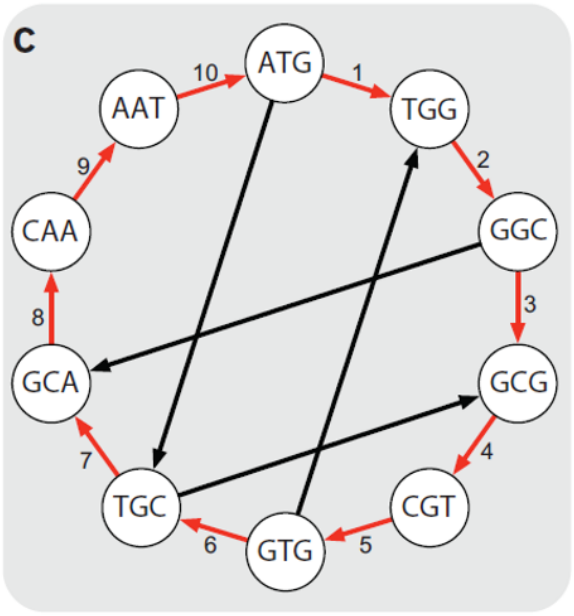
Introducción a la Bioinformática

Sequencing Methods

Sequence Assembly

Short Read Mapping



Hamiltonian cycle
Visit each vertex once

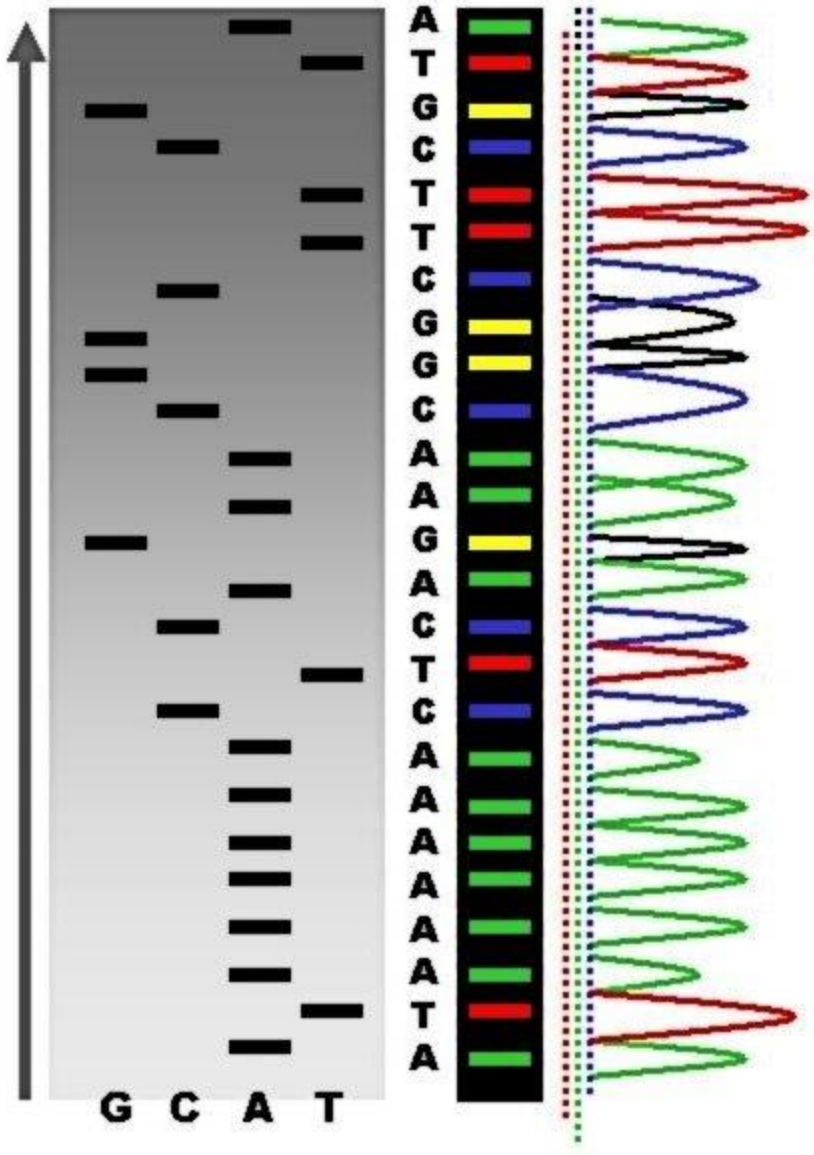
Fernán Agüero
Instituto de Investigaciones Biotecnológicas
Universidad Nacional de General San Martín
fernan@iib.unsam.edu.ar

Next-gen sequencing

Efficiency (bp/person/year)		
	1870	Miescher: Discovers DNA
	1940	Avery: Proposes DNA as 'Genetic Material' Watson & Crick: Double Helix Structure of DNA
1	1953	Holley: Sequences Yeast $tRNA^{Ala}$
15	1965	Wu: Sequences λ Cohesive End DNA
150	1970	Sanger: Dideoxy Chain Termination Gilbert: Chemical Degradation
1,500	1977	Messing: M13 Cloning
15,000	1980	Hood et al.: Partial Automation
25,000		
50,000	1986	
200,000	1990	• Cycle Sequencing • Improved Sequencing Enzymes • Improved Fluorescent Detection Schemes
50,000,000	2002	
100,000,000,000	2009	• Next Generation Sequencing •Improved enzymes and chemistry •New image processing

Sanger sequencing (old technology)

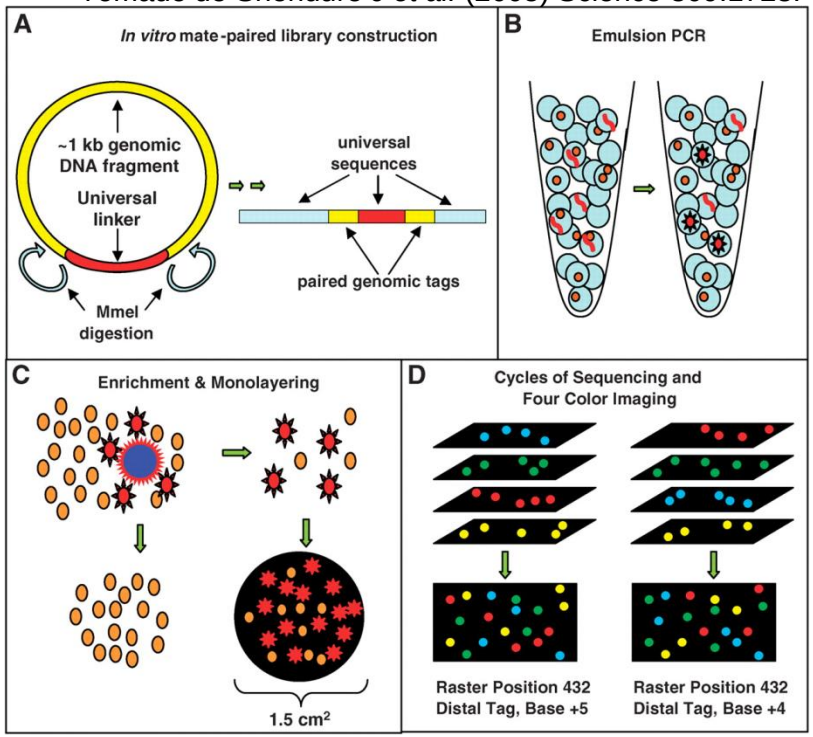
- Clonar el DNA.
- Generar una escalera de moléculas *etiquetadas* (con fluoróforos) o marcadas radioactivamente
- Cada fragment difiere en 1 nucleótido del proximo
- Separar la mezcla en alguna matriz (electroforesis).
- Detectar cada fragmento
- Interpretar los picos de emisión como una cadena de bases (DNA).
- Se generan cadenas de 500 a 1,000 bases de longitud
- 1 secuenciador genera ~ 57,000 nucleotidos/corrida
- Ensamblar las cadenas en un *todo*



New sequencing technologies

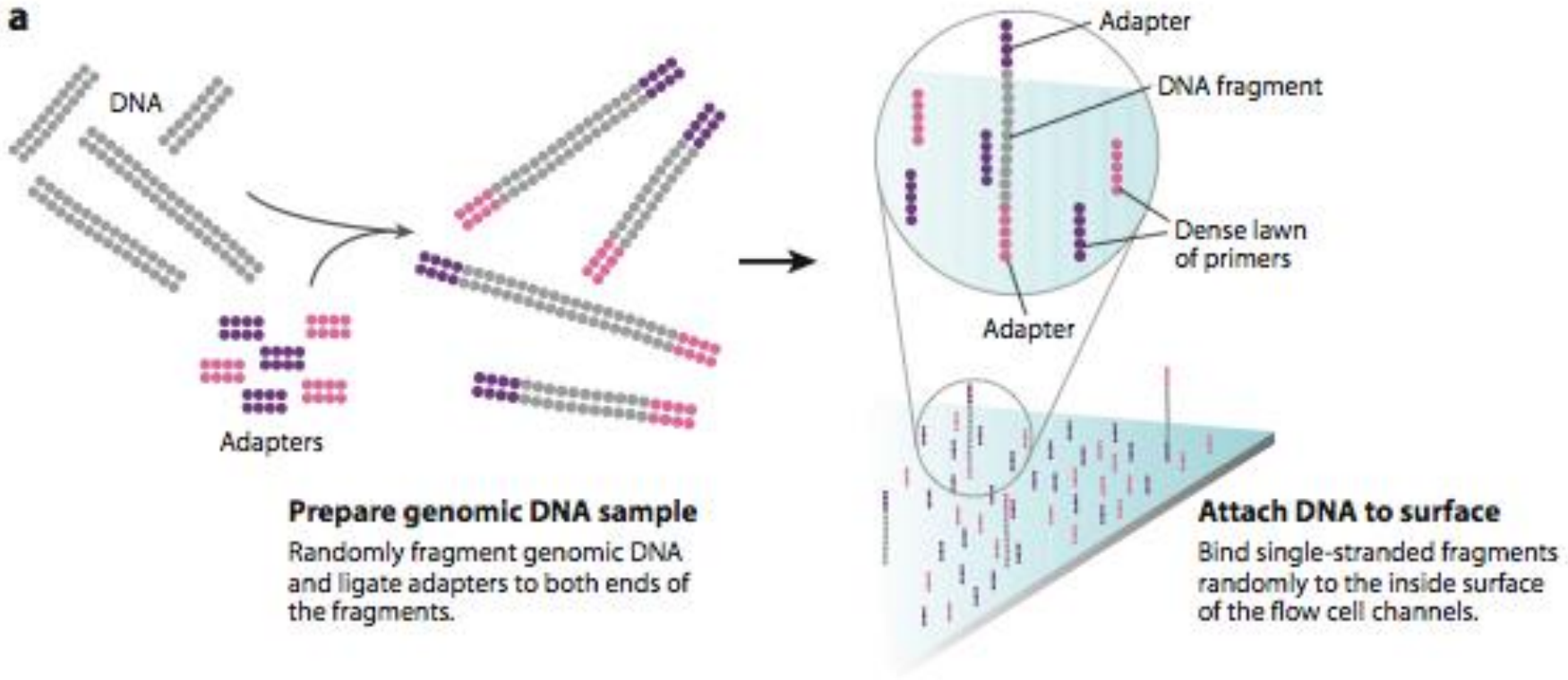
- **Breakthrough**
  - **Polymerase colonies – *polony / polonies***
    - In situ localized amplification and contact replication of many individual DNA molecules. Mitra RD, Church GM. (1999) Nucleic Acids Res. 27: e34.
  - **Se elimina la necesidad de clonar moléculas en *E. coli***
  - **Multiplex-amplification, manteniendo agrupamiento físico de amplicones idénticos**
    - Se *amplifican* y *clonan* moléculas en el tubo
    - Emulsion-PCR (beads)
    - *In situ* polonies (matrix)

Tomado de Shendure J *et al.* (2005) Science 309:1728.



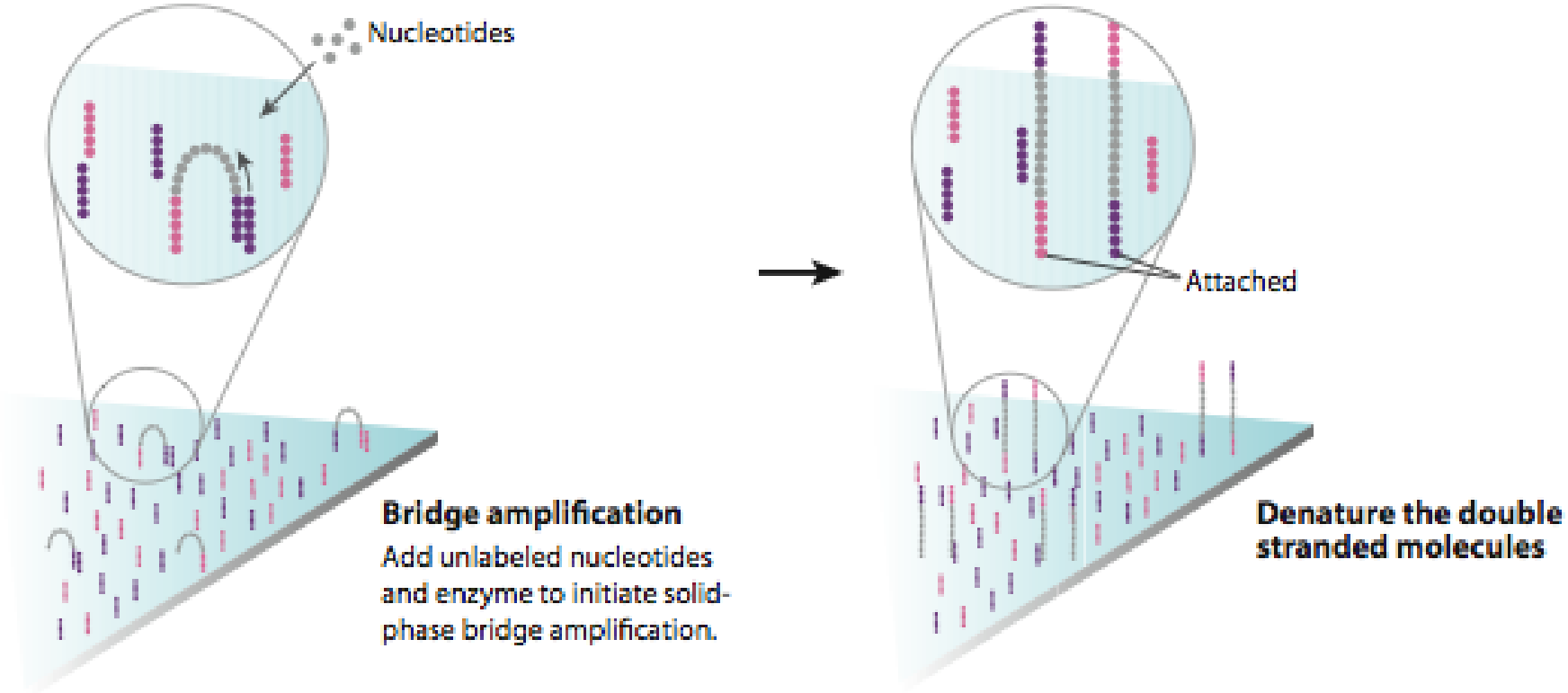
New sequencing technologies: Illumina / solexa

- Construcción de bibliotecas
- Attachment al soporte



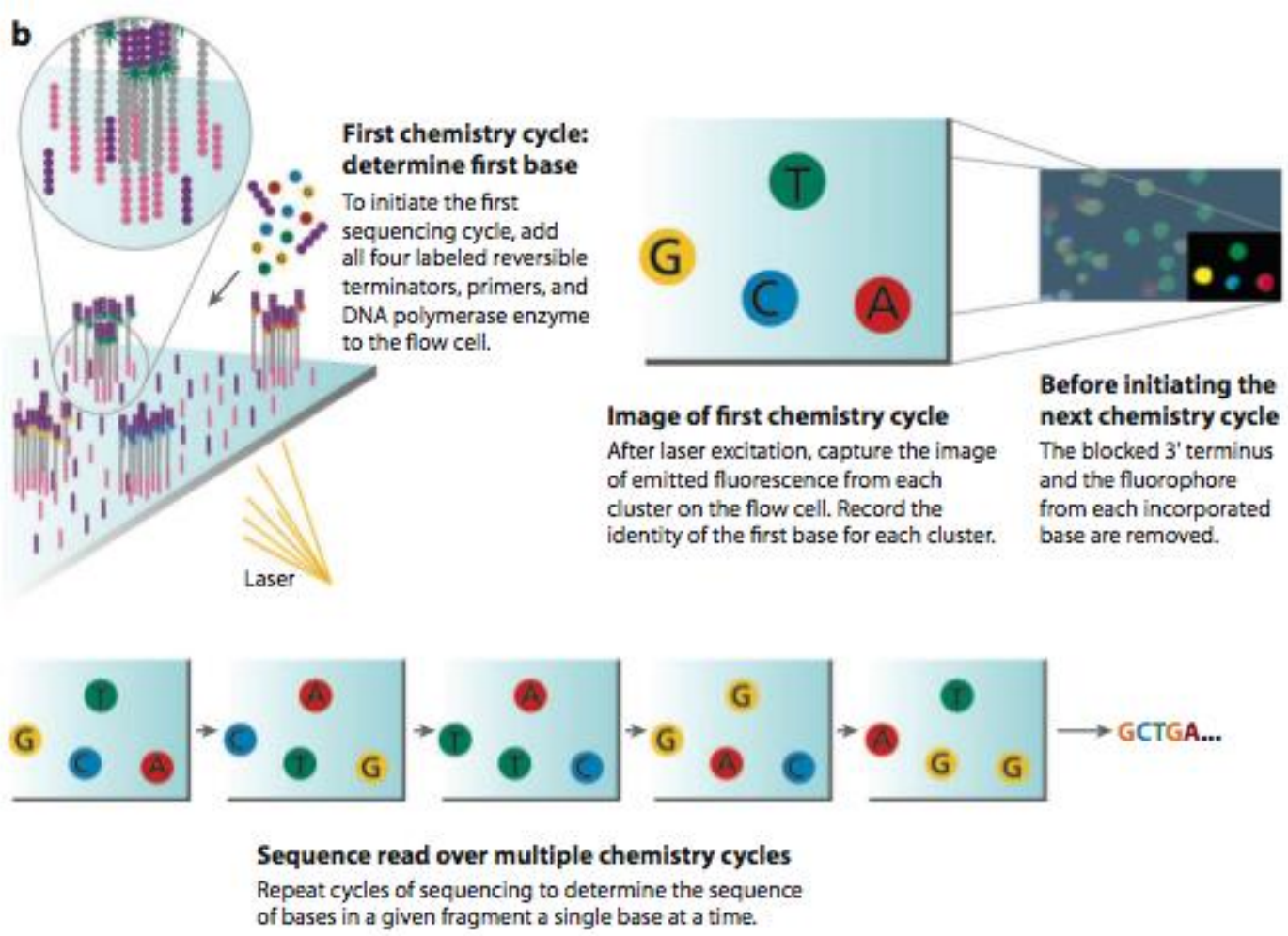
New sequencing technologies: Illumina / solexa

- Amplificación de las colonias



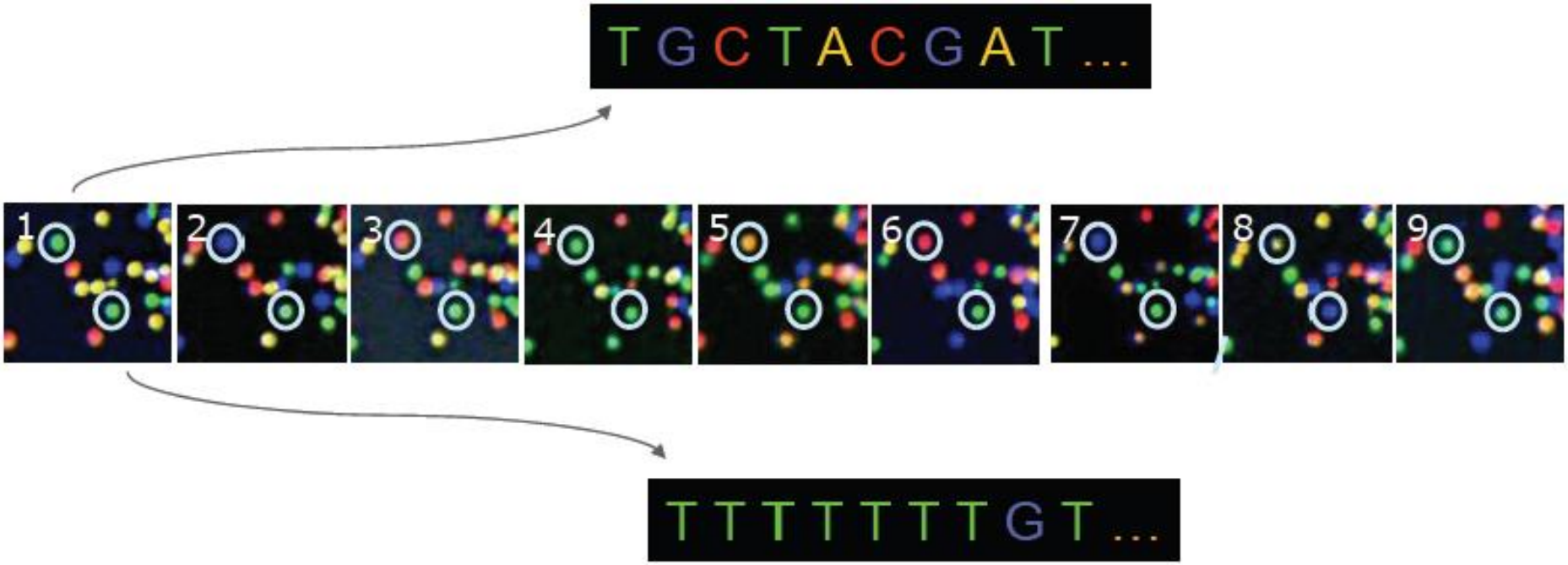
New sequencing technologies: Illumina / solexa

- Reacciones de extensión + lectura del slide usando laser



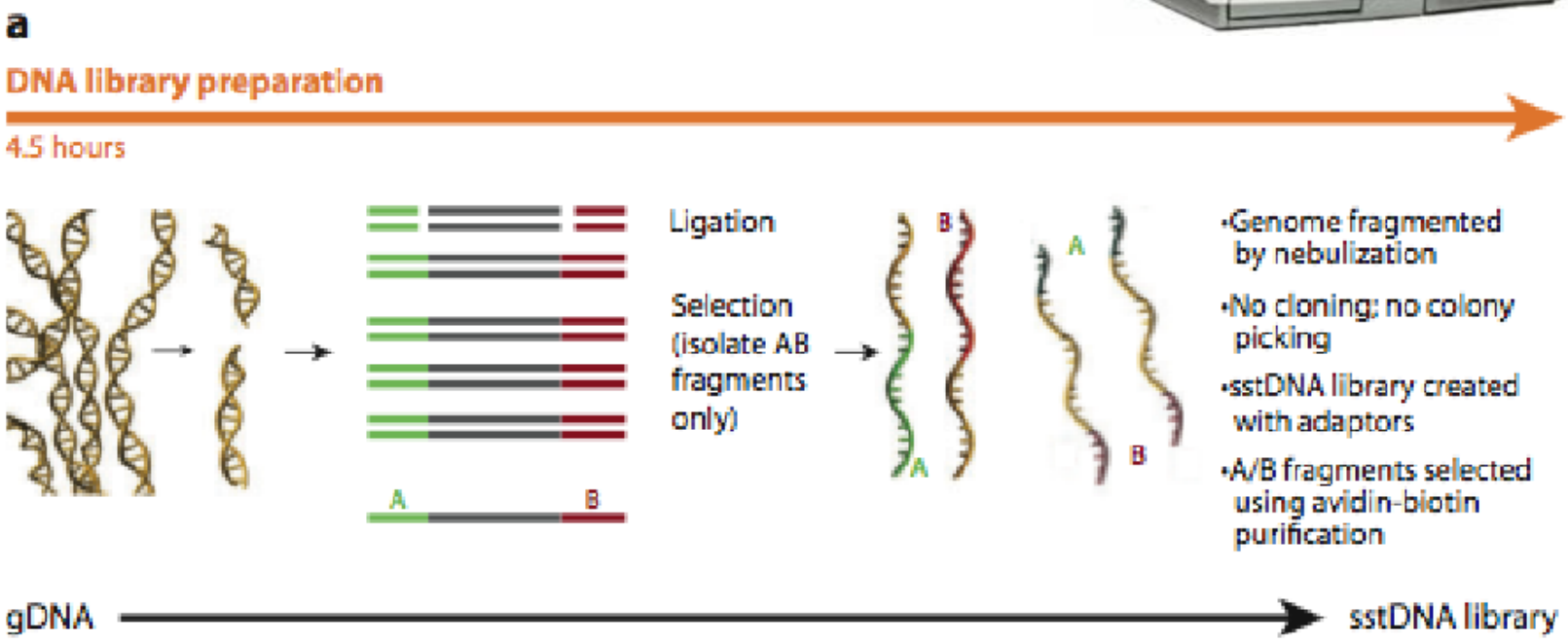
New sequencing technologies: Illumina / solexa

- Base calling – Asignación de bases en la secuencia



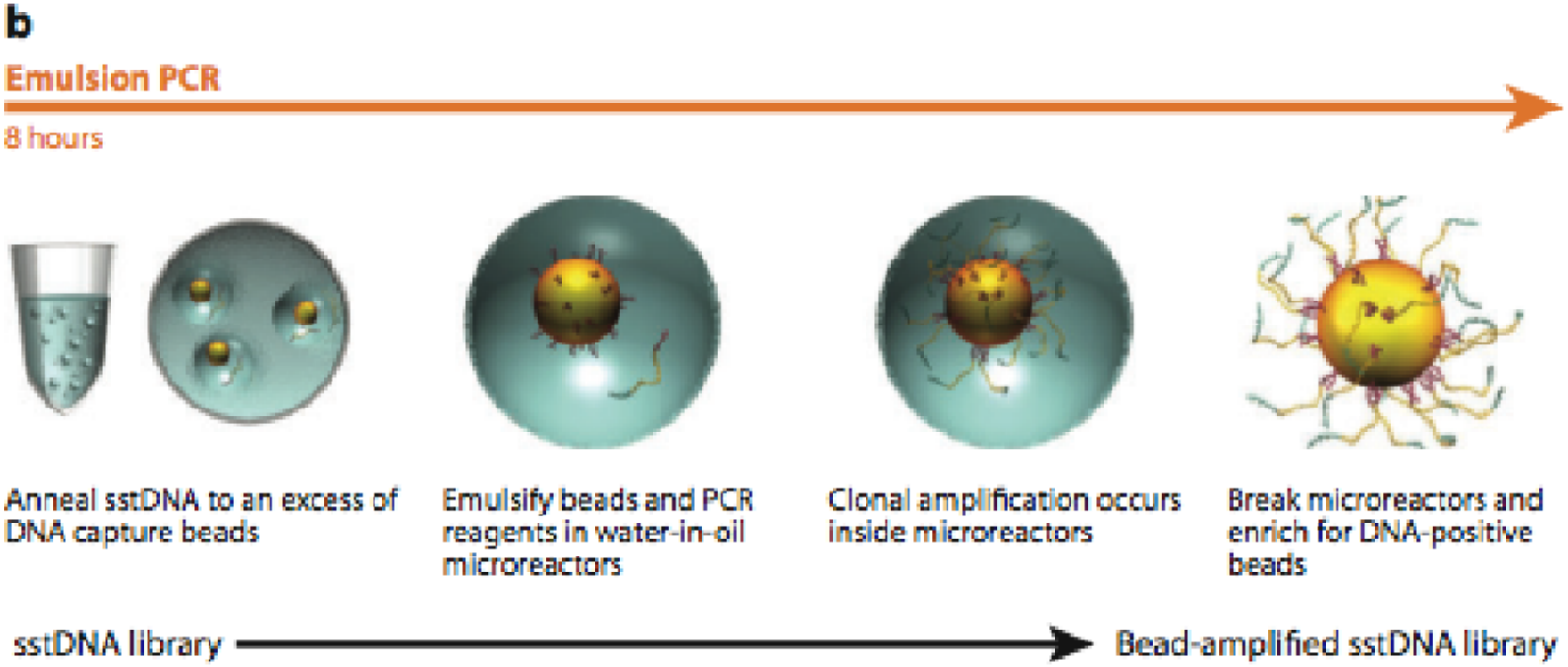
New sequencing technologies: 454

- **Roche 454**
  - **Also known as *pyrosequencing***
  - **500 million bp/run**
  - **10 hr/run**
  - **400-500 bp/read & > 1 M reads**



