BMEG3102 Bioinformatics

Lecture 8. High-throughput Data Processing and Analysis

Qi Dou

Email: qidou@cuhk.edu.hk

Office: Room 1014, 10/F, SHB

BMEG3102 Bioinformatics

The Chinese University of Hong Kong

Lecture outline

1. Ome, omic and omics

2. Genomics (DNA)

- Sequencing methods
- Computational problems in reconstructing sequence

3. Transcriptomics (RNA)

- Microarrays and RNA-sequencing
- Data clustering and classification

Part 1

Ome, Omic and Omics

Ome, omic, and omics

- Traditionally, biologists study one or a few biological objects at a time
  - Hypothesis driven

Ome, omic, and omics

- Traditionally, biologists study one or a few biological objects at a time
  - Hypothesis driven
- Now it is possible to study many biological objects at the same time
  - Data driven

Ome, omic, and omics

- Traditionally, biologists study one or a few biological objects at a time
  - Hypothesis driven
- Now it is possible to study many biological objects at the same time
  - Data driven
- Suppose we want to study a type of objects or phenomena, X
  - "X-ome": A large amount of data related to X, or the whole set of X
  - "X-omic": To study a large amount of data related to X
  - "X-omics": The area of studying a large amount of data related to X

What: Different kinds of X-omics

Object/ phenomenon type, X	X-ome	X-omics
Genes/ DNA	Genome	Genomics (The study of all genes/whole set of DNA)
Transcripts/ transcription	Transcriptome	Transcriptomics (The study of gene expression levels)
Exons/ transcription	Exome	Exomics
Proteins	Proteome	Proteomics (The study of protein identity and abundance)
Metabolism	Metabolome	Metabolomics (The study of metabolic reactions)
DNA methylation	Methylome	Methylomics
Non-coding RNAs, DNA methylation, histone modifications	Epigenome	Epigenomics (The study of inheritable non-DNA signals)
Population of co-existing species in an environment	Metagenome	Metagenomics (The study of different genomes, transcriptomes, etc. in a common environment)
Phenotypes	Phenome	Phenomics
Interactions	Interactome	Interactomics
...	...	...

How: High-throughput experiments

- Key idea in omic research: high-throughput experiments by means of...
  - Parallelization

How: High-throughput experiments

- Key idea in omic research: high-throughput experiments by means of...
  - Parallelization
- Examples:
  - Measuring the expression levels of a small number of genes (e.g., RT-PCR) vs. measuring the expression levels of a large number of genes in parallel (microarray or RNA-seq)
  - Measuring the interaction between a protein and a particular piece of DNA (e.g., DNase I footprinting) vs. measuring the interactions between a protein and all regions in the genome in parallel (ChIP-chip or ChIP-seq)

Why: Strengths and weaknesses

- Strengths of the omic approach:
  - High-throughput: fast, less tedious, relatively inexpensive
  - Comprehensive
  - Relatively unbiased
  - Easier to study interactions and combinatorial effects

Why: Strengths and weaknesses

- Strengths of the omic approach:
  - High-throughput: fast, less tedious, relatively inexpensive
  - Comprehensive
  - Relatively unbiased
  - Easier to study interactions and combinatorial effects

- Weaknesses:
  - Noise
  - Secondary effects
  - Lack of clear hypotheses
  - High initial cost (the machines)

Typical omic workflow

1. Production of data
2. Data processing
   - Quality control
   - Data normalization
3. Data analysis (pattern discovery)
4. Data annotation and comparisons
   - Evaluation of statistical significance
5. Selection and summarization of results
6. Hypothesis formation
7. Experimental validation

Part 2

Genomics

Sequencing

- We have studied many problems related to sequences
  - Alignments
  - Estimation of actual number of substitutions based on the observed number
  - Phylogenetic tree reconstruction
  - Secondary structure prediction

- How did we get the sequences in the first place?
  - Sequencing
    - Input: Cell sample containing the DNA
    - Output: The string representation of the DNA sequence

Sanger sequencing

- For sequencing DNA
- Low-throughput, but high reliability
- Can sequence up to 300-1000 nucleotides per reaction
  - Versus ~100nt for high-throughput experiments
- Used for sequencing the first human genome
- Method of choice for common laboratory use
- Now also used for validating results obtained from high-throughput, “next-generation” sequencing

Next generation sequencing

- "Second generation", "next generation", or "massively parallel" sequencing

- Going parallel
  - Platform (droplet vs. solid-phase)
  - Immobilization (primer vs. template vs. polymerase)

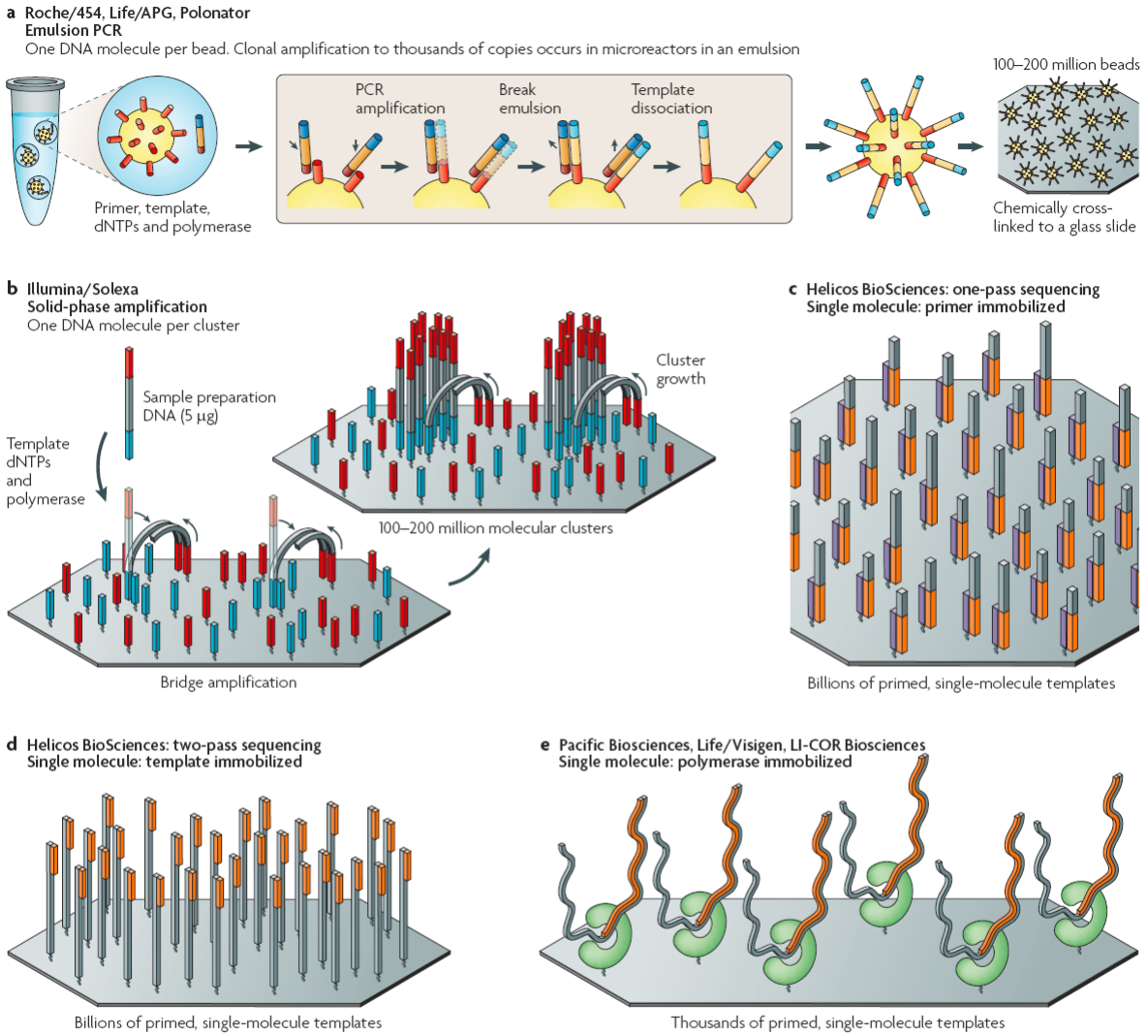


Image credit: Metzker, *Nature Reviews Genetics* 11:31-46, (2010)

Comparison of technologies

Table 1 Comparison of next generation sequencing platforms

Company	Sequencing Principle	Detection	System platform	Read length (bp)	Number of Reads	Time/run	Throughput/run	Accuracy	Machine cost ($)	Advantage	Disadvantage
Illumina	Reversible terminator sequencing by synthesis	Fluorescence/ Optical	HiSeq 2500/1500	36/50/100	3 billion (SE)	2–11 days	600 GB	> 99%	740,000	Very high throughput; Cost-effectiveness; Steadily improving read lengths; Massive throughput	Long run time; Short read lengths; Expensive instrument; Lower error rate
			Genome Analyzer IIx	35/50/75/100	320 million (SE)	2–14 days	95 GB	> 99%	250,000	High throughput; The most widely used platform	Low multiplexing capability of samples
			MiSeq	25/36/100/ 150/250	17 million (SE)	4–27 hours	8.5 GB	> 99%	125,000	High throughput; Cost-effectiveness; Short run times; Appropriate throughput for microbial applications; Minimal hands-on time; High coverage	Short read lengths
Roche	Pyrosequencing	Optical	454 GS FLX+	700	1 million	23 hours	0.7 GB	99.997%	450,000	High throughput; Longer read lengths; Short run times; High coverage	Appreciable hands-on time; High reagent costs; Higher error rate in homopolymers regions
			454 GS Junior	400	1 million	10 hours	0.035 GB	> 99%	108,000	Longer read lengths; Short run times	
Helicos Biosciences	Single molecule sequencing	Fluorescence/ Optical	Heliscope	25–55 (average: 32)	600–800 million	8 days	37 GB	99.99%	999,000	Single-molecule nature of technology; Non-bias representation of templates for genome	Expensive instrument; Very short read lengths (increase cost and difficulty of assembly); Higher error rate
ABI Life Technologies	Ligation	Fluorescence/ Optical	5500 SOLiD	75+35	1.4 billion	7 days	90 GB	99.99%	350,000	High throughput; Lowest reagent cost	Long run times; Very short read lengths (increase cost and difficulty of assembly)
			5500xl SOLiD	75+35	2.8 billion	7 days	180 GB	99.99%	595,000	Very high throughput; Low error rate; Massive throughput	
	Proton detection	Change in pH detected by Ion-Sensitive Field Effect Transistors (ISFETs)	Ion Personal Genome Machine (PGM)	35/200/400	12 million	2 hours	2 GB	> 99%	80,000	Short run times; Low cost per sample; Appropriate throughput for microbial applications; Direct measurement of nucleobase incorporation events	Appreciable hands-on time; High reagent costs; Higher error rate in homopolymers (sequential washing steps)
			Ion Proton Chip I/II	Up to 200	60-80 million	2 hours	10 GB / 100 GB	> 99%	243,000	Short run times; Flexible chip reagents	Instrument not available at time of writing
Pacific Biosciences	Real-time, single molecule DNA sequencing	Fluorescence/ Optical	PacBio RS	Average: 3000	~50 K	2 hours	13 GB	84–85%	750,000	Short run times; Very long read lengths; Low reagent costs; Simple sample preparation	No paired reads; Highest error rates; Expensive instrument; Difficult installation
Oxford Nanopore	Nanopore exonuclease sequencing	Electrical Conductivity	gridION	Tens of Kb	4–10 million	According to experiment	Tens of GB	96%	According to experiment	Extremely long read lengths; Low cost of α-HL nanopore production; Customization; No fluorescent labeling; No optics	4% error rates; Cleaved nucleotide may be read in the wrong order; Difficult to fabricate a device with multiple parallel pores

