BCGES short course, session 1, introduction

Vincent Plagnol

UCL Genetics Institute

Some technical details about the class

- Co-taught between myself and Taane Clark from the LSHTM.
- My slides and practicals are available on Github.
  - This is a collaborative editing side, and I strongly recommend becoming familiar with it.
- The practicals all use linux, and some familiarity with the command line will help without being necessary.
  - Sections 1-3 of this manual should be all you need.
  - If you struggle with command lines, text editing... spend some extra time this evening to go through this.
- Several practicals use the programming language `R`
  - Again, some familiarity would allow you to get more out of the course.

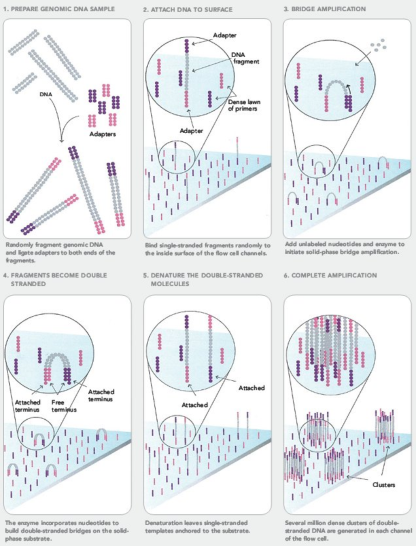
Key concepts/tools

- Introduction to vocabulary, concepts
- Fastq format
- BAM and CRAM format
- Using the Galaxy server for basic manipulations

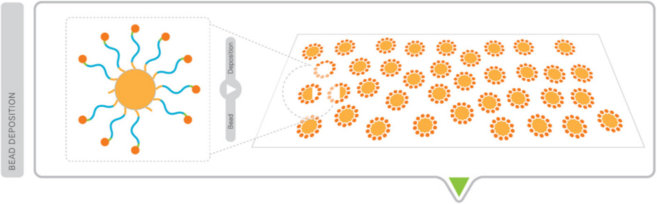
Outline

Outline

The Illumina principle...



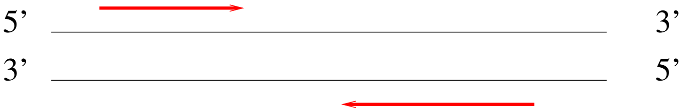
...is pretty much the same as the Solid idea



Stoechiometry is set such that each bead clonaly amplifies a single DNA molecule.

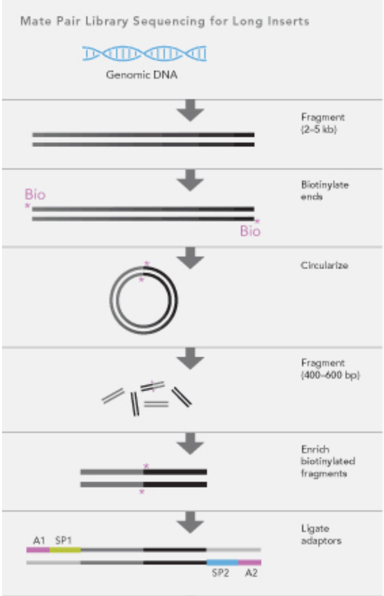
Outline

Illumina paired-end reads



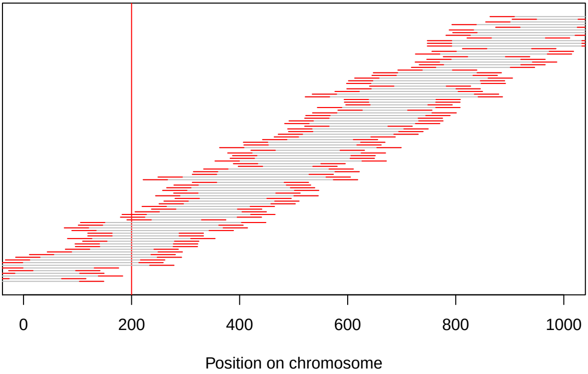
Illumina technology always reads from the 5' end to the 3' end. The first read can be either of the two but the direction is set, and in the absence of something unusual both reads point toward each other and may overlap in the middle.

Mate-pair reads



Mate Pair library preparation is designed to generate short fragments that consist of two segments that originally had a separation of several kilobases in the genome. Fragments of sample genomic DNA are end-biotinylated to tag the eventual mate pair segments. Self-circularization and refragmentation of these large fragments generates a population of small fragments, some of which contain both mate pair segments with no intervening sequence. These Mate Pair fragments are enriched using their biotin tag. Mate Pairs are sequenced using a similar two-adapter strategy as described for paired-end sequencing.

