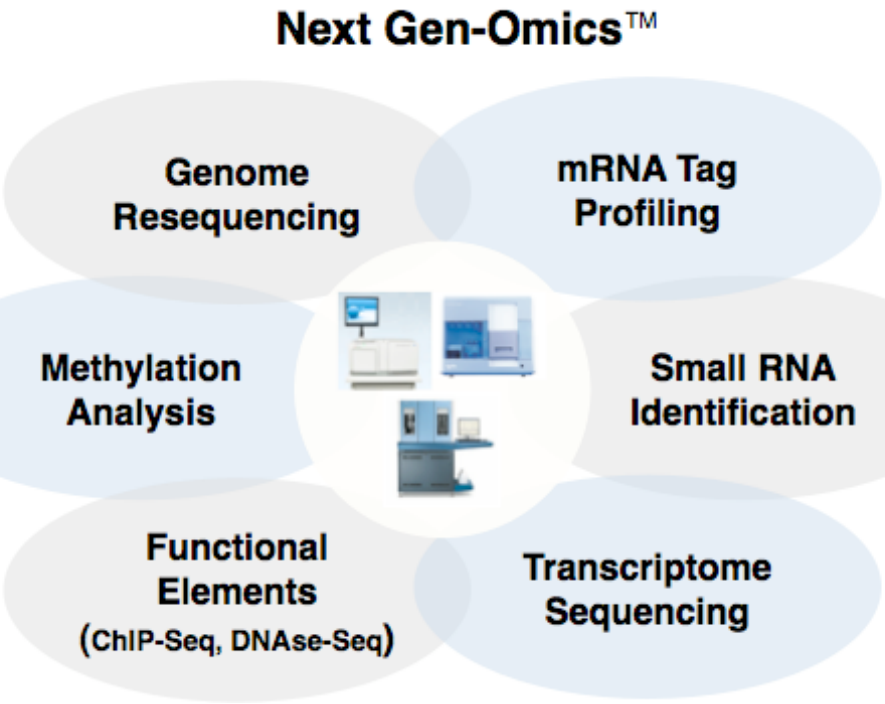
Looking at raw NGS data

O. Harismendy, PhD

MED263 – W2017

NGS is ubiquitous

- Mutations
- Structural variations
- *De novo* sequencing
- Targeted sequencing



- Silenced DNA
- Imprinted regions

- Novel RNA discovery
- Digital transcriptome
- New isoforms

- Location of proteins on DNA
- Level of Occupancy

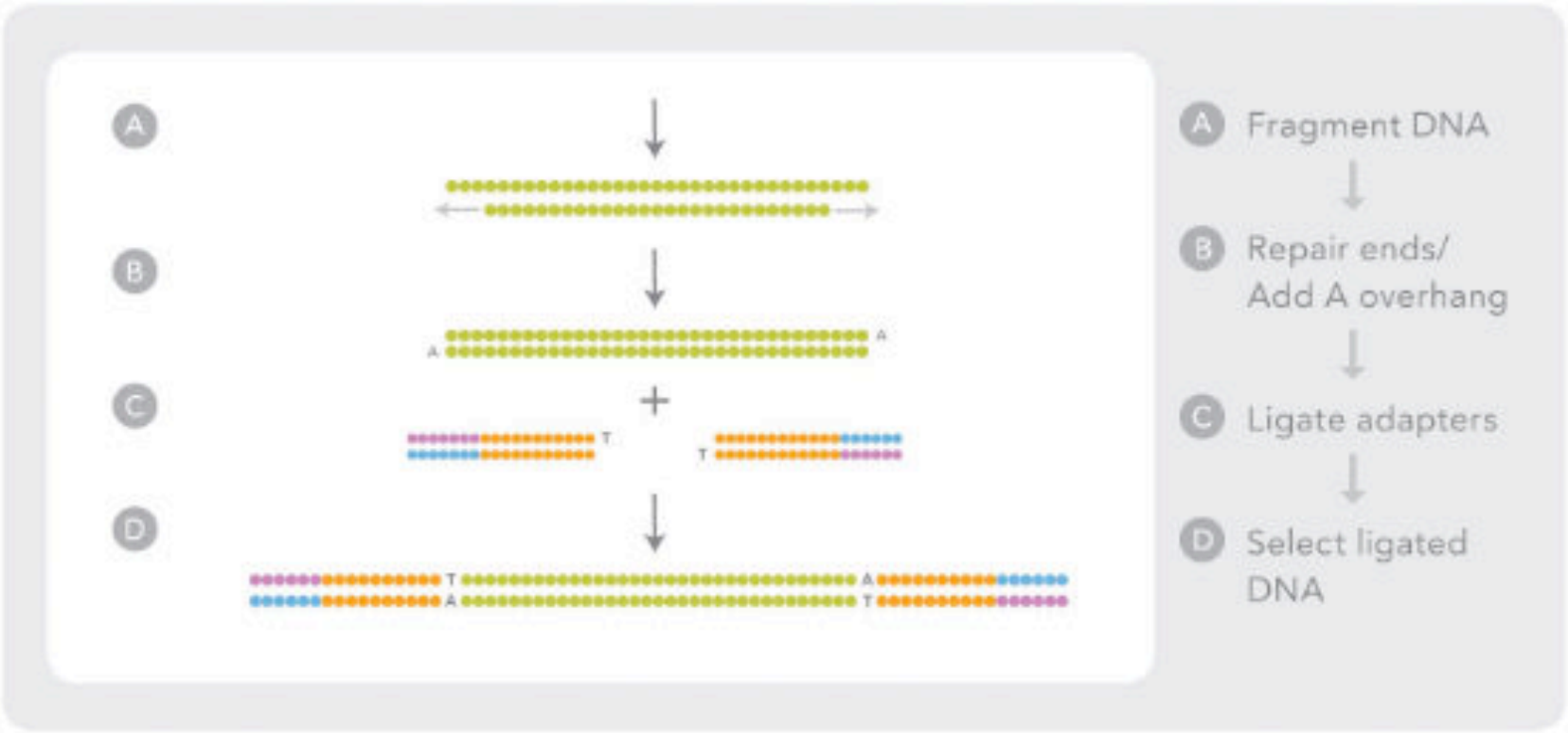
Sequencing is providing a universal read out for many biological experiments