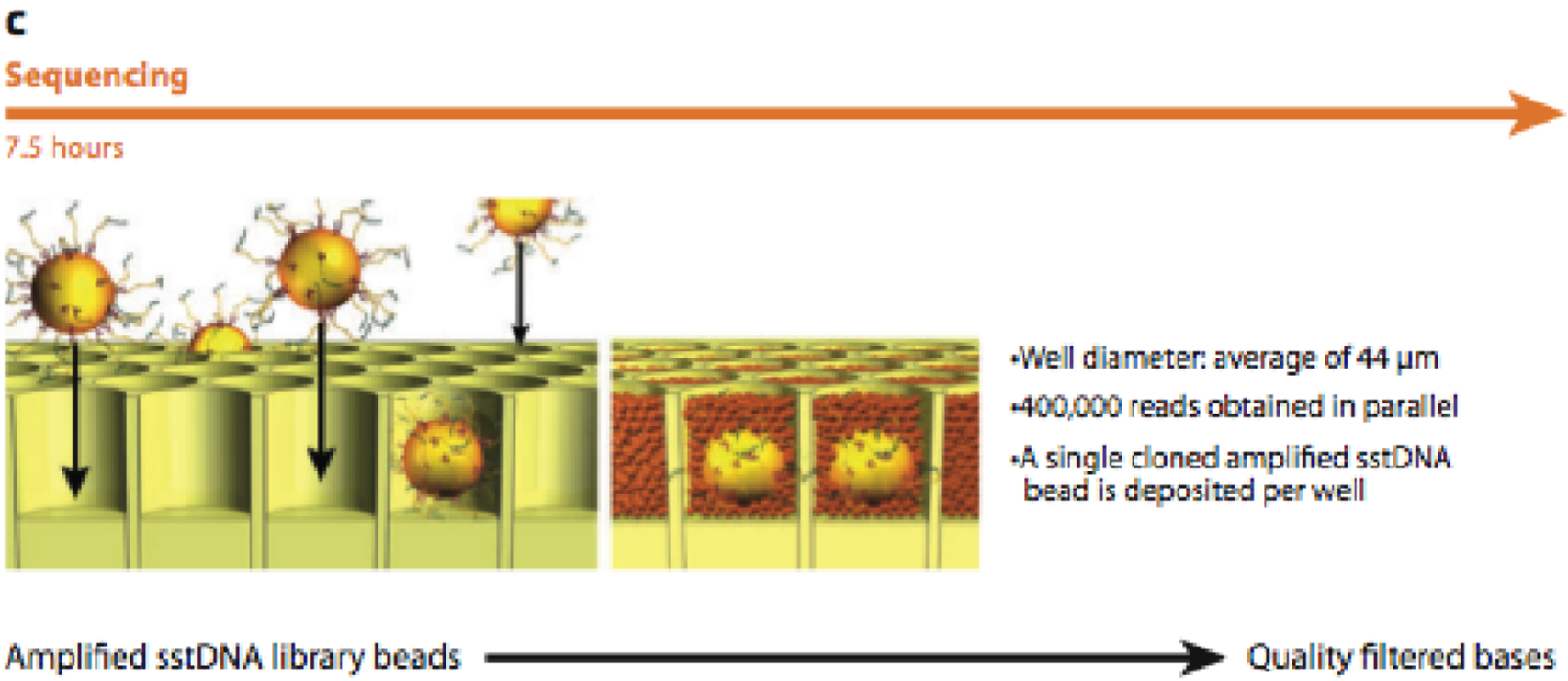
New sequencing technologies: 454

- PCR en emulsión



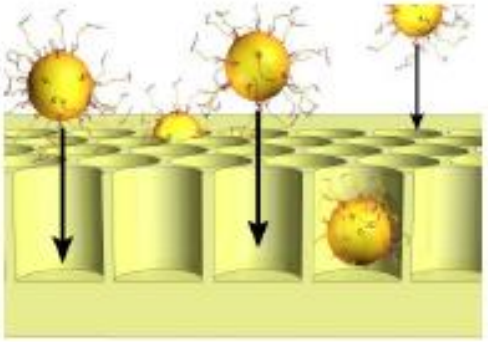
New sequencing technologies: 454

- Secuenciación en nanowells



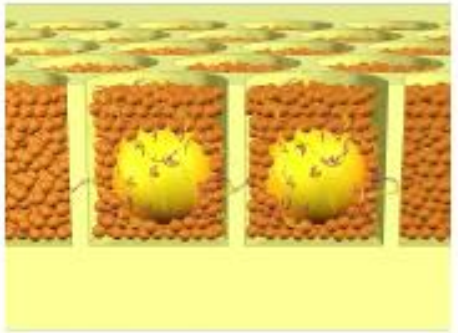
454 sequencing explained

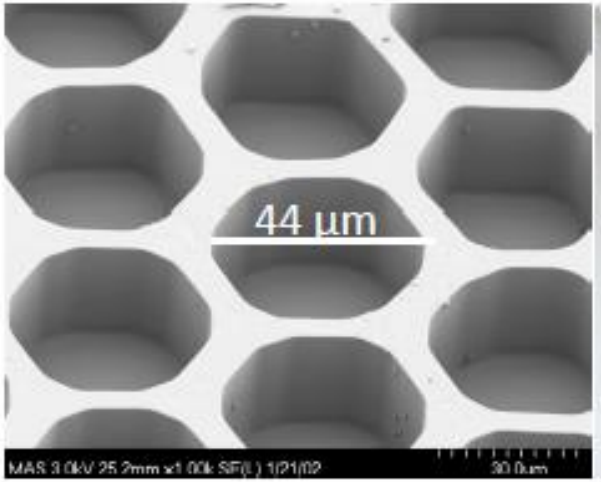
Load beads into PicoTiter™Plate

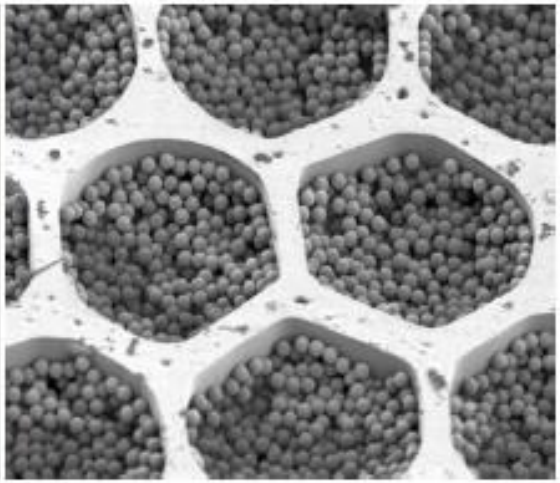


Load Enzyme Beads

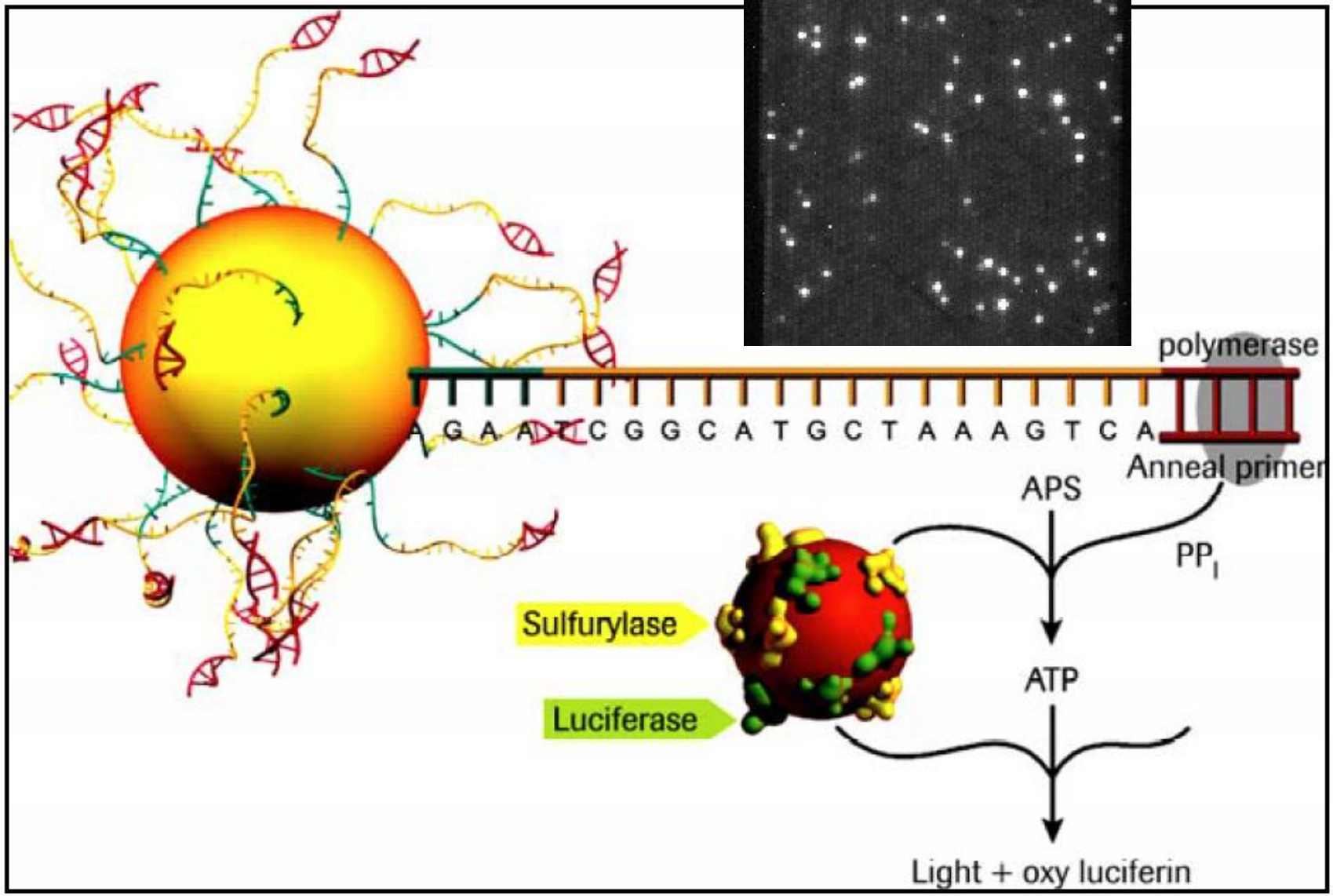
Centrifuge Step







454 sequencing explained

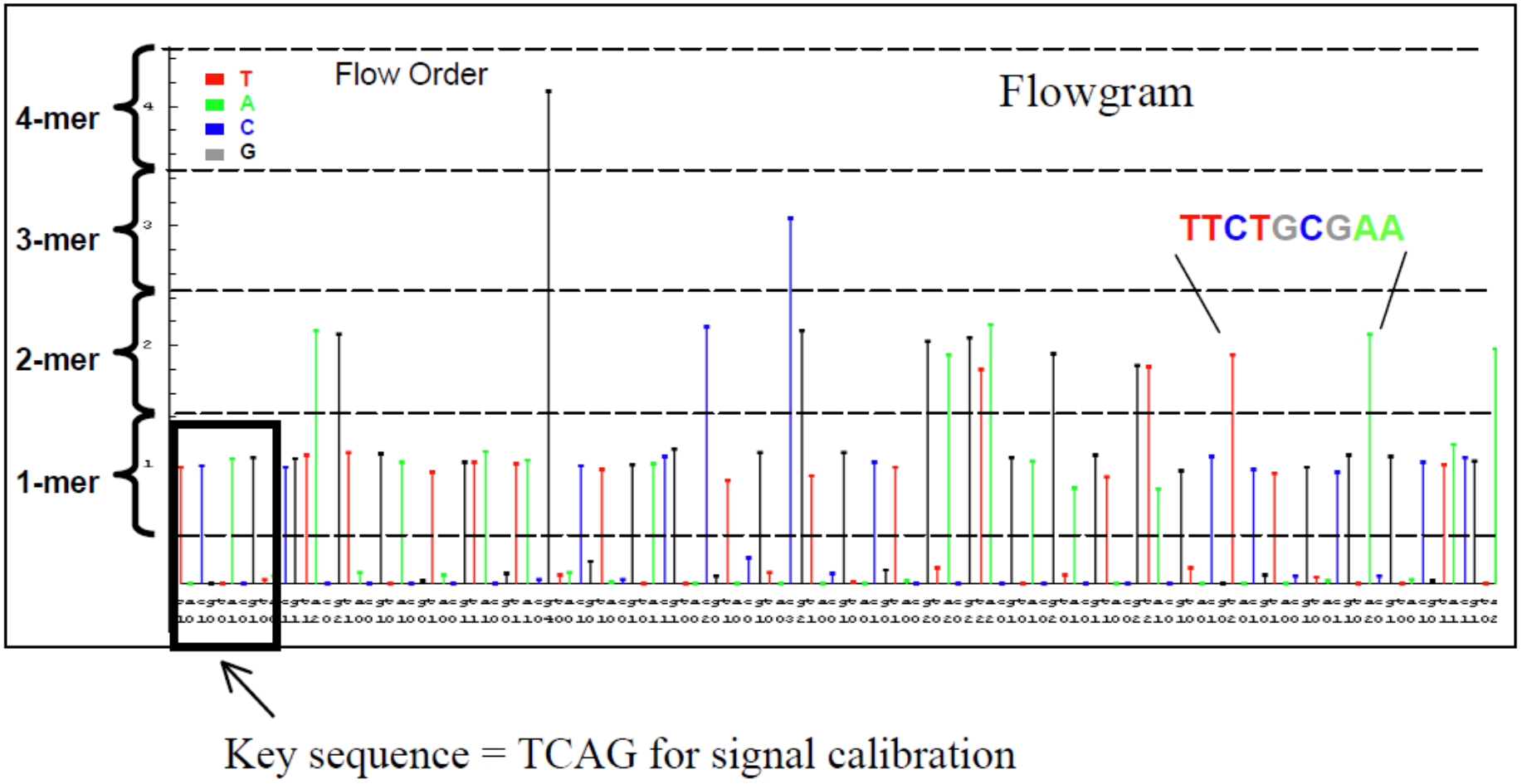


454 sequencing explained

- Cada base se inyecta en forma secuencial en la platina de reacción (PicoTiter Plate), de a una por vez
  - Por ejemplo, 100 veces para un secuenciador 454-FLX
- Si el nucleótido es complementario al molde, se polimeriza en la cadena naciente. La reacción genera pirofosfato, que es transformado en una señal luminosa
- La señal es leida por una cámara
- La intensidad de la señal es proporcional al número de nucleótidos incorporados
  - Si hay 3 'T' en el molde, la luz emitida va a ser ~ 3 veces la de una sola 'T'
- La secuencia se lee a partir de un '*flowgram*'

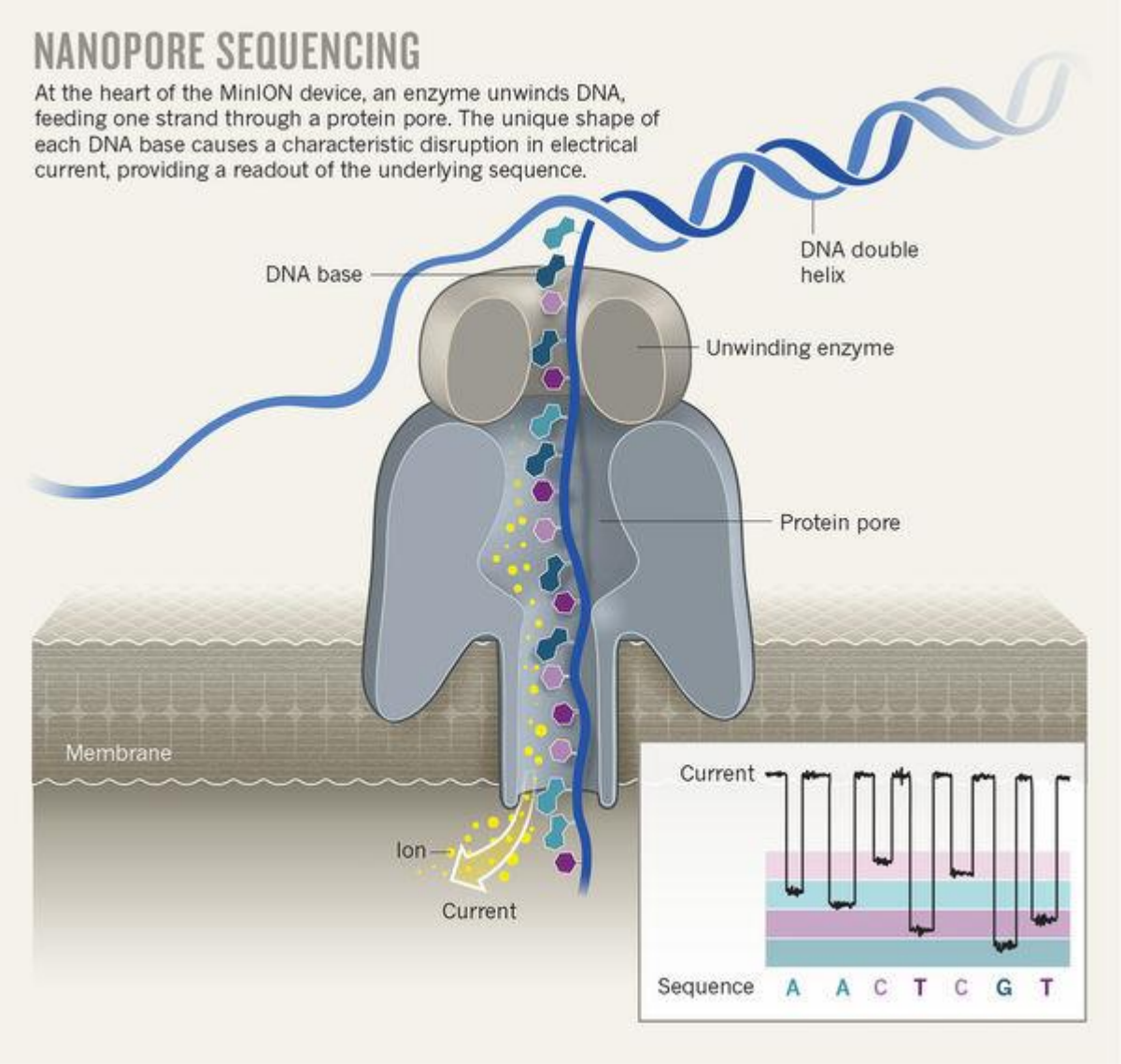
Margulies M, et al (2005) Genome sequencing in microfabricated high-density picolitre reactors. Nature DOI: 10.1038/nature03959

A 454 flowgram

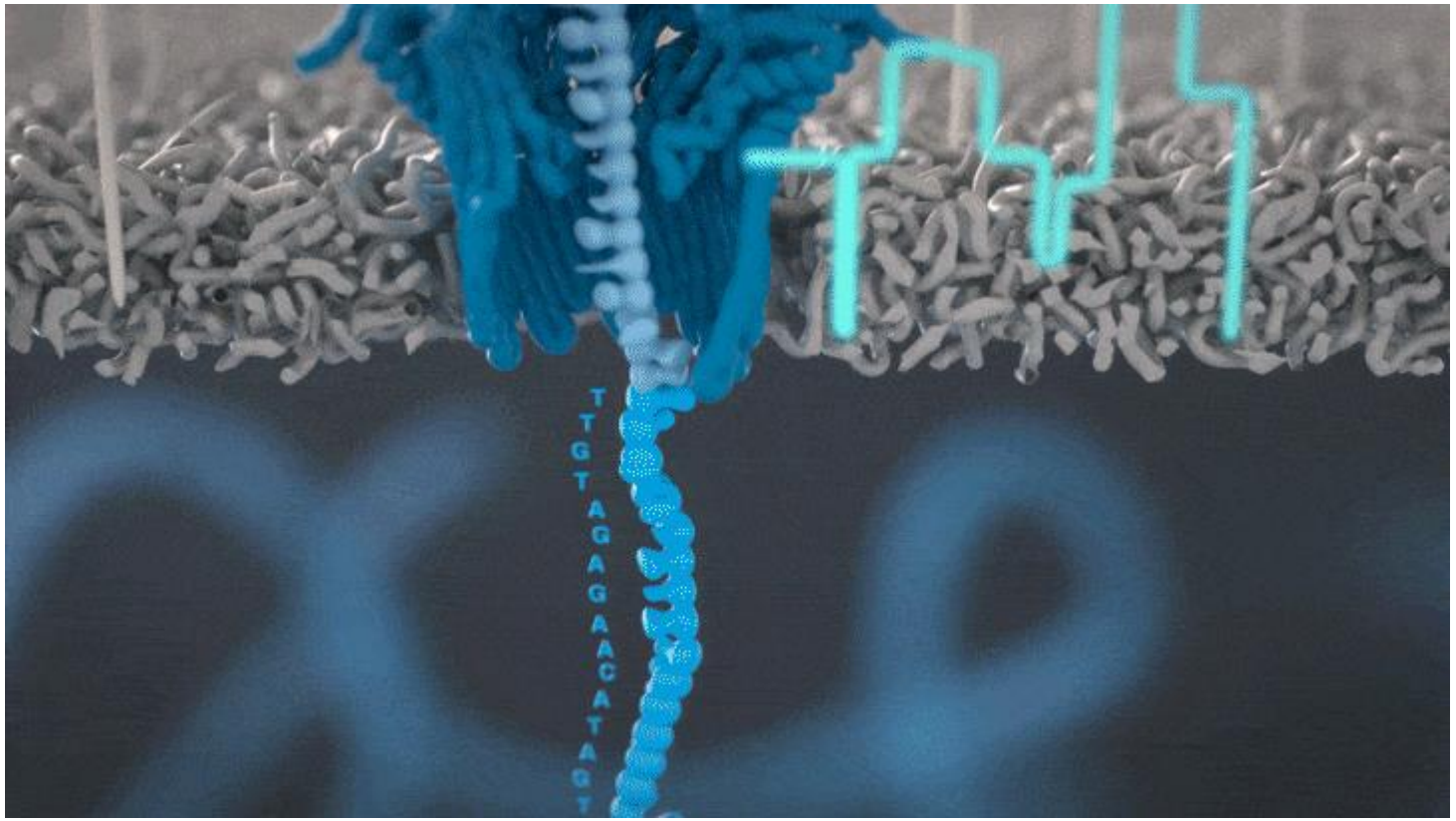


Oxford Nanopore

A **nanopore** is a nano-scale hole. In its devices, **Oxford Nanopore** passes an ionic current through **nanopores** and measures the changes in current as biological molecules pass through the **nanopore** or near it. The information about the change in current can be used to identify that molecule.



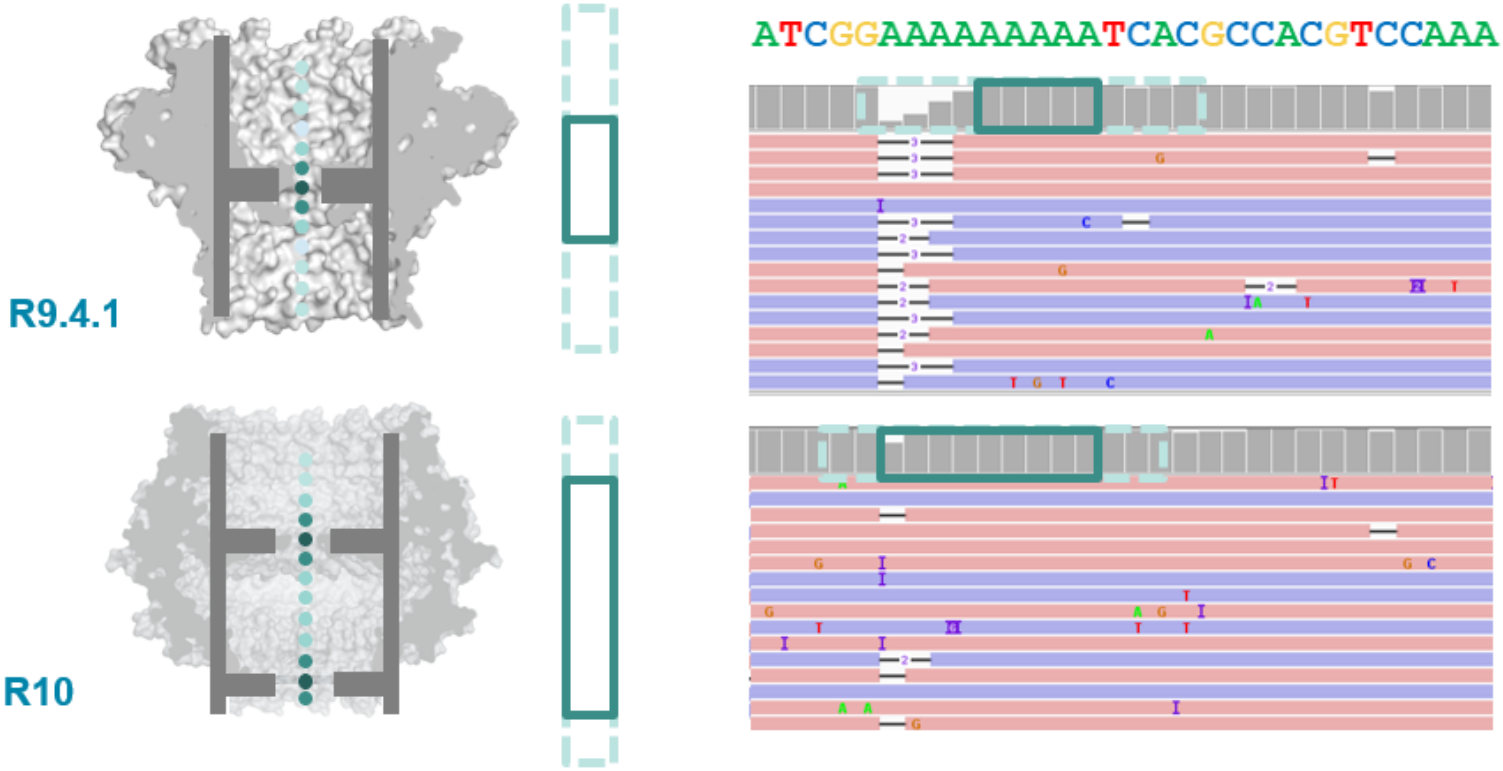
Nanopore: capturing the signal



When sequencing DNA or RNA with nanopores, the ***changes in current*** caused by the strand of DNA or RNA as it passes through the pore **are recorded**. The processive movement of bases through the pore leads to a ***continual change in current, known as the "squiggle"***. MinKNOW software processes the squiggle into reads in real-time, each read corresponding to a single strand of DNA/RNA. These reads are written out into POD5 files. This raw data contains information on not only canonical bases but also base modifications, such as methylation.

Oxford Nanopore

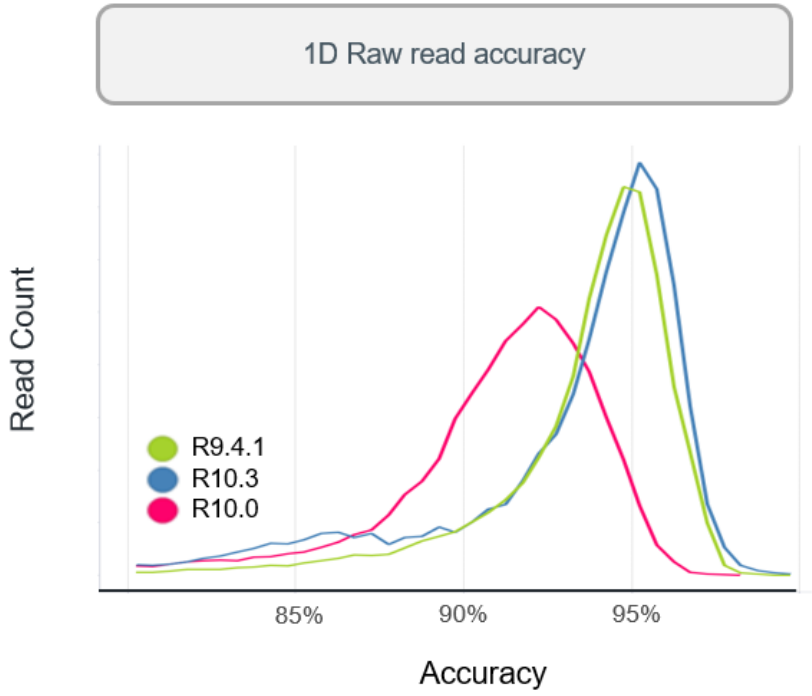
La evolución de la química de secuenciación incluye el uso de distintos tipos de nanoporos que van mejorando precisión de lectura, velocidad, throughput, etc.

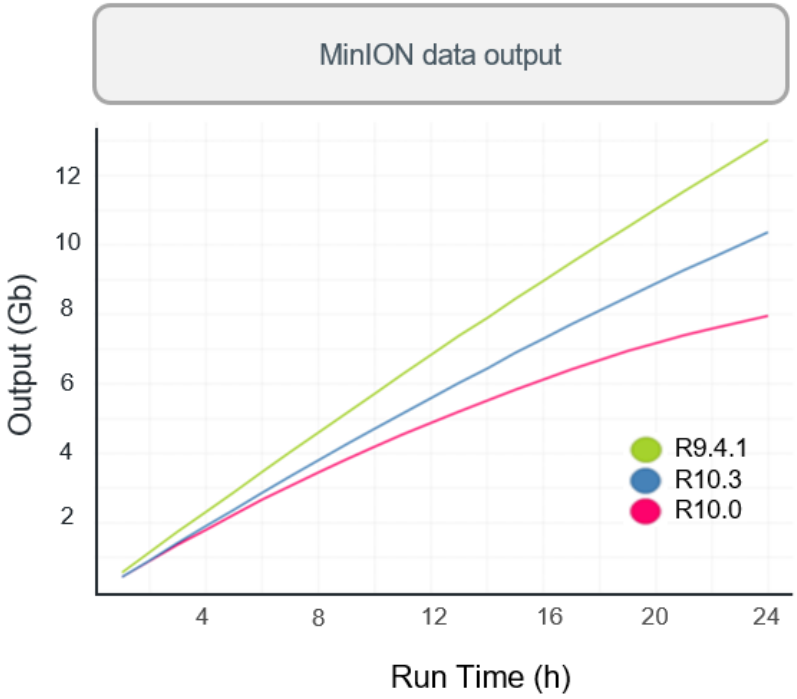


R10 = Marzo, 2019,
https://nanoporetech.com/about-us/news/r103-newest-nanopore-high-accuracy-nanopore-sequencing-now-available-store

Oxford Nanopore

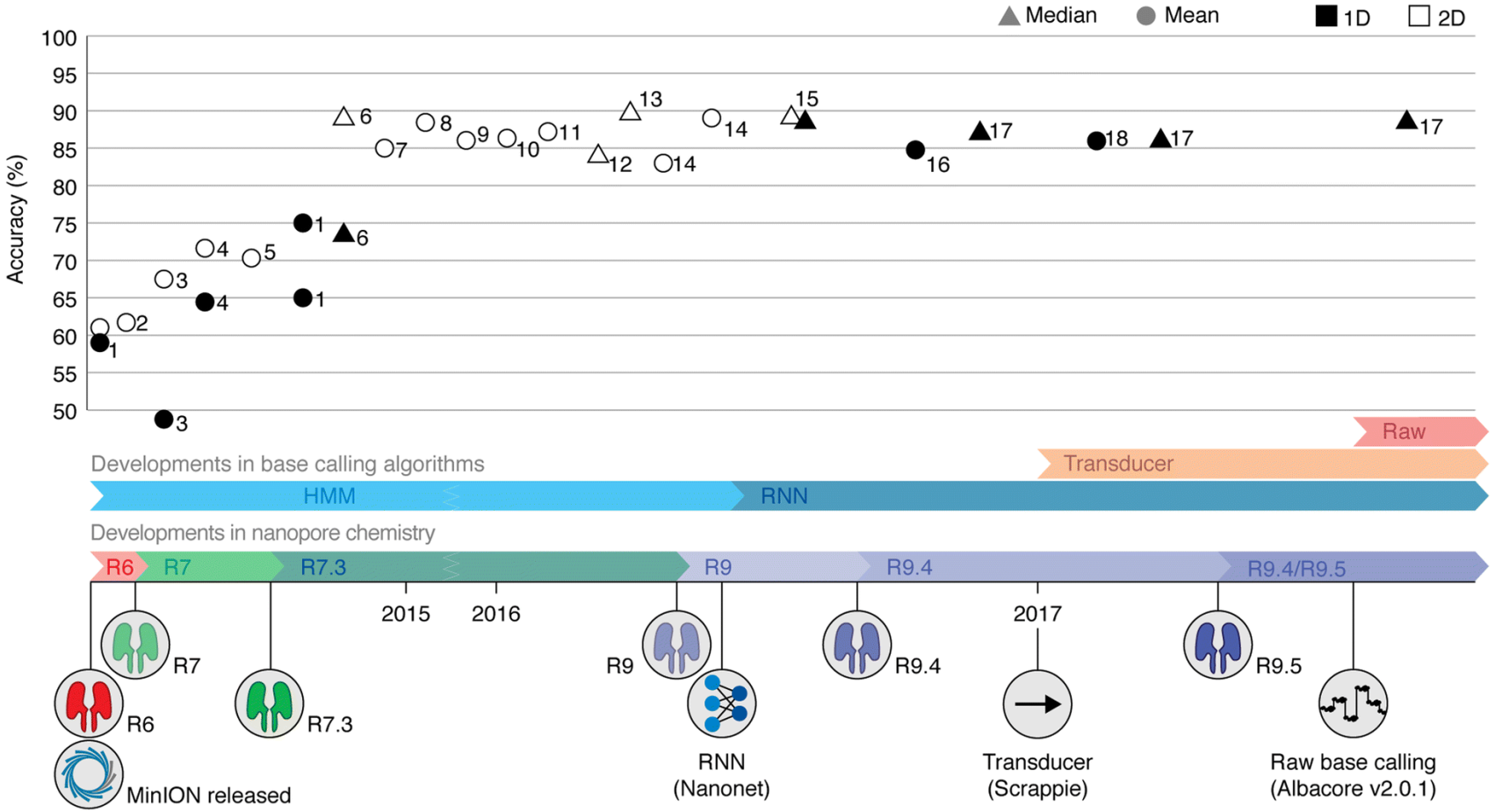
La evolución de la química de secuenciación incluye el uso de distintos tipos de nanoporos que van mejorando precisión de lectura, velocidad, throughput, etc.





