Week 8

- *Maximum Likelihood trees*
- *Genome Assembly*
  - Whole genome shotgun assembly
  - DNA sequencing
  - Kmers
  - de Bruijn graph assembly
  - Overlap-Layout-Consensus assembly
  - Repeats
  - Scaffolding
- *Gene Modeling*
  - *Ab initio* gene prediction
    - Markov models
    - Hidden Markov models
- Comparative gene modeling

Multiple Alignment and Trees

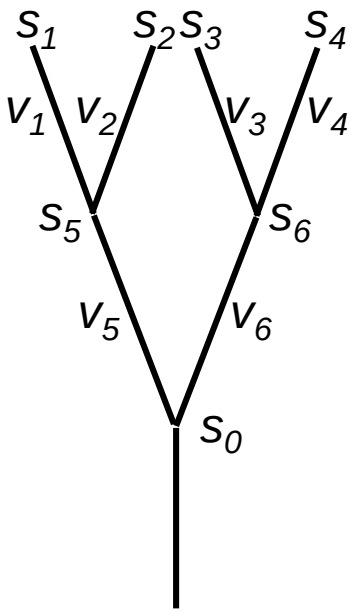
Maximum Likelihood Methods

- *Estimate topology and branch length viewing evolution as a random process*
- *Requires a probability model of evolution as a function of time.*
  - For DNA one can use Jukes-Cantor model (all nucleotides have same substitution rates), or Kimura model (different rates for transitions, R→R or Y →Y, and transversion, R→Y or Y →R).
  - For proteins one can use Dayhoff, but in the probability form not the log-odds form.
  - most common is the general time-reversible (GTR) model for an equilibrium base frequency vector $\vec{\pi} = (\pi_1, \pi_2, \pi_3, \pi_4)$

$$Q = \begin{pmatrix} -(x_1 + x_2 + x_3) & x_1 & x_2 & x_3 \\ \frac{\pi_1 x_1}{\pi_2} & -(\frac{\pi_1 x_1}{\pi_2} + x_4 + x_5) & x_4 & x_5 \\ \frac{\pi_1 x_2}{\pi_3} & \frac{\pi_2 x_4}{\pi_3} & -(\frac{\pi_1 x_2}{\pi_3} + \frac{\pi_2 x_4}{\pi_3} + x_6) & x_6 \\ \frac{\pi_1 x_3}{\pi_4} & \frac{\pi_2 x_5}{\pi_4} & \frac{\pi_3 x_6}{\pi_4} & -(\frac{\pi_1 x_3}{\pi_4} + \frac{\pi_2 x_5}{\pi_4} + \frac{\pi_3 x_6}{\pi_4}) \end{pmatrix}$$

  - x_1 = *rate*(A → G) = r(G → A) so we can write r(A ↔ G)
  - x_2 = r(A ↔ C), x_3 = r(A ↔ T), x_4 = r(G ↔ C), x_5 = r(G ↔ T), x_6=r(C ↔ T)

Maximum Likelihood Methods



s_1 etc. are the bases or residues in the extant and ancestral taxa. *only* s_1, s_2, s_3, and s_4 are observable

$v = \lambda t$ where λ is the substitution rate and t is absolute time

$P_{i,j}(v)$ is the probability that the residue at node s_i becomes residue at node s_j in time v

g_0 is the prior probability of the bases or nucleotides at any position

$$L = g_0 P_{0,5}(v_5)\, P_{5,1}(v_1)\, P_{5,2}(v_2)\, P_{0,6}(v_6)\, P_{6,3}(v_3)\, P_{6,4}(v_4)$$

Maximum Likelihood

- $L = g_0 P_{0,5}(v_5)\, P_{5,1}(v_1)\, P_{5,2}(v_2)\, P_{0,6}(v_6)\, P_{6,3}(v_3)\, P_{6,4}(v_4)$
- *We don't know the identity of the bases or residues at s5, s6, or s0 so the likelihood must be summed over all possibilities:*

$$L = \sum_{s_0} \sum_{s_5} \sum_{s_6} g_0 P_{0,5}(v_5)\, P_{5,1}(v_1)\, P_{5,2}(v_2)\, P_{0,6}(v_6)\, P_{6,3}(v_3)\, P_{6,4}(v_4)$$

- *Felsenstein suggests a simple model:*
  $P_{i,j} = e^{-v\delta_{i,j}} + (1-e^{-v})g_j$ *where* $\delta_{i,j}$ *is 1 when i=j and 0 otherwise*
- *Must vary all v to maximize likelihood and the try numerous topologies to find the highest likelihood tree*

Maximum Likelihood

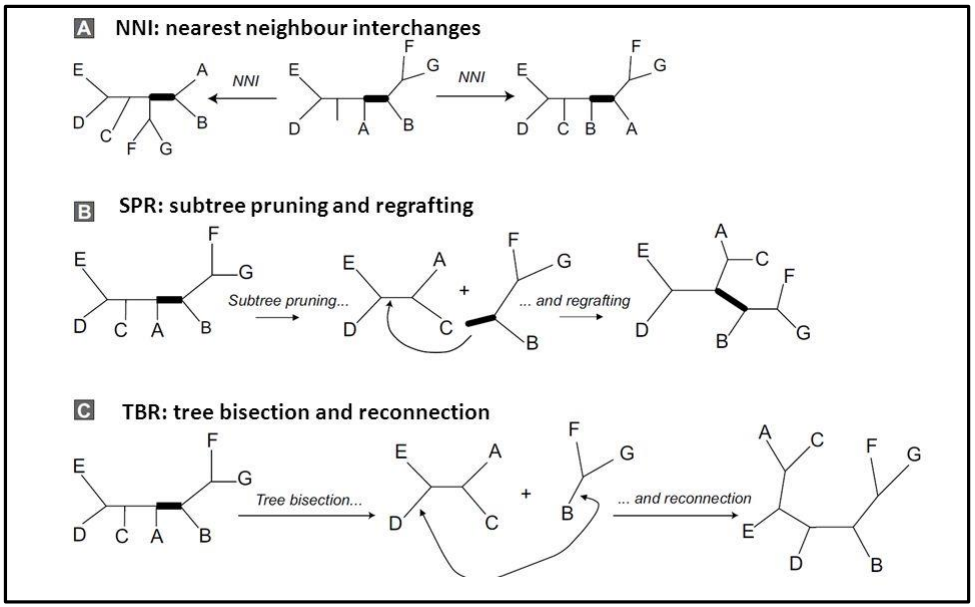
- *Problems*
  - The value of *v* varies with position due to conserved and unconserved regions of the sequences
  - The probability of a substitution $P_{i,j}$ is also position specific
  - both must be must be estimated for every site
  - Every tree topology must be evaluated separately

  - many parameters to estimate
    - very high computational requirements
    - must have a lot of data or estimates are unreliable

Multiple Alignment and Trees

Problem

- *Too many trees to exhaustively check every topology*
- *Parsimony and ML (and Bayesian methods) do not create the topology*
- *No exact solution*
  - Start from a close tree (e.g., from a distance method)
  - make small rearrangements and see if the tree quality improves (likelihood ratio test), i.e., the tree with the highest probability of generating the observed data (sequences) is preferred
  - subtree pruning and regrafting
  - Tree bisection and reconnection



Multiple Alignment and Trees

Distance methods (UPGMA, FM, & N-J)

Optimality criterion

- *in constructing tree – none*
- *for evaluating tree – sum of squared distances between data and tree.*

Advantages:

- *Can be used on indirectly-measured distances (immunological, hybridization).*
- *Distances can be 'corrected' for unseen events.*
- *The fastest of the methods available (N-J is very fast! UPGMA is faster).*
- *Can therefore analyze very large datasets quickly (needed for Coronavirus, etc.).*
- *Can be used for some types of rate and date analysis.*

Disadvantages:

- *Similarity and relationship are not necessarily the same thing*
- *Clustering by similarity does not necessarily give a cladistictree.*
- *Cannot be used for morphological character analysis!*
- *Greedy approach may not find the global optimum*

Multiple Alignment and Trees

Parsimony methods

Optimality criterion

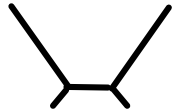
- *in constructing tree – none*
- *for evaluating tree – the fewest number of evolutionary events (e.g., nucleotide substitutions, amino acid replacements) required to explain the sequences.*

Advantages:

- *Simple, intuitive, and logical (many possible by 'pencil-and-paper').*
- *Can be used on molecular and non-molecular (e.g., morphological) data.*
- *Can tease apart types of similarity (shared-derived, shared-ancestral, homoplasy)*
- *Can be used for character and rate analysis.*
- *Can be used to infer the sequences of the extinct (hypothetical) ancestors.*

Disadvantages:

- *Are simple, intuitive, and logical (derived from "Medieval logic", not statistics!)*
- *Can be fooled by high levels of homoplasy (multiple substitutions at the same site).*
- *Can become positively misleading in the "Felsenstein Zone" (long branches attract)*
- *How do you find the tree topology?*



Multiple Alignment and Trees

Maximum likelihood (ML) methods

Optimality criterion

- *Probability that a proposed model of the evolutionary process and the proposed unrooted tree would give rise to the observed data.*
- *The tree found to have the highest ML value is considered to be the preferred tree.*

Advantages:

- *Inherently statistical and evolutionary model-based.*
- *Usually the most 'consistent' of the methods available.*
- *Can be used for character (can infer the exact substitutions) and rate analysis.*
- *Can be used to infer the sequences of the extinct (hypothetical) ancestors.*
- *Can help account for branch-length effects in unbalanced trees*
- *Can account for differences in evolutionary rate*
- *Can be applied to nucleotide or amino acid sequences, and other types of data.*

Disadvantages:

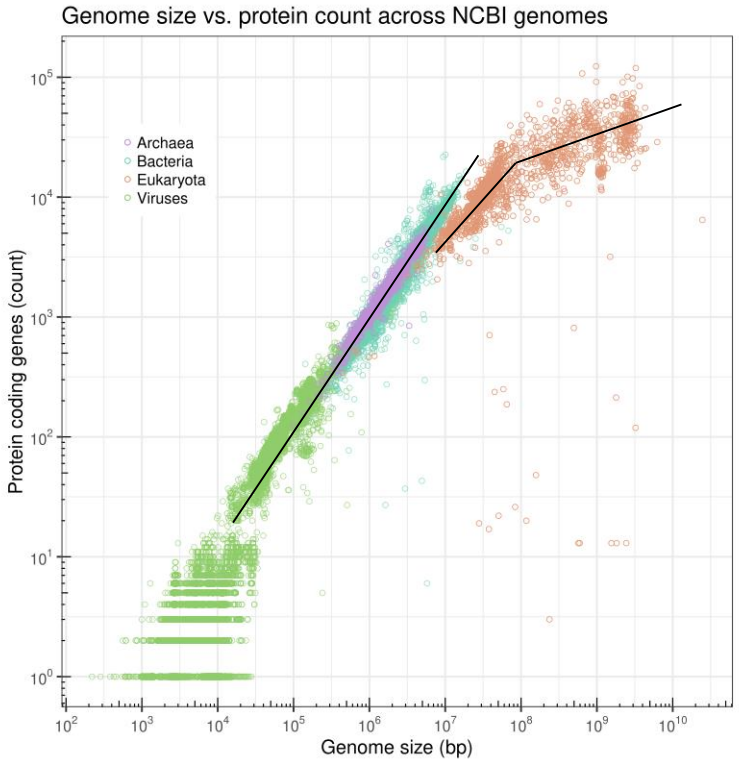
- *Not as simple and intuitive as many other methods.*
- *Computationally very intensive (Iimits number of taxa and length of sequence).*
- *Like parsimony, can be fooled by high levels of homoplasy.*
- *Violations of the assumed model can lead to incorrect trees.*
- *How do you find the tree topology?*

Genome Sequencing and Annotation

Genome Sizes

- *Genome size does not correlate with evolutionary status, nor is the number of genes proportional to genome size. (C-value paradox)*

Species	Genome	Genes	Genes/100kb
Protopterus aethiopicus (marbled lungfish)	133 Gb	19181	0.01
Neoceratodus forsteri (giant lungfish)	27.1 Gb	31120	0.1
Picea abies (Norway spruce)	19.6 Gb	28534	0.1
Triticum aestivum (wheat)	16.8 Gb	95000	0.6
Alsophila spinulosa (tree fern)	6230 Mb	67831	1.1
Homo sapiens (human)	3200 Mb	87306	2.7
Mus musculus (mouse)	2588 Mb	68968	2.7
Zea mays B73 (corn)	2365 Mb	39756	1.7
Balaenoptera musculus (blue whale)	2105 Mb	52259	2.5
Gymnosporangium confusum (rust fungus)	893.2 Mb	15722	1.8
Mycena galopus (wood fungus)	211.4 Mb	49684	23.5
Drosophila melanogaster (fruit fly)	138.9 Mb	30717	22.1
Arabidopsis thaliana (plant)	120.1 Mb	27334	22.8
Caenorhabditis elegans (roundworm)	103.3 Mb	28546	27.6
Aspergillus fumigatus (fungus)	28.5 Mb	9398	32.9
Saccharomyces cerevisiae (yeast)	11.8 Mb	5410	45.8
Escherichia coli (bacterium)	4.64 Mb	4298	92.6



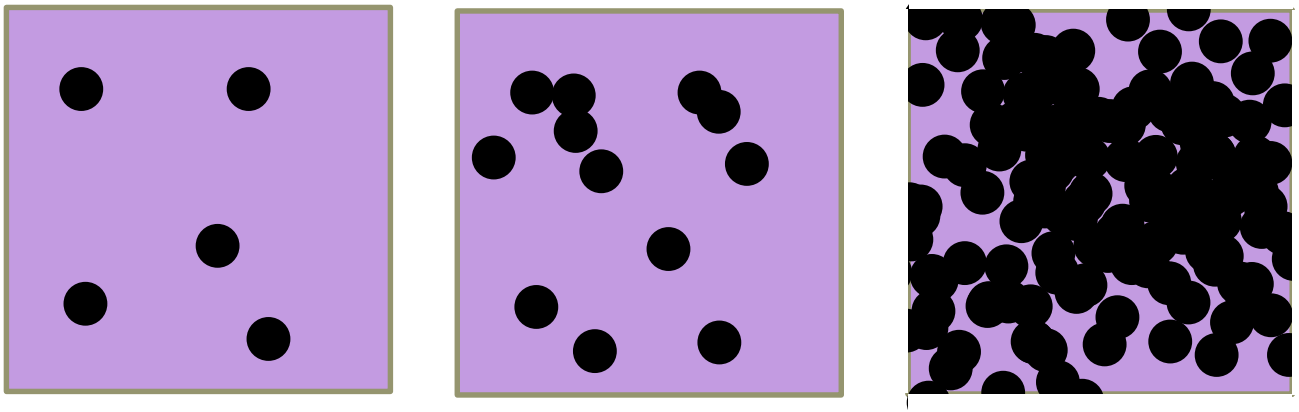
Genome Assembly

- *Original plan for human genome*
  - Isolate chromosomes
  - clone in Bacterial Artificial Chromosome (BAC) vectors (~200kb inserts)
  - Find "golden path" through BACs to minimize sequencing (tiling)
  - Subclone BACs into plasmids
  - Sequence using dideoxy chain termination (Sanger) sequencing
  - Optimistically estimated to take $3 billion and take 15 years
  - Initiated in 1990 – claimed to be the largest collaborative science project ever

Genome Assembly

Like raindrops falling on a sidewalk

- *In principle it takes infinite drops*
  - in reality no, both the sidewalk and the drops have finite size



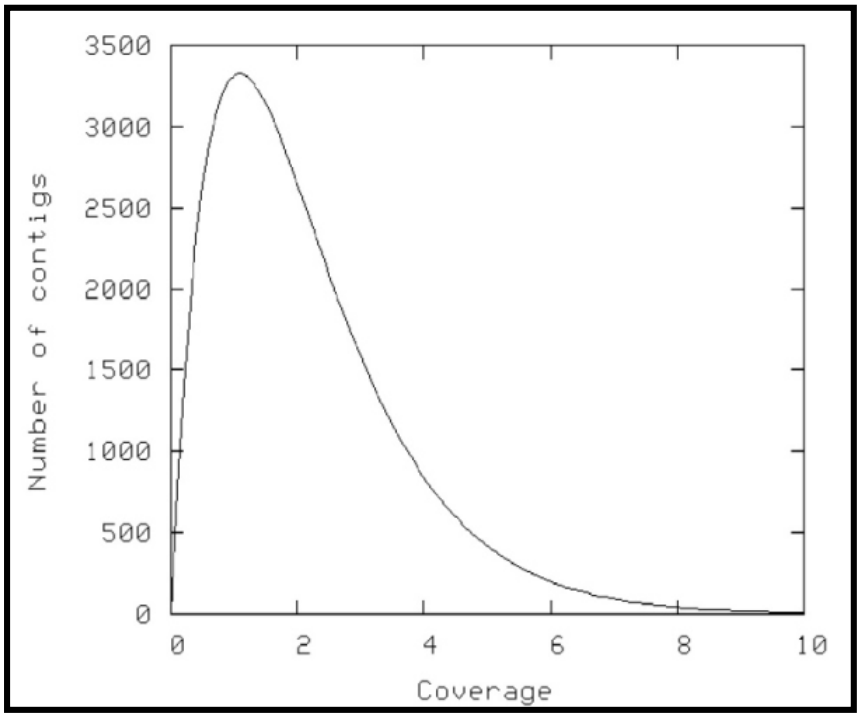
  - At first none overlap, then a few overlap, and finally almost all overlap

Genome Assembly

How much sequence do you need - 1988

- Lander ES, Waterman MS, Genomic mapping by fingerprinting random clones: a mathematical analysis, Genomics 2: 231-239 (1988)

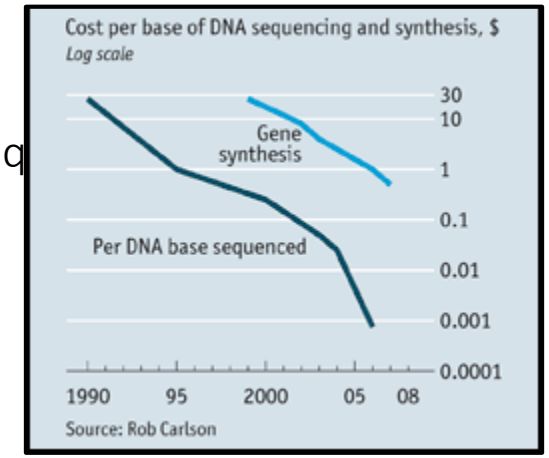
- *Depends on*
  - genome size (G)
  - sequence length (L)
  - number of sequences (N)
- *coverage = L N / G*
- *In 1st generation sequencing 16 X coverage was a good target*
- *Today >100 X is common*
- *For the human genome*
  - 15 X coverage
  - 500 base reads
  - 3.3 x 10e9 bp
  - 99 million reads = $ 3 billion @ $30/base,
    = $9.9 million @ $0.10/base



Genome Assembly

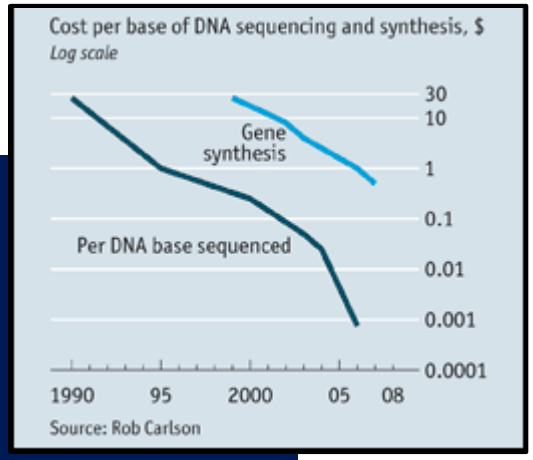
Whole Genome Shotgun (WGS) Assembly

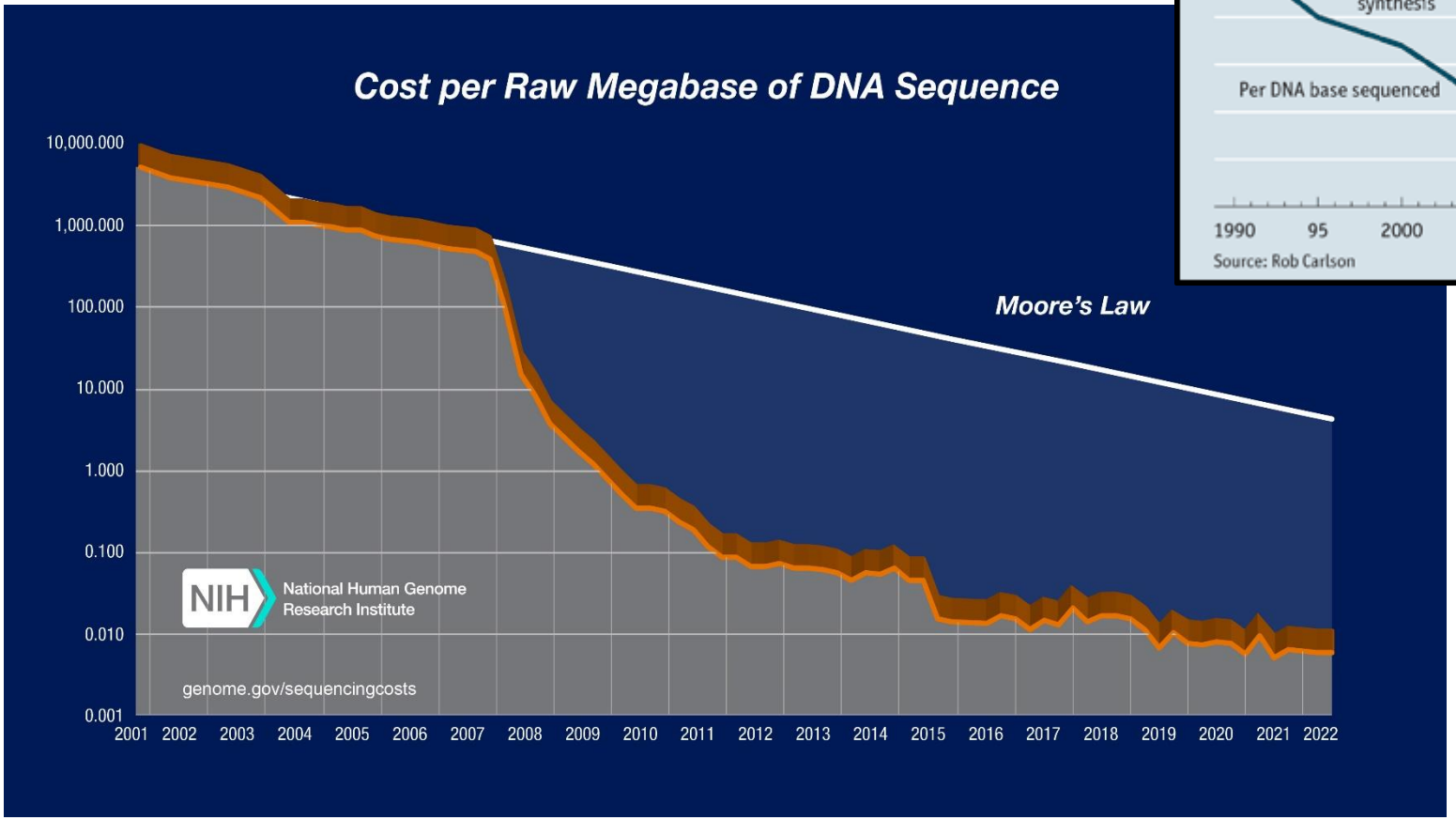
- *1998 – Craig Venter*
  - Why mess around with all the subcloning and tiling
  - why not just fragment the whole genome randomly and seq
    Whole Genome Shotgun (WGS)
- *1998 – NIH*
  - It'll never work
  - You have to sequence too many clones
  - You won't be able to put it together
- *2000 – Celera completes draft sequence using WGS approach*
  - Funding $300 Million



Sequencing Basics

Sequencing is now (extremely) cheap





Cost by Platform (2022)

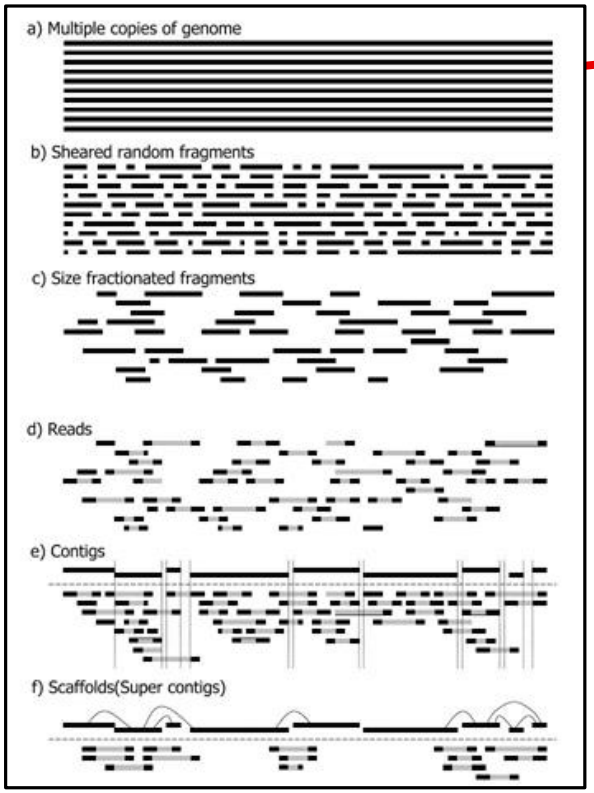
- *currently installed sequencers can sequence 7.5 x 10^6 gigabases/day!*
- *about 85% is Illumina short read sequence*