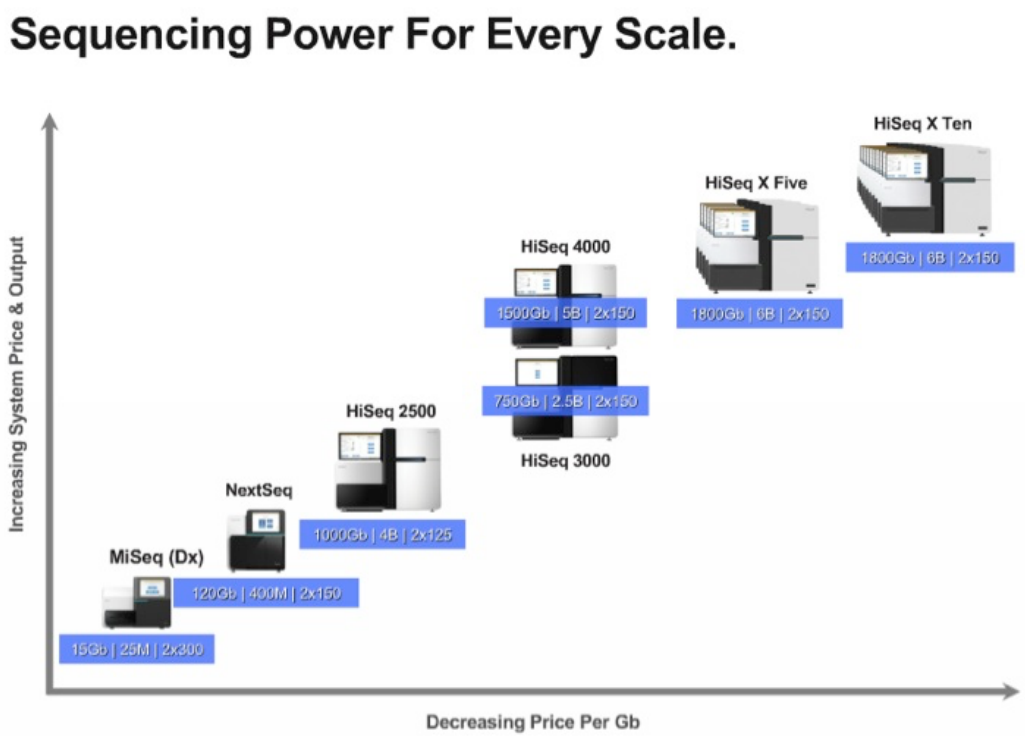


Image credit: Lee et al., *Translational Cancer Research* 2:1 (2013); http://blog.genohub.com/wp-content/uploads/2015/01/Slide1.jpg

Sequencing cost

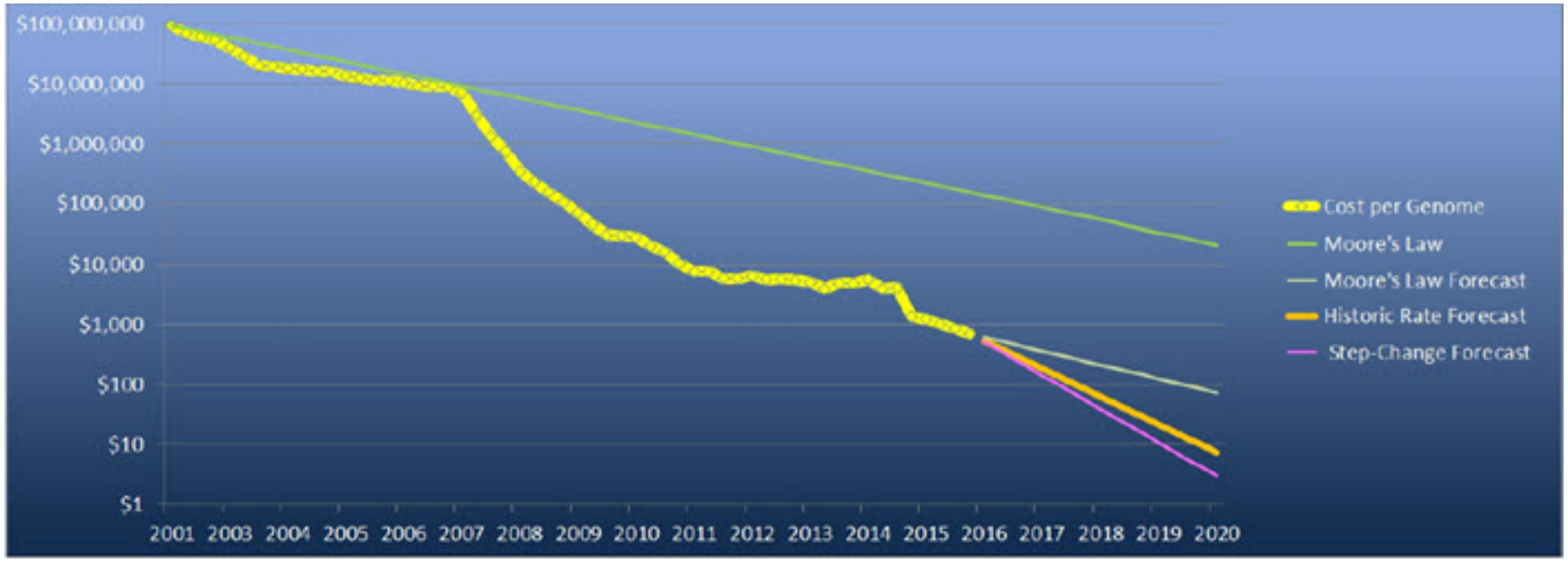


Figure 1: Human Genome Sequencing Costs. Data Source: NHGRI Genome Sequencing Program (GSP)

Image source: https://insidehpc.com/2016/11/enabling-personalized-medicine-through-genomic-workflow-acceleration/

Third generation sequencing

- Characteristics:
  - Longer reads
  - Higher error rate (currently)
  - Higher cost (currently)
  - Single-cell sequencing

- Example: Pacific Biosciences' Single Molecule Real-Time (SMRT) sequencing
  - Several hundred base pairs or more
  - >10 times higher error rate than NGS

Third generation sequencing technologies

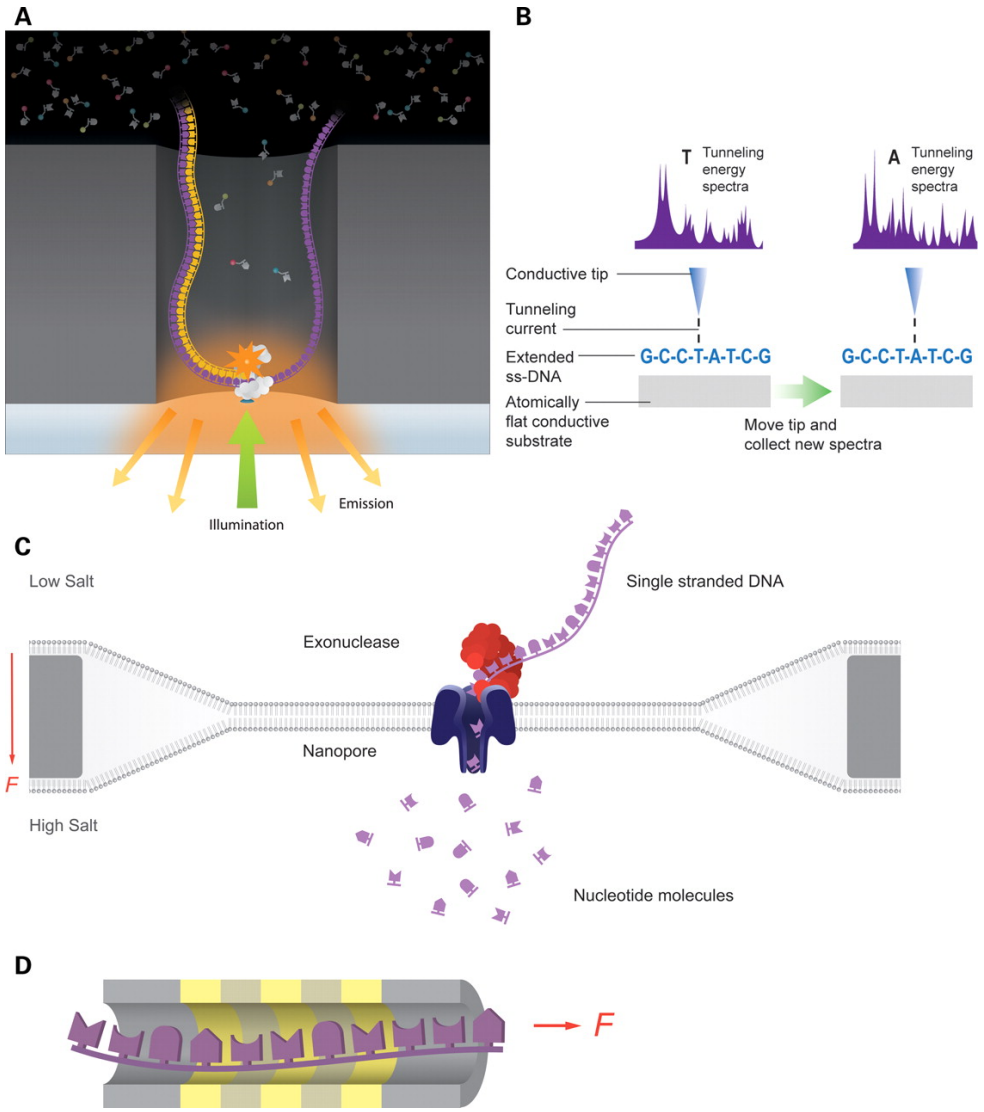


Image credit: Schadt et al., *Human Molecular Genetics* 19(R2):227-240, (2010)

How third-generation DNA-sequencing technologies work. Third-generation DNA-sequencing technologies are distinguished by direct inspection of single molecules with methods that do not require wash steps during DNA synthesis. (**A**) Pacific Biosciences technology for direct observation of DNA synthesis on single DNA molecules in real time. A DNA polymerase is confined in a zero-mode waveguide and base additions measured with florescence detection of gamma-labeled phosphonucleotides. (**B**) Several companies seek to sequence DNA by direct inspection using electron microscopy similar to the Reveo technology pictured here, in which an ssDNA molecule is first stretched and then examined by STM. (**C**) Oxford Nanopore technology for measuring translocation of nucleotides cleaved from a DNA molecule across a pore, driven by the force of differential ion concentrations across the membrane. (**D**) IBM's DNA transistor technology reads individual bases of ssDNA molecules as they pass through a narrow aperture based on the unique electronic signature of each individual nucleotide. Gold bands represent metal and gray bands dielectric layers of the transistor.

Sequencing a long DNA

- How to sequence DNA longer than what a single reaction can achieve?
  - Cut the DNA into shorter fragments

Sequencing a long DNA

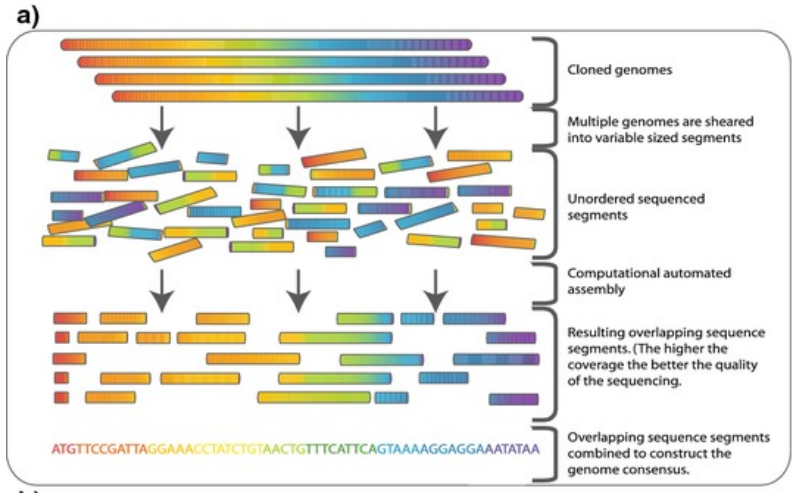
- How to sequence DNA longer than what a single reaction can achieve?
  - Cut the DNA into shorter fragments

- How to get back the whole sequence?
  - If the short fragments do not overlap, we need to record the exact order
  - Experimentally infeasible: 3×10^9 / 1000 = 3×10^6 fragments