Input Requirements:	R9.4.1	R10.3	R10.0
	5 – 50 fmol	25 – 75 fmol	50 – 100 fmol

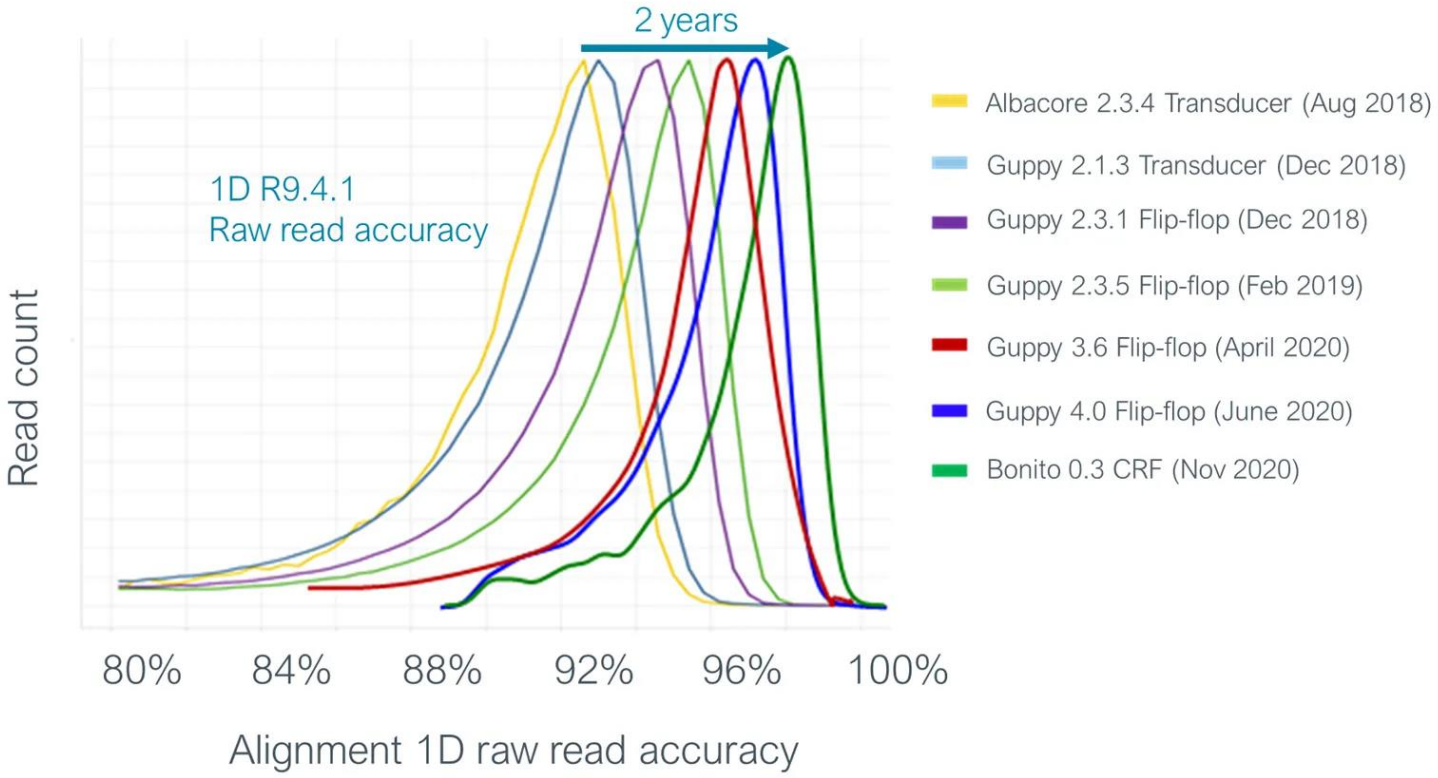
Nanopore sequencing read accuracy



Timeline of reported MinION read accuracies and Oxford Nanopore Technologies (ONT) technological developments. Nanopore chemistry updates and advances in base-caller software are represented as colored bars. The plotted accuracies are ordered on the basis of the chemistry and base-calling software used, not according to publication date. Based on data from 1 [9]; 2 [10]; 3 [50]; 4 [51]; 5 [33]; 6 [28]; 7 [52]; 8 [53]; 9 [54]; 10 [29]; 11 [31]; 12 [48]; 13 [46]; 14 [55]; 15 [11]; 16 [5]; 17 [13]; 18 [3]. HMM Hidden Markov Model, RNN Recurrent Neural Network

Rang, FJ, Kloosterman, WP & de Ridder, J. *Genome Biol* **19**, 90 (2018). https://doi.org/10.1186/s13059-018-1462-9

Evolution of nanopore sequencing read accuracy

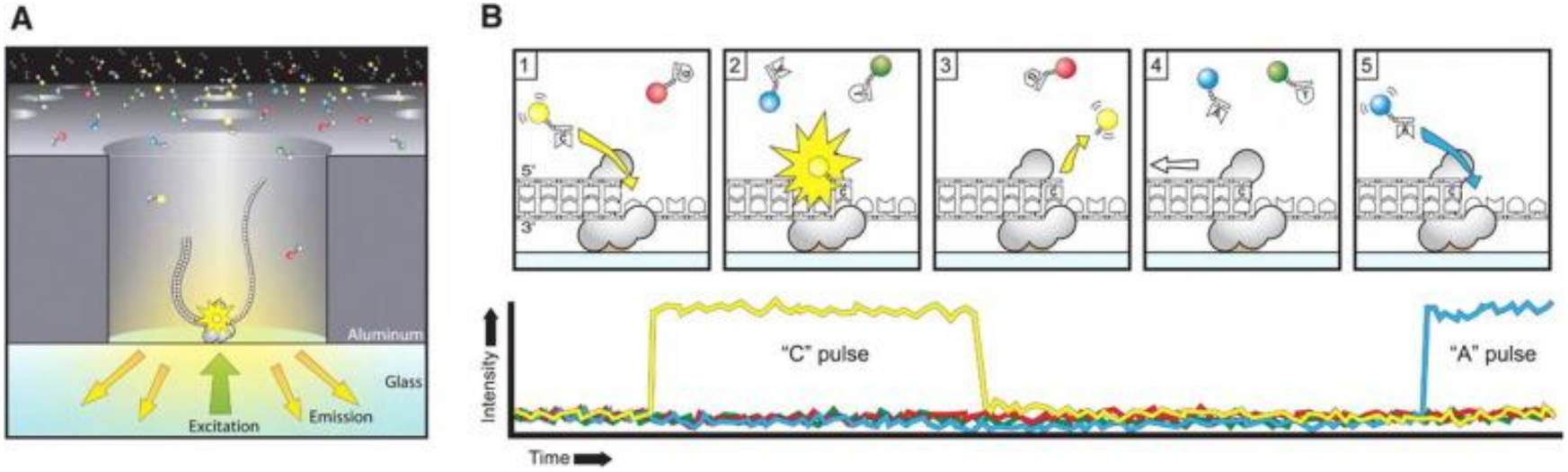


Pacific Biosciences (PacBio)

- The zero-mode waveguide (ZMW) is a nanophotonic confinement structure
- ZMW holes are ~70 nm in diameter and ~100 nm in depth.
- Due to the behavior of light when it travels through a small aperture, the optical field decays exponentially inside the chamber.
- The volume in a ZMW is ~20 zeptoliters (20 X 10^{-21} liters)
- Within this volume, the activity of DNA polymerase incorporating a single nucleotide can be readily detected

Circular DNA

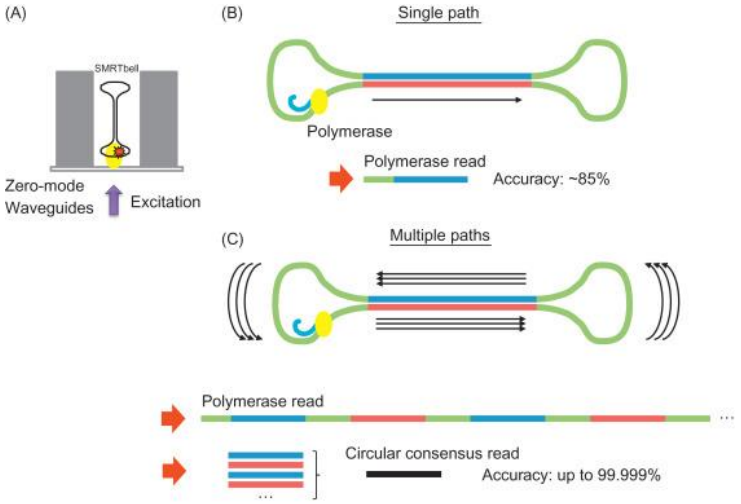
Zero mode waveguide unit (ZMW)

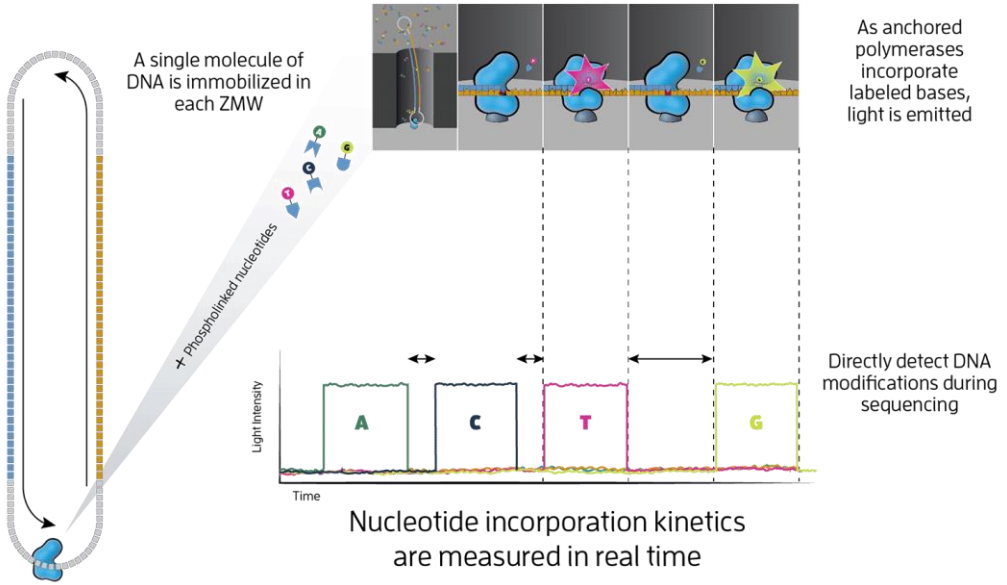


PacBio circular sequencing

No hay PCR o amplificación

Single-Molecule Sequencing

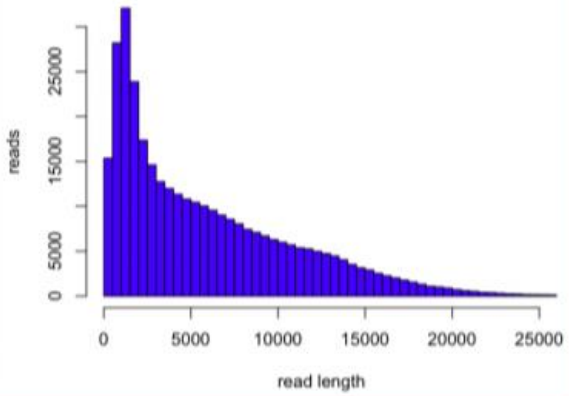




https://youtu.be/_lD8JyAbwEo

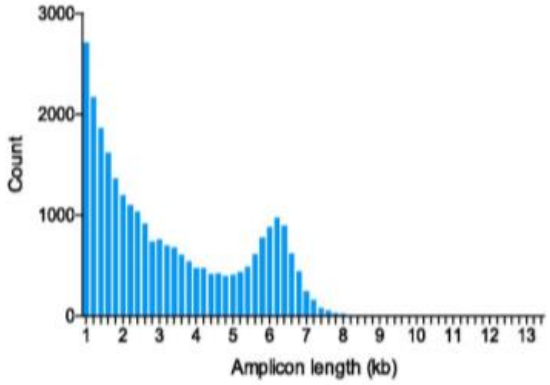
Single Molecule Sequencing Technologies

PacBio RS II

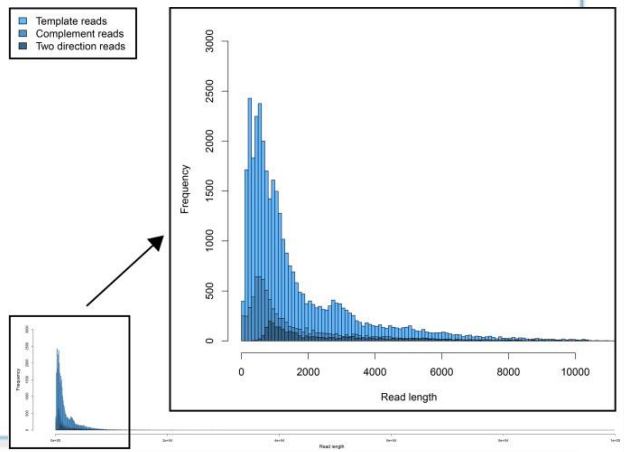


Moleculo

illumina
moleculo



Oxford Nanopore



Historia de la Genómica: Base quality scores

El uso de valores de calidad para la asignación de bases a partir de picos en un cromatograma comenzó con el paquete phred/phrap/consed. www.phrap.org

Phred/Phrap/Consed es un paquete de software utilizado para:

- Leer cromatogramas (trace files)
- Asignar valores de calidad a las bases individuales de una secuencia
- Identificar y enmascarar secuencias correspondientes a vector (plásmido) o secuencias repetitivas
- Ensamblar secuencias individuales en contigs
- Visualizar assemblies (contigs)
- Hacer 'sequence finishing' auto dirigido (automatic finishing)

Phred: a basecaller

- Genome Res 8 (1998): 175
- Genome Res 8 (1998): 186

RESEARCH

Base-Calling of Automated Sequencer Traces Using *Phred.* I. Accuracy Assessment

Brent Ewing,[1] LaDeana Hillier,[2] Michael C. Wendl,[2] and Phil Green[1,3]

[1]Department of Molecular Biotechnology, University of Washington, Seattle, Washington 98195-7730 USA;
63108 USA

RESEARCH

Base-Calling of Automated Sequencer Traces Using *Phred.* II. Error Probabilities

Brent Ewing and Phil Green[1]

Department of Molecular Biotechnology, University of Washington, Seattle, Washington 98195-7730 USA

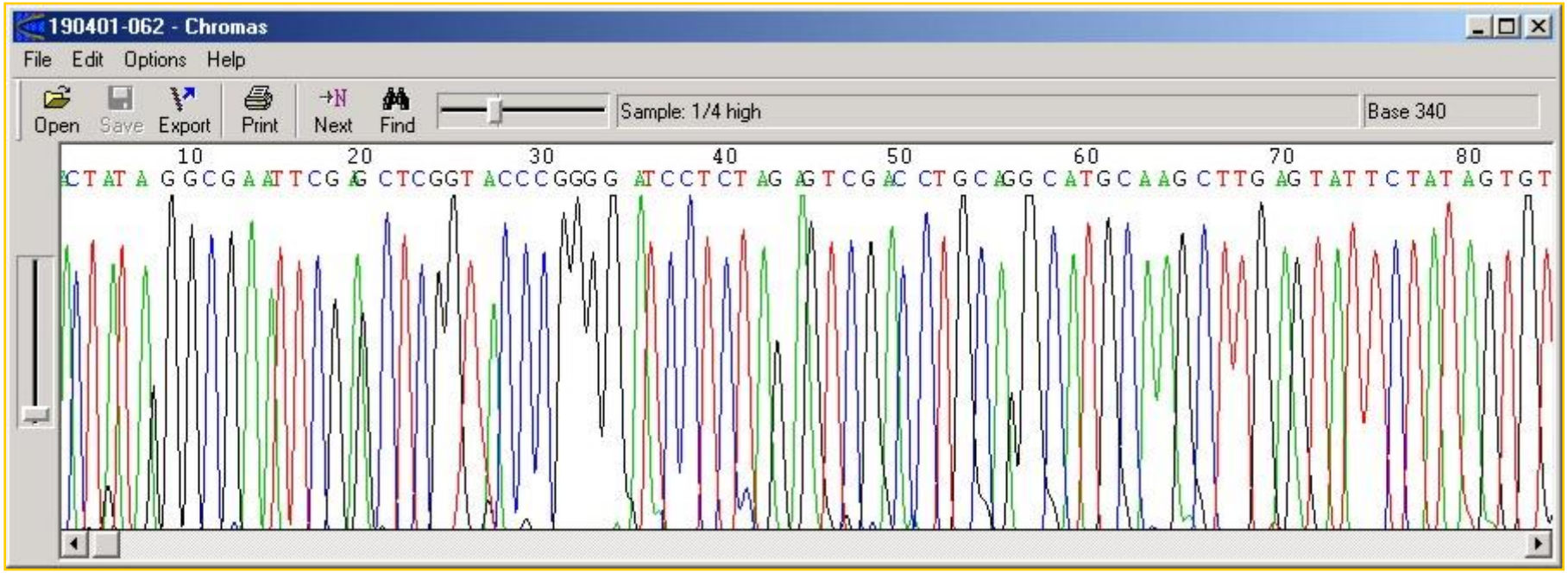
8:186–194 ©1998 by Cold Spring Harbor Laboratory Press ISSN 1054-9803/98 $5.00; www.genome.org

Phred

- **Phred is a program that performs several tasks:**
  - **Reads trace files – compatible with most file formats: SCF (standard chromatogram format), ABI (373/377/3700), ESD (MegaBACE) and LI-COR.**
  - **Calls bases – attributes a base for each identified peak with a lower error rate than the standard base calling programs.**
  - **Assigns quality values to the bases – a "Phred value" based on an error rate estimation calculated for each individual base.**
  - **Creates output files – base calls and quality values are written to output files.**

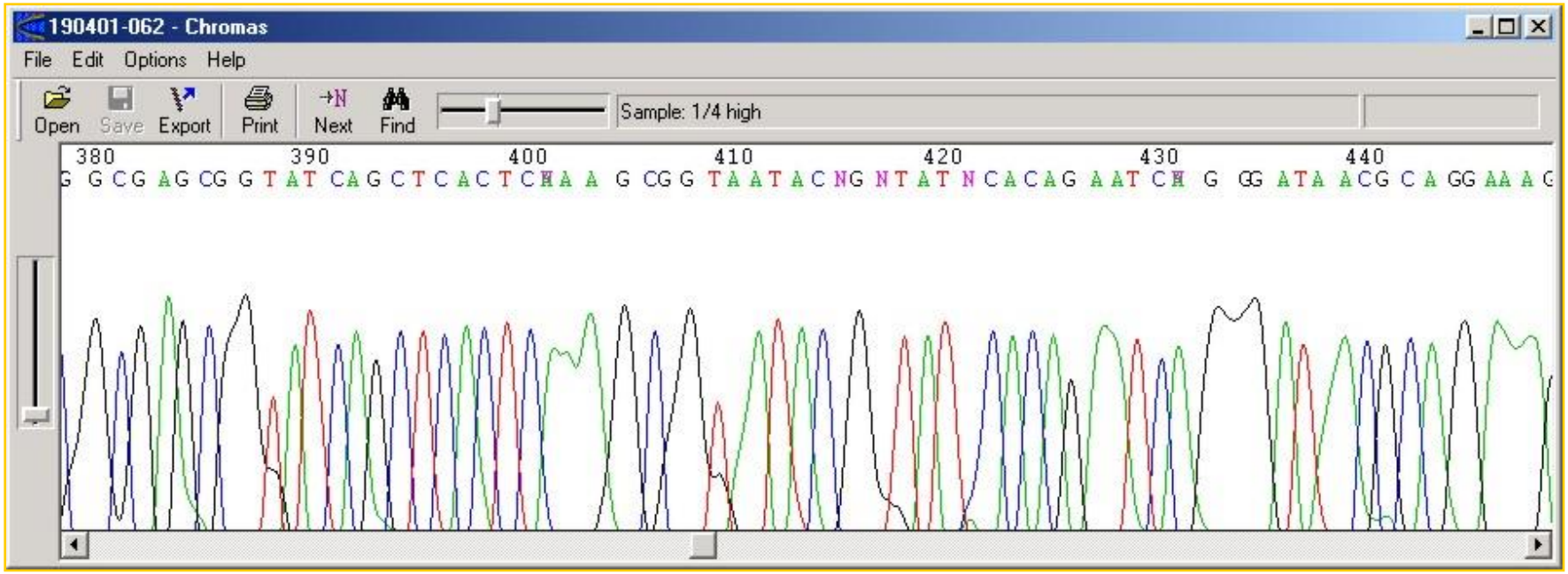
Trace files

- Alta calidad, sin ambigüedad



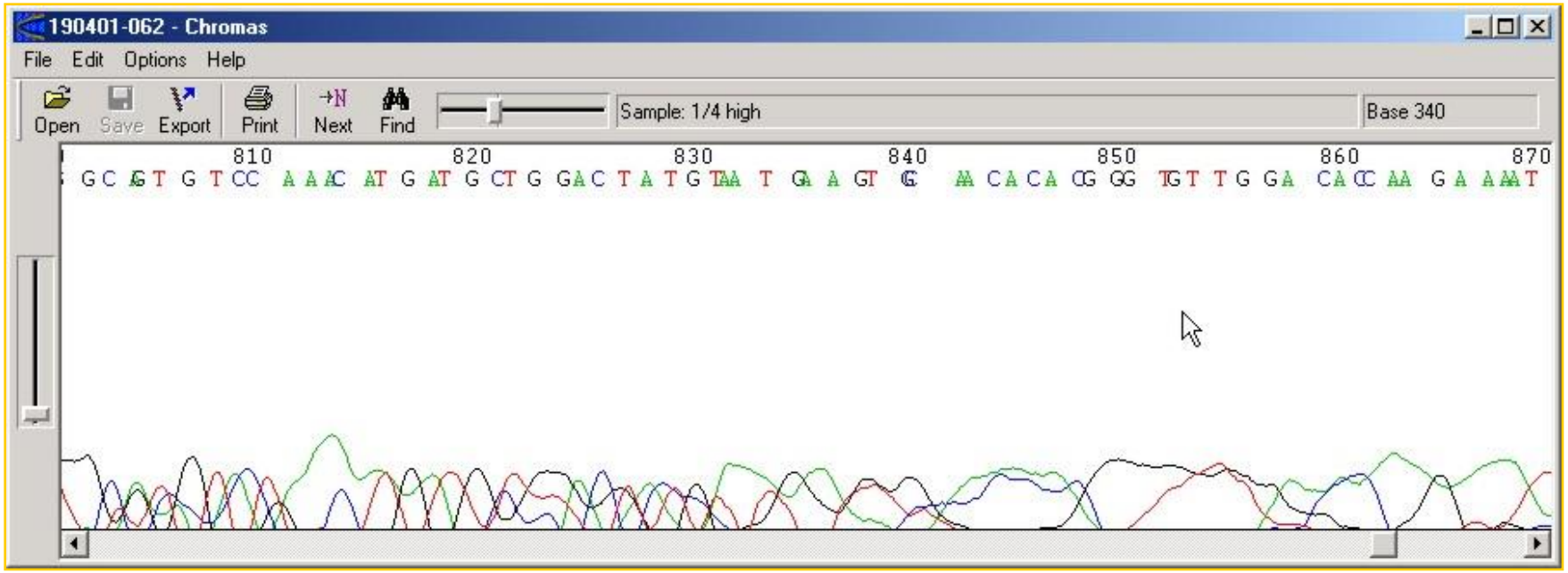
Trace files

- Calidad media, algunas ambigüedades



Trace files

- **Baja calidad**
  - **la confianza en la asignación de bases es menor**



Phred qualities

$$q = -10 \times log_{10}(p)$$

Donde:

- **q** = quality value
- **p** = estimated probability error for a base call

Phred quality scores are logarithmically linked to error probabilities

Phred Quality Score	Probability of incorrect base call	Base call accuracy
10	1 in 10	90 %
20	1 in 100	99 %
30	1 in 1000	99.9 %
40	1 in 10000	99.99 %
50	1 in 100000	99.999 %

http://en.wikipedia.org/wiki/Phred_quality_score

$$p = 10^{\frac{-q}{10}}$$

Donde:

- **q** = quality value
- **p** = estimated probability error for a base call

Q = quality value	P = estimated probability of error
0	1
1	0.794
2	0.631
3	0.501
4	0.398
…	…
10	0.1
20	0.01
30	0.001

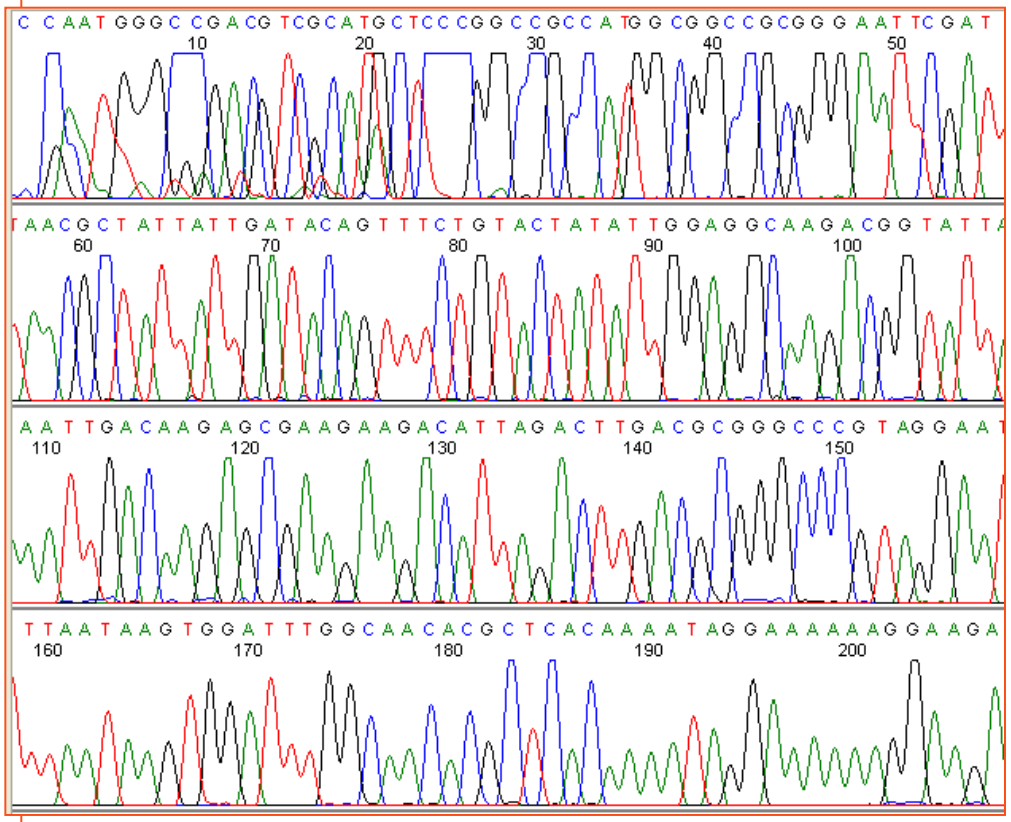
Phred: PHD files

```
BEGIN_SEQUENCE 01EBV10201A02.g

BEGIN_COMMENT

CHROMAT_FILE: EBV10201A02.g
ABI_THUMBPRINT:
PHRED_VERSION: 0.990722.g
CALL_METHOD: phred
QUALITY_LEVELS:99
TIME: Thu May 24 00:18:58 2001
TRACE_ARRAY_MIN_INDEX: 0
TRACE_ARRAY_MAX_INDEX: 12153
TRIM:
CHEM: term
DYE: big

END_COMMENT

BEGIN_DNA
t 8 5
c 13 17
a 19 26
c 19 32
t 24 2221
a 24 2232
a 22 2245
a 27 2261
g 25 2272
c 19 2286
c 12 2302
t 19 2314
g 12 2324
g 15 2331
g 19 2346
g 23 2363
t 33 2378
g 36 2390
c 44 2404
c 44 2419
t 39 2433
a 39 2446
a 34 2460
t 35 2470
g 34 2482
t 16 8191
g 19 8200
t 13 8211
c 13 8229
g 4 8241
n 4 8253
c 4 8263
t 10 8276
t 9 8286
c 12 8301
t 16 8313
c 12 8329
c 12 8336
c 15 8343
t 19 8356
c 9 8371
g 13 8386
g 14 8397
a 7 8417
g 9 8427
g 4 8445
t 6 11908
a 6 11921
g 6 11927
t 6 11947
c 6 11953
a 6 11964
g 6 11981
c 4 11994
n 4 12015
c 4 12037
n 4 12044
n 4 12058
n 4 12071
n 4 12085
n 4 12098
n 4 12111
n 4 12124
c 4 12144
n 4 12151
END_DNA

END_SEQUENCE
```

Phred: QUAL files

- Quality values in FASTA format

```
>106   542     0    542  ABI trimmed
15 15 16 16 16 13 14 16 16 17 16 12 14 15 19 13 15
18 19 18 13 22 29 20 10 13 11 13 13 19 23 25 26 22
23 25 25 29 33 29 19 12 12 16 25 27 48 48 44 40 40
40 40 40 40 35 35 35 35 35 35 40 51 51 45 45 45 45
45 45 51 45 45 45 45 45 45 45 51 51 56 56 56 51 51
45 45 45 45 51 51 51 45 45 45 45 45 45 45 45 45 45
51 51 51 51 51 45 45 45 51 51 51 51 56 56 56 56 56
56 56 56 56 56 56 51 51 51 51 51 51 51 51 51 51 51
51 51 51 56 51 51 39 39 35 35 40 40 56 51 56 56 56
56 56 56 56 56 56 56 56 56 56 51 51 51 51 51 51 51
51 56 56 56 56 56 56 56 56 56 56 56 45 45 45 45 45
45 56 56 45 45 45 45 45 45 56 56 56 56 56 51 51 51
56 56 56 56 56 56 56 56 51 51 51 51 51 51 56 56 56
56 56 56 56 56 56 56 56 51 51 51 51 51 51 45 45 45 41
45 51 56 56 56 56 56 56 56 56 56 56 56 56 56 51 51
51 51 51 56 56 56 51 51 51 51 51 56 56 56 56 56 56
56 56 56 56 56 56 56 51 51 51 51 51 56 56 56 56 56 56
56 56 56 56 56 51 51 45 45 37 37 37 40 45 45 45 45 51
51 51 51 51 51 56 56 45 45 45 45 45 45 56 56 51 40
40 40 40 40 40 51 51 51 56 56 56 56 56 56 56 56 56
56 56 56 51 51 51 51 40 40 45 45 40 40 40 40 45 45
56 45 45 45 45 45 51 56 56 56 51 39 39 35 35 35 37
46 51 51 51 51 51 56 56 56 51 51 51 51 51 51 51 40
40 40 40 40 40 40 40 40 40 40 34 34 34 32 40 40 32
32 32 32 32 32 32 32 29 29 31 40 56 56 56 40 51 51
51 43 43 56 56 56 56 45 40 40 40 40 39 40 40 40 40
40 51 44 44 40 40 40 40 39 32 29 29 27 29 31 34 34
32 25 25 18 13 13 19 32 40 40 34 29 29 29 40 40 24
17 8 8 9 19 24 40 29 29 25 27 29 29 27 20 14 12 9
9 12 9 10 15 18 24 25 21 23 24 24 27 29 32 33 33 27
23 18 18 23 21 25 29 29 29 29 29 32 40 23 19 9 9 9
15 24 29 29 29 29 29 40 40 32 32 24
```

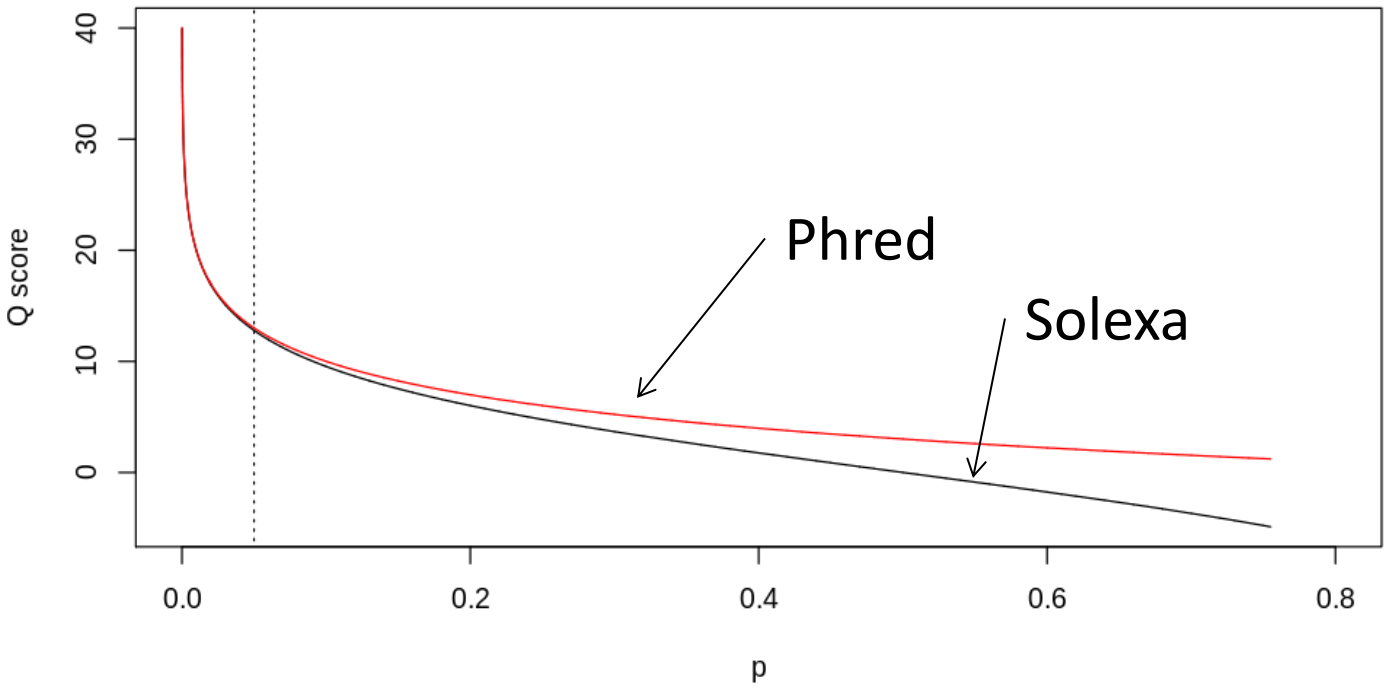


Quality values: Solexa (Illumina)

Solexa (Illumina) qualities for their version 1.3 pipeline

$$q = -10 \times log_{10}(\frac{p}{1-p})$$

Odds



Relationship between *Q* and *p* using the Sanger (red) and Solexa (black) equations (described above). The vertical dotted line indicates $p = 0.05$, or equivalently, $Q \approx 13$.