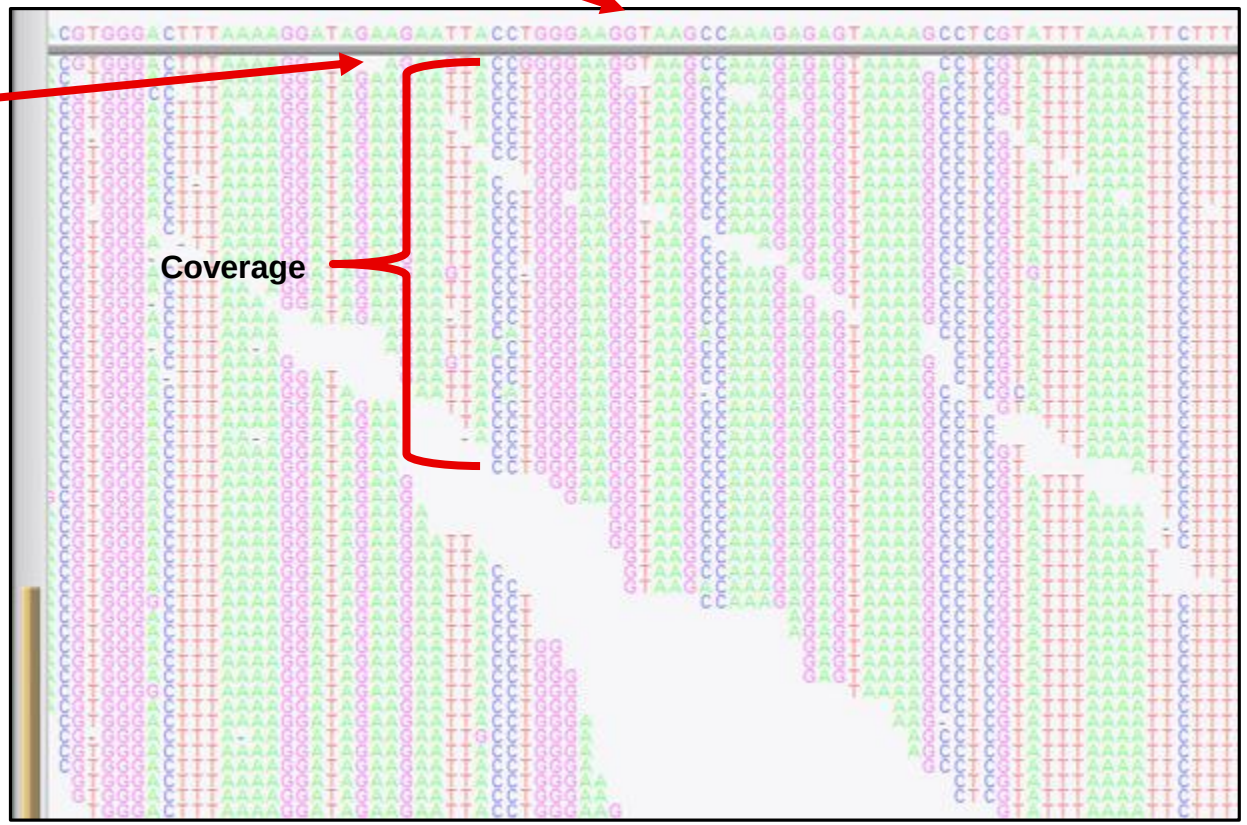
Platform	Read length	Yield (Gb)	Gb/d	$ / Gb	Installed		
Short Read							
ILMN iSeq 100 1fcell	PE150	1.2	1.56	485	1400	85%	Illumina
ILMN MiniSeq 1fcell	PE150	7.5	7.5	233.3	1700		
ILMN MiSeq 1fcell	PE300	15	7.5	118	8400		
ILMN NextSeq 550 1fcell	PE150	120	100	43.8	4000		
ILMN HiSeq 4000 2fcells	PE150	1500	400	25	400		
ILMN NextSeq 2000 P3 1fcell	PE150	360	150	17.3	1175		
ILMN NovaSeq S4 v1.5 2fcells	PE150	6000	3600	4.84	1485		
MGI DNBSEQ-T10×4RS 8fcells	PE150	70000	20000	1	6	3.2%	MGI Technologies (Complete Genomics)
MGI DNBSEQ-T7 4fcells	PE150	6000	6000	5	20		
TMO Ion S5 550 1chip	200	25	25	66.8	2220	0.7%	Ion Torrent (Thermo Fisher)
Long Read							
Singular Genomics G4 F3 4fcell (late 2022)	PE150	400	505	8	2		
PACB 8M Sequel II/IIe v2.0 chem 1 flowcell	45K-185K	180	650	7.2	577	5.0%	Pacific Biotechnologies (PacBio)
ONT MinION Mk1b 1fcell 512 channels	20-2000000	40	15	8.5	5401	6.0%	Oxford Nanopore Technologies
ONT MinION Mk1c 1fcell 512 channels	20-2000000	40	15	8.5	100		
ONT GridION X5 5fcells	20-2000000	200	75	8.5	782		
ONT PromethION 48fcells 10,700 channels	20-2000000	9600	4667	3	%67		

Genome Assembly

Short reads -> whole genome

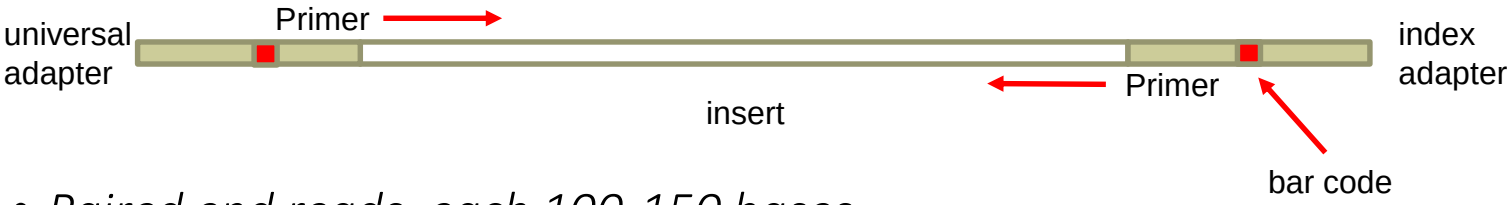
- *Reads typically 100-150 bases long*
- *Error rate ~ 0.01 – 0.1%*
- *10^8 – 10^9 reads*



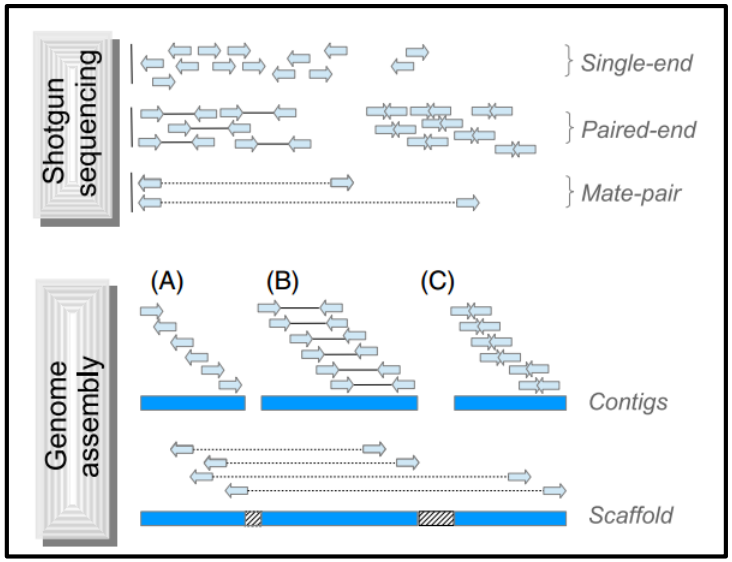


Genome Assembly

Illumina TruSeq System

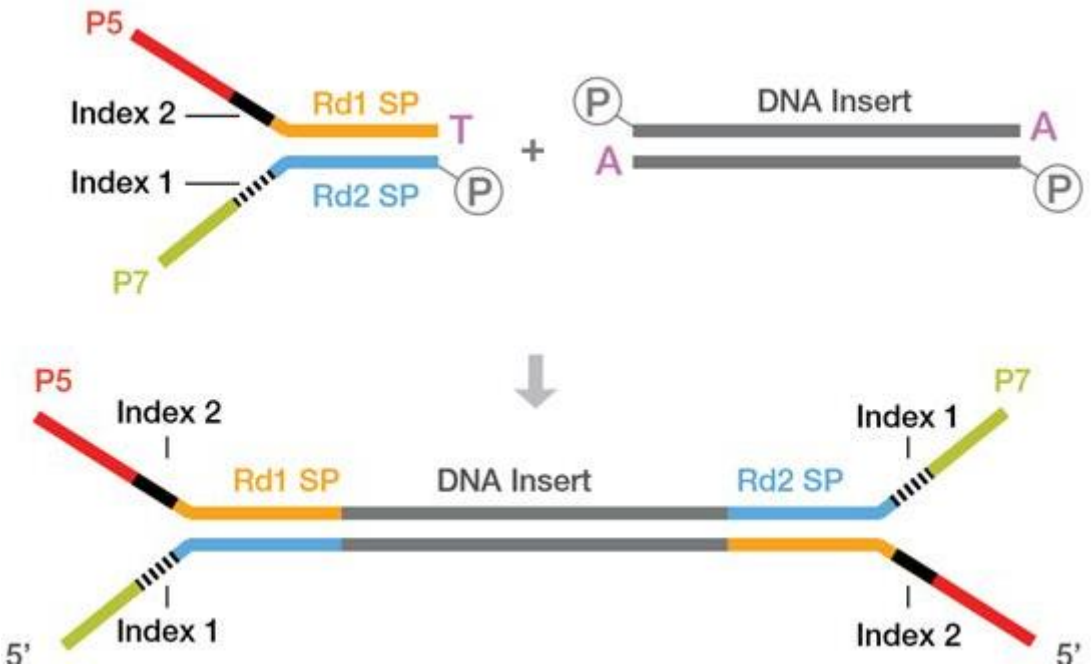


- *Paired end reads, each 100-150 bases*
- *Vocabulary*
  - paired-end
  - mate-pair
  - contig
  - scaffold



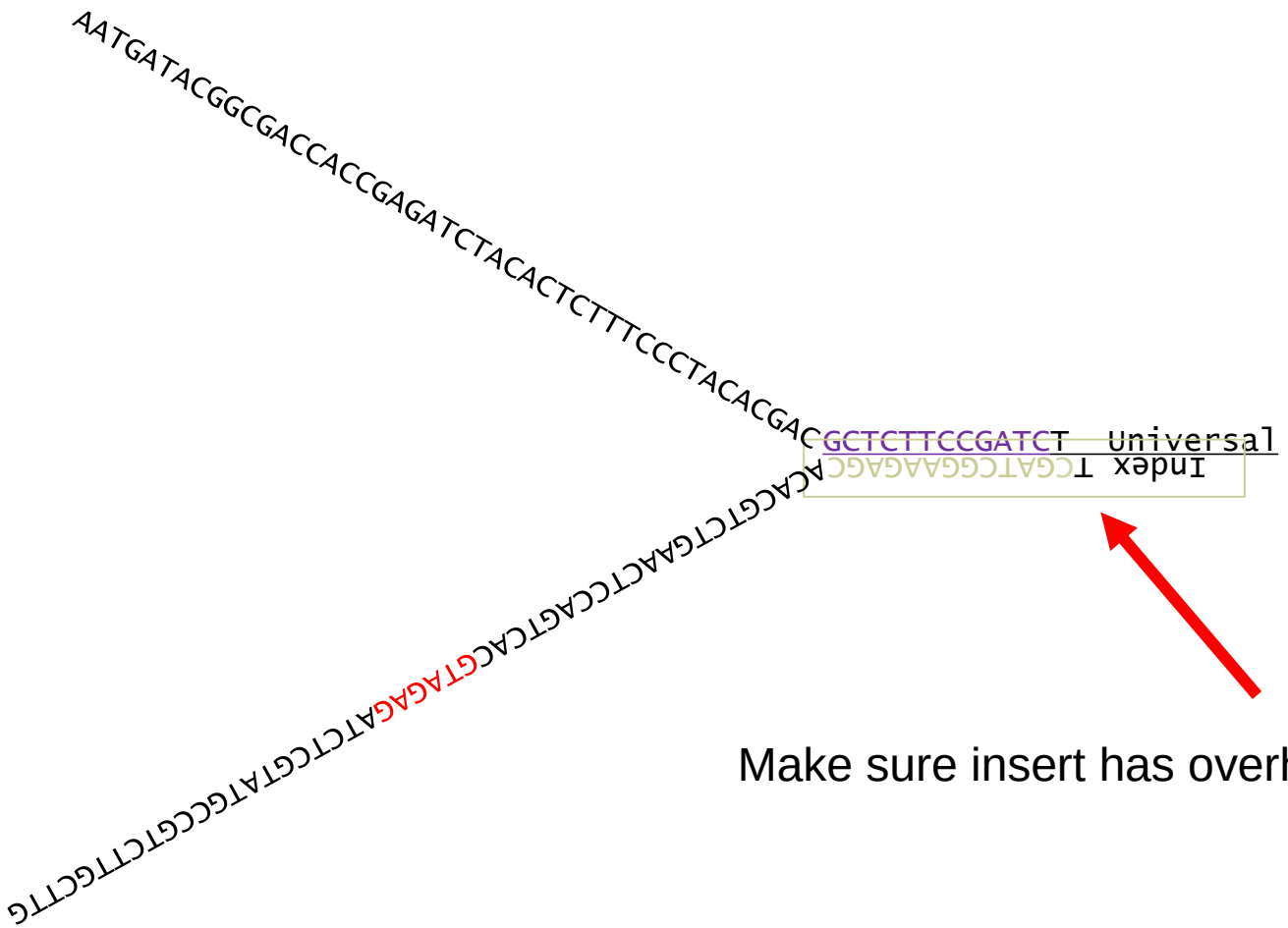
Ekblom, fig 2 (partial)

TruSeq dual index Adapters (since 2017)



Genome Assembly

Truseq adapters

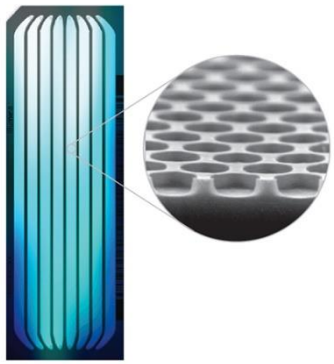


Make sure insert has overhanging A

Genome Assembly

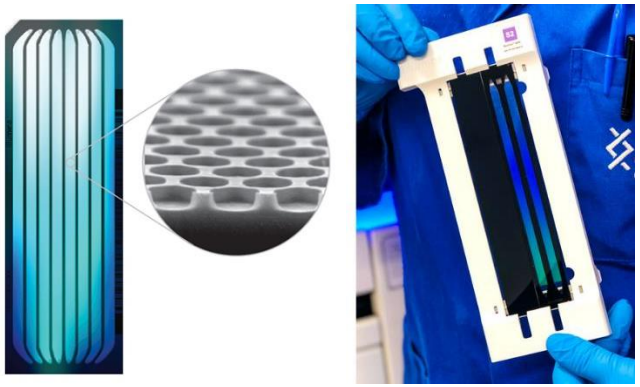
Illumina Sequencing technology

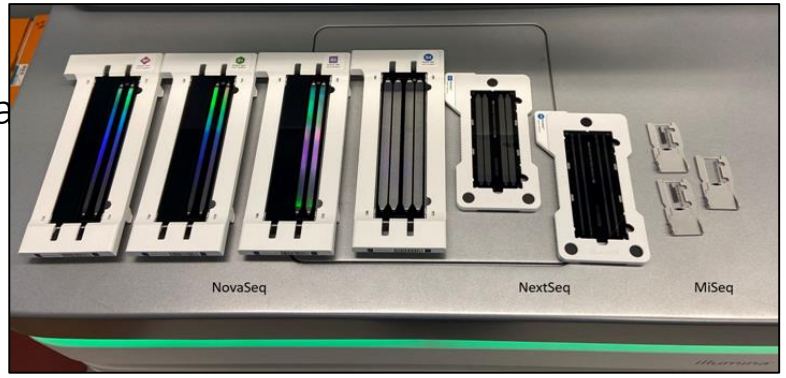
- *Most commonly used technology, > 85%*
- *Typically 100-150 base reads, but can go up to 250-300 bases*
- *Truseq adapter sequence reduces "empty" sequences and artifacts*
- *Random fragments are run into patterned flowcell and amplified into local clusters*
  - Slide has covalently attached DNA complementary to the adapter
  - Use bridge amplification to create local clusters
    - Similar to emulsion PCR described in Momand



Illumina Sequencing technology

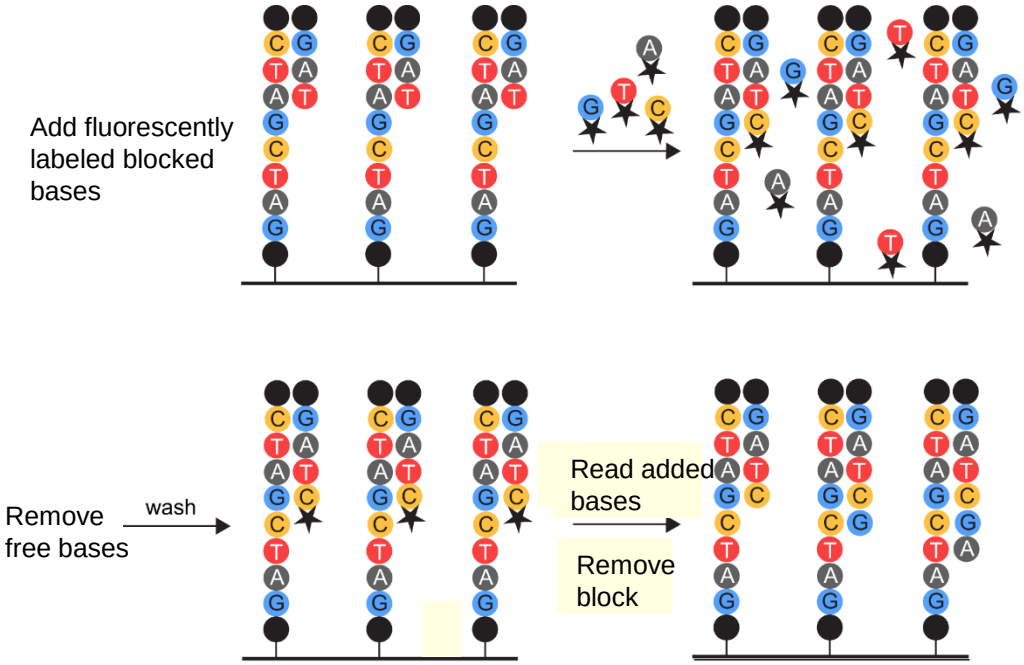
- *Most commonly used technology, > 90%*
- *Typically 100-150 base reads, but can go up to 250*
- *Truseq adapter sequence reduces "empty" sequenc*
- *Sequence is determine by synthesizing cDNA strand*
  - but a single DNA strand can't be detected
- *Random fragments are run into patterned flowcell and amplified into local clusters*
  - Slide has covalently attached DNA complementa
  - Use bridge amplification to create local clusters

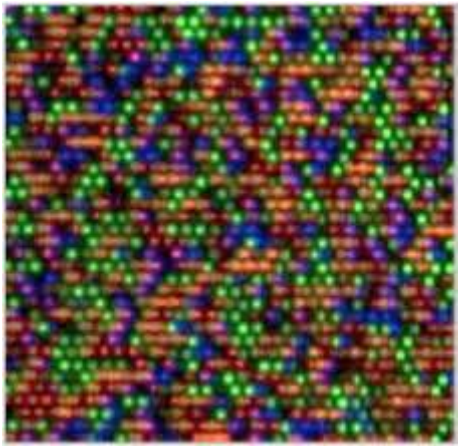




Illumina sequencing

1. Bind (hybridize to reverse complement) polymerase primer
2. Add one base (fluorescent), base is chemically blocked and cannot be extended
3. detect the single added base with accurate CCD camera
4. unblock, wash out unused reagents, return to 2

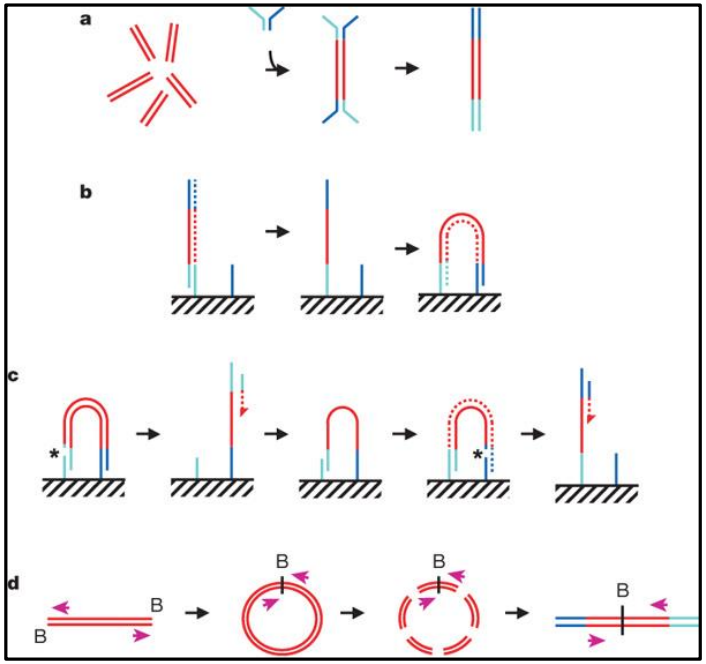


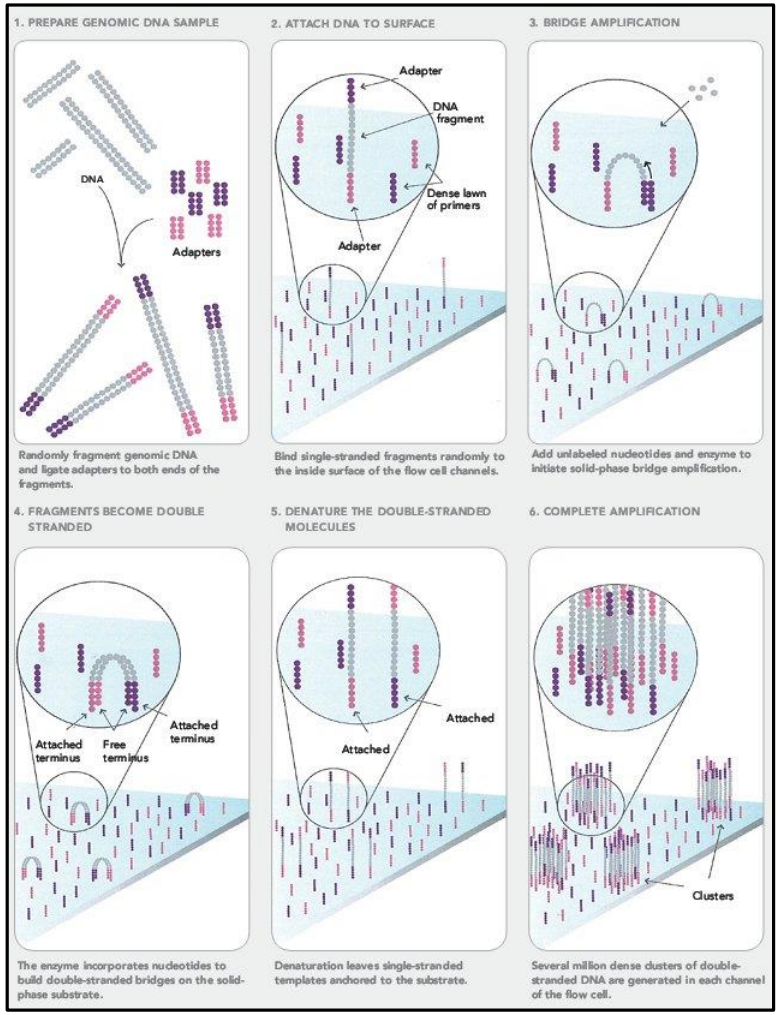


Genome Assembly

Illumina Sequencing

- *Bridge amplification creates clusters*
- *Sequence from each end using different primers*



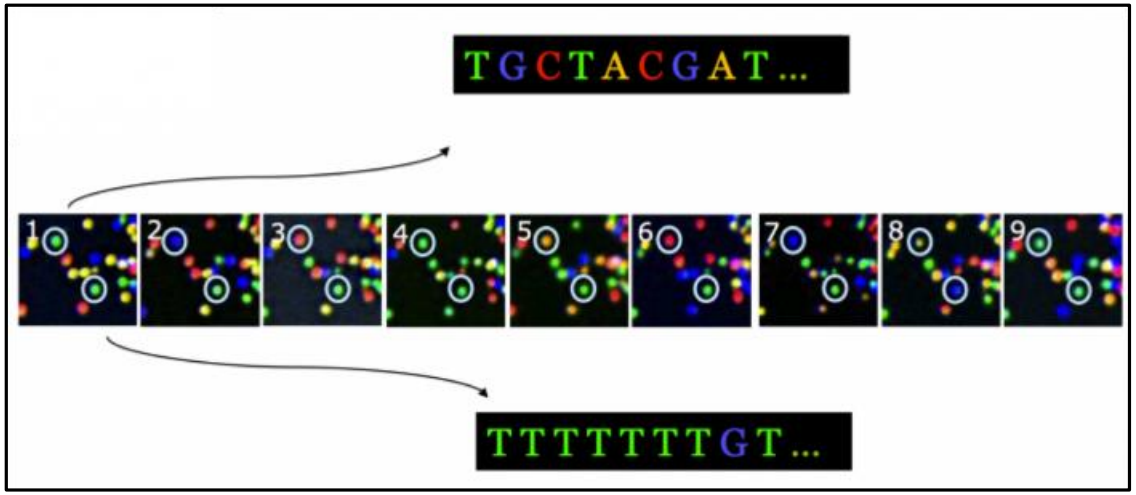


Genome Assembly

Illumina process

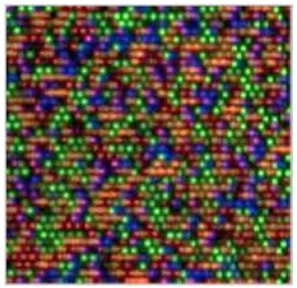
Sequencing by synthesis

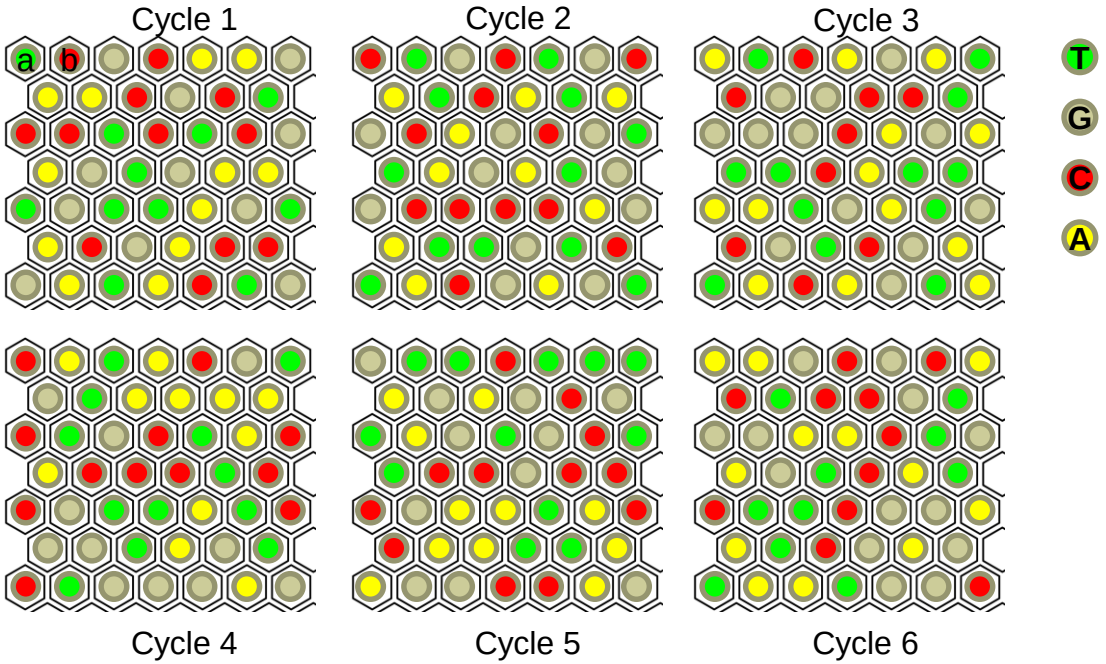
1. *Bind polymerase primer*
2. *Add one base (fluorescent), base is chemically blocked and cannot be extended*
3. *detect the single added base*
4. *unblock, return to 2*