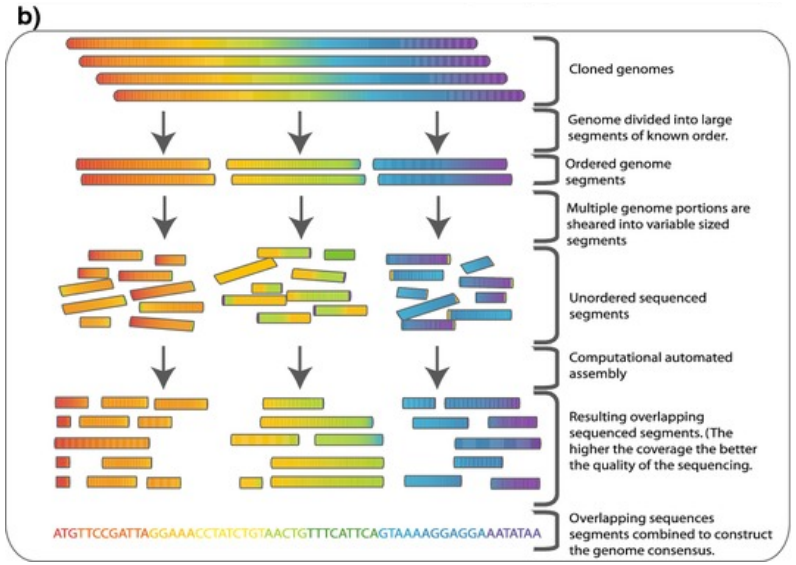
Sequencing a long DNA

- How to sequence DNA longer than what a single reaction can achieve?
  - Cut the DNA into shorter fragments
- How to get back the whole sequence?
  - If the short fragments do not overlap, we need to record the exact order
  - Experimentally infeasible: 3×10^9 / 1000 = 3×10^6 fragments
- Key idea: cut randomly, with overlaps
  - 60x coverage means on average each position is covered by 60 sequencing reads
  - "Shotgun sequencing"

Shotgun sequencing



Whole genome in a single run



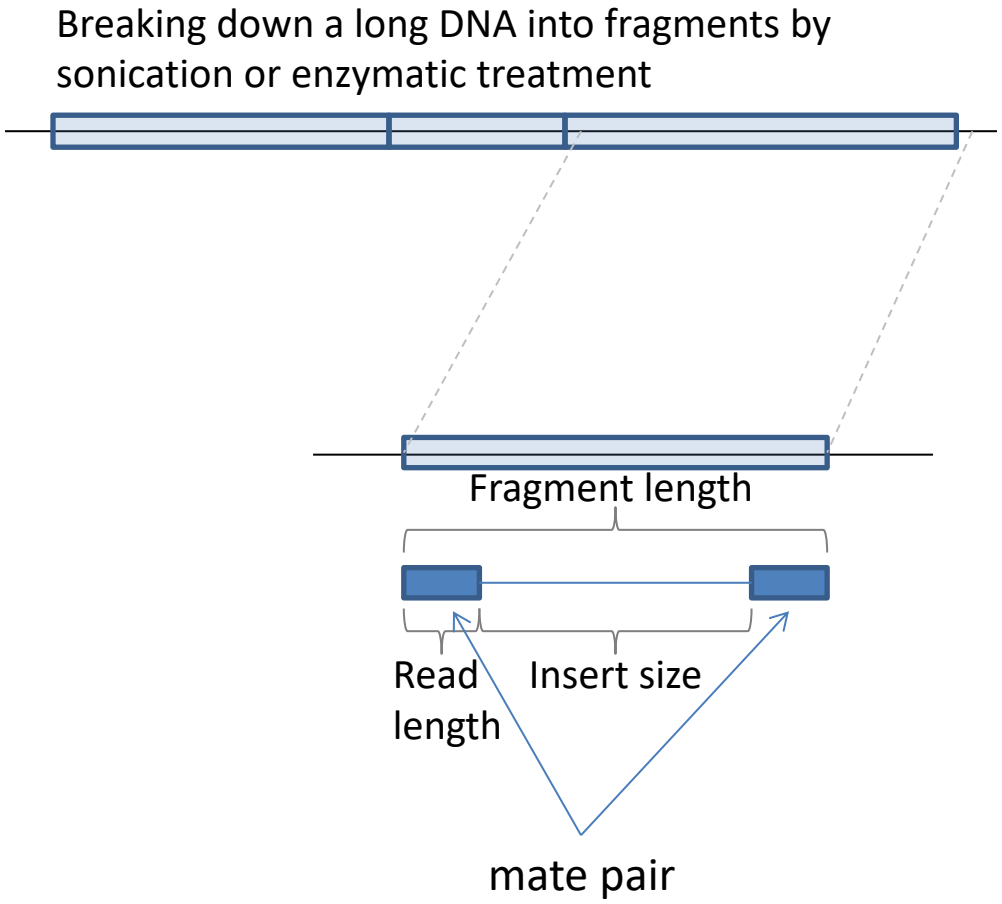
Hierarchical approach

Image credit: Commins et al., *Biological Procedures Online* 11(1):52-78, (2009)

The resulting raw data

- Raw data:
  - Single-end sequencing: one sequencing read (i.e., a short string) per fragment
  - Paired-end sequencing: two sequencing reads per fragment

- Quality score:
  - How reliable each sequenced base is
    - While sequencing is quite reliable, errors do occur (e.g., due to unclear signals)



FASTQ file format (for raw reads)

- Sequence (FASTA) + Quality (Q)
- Each sequence occupies four lines:

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*(((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

  - Line 1: @, followed by sequence ID and descriptions
  - Line 2: sequence
  - Line 3: +, optionally followed by sequence ID and descriptions
  - Line 4: quality scores
    - Standard: int score = (ASCII code of character) – 33;
      - E.g., '!' means a quality (Phred) score of 33 – 33 = 0, i.e., very bad
      - If probability of base-calling error is p, then Phred score is $-\log_{10} p$
    - Illumina has a different standard (with different versions)

Example source: Wikipedia

ASCII table and Phred score

Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	[NULL]
1	1	1	1	[START OF HEADING]
2	2	10	2	[START OF TEXT]
3	3	11	3	[END OF TEXT]
4	4	100	4	[END OF TRANSMISSION]
5	5	101	5	[ENQUIRY]
6	6	110	6	[ACKNOWLEDGE]
7	7	111	7	[BELL]
8	8	1000	10	[BACKSPACE]
9	9	1001	11	[HORIZONTAL TAB]
10	A	1010	12	[LINE FEED]
11	B	1011	13	[VERTICAL TAB]
12	C	1100	14	[FORM FEED]
13	D	1101	15	[CARRIAGE RETURN]
14	E	1110	16	[SHIFT OUT]
15	F	1111	17	[SHIFT IN]
16	10	10000	20	[DATA LINK ESCAPE]
17	11	10001	21	[DEVICE CONTROL 1]
18	12	10010	22	[DEVICE CONTROL 2]
19	13	10011	23	[DEVICE CONTROL 3]
20	14	10100	24	[DEVICE CONTROL 4]
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]
22	16	10110	26	[SYNCHRONOUS IDLE]
23	17	10111	27	[ENG OF TRANS. BLOCK]
24	18	11000	30	[CANCEL]
25	19	11001	31	[END OF MEDIUM]
26	1A	11010	32	[SUBSTITUTE]
27	1B	11011	33	[ESCAPE]
28	1C	11100	34	[FILE SEPARATOR]
29	1D	11101	35	[GROUP SEPARATOR]
30	1E	11110	36	[RECORD SEPARATOR]
31	1F	11111	37	[UNIT SEPARATOR]
32	20	100000	40	[SPACE]
33	21	100001	41	!
34	22	100010	42	"
35	23	100011	43	#
36	24	100100	44	$
37	25	100101	45	%
38	26	100110	46	&
39	27	100111	47	'
40	28	101000	50	(
41	29	101001	51	)
42	2A	101010	52	*
43	2B	101011	53	+
44	2C	101100	54	,
45	2D	101101	55	-
46	2E	101110	56	.
47	2F	101111	57	/
48	30	110000	60	0
49	31	110001	61	1
50	32	110010	62	2
51	33	110011	63	3
52	34	110100	64	4
53	35	110101	65	5
54	36	110110	66	6
55	37	110111	67	7
56	38	111000	70	8
57	39	111001	71	9
58	3A	111010	72	:
59	3B	111011	73	;
60	3C	111100	74	<
61	3D	111101	75	=
62	3E	111110	76	>
63	3F	111111	77	?
64	40	1000000	100	@
65	41	1000001	101	A
66	42	1000010	102	B
67	43	1000011	103	C
68	44	1000100	104	D
69	45	1000101	105	E
70	46	1000110	106	F
71	47	1000111	107	G
72	48	1001000	110	H
73	49	1001001	111	I
74	4A	1001010	112	J
75	4B	1001011	113	K
76	4C	1001100	114	L
77	4D	1001101	115	M
78	4E	1001110	116	N
79	4F	1001111	117	O
80	50	1010000	120	P
81	51	1010001	121	Q
82	52	1010010	122	R
83	53	1010011	123	S
84	54	1010100	124	T
85	55	1010101	125	U
86	56	1010110	126	V
87	57	1010111	127	W
88	58	1011000	130	X
89	59	1011001	131	Y
90	5A	1011010	132	Z
91	5B	1011011	133	[
92	5C	1011100	134	\
93	5D	1011101	135	]
94	5E	1011110	136	^
95	5F	1011111	137	_
96	60	1100000	140	`
97	61	1100001	141	a
98	62	1100010	142	b
99	63	1100011	143	c
100	64	1100100	144	d
101	65	1100101	145	e
102	66	1100110	146	f
103	67	1100111	147	g
104	68	1101000	150	h
105	69	1101001	151	i
106	6A	1101010	152	j
107	6B	1101011	153	k
108	6C	1101100	154	l
109	6D	1101101	155	m
110	6E	1101110	156	n
111	6F	1101111	157	o
112	70	1110000	160	p
113	71	1110001	161	q
114	72	1110010	162	r
115	73	1110011	163	s
116	74	1110100	164	t
117	75	1110101	165	u
118	76	1110110	166	v
119	77	1110111	167	w
120	78	1111000	170	x
121	79	1111001	171	y
122	7A	1111010	172	z
123	7B	1111011	173	{
124	7C	1111100	174	|
125	7D	1111101	175	}
126	7E	1111110	176	~
127	7F	1111111	177	[DEL]

Image source: Wikimedia

ASCII table and Phred score

Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	[NULL]
1	1	1	1	[START OF HEADING]
2	2	10	2	[START OF TEXT]
3	3	11	3	[END OF TEXT]
4	4	100	4	[END OF TRANSMISSION]
5	5	101	5	[ENQUIRY]
6	6	110	6	[ACKNOWLEDGE]
7	7	111	7	[BELL]
8	8	1000	10	[BACKSPACE]
9	9	1001	11	[HORIZONTAL TAB]
10	A	1010	12	[LINE FEED]
11	B	1011	13	[VERTICAL TAB]
12	C	1100	14	[FORM FEED]
13	D	1101	15	[CARRIAGE RETURN]
14	E	1110	16	[SHIFT OUT]
15	F	1111	17	[SHIFT IN]
16	10	10000	20	[DATA LINK ESCAPE]
17	11	10001	21	[DEVICE CONTROL 1]
18	12	10010	22	[DEVICE CONTROL 2]
19	13	10011	23	[DEVICE CONTROL 3]
20	14	10100	24	[DEVICE CONTROL 4]
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]
22	16	10110	26	[SYNCHRONOUS IDLE]
23	17	10111	27	[ENG OF TRANS. BLOCK]
24	18	11000	30	[CANCEL]
25	19	11001	31	[END OF MEDIUM]
26	1A	11010	32	[SUBSTITUTE]
27	1B	11011	33	[ESCAPE]
28	1C	11100	34	[FILE SEPARATOR]
29	1D	11101	35	[GROUP SEPARATOR]
30	1E	11110	36	[RECORD SEPARATOR]
31	1F	11111	37	[UNIT SEPARATOR]
32	20	100000	40	[SPACE]
33	21	100001	41	!
34	22	100010	42	"
35	23	100011	43	#
36	24	100100	44	$
37	25	100101	45	%
38	26	100110	46	&
39	27	100111	47	'
40	28	101000	50	(
41	29	101001	51	)
42	2A	101010	52	*
43	2B	101011	53	+
44	2C	101100	54	,
45	2D	101101	55	-
46	2E	101110	56	.
47	2F	101111	57	/
48	30	110000	60	0
49	31	110001	61	1
50	32	110010	62	2
51	33	110011	63	3
52	34	110100	64	4
53	35	110101	65	5
54	36	110110	66	6
55	37	110111	67	7
56	38	111000	70	8
57	39	111001	71	9
58	3A	111010	72	:
59	3B	111011	73	;
60	3C	111100	74	<
61	3D	111101	75	=
62	3E	111110	76	>
63	3F	111111	77	?
64	40	1000000	100	@
65	41	1000001	101	A
66	42	1000010	102	B
67	43	1000011	103	C
68	44	1000100	104	D
69	45	1000101	105	E
70	46	1000110	106	F
71	47	1000111	107	G
72	48	1001000	110	H
73	49	1001001	111	I
74	4A	1001010	112	J
75	4B	1001011	113	K
76	4C	1001100	114	L
77	4D	1001101	115	M
78	4E	1001110	116	N
79	4F	1001111	117	O
80	50	1010000	120	P
81	51	1010001	121	Q
82	52	1010010	122	R
83	53	1010011	123	S
84	54	1010100	124	T
85	55	1010101	125	U
86	56	1010110	126	V
87	57	1010111	127	W
88	58	1011000	130	X
89	59	1011001	131	Y
90	5A	1011010	132	Z
91	5B	1011011	133	[
92	5C	1011100	134	\
93	5D	1011101	135	]
94	5E	1011110	136	^
95	5F	1011111	137	_
96	60	1100000	140	`
97	61	1100001	141	a
98	62	1100010	142	b
99	63	1100011	143	c
100	64	1100100	144	d
101	65	1100101	145	e
102	66	1100110	146	f
103	67	1100111	147	g
104	68	1101000	150	h
105	69	1101001	151	i
106	6A	1101010	152	j
107	6B	1101011	153	k
108	6C	1101100	154	l
109	6D	1101101	155	m
110	6E	1101110	156	n
111	6F	1101111	157	o
112	70	1110000	160	p
113	71	1110001	161	q
114	72	1110010	162	r
115	73	1110011	163	s
116	74	1110100	164	t
117	75	1110101	165	u
118	76	1110110	166	v
119	77	1110111	167	w
120	78	1111000	170	x
121	79	1111001	171	y
122	7A	1111010	172	z
123	7B	1111011	173	{
124	7C	1111100	174	|
125	7D	1111101	175	}
126	7E	1111110	176	~
127	7F	1111111	177	[DEL]