Illumina Technology

SEQUENCING BY SYNTHESIS

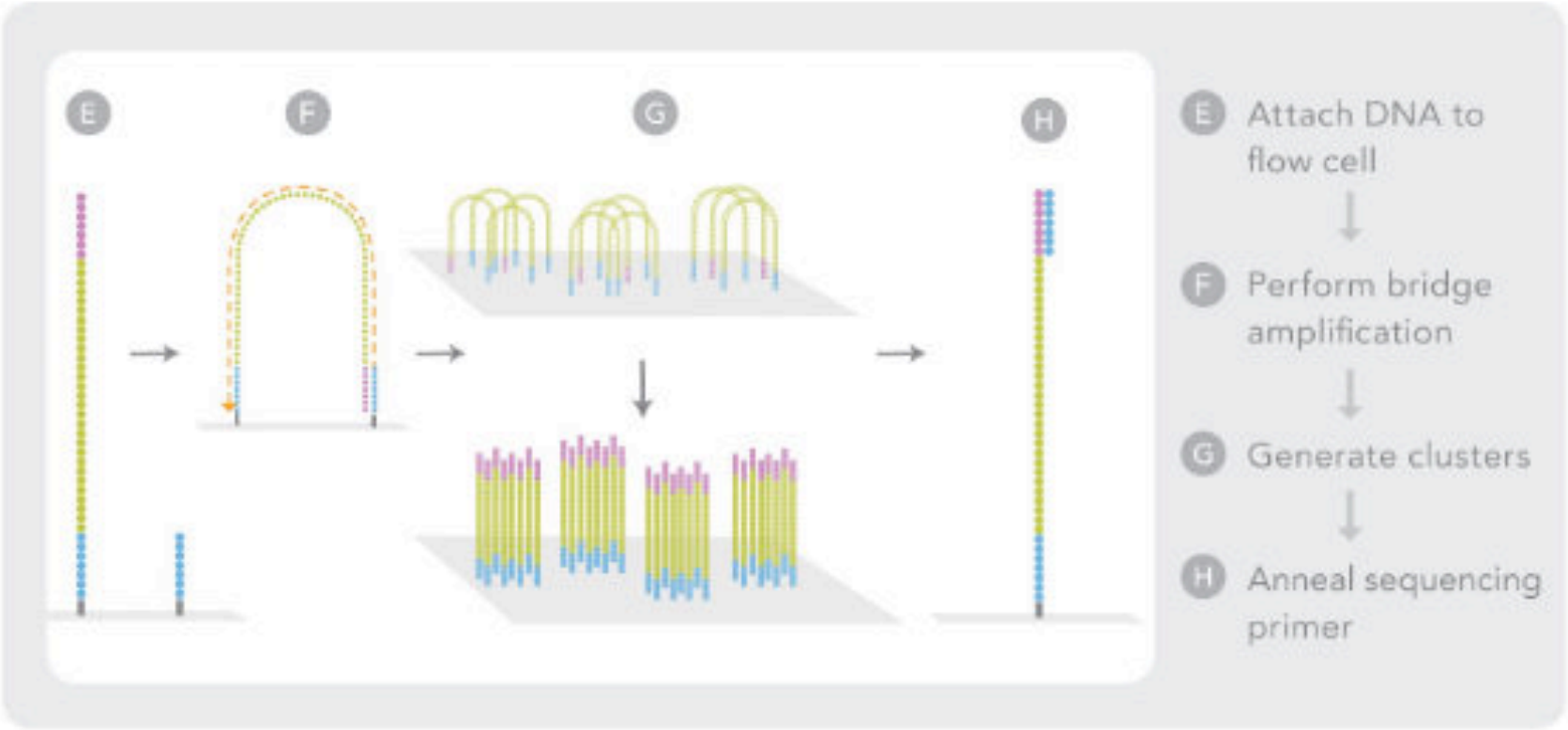
Illumina Technology

- Library Preparation (TruSeq)