FASTQ Format

El formato **FASTQ** guarda información de secuencia y de calidad en el mismo archivo.

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

@ = linea de texto que contiene al identificador
+ = separador (arriba la secuencia, abajo la calidad)

Los valores de calidad están ***codificados.***
Los caracteres "@" y "+" pueden aparecer en esta cadena de caracteres!

Sanger format = Phred Q (0 – 93) se codifica utilizando los códigos ASCII 33 al 126
Solexa 1.0 = Phred Q (-5 – 62) se codifica utilizando ASCII 59 al 126
Solexa 1.3 = Phred Q (0 – 62) se codifica utilizando ASCII 64 al 126
Solexa 1.8 = Sanger format (Phred Q + 33)

ASCII Table

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

```
@HWUSI-EAS582_157:6:1:1:1501/1
NCACAGACACACACGAACACACAAAGACATGCCCATATGAAGAT
+
%.7786867:7785568587465750588 73/347777476035
@HWUSI-EAS582_157:6:1:1:1606/1
NCTGGCACCTTGATTTTGGACTTCCCAGCCTCCAGAACTGTGAG
+
%1948988888798988366898888648998788898888588
@HWUSI-EAS582_157:6:1:1:453/1
NCTGCTTGCACCCCTGAAGTCACTGATCACATTTCAGGGTCACC
+
%/8689989888888676688889866447889884134888885
@HWUSI-EAS582_157:6:1:1:1844/1
NGATTGACATTGGCAAAGAGGACAACTGATTGCAAACTTCACAC
+
%-7;:::::;86499;75574586::635:62687666887879
@HWUSI-EAS582_157:6:1:1:1707/1
NAGGCTCAGGCGCACGGCCTACATCGTCGCTGTCGGCCAAGGGG
+
```

"Read" (sequence)

Quality scores (phred-33)

Illumina sequence identifiers

Sequences from the Illumina software use a systematic identifier:

```
@HWUSI-EAS100R:6:73:941:1973#0/1
```

HWUSI-EAS100R	the unique instrument name
6	flowcell lane
73	tile number within the flowcell lane
941	'x'-coordinate of the cluster within the tile
1973	'y'-coordinate of the cluster within the tile
#0	index number for a multiplexed sample (0 for no indexing)
/1	the member of a pair, /1 or /2 *(paired-end or mate-pair reads only)*

http://en.wikipedia.org/wiki/FASTQ_format

Qué hacer con las secuencias?

Obtuvimos nuestros datos: y ahora qué?

- **Analizar la calidad**
- **Pre-procesar (filtrar, recortar)**

Esto permite identificar contaminaciones, y problemas en la construccón de las bibliotecas, y mejorar los datos para los pasos subsiguientes.

- **Ensamblar**
- **Mapear contra referencia**

Analizar calidad global

Se analiza la calidad de toda la corrida!

Una herramienta muy útil es **FASTQC**

FastQC

Function	A quality control tool for high throughput sequence data.
Language	Java
Requirements	A suitable Java Runtime Environment
Code Maturity	Stable. Mature code, but feedback is appreciated.
Code Released	Yes, under GPL v3 or later.
Initial Contact	Simon Andrews

Download Now

Hay otros:

FASTX

PRINSEQ

TagCleaner

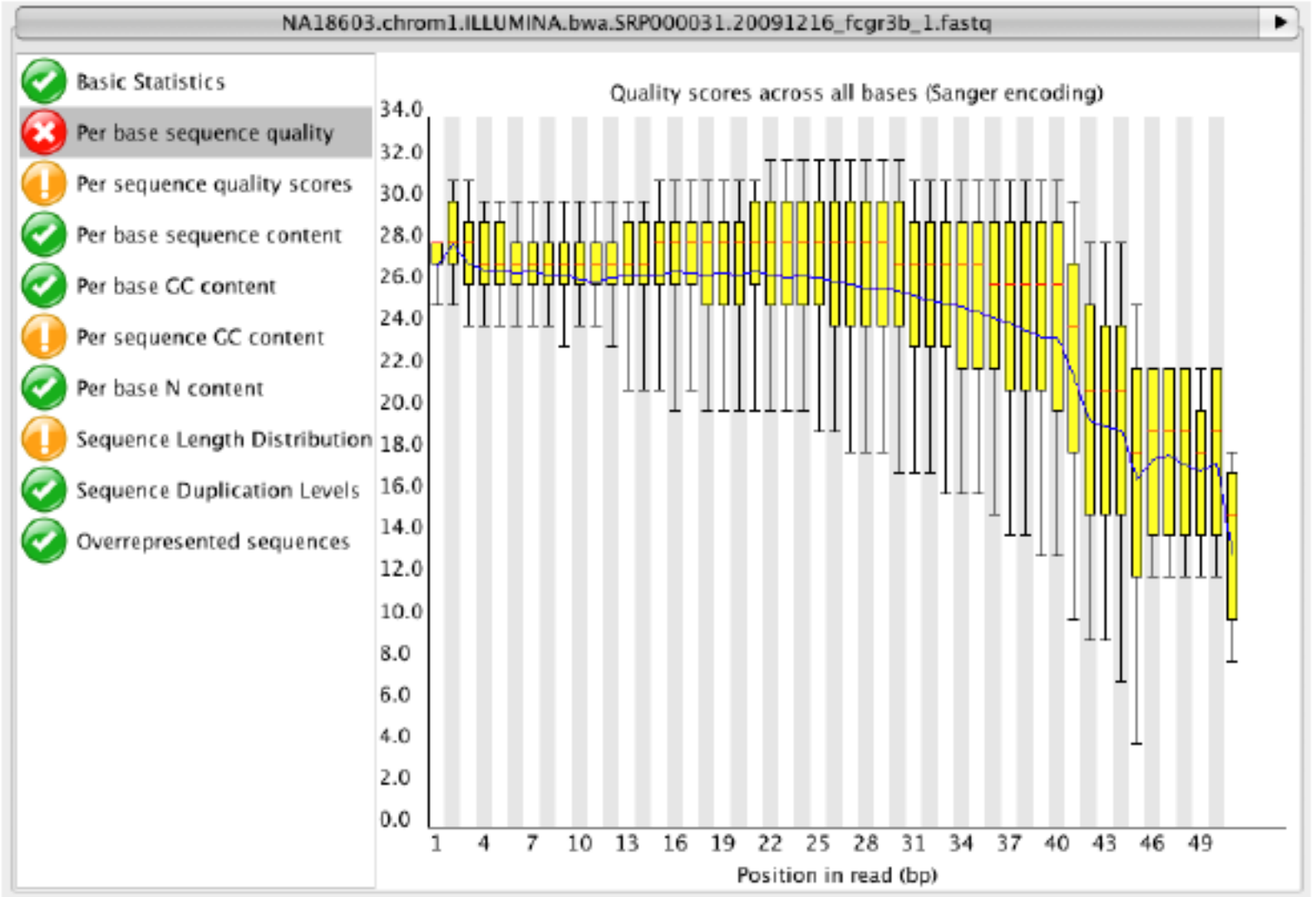
http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc/

FASTQC

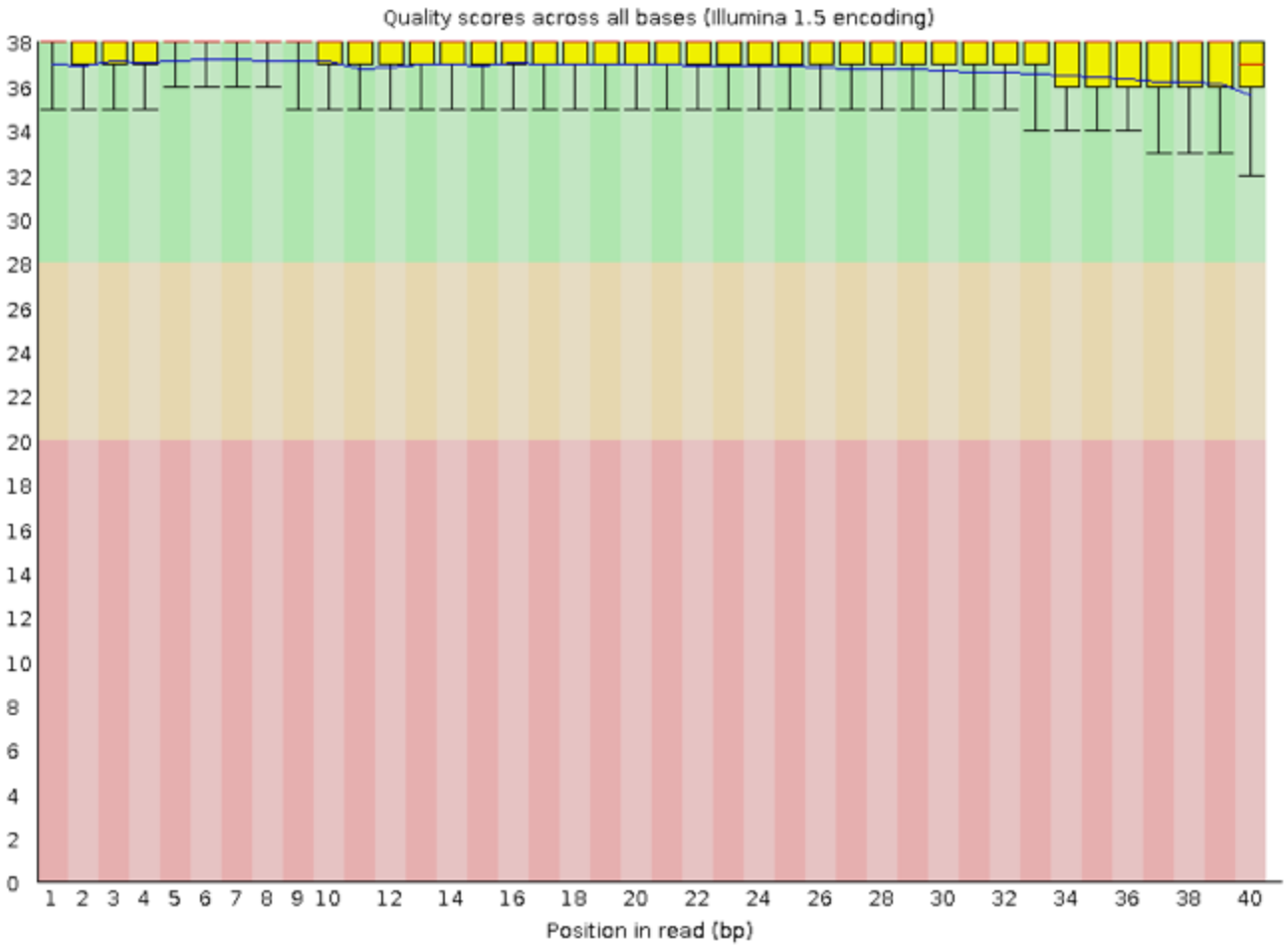
Qué cosas se chequean:

- **Calidad**
  - Calidad por base
  - Calidad por lectura
- **Composición**
  - Por base
  - Perfil de composición de GC
- **Identificación de contaminantes**
  - Secuencias sobre-representadas (k-mers)
  - Niveles de duplicación

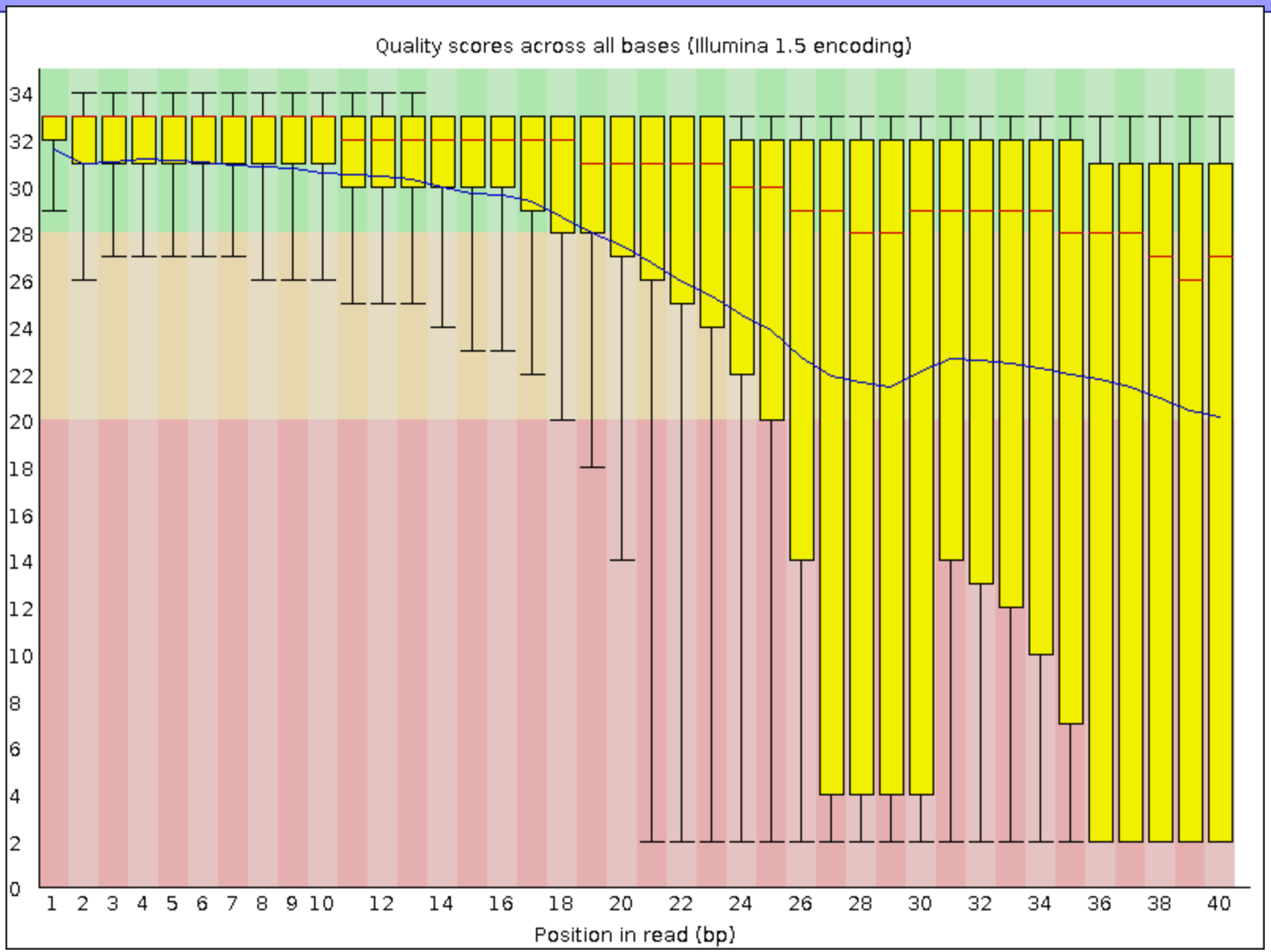
FASTQC

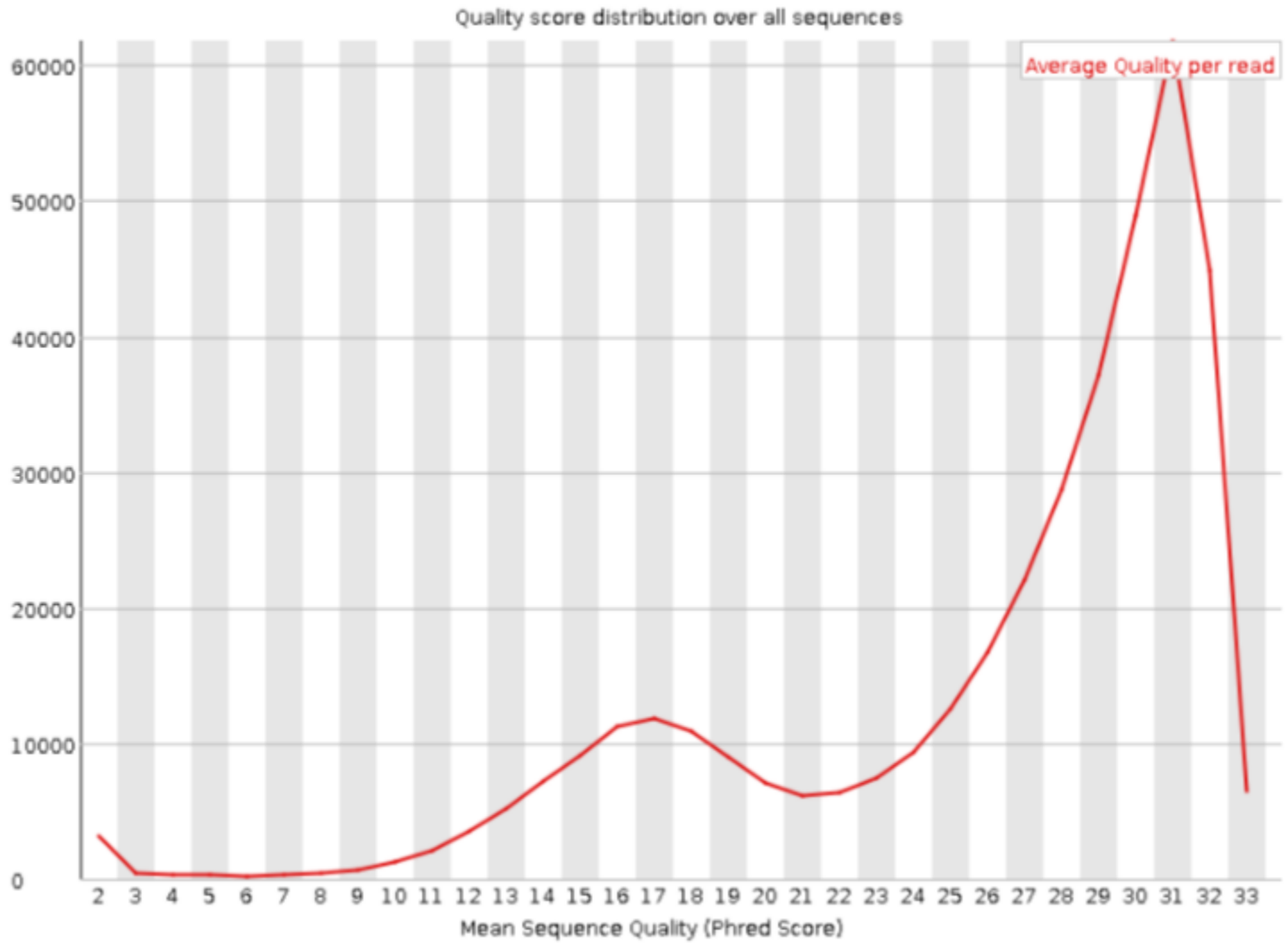

NA18603.chrom1.ILLUMINA.bwa.SRP000031.20091216_fcgr3b_1.fastq
Basic Statistics
Per base sequence quality
Per sequence quality scores
Per base sequence content
Per base GC content
Per sequence GC content
Per base N content
Sequence Length Distribution
Sequence Duplication Levels
Overrepresented sequences
Quality scores across all bases (Sanger encoding)
34.0
32.0
30.0
28.0
26.0
24.0
22.0
20.0
18.0
16.0
14.0
12.0
10.0
8.0
6.0
4.0
2.0
0.0
1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49
Position in read (bp)

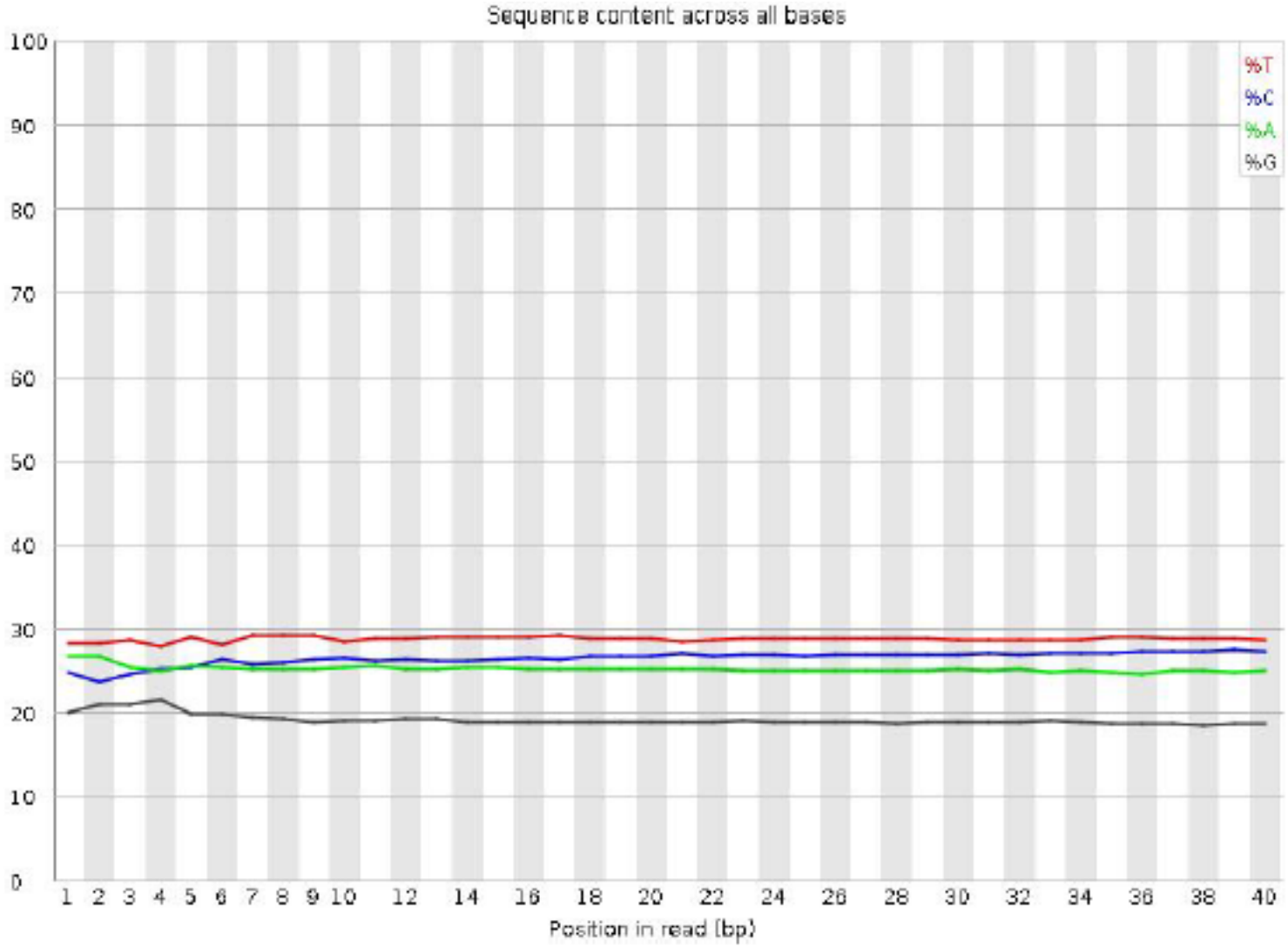
FASTQC

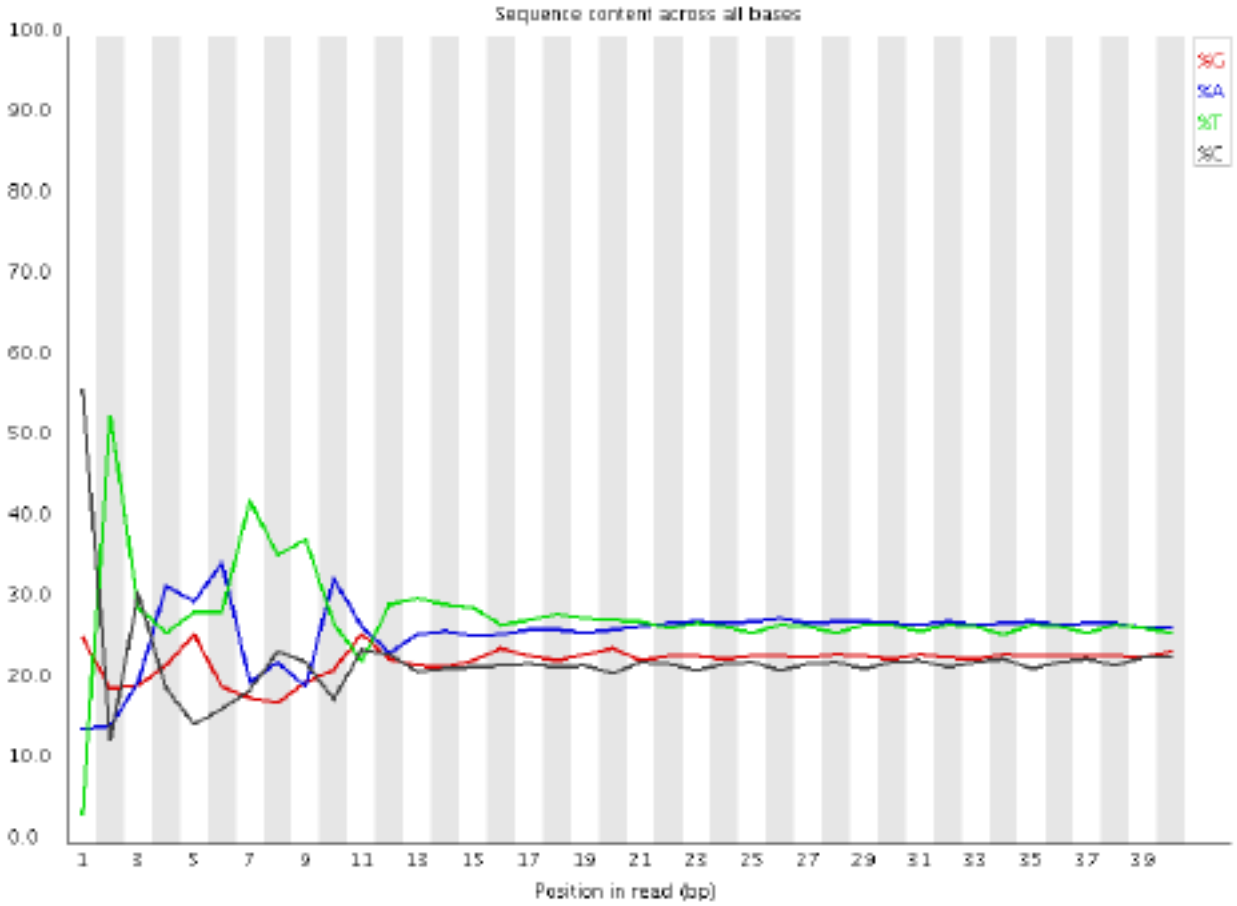

Quality scores across all bases (Illumina 1.5 encoding)
38
36
34
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
0
1 2 3 4 5 6 7 8 9 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40
Position in read (bp)

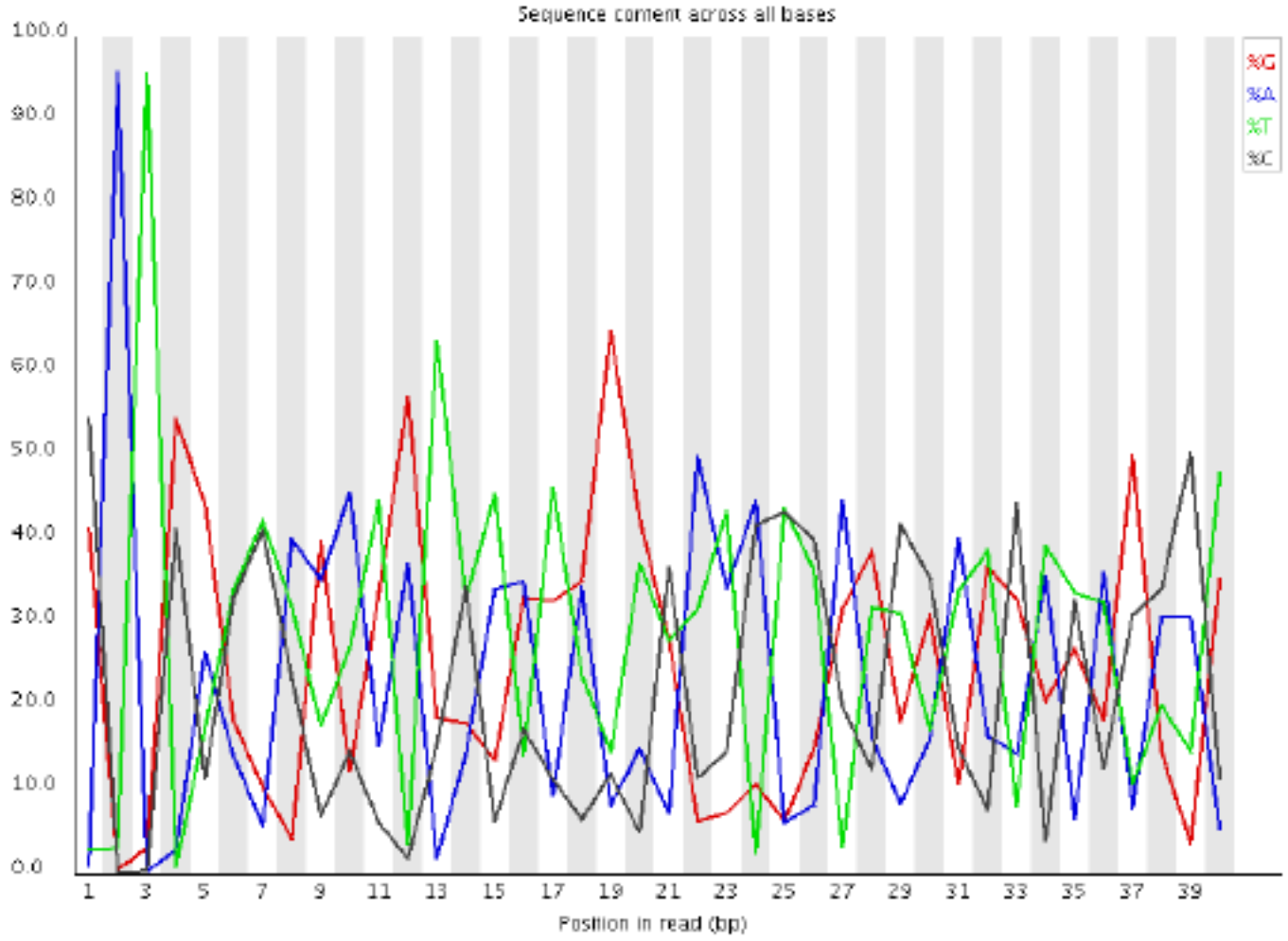
FASTQC


Quality scores across all bases (Illumina 1.5 encoding)
34
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
0
1 2 3 4 5 6 7 8 9 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40
Position in read (bp)


Quality score distribution over all sequences
Average Quality per read
60000
50000
40000
30000
20000
10000
0
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
Mean Sequence Quality (Phred Score)


Sequence content across all bases
%T
%C
%A
%G
100
90
80
70
60
50
40
30
20
10
0
1 2 3 4 5 6 7 8 9 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40
Position in read (bp)


Sequence content across all bases
100.0
90.0
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
Position in read (bp)
%G
%A
%T
%C


Sequence content across all bases
100.0
90.0
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
Position in read (bp)
%G
%A
%T
%C

Análisis de abundancia de kmeros

Qué son los K-meros

Un K-mero es un fragmento de una secuencia, de longitud K dentro de una cadena de bases (una secuencia mas larga de DNA).