Base Calling

- *Each cycle one base is added and read*
  - a = TCACGA
  - b = CTTATA

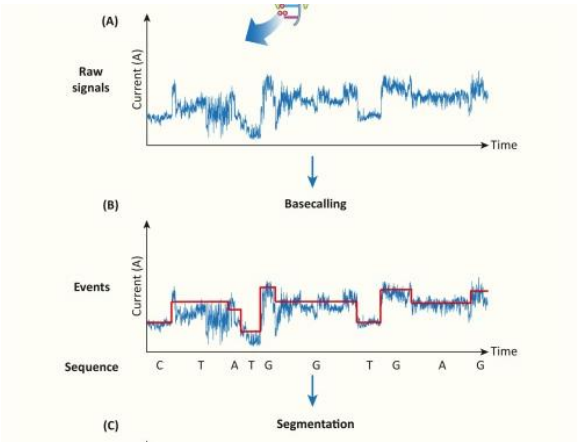


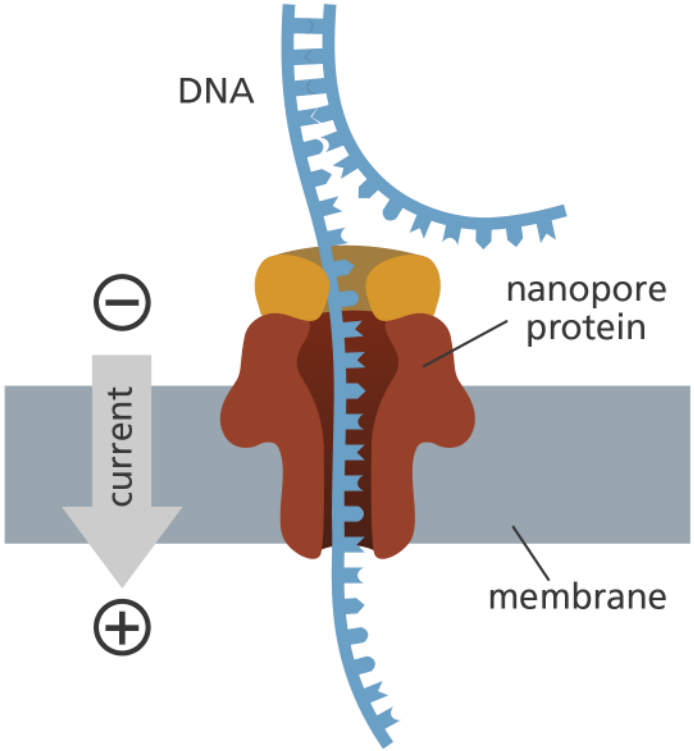


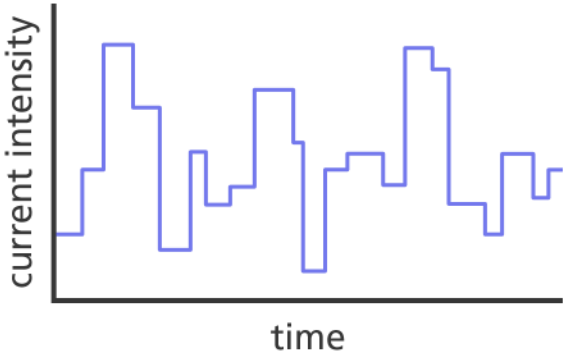
Genome Assembly

Oxford Nano Technologies (Nanopore)

- *single molecule sequencing*
- *long reads 10-100 k, ONT claims 99% accuracy*





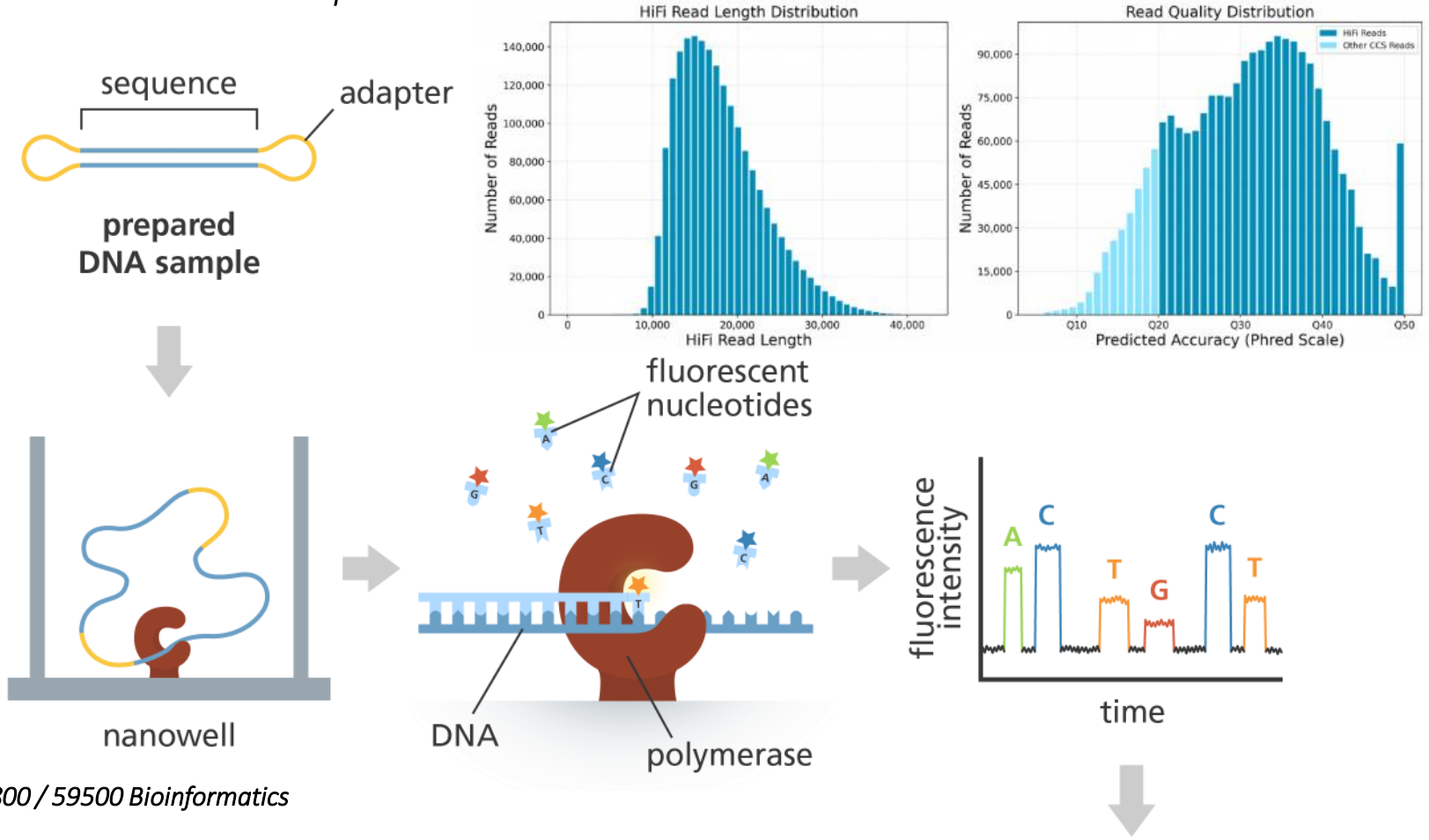


A C T G C T ...

Genome Assembly

Pacific Biosystems (Pac Bio)

- *Single molecule sequencing*
- *long reads 10-40 k*
- *circular consensus sequence (CCS), now called HiFi, 99.9% accuracy*



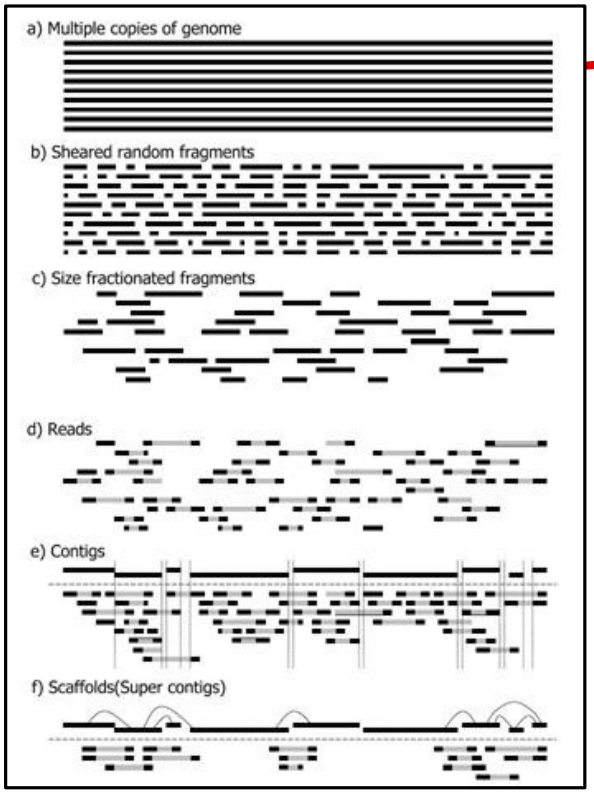
Genome Assembly

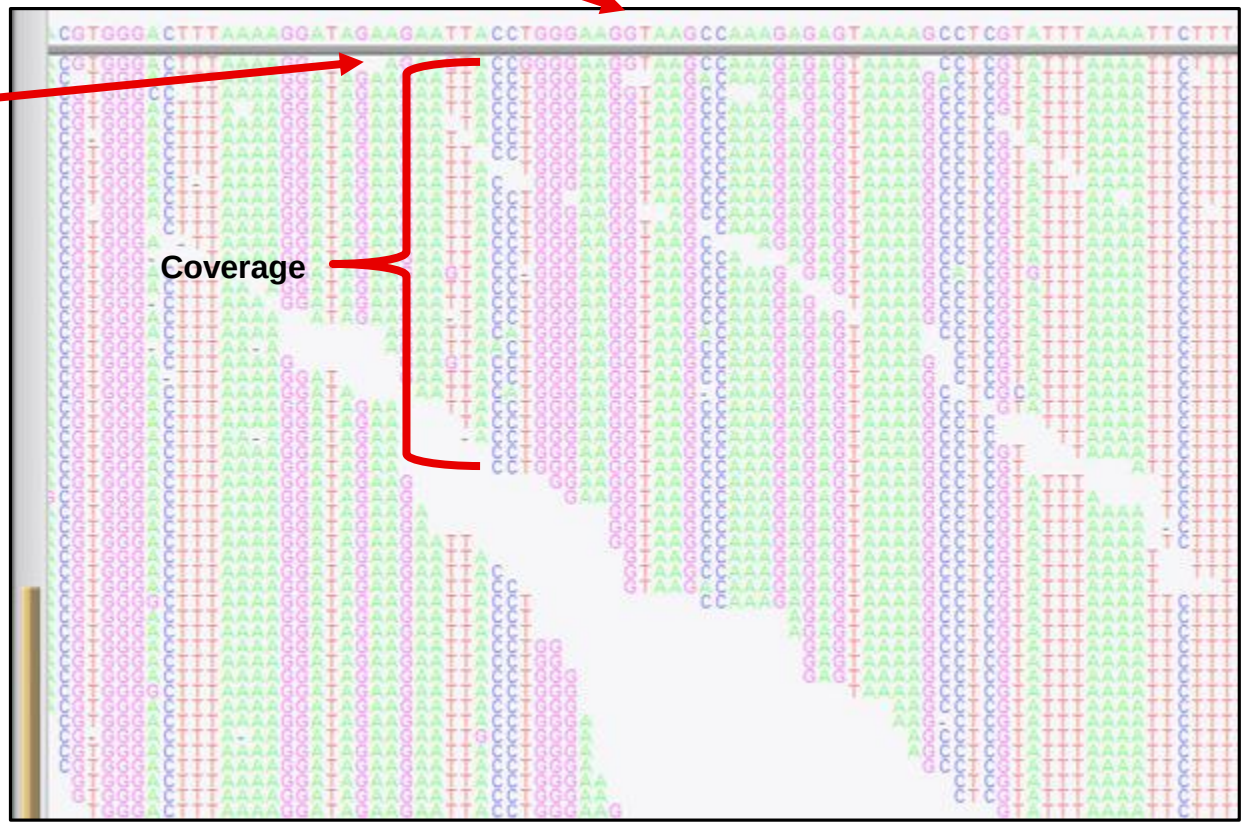
- *Original plan for human genome*
  - Isolate chromosomes
  - clone in Bacterial Artificial Chromosome (BAC) vectors (~200kb inserts)
  - Find "golden path" through BACs to minimize sequencing (tiling)
  - Subclone BACs into plasmids
  - Sequence using dideoxy chain termination (Sanger) sequencing
  - Optimistically estimated to take $3 billion and take 15 years
  - Initiated in 1990 – claimed to be the largest collaborative science project ever

Genome Assembly

Short reads -> whole genome

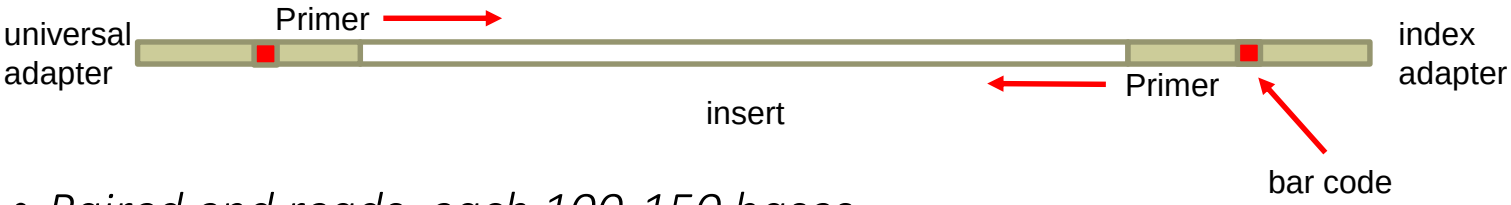
- *Reads typically 100-150 bases long*
- *Error rate ~ 0.01 – 0.1%*
- *10^8 – 10^9 reads*



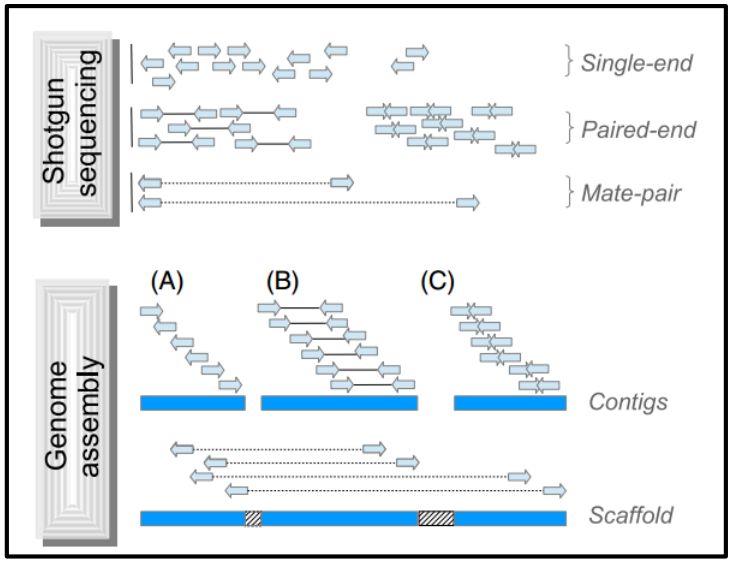


Genome Assembly

Illumina TruSeq System



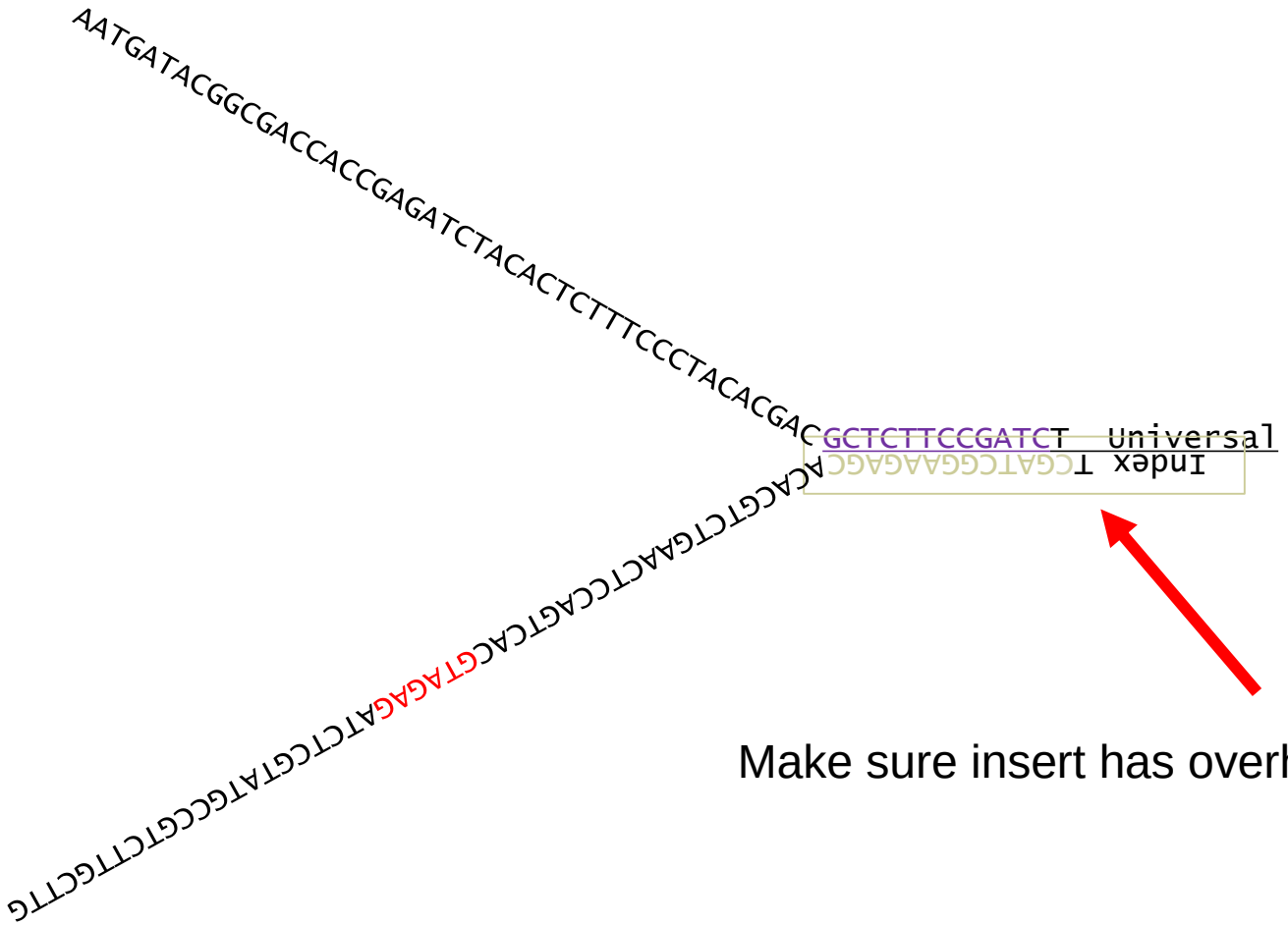
- *Paired end reads, each 100-150 bases*
- *Vocabulary*
  - paired-end
  - mate-pair
  - contig
  - scaffold



Ekblom, fig 2 (partial)

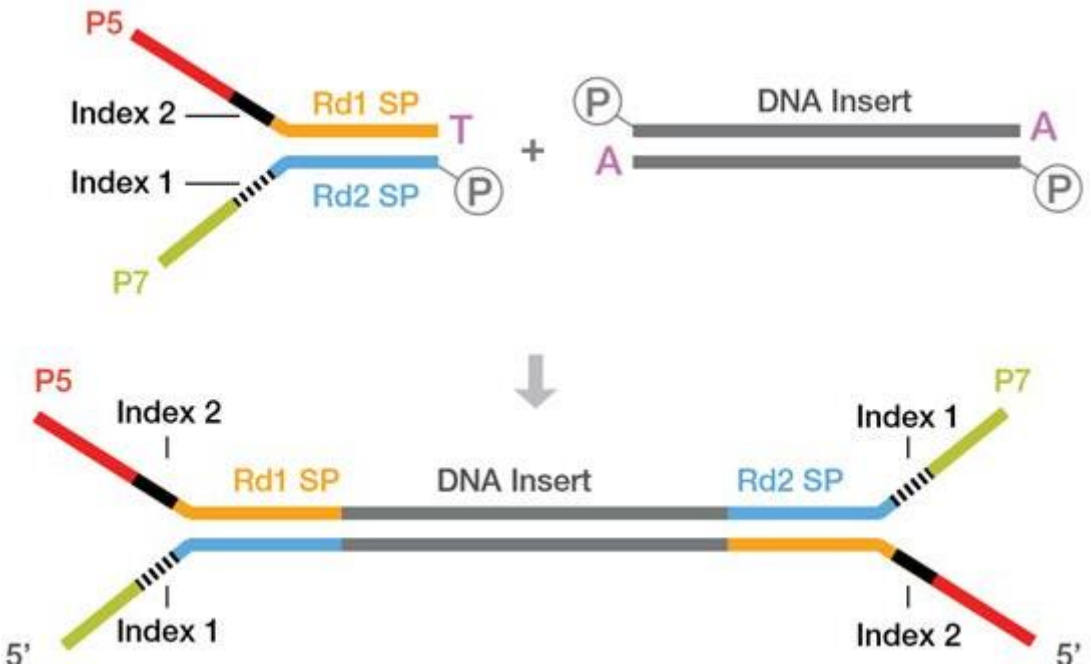
Genome Assembly

Truseq adapters



Make sure insert has overhanging A

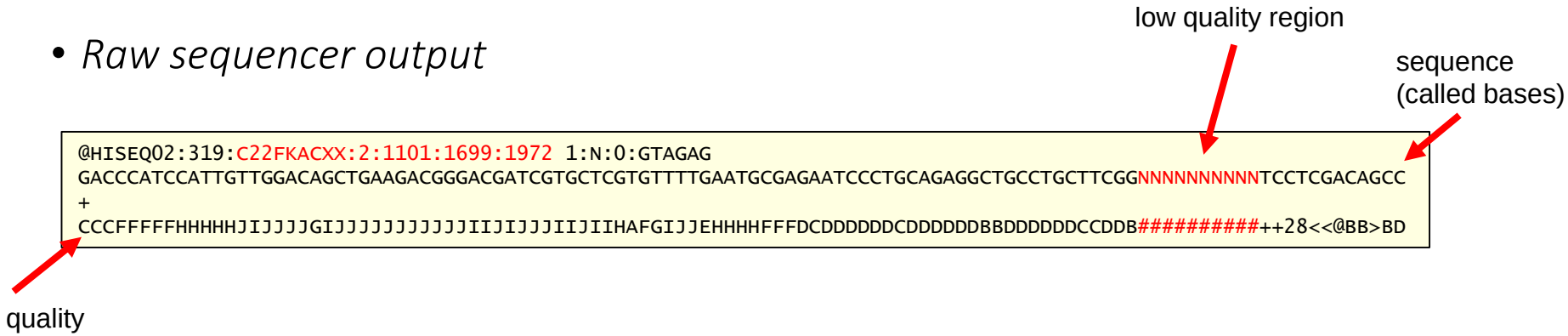
TruSeq dual index Adapters (since 2017)



Sequencing Basics

Fastq format

- *Raw sequencer output*



- *line1: @, followed by ID information, Illumina shown in example*
  - instrument:run:flowcell:lane:tile:x:y pair:filtered:control:bar-code
- *line2: Sequence*
- *line 3: +*
  - this line may contain any information, sometimes ID is repeated
- *line 4: Quality, ASCII encoded*
  - note that @ is a valid quality character

Sequencing Basics

What is quality?

- *Probability that the called base is incorrect*
  - Introduced in the Phred program (Phil Green, UWa) ca. 1998
  - Encoded as a single ASCII letter (so it is collinear with the sequence)
- *Quality score, $Q = -10 \log_{10} \varepsilon$*
  - ε is the expected error rate (probability of calling an incorrect base)
- *20 (P=0.01) is a commonly used cutoff (why?)*

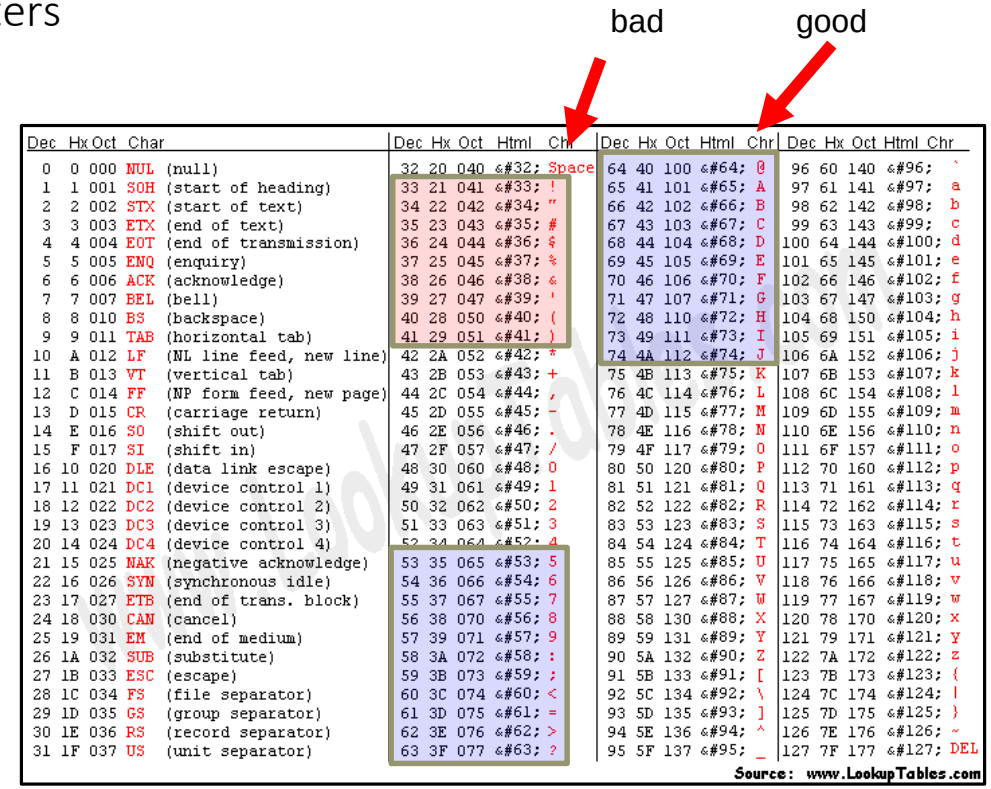
Phred Quality Score	Letter Code	Error Probability	Base Call Accuracy
10	+	0.1	90%
13	.	0.05	95%
20	5	0.01	99%
30	?	1e-3	99.9%
40	I	1e-4	99.99%
50	S	1e-5	99.999%

Sequencing Basics

What is quality?