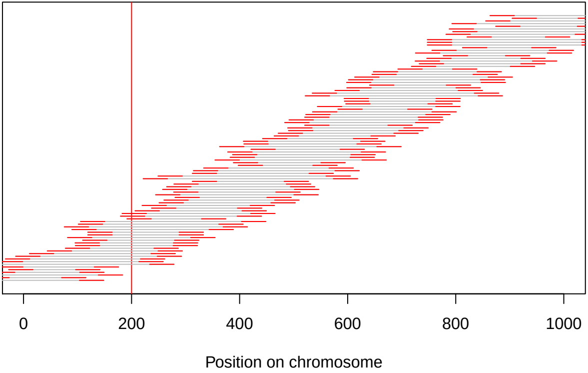
Read depth



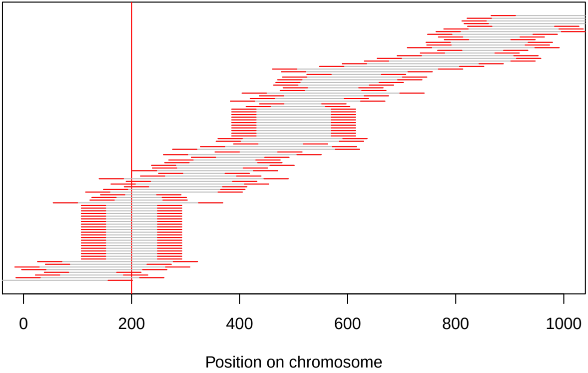
Read depth is not a good summary of data quality

- If the depth is uneven, a high mean read depth may not be informative.
- Sequence capture technologies introduce a lot of variability in read depth.
- So a 30x read depth is to a large extent useful to make sure that a large fraction of the target region (say 90% of the exome) is covered with at least 10x.
- The mean read depth may not need to be as large for a full genome (because of the absence of capture step).

The read clonality problem



The read clonality problem



Sample multiplexing (1)

- A full sequencing run is often too much for a single sample.
- One needs to find ways to split the sequencing across samples.
- The first option is physical separation on the sequencing slide:
  - The Illumina flow cell has 8 lanes so 8 plex sequencing is obvious.
  - Physical separation wastes some space on the slide.
- Better: introduce next to the adapter a sample specific barcode (6 bp for Illumina).
- Then assign each read to the sample it originated from.

Sample multiplexing (2)

- Sample multiplexing has been discussed for a long time now.
- In my opinion, it still is not completely solved:
  - Standard kits use 12 plex sequencing.
  - But we are often interested in much higher multiplexing level (100s or more).
  - A few companies can now provide this type of service (Fluidigm in particular).
- When the multiplexing level is high, the cost of library preparation becomes an issue.

Sequence capture (1)

- We also typically want to target a genomic region, or the exome, rather than full genome sequencing.
- This requires enriching the DNA for a specific target region, a process known as sequence capture.
- An alternative to sequence capture is PCR based amplification but it creates artifacts of its own.
  - It is however very efficient at generating a very high read depth.
  - It does prevent us from identifying duplicate reads properly.
- Two main companies are competing for sequence capture products: Nimblegen and Illumina.

Sequence capture (2)

- Sequence capture can be achieved in different ways.
- First generation used probes on a solid surface.
- More recent assays used probes in solution, and it seems to give more consistent and higher quality results.
- Many parameters affect capture efficiency:
  - Quality of the DNA.
  - Size of the target region.
  - GC content.

Cloud computing

- Remote servers are now available to perform some of these tasks, like the Galaxy server
- A true solution for cloud computing is the Amazon server.
- Amazon provides an incredibly flexible solution, probably more flexible than Galaxy.
- Of course there are costs, see for example *Biomedical Cloud Computing With Amazon Web Services*, PLoS Computational biology, 2011.
  - Aligning a full human genome cost $XXX according to their computations (your guess).
  - See the answer here.

Outline

Why do we need a fastq format? (1)

- Best reference: Wikipedia page on fastq.
- This is the simplest and most generic flat text format to store sequencing reads of arbitrary size.
  - Store the most likely call and a quality associated to it.
- A weakness: the second most likely call is not stored.
- It is a flat text file so very easy to share across platforms and software.