- Example: The character '*' has ASCII code of 42
- Suppose the probability of sequencing error of that base is p, then
  - $-\log_{10} p = 42 - 33$
  - $\Rightarrow p = 10^{-9}$
  - $\Rightarrow$ Reported base is unlikely to be an error

Image source: Wikimedia

Getting back the original sequence

- After sequencing the ends of the fragments (the “reads”), how to get back the original sequence?

- Two main approaches:
  - Sequence assembly (“*de novo* assembly”)
  - Sequence alignment (“re-sequencing”)

Sequence assembly

- From the short fragments, reconstruct the original long sequence
  - Not always able to get one single final sequence
  - If not, each assembled sequence is called a contig (remember seeing this term in GenBank?)

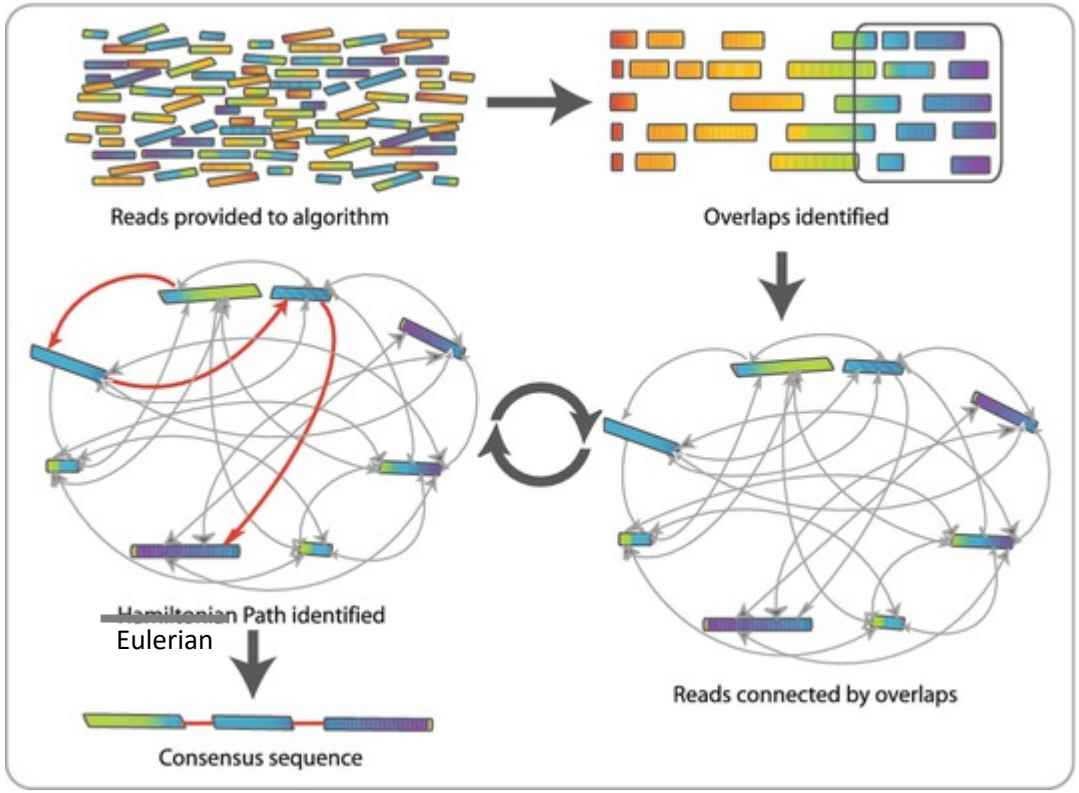


Image credit: Commins et al., *Biological Procedures Online* 11(1):52-78, (2009)

Sequence assembly methods

- Many methods proposed for different sequencing methods
  - ABySS
  - Euler
  - SOAPdenovo
  - Velvet
  - …
- We will just study some basic ideas

de Bruijn graph

- A set of nodes, each with a length-k sub-sequence
- There is a directed edge from a node A to a node B if the length-(k-1) suffix of A is equal to the length-(k-1) prefix of B
- Example:

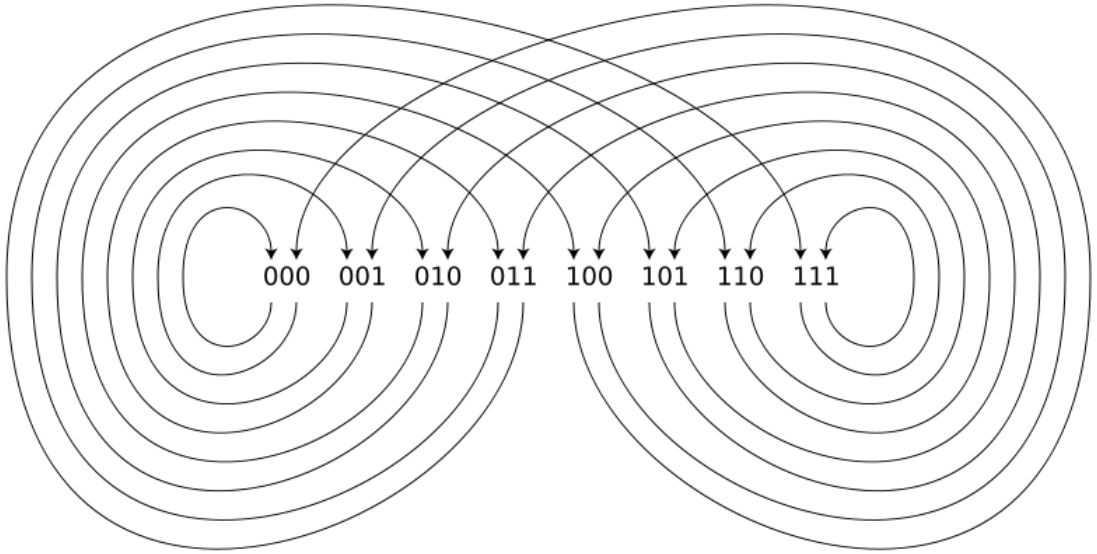


Image credit: Wikipedia

How to pronounce “de Bruijn”? You may take a look at this page:
http://thegenomefactory.blogspot.hk/2013/08/how-to-pronounce-de-bruijn.html

de Bruijn graph and sequencing

- After sequencing, we obtain a set of short reads, each consisting of a sub-sequence
- We can construct a de Bruijn graph by considering k-mers of the sub-sequences
  - Two nodes are connected if they appear in consecutive positions on a single read
- Then based on the graph structure, we try to deduce the original long sequence
  - Theoretically, we try to find a path that visits every edge exactly once
  - In practice, there are many issues

Illustration: Constructing the de Bruijn graph

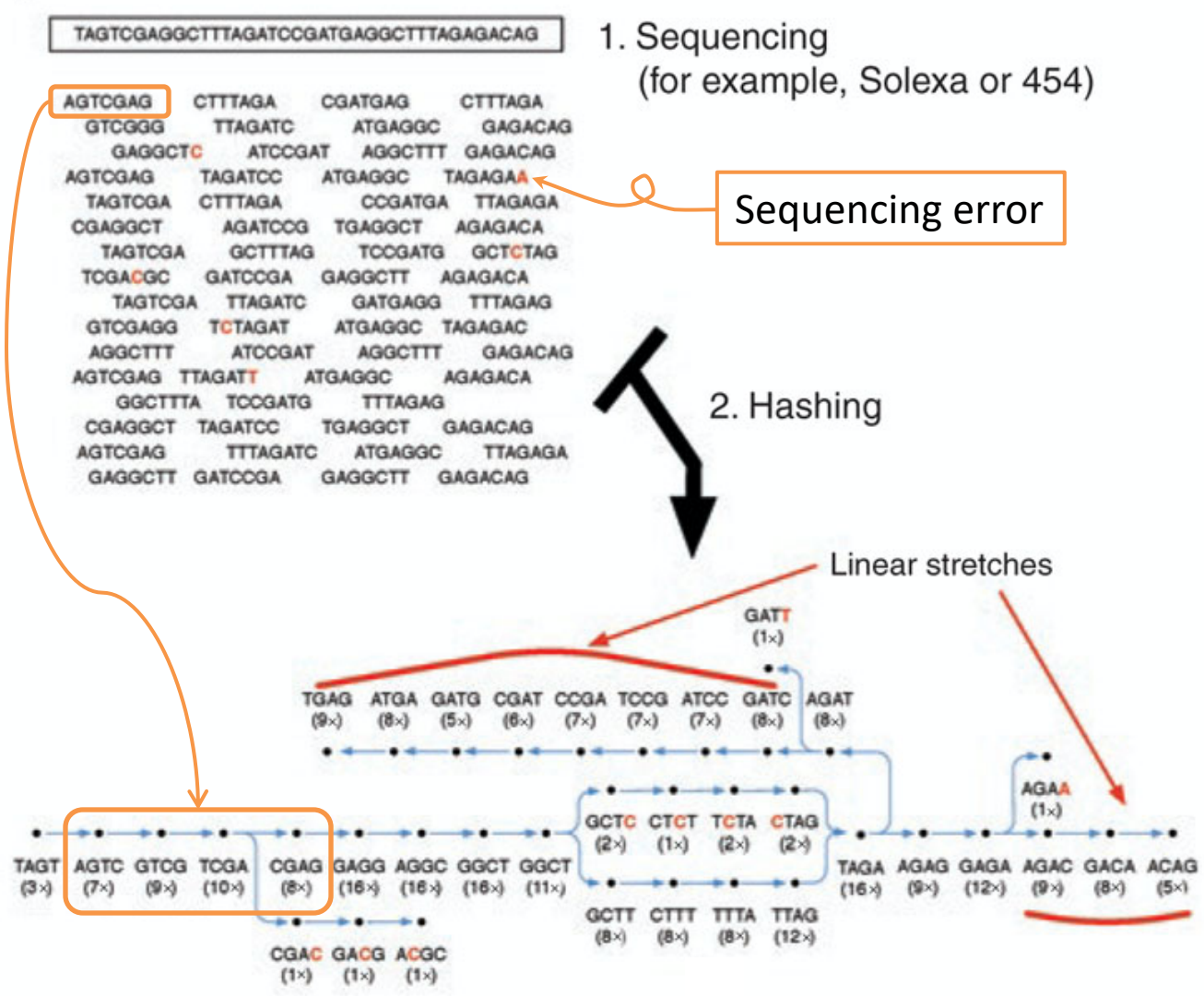


Image credit: Flicek and Birney, *Nature Methods* 6(11s):S6-S12, (2009)