- *ASCII encoding*
- *Every letter in a computer is represented as a digital value*
  - ASCII code (American Standard Code for Information Interchange)
  - In FastQ files, quality+33 is used to encode the quality
  - The first 32 are non-printing characters

Phred Quality Score	Letter Code	Error Probability	Base Call Accuracy
10	+	0.1	90%
13	.	0.05	95%
20	5	0.01	99%
30	?	1e-3	99.9%
40	I	1e-4	99.99%
50	S	1e-5	99.999%



Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Sequencing Basics

Fastq format

sequence

```
@HISEQ02:319:C22FKACXX:2:1101:1699:1972 1:N:0:GTAGAG
GACCCATCCATTGTTGGACAGCTGAAGACGGGACGATCGTGCTCGTGTTTTGAATGCGAGAATCCCTGCAGAGGCTGCCTGCTTCGGNNNNNNNNNNTCCTCGACAGCC
+
CCCFFFFFHHHHHJIJJJJGIJJJJJJJJJJJIIJIJJJIIJIIHAFGIJJEHHHHFFFDCDDDDDDCDDDDDDBBDDDDDDCCDDB##########++28<<@BB>BD
```

quality

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

$\varepsilon = 10^{-Q/10}$

I = ascii 73

Quality = 73 – 33 = 40

Quality = $-10 \log_{10} \varepsilon$

$\varepsilon = 10^{-4}$

\# = ascii 35

Q = 35 – 33 = 2

$\varepsilon = 10^{-0.2} = 0.63$

= totally bogus

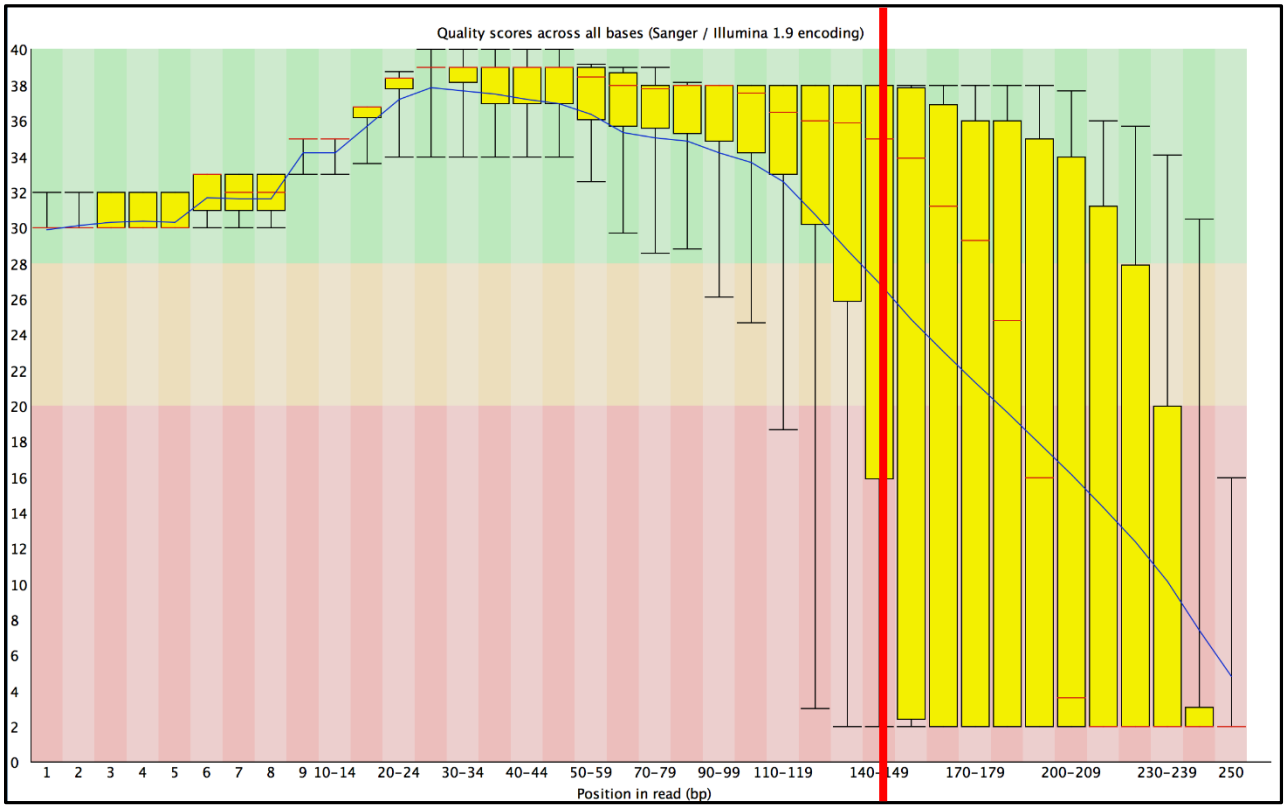
Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Genome Assembly

Illumina Quality

- *Quality filtering is important to assembly quality*
- *20 is a common cutoff for base quality (1% error)*



Genome Assembly

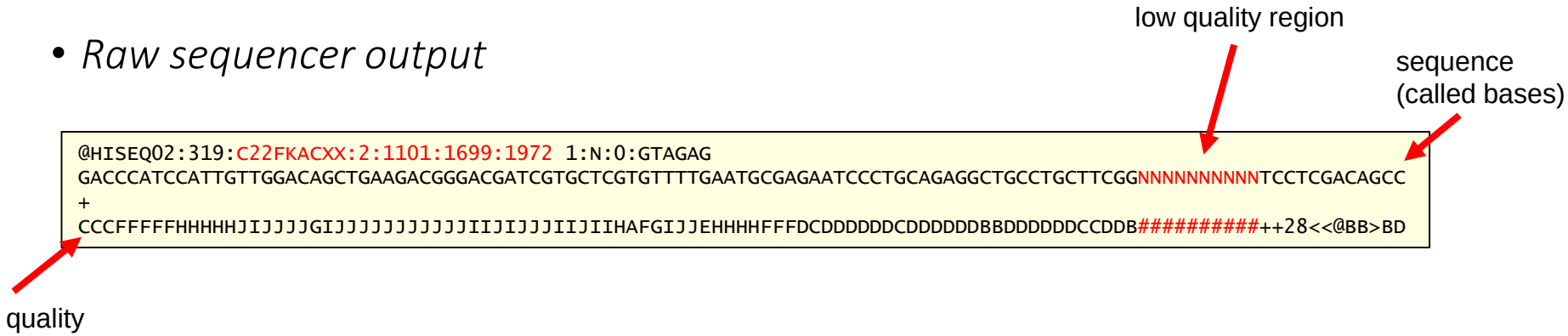
Week 9

- *Genome Sequencing*
  - Shotgun Sequencing
    Sequence quality
- *Genome Assembly*
  - Overlap-layout-consensus
  - Debruijn Graph

Sequencing Basics

Fastq format

- *Raw sequencer output*



- *line1: @, followed by ID information, Illumina shown in example*
  - instrument:run:flowcell:lane:tile:x:y pair:filtered:control:bar-code
- *line2: Sequence*
- *line 3: +*
  - this line may contain any information, sometimes ID is repeated
- *line 4: Quality, ASCII encoded*
  - note that @ is a valid quality character

Sequencing Basics

What is quality?

- *Probability that the called base is incorrect*
  - Introduced in the Phred program (Phil Green, UWa) ca. 1998
  - Encoded as a single ASCII letter (so it is collinear with the sequence)
- *Quality score, $Q = -10 \log_{10} \varepsilon$*
  - ε is the expected error rate (probability of calling an incorrect base)
- *20 (P=0.01) is a commonly used cutoff (why?)*

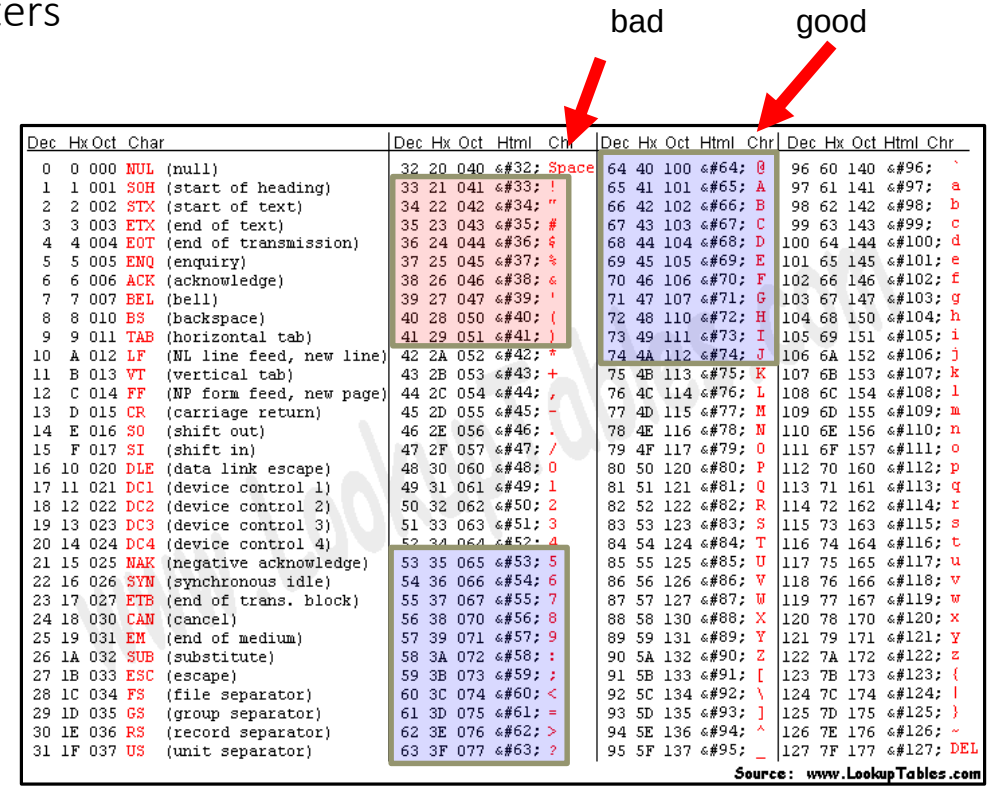
Phred Quality Score	Letter Code	Error Probability	Base Call Accuracy
10	+	0.1	90%
13	.	0.05	95%
20	5	0.01	99%
30	?	1e-3	99.9%
40	I	1e-4	99.99%
50	S	1e-5	99.999%

Sequencing Basics

What is quality?

- *ASCII encoding*
- *Every letter in a computer is represented as a digital value*
  - ASCII code (American Standard Code for Information Interchange)
  - In FastQ files, quality+33 is used to encode the quality
  - The first 32 are non-printing characters

Phred Quality Score	Letter Code	Error Probability	Base Call Accuracy
10	+	0.1	90%
13	.	0.05	95%
20	5	0.01	99%
30	?	1e-3	99.9%
40	I	1e-4	99.99%
50	S	1e-5	99.999%



Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Sequencing Basics

Fastq format

sequence

```
@HISEQ02:319:C22FKACXX:2:1101:1699:1972 1:N:0:GTAGAG
GACCCATCCATTGTTGGACAGCTGAAGACGGGACGATCGTGCTCGTGTTTTGAATGCGAGAATCCCTGCAGAGGCTGCCTGCTTCGGNNNNNNNNNNTCCTCGACAGCC
+
CCCFFFFFHHHHHJIJJJJGIJJJJJJJJJJJIIJIJJJIIJIIHAFGIJJEHHHHFFFDCDDDDDDCDDDDDDBBDDDDDDCCDDB##########++28<<@BB>BD
```

quality

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

$\varepsilon = 10^{-Q/10}$

I = ascii 73

Quality = 73 – 33 = 40

Quality = $-10 \log_{10} \varepsilon$

$\varepsilon = 10^{-4}$

\# = ascii 35

Q = 35 – 33 = 2

$\varepsilon = 10^{-0.2} = 0.63$

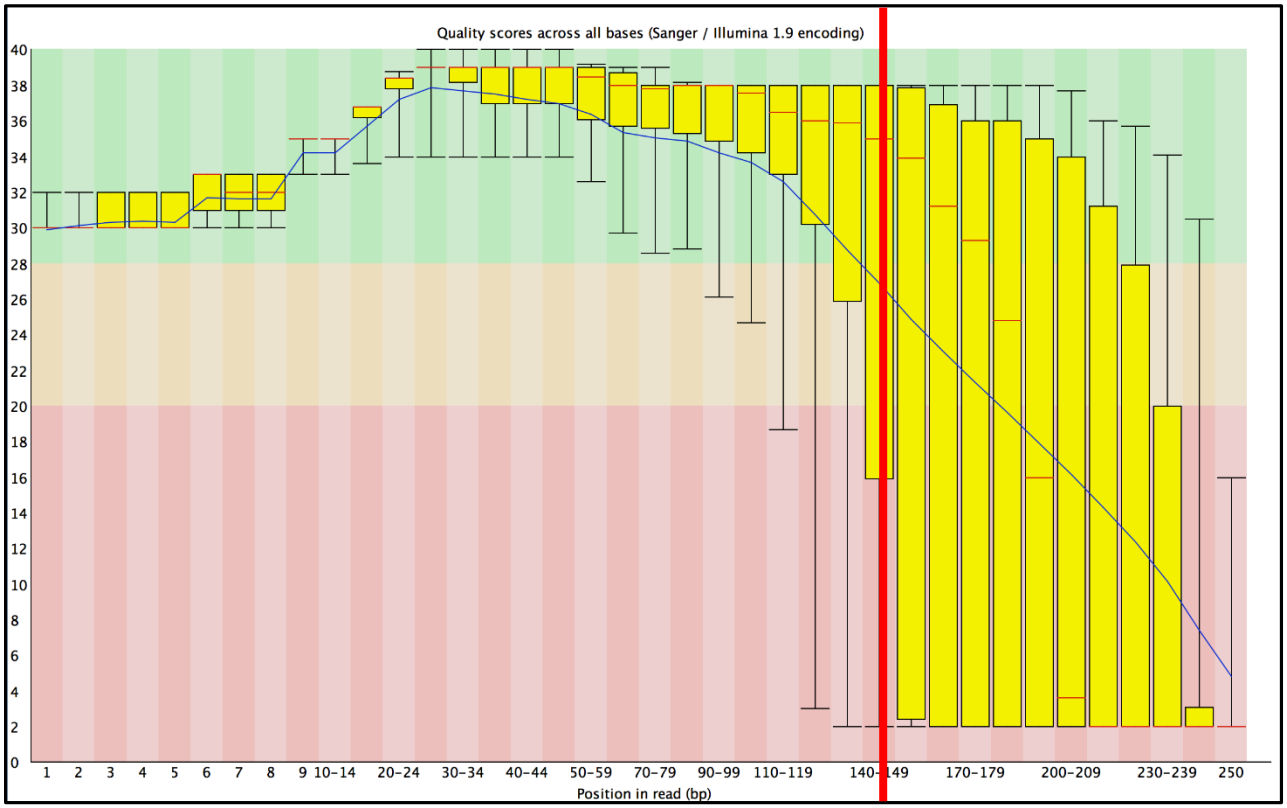
= totally bogus

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Illumina Quality

- Quality filtering is important to assembly quality
- 20 is a common cutoff for base quality (1% error)

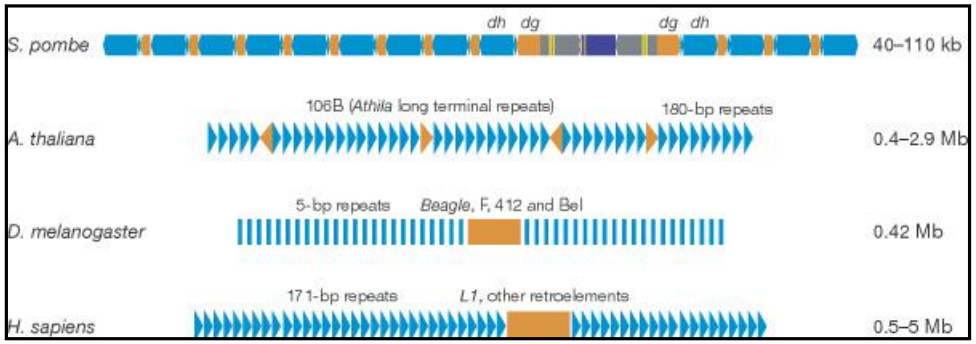


Whole Genome Shotgun (WGS) Assembly

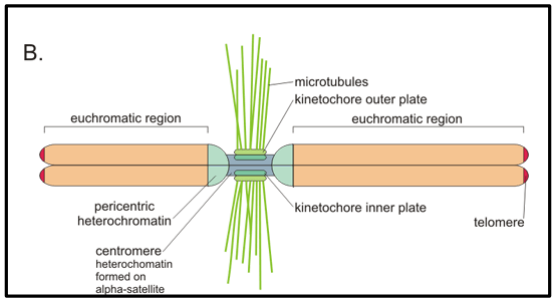
- *Problems*
  - Genome is large, bacteria typically have 5 x 10^6 bases
    - many eukaryotes have 10^9 bases
  - Reads are small, 100-150 bases
  - Genomes contain repetitive regions
    - Centromeres, telomeres, satellite (heterochromatin)
    - transposons
    - homopolymer repeats / microsatellites
  - Many genomes are diploid, chromosomes may vary in sequence and/or structure
  - Cells contain organelles with their own DNA (mitochondria, chloroplast)
  - Genomes contain duplicated segments (especially ribosomal RNA operons)
  - Cells contain parasites and pathogens (even bacteria)
  - Laboratory contamination
- *How can you reconstruct the complete genome sequence?*
- *<u>Can</u> you reconstruct the complete genome sequence?*

Genome Assembly

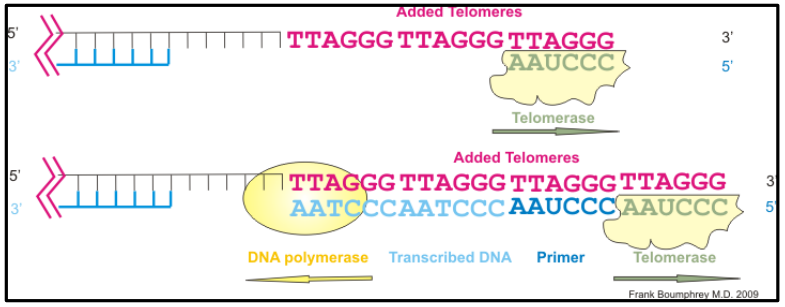
Repeats in genomes



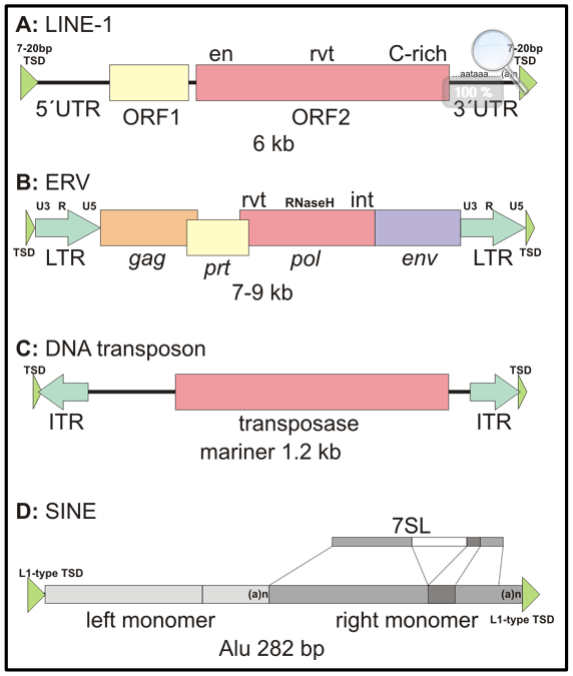
short repeats



Centromeres



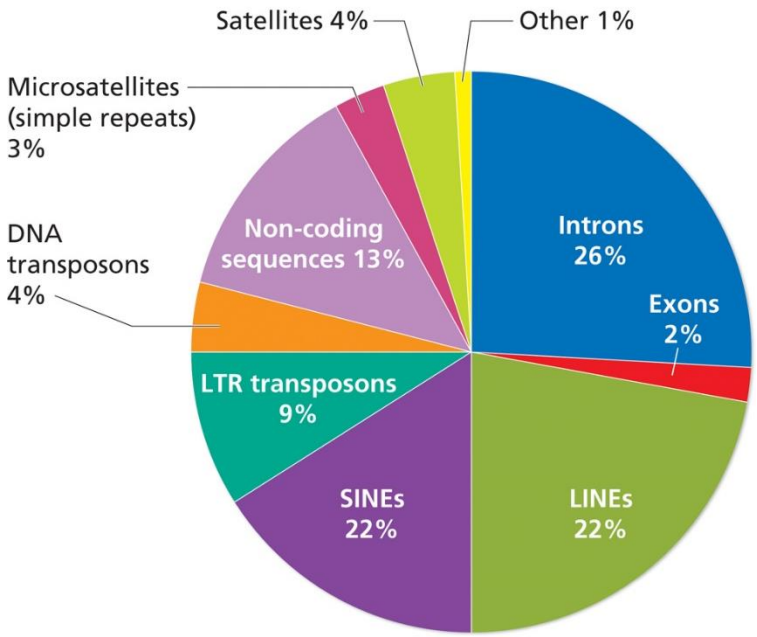
Telomeres

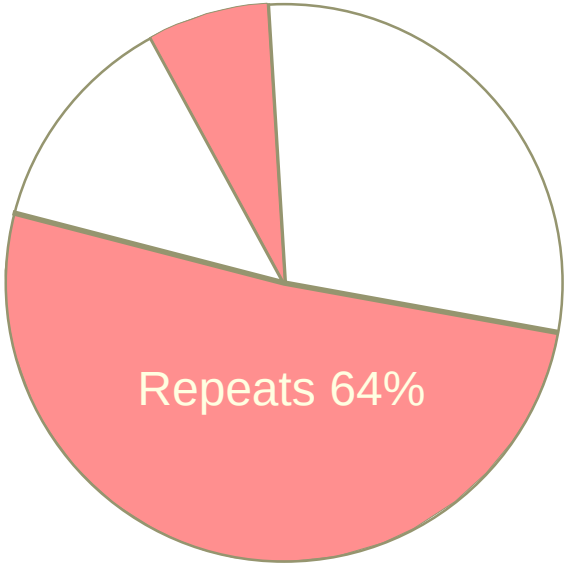


LINES
RT
DT
SINEs

Genome Assembly

Human Genome





Genome Assembly

Shotgun Assembly

- *Overlap-Layout-Consensus*
  - Mostly used for long reads, i.e., before 2008 or Pac-Bio/Nanopore
  - Align all sequence reads to find how they overlap
    - Human genome: $3x10^9$ x 20X coverage x 100 base reads = 600 M reads
    - Many pairwise alignments
      $1.8x10^{17}$ comparisons
      memory (assume 2 byte int) 2.9 billion Gb
  - Figure out the best layout (hard)
  - Hamiltonian (Eulerian) path
  - Generate consensus (easy)
- *De Bruijn Graph method*
  - Most commonly used
  - Break reads up into kmers
    - How many kmers in a genome: Genome size – k + 1
  - Overlap kmers (Burrows Wheeler transform)
  - Construct De Bruijn graph(s)
  - Find path through graph(s) ⇒ contig
  - Use paired-end and mate-pair sequences to create scaffolds
  - Fill remaining gaps (gap closing)

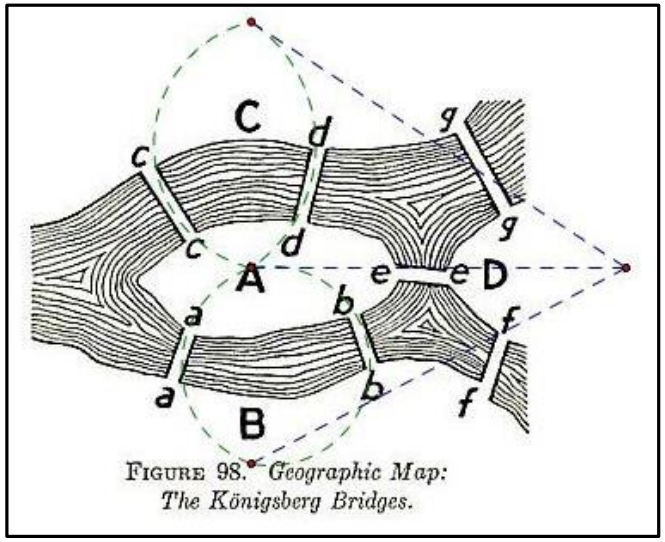


FIGURE 98. *Geographic Map: The Königsberg Bridges.*

Genome Assembly

Overlap – layout – consensus Assembly

- *Used for long-read sequence (ONT/PacBio)*
- *First step is often error correction because long read sequences typically have higher error rate (especially indels)*
  - Correct by overlapping long reads
  - Correct using short reads
- *For every pair of sequence fragments, find the ones that overlap*
  - finding overlaps is an alignment problem (usually gaps are not considered)
  - arrange them in sequential order according to overlaps
  - when a position is covered by several fragments, keep the consensus letter
  - result -> contiguous sequences = contigs

Genome Assembly

Overlap – layout – consensus assembly

- *DNA is just a text*
- *Can we reconstruct a text from random fragments?*
- *The assembly problem is to reconstruct the genome text from many small fragments (sequence reads)*

Hamlet:
To be, or not to be--that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune
Or to take arms against a sea of troubles
And by opposing end them.

without punctuation, 153 characters

tobeornottobethatisthequestionwhethertisnobl
erinthemindtosuffertheslingsandarrowsofoutra
geousfortuneortotakearmsagainstaseaoftroub
lesandbyopposingendthem

Sequence Assembly

Reconstructing the genome sequence from short fragments

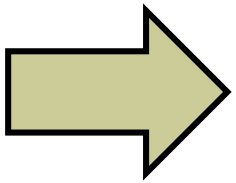
- *A metaphor, assume*
  - I am on a desert island and I want to send you the Hamlet text
  - But my radio (telegraph) is defective and all you receive is 8 random letters from somewhere (anywhere) in the text
  - Solution – resend the message multiple times
  - Problem – you have to reassemble the pieces and neither of us knows where in the message each piece came from

  - The message we want to send is the genome
  - The short fragments are sequence reads

Overlap – layout – consensus assembly

- *I randomly sampled 11 thirty letter words*
- *Let's see if we can reassemble Hamlet's soliloquy*
- *11 x 30= 330*
- *coverage = 330/153 = 2.15*

```
tobeornottobethatisthequestionwhetherti
snoblerinthemindtosuffertheslingsandarr
owsofoutrageousfortuneortotakearmsagain
staseaoftroublesandbyopposingendthem
```



```
stionwhethertisnoblerinthemind
ethatisthequestionwhethertisno
gainstaseaoftroublesandbyoppos
tosuffertheslingsandarrowsofou
sandarrowsofoutrageousfortuneo
ertisnoblerinthemindtosufferth
themindtosuffertheslingsandarr
staseaoftroublesandbyopposinge
noblerinthemindtosuffertheslin
inthemindtosuffertheslingsanda
rtisnoblerinthemindtosufferthe
```