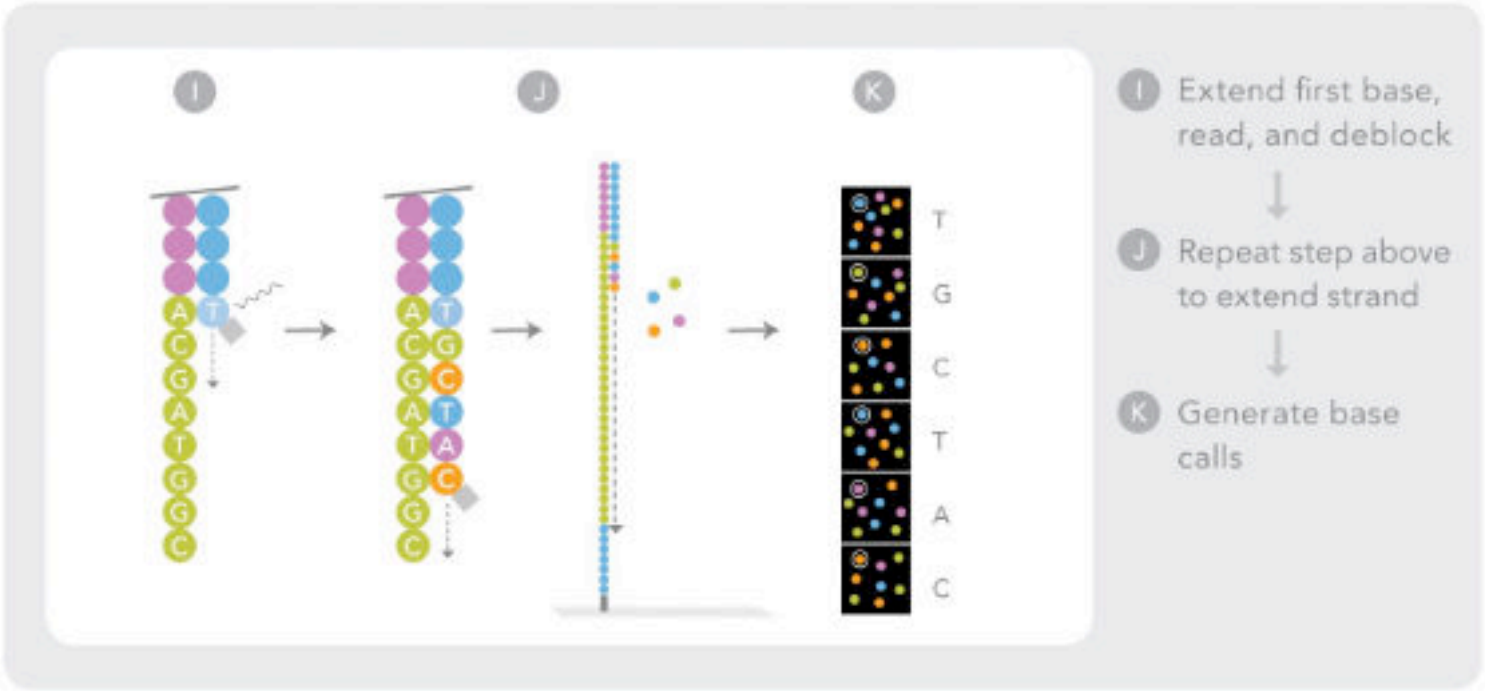
Illumina Technology

Clonal amplification

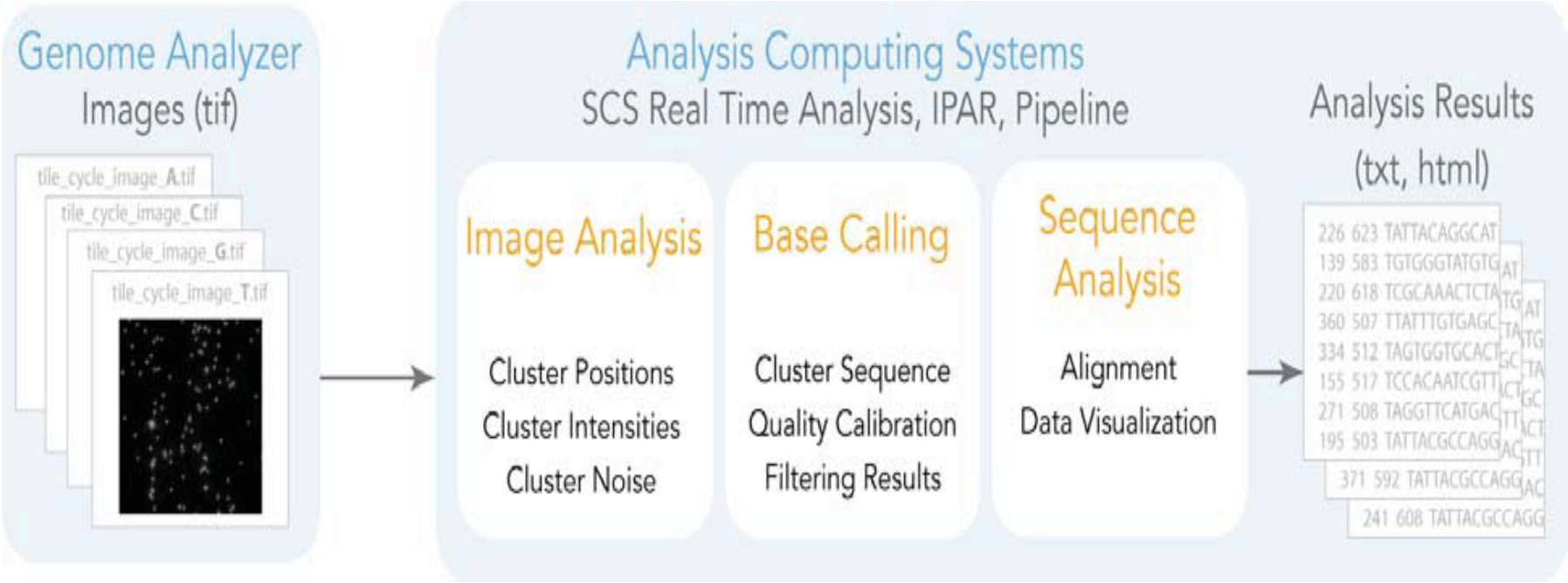


Illumina Technology

Sequencing by synthesis



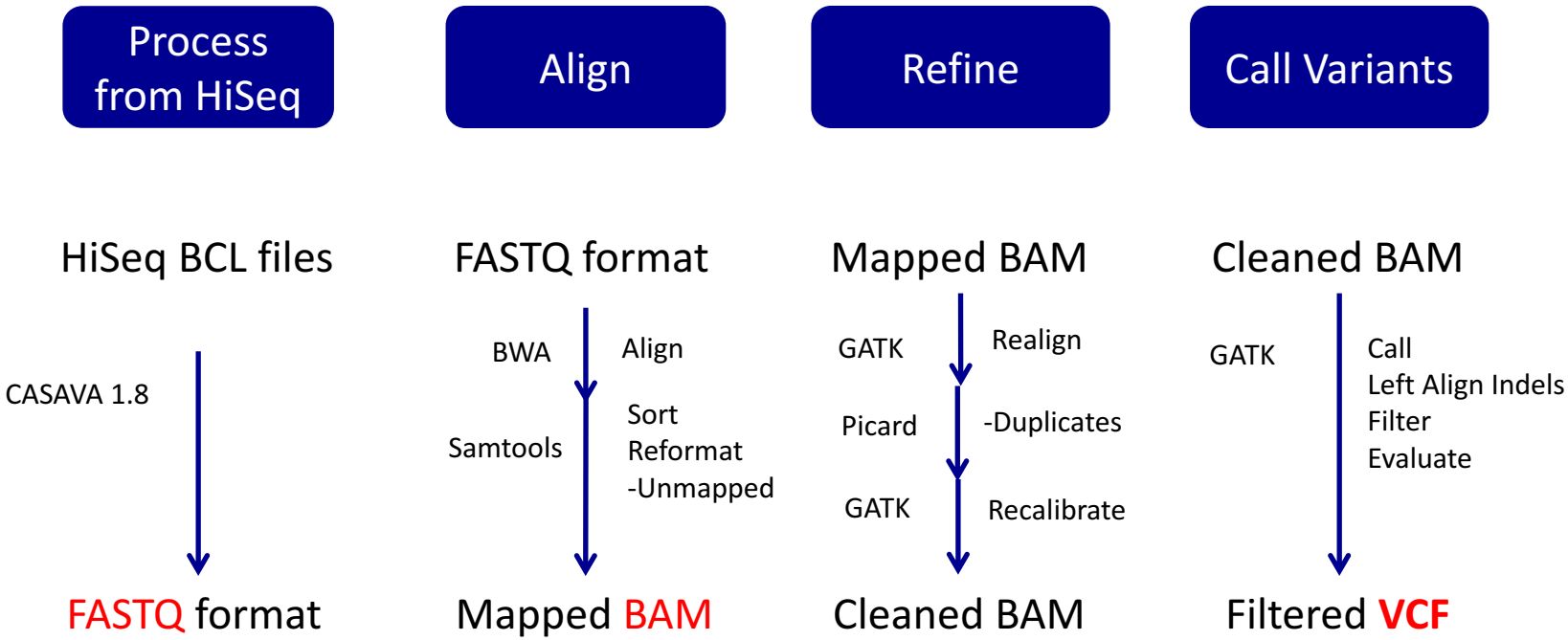
Real Time Analysis



Multi-Step analysis

- Quality Control
- Read Alignment
- Alignment Refinement
- Variant Calling