Bases	K-mer size	Total possible kmers
4	1	4
4	2	16
4	3	64
4	4	256
4	5	1,024
4	6	4,096
4	7	16,384
4	8	65,536
4	9	262,144
4	10	1,048,576
4	…	…
4	21	4.4e+12
4	27	1.8e+16
4	31	4.6e+18

Por ejemplo: Todos los 2-meros de la secuencia AATTGGCCG son AA, AT, TT, TG, GG, GC, CC, CG. Y todos los 3-meros son AAT, ATT, TTG, TGG, GGC, GCC, CCG.

El número possible de K-meros se incrementa exponencialmente a medida que aumenta K (4^K).

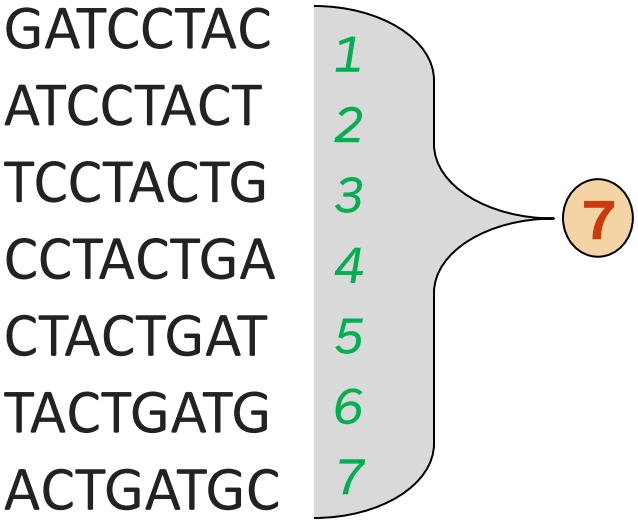
Análisis de abundancia de kmeros

Para una determinada secuencia de longitud ***L***, y un tamaño de K-meros ***K***, los posibles k-meros son ***(L – k) + 1***

K = 7 ; L = 18

(L – k) + 1 = 14 – 7 + 1 = ***8***

GATCCTACTGATGC



K = 18

Genome Sizes	Total K-mers of k=18	% diff in genome estimation
L	N=(L-K)+1	–
100	83	17
1000	983	1.7
10000	9983	0.17
100000	99983	0.017
1000000	999983	0.0017

Análisis de abundancia de kmeros

Cuando secuenciamos un genoma:

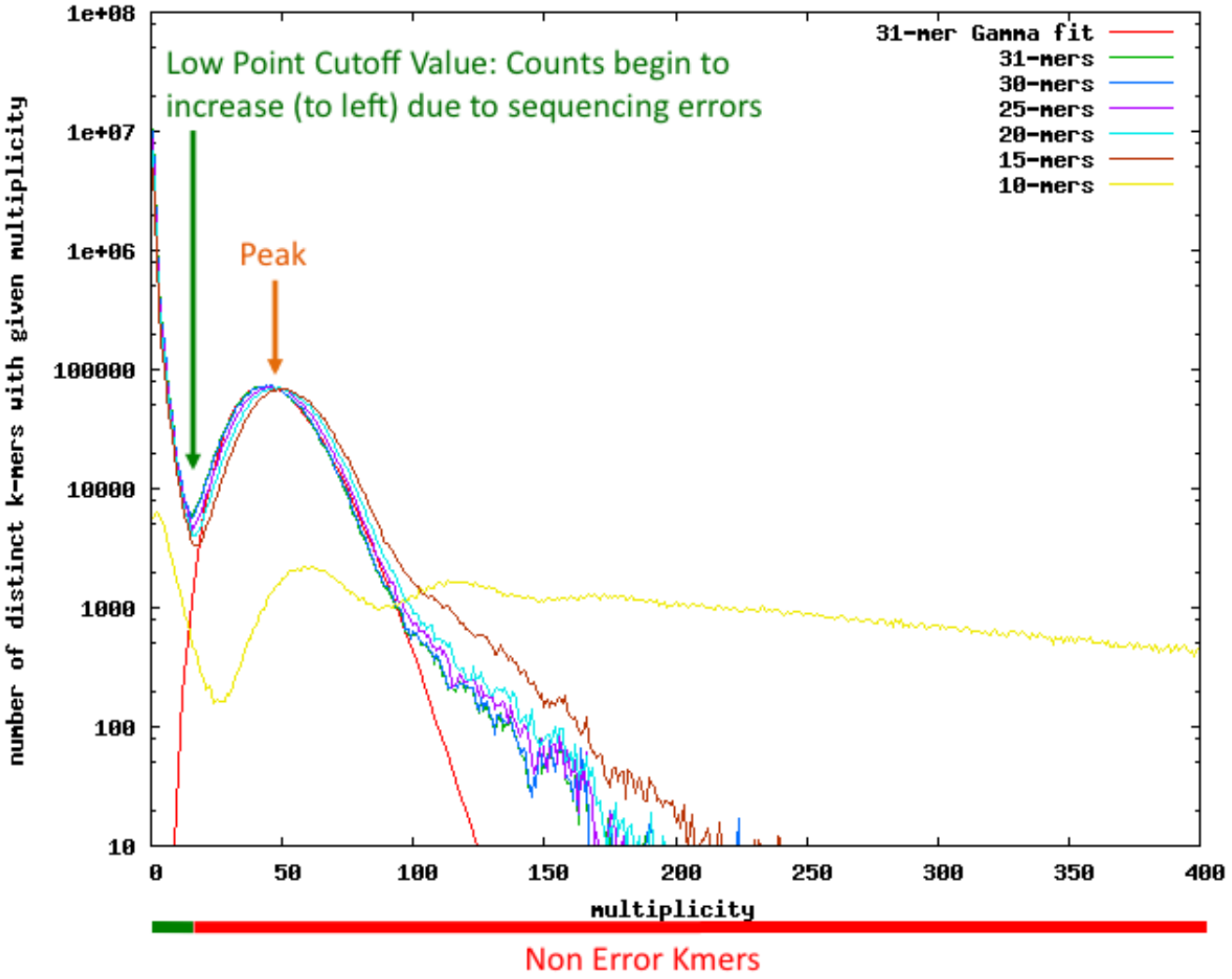
- Puede no haber cobertura ***uniforme***
  - ***Variabilidad técnica: amplificación sesgada de algunas regiones (PCR)***
  - ***Variabilidad biológica: secuencias repetitivas (perfectas o imperfectas)***
- Pero además!
  - ***Nunca secuenciamos 1 solo genoma!***
  - ***Secuenciamos un conjunto de genomas!!! (ADN aislado de una población de células)***

Tamaño del genoma (L)
Número de k-meros en el genoma (n)
Tamaño de k-meros (k)
Número de copias del genoma (C)

$$N = [(L - k) + 1] * C$$

Análisis de abundancia de kmeros

kmers = secuencias de longitud *k*



Metodos, algoritmos, técnicas para ensamblar genomas

SEQUENCE ASSEMBLY

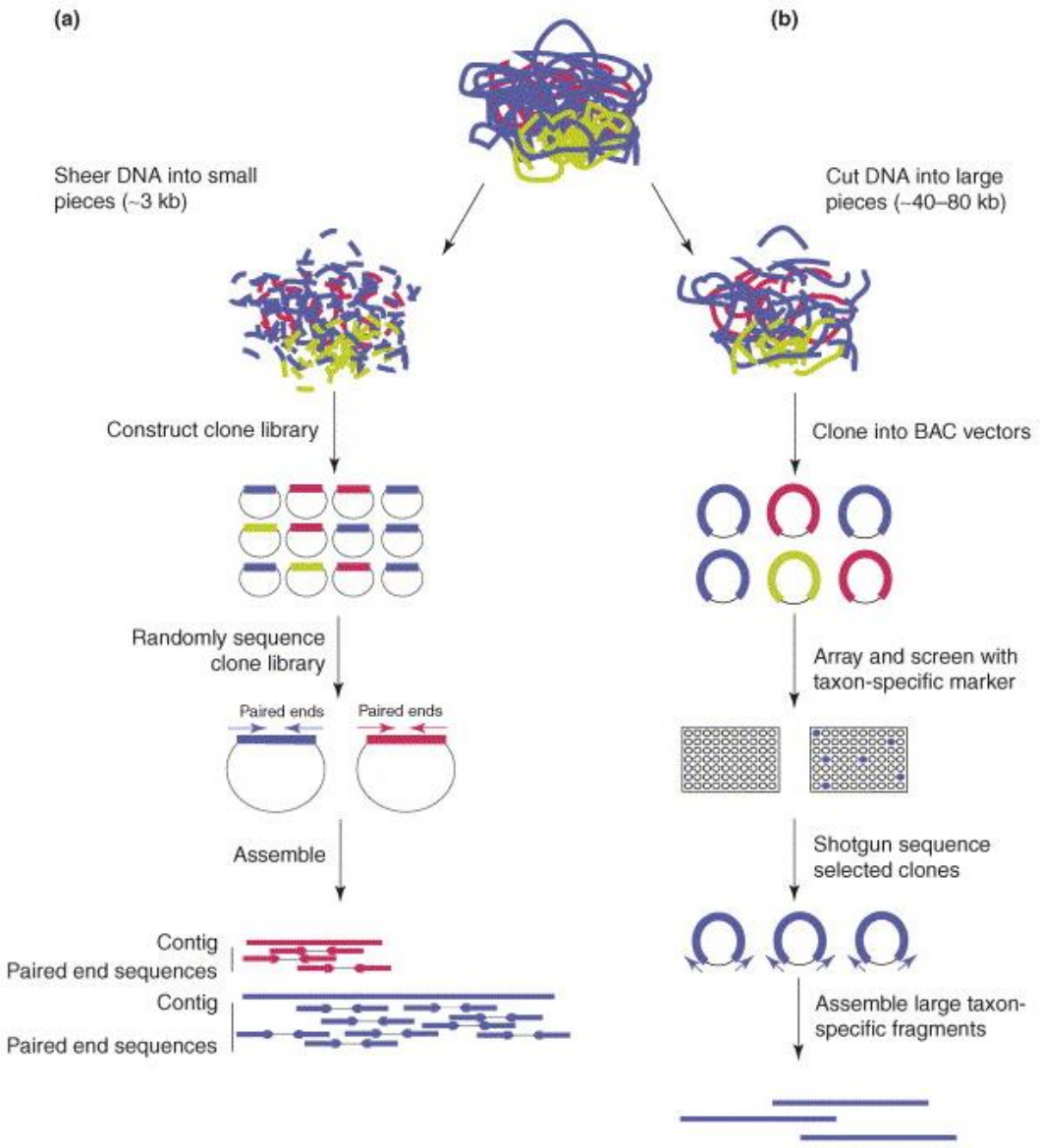
Qué es un assembly?

Dada una colección de lecturas (*"reads"*) con secuencia de DNA conocida, y una lista de *datos adicionales* sobre sus posicionamientos, encontrar la secuencia de ADN de la molécula original.

Datos adicionales **= datos opcionales, auxiliares que pueden ayudar a posicionar las secuencias**

Ensamblar secuencias

Shotgun sequencing



Population genomics in natural microbial communitiesAuthor links open overlay panel. Rachel J. Whitaker, Jillian F. Banfield. doi.org/10.1016/j.tree.2006.07.001

Terminología / Jerga

Assembly

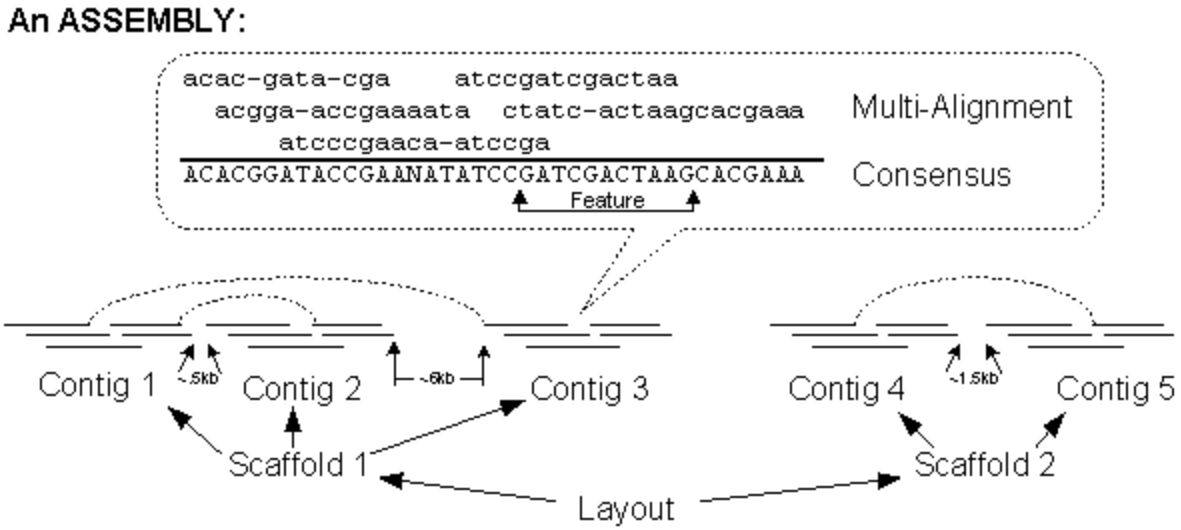
- Un conjunto de ***scaffolds***

Scaffold

- Un conjunto de ***contigs*** ordenados y orientados

Contig

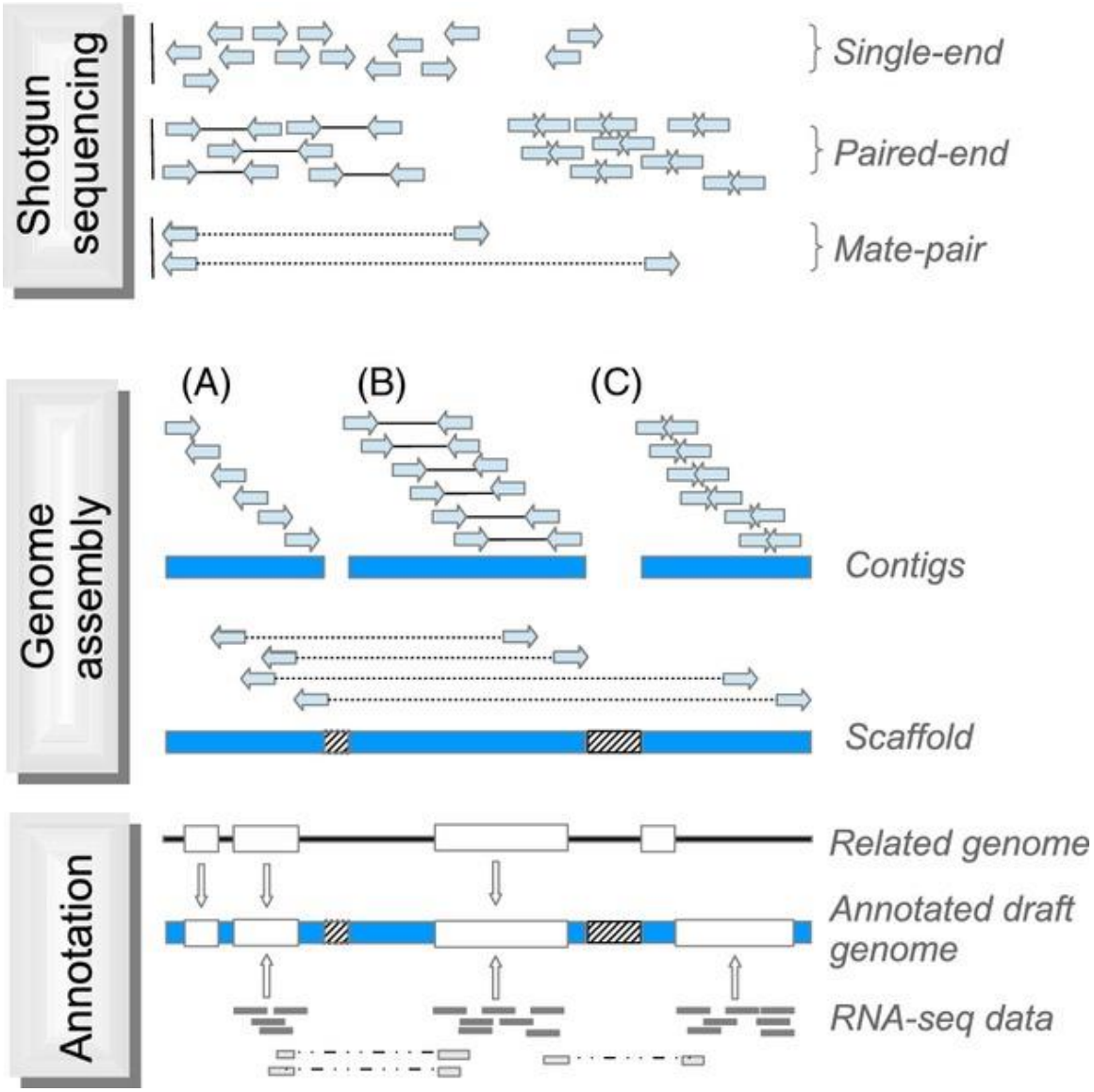
- Un conjunto de ***reads***
- Un ***layout*** que posiciona y ordena todos los ***reads*** sin dejar gaps
- Un alineamiento múltiple de los reads
- Una secuencia consenso



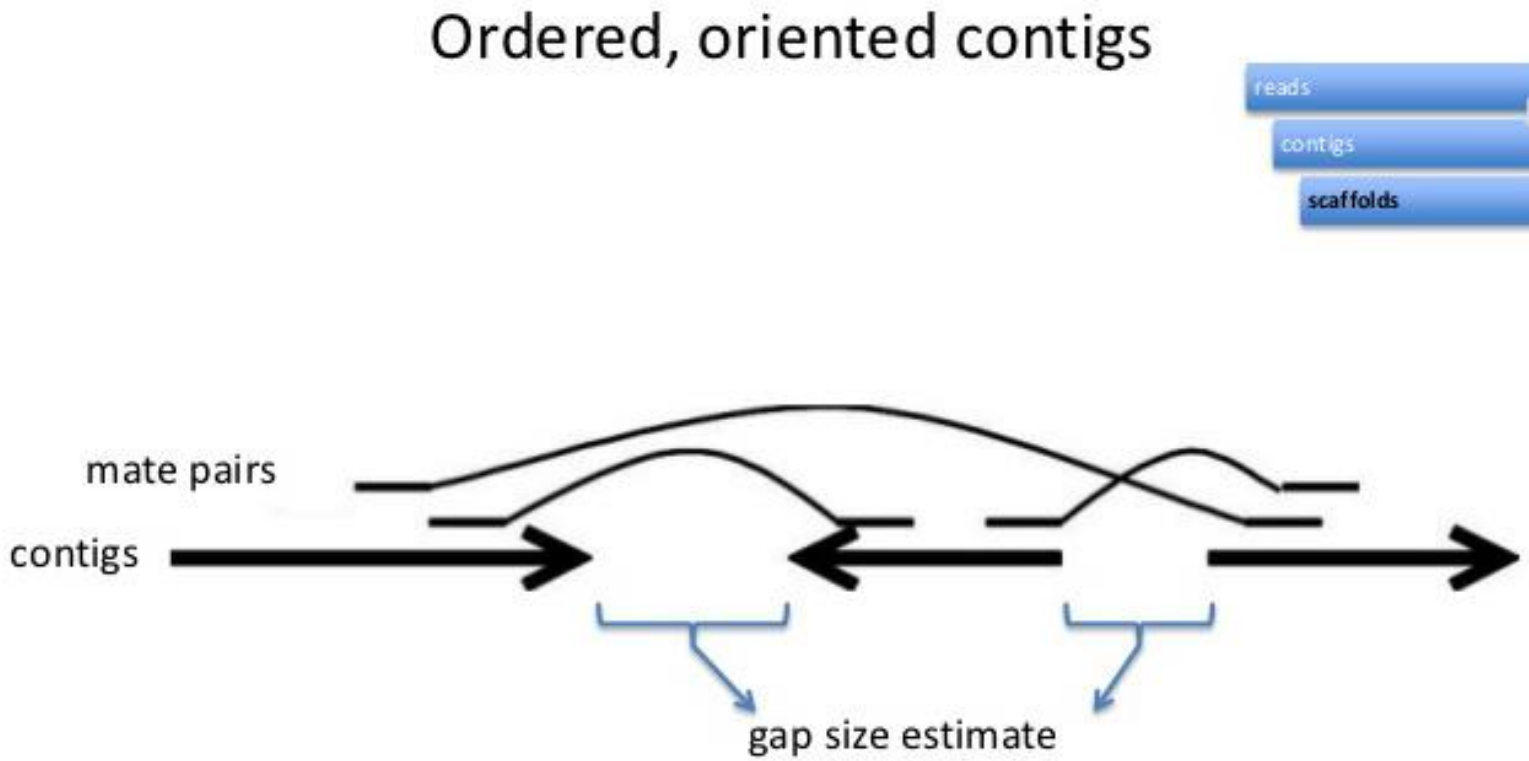
http://wgs-assembler.sourceforge.net/wiki/index.php/Celera_Assembler_Terminology

Genome sequencing, assembly and annotation: overview

- Distintos tipos de estrategias ***shotgun***
- Uso de la información del shotgun para guiar el assembly



Insert sizes, paired reads

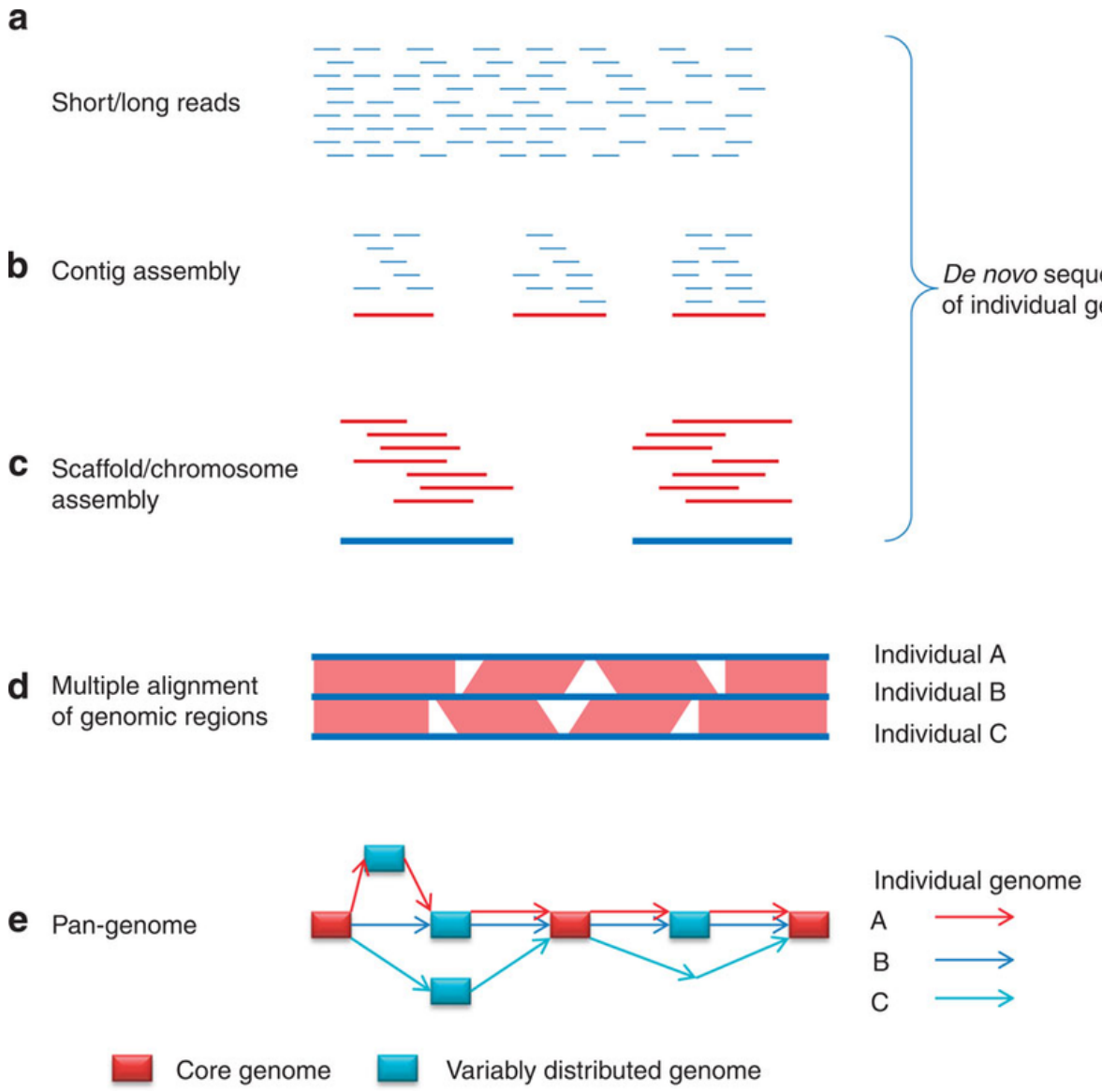


Las **lecturas apareadas** permiten **agrupar y orientar contigs**. El **tamaño de inserto** permite obtener estimaciones de tamaño de **gaps**.