Genome Assembly

Overlap – layout – consensus assembly

- *How well did i cover the message?*
- *Since I never even received the red parts I can't reconstruct (assemble) them*
- *For convenience, i'll call the overlapping parts contigs*
  - There are two contigs, 89 and 40 characters each
  - message reconstructed is 89+40/153 = 84 % of original message
  - If I don't know the message, I can't order the contigs

```
tobeornottobethatisthequestionwhethertisnoblerinthemindtosuffertheslingsandarrowsofoutrageousfortuneo
           ethatisthequestionwhethertisno
                       stionwhethertisnoblerinthemind
                                 ertisnoblerinthemindtosufferth
                                  rtisnoblerinthemindtosufferthe
                                      noblerinthemindtosuffertheslin
                                            inthemindtosuffertheslingsanda
                                              themindtosuffertheslingsandarr
                                                     tosuffertheslingsandarrowsofou
                                                                sandarrowsofoutrageousfortuneo

rtotakearmsagainstaseaoftroublesandbyopposingendthem
           gainstaseaoftroublesandbyoppos
               staseaoftroublesandbyopposinge
```

Overlap – layout – consensus Assembly

- *How do you measure assembly quality?*
  - Number of contigs (fewer is better until you reach the number of chromosomes)
  - N50 (larger is better until?)
- *Both favor over-assembled/incorrectly assembled sequences*

sorted by size (total 106 characters)

	length	cumulative	fraction
troublesandbyopposingend	24	106	1.000
dtosufferthesling	17	82	0.774
bethatistheque	14	65	0.613
stionwhethe	11	51	0.481
rowsofou	8	40	0.377
rinthemi	8	32	0.302
totakear	8	24	0.226
staseaof	8	16	0.151
rageousf	8	8	0.075

N50 = 11.4
half the letters are
in shorter contigs

Genome Assembly

Overlap – layout – consensus Assembly

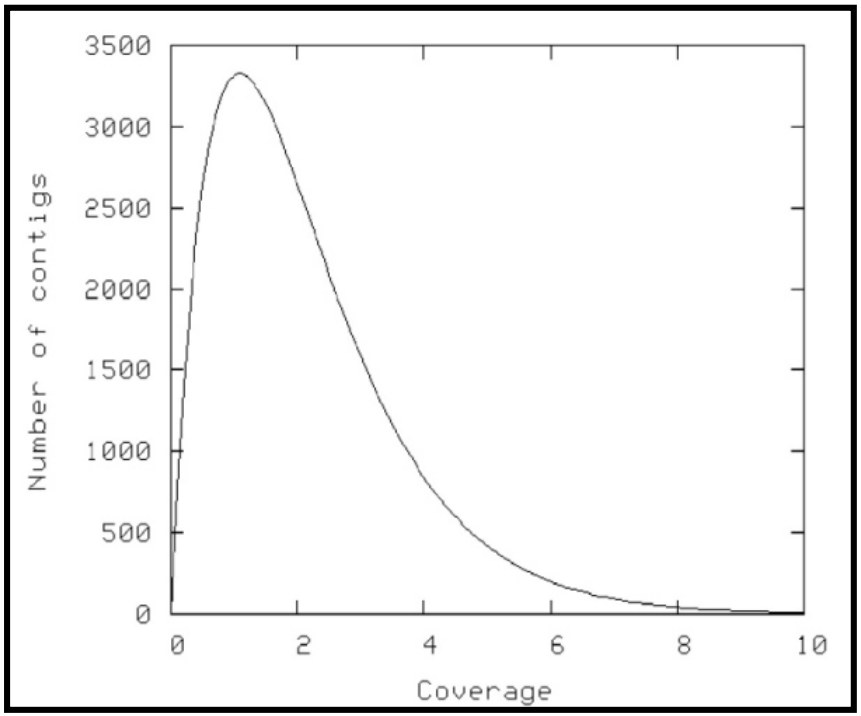
- *Problems*
- *In this simulation, reads are a substantial fraction of the length of the message (genome). Never true in RL even with long reads*
- *Sample needs to be much deeper if we want to get a complete message*
  - From the coverage, i figured i'd have a problem
    - Coverage = total number of letters received divided by the length of the message
    - Coverage = N * L / G
      N = number of fragments
      L = length of fragments
      G = size of genome
  - sequences = 11 x 30 = 330 characters
    genome = 153 characters
    coverage = N * L / G = 330 / 153 = 2.15
    N = number of fragments
    L = length of fragments
    G = size of genome
- *Long read sequencing typically has high error rate and many indels*

Genome Assembly

How much sequence do you need (ca. 2004)

- Lander ES, Waterman MS, Genomic mapping by fingerprinting random clones: a mathematical analysis, Genomics 2: 231-239 (1988)

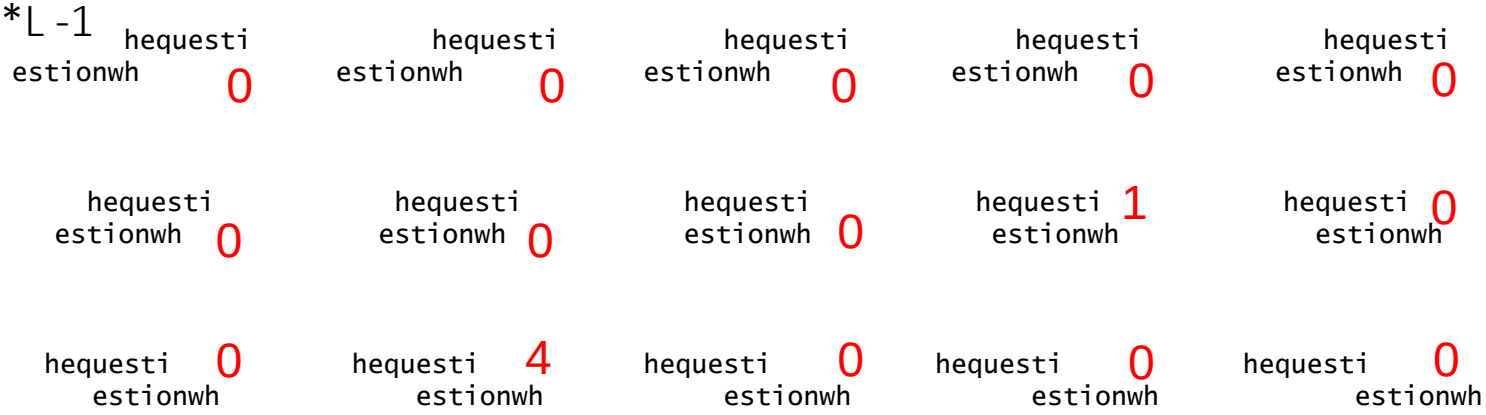
- *Depends on*
  - genome size (G)
  - sequence length (L)
  - number of sequences (N)
- *coverage = L N / G*
- *In 1st generation sequencing 16 X coverage was a good target*
- *Today >100 X is common*
- *For the human genome*
  - 15 X coverage
  - 500 base reads
  - 3.3 x 10e9 bp
  - 99 million reads = $ 3 billion @ $30/base,
    = $9.9 million @ $0.10/base



Overlap – layout – consensus Assembly

- *OLC is expensive because of the overlap step*
- *for n fragments you have to do $n^2/2$ comparisons*
- *A simple way of finding overlaps is to slide the fragments past each other and count the number of matches (this isn't a good method, just a thought example)*
- *for each pair*
  - assume that we slide each fragment past the other and count the number of matches.

  2*L -1



Genome Assembly

OLC / Hybrid Assembly

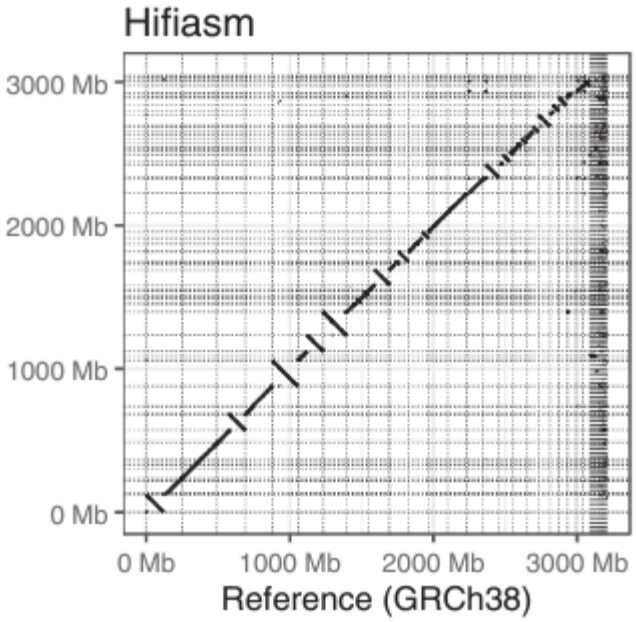
- *Canu, Flye, Miniasm, Masurca, Goldrush*
  - Hifiasm, HiCanu for Hifi reads
- *Speed up overlap using kmer based comparisons*
- *Overlap graph is messy due to errors and repeats*

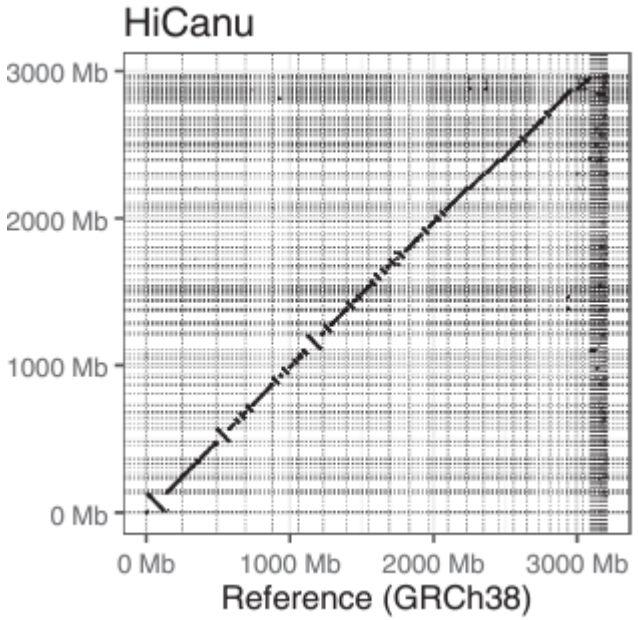
Quality evaluation of different genomes on HiFi datasets measured in terms of contiguity, correctness and completeness for the assembly with *Hifiasm* and *HiCanu*.

Quality evaluation	Metric	*E. coli* k12		*S. pombe*		*D. melanogaster*		*S. tuberosum*		*H. sapiens* (HG002)	
		HiCanu	Hifiasm	HiCanu	Hifiasm	HiCanu	Hifiasm	HiCanu	Hifiasm	HiCanu	Hifiasm
Contiguity	N50 (Mbp)	4.54	4.66	4.61	9.66	20.87	22.94	3.86	24.27	36.65	87.11
	NG50 (Mbp)	4.54	4.66	4.61	9.66	20.87	23.95	5.14	28.88	38.18	78.86
	Number of contigs	217	4	72	75	59	281	562	1610	1169	934
	GC (%)	50.61	50.74	36.29	36.56	41.71	41.29	34.48	35.60	40.54	40.75
	Largest contig (Mbp)	4.54	4.66	5.62	9.66	24.41	28.21	19.10	60.22	121.36	176.22
	Genome (Mbp)	9.01	4.73	15.93	16.63	179.69	177.77	942.55	953.14	3326.85	31,126.15
Correctness	Missassemblies	10	6	139	196	4027	6255	78,237	67,167	18,844	18,715
	Missmatches	314	8	3613	4289	586,895	616,356	9,593,452	8,807,798	5,359,406	5,542,121
	Fully unaligned contig	0	0	0	0	7	55	9	62	3	12
	Partially unaligned contigs	0	0	3	1	31	92	522	828	330	187
	Genome fraction (%)	99.998	99.998	99.855	95.937	88.578	94.576	70.91	77.078	93.986	94.45
	Total aligned (Mbp)	9.01	4.73	15.81	16.41	175.02	164.29	697.13	641.38	3260.63	3002.72
Completeness	Missing genes	0	0	18.4	18.3	0.3	0.5	2.6	1.4	3.6	3.3
	Fragmented genes	0	0	1.7	1.7	0.1	0.1	0.1	0.1	1.6	1.1
	Identified genes	100	100	79.9	80	99.6	99.4	97.3	98.5	94.8	95.6
	Total	124	124	1706	1706	1367	1367	5950	5950	13,780	13,780

Genome Assembly

OLC/Hybrid Assembly





Genome Assembly

Overlap – layout – consensus Assembly

- $O \approx (n^2/2)(2L - 1) \approx Ln^2$
  since L is small compared to n, we usually say the complexity of OLC is n^2 or more simply $O(n^2)$
- Also known as quadratic
  every time you double the size of the problem, the required time goes up by 4
- 1000 reads $1000^2 = 1M$ – not scary
- 100 Mreads $(10^8)^2 = 10^{16}$ – scary
  at 10^6 calculations/sec = 31 years

- it takes about 60 Kreads for 1X coverage of human genome

Kmers

- *What's a kmer?*
- *Each sequence gets tabulated as the total number of overlapping words of length k*
- *kmer words, for k =5*

```
tobethat
tobet
 obeth
  betha
   ethat
```

original read (8 letters)

L – k + 1 kmers

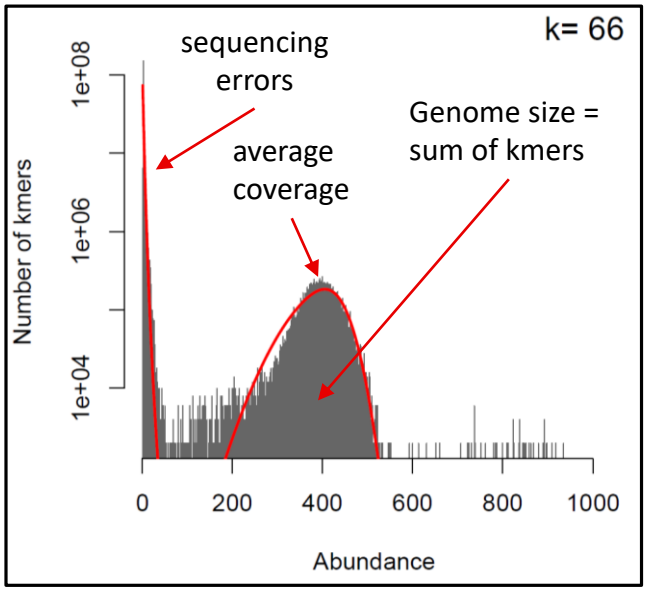
Kmers

k	n kmers
1	4
2	16
3	64
4	256
5	1024
6	4096
7	16384
8	65536
9	262144
10	1.05e6
15	1.07e9
20	1.10e12

- *A more scalable approach computationally breaks the sequence fragments into short fragments, each k bases long = kmers*
- *for short kmers, k=1 .. 10, each kmer occurs many times randomly in a genome because there are only a small number of possible kmers*
- *how long does a kmer need to be to be unique in the genome*
  - bacterial genomes are small, only about 5e6 bases
  - for k=20, the probability of finding the same kmer in different locations is very small P = 5e6/1.1e12 = 4.5e-6
- *assuming sequence is unique (no duplicate sequences)*
- *assuming sequence is random (all letters occur equally)*

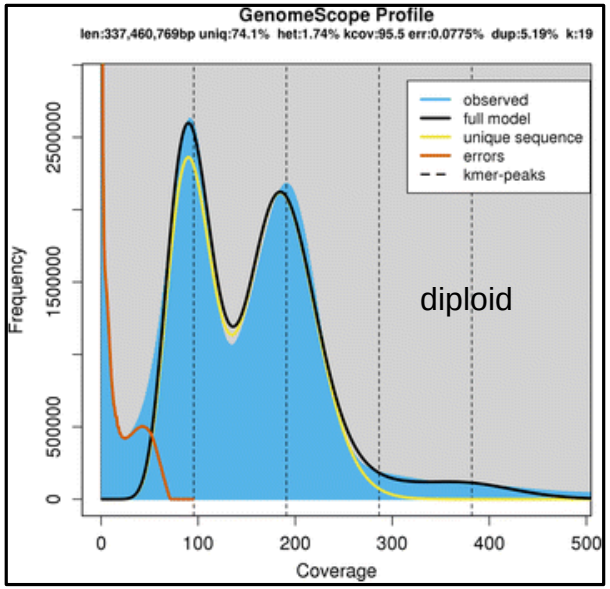
Estimating genome size from kmers

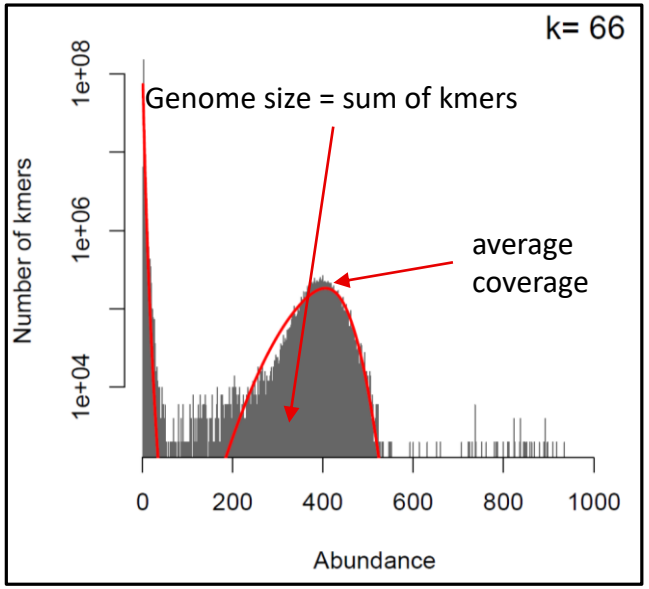
- *One kmer for each unique base in genome (assuming no duplication)*
- *if we deep sequence (e.g., coverage ≥ 100), all kmers should occur 100X +/- some randomness*
- *error sequences have low abundance*
  - in illumina sequencing the error rate is low (<0.1%)
  - most sequence reads have no errors
  - sequences with errors usually only differ by 1-2 bases, and never at identical positions
- *"good" peak ~180-500*
  - 22.99 M unique kmers
  - average coverage 389.9
- *What is the best k? The one that gives the highest number of non-error kmers*



Genome size from kmers

- duplicated sequences look like they have higher coverage
  - bacterial sequences have repetitive sequences
  - rRNA genes (vibrio has 12) are close to identical
  - other genes may be duplicated
  - some regions, such as origins of replication, my be repetitive
- variants in the population create less common than average kmers
  - variants are non-random





Sequence Assembly

Reconstructing the genome sequence from short fragments

- *A metaphor, assume*
  - I am on a desert island and I want to send you the Hamlet text
  - But my radio (telegraph) is defective and all you receive is 8 random letters from somewhere (anywhere) in the text
  - Solution – resend the message multiple times
  - Problem – you have to reassemble the pieces and neither of us knows where in the message each piece came from

  - The message we want to send is the genome
  - The short fragments are sequence reads

Method 2: De Bruijn Graph Assembly

- *A faster approach*
- *break each fragment (read) down into smaller words*
- *index the words so that you can rapidly tell which ones overlap (Burrows-Wheeler transform)*
- *build up a graph of overlapping words*
- *trace the consensus contig*
  - Ideally a path that visits each node exactly once (Hamiltonion path)
  - or visit each edge once (Eulerian path)

Debruijn graph assembly

- Repeat the Hamlet example
- Try for 3X coverage with 8 letter fragments 156 * 3 / 8 ≈ 58 fragments
- You can see some words occur more that once

```
obeornot
ottobeth
obethati
bethatis
hatisthe
uestionw
stionwhe
ionwheth
nwhether
whethert
ethertis
ethertis
hertisno
hertisno
hertisno
blerinth
lerinthe
lerinthe
erinthem
inthemin
... 38 more
```

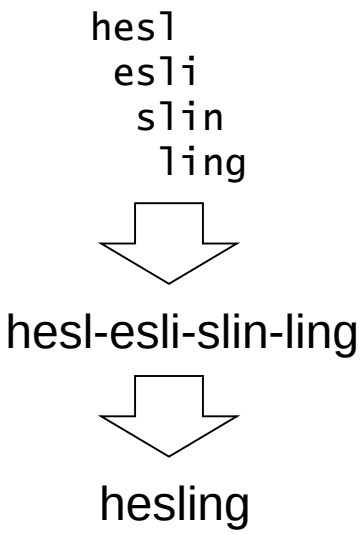
De Bruijn Graph Assembly

- *for this example I'm using 4mer words*
  - for bacteria we need to use something much bigger
  - long enough that words don't randomly match
- *tips are words with no overlap*
  - left tips: hesl otto bler uest indt posi bles eous
  - no overlap on left
  - right tips: ndth oppo isno emin rnot oftr fout
  - no overlap on right

agai 3	gsan 1	osin 1	tune 3
ains 4	hati 3	osuf 3	uest 1
akea 3	hemi 1	otak 3	uffe 2
anda 2	hert 6	otto 1	uneo 3
andb 2	hesl 1	ousf 1	usfo 1
aoft 2	heth 3	owso 4	whet 3
arms 1	indt 1	posi 1	wsof 3
arro 4	inge 4	rint 4	yopp 5
asea 4	ings 1	rmsa 1	
atis 2	inst 4	rnot 1	
beor 1	inth 5	rows 3	
beth 3	ionw 3	rrow 3	
bler 1	isno 3	rtis 5	
bles 1	isth 1	rtot 4	
byop 5	kear 2	rtun 2	
darr 2	leri 3	saga 2	
dbyo 5	lesa 1	sand 3	
dtos 1	ling 1	seao 3	
eaof 2	msag 1	sfor 1	
earm 1	ndar 2	sing 1	
emin 1	ndby 5	slin 1	
endt 3	ndth 3	sofo 1	
eorn 1	ndto 1	stas 3	
eort 5	neor 4	sthe 1	
eous 1	ngen 4	stio 2	
erin 4	ngsa 1	suff 2	
erti 5	nsta 4	take 2	
esan 1	nthe 4	tase 3	
esli 1	nwhe 3	that 2	
esti 1	obeo 1	them 2	
etha 2	obet 2	ther 4	
ethe 4	ofou 1	tion 2	
fert 2	oftr 2	tisn 3	
ffer 2	onwh 2	tist 1	
fort 2	oppo 4	tobe 1	
fout 1	orno 1	tosu 1	
gain 3	orto 4	tota 5	
gend 3	ortu 1	ttob 1	

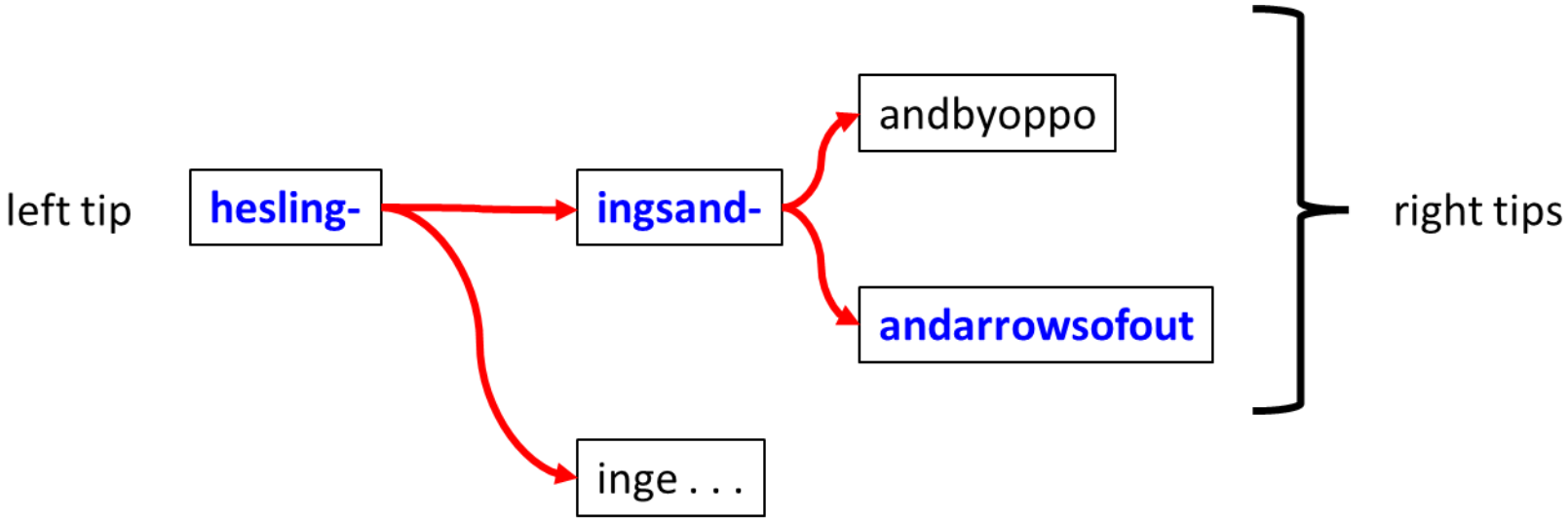
De Bruijn Graph Assembly

- *Overlapping words can be found very quickly using Burrows-Wheeler transform (essentially a suffix tree)*
- *number of words is proportional to N (the genome size), not N^2*
  *!! **NOT QUADRATIC***
- *Step 1: compress overlapping words with branches to produce the nodes in the De Bruijn grap . words must overlap by k-1 letters.*
- *Example of compression*