- Variant Filtering
- Variant Annotation
- Visualization
  - Alignment
  - Variants

Alignment and Variant Calling Pipeline



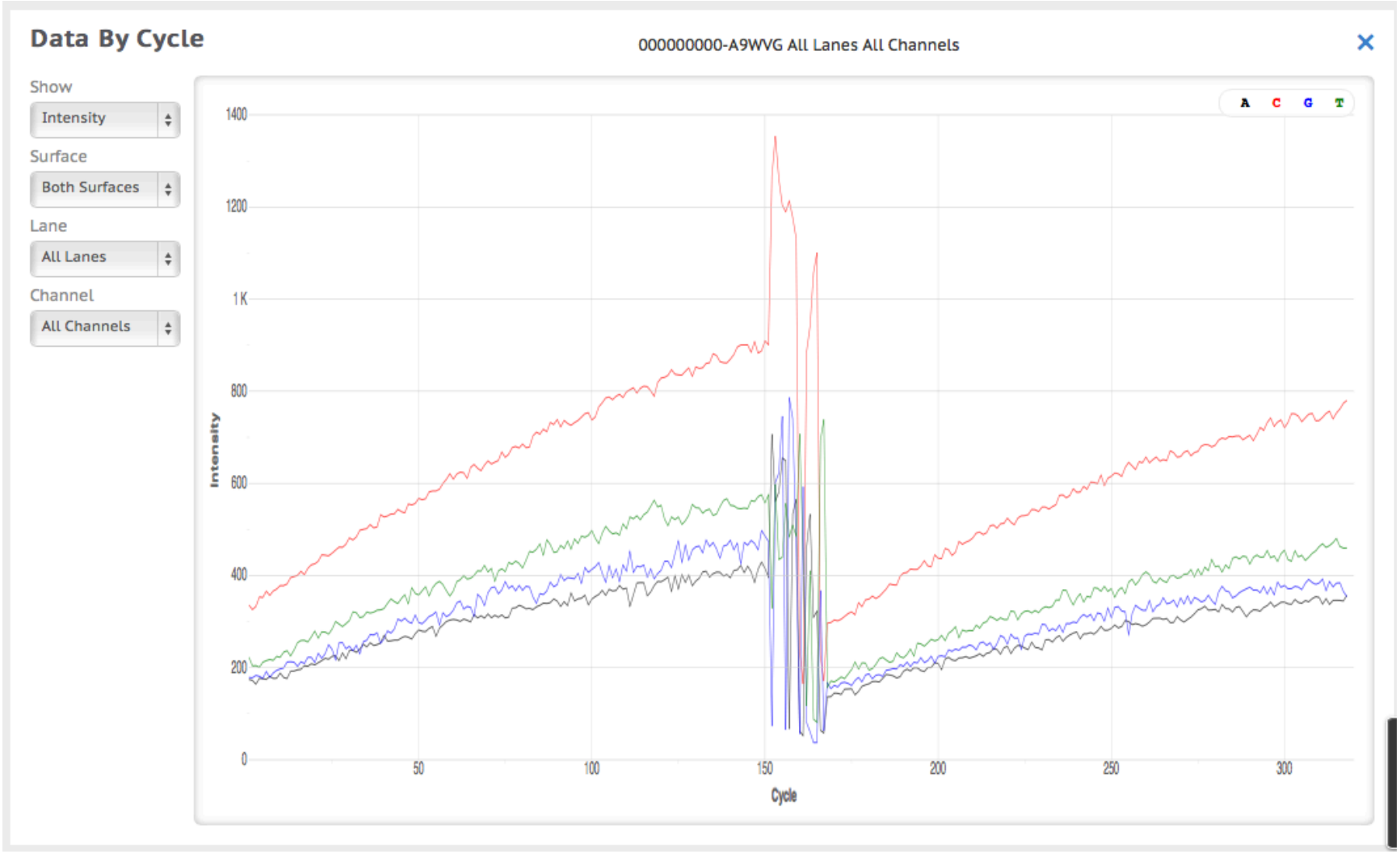
File Sizes

	Depth	FASTQ.gz	BAM	VCF.gz
Genome	30x	100-200Gb	100-200Gb	2Gb
Exome (50 Mb)	120x	5-10 Gb	10-30Gb	3-6Mb
Deep Targeted (150 kb)	1500x	300-500Mb	150Mb	3-6Mb