Simplification

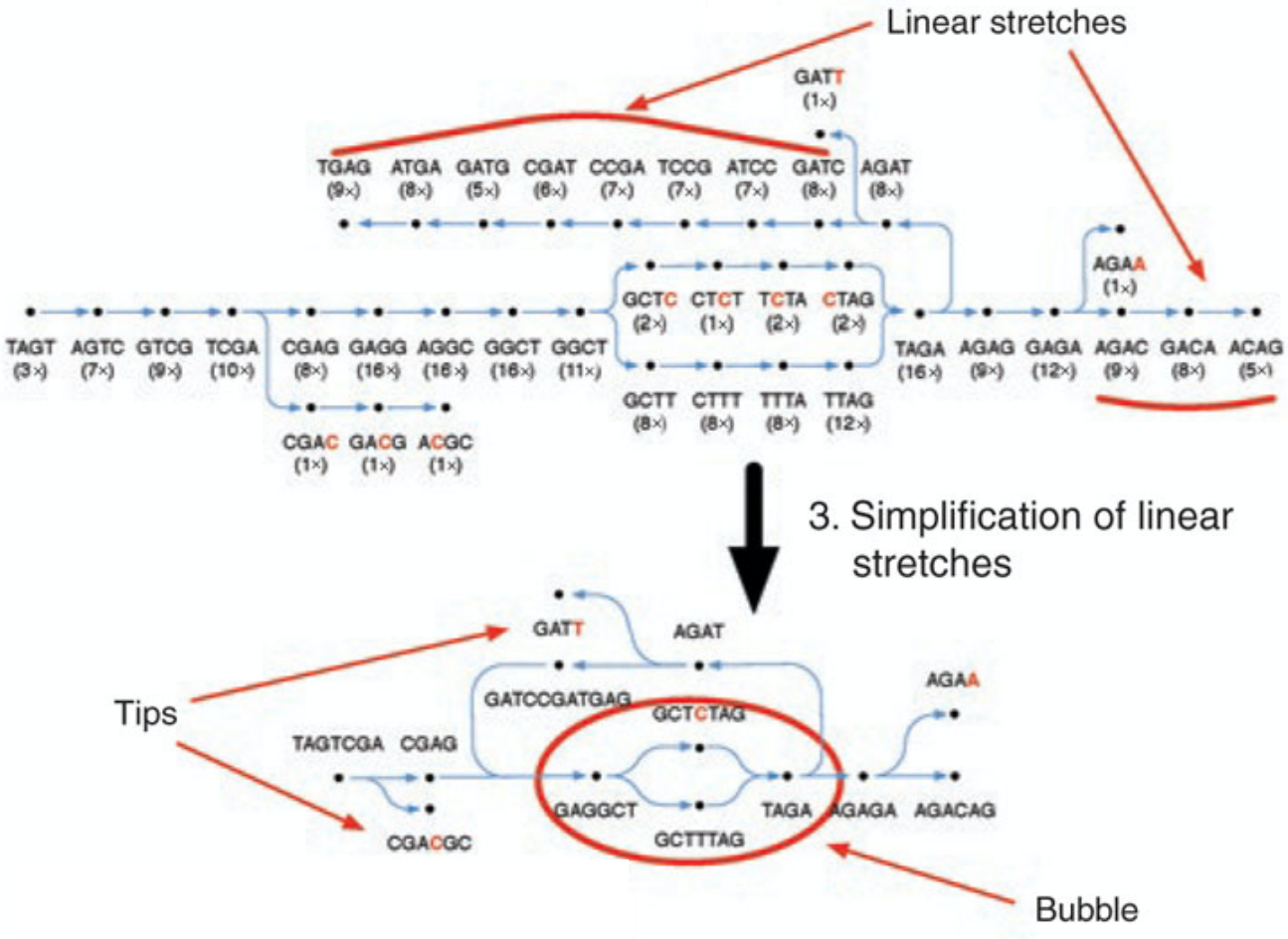


Image credit: Flicek and Birney, *Nature Methods* 6(11s):S6-S12, (2009)

Error detection and removal

- Signals of errors:
  - Tips
  - Bubbles
  - Low-coverage paths

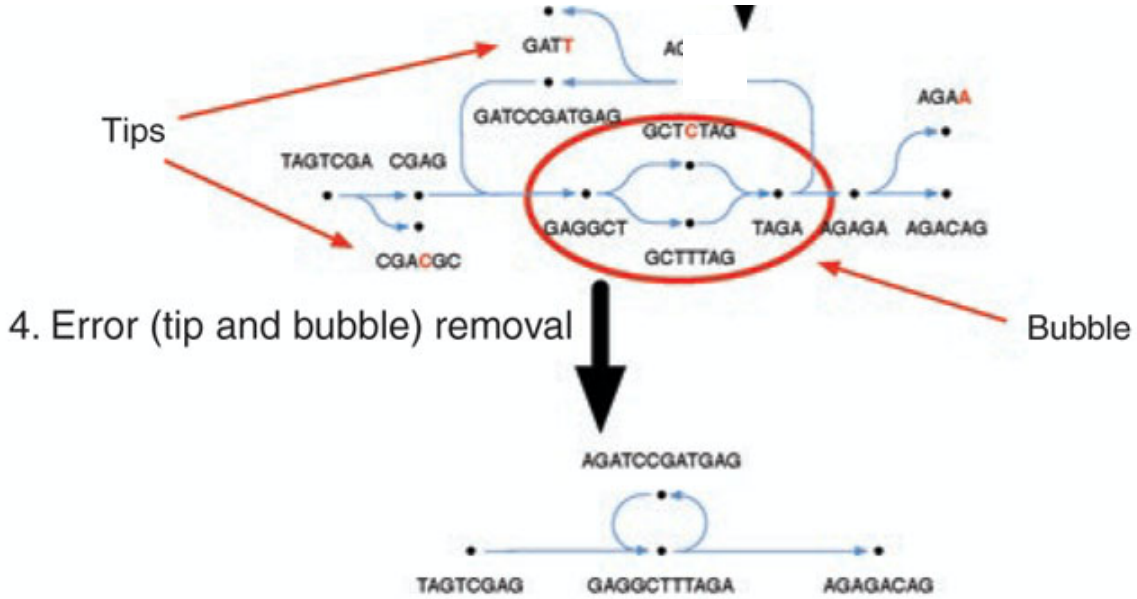
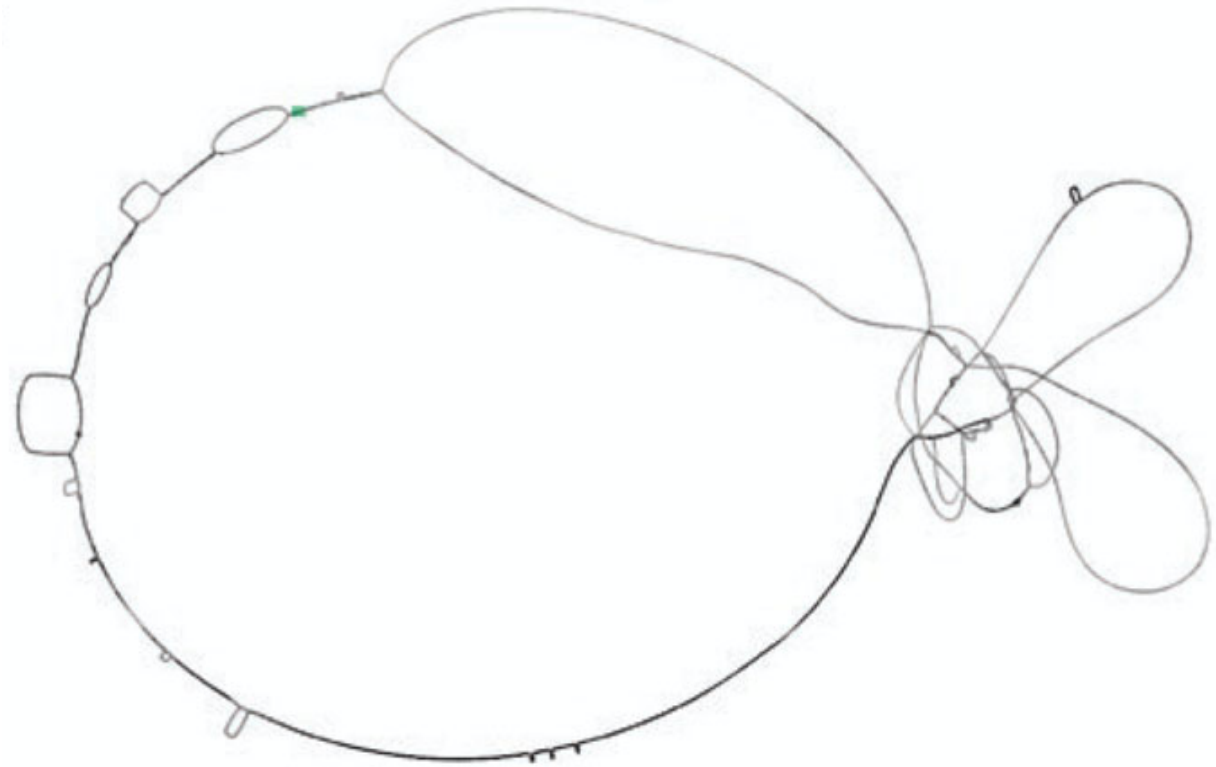


Image credit: Flicek and Birney, *Nature Methods* 6(11s):S6-S12, (2009)

The final graph: A real example

- Artificial mixing of two similar DNA sequences



Mapping and alignment

- It is not very common to perform *de novo* assembly nowadays
  - Reason 1: Current high-throughput technologies give very short reads (about or less than 100nt). The resulting de Bruijn graphs are very hard to simplify
    - For example, it is hard to deal with repeat regions
  - Reason 2: Databases contain many DNA sequences that can be used as references

- Instead, it is more common to perform mapping by means of alignment
  - Aligning many (billions of) short reads to a long reference
  - Perform assembly only when a good reference is not available

Common short-read alignment methods

- BFAST
- BOWTIE
- BWA
- ELAND
- Maq
- SHRiMP, SHRiMP2
- SOAP, SOAP2, SOAP3
- ...