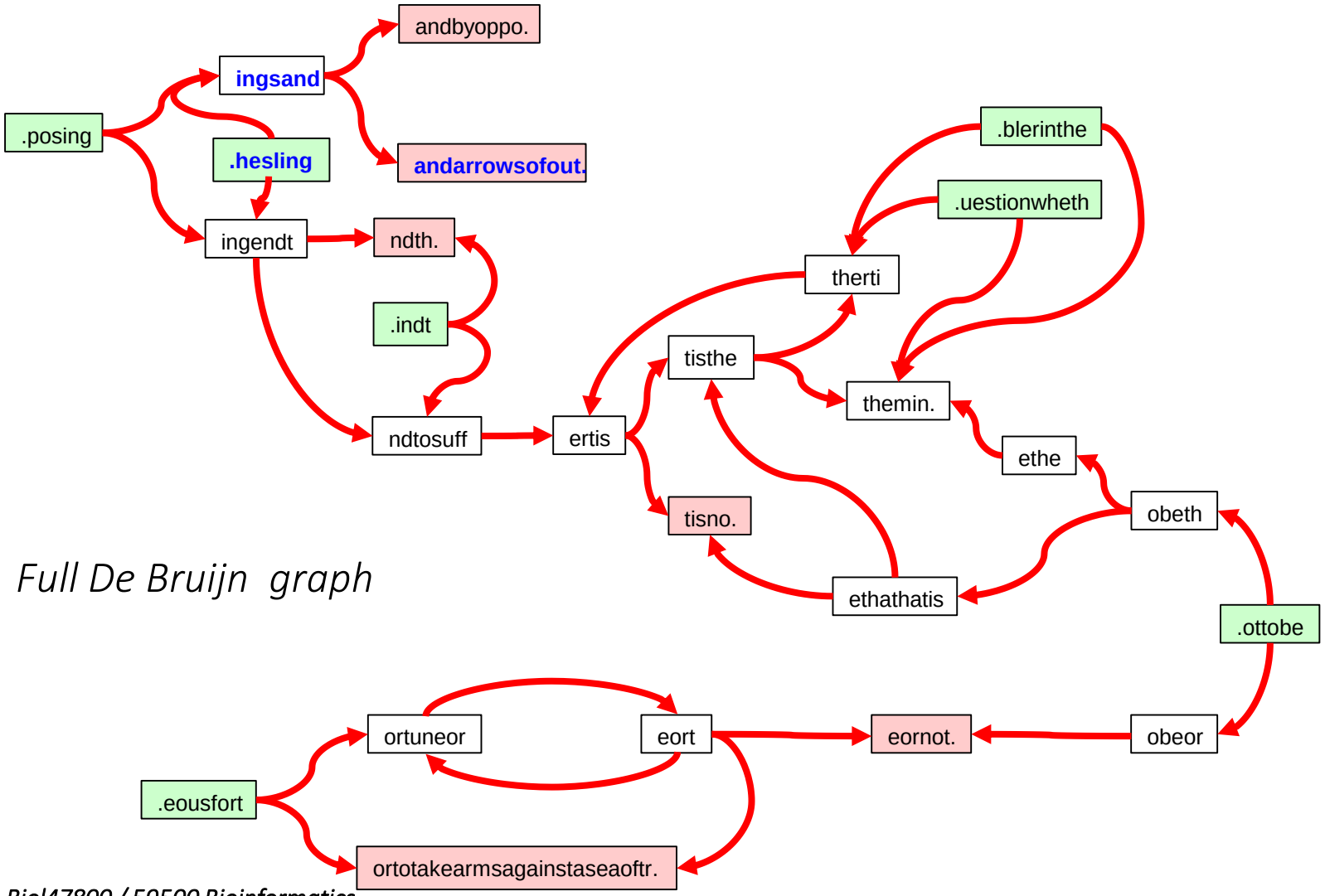


De Bruijn Graph Assembly

- *Step 2: Join nodes together according to overlaps*



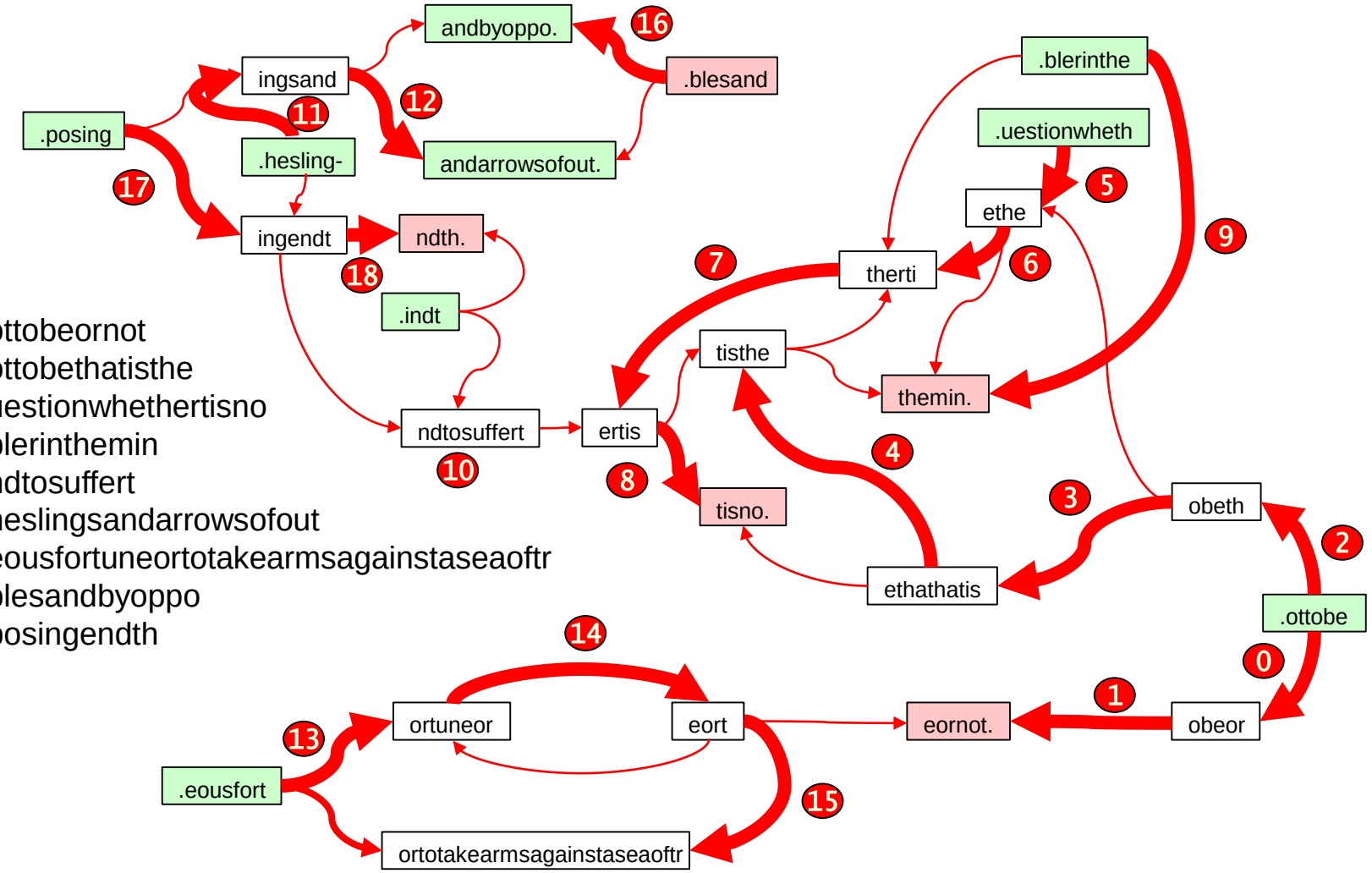
De Bruijn Graph Assembly



- *Full De Bruijn graph*

De Bruijn Graph Assembly

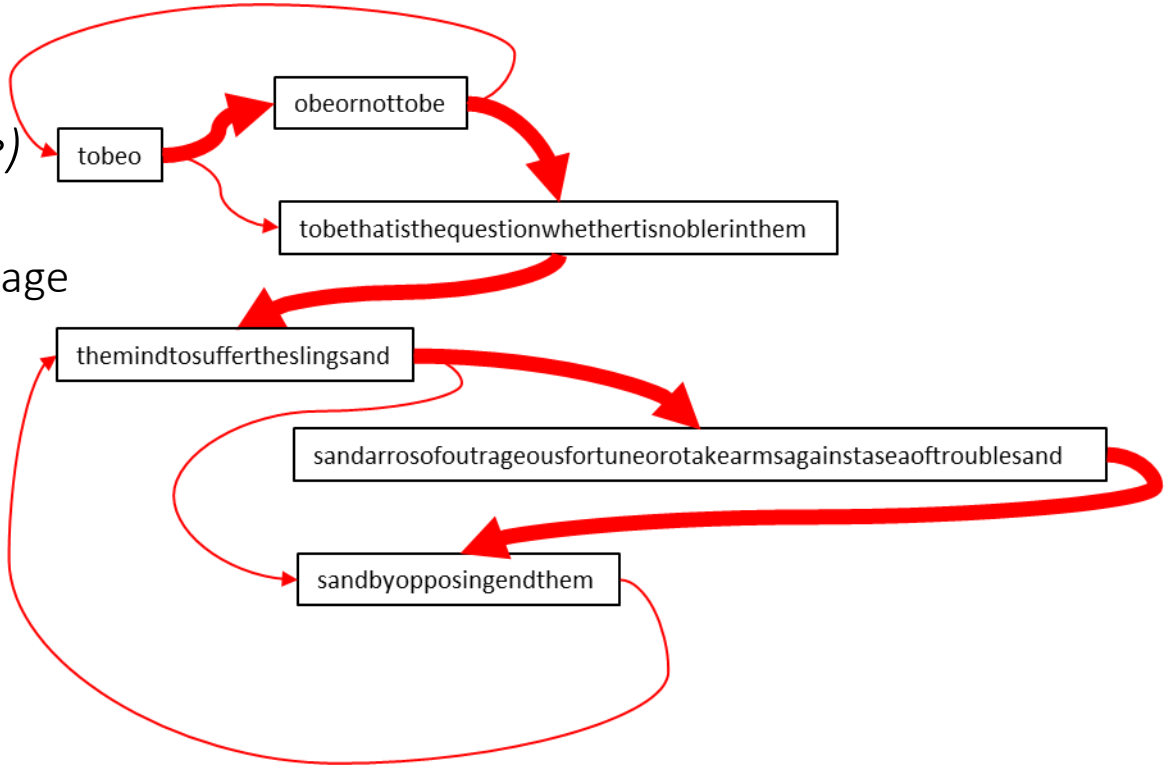
- *Step 3: trace contig sequences*



0 – 1 ottobeornot
2 – 4 ottobethatisthe
5 – 8 uestionwhethertisno
9 blerinthemin
10 ndtosuffert
11 – 12 heslingsandarrowsofout
13 – 15 eousfortuneortotakearmsagainstaseaoftr
16 blesandbyoppo
17 – 18 posingendth

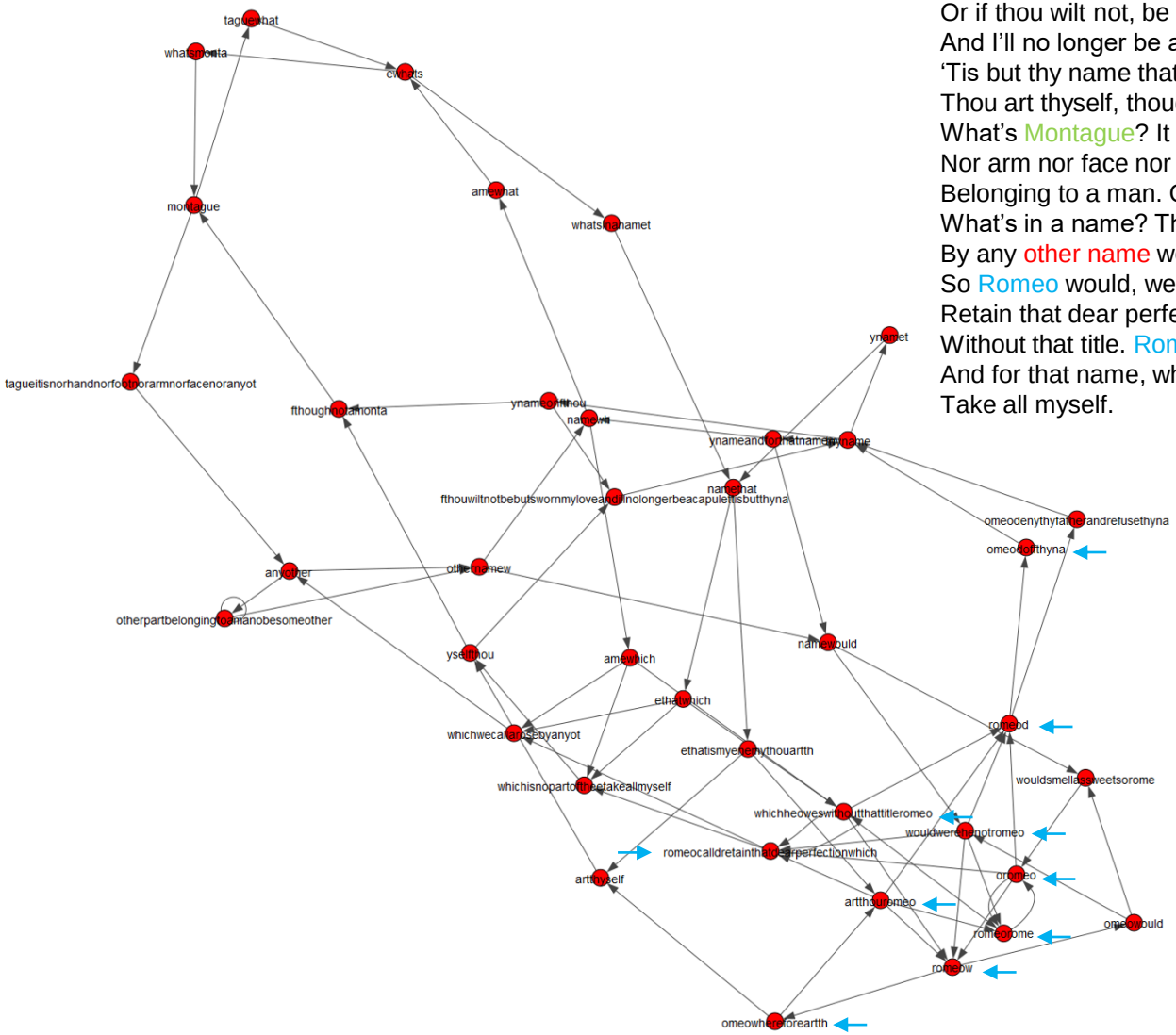
De Bruijn Graph Assembly

- *Difficult to trace correct message (genome sequence)*
  - coverage is too low to completely recover the message
  - repeats cause cycles in the Debruijn graph
  - k is to short to resolve repeats
- *example with complete coverage and 5mer words*



Genome Assembly

k=6

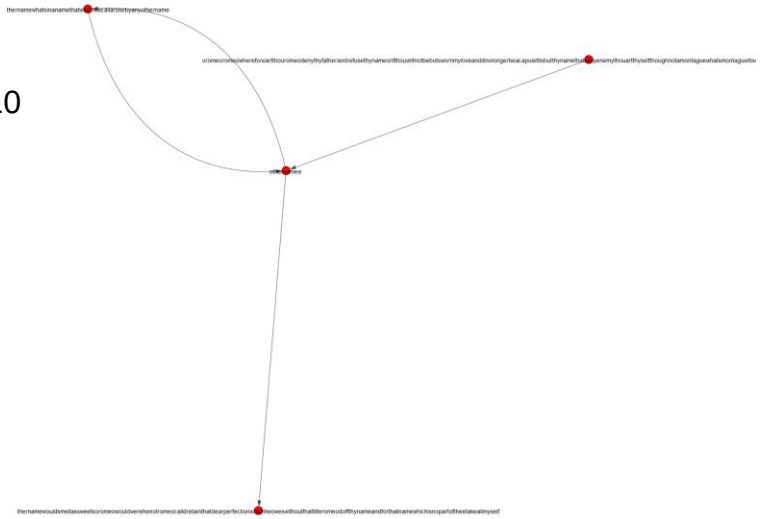


O Romeo, Romeo, wherefore art thou Romeo?
Deny thy father and refuse thy name.
Or if thou wilt not, be but sworn my love
And I'll no longer be a Capulet.
'Tis but thy name that is my enemy:
Thou art thyself, though not a Montague.
What's Montague? It is nor hand nor foot
Nor arm nor face nor any other part
Belonging to a man. O be some other name.
What's in a name? That which we call a rose
By any other name would smell as sweet;
So Romeo would, were he not Romeo call'd,
Retain that dear perfection which he owes
Without that title. Romeo, doff thy name,
And for that name, which is no part of thee,
Take all myself.

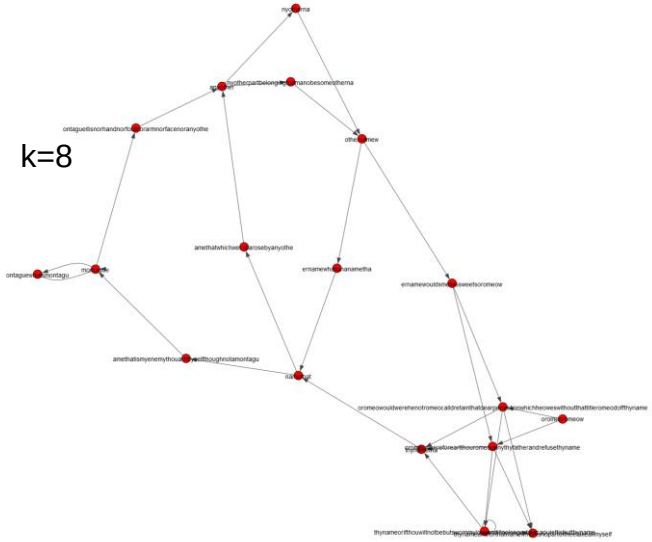
Genome Assembly

k=7

k=10



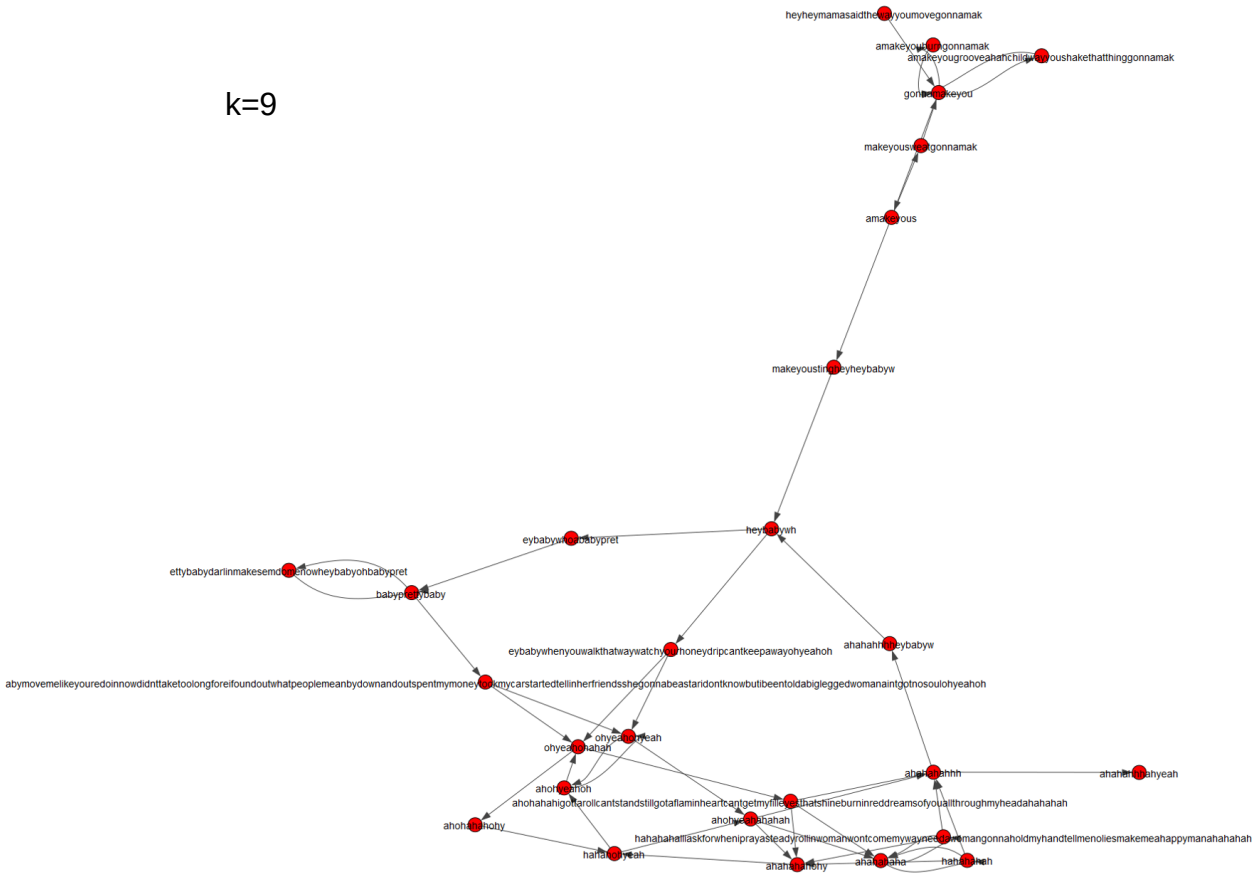
k=8



increasing k improves the graph when k > repeat length

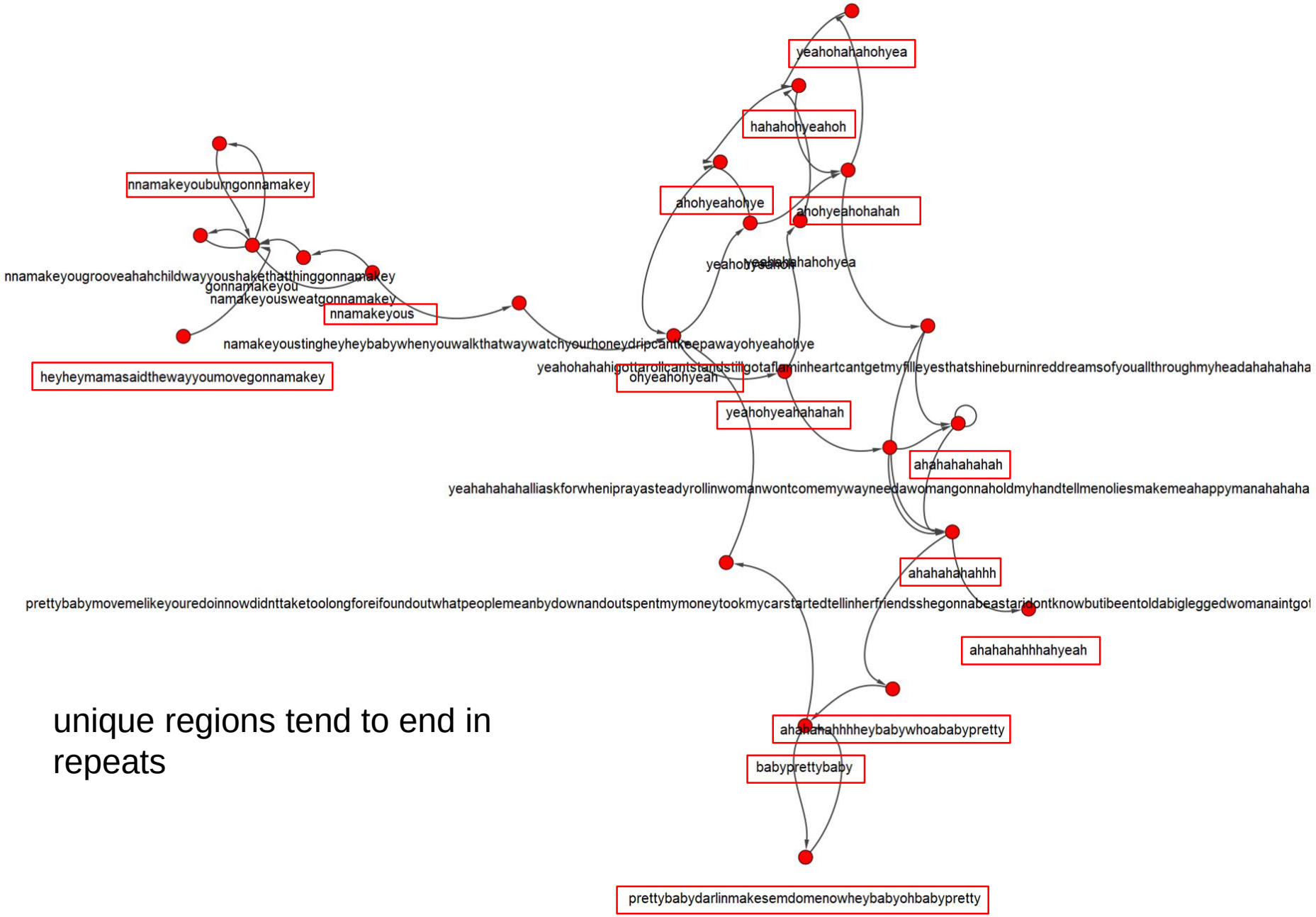
Genome Assembly

A repetitious text



Black Dog
Led Zeppelin (1971)
https://www.youtube.com/watch?v=2KPEHohJMuw

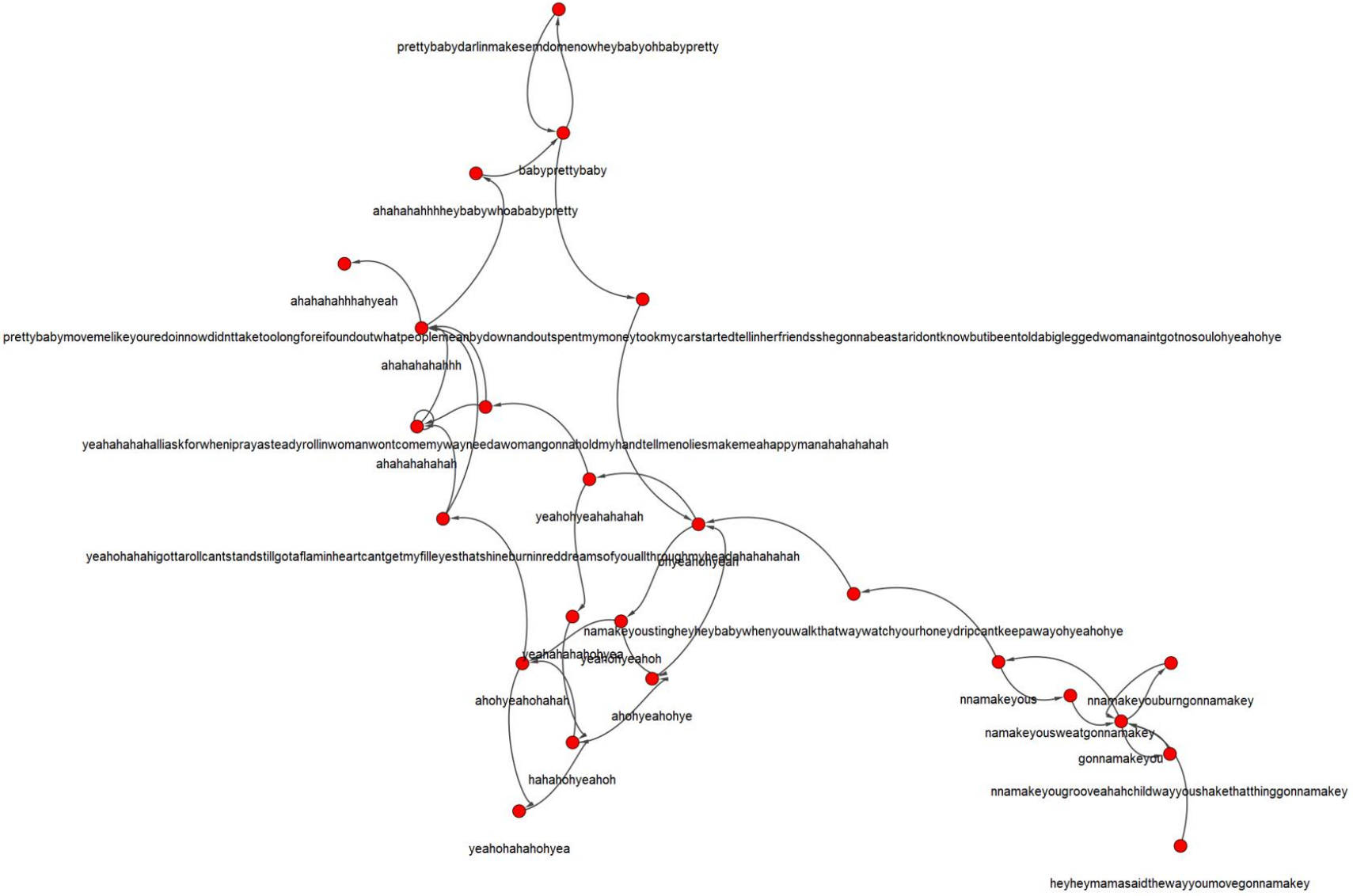
Hey hey mama said the way you move
Gonna make you sweat, gonna make you groove
Ah, ah, child, way you shake that thing
Gonna make you burn, gonna make you sting.
Hey hey baby when you walk that way
Watch your honey drip, can't keep away
Oh yeah, oh yeah, oh, ah, ah
Oh yeah, oh yeah, oh, ah, ah.
I gotta roll, can't stand still
Got a flamin' heart, can't get my fill
Eyes that shine, burnin' red
Dreams of you all through my head
Ah ah, ah ah, ah ah, ah ah, ah ah, ah ah, ahhh
Hey, baby, whoa baby, pretty baby
Darlin' makes 'em do me now
Hey, baby, oh baby, pretty baby
Move me like you're doin' now
Didn't take too long 'fore I found out
What people mean by down and out
Spent my money, took my car
Started tellin' her friends she gonna be a star
I don't know, but I been told
A big-legged woman ain't got no soul
Oh yeah, oh yeah, ah, ah, ah
Oh yeah, oh yeah, ah, ah, ah
All I ask for when I pray
A steady rollin' woman won't come my way
Need a woman gonna hold my hand
Tell me no lies, make me a happy man
Ah ah, ah ah, ah ah, ah ah, ah ah, ah ah, ahhh.
Ah, yeah!'"