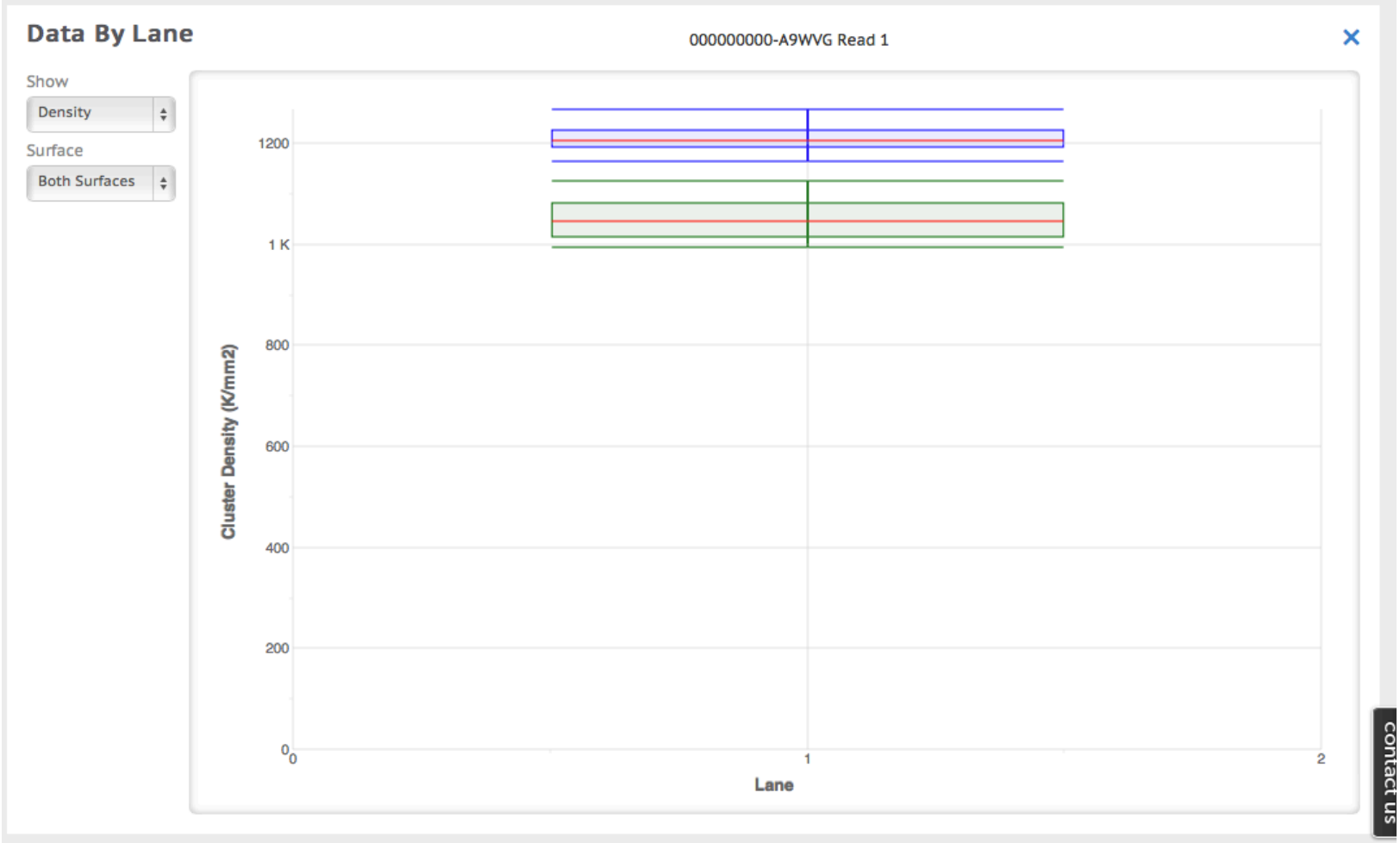
Assume 100 bp read length

INSTRUMENT QC

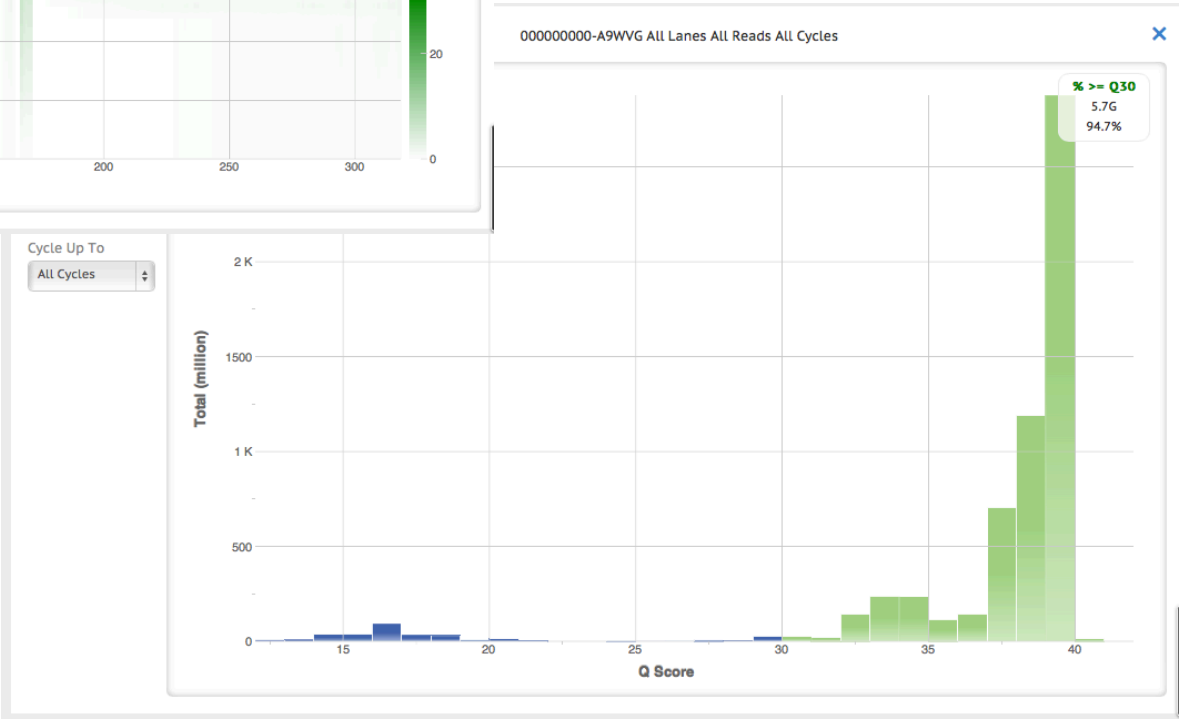
Instrument QC



Instrument QC



Instrument QC



Instrument QC

MiSeq specifications

Cluster Generation and Sequencing

MISEQ REAGENT KIT V2

READ LENGTH	TOTAL TIME*	OUTPUT
1 × 36 bp	~4 hrs	540-610 Mb
2 × 25 bp	~5.5 hrs	750-850 Mb
2 × 150 bp	~24 hrs	4.5-5.1 Gb
2 × 250 bp	~39 hrs	7.5-8.5 Gb