Mapping short reads to a reference genome

- Basically, a local sequence alignment problem
  - Dynamic programming?
    - Reference too long (~3 billion for human)
    - Too many reads (up to billions)
  - BLAST/FASTA?
    - Still too slow
    - No need to be so flexible. Can be more stringent by allowing fewer mismatches
  - Faster data structures
    - Hash tables (similar to what BLAST and FASTA use)
    - Suffix tree/trie/array
    - Burrows Wheeler Transform (BWT)

Efficient data structures

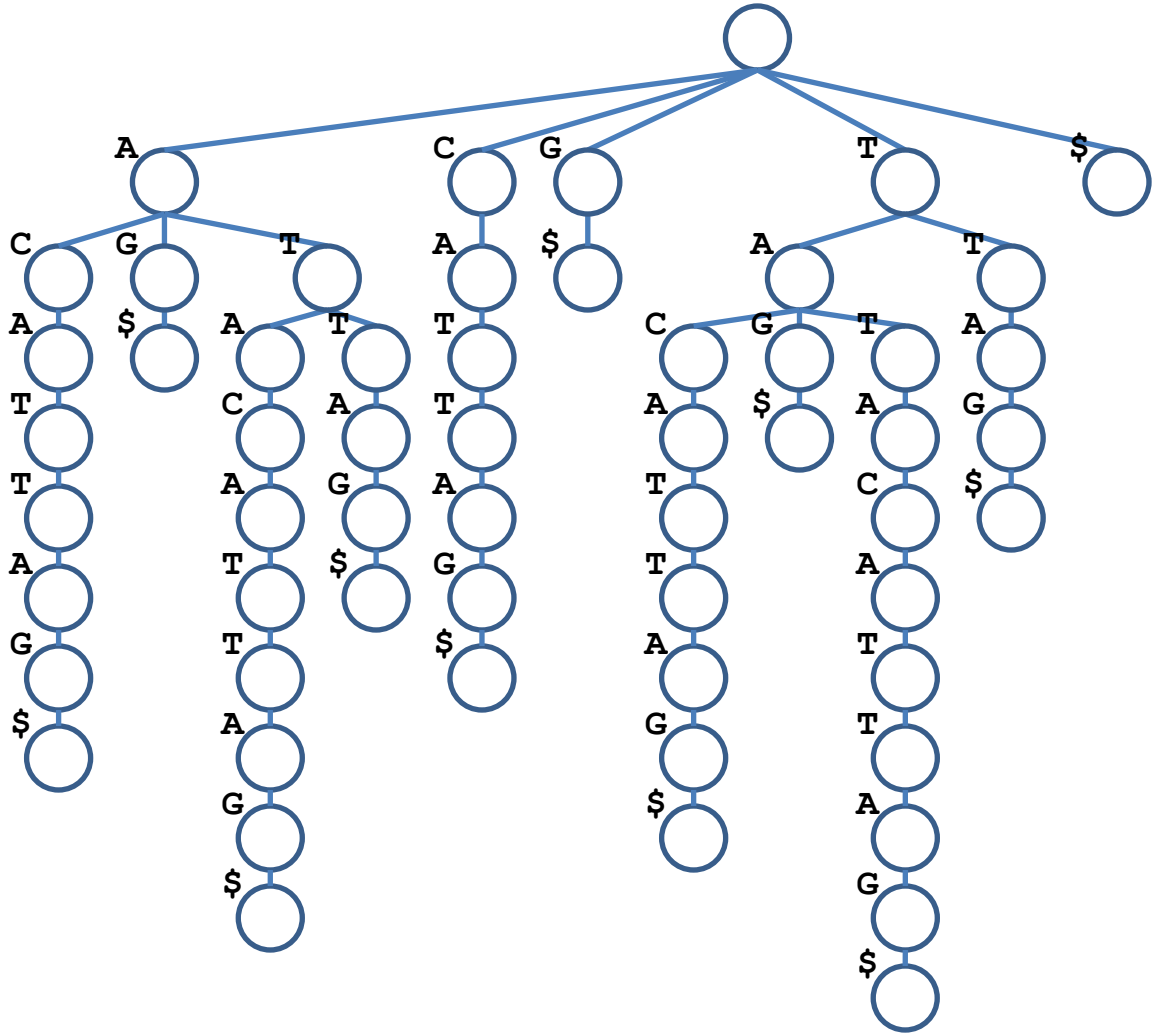
- The key to these fast methods is to construct an index of the reference and/or the reads, so that near-exact searching can be very efficient
  - Similar to an index/glossary at the end of a book

- Example: Suppose we want to check whether a short substring appears in the string `TATACATTAG$` (the `$` symbol indicates the end of the string)
  - `CAT`? Yes: Position 5
  - `GAG`? No

An example data structure: Suffix trie

- *s*=TATACATTAG$
- Suffixes:

TATACATTAG$
ATACATTAG$
TACATTAG$
ACATTAG$
CATTAG$
ATTAG$
TTAG$
TAG$
AG$
G$
$



SAM file format (for alignment results)

- Alignment:

```
Coor    12345678901234  5678901234567890123456789012345
ref     AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT
+r001/1       TTAGATAAAGGATA*CTG
+r002         aaaAGATAA*GGATA
+r003       gcctaAGCTAA
+r004                     ATAGCT..............TCAGC
-r003                           ttagctTAGGC
-r001/2                                      CAGCGCCAT
```

- SAM format:

```
@HD VN:1.3 SO:coordinate
@SQ SN:ref LN:45
r001 163 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002   0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003   0 ref  9 30 5H6M       *  0   0 AGCTAA   *   NM:i:1
r004   0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003  16 ref 29 30 6H5M       *  0   0 TAGGC    *   NM:i:0
r001  83 ref 37 30 9M         =  7 -39 CAGCGCCAT         *
```

  - CIGAR (Compact Idiosyncratic Gapped Alignment Report) string:
    M: alignment match; S: substitution (mismatch); I: insertion; D: deletion, etc.

Example source: http://samtools.sourceforge.net/SAM1.pdf

Other common file formats

- See http://genome.ucsc.edu/FAQ/FAQformat.html for a list of commonly used file formats in addition to FASTQ and SAM

Part 3

Transcriptomics

Gene expression

- We have only one genome, but why do we have many different types of cells?
  - Expression of different genes:
    - Amount of RNAs produced
    - Amount of proteins produced
- Examining gene expression is a first step to understanding function
  - Also a way to see what has gone wrong (e.g., in cancer cells)
- As explained before, gene expression is regulated by a complex set of mechanisms

RNA level vs. protein level

- For protein-coding genes, activity is best reflected by protein abundance
  - However, it is difficult to measure
- Instead, amount of RNAs produced is usually used as a proxy of protein abundance
  - RNA level does not perfectly correlate with protein level due to post-transcriptional regulation, RNA degradation, translation efficiency, etc.
  - Transcription rate is yet another measurement.
- We will use the term “gene expression” to mean RNA level (number of copies of an RNA in a cell)

Measuring gene expression levels

- High-throughput methods:
  - Microarrays
    - Design probes
    - Convert RNA back to DNA
      - Since DNA is more stable
      - Called complementary DNA, or cDNA
    - Hybridization
    - Fluorescent dye as measurable output
  - cDNA sequencing (RNA-seq)
    - Convert RNA back to cDNA
    - Sequence DNA
    - Map back to genome
    - Count number of reads from each gene

Microarrays

- Basic ideas:
  - We need to know the DNA sequences of the genes
  - For each gene, we design short sequences that are unique to the gene
    - Usually 25-75 nucleotides
  - When RNA is converted back to DNA, if it is complementary to a probe, it will bind to the probe – “hybridization”
    - Ideally only for perfect match, but sometimes hybridization also happens with some mismatches

Illustrations: the arrays

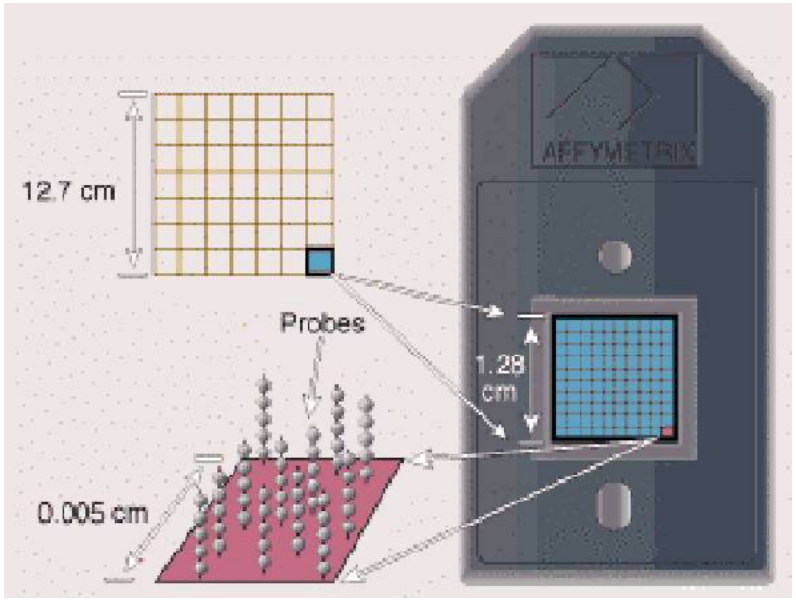


Image sources: http://www4.carleton.ca/jmc/catalyst/2006s/images/dk-PersMed3.jpg, http://bioweb.wku.edu/courses/biol566/Images/stemAffyChip.jpg

Illustrations: hybridization

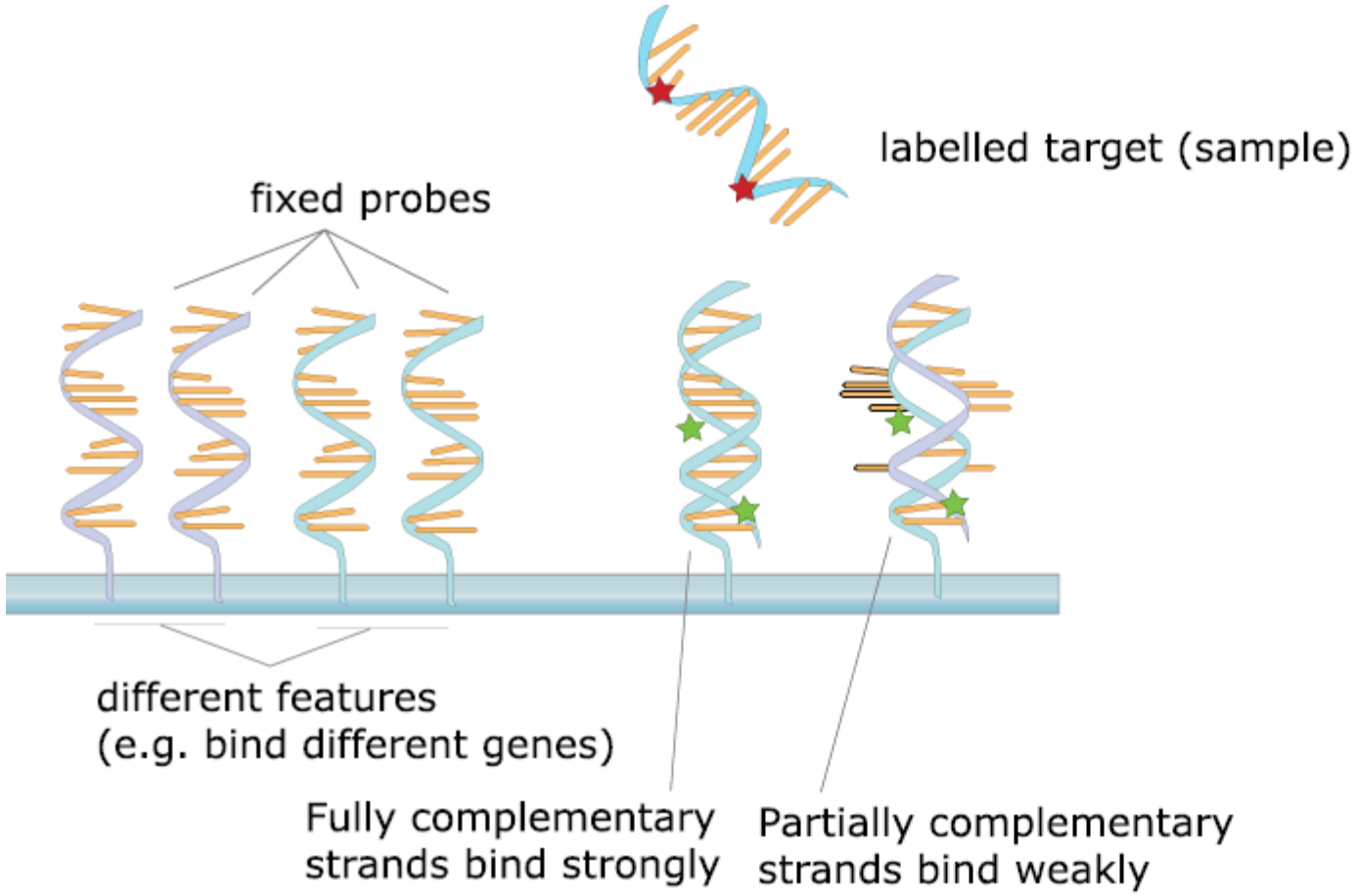


Image credit: Wikipedia

Microarrays

- Basic ideas (cont'd):
  - Detection: using florescent dye, more hybridization gives stronger signal
  - Thousands or tens of thousands of such experiments are performed at the same time by having many small wells on a solid surface, each with a different type of probes – the microarray
  - When cDNA is added to a microarray, they will hybridize to complementary probes