unique regions tend to end in repeats

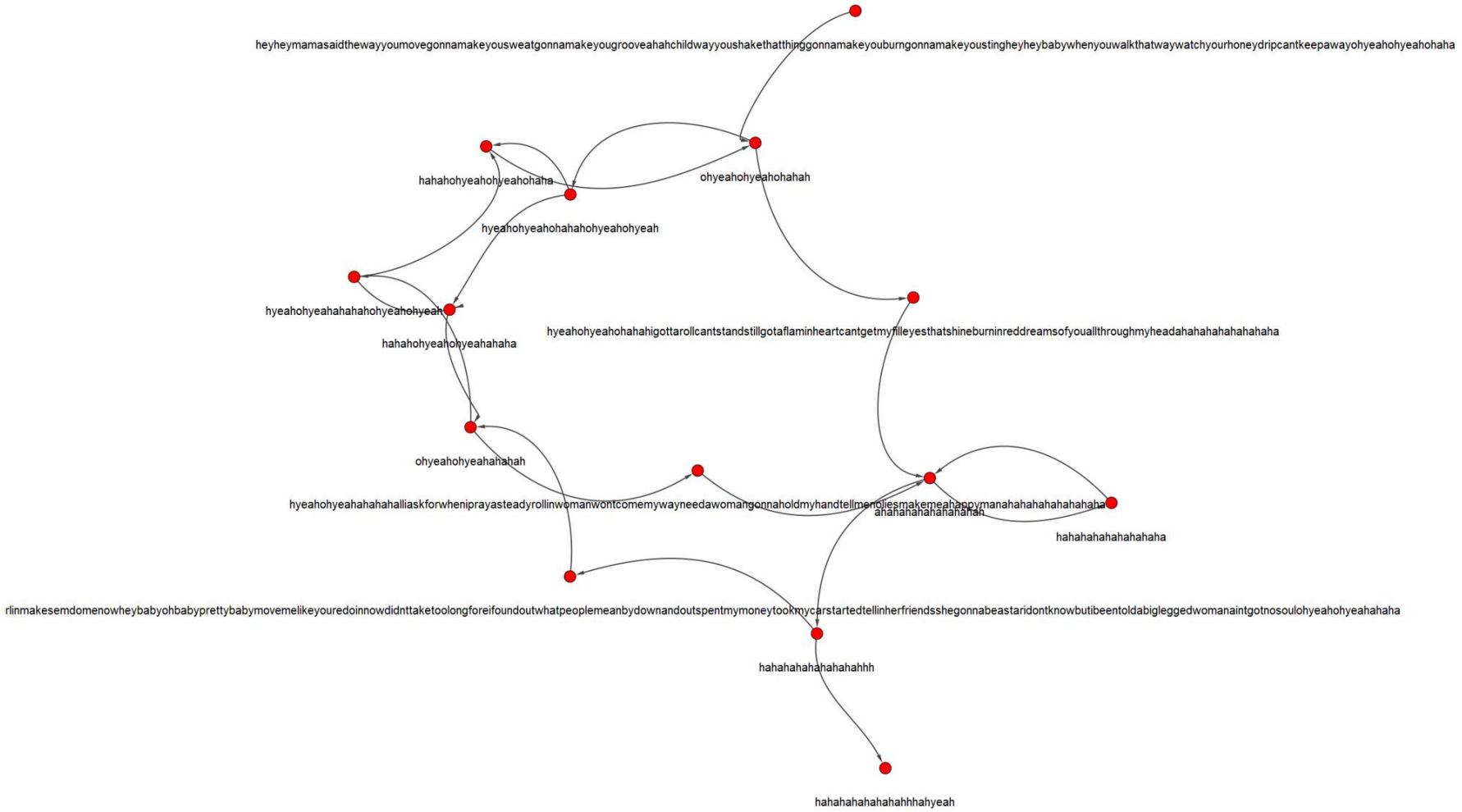
Genome Assembly

k=11



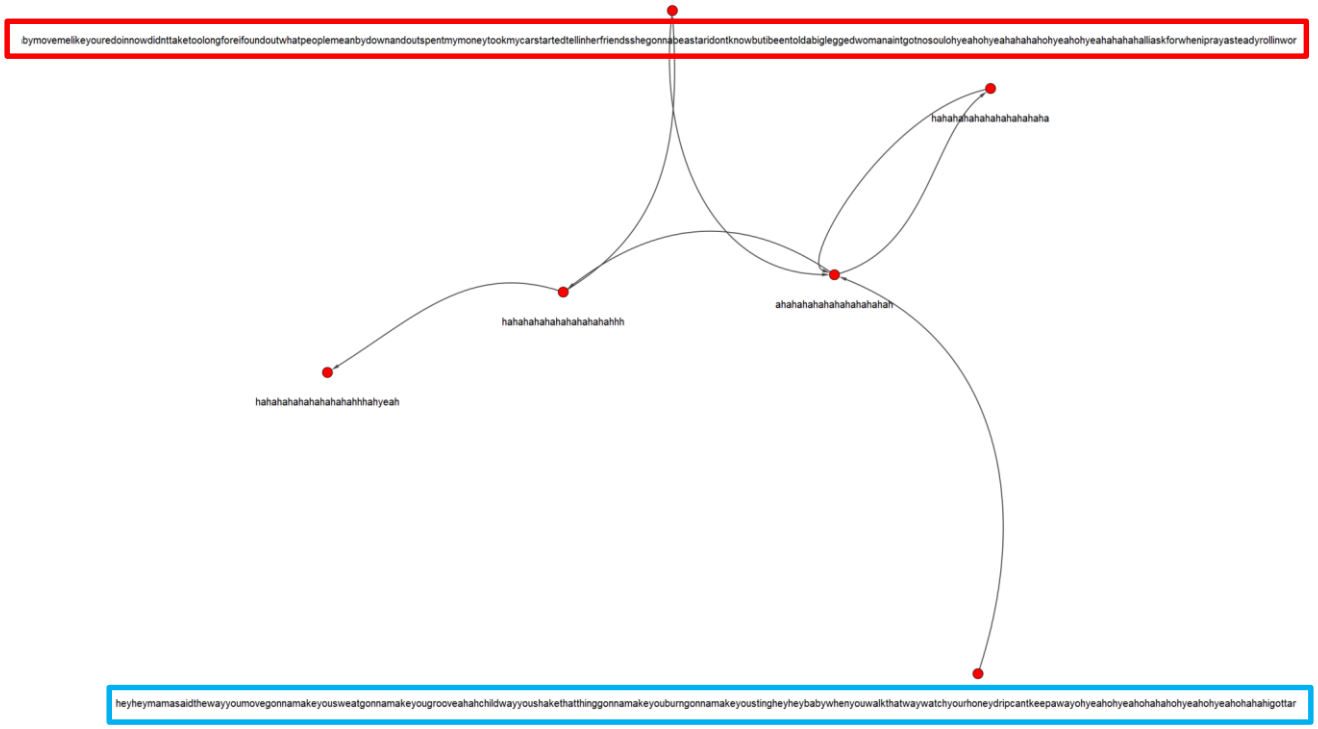
Genome Assembly

k=18



Genome Assembly

k=22



Hey hey mama said the way you move
Gonna make you sweat, gonna make you groove
Ah, ah, child, way you shake that thing
Gonna make you burn, gonna make you sting.
Hey hey baby when you walk that way
Watch your honey drip, can't keep away
Oh yeah, oh yeah, oh, ah, ah
Oh yeah, oh yeah, oh, ah, ah.
I gotta roll, can't stand still
Got a flamin' heart, can't get my fill
Eyes that shine, burnin' red
Dreams of you all through my head
Ah ah, ah ah, ah ah, ah ah, ah ah, ah ah, ahhh
Hey, baby, whoa baby, pretty baby
Darlin' makes 'em do me now
Hey, baby, oh baby, pretty baby
Move me like you're doin' now
Didn't take too long 'fore I found out
What people mean by down and out
Spent my money, took my car
Started tellin' her friends she gonna be a star
I don't know, but I been told
A big-legged woman ain't got no soul
Oh yeah, oh yeah, ah, ah, ah
Oh yeah, oh yeah, ah, ah, ah
All I ask for when I pray
A steady rollin' woman won't come my way
Need a woman gonna hold my hand
Tell me no lies, make me a happy man
Ah ah, ah ah, ah ah, ah ah, ah ah, ah ah, ahhh.
Ah, yeah!"'

De Bruijn graph assembly

- *Issues*
  - Errors increase the complexity of the graph
    - Repeats create bubbles and cycles in the De Bruijn graph
    - Sequencing errors create extraneous tips (because the words don't overlap with anything)
  - Repeats greatly increase the complexity of the graph by creating cycles
    - To resolve repeats requires read pairs that are further apart than the repeat length
  - The graph is usually simplified (pruned) using
    - kmer / sequence depth

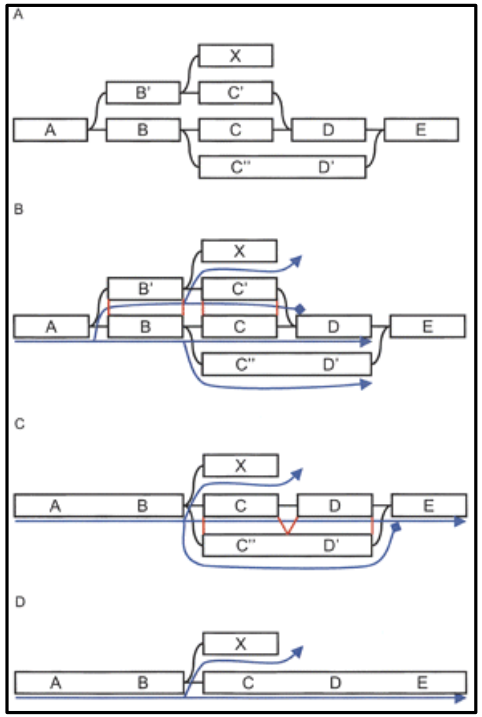
      repeats have greater depth

      singletons are usually errors
    - paired reads typically 350-500 base (do not overlap)

      should have correct direction and spacing in assembly

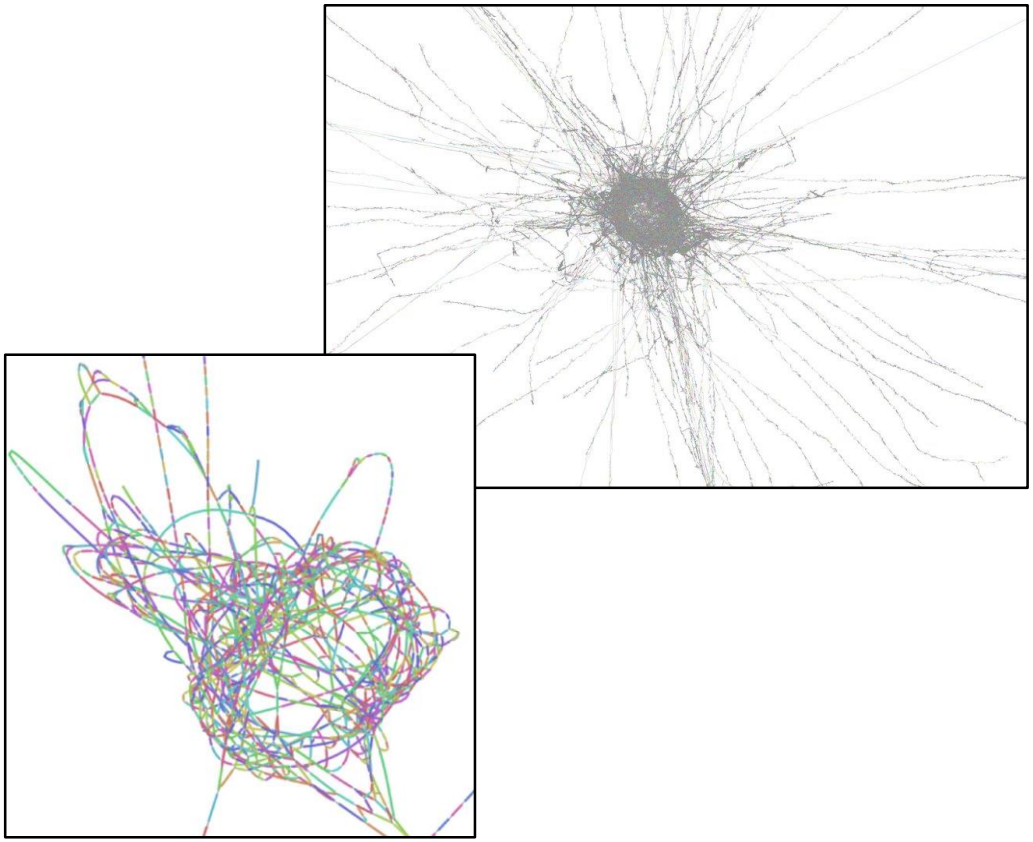
Velvet Assembler

- *One of the first De Bruijn assemblers*
- *Pruning*
  - tips – a chain of nodes disconnected on one end
    caused by sequencing errors OR coverage gaps
    errors tend to be short (rule trim if < 2 kmer)
    errors tend to have low multiplicity at junction
  - bubbles – paths that leave and return
    caused by sequence variation (SNPs)
    length/multiplicity rule
    shorter, higher multiplicity paths are preferred
  - Erroneous connections
    duplicate sequences + errors
    errors will have low coverage, so will areas with low coverage



De Bruijn graph

- *Real De Bruijn graphs are very complicated*



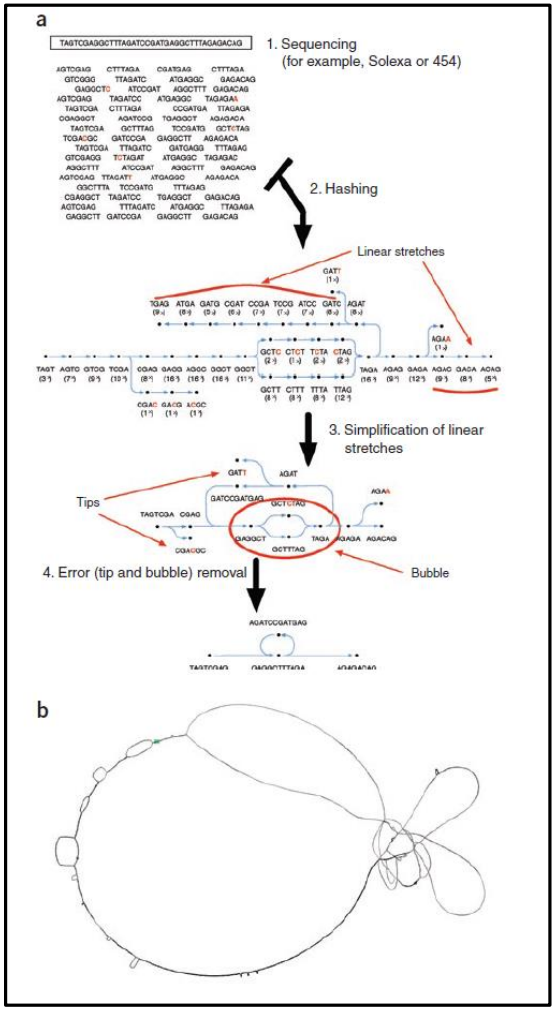


Fig 3. Flicek & Birney, 2009

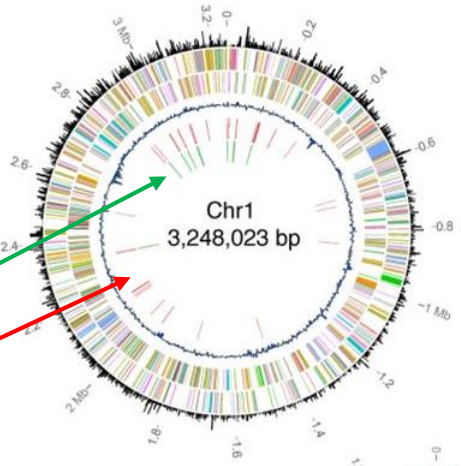
Problems with De Bruijn assembly

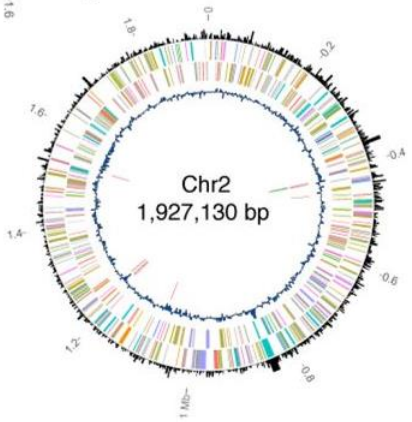
- *Reconstructed sequence is fragmented due to lack of overlap*
  - Get high coverage NGS (50X +)
- *Read errors (and polymorphisms) cause bubbles (false edges and nodes)*
  - Every sequencing error creates k-1 spurious kmers
  - hard to distinguish from natural variation (heterozygosity, polymorphism)
  - kmer count can identify random errors
- *Kmer issues*
  - too short – many bubbles and false overlaps
  - too long – overlaps missed due to sequence errors
  - kmers should be long enough to be unique in coding regions
    - usually lengths from 30-120, but it's difficult to predict best length
- *Repeats*
  - Real biological sequences contain long repeated sequences

Debruijn lessons

- *the right k depends on the length of the text*
- *the right k depends on amount of repetitive text*
- *complete unambiguous assembly is impossible when*
  - *k* is less than the length of repeats
  - the lengths of the sequence reads is less than the length of the repeats

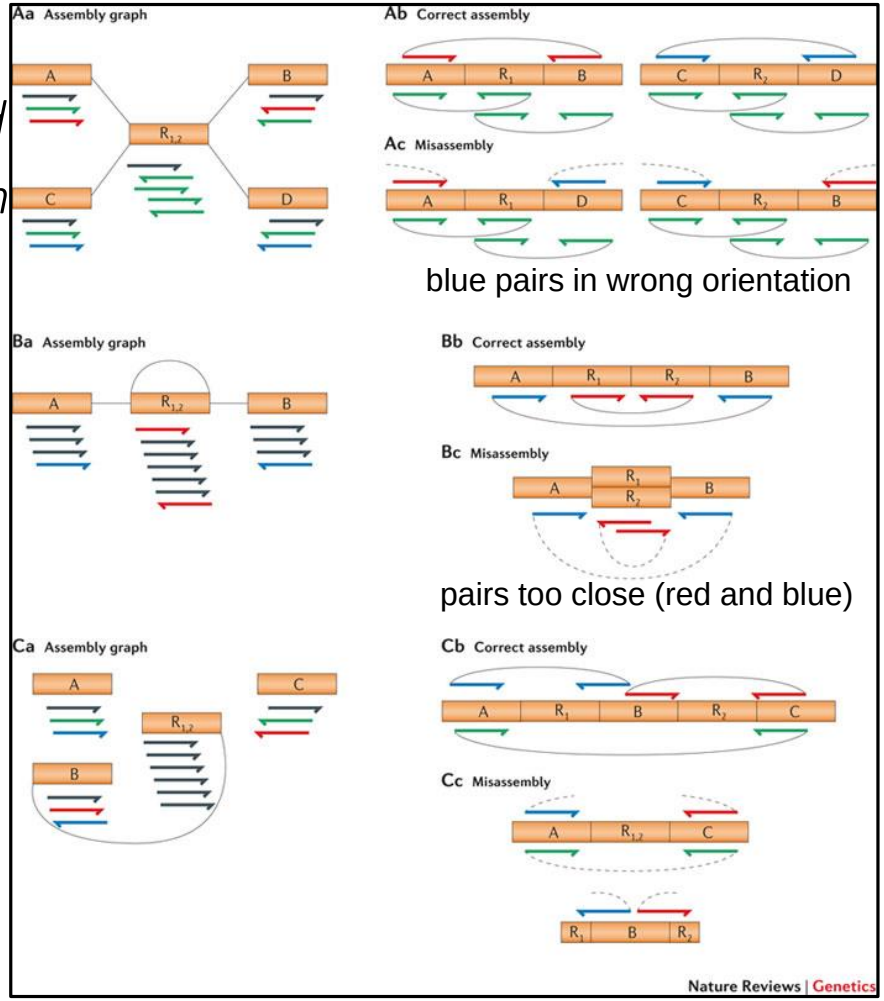
- *Even bacteria are repetitious*
- *Vibrio natriegens*
  - text = 5,175,153 base pairs, two chromosomes
    - 3,248,023 bp and
    - 1,927,310 bp
  - *11 rRNA operons*
  - *129 tRNA*



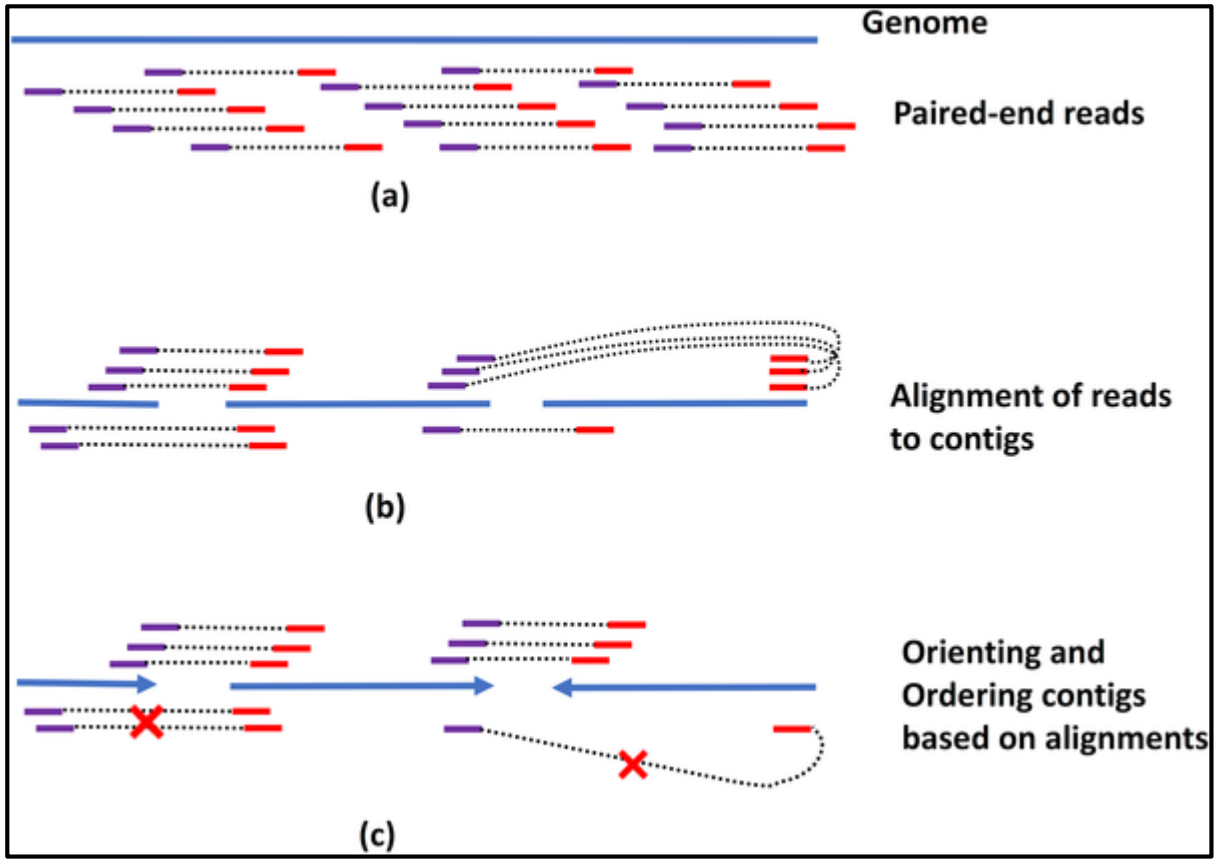


Repeats

- *Can you detect the misassemblies caused*
- *Yes! Repeats result in systematic errors in genome model*
- *You can see*
  - compression
  - expansion
  - misorientation
- *Programs*
  - Quast
  - REAPR
  - CGAL
  - ALE