From genomes to pan-genomes

Hacia la genómica de poblaciones

- **Core genome**
- **Variable genome**



Sequence assembly problem

Mapear “palabras” en cadenas de texto más largas es un problema conocido: “Exact string matching”

Naïve algorithm

Sequence assembly problem

“Exact string matching”

Naïve algorithm

Sequence assembly problem

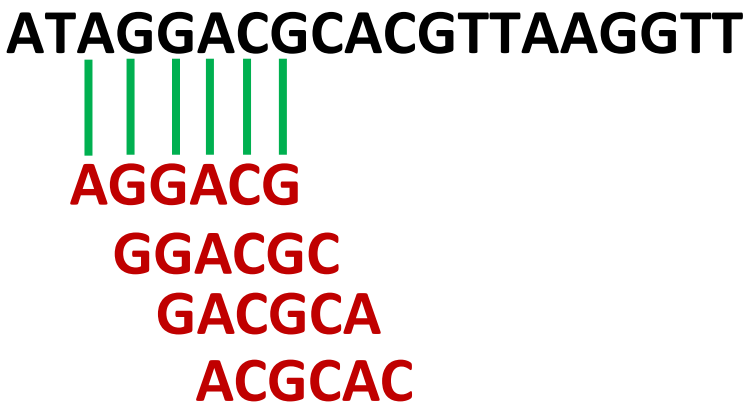
“Exact string matching”

Naïve algorithm

Sequence assembly problem

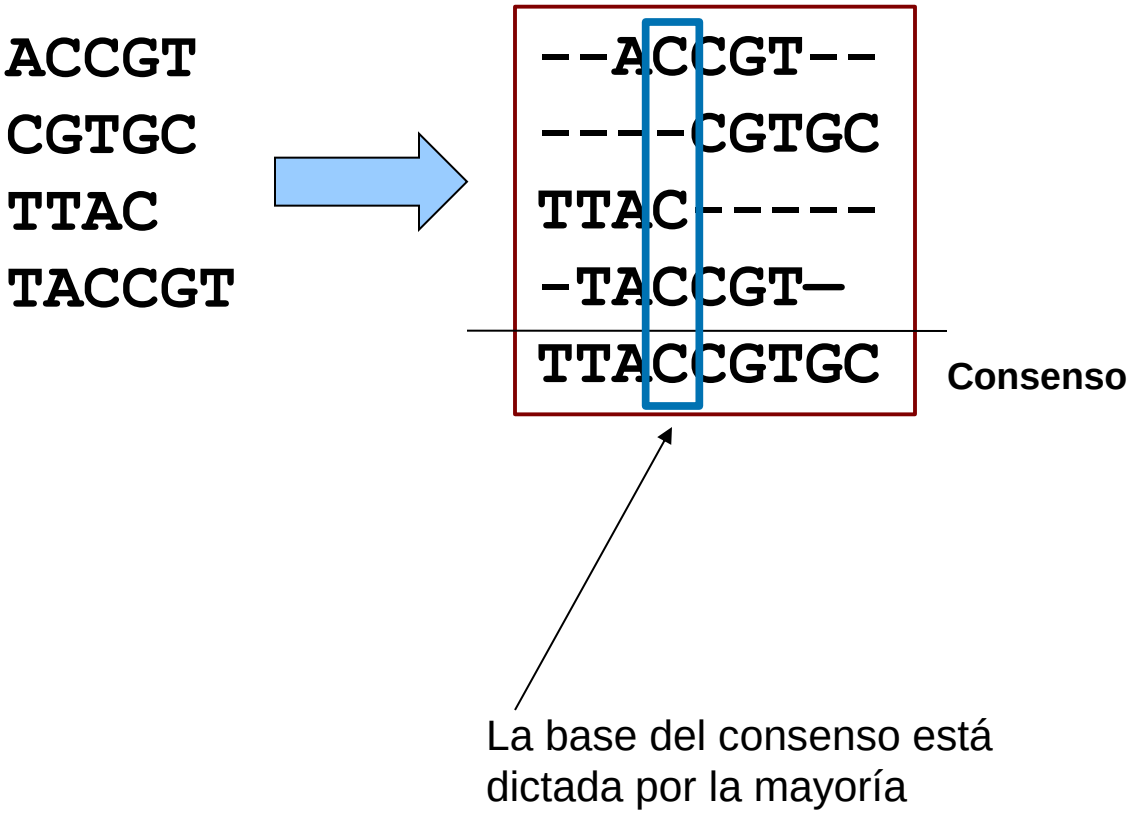
“Exact string matching”

Naïve algorithm



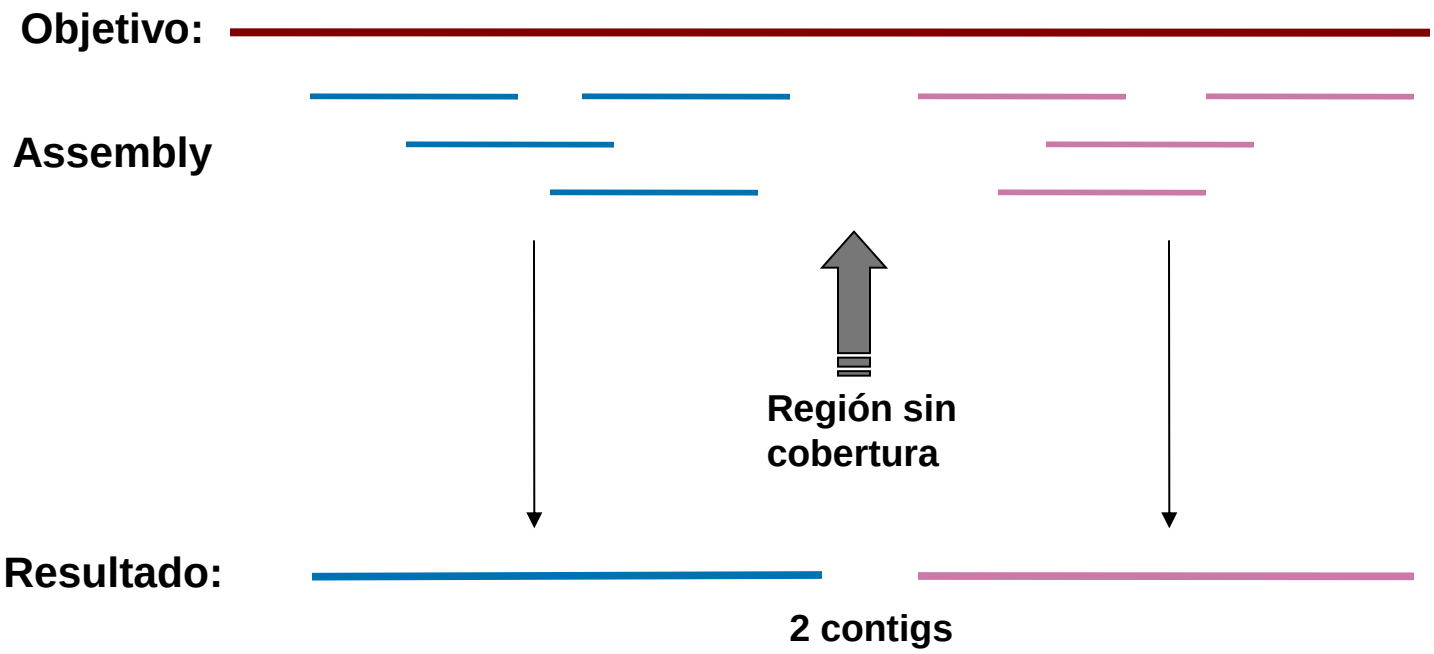
El problema de ensamblar secuencias

- Fragment assembly problem
- El caso *ideal*



Control de calidad del ensamblado

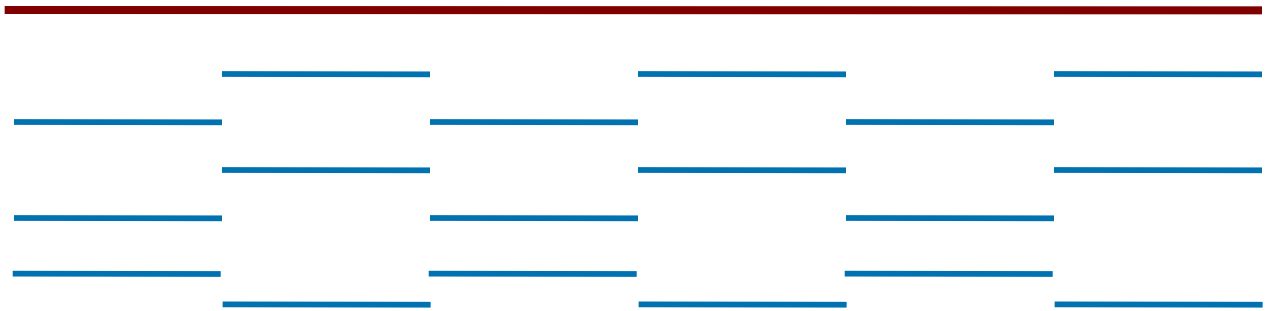
- Quality metrics



Control de calidad del ensamblado

- **Linkage –**
  - **grado de solapamiento (*overlap*) de los fragmentos**

Objetivo:



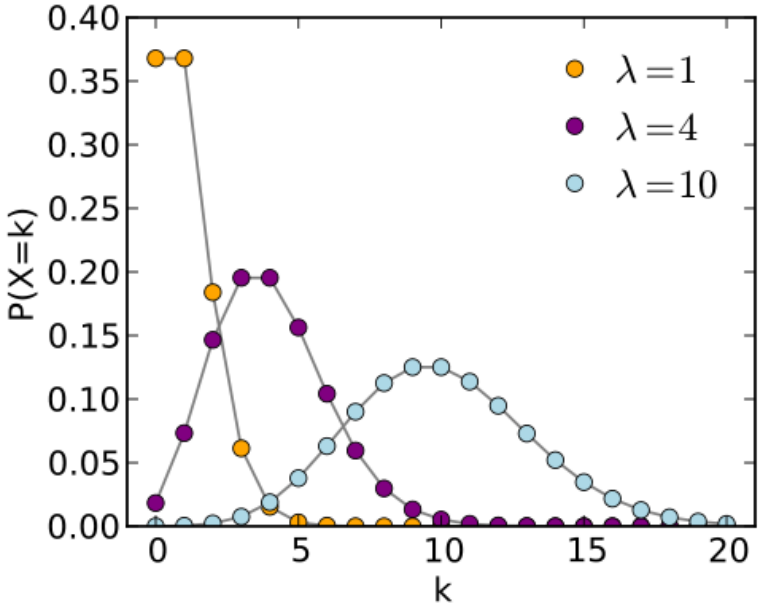
- **Alta** cobertura (coverage)
- Solapamiento promedio **pobre**
- Solapamiento *mínimo* también **pobre**

Cobertura de secuenciación

- **Cálculos de cobertura**
  - **Queremos secuenciar cada base 5x para tener un nivel de error aceptable**
  - **Qué cobertura promedio necesitamos para asegurar que el 95% de un genoma se secuencie *al menos* 5 veces?**
- **Se usa la distribución de Poisson**
  - **Si el número esperado de eventos (ocurrencias) es λ entonces, la probabilidad de observar exactamente κ eventos es**

$$f(k;\lambda) = \frac{\lambda^{\kappa} e^{-\lambda}}{\kappa!}$$

Distribución de Poisson



Ejemplo

- **Cobertura promedio = 5X**
  - **Número de veces *esperado* que va a ser leída una base (λ)**
- **La probabilidad de una base de haber sido secuenciada 10 veces es**
  - **Número de veces *observado* (κ)**

$$f(10;5) = \frac{5^{10}e^{-5}}{10!} = 0.018$$

- **0.018 (1.8%) del genoma va a ser leido 10 veces**
  - **Es decir: 1.8 % del genoma va a tener una cobertura de 10x**

Por qué es importante la cobertura?

Respuesta: Base quality values

$$q = -10 \times log_{10}(p)$$

Donde:

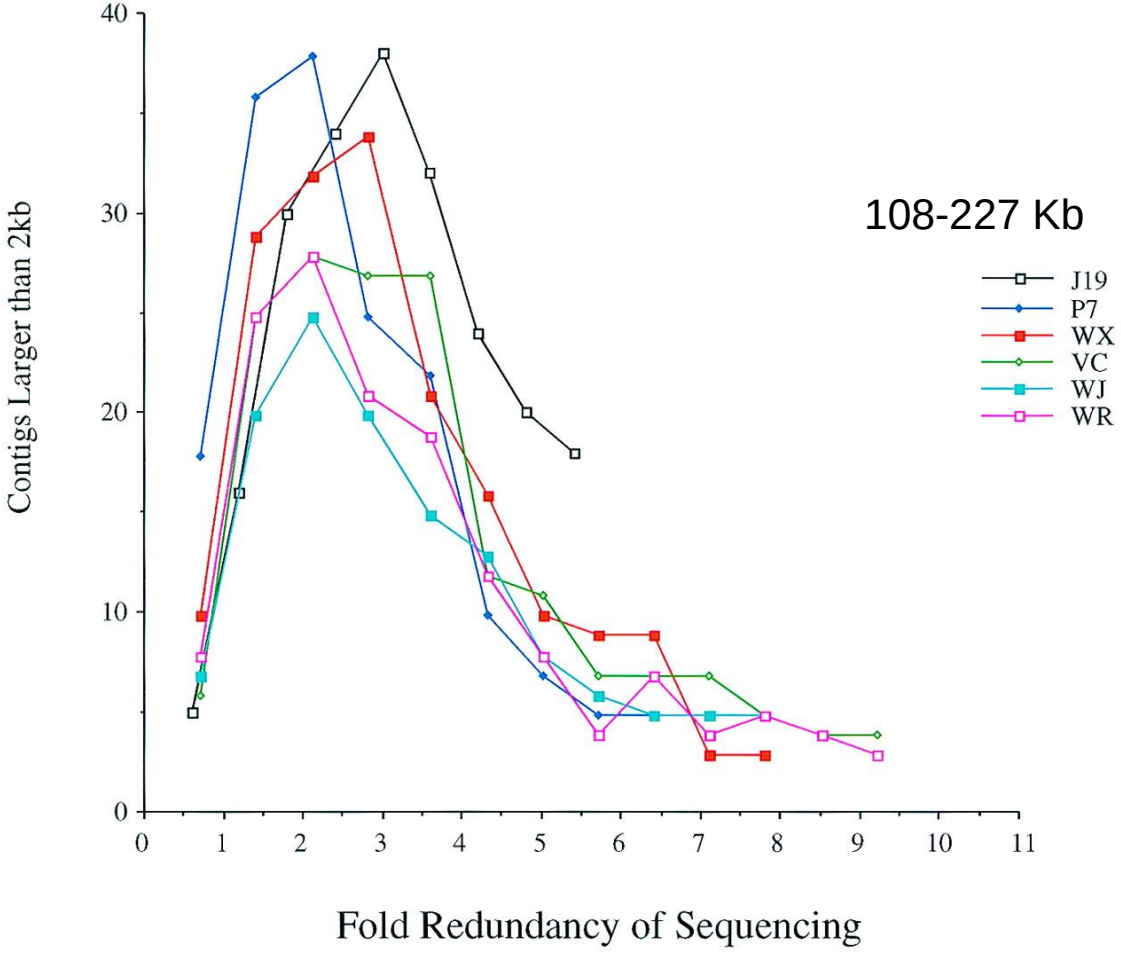
- **q = quality value**
- **p = estimated probability error for a base call**

Ejemplos:

- **q = 20 significa p = 10^{-2} (1 error cada 100 bases)**
- **q = 30 significa p = 10^{-3} (1 error cada 1000 bases)**
- **q = 40 significa p = 10^{-4} (1 error cada 10000 bases)**

Modelos para estimar gaps

- **Shotgun sequencing**
  - **Los clones que serán secuenciados se seleccionan al azar**
  - **Genera redundancia**
  - **La cobertura aumenta con el número de secuencias (pero no en forma lineal**

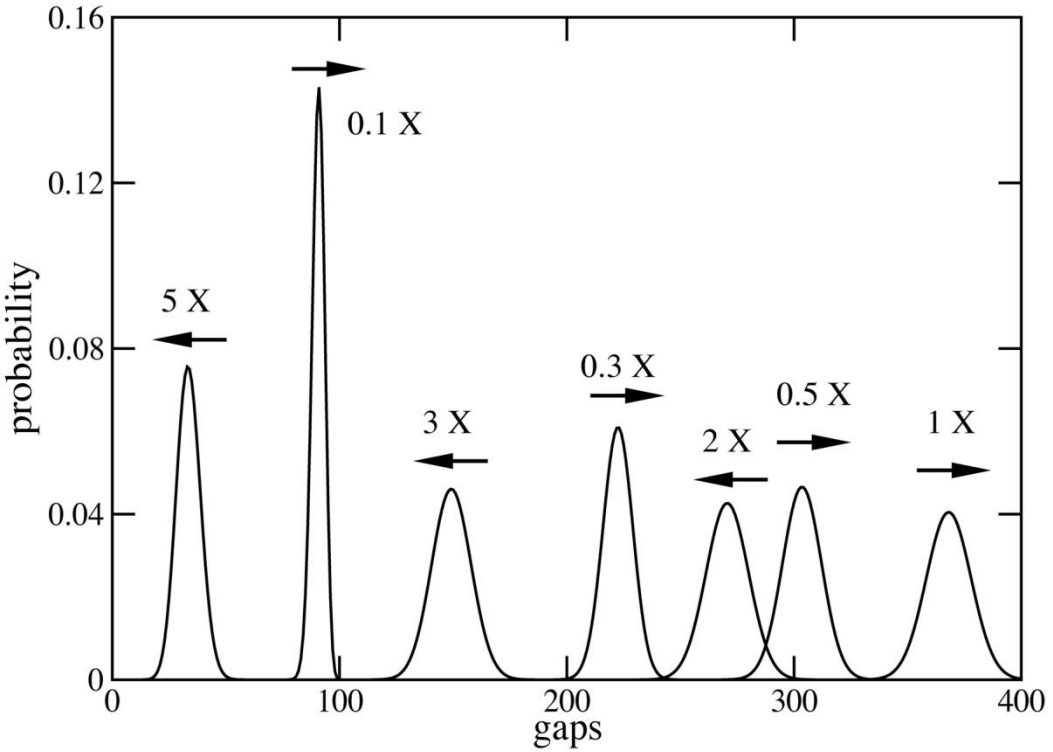


Contig formation at lower redundancy of sequencing. The number of contigs that were larger than 2 kb was calculated for each low redundancy simulation. The fold redundancy of each clone was calculated based on the number of bases that had a Phred value >20. The projects that were examined are listed at right. Tomado de Bouck *et al.* (1998) Genome Res 8: 1074.

Modelos para estimar gaps

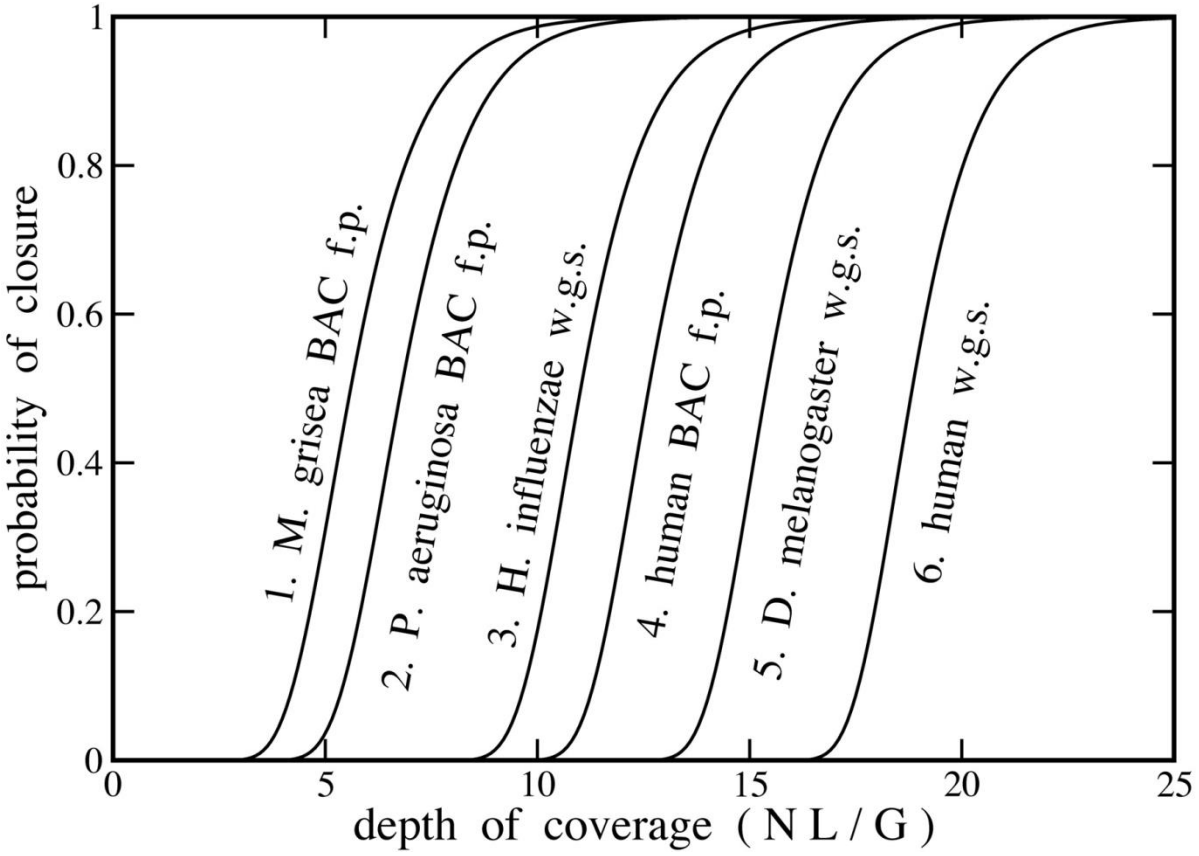
- **Wendl MC and Waterston RH. (2002). Generalized Gap Model for Bacterial Artificial Chromosome Clone Fingerprint Mapping and Shotgun Sequencing. Genome Res 12: 1943.**
  - **Función de densidad de probabilidades para *i* gaps en N clones**

Evolution of probability density function for a hypothetical project (L/G = 0.001, T/L = 0) up to 5× coverage as evaluated by equation 4. Arrows indicate whether the average number of gaps is increasing (→) or decreasing (←) for each distribution.
L = clone length
G = project length



Modelos para estimar gaps: closure

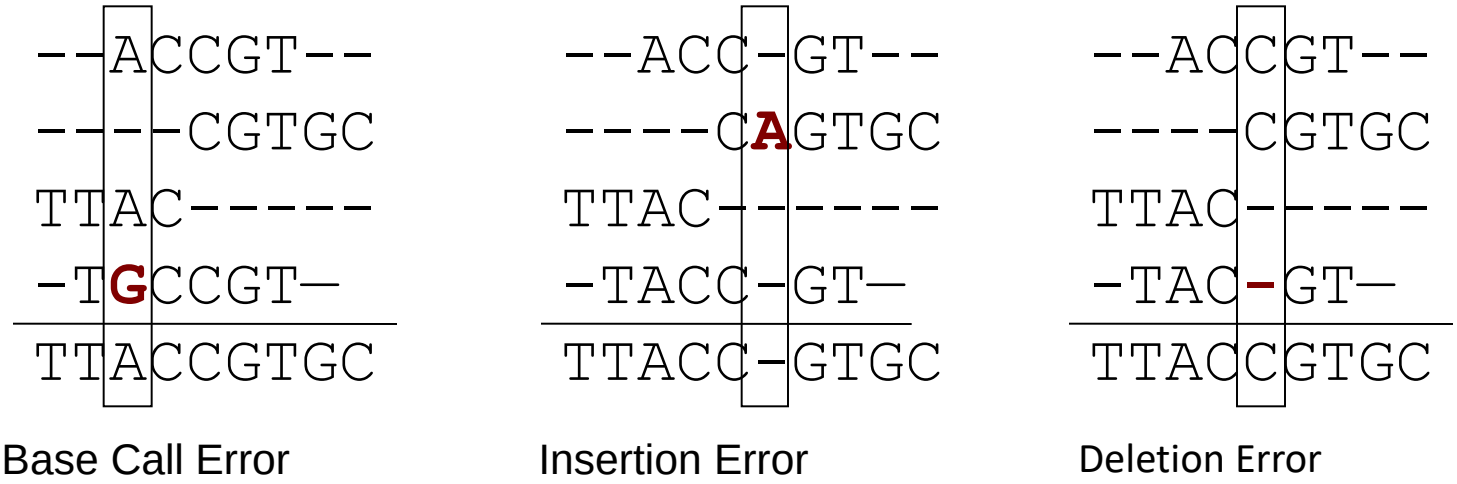
- **Closure**
  - **En proyectos de secuenciación por shotgun *closure* se refiere al momento donde un aumento de cobertura ya no produce cambios en la disminución del número de gaps (aumento del No. de contigs)**
  - **Probability of closure, p(0,N)**



Probability of closure as a function of depth of coverage for various projects: 1. Zhu et al. (1999); 2. Dewar et al. (1998); 3. Fleischmann et al. (1995); 4. McPherson et al. (2001); 5. Adams et al. (2000); 6.Venter et al. (2001). Abbreviations "f.p." and "w.g.s." represent fingerprint mapping and whole genome shotgun sequencing projects, respectively. Cases 1 and 2 were evaluated using equation 4, whereas the remaining cases were determined using equation 9. **Tomado de:** Wendl MC and Waterston RH (2002). Genome Res 12: 1943

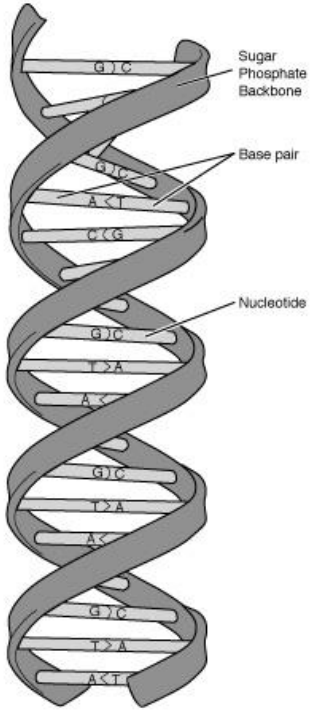
Errores usuales

- **Artefactos de clonado**
  - **Quimeras (dos insertos ligados en el mismo vector)**
- **Errores en la asignación de las bases**



Cosas que hay que resolver

- **Los fragmentos secuenciados pueden provenir de cualquiera de las 2 hebras del ADN originario**



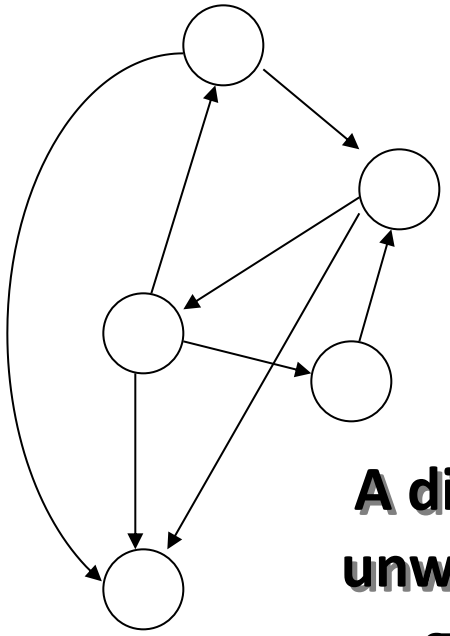

```
CACGT     →   CACGT
ACGT      →   -ACGT
ACTACG    ←   --CGTAGT
GTACT     ←   -----AGTAC
ACTGA     →   ---------ACTGA
CTGA      →   ----------CTGA
```

Sequence assembly models

- **Shortest common superstring (SCS)**
  - **Input: una colección $\mathcal{F}$ de cadenas de caracteres (fragmentos)**
  - **Output: la cadena más corta posible *S* en la cual se cumpla que**
    - Por cada $f \in \mathcal{F}$, S **es una supercadena de** f
- **Ejemplo 1**
  - $\mathcal{F}$ **= { ACT, CTA, AGT }**
  - S **= ACTAGT**
- **Ejemplo 2**
  - **Alfabeto = 0,1**
  - **Todos los 3-mers posibles para este alfabeto**
  - $\mathcal{F}$ **= { 000, 001, 010, 011, 100, 101, 110, 111 }**
  - S **= 0001110100**

Grafos

- La mayoria de las soluciones de reconstruccion de contigs a partir de fragmentos se resuelven modelando el problema como un ***grafo***
- **Un grafo es una coleccion de nodos y aristas (o vertices) que conectan los nodos**
  - **Dirigidos vs no dirigidos**
  - **Pesados (weighted) vs unweighted**
- **Vamos a ver mas sobre grafos …**

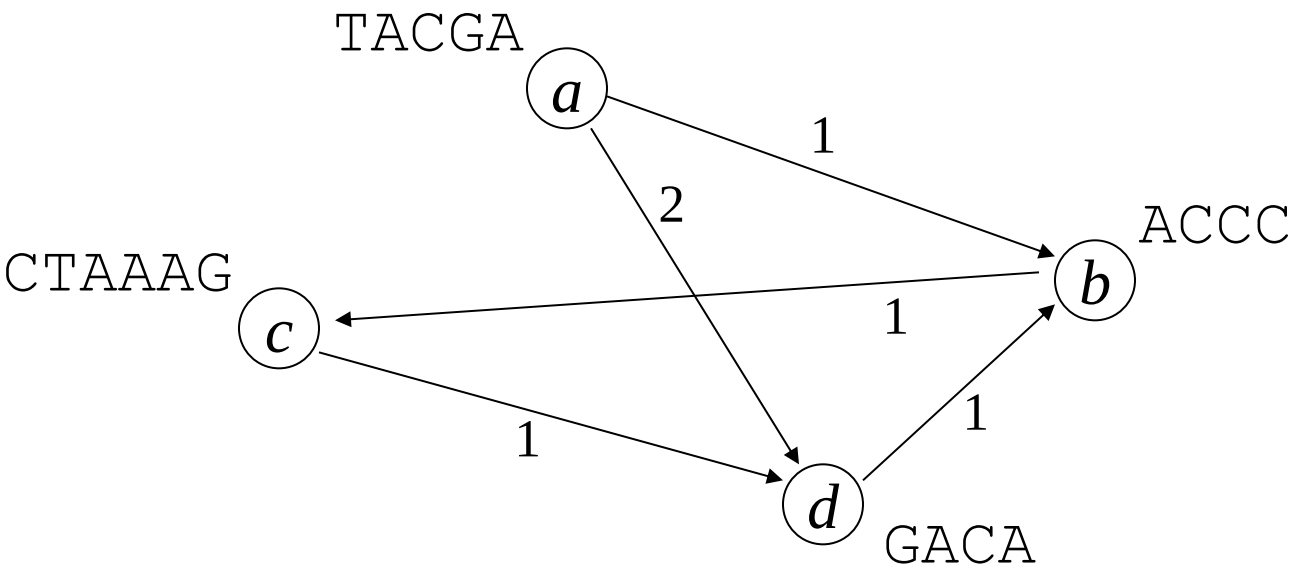


A directed, unweighted graph

Grafo de maximo solapamiento

Maximum overlap graph

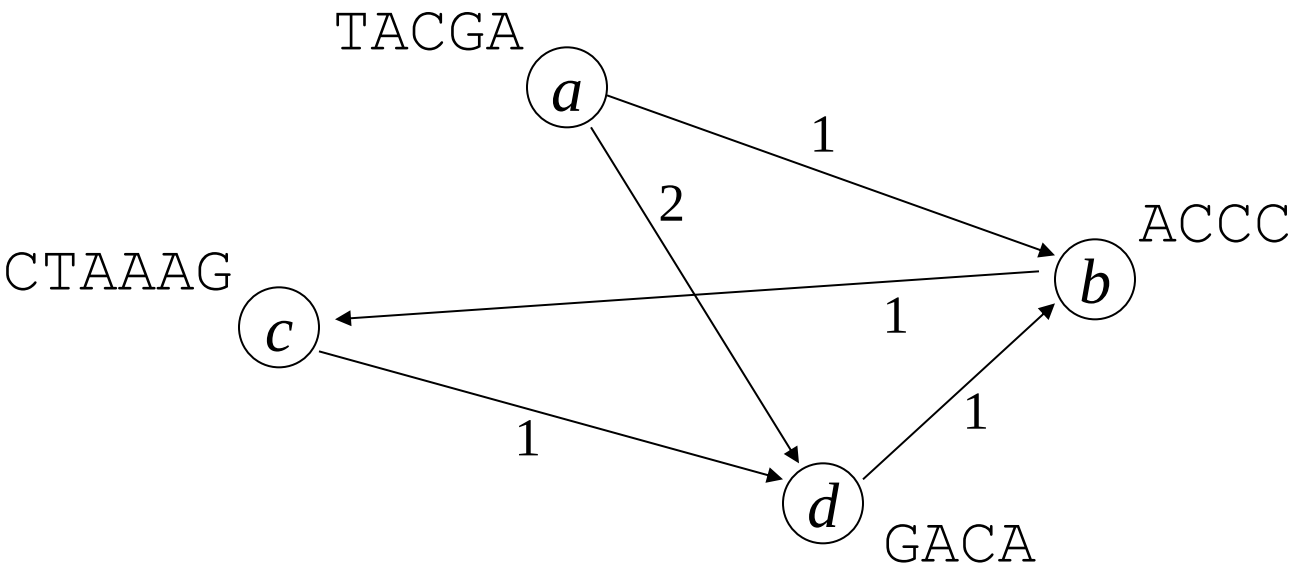
- El *peso* de cada vertice (u,v) corresponde a la longitud maxima de solapamiento entre un *prefijo* de u y un *sufijo* de v



Assembly graphs

El camino *dbc* corresponde al alineamiento:

```
GACA--------
---ACCC-----
------CTAAAG
```



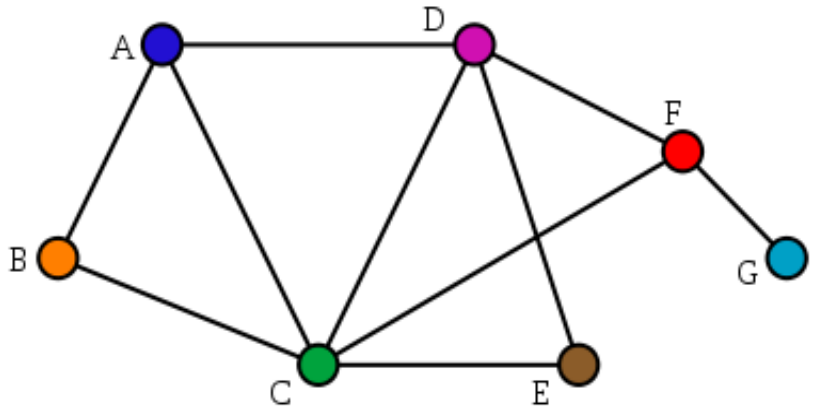
Assembly graphs

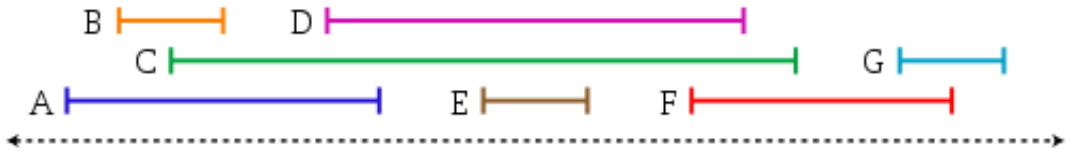
- Cada ***camino*** (path) dentro de un grafo, que recorra todos los nodos es un ***superstring***
  - **Los vertices con peso = 0 corresponden a alineamientos del tipo**

```
GACA---------
----GCCC-----
---------TTAAAG
```

  - **Vertices con pesos mas altos, producen alineamientos con mayor overlap (y por lo tanto cadenas mas cortas)**
  - **El superstring comun mas corto (SCS) es el camino con mayor peso que cubre todos los nodos**
- **Problema:**
  - **Input: un grafo dirigido, con pesos**
  - **Output: el camino con mayor peso (score) que recorre todos los nodos**
  - ***Suena familiar?***

- Grafos de intervalos
  - **Resultan de representar intervalos en forma de *grafo***
  - **Los intervalos son la proyección 1D del grafo**
    - 1 nodo por cada fragmento o intervalo
    - 1 arista entre cada par de intervalos que se ***solapan***



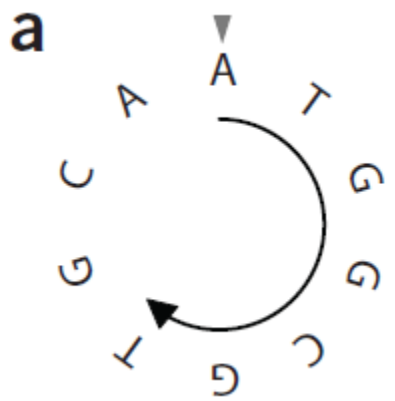


Tomado de http://en.wikipedia.org/wiki/Interval_graph

Ensamblando un genoma con grafos

Reconstruir (ensamblar) un genoma circular:

ATGGCGTGCA



A partir de una serie de reads:

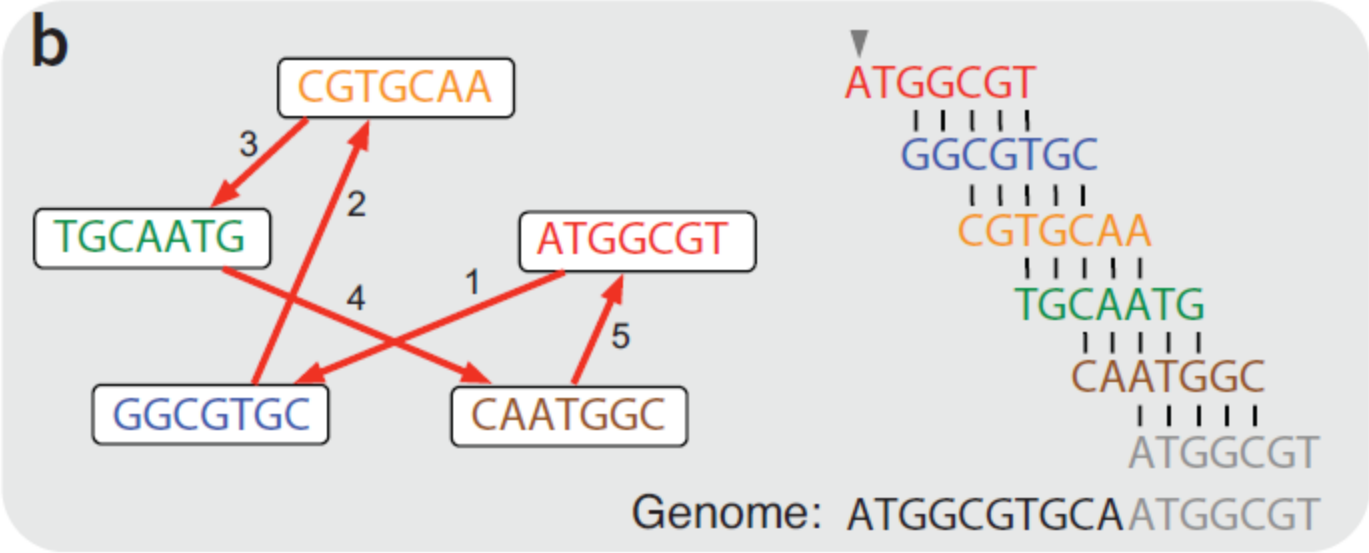
CGTGCAA

TGCAATG

ATGGCGT

GGCGTGC

CAATGGC



Solucion 1: Hamiltonian cycles

Una posible solución es representar las secuencias como un grafo de *k-mers*, donde los *edges* indiquen sufijos compartidos entre nodos. Ensamblar es buscar un camino en el grafo que pase por **todos** los *k-mers* (nodos).