Illustrations: workflows

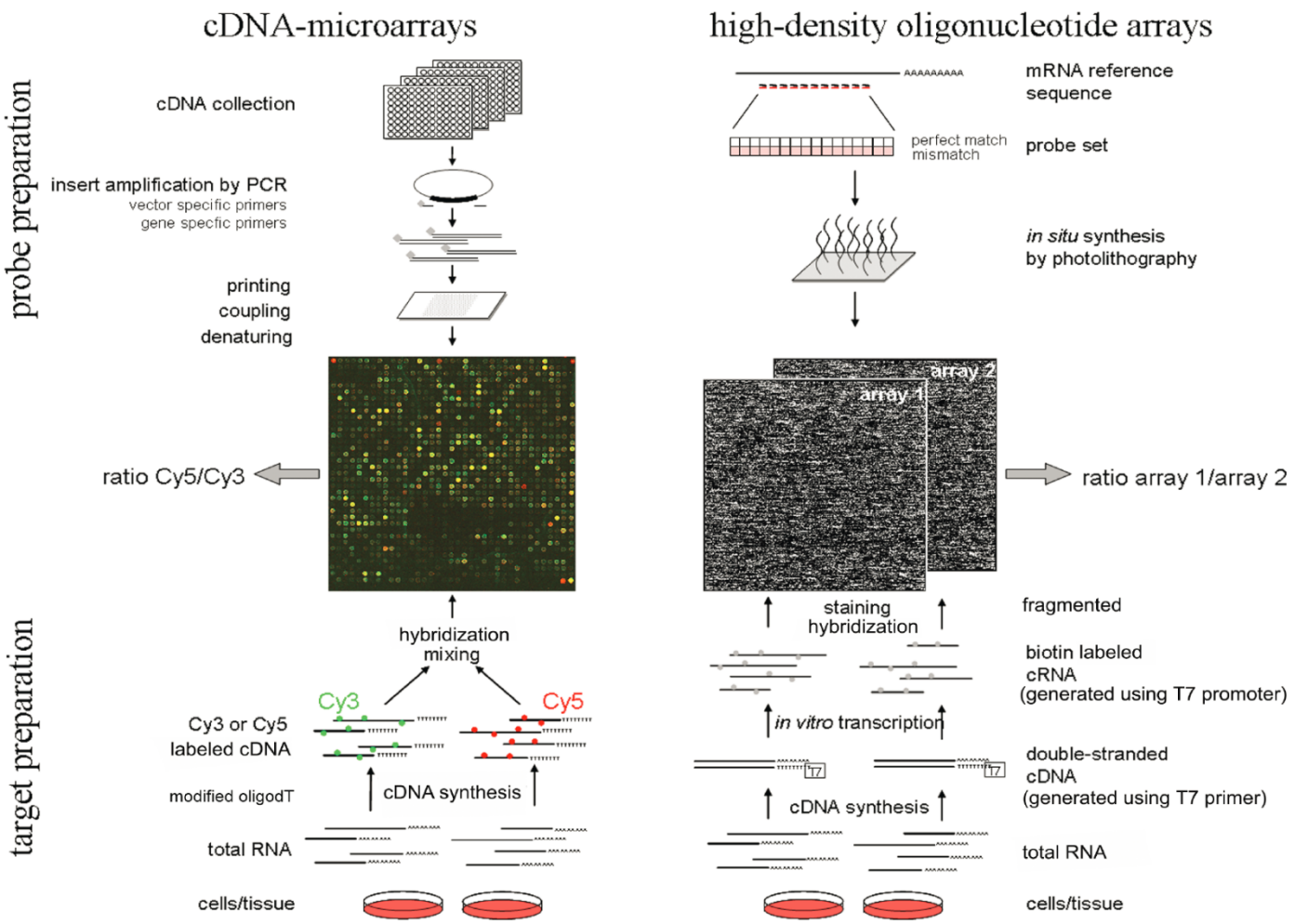


Image source: http://www.stat.berkeley.edu/users/terry/Classes/s246.2004/Week9/2004L17Stat246.pdf

Processing microarray data

- Microarray data are relatively noisy
  - Cross-hybridization (binding to an unexpected probe)
  - Background signals
  - Sensitivity to experimental condition

- There are many steps in microarray data processing. We do not go into the details
  - Combining values from different probes
  - Normalization
  - Filtering
  - ...

RNA-seq

- Convert RNAs back to cDNAs, sequence them, and identify which genes they correspond to
  - Better signal-to-noise ratio than microarrays
    - Especially important for genes with low expression
  - Wider signal range
  - No need to have prior knowledge about the sequences
  - If a sequence is not unique to a gene, cannot determine which gene it comes from
    - Also a problem for microarrays

RNA-seq

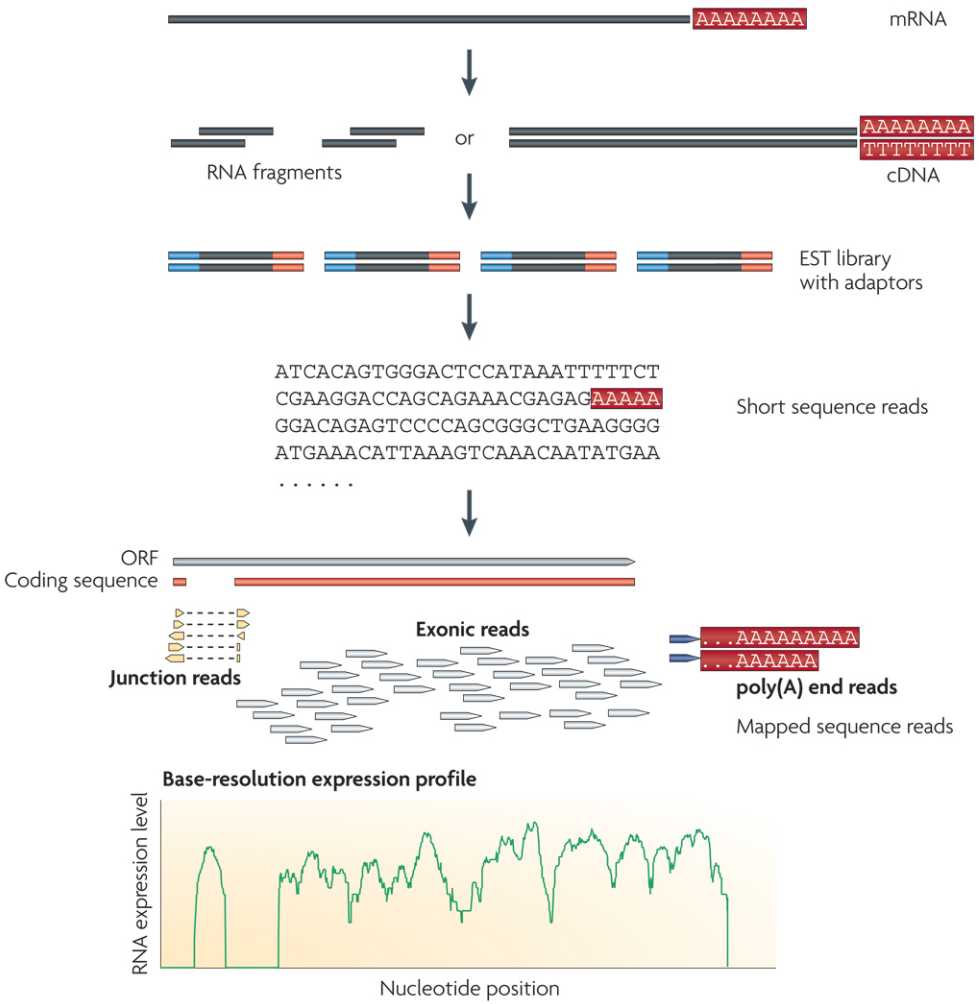


Image credit: Wang et al., *Nature Review Genetics* 10(1):57-63, (2009)

Processing RNA-seq data

- Again, many steps and we will not go into the details
  - Quality check
  - Read trimming and filtering
  - Read mapping
  - Data normalization
  - ...

Measuring expression levels

- How to compute an expression level from a distribution of read counts?
  - Calculate the average
  - Based on a statistical model
- Normalization: If expression levels of ene in different datasets are to be compared
  - Longer genes are expected to get more reads
  - For a dataset with more reads, each gene gets more reads on average
  - RPKM: Reads per Kilobase of exons per Million reads
    - There are more advanced methods

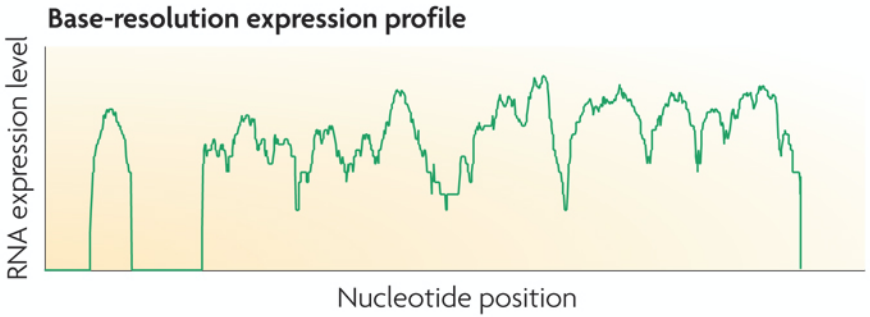


Image credit: Wang et al., *Nature Reviews Genetics* 10(1):57-63, (2009)

Analyzing gene expression data

- Ultimately, from a high-throughput gene expression experiment we get a vector of real numbers, corresponding to the expression values of the genes
- We will briefly study two types of analysis
  - Clustering
  - Classification

Clustering gene expression data

- The data:
  - A matrix of real numbers
  - Each row corresponds to a gene
  - Each column corresponds to a sample/experiment:
    - A particular condition
    - A cell type (e.g., cancer)
- Questions:
  - Any genes that show similar changes of their expression levels across experiments?
  - Any samples with similar sets of genes expressed?

Hierarchical clustering

- There are many clustering algorithms
- We have learned one in phylogenetic tree reconstruction: UPGMA
  - A hierarchical clustering algorithm
  - In each step, we merge two clusters that are most similar, until all clusters have been merged into one
- Can apply the same idea in clustering gene expression data
  - Of course here branch lengths do not mean evolutionary time

Two-way hierarchical clustering

- Things to consider when clustering gene expression data:
  - We want to both cluster genes and cluster samples – two-way clustering
  - May use either distance matrix or similarity matrix
    - Euclidean distance (if absolute expression levels matter):

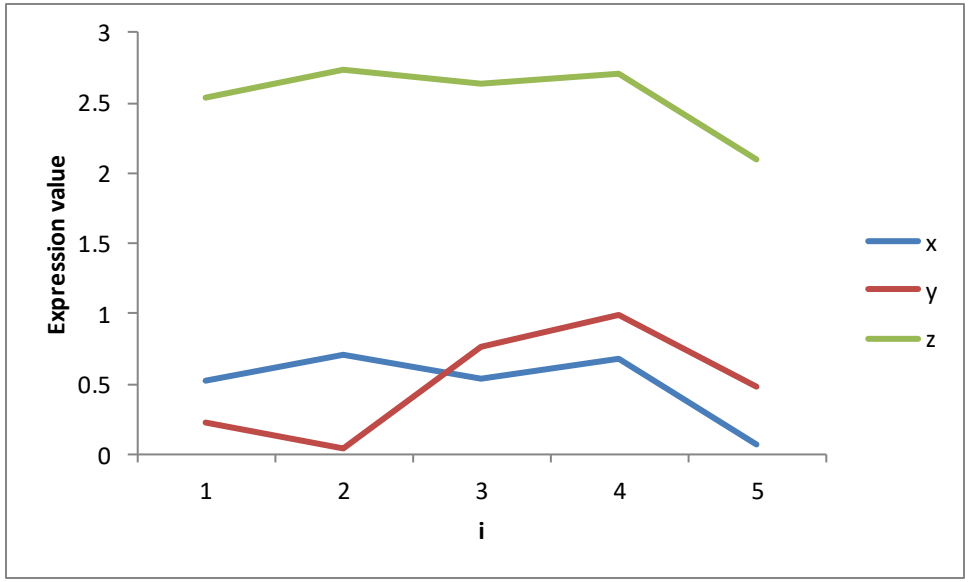
$$d(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$$

    - Pearson correlation (if only the trend matters):

$$r(x, y) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y_i - \bar{y})^2}}$$

      - Between -1 and 1
      - Positive: correlated; negative: anti-correlated; 0: uncorrelated

Absolute distance vs. correlation



- d(x, y) = 0.92 < d(x, z) = 4.56
  - x is closer (i.e., more similar) to y than z
- r(x, y) = 0.01 < r(x, z) = 0.99
  - x is more similar to z than y