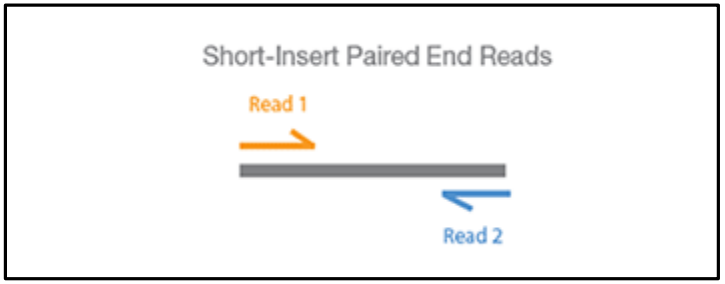


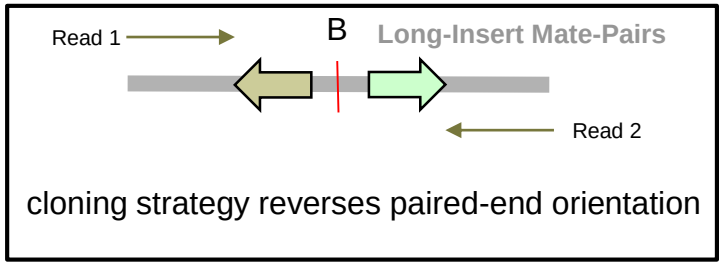
Scaffolding the assembly

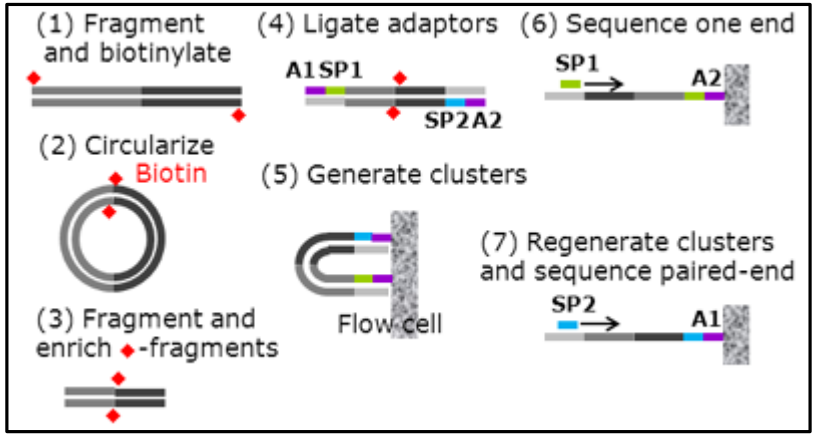


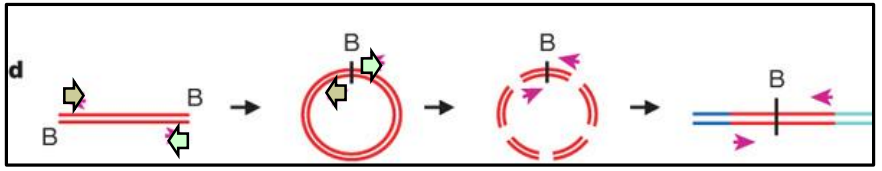
Repeats/Scaffolding

- *Must have some information that allows distant reads to be placed with respect to each other*
- *paired end reads (100s of bases)*
- *mate-pair reads (1000s of bases)*
- *long clones such as fosmids*
  - not used much now
- *long reads – ONT or PacBio (5-30 kbase)*





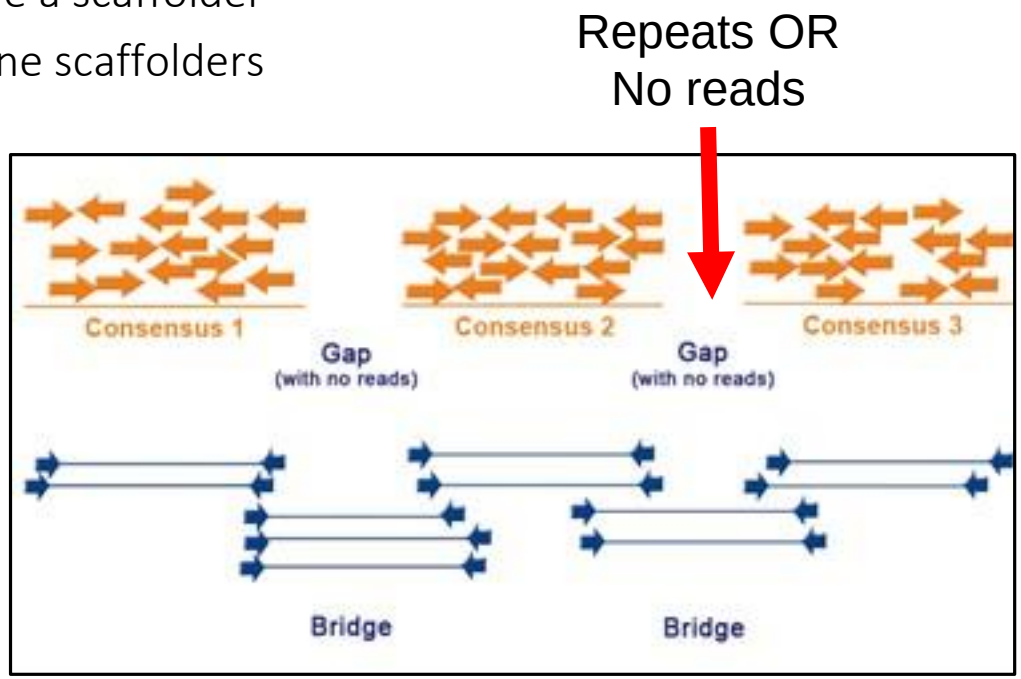




Mate Pairs: One common technique
- isolate long fragment, e.g., 5 kb
- circularize (including tag at junction) fragment
- isolate desired size fragment
- attach adapters and sequence

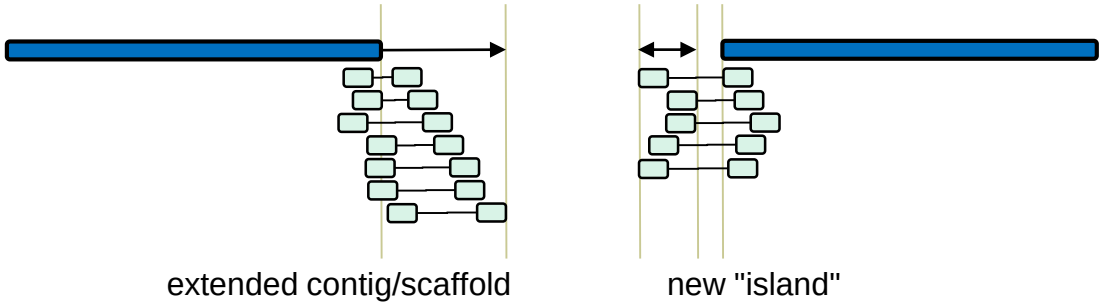
Scaffolding and gap filling/closing

- *Scaffold – contigs with defined orientation, order, and spacing (Ns)*
  - Spacing (number of Ns) is determined from the insert size in library
  - Library size errors make estimate of spacing less accurate
  - typically uses both paired-end and mate-pair information, or long-read sequences
  - most assemblers include a scaffolder
  - there are also standalone scaffolders



Gap filling/closing

- *Gap Filling/closing/finishing/polishing*
  - Perform initial assembly
  - scaffold to estimate sizes of gaps
  - search for paired reads that have one end in a contig
  - attempt to fill in gap region
  - many gaps can be closed with repetitive application of gap filling
  - gap filling is easier than initial assembly
    - most reads have been placed
    - size information limits which reads can be used



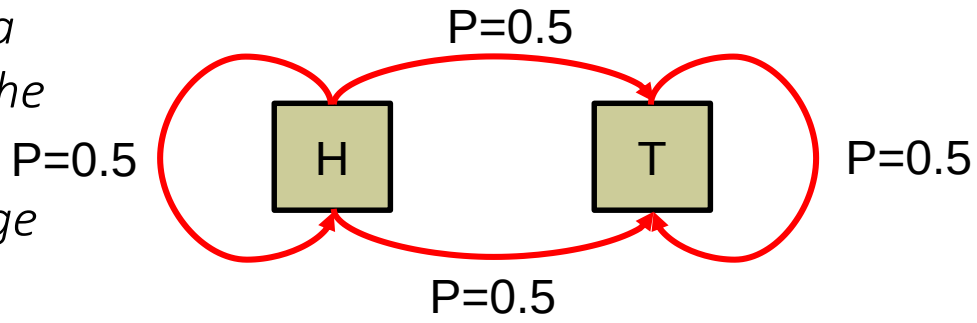
Biol 47800 Introduction to Bioinformatics

Week 10

- *Gene Modeling*
- *Genome Functional Annotation*
  - Comparative/extrinsic
  - Combined
- *Review*

Markov Models

- *A Markov model is a probabilistic process made up of states and transitions*
- *Transitions between states have probabilities*
- *the Dayhoff PAM model is a Markov model that gives the probability that an ancestral residue will change another during a unit of evolutionary time*

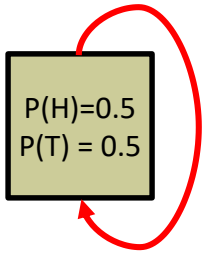


- *The PAM model is a first order Markov model since the new residue depends only on the current residue*
- *The coin flipping model, above, is a zero order model – transitions have the same probability whether the current state is heads or tails*

Gene Modeling

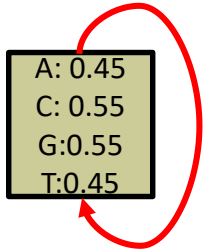
Markov Models

- *Often states are associated with an action. A common one is to emit a letter with a certain probability (emission probability).*
- *A Markov model that simulates tossing a fair coin*



P=1.0

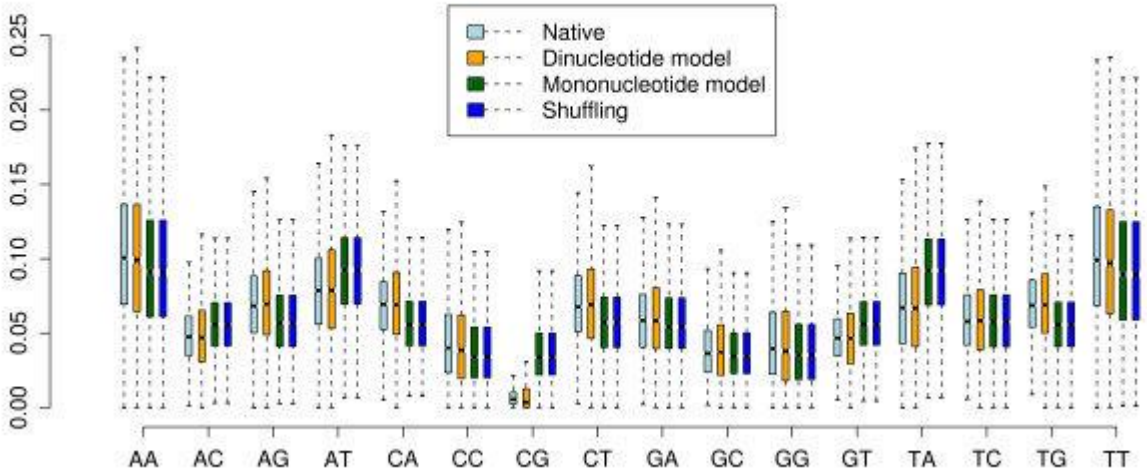
- *A Markov model for generating a random DNA sequence (45% AT) could be*



P=1.0

Markov Models

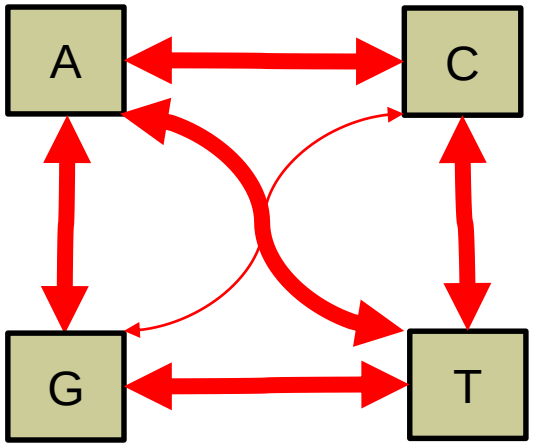
- *In biological sequence, the next base or residue often depends on the current base*
- *This can be clearly seen in the much lower frequency of CG sequences in eukaryotes*



Gene Modeling

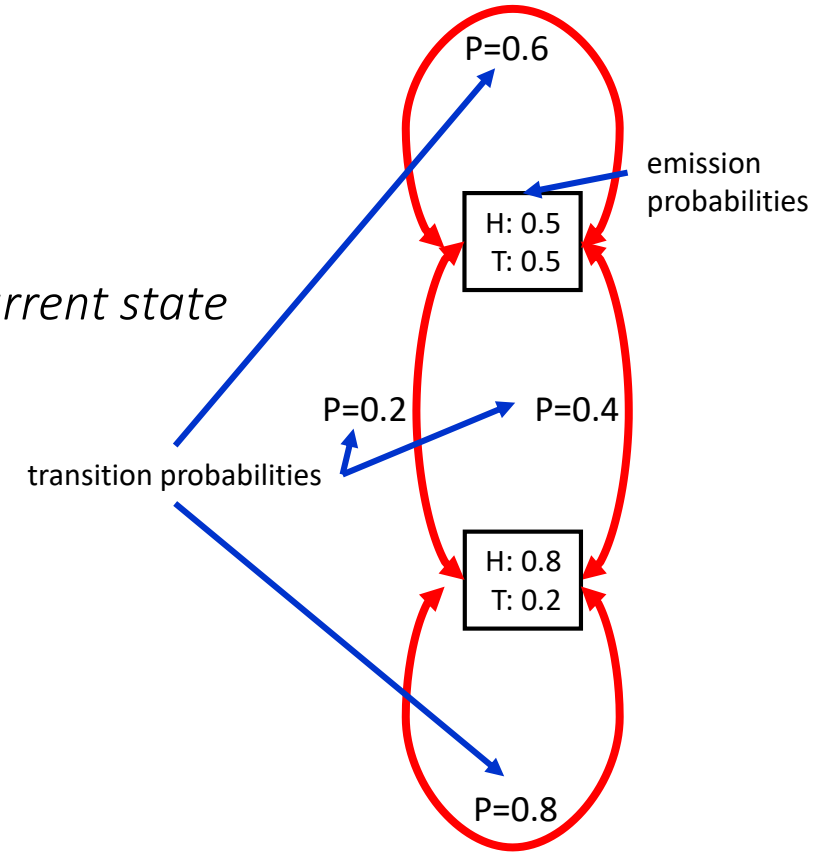
Markov Models

- *1st order DNA model (dinucleotide model)*
- *Experience says that good random DNA models require 5th or 6th order models*
- *sixth order model has $4^6 = 4096$ states*



Markov Models

- *Two coins, one fair, one unfair*
- *states: fair, unfair*
- *first order, transition depends on current state*

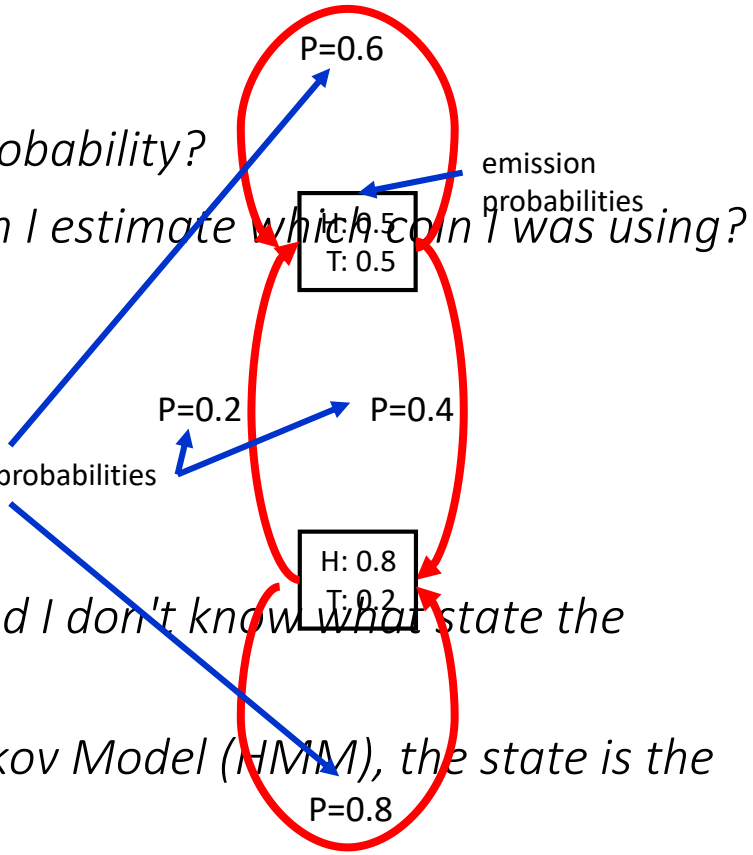


Gene Modeling

Markov Models

- *What if I secretly switch coins with some probability?*
- *If I generate some sequence of H and Ts can I estimate which coin I was using? For instance*

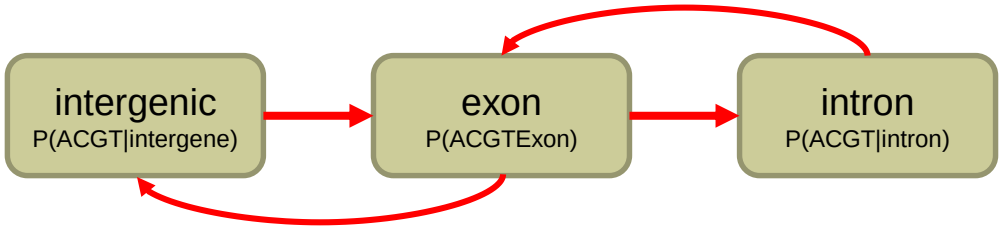
HTHHHTHHTTHTHTHTHTHTHHHTHTH
UUUUUUUUFFFFUUFFFFFFUUUUUFF



- *In this case I only have the model output and I don't know what state the model is in at each point*
- *If I don't know the state, it is a Hidden Markov Model (HMM), the state is the hidden information that we want to find*

Markov Models

- *Why do I want to guess the coin?*
- *Consider a simple model of a gene*



- *Can we guess the states if we know the sequence of A, C, G, T emitted by this machine?*

Gene Modeling

Markov Models

- *Discovering the state in a HMM*

  For HTHHTHTTHHH

  HTHHTHTTHHH

- *I first need to know the initial probability that I am in state F (fair) or unfair (U)*
  - the prior probability - when I have no observations
- *Then I can update my estimate based on my first observation, H*

- *By Bayes' rule* $\boldsymbol{P(F|H) = \frac{P(F)P(H|F)}{P(H)}}$

  - P(H|F) = prob. head in fair state = 0.5
  - P(F) = prob. of fair state = .33 (prior probability of fair state, I made this up)
  - P(H) = marginal probability of H = average fraction of heads

  - P(F|H) = P(H|F) P(F) / P(H) = 0.238
  - P(U|H) = P(H|U) * P(U) / P(H) = 0.761

When I have evidence of just one head

Markov Models

HTHHTHTTHHH

- *Next flip is T*
- *the prior probability of F and U are the conditional probabilities based on the first H, times the transition probabilities (the probability I change coins)*

  - P(F) = P(F|H) * P(F→F) + P(U|H) * P(U→F) = 0.238 *0.6 + 0.761*0.2 = 0.295
  - P(U) = P(U|H) * P(U→U) + P(F|H) * P(F→U) = 0.761*0.8 + 0.238*.4 = 0.704

  - P(F|HT) = P(T|F) * P(F) / P(T) = 0.5 * 0.295 / 0.289 = 0.511
  - P(U|HT) = P(T|U) * P(U) / P(T) = 0.8 * 0.704 / 0.289 = 0.488
  - P(T) = P(T|U)*P(U) + P(T|F)+P(F) = 0.5*0.295 + 0.2*0.704 = 0.289

Markov Models

- *continuing for each flip, I calculate the following probabilities for the two states*

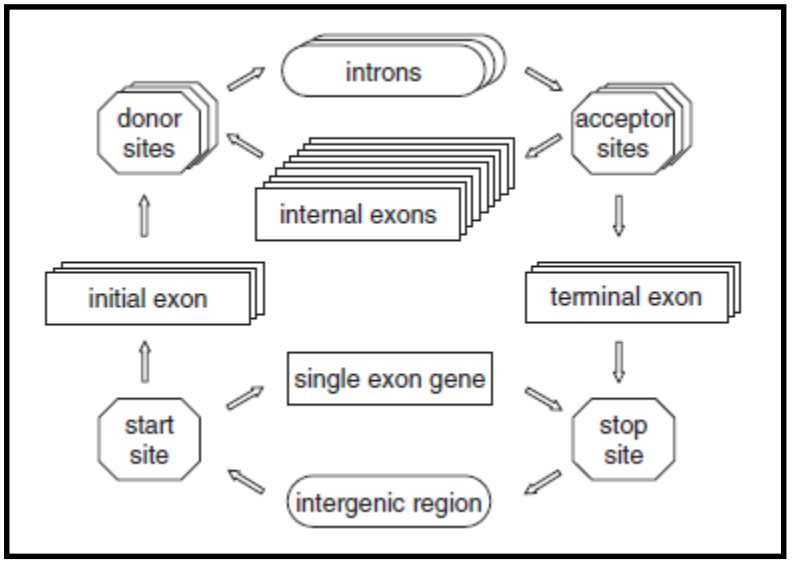
	H	T	H	H	T	H	T	T	H	H	H
Fair	0.24	0.51	0.30	0.23	0.51	0.30	0.54	0.64	0.34	0.24	0.21
Unfair	0.76	0.49	0.70	0.77	0.49	0.70	0.46	0.36	0.66	0.76	0.79

- *Now I can guess which coin I am using at each step, the most probable one!*

Gene Modeling

Markov Models for Gene Prediction

- *Genemark models (http://exon.biology.gatech.edu/)*
  - 3 forward, three reverse reading frames
  - 5th order model (4096 hexamers)
  - Non-coding (homogeneous model)
  - Initial, internal, terminal exons, and single exon genes
  - Introns
  - Intergenic regions
  - Initiation site, termination site
  - Splice donor, splice acceptor



Markov Chains – GeneMark

- *E. coli*

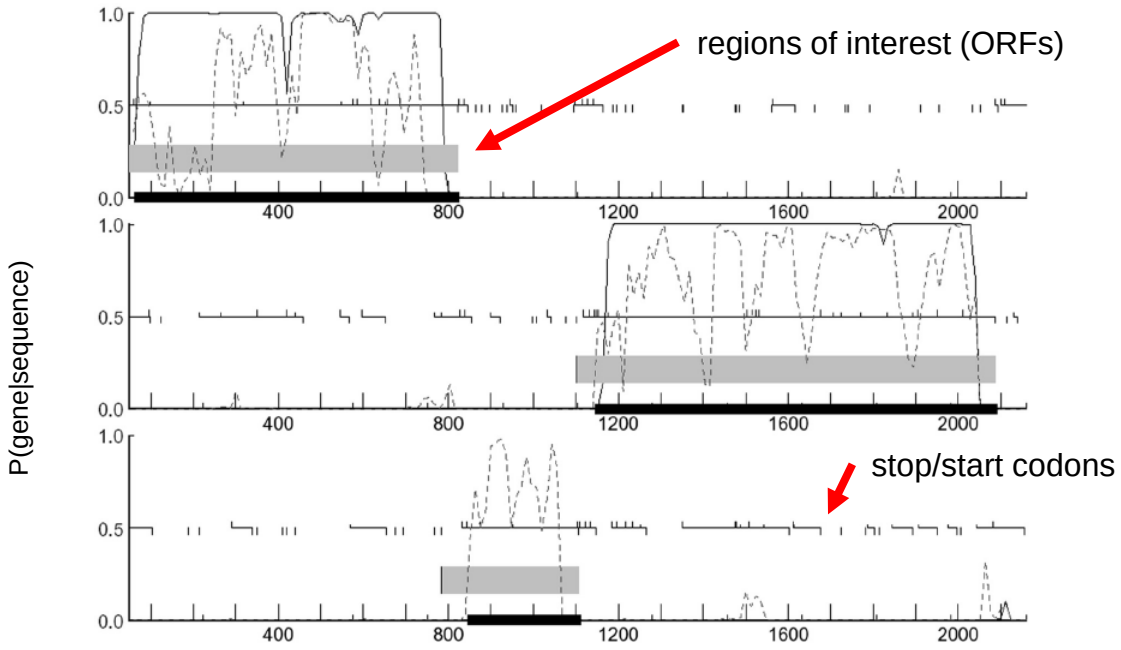


Figure 1. Graphical output from the combination of GeneMark and GeneMark.hmm for a fragment of the *Escherichia coli* K12 genome. The solid black and dashed traces indicate the coding potential calculated by the GeneMark program using the Typical and Atypical Markov chain models of coding DNA, respectively. Only the three reading frames in the direct strand are shown as there are no genes (either predicted or annotated) on the reverse strand in this section of the genome. The thick black horizontal bars indicate the locations of the predictions made by GeneMark.hmm. The thick grey horizontal bars indicate 'regions of interest' provided by the GeneMark program. The thin black horizontal lines indicate (longest) ORFs observed in each reading frame; ticks extending above and below this line indicate potential start and stop codons, respectively.

Gene Modeling

Markov Models

- *Markov models are "trained" on known well-annotated genes*
- *Every organism is different!*
- *How much data might we need?*
  - $4^6 = 2^{12}$ words = 4096 6mer words
  - Assume we want a good count, say at least 10 of each word
  - Assume that the rarest word is 0.01 times as frequent as the average
  - 4096 * 100 * 10 = 4,096,000 bases
  - 4Mb x 7 models (6 exon , 2 intron model, 1 intergenic) = 48 Mb
  - Need well annotated sequence (i.e., gold standard)
- *What if some words still don't appear*
  - Interpolated model (Glimmer)
  - Self-training model

Gene Modeling

Markov Chains – GeneMark self-training

- *Learning model parameters without knowing anything about the organism!*

- *Initial Setup*
  - Donor / acceptor = GT / AG
  - Initiation / termination = ATG / TGA, TAG, TAA
  - Exon – 5th order inhomogeneous model
    - derived from non-overlapping ORFs > 1000 bases long
    - even in eukaryotes very long ORFs are likely to be (single exon) genes
- *Predict introns and exons, then update the intron/exon models based on the 6mer frequencies of the predicted regions, weighted by the probability the prediction is correct*
- *Iterate*
- *Refinement of additional parameters*
  - In first iterations, update coding and non-coding models, but not lengths
  - Then train sites (splice & initiation/termination)
  - Then train lengths of coding/non-coding

Gene Modeling

Markov Chains – GeneMark

- *Unsupervised vs Supervised*

	A.thaliana Unsupervised		Supervised		*C.elegans* Unsupervised		Supervised		*D.melanogaster* Unsupervised		Supervised	
Nucleotide	**97.7** / **94.8**	**96.3**	97.2 / 94.3	95.8	**99.1** / 93.6	96.4	97.8 / **95.5**	**96.7**	97.9 / 92.9	95.4	**98.1** / **93.1**	**95.6**
Internal exons	**91.2** / 87.8	89.5	**91.2** / **88.5**	**89.9**	**94.0** / **91.3**	**92.7**	90.9 / 90.8	90.9	**91.3** / 89.7	**90.5**	87.2 / **90.2**	88.7
Initiation sites	**80.1** / **76.5**	**78.3**	**80.1** / 71.9	76.0	**85.8** / **68.9**	**77.4**	79.2 / 67.4	73.3	**83.9** / 73.5	**78.7**	83.4 / **74.3**	**78.9**
Termination sites	87.5 / **83.1**	**85.3**	**88.3** / 78.6	83.5	**95.1** / 75.3	85.2	94.0 / **79.6**	**86.8**	89.2 / 77.2	83.2	89.5 / **78.8**	**84.2**
Donor sites	**94.0** / **90.3**	**92.2**	**94.0** / 89.8	91.9	**96.2** / 90.8	**93.5**	93.7 / **91.4**	92.6	**92.8** / 87.2	**90.0**	91.3 / **89.1**	**90.2**
Acceptor sites	**94.0** / **90.2**	**92.1**	93.6 / 89.2	91.4	**97.3** / 91.6	**94.5**	95.2 / **92.8**	94.0	**93.0** / 87.0	**90.0**	90.5 / **87.9**	89.2

Boldface highlights the higher value in comparison of unsupervised and supervised modes (ES-3.0 versus E-3.0).

Lomsadze et al., Nucleic Acids Research 33, 6494-6506 (2005)