Reads:

CGTGCAA

TGCAATG

ATGGCGT

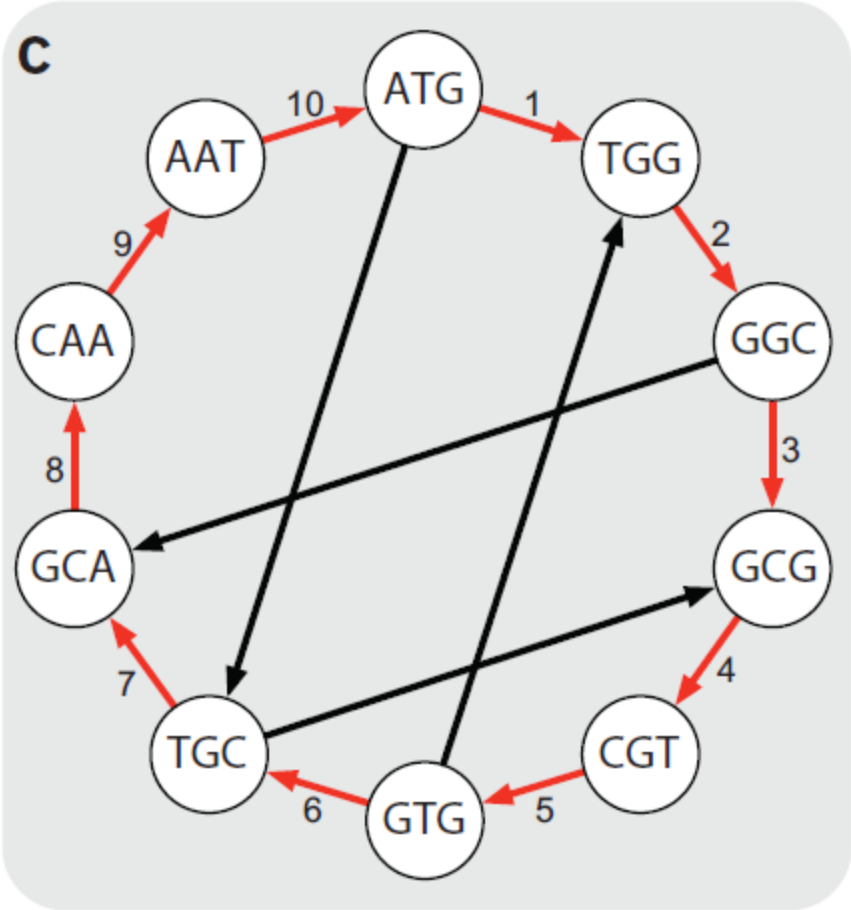
GGCGTGC

CAATGGC

Para k=3, los k-mers son:

CGT, GTG, TGC, GCA, CAA,

AAT, ATG, TGG, GGC, GCG



Hamiltonian cycle
Visit each vertex once

Solucion 2: Eulerian cycles

Otra solución posible: representar las secuencias como un grafo de *k-mers*, donde cada *edge* es un k-mer, y donde los nodos son prefijos y sufijos de cada k-mer. En este caso hay que buscar un camino que pase por todos los *k-mers* (ejes).

Reads:

CGTGCAA

TGCAATG

ATGGCGT

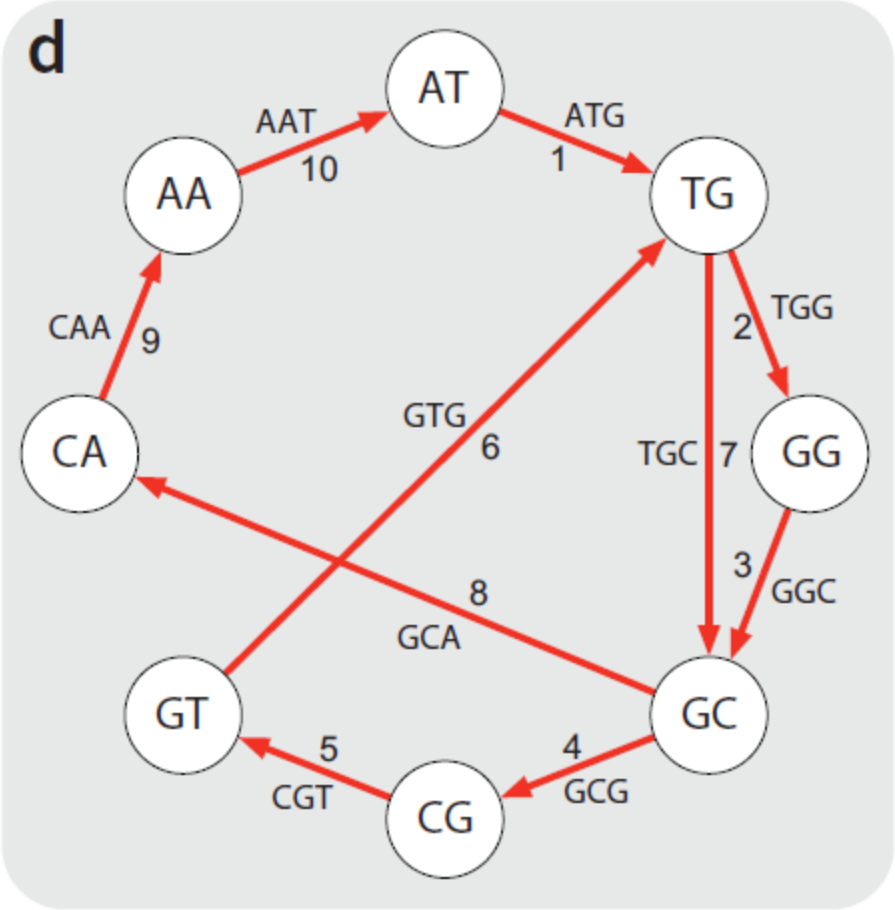
GGCGTGC

CAATGGC

Para k=3, los k-mers son:

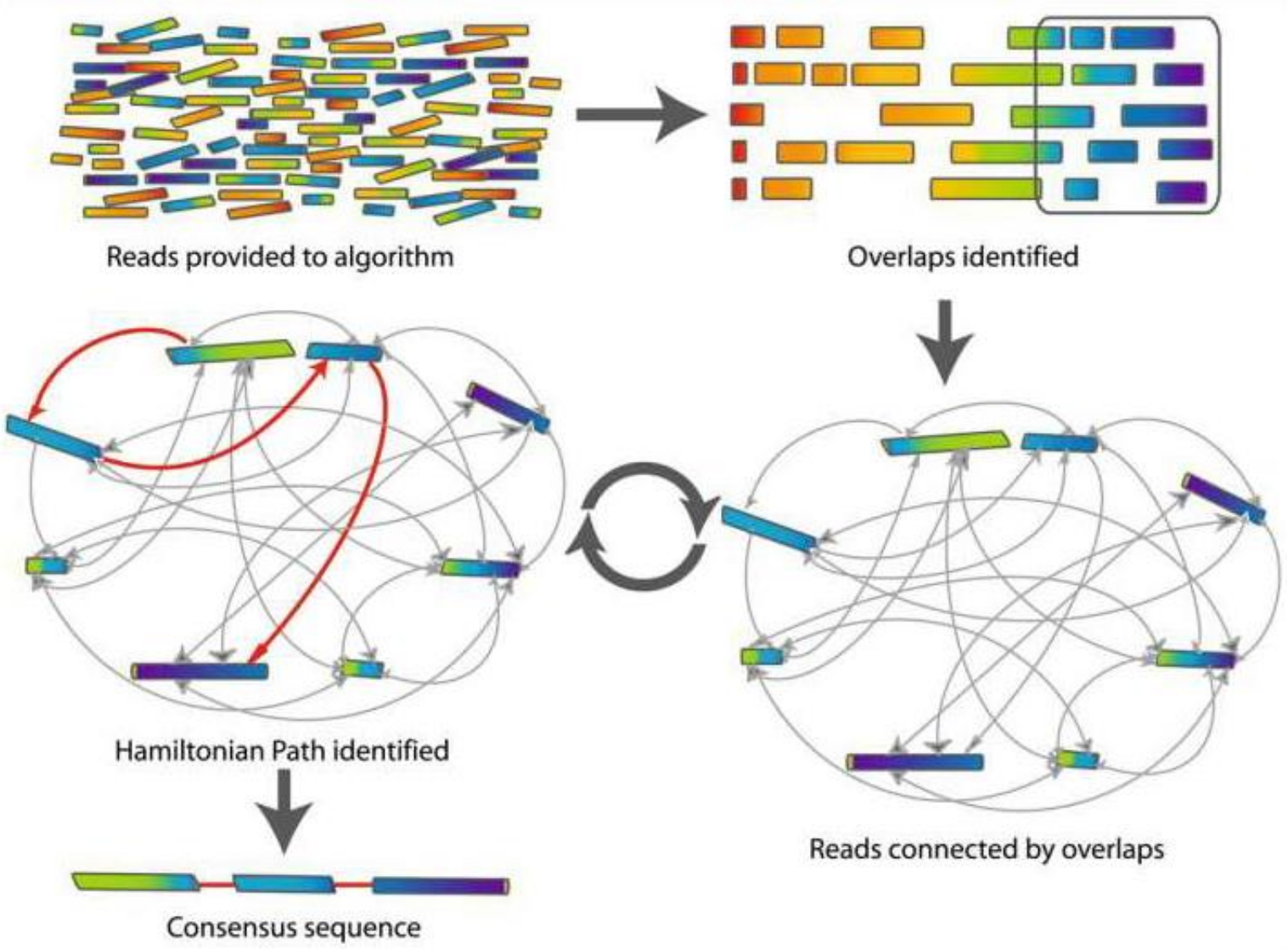
CGT, GTG, TGC, GCA, CAA,

AAT, ATG, TGG, GGC, GCG



Eulerian cycle
Visit each edge once

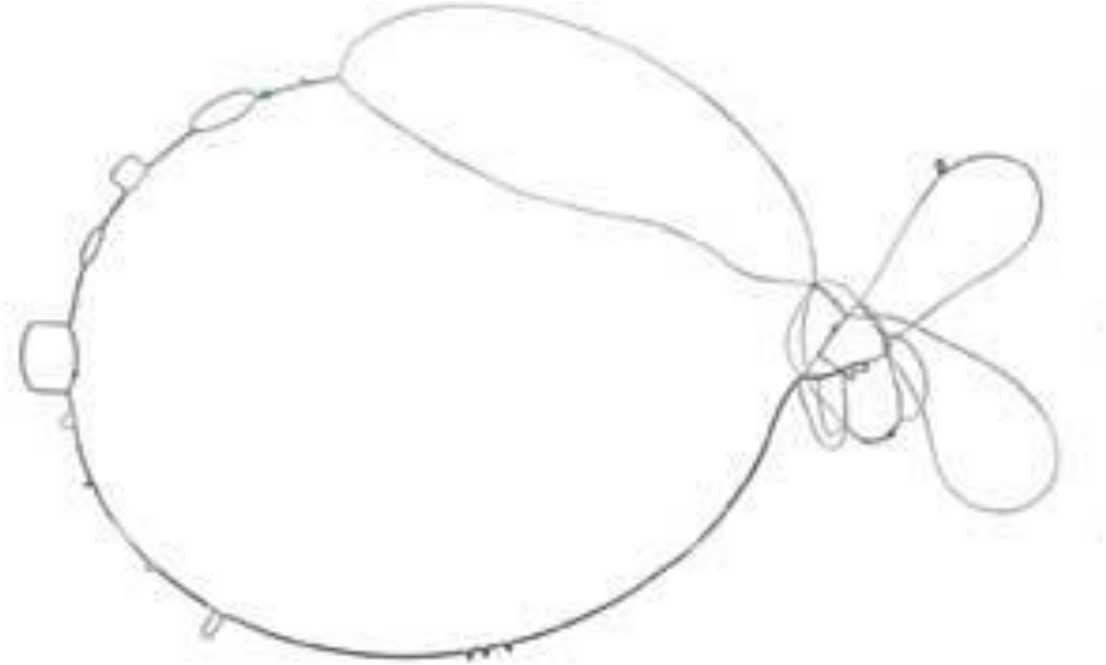
Genome assembly using graphs: overview



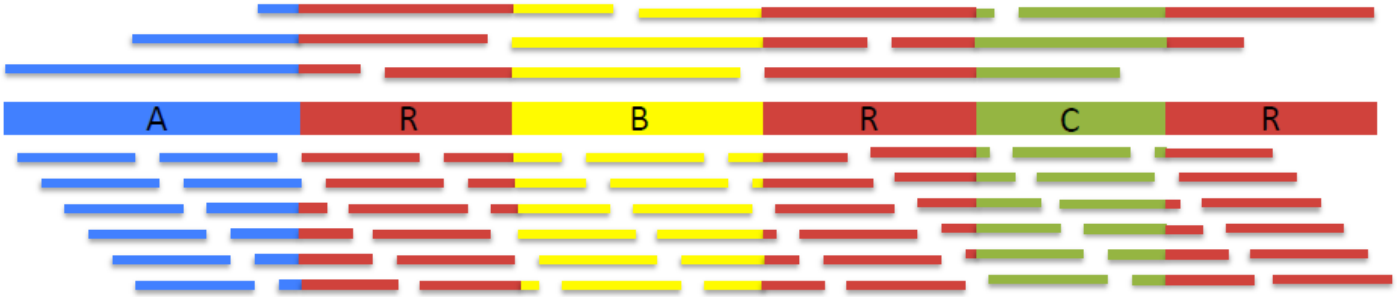
Assembly of contigs using graphs

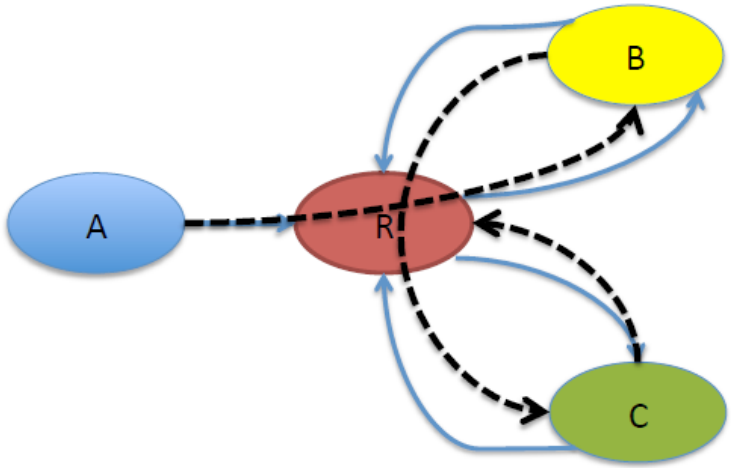
- Zhang *et al.* (1994). An algorithm based on graph theory for the assembly of contigs in physical mapping of DNA. Bioinformatics 10: 309–317
  - **"An algorithm is described for mapping DNA contigs based on an interval graph (IG) representation … CPU time is essentially linear with respect to the number of cosmids analyzed"**

Bacterial genome assembled using a de Bruijn graph



Assembly complexity





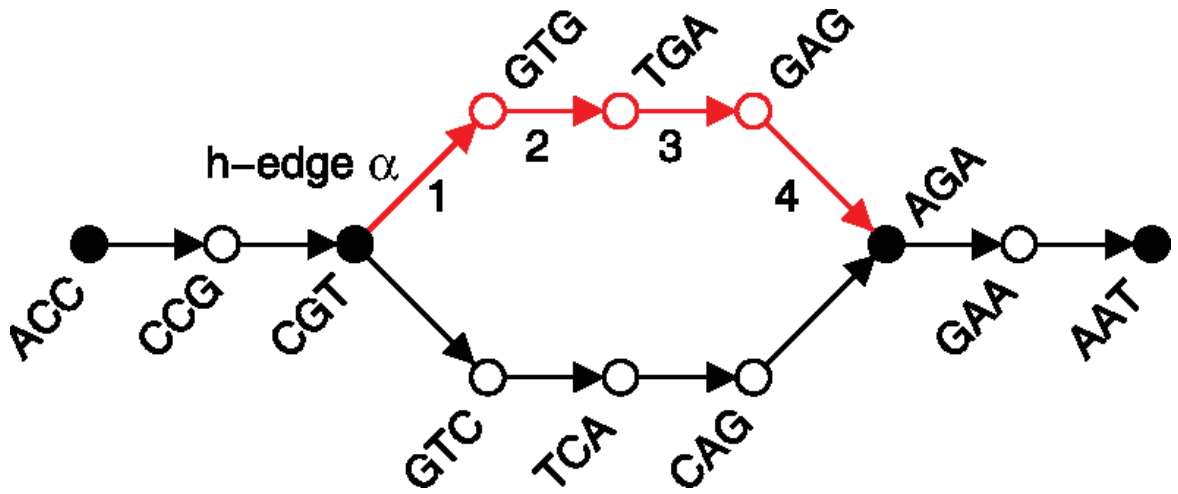
De Bruijn graphs: selection of k

- The choice of k is important to the construction of a de Bruijn graph
- smaller k results in more tangled graphs (more repeats will be glued)
  - Smaller k works better with low coverage regions
- larger k may not adequately detect overlaps, leading to fragmented graphs.
  - Larger k works better with high coverage regions

VelvetOptimiser: script written to optimise the k-mer size and coverage cutoff parameters for Velvet. https://github.com/tseemann/VelvetOptimiser

Multi-size de Bruijn graphs

Spades uses several values for *k* (manually set or inferred automatically) to create a ***multisized*** graph that minimized tangledness and fragmentation by combining various *k*-mers



SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing. Anton Bankevich et al Journal of Computational Biology 19, 2012 https://doi.org/10.1089/cmb.2012.0021

Assembly of long error-prone reads using de Bruijn graphs
Yu Lin, Jeffrey Yuan, Mikhail Kolmogorov, Max W. Shen, Mark Chaisson, and Pavel A. Pevzner
PNAS 2016 https://doi.org/10.1073/pnas.1604560113

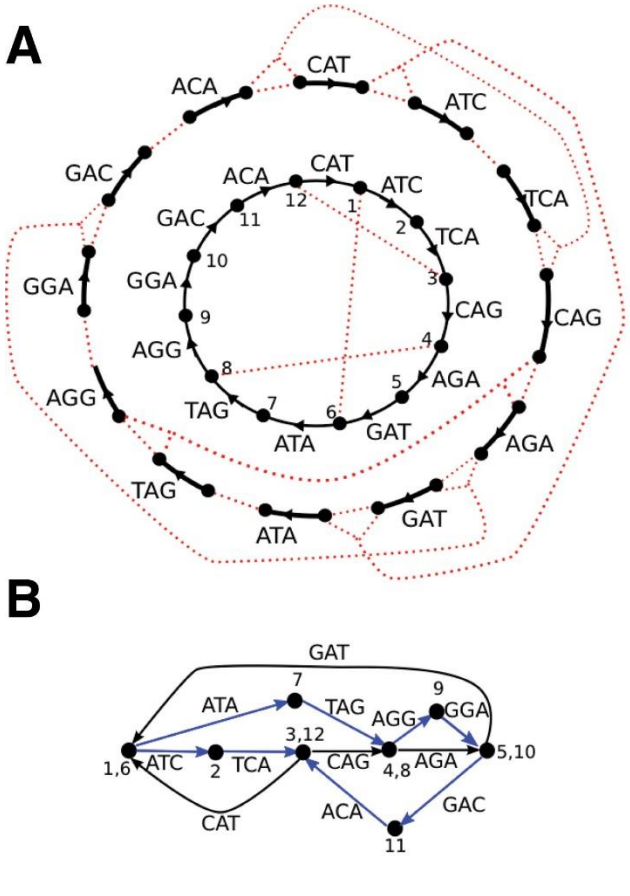
SPAdes: multi-sized de Bruijn graphs

Genome = CATCAGATAGGA

Reads (4-mers) = {ACAT, CATC, ATCA, TCAG, CAGA, AGAT, GATA, TAGG, GGAC}

Missing = {ATAG, AGGA, GACA}

Theoretical (3-mers) = {all}



Multisized de Bruijn graph. A circular Genome CATCAGATAGGA is covered by a set of Reads consisting of nine 4-mers, {ACAT, CATC, ATCA, TCAG, CAGA, AGAT, GATA, TAGG, GGAC}. Three out of 12 possible 4-mers from Genome are missing from Reads (namely {ATAG,AGGA,GACA}), but all 3-mers from the Genome are present in the Reads. (A) The outside circle shows a separate black edge for each 3-mer from Reads. Dotted red lines indicate vertices that will be glued. The inner circle shows the result of applying some of the glues. (B) The graph DB(Reads, 3) resulting from all the glues is tangled. The three h-paths of length 2 in this graph (shown in blue) correspond to h-reads ATAG, AGGA, and GACA. Thus $Reads_{3,4}$ contains all 4-mers from Genome. (C) The outside circle shows a separate edge for each of the nine 4-mer reads. The next inner circle shows the graph DB(Reads, 4), and the innermost circle represents the Genome. The graph DB(Reads, 4) is fragmented into 3 connected components. (D) The multisized de Bruijn graph DB (Reads, 3, 4). Figure and text from Bankevich *et al.* 2012.

Short read mapping using Suffix Trees/Arrays

Qué es un Suffix Array?

Consideremos una cadena a indexar: *"AGGAGC$" ($ = ultima posición)*

i	0	1	2	3	4	5	6
S[i]	A	G	G	A	G	C	$

Cadena indexada

List of suffixes

Suffix	I
AGGAGC$	0
GGAGC$	1
GAGC$	2
AGC$	3
GC$	4
C$	5
$	6

Ordered list of suffixes

Suffix	I
$	6
AGC$	3
AGGAGC$	0
C$	5
GAGC$	2
GC$	4
GGAGC$	1

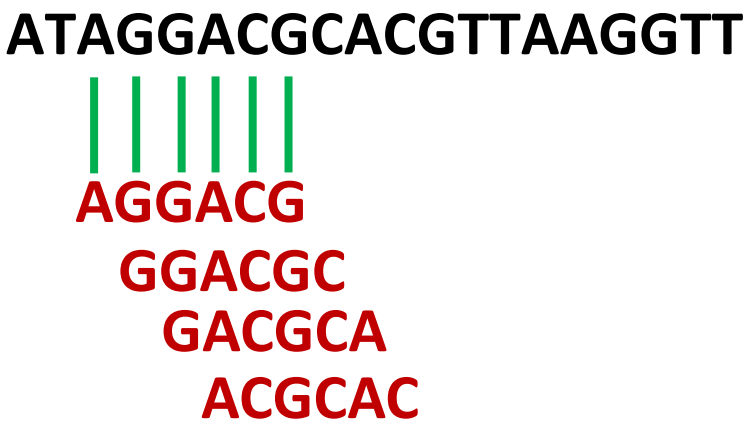
Suffix Array

i	A[i]
0	6
1	3
2	0
3	5
4	2
5	4
6	1

Cuáles son todos los sufijos que empiezan con AG?

Applications of suffix trees

Se acuerdan del “Exact string matching” (Naïve algorithm)?



Cómo aplicarían Suffix Arrays a este problema?

Suffix Trees/Arrays son la base algorítmica del programa BWA (Burrows-Wheeler Aligner, short read alignment)

Read mapping using hashing algorithm

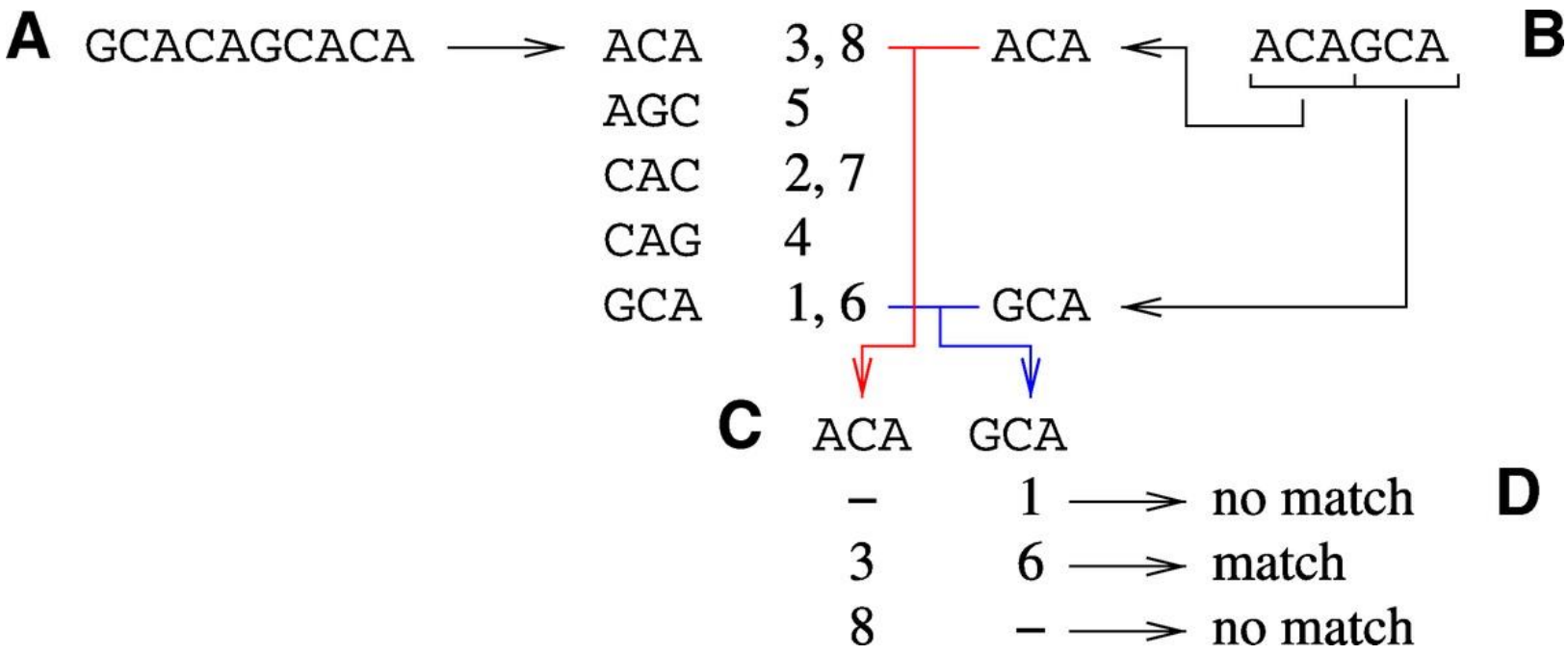
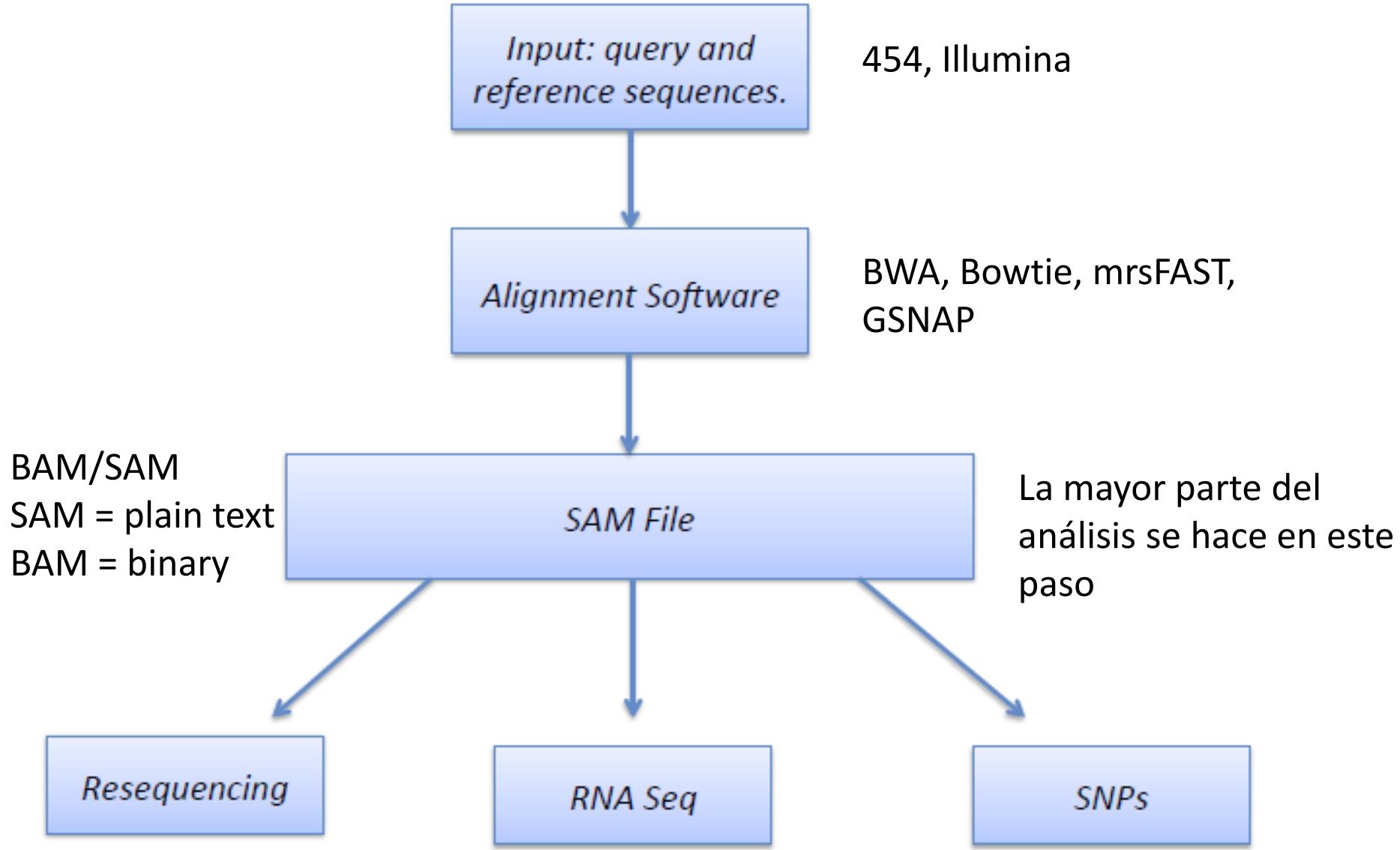


FIG. 1. The hashing algorithm. **(A)** The genome is cut into overlapping 3-mers, and their respective positions in the genome are stored. **(B)** The read is cut into 3-mers. The 3-mers from the reads are compared to 3-mers from the genome using a hashing procedure. **(C)** Positions for each seed are sorted and compared to the other seeds. **(D)** Compatible positions are kept.