MISEQ REAGENT KIT V3

READ LENGTH	TOTAL TIME*	OUTPUT
2 × 75 bp	~20 hrs	3.3-3.8 Gb
2 × 300 bp	~55 hrs	13.2-15 Gb

Reads Passing Filter†

MISEQ REAGENT KIT V2

Single Reads	12-15 M
Paired-End Reads	24-30 M

MISEQ REAGENT KIT V3

Single Reads	22-25 M
Paired-End Reads	44-50 M

Quality Scores††

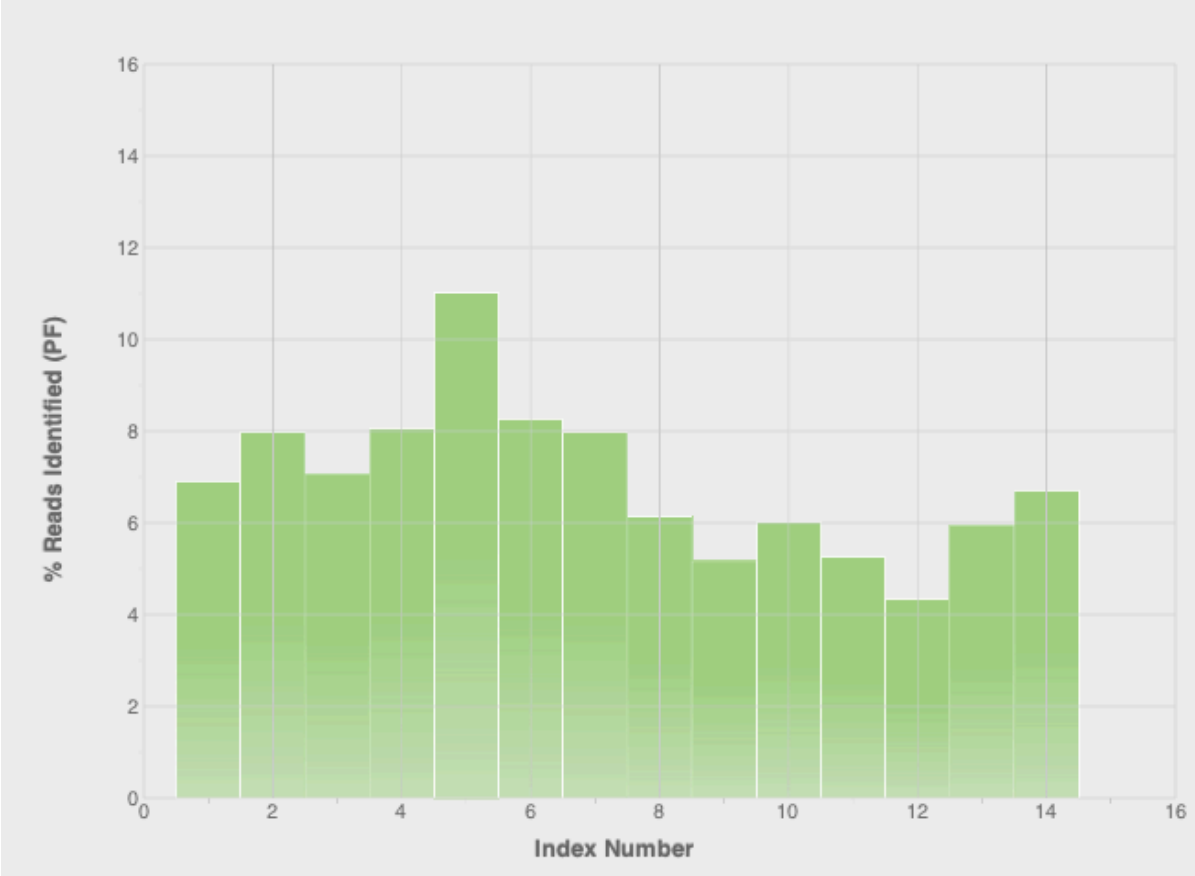
MISEQ REAGENT KIT V2

> 90% bases higher than Q30 at 1 x 36 bp
> 90% bases higher than Q30 at 2 x 25 bp
> 80% bases higher than Q30 at 2 x 150 bp
> 75% bases higher than Q30 at 2 x 250 bp

MISEQ REAGENT KIT V3

> 85% bases higher than Q30 at 2 x 75 bp
> 70% bases higher than Q30 at 2 x 300 bp

Instrument QC



FASTQ

FASTQ

```
@IL31_4368:1:1:996:8507/2
TCCCTTACCCCCAAGCTCCATACCCTCCTAATGCCCACACCTCTTACCTTAGGA
+
FFCEFFFEEFFFFFFFEFFEFFFEFCFC<EEFEFFFCEFF<;EEFF=FEE?FCE
@IL31_4368:1:1:996:21421/2
CAAAAACTTTCACTTTACCTGCCGGGTTTCCCAGTTTACATTCCACTGTTTGAC
+
>DBDDB,B9BAA4AAB7BB?7BBB=91;+*@;5<87+*=/*@@?9=73=.7)7*
@IL31_4368:1:1:997:10572/2
GATCTTCTGTGACTGGAAGAAAATGTGTTACATATTACATTTCTGTCCCCATTG
+
E?=EECE<EEEE98EEEEAEEBD??BE@AEAB><EEABCEEDEC<<EBDA=DEE
@IL31_4368:1:1:997:15684/2
CAGCCTCAGATTCAGCATTCTCAAATTCAGCTGCGGCTGAAACAGCAGCAGGAC
+
EEEEDEEE9EAEEDEEEEEEEEEECEEAAEEDEE<CD=D=*BCAC?;CB,<D@,
@IL31_4368:1:1:997:15249/2
AATGTTCTGAAACCTCTGAGAAAGCAAATATTTATTTTAATGAAAAATCCTTAT
+
EDEEC;EEE;EEE?EECE;7AEEEEEE07EECEA;D6D>+EE4E7EEE4;E=EA
@IL31_4368:1:1:997:6273/2
ACATTTACCAAGACCAAAGGAAACTTACCTTGCAAGAATTAGACAGTTCATTTG
+
EEAAFFFEEFEFCFAFFAFCCFFEFEF>EFFFFB?ABA@ECEE=<F@DE@DDF;
@IL31_4368:1:1:997:1657/2
CCCACCTCTCTCAATGTTTTCCATATGGCAGGGACTCAGCACAGGTGGATTAAT
(...)
```