Why do we need a fastq format? (2)

- A typical exome dataset: 43,406,971 reads, each 76bp long.
  - That is 3.3 billion bp, hence the size of the human genome,
- Each base pair has a quality associated to it, called Phred score.
- Storing a number, even an integer is not efficient.
  - Typical int format is stored on 32 bits (hence numbers capped by 2^{32}.
  - Also we need a text format, more reliable and portable.
- The best solution is to store qualities as characters, mapping each ASCII character to its associated number.

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

Source: www.LookupTables.com

What is a Phred quality meant to be?

- In an ideal world a quality of x means a probability $10^{-x/10}$ that the call is wrong.
- So a completely random call means 75% chances to be wrong.
- This more or less matches a minimum score of 2 as $10^{-0.2}$ is equal to 0.63.
- In practice Phred scores are usually poorly calibrated so the interpretation is not straightforward.
  - See this blog post for example.

What does the maximum Phred quality mean?

- Phred scores are typically capped at 40.
- It represents a best case scenario of 1/10,000 error rate.
- Indeed, of the error was introduced at the PCR stage, the scanner will not capture this information and can give a perfect quality.
- As a consequence, the maximum Phred score is a measure of the accuracy of the library preparation, more than the sequencing itself.

Several flavors of fastq formats

- The Sanger fastq: qualities = ASCII code - 33
- The Illumina fastq: qualities = ASCII code - 64
- The latest Illumina fastq (CASAVA 1.8): qualities = ASCII code - 33
  - This format adds other technical refinements, including a Y/N flag for each read (Y means failed QC).
  - Best is to look at some examples and see the differences.

Splitting the fastq files into smaller chunks

- The standard Illumina pipeline splits the fastq into a large number of smaller fastq files.

```
[vincentplagnol@ugi-151040 Sergey_Illumina]$ ls -ltrh CamFid_039FQ_GCCAAT_L004_R*
-rwxrwxrwx 1 vincentplagnol users 358M Feb  9  2012 CamFid_039FQ_GCCAAT_L004_R1_008.fastq.gz
-rwxrwxrwx 1 vincentplagnol users 345M Feb  9  2012 CamFid_039FQ_GCCAAT_L004_R1_001.fastq.gz
-rwxrwxrwx 1 vincentplagnol users 356M Feb  9  2012 CamFid_039FQ_GCCAAT_L004_R1_002.fastq.gz
-rwxrwxrwx 1 vincentplagnol users 357M Feb  9  2012 CamFid_039FQ_GCCAAT_L004_R1_003.fastq.gz
-rwxrwxrwx 1 vincentplagnol users 355M Feb  9  2012 CamFid_039FQ_GCCAAT_L004_R1_004.fastq.gz
-rwxrwxrwx 1 vincentplagnol users 361M Feb  9  2012 CamFid_039FQ_GCCAAT_L004_R1_005.fastq.gz
-rwxrwxrwx 1 vincentplagnol users 348M Feb  9  2012 CamFid_039FQ_GCCAAT_L004_R1_006.fastq.gz
-rwxrwxrwx 1 vincentplagnol users 354M Feb  9  2012 CamFid_039FQ_GCCAAT_L004_R1_007.fastq.gz
-rwxrwxrwx 1 vincentplagnol users 355M Feb  9  2012 CamFid_039FQ_GCCAAT_L004_R1_009.fastq.gz
-rwxrwxrwx 1 vincentplagnol users 206M Feb  9  2012 CamFid_039FQ_GCCAAT_L004_R1_010.fastq.gz
-rwxrwxrwx 1 vincentplagnol users 340M Feb  9  2012 CamFid_039FQ_GCCAAT_L004_R2_001.fastq.gz
-rwxrwxrwx 1 vincentplagnol users 346M Feb  9  2012 CamFid_039FQ_GCCAAT_L004_R2_002.fastq.gz
-rwxrwxrwx 1 vincentplagnol users 345M Feb  9  2012 CamFid_039FQ_GCCAAT_L004_R2_003.fastq.gz
-rwxrwxrwx 1 vincentplagnol users 343M Feb  9  2012 CamFid_039FQ_GCCAAT_L004_R2_004.fastq.gz
-rwxrwxrwx 1 vincentplagnol users 347M Feb  9  2012 CamFid_039FQ_GCCAAT_L004_R2_005.fastq.gz
-rwxrwxrwx 1 vincentplagnol users 340M Feb  9  2012 CamFid_039FQ_GCCAAT_L004_R2_006.fastq.gz
-rwxrwxrwx 1 vincentplagnol users 344M Feb  9  2012 CamFid_039FQ_GCCAAT_L004_R2_007.fastq.gz
-rwxrwxrwx 1 vincentplagnol users 345M Feb  9  2012 CamFid_039FQ_GCCAAT_L004_R2_008.fastq.gz
-rwxrwxrwx 1 vincentplagnol users 341M Feb  9  2012 CamFid_039FQ_GCCAAT_L004_R2_009.fastq.gz
-rwxrwxrwx 1 vincentplagnol users 195M Feb  9  2012 CamFid_039FQ_GCCAAT_L004_R2_010.fastq.gz
```