Journal of Computational Biology. June 2012, 19(6): 796-813.
DOI: 10.1089/cmb.2012.0022

Short read mapping/alignment



SAM format

Contiene un header (opcional) y una sección de alineamientos

```
@HD     VN:1.0     SO:coordinate
@SQ     SN:1     LN:249250621  AS:NCBI37     UR:file:/data/local/ref/GATK/human_g1k_v37.fasta
        M5:1b22b98cdeb4a9304cb5d48026a85128
@SQ     SN:2     LN:243199373  AS:NCBI37     UR:file:/data/local/ref/GATK/human_g1k_v37.fasta
        M5:a0d9851da00400dec1098a9255ac712e
@SQ     SN:3     LN:198022430  AS:NCBI37     UR:file:/data/local/ref/GATK/human_g1k_v37.fasta
        M5:fdfd811849cc2fadebc929bb925902e5
@RG     ID:UM0098:1     PL:ILLUMINA     PU:HWUSI-EAS1707-615LHAAXX-L001     LB:80     DT:2010-05-05T20:00:00-0400     SM:SD37

1:497:R:-272+13M17D24M     113     1     497     37     37M     15     100338662     0
        CGGGTCTGACCTGAGGAGAACTGTGCTCCGCCTTCAG     0;==-==9;>>>>>=>>>>>>>>>>>>>=>>>>>>>>>>     XT:A:U     NM:i:0
        SM:i:37     AM:i:0     X0:i:1     X1:i:0     XM:i:0     XO:i:0     XG:i:0     MD:Z:37
19:20389:F:275+18M2D19M     99     1     17644     0     37M     =     17919     314
        TATGACTGCTAATAATACCTACACATGTTAGAACCAT     >>>>>>>>>>>>>>>>>>>><>>><<>>4:;>>:<9     RG:Z:UM0098:1
        XT:A:R     NM:i:0     SM:i:0     AM:i:0     X0:i:4     X1:i:0     XM:i:0     XO:i:0     XG:i:0
        MD:Z:37
19:20389:F:275+18M2D19M     147     1     17919     0     18M2D19M     =     17644     -314
        GTAGTACCAACTGTAAGTCCTTATCTTCATACTTTGT     ;44999;499<8<8<<<8<<><<<<<><7<;<<<><<     XT:A:R     NM:i:2
        SM:i:0     AM:i:0     X0:i:4     X1:i:0     XM:i:0     XO:i:1     XG:i:2     MD:Z:18^CA19
9:21597+10M2I25M:R:-209     83     1     21678     0     8M2I27M     =     21469     -244
        CACCACATCACATATACCAAGCCTGGCTGTGTCTTCT     <;9<<5><<<<><<<><><><><9><><>>9><><>     XT:A:R     NM:i:2
        SM:i:0     AM:i:0     X0:i:5     X1:i:0     XM:i:0     XO:i:1     XG:i:2     MD:Z:35
```

Current Read Mapping Tools

Table 2.

Global characteristics of the mapping tools

Tool	*Format*	*Algorithm*	*Threads*	*Gaps*	*Mismatches*
BWA	SAM	BWT	yes	yes	yes
Novoalign	SAM	hash the ref.	yes	yes	yes
Bowtie	SAM	BWT	yes	no	yes
SOAP2	perso	BWT	yes	no	at most 2
BFAST	SAM	hash the ref.	yes	yes	yes
SSAHA2	SAM	hash the ref.	no	no	yes
MPscan	perso	suffix tree	no	no	no
GASSST	SAM	hash the ref.	yes	yes	yes
PerM	SAM	hash the ref.	no	no	yes

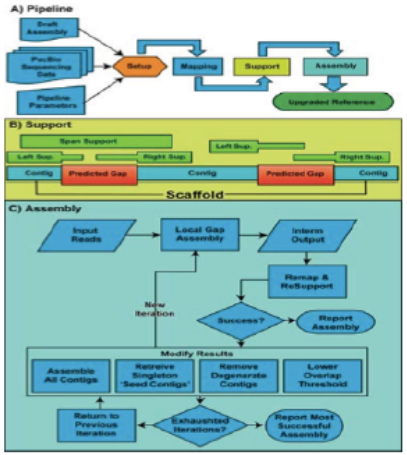
SAM, Sequence Alignments Map.

https://github.com/bwa-mem2/bwa-mem2

Schbath S, Martin V, Zytnicki M, Fayolle J, Loux V, Gibrat JF. Mapping reads on a genomic sequence: an algorithmic overview and a practical comparative analysis. J Comput Biol. 2012 19(6):796-813. doi: 10.1089/cmb.2012.0022. PMID: 22506536; PMCID: PMC3375638.

PacBio Assembly Algorithms

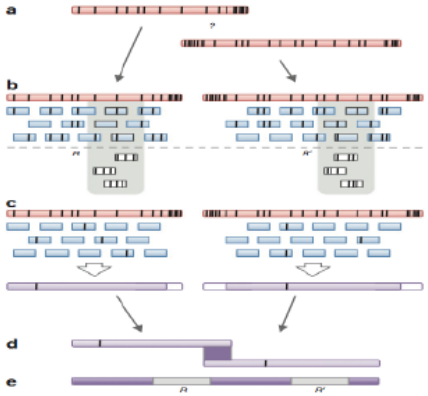
PBJelly



Gap Filling and Assembly Upgrade

English *et al* (2012)
PLOS One. 7(11): e47768

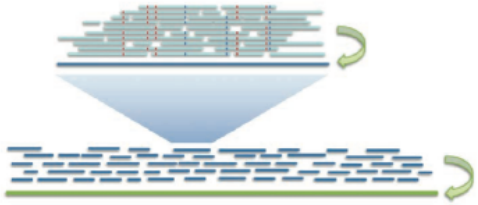
PacBioToCA & ECTools



Hybrid/PB-only Error Correction

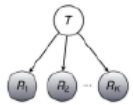
Koren, Schatz, *et al* (2012)
Nature Biotechnology. 30:693–700

HGAP & Quiver



$$\Pr(\mathbf{R} \mid T)$$

$$\Pr(\mathbf{R} \mid T) = \prod_k \Pr(R_k \mid T)$$



Quiver Performance Results
Comparison to Reference Genome
(*M. ruber*; 3.1 MB; SMRT® Cells)

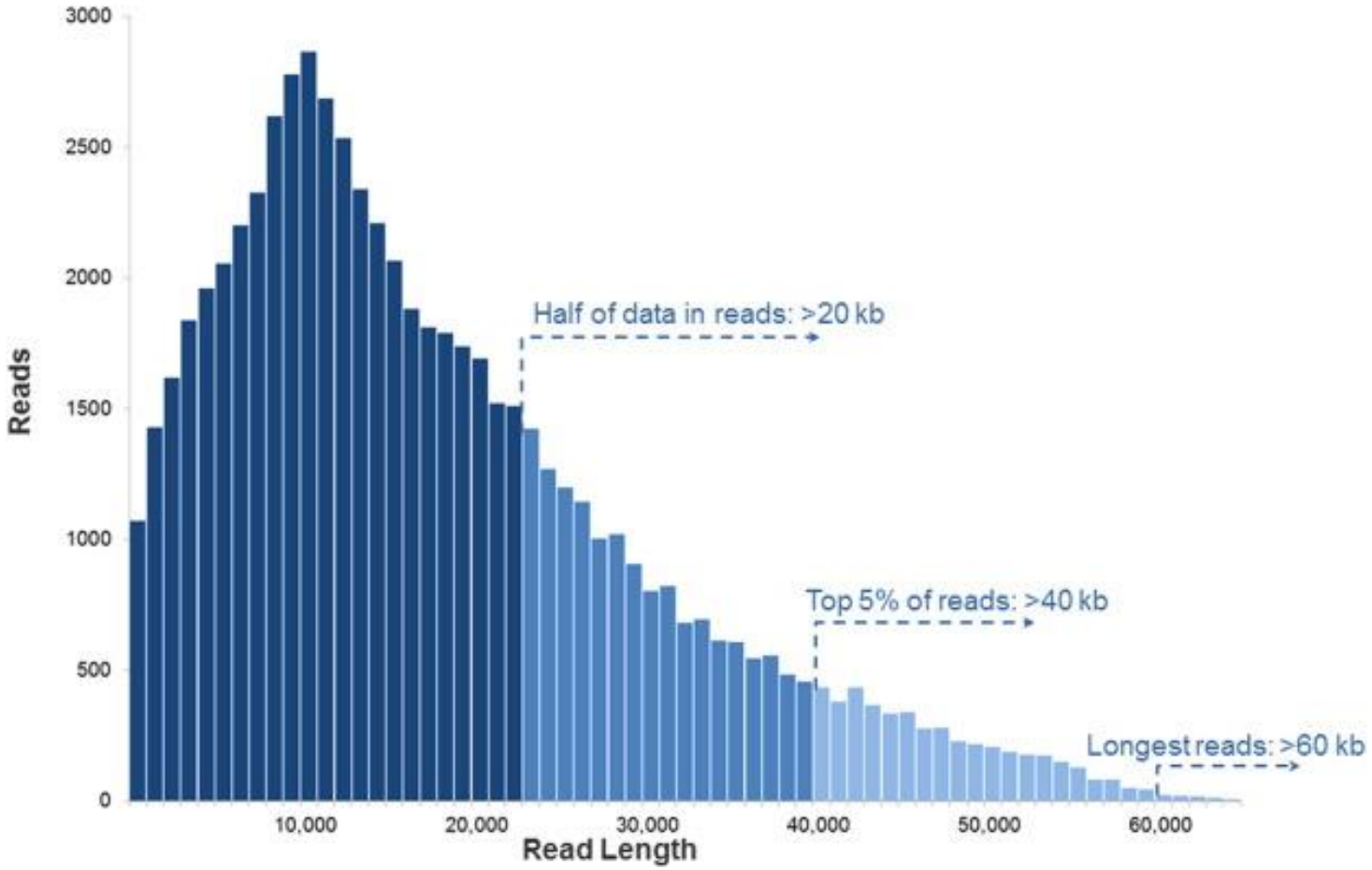
	Initial Assembly	Quiver Consensus
QV	43.4	54.5
Accuracy	99.99540%	99.99964%
Differences	141	11

PB-only Correction & Polishing

Chin *et al* (2013)
Nature Methods. 10:563–569

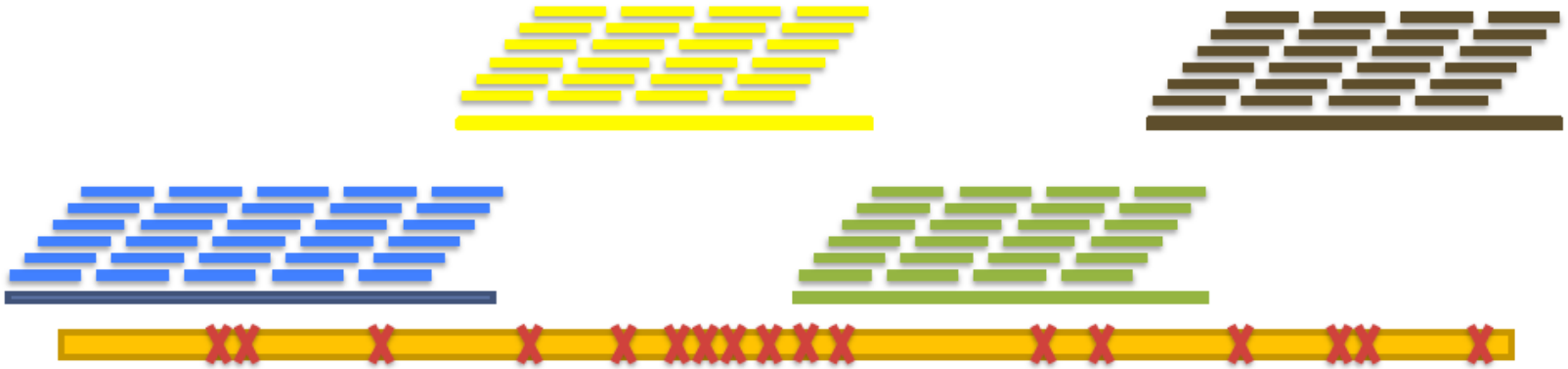
< 5x PacBio Coverage > 50x

PacBio


3000
2500
2000
1500
1000
500
0
Reads
Half of data in reads: >20 kb
Top 5% of reads: >40 kb
Longest reads: >60 kb
10,000
20,000
30,000
40,000
50,000
60,000
Read Length

ECTools: Error Correction with pre-assembled reads

https://github.com/jgurtowski/ectools



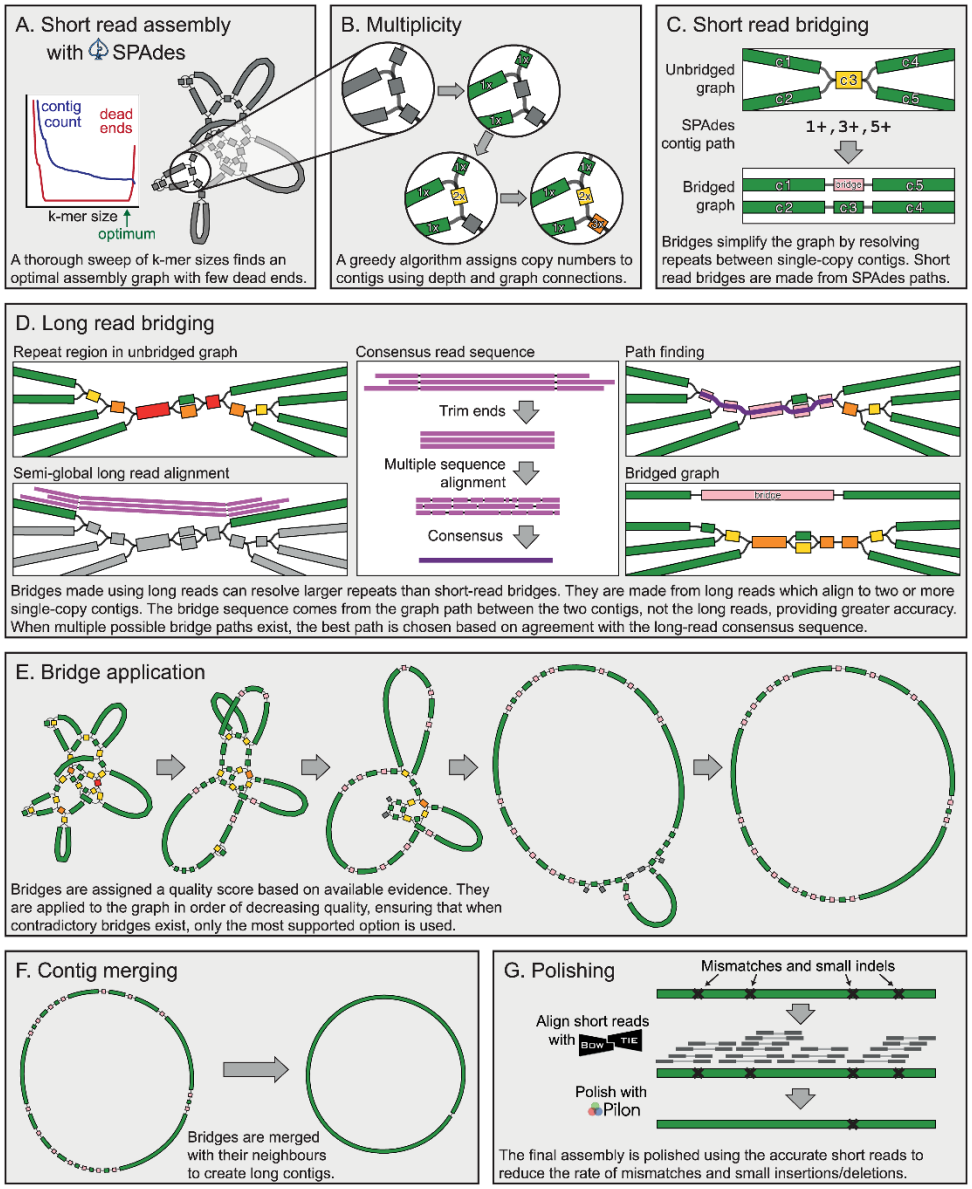
Short Reads -> Assemble Unitigs -> Align & Select - > Error Correct

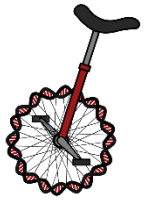
Can Help us overcome:

1. Error Dense Regions – Longer sequences have more seeds to match
2. Simple Repeats – Longer sequences easier to resolve

However, cannot overcome Illumina coverage gaps & other biases

Hybrid assemblies: short + long reads





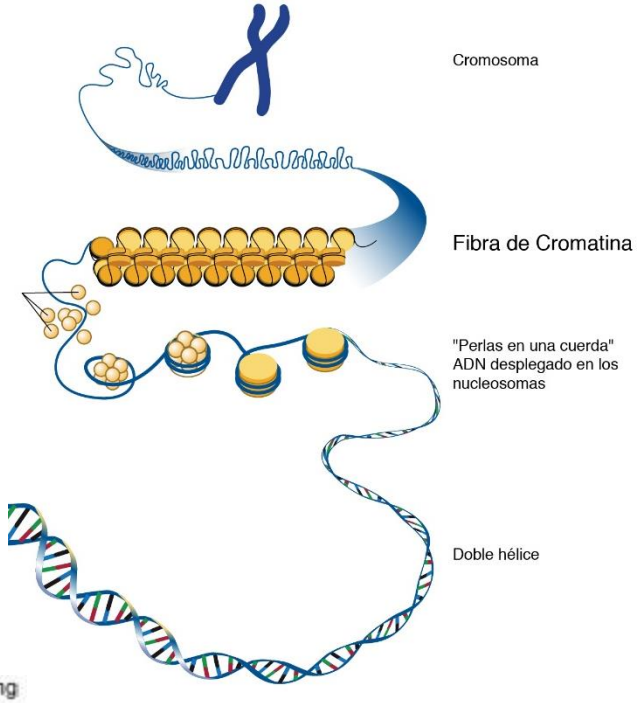
Unicycler

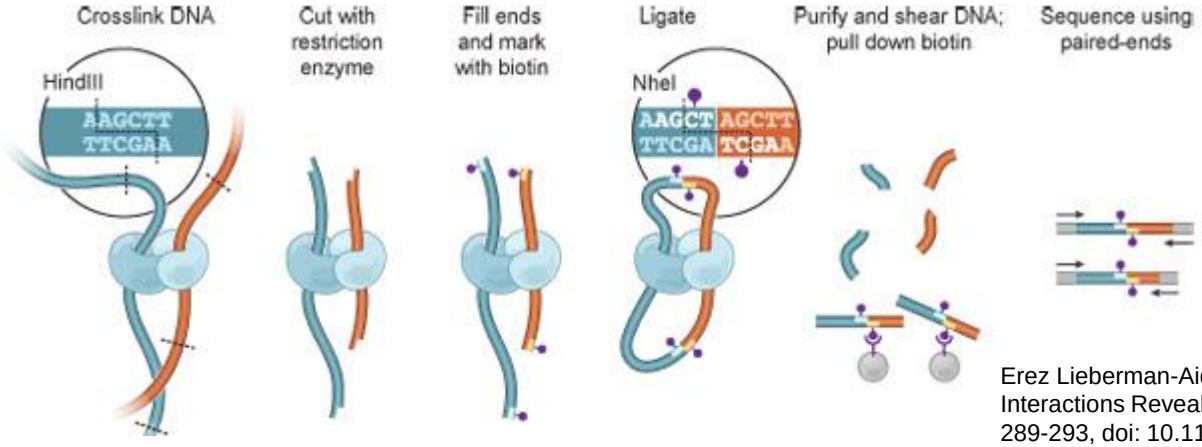
- Unicycler is designed specifically for ***hybrid assembly*** (that is, using both short- and long-read sequencing data) of small (e.g., bacterial, viral, organellar) genomes.
- Unicycler employs a multi-step process that utilizes a number of software tools

Ultimo hito: estudio de estructura cromosómica

Hi-C: proximity ligation + secuenciación

- Estudio *no sesgado* a escala genómica de interacciones a nivel de la cromatina
- Revela arquitectura cromosómica a diferentes niveles
  - **Territorios cromosómicos**
  - **Regiones donde la cromatina es abierta vs cerrada**
  - **Estructura de la cromatina a escala de megabases (millones de bases)**



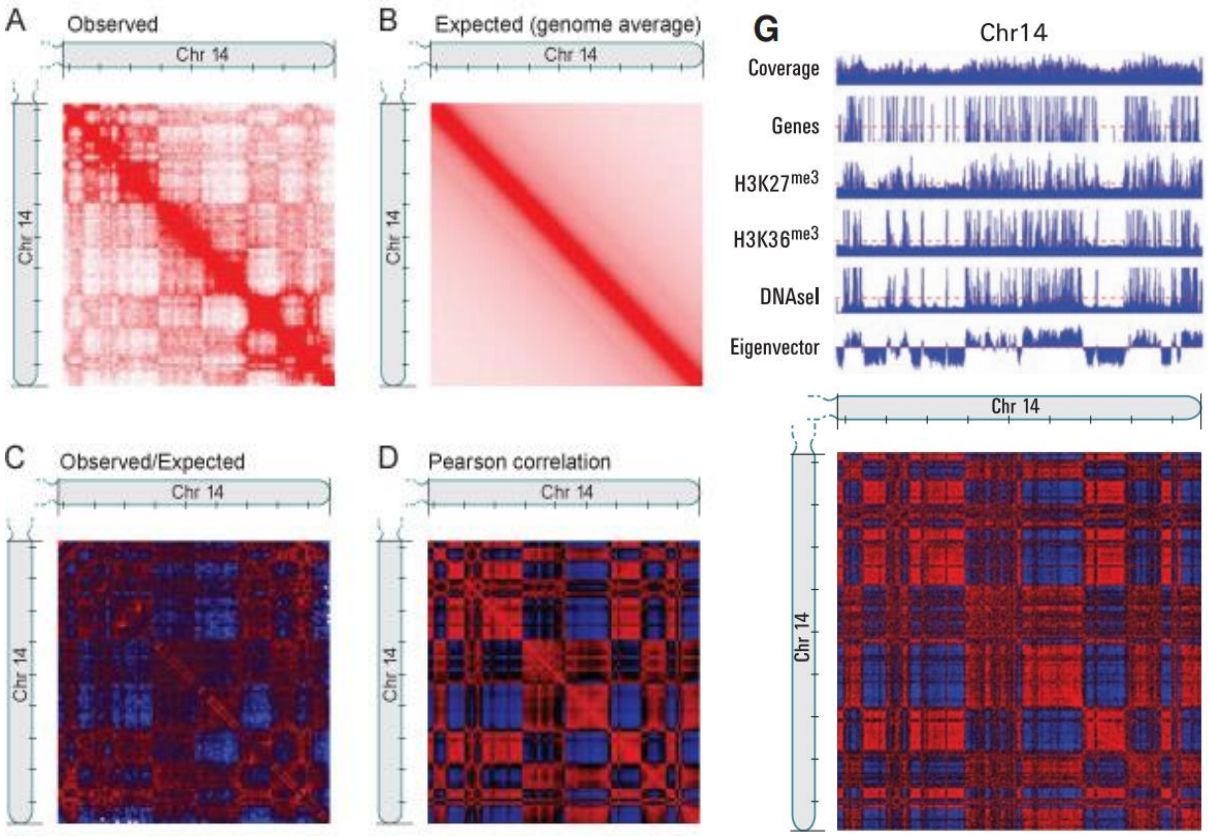


Erez Lieberman-Aiden E, et al. Comprehensive Mapping of Long-Range Interactions Reveals Folding Principles of the Human Genome. Science (326), 289-293, doi: 10.1126/science.1181369 (2009).
van Berkum, NL, et al. Hi-C: A Method to Study the Three-dimensional Architecture of Genomes. J. Vis. Exp. (39), e1869, doi:10.3791/1869 (2010).

Hi-C: proximity ligation + sequencing

Hi-C revela interacciones *intercromosómicas* + *intracromosómicas*

Ayudan a mejorar ensambles: corrijen mala asignación de contigs a scaffolds o cromosomas), además de identificar inversions y translocaciones, en estudios comparativos.



Erez Lieberman-Aiden E, et al. Comprehensive Mapping of Long-Range Interactions Reveals Folding Principles of the Human Genome. Science (326), 289-293, doi: 10.1126/science.1181369 (2009).

Assessment / Validation

ASSEMBLY VALIDATION

Métricas para evaluar assemblies

N50

- Calidad del ensamble en términos de ***contiguidad***
- ***N50*** **es similar a una mediana o media de longitudes de contigs**
- ***Es la longitud del contig más corto*** **a partir del cual el 50% de las bases se encuentran repartidas entre él mismo y otros contigs más cortos**
- **Ej si tenemos 7 contigs con longitudes**
  - **1, 1, 3, 5, 8, 12, 20**
- **El N50 es 12 porque:**
  - **1 + 1 + 3 + 5 + 8 + 12 + 20 = 50 (la longitud acumulada de todo el ensamble)**
  - **50/2 = 25 (la mitad de la longitud sumada de todo el ensamble)**
  - **Y si empezamos desde el contig más corto y vamos sumando hasta conseguir llegar a una longitud acumulada >=25 …**
  - **1 + 1 + 3 + 5 + 8 + 12 = 30**
  - **O sea, el sexto contig (de longitud 12) es el primero en el que alcanzamos o pasamos la mitad de la longitud del ensamble**

Métricas para evaluar assemblies

L50

- El *número mínimo de contigs* cuya longitud suma 50% del tamaño del ensamble
- Ej si tenemos 7 contigs con longitudes
  - 1, 1, 3, 5, 8, 12, 20
  - El L50 es 6 porque seis es el número de contigs con los que alcanzamos o pasamos la mitad de la longitud del ensamble

N90

- Similar al N50 (pero pide 90% de las bases)

NG50

- Similar al N50 pero en lugar de referirse a la longitud total del ensamble, se refiere a la longitud total del *genoma*
- *Util porque el N50 no permite comparar entre ensambles de diferentes tamaños (pero NG50 si)*

Y hay más métricas …. Ver https://en.wikipedia.org/wiki/N50,_L50,_and_related_statistics

Métricas para evaluar ensambles

Una métrica es solamente eso. Una herramienta.

Usarla con cuidado!

PROBLEMAS con el N50!

Si intentamos optimizar el N50 podemos forzar (recompensar) malos ensambles

- Un assembler agresivo puede excederse al unir contigs simplemente buscando incrementar el N50
  - **Ej 1, 1, 3, 5, 8, 12, 20 (contigs del ejemplo anterior, N50 = 12)**
  - **1, 1, 3, 5, 8, 20, 20 (aggressive join de los contigs de longitudes 8 y 12)**
  - **Ahora el N50 es 20**

Validación de los ensambles

Auto-consistencia

- Mapear de nuevo reads contra contigs
- Chequear errores o inconsistencias

Segunda opinion / validación externa

- Usar dos métodos de secuenciación complementarios
  - Illumina + PacBio
  - Illumina + Nanopore
- Validar regiones por PCR
  - Util para validar o para resolver regiones dificiles
- Hi-C (chromatin contact maps)
  - Hi-C, 3-C Seq, Capture-C
  - Familia de métodos para caracterizar interacciones a nivel de cromatina
  - Mapas de regiones del genoma que están cercanas entre si
- Mapa óptico global del genoma
  - https://en.wikipedia.org/wiki/Optical_mapping

Material de lectura

- J. Setubal and J. Meidanis, Introduction to Computational Molecular Biology, PWS Publishing Company, Boston, 1997
- D. Gusfield, Algorithms on Strings, Trees and Sequences, Cambridge Universtiy Press, 1997.
- Compeau PEC, Pevzner PA, Tesler G. How to apply de Bruijn graphs to genome assembly. Nature Biotechnol 29: 987, 2011.
- Li H, Holmer N. A survey of sequence alignment algorithms for next-generation sequencing. Briefings in Bioinformatics 11: 473, 2010.
- Riberiro FJ et al. Finished bacterial genomes from shotgun sequence data. Genome Res 22: 2270, 2012
- Nagarajan, N., & Pop, M. (2013). Sequence assembly demystified. Nature Reviews Genetics, 14(3), 157–167. doi:10.1038/nrg3367
- Rice, E. S., & Green, R. E. (2018). New Approaches for Genome Assembly and Scaffolding. Annual Review of Animal Biosciences, 7(1). doi:10.1146/annurev-animal-020518-115344