- Instrument serial #
- Lane
- Swath
- X coord
- Y coord
- Read direction

FASTQ read ID variations

@EAS139:136:FC706VJ:2:2104:15343:197393 1:Y:18:ATCACG

EAS139	the unique instrument name
136	the run id
FC706VJ	the flowcell id
2	flowcell lane
2104	tile number within the flowcell lane
15343	'x'-coordinate of the cluster within the tile
197393	'y'-coordinate of the cluster within the tile
1	the member of a pair, 1 or 2 *(paired-end or mate-pair reads only)*
Y	Y if the read is filtered, N otherwise
18	0 when none of the control bits are on, otherwise it is an even number
ATCACG	index sequence

PHRED-like quality

```
 SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....................................................
 ..........................XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......................
 ..............................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII......................
 ..................................JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ......................
 LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL.....................................................
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
  |                          |    |        |                              |                     |
33                          59   64       73                            104                   126
 0........................26...31.......40
                          -5....0........9.............................40
                               0........9.............................40
                                  3.....9.............................40
 0.2......................26...31.........41

S - Sanger        Phred+33,  raw reads typically (0, 40)
X - Solexa        Solexa+64, raw reads typically (-5, 40)
I - Illumina 1.3+ Phred+64,  raw reads typically (0, 40)
J - Illumina 1.5+ Phred+64,  raw reads typically (3, 40)
    with 0=unused, 1=unused, 2=Read Segment Quality Control Indicator (bold)
    (Note: See discussion above).
L - Illumina 1.8+ Phred+33,  raw reads typically (0, 41)
```

Phred=-10log10(error)

ASCII-Code

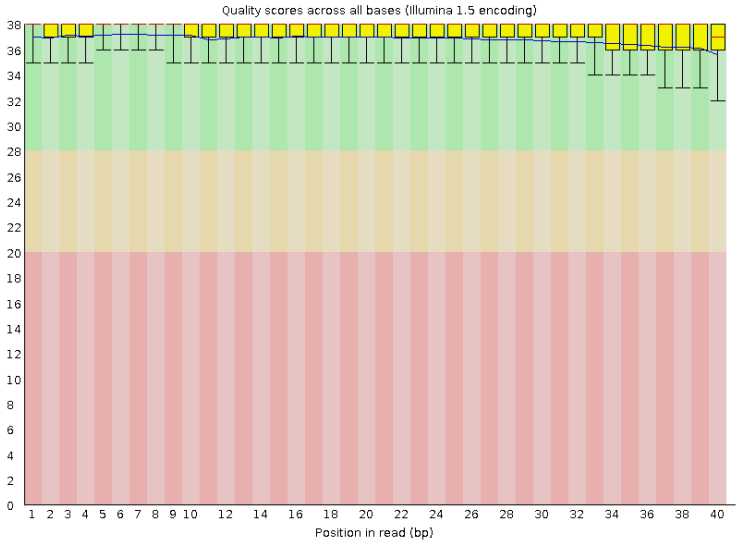
ASCII value	Character	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character
000	(null)	NUL	032	(space)	064	@	096	
001	☺	SOH	033	!	065	A	097	a
002	☻	STX	034	"	066	B	098	b
003	♥	ETX	035	#	067	C	099	c
004	♦	EOT	036	$	068	D	100	d
005	♣	ENQ	037	%	069	E	101	e
006	♠	ACK	038	&	070	F	102	f
007	(beep)	BEL	039	'	071	G	103	g
008	◘	BS	040	(	072	H	104	h
009	(tab)	HT	041	)	073	I	105	i
010	(line feed)	LF	042	*	074	J	106	j
011	(home)	VT	043	+	075	K	107	k
012	(form feed)	FF	044	,	076	L	108	l
013	(carriage return)	CR	045	-	077	M	109	m
014	♫	SO	046	.	078	N	110	n
015	☼	SI	047	/	079	O	111	o
016	►	DLE	048	0	080	P	112	p
017	◄	DC1	049	1	081	Q	113	q
018	↕	DC2	050	2	082	R	114	r
019	!!	DC3	051	3	083	S	115	s
020	¶	DC4	052	4	084	T	116	t
021	§	NAK	053	5	085	U	117	u
022	▬	SYN	054	6	086	V	118	v
023	↨	ETB	055	7	087	W	119	w
024	↑	CAN	056	8	088	X	120	x
025	↓	EM	057	9	089	Y	121	y
026	→	SUB	058	:	090	Z	122	z
027	←	ESC	059	;	091	[	123	{
028	(cursor right)	FS	060	<	092	\	124	|
029	(cursor left)	GS	061	=	093	]	125	}
030	(cursor up)	RS	062	>	094	^	126	~
031	(cursor down)	US	063	?	095	_	127	⌂

