al., Ultrafast and memory-efficient alignment of short DNA sequences to the human genome, *Genome Biol.*, 10(3):R25, 2009.

Additional reading

- Lecture slides by Ben Langmead https://www.cs.jhu.edu/~langmea/resources/lecture_notes/bwt_and_fm_index.pdf
- A book by Veli MÃ¤kinen et al. “Genome-Scale Algorithm Design”, Cambridge Univ. Press, 2015. http://www.genome-scale.info