- This is apparently useful for the standard CASAVA pipeline that comes with Illumina instruments.

QC tool for fastq file

- A researcher at the Babraham has put together some tools to check that a fastq file is OK.
- See the webpage: FastQC
- It does some useful checks for base quality, over-representation of k-mers...
- In case some issue arises, it is a good tool to use and straightforward to run.

Storage considerations

- How much a fastq file is of course a function of what is being sequenced.
- A standard human exome (38 Mb capture, 30x read depth) will require roughly 8 Gb fastq file.
- Compression is key: this storage requirement goes down to 3 Gb after using bzip2 compression.
    - Note that bzip2 compression is more effective than gzip.
- 1 Tb of data can store roughly 300 human exomes.
- Full human sequence is another challenge (easily 200 Gb per sample).

How Illumina quality control is set

- To remove the least reliable data from the analysis results, often derived from overlapping clusters, raw data is filtered to remove any reads that do not meet the overall quality as measured by the Illumina chastity filter.
- The chastity of a base call is calculated as the ratio of the brightest intensity divided by the sum of the brightest and second brightest intensities.
- Clusters pass filter if no more than one base call in the first 25 cycles has a chastity of > 0.6.
- Remaining cycles are ignored.
- More information on the Illumina support page: Support page

Outline

Why do we need a BAM format?

- As you will discuss extensively in the rest of this class, we typically want to map the short reads to a known reference genome.
- The BAM format is a binary format for storing aligned sequence data
  - Which means that it stores the reads plus the alignment position.
- BAM is the binary (compressed) version, and an equivalent uncompressed format exists (SAM format).

The BAM format is sorted and indexed

- A typical use of a BAM file is to extract information about a specific slice of sequence, not the whole genome or exome.
- It is therefore key to be able to access these slices very rapidly, without having to go through the whole file.
- To this end, BAM files can be sorted and indexed, which allows constant time access to any fraction of the BAM file.
- Best place to learn is the SAM/BAM reference file.

The choice of the reference sequence matters

- A BAM file is dependent on the sequence it was aligned against.
- This information is encoded in the header of the BAM file.
- It is useful to get used to reading the headers of BAM files.
  - This can be done using `samtools view -H`.
  - All the information about what has happened to the BAM file can be found in the headers, as well as the reference genome.

Basic `samtools` manipulation

- It is a good idea to read through the samtools manual.
- With the recent release of `samtools` v1.0, you should transition toward this location to update `samtools`.
- Most tools have sophisticated options but `samtools view` in particular can do useful thing:
  - Subset a specific gene/region/chromosome, potentially over the web.
  - Request specific flags for the reads.

The flags associated with reads in BAM files

Flag	Chr	Description
0x0001	p	the read is paired in sequencing
0x0002	P	the read is mapped in a proper pair
0x0004	u	the query sequence itself is unmapped
0x0008	U	the mate is unmapped
0x0010	r	strand of the query (1 for reverse)
0x0020	R	strand of the mate
0x0040	1	the read is the first read in a pair
0x0080	2	the read is the second read in a pair
0x0100	s	the alignment is not primary
0x0200	f	the read fails platform/vendor quality checks
0x0400	d	the read is either a PCR or an optical duplicate

From the `samtools` manual

Even more compression with CRAM

- The EBI has a technical page about the CRAM format.
- Here is the news release, with full availability on June 2013.
- The very recently released version of `samtools` fully incorporates the BAM format.
- See this paper for more information on the CRAM format.

Outline

Galaxy servers

- The main server of Galaxy is located here: http://main.g2.bx.psu.edu.
- But other instances exist.
- Various Galaxy servers deliver slightly different packages and options.
- It is even possible to install your own server if you found it useful.
- There are also options for data sharing, and sharing analysis protocols which are really useful.
  - The idea is to make the bioinformatic analysis of sequence data more reproducible.