The whole procedure

1. Compute the distance/similarity between every pair of rows, using the columns as features
2. Use the distance/similarity matrix to perform a hierarchical clustering of the rows
3. Compute the distance/similarity between every pair of columns, using the rows as features
4. Use the distance/similarity matrix to perform a hierarchical clustering of the columns

An example: different subtypes of cancer

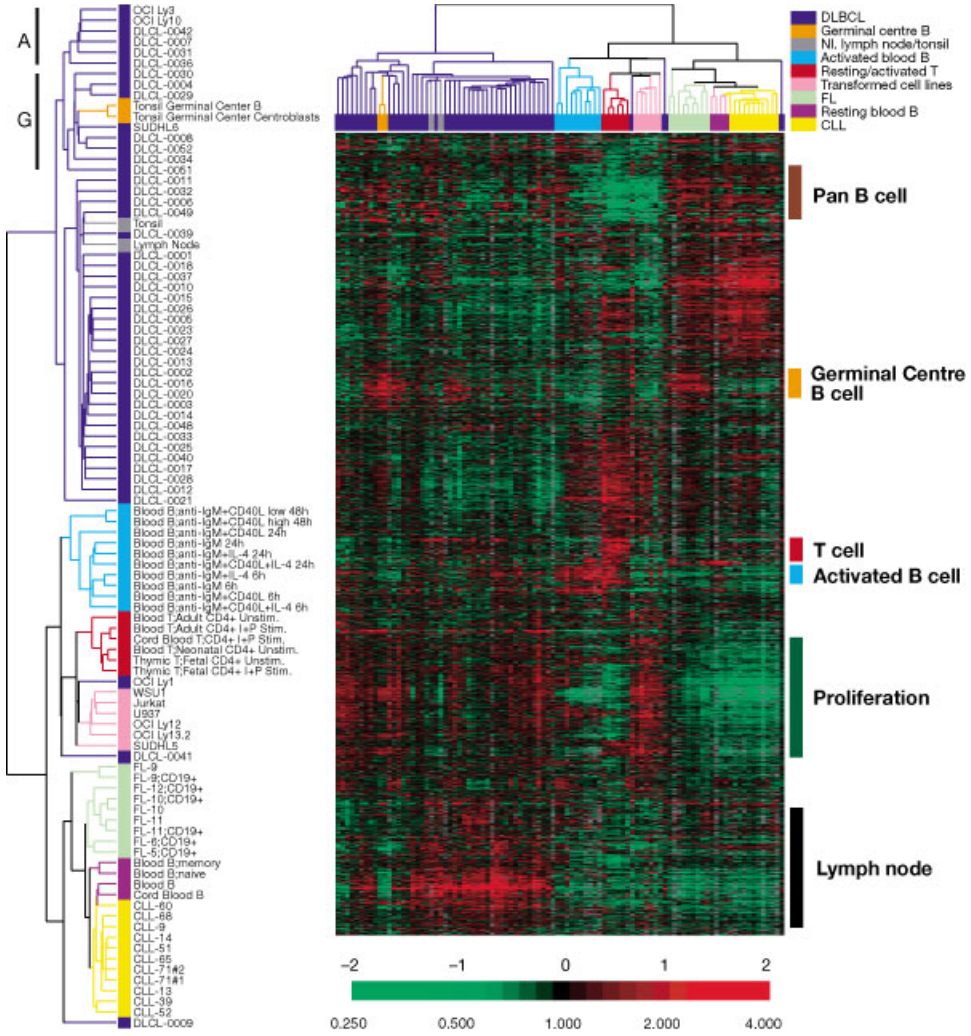


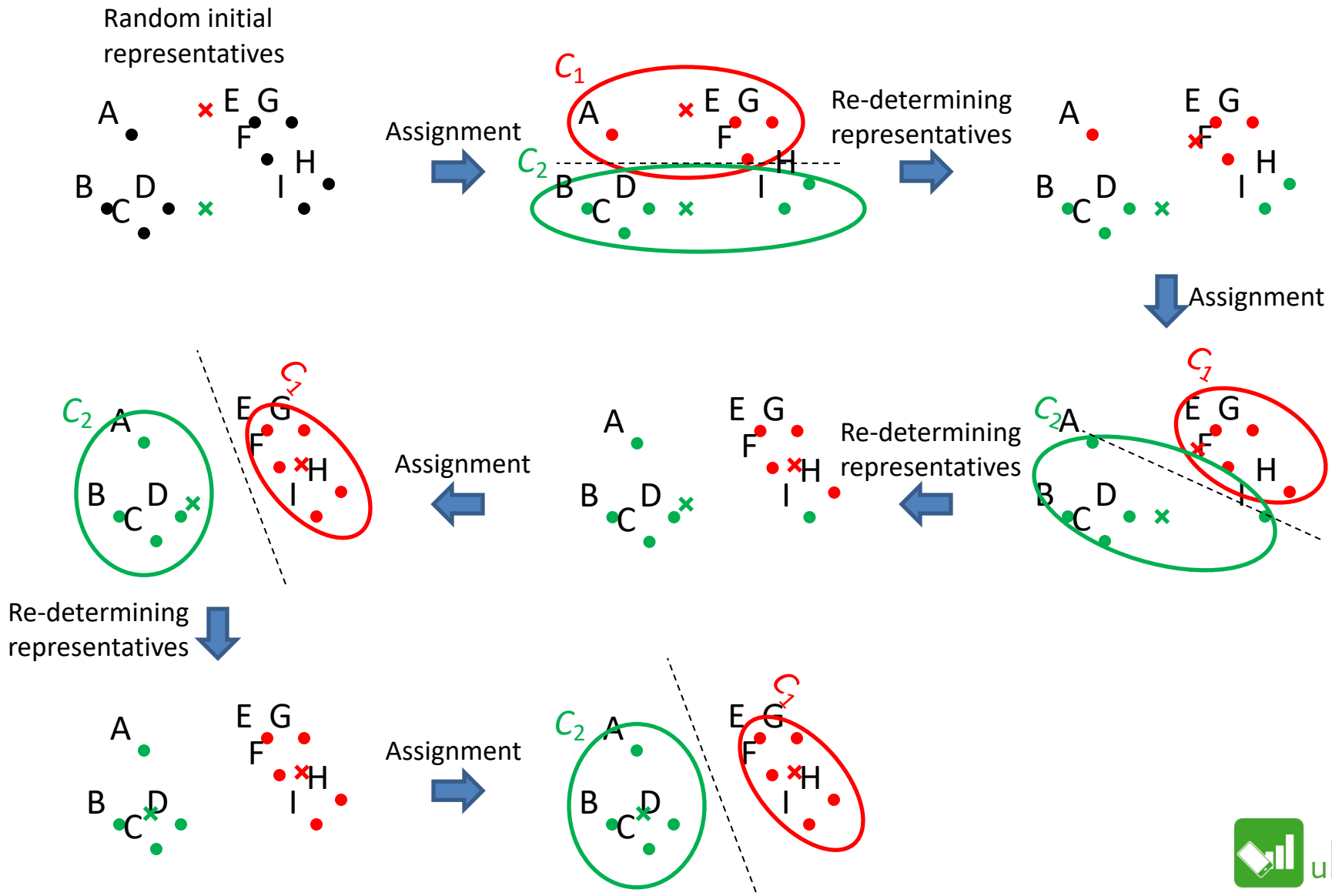
Image credit: Alizadeh et al., *Nature* 403(6769):503-533, (2000)

- Note the separation of different types of samples (labeled by different colors)

K-means

- K-means is another classical clustering algorithm
  - MacQueen, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability 281-297, (1967)
- Instead of hierarchically merging clusters, k-means iteratively partitions the objects into *k* clusters by repeating two steps until stabilized:
  1. Determining cluster representatives
     - Randomly determined initially
     - Centroids of current members in subsequent iterations
  2. Assigning each object to the cluster with the closest representative

Example (k=2)


Random initial representatives
Assignment
Re-determining representatives
Assignment
Re-determining representatives
Assignment
Re-determining representatives
Assignment
C_1
C_2

Unsupervised vs. supervised learning

- In clustering analysis, we grouped objects (genes or samples) together purely based on expression values
  - Did not use any knowledge about the samples (e.g., type of cancer)
  - "Unsupervised" learning
- Alternatively, one may try to build a model that can distinguish different types of genes/samples, using known labels
  - "Supervised" learning
  - Goal: Given a new object without label, use the model to predict it

Classification: supervised learning

- If the target is discrete labels, the supervised learning problem is called classification
- There are many methods for performing classification
  - Bottom line: You need to know the difference between clustering and classification
  - We will introduce one very simple method for classification

Nearest neighbor

- One simple method is to predict the label of the object as the same as that of the most similar object
  - Again, similarity can be measured by different ways
  - Can generalize to consider k nearest neighbors instead of the single most similar one – let them vote

Example

- Suppose we have this set of gene expression data:

	S1	S2	S3	S4	S5
Gene 1	1	2	4	3	3
Gene 2	2	5	2	9	5
Gene 3	2	3	6	1	4
Gene 4	5	7	1	2	6
Cancer type	A	A	B	B	?

- Similarity Between S5 and the other samples based on Pearson correlation:

	S1	S2	S3	S4
Correlation with S5	0.89	0.99	-0.76	0.18

- Therefore, S5 is predicted to be of type A (due to S2)

An example

- Predicting the survival of medulloblastoma (a form of brain tumour) patients (mainly children)
- Method: a variation of k nearest neighbors

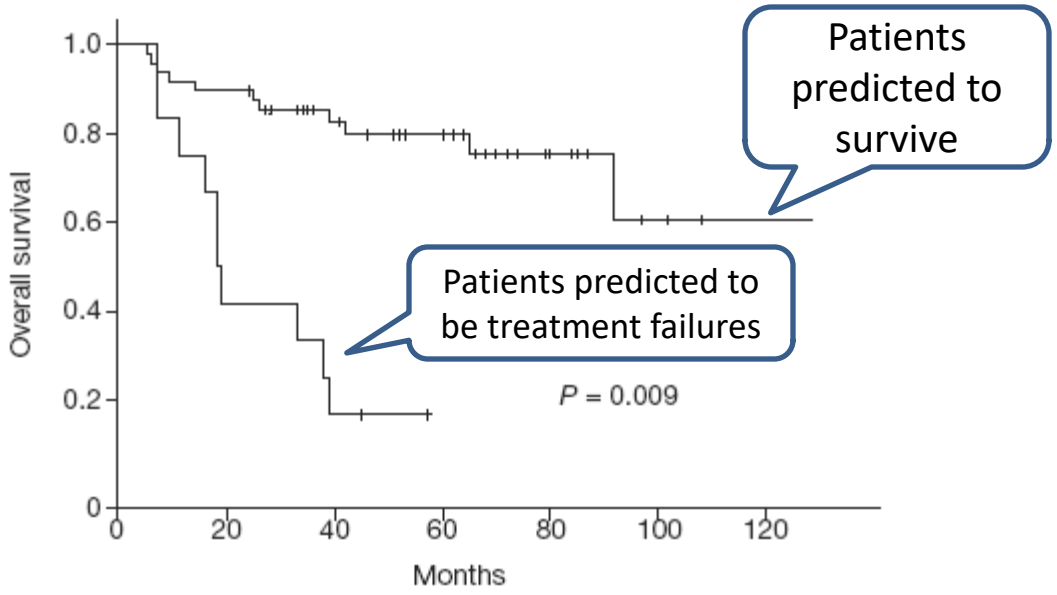


Image credit: Pomeroy et al., *Nature* 415(6870):436-442, (2002)

Epilogue

Summary and Further Readings

Further readings

- SEQAnswers: An important online forum for discussing almost everything about high-throughput sequencing
- There are many good review and assessment papers. For example:
  - Sequence assembly: Compeau et al., How to Apply de Bruijn Graphs to Genome Assembly. *Nature Biotechnology* 29(11):987-991, (2011)
  - RNA-seq: Marioni et al., RNA-seq: An Assessment of Technical Reproducibility and Comparison with Gene Expression Arrays. Genome Research 18(9):1509-1517, (2008)
  - Differential expression: Dillies et al., A Comprehensive Evaluation of Normalization Methods for Illumina High-Throughput RNA Sequencing Data Analysis. *Briefings in Bioinformatics* doi:10.1093/bib/bbs046, (2012)