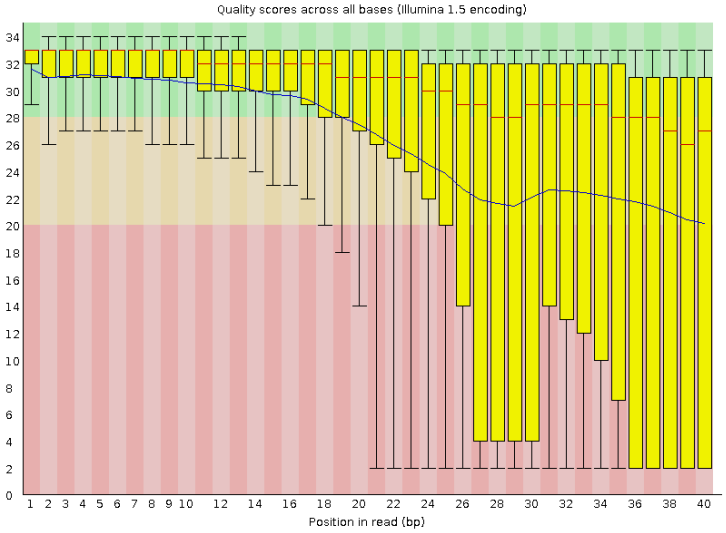
FASTQ QC

FASTQC

Measure	Value
Filename	good_sequence_short.txt
File type	Conventional base calls
Encoding	Illumina 1.5
Total Sequences	250000
Sequences flagged as poor quality	0
Sequence length	40
%GC	45

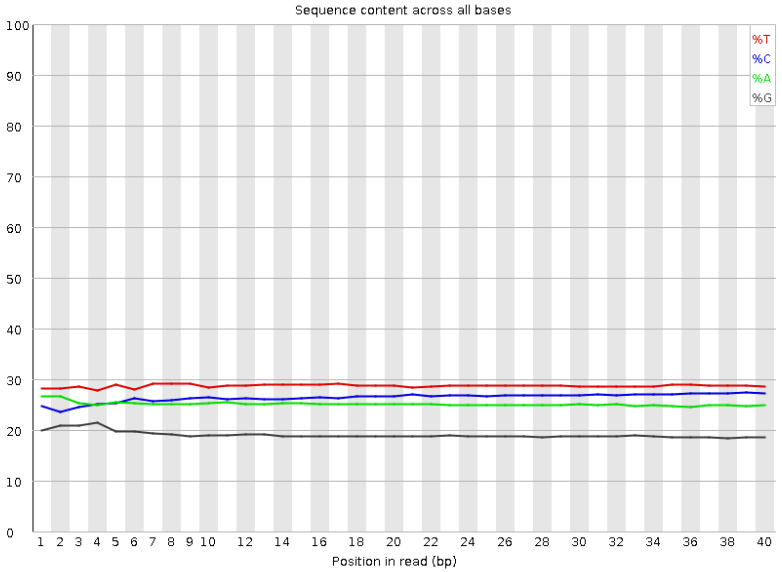
- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content
- Kmer Content

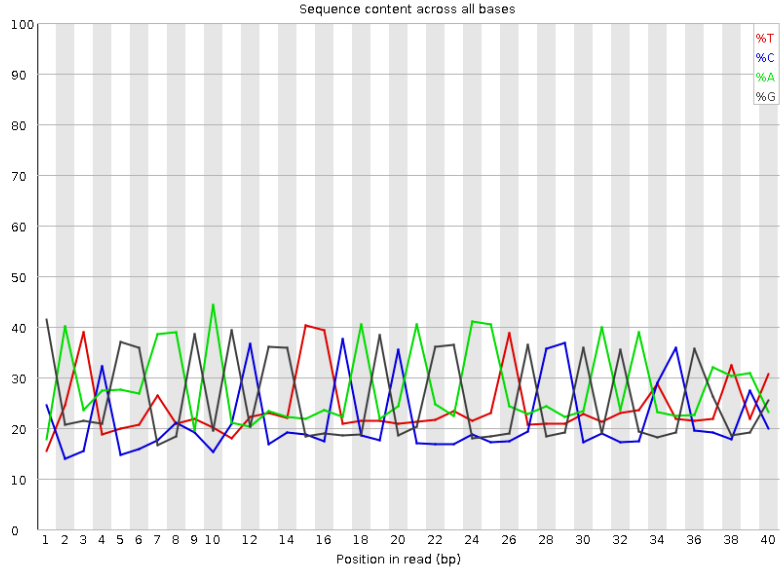


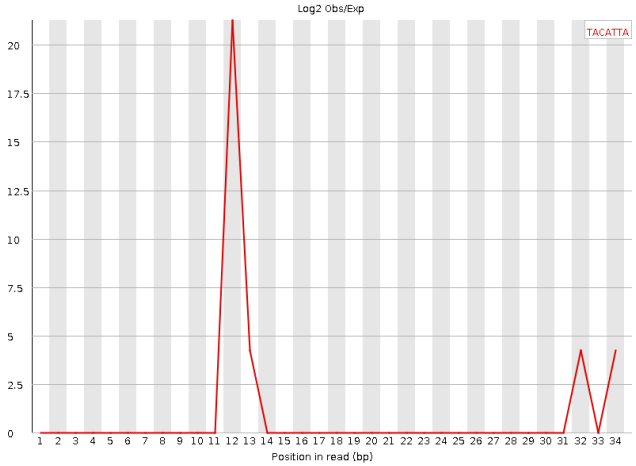


http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

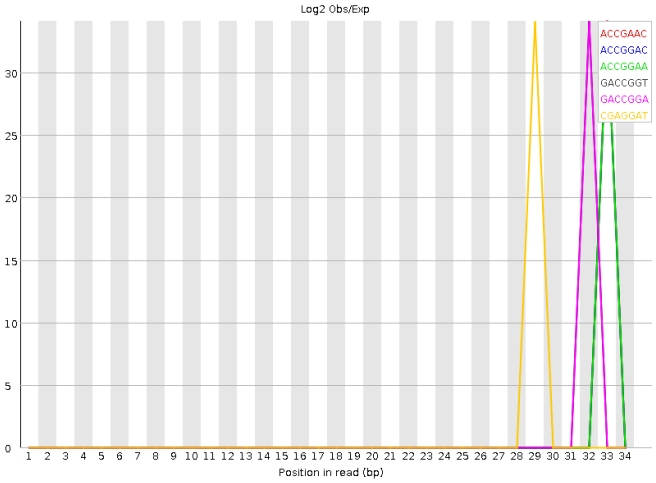
FASTQC Adapter Contamination







Sequence	Count	PValue	Obs/Exp Max	Max Obs/Exp Position
TACATTA	40	0.003151852	21.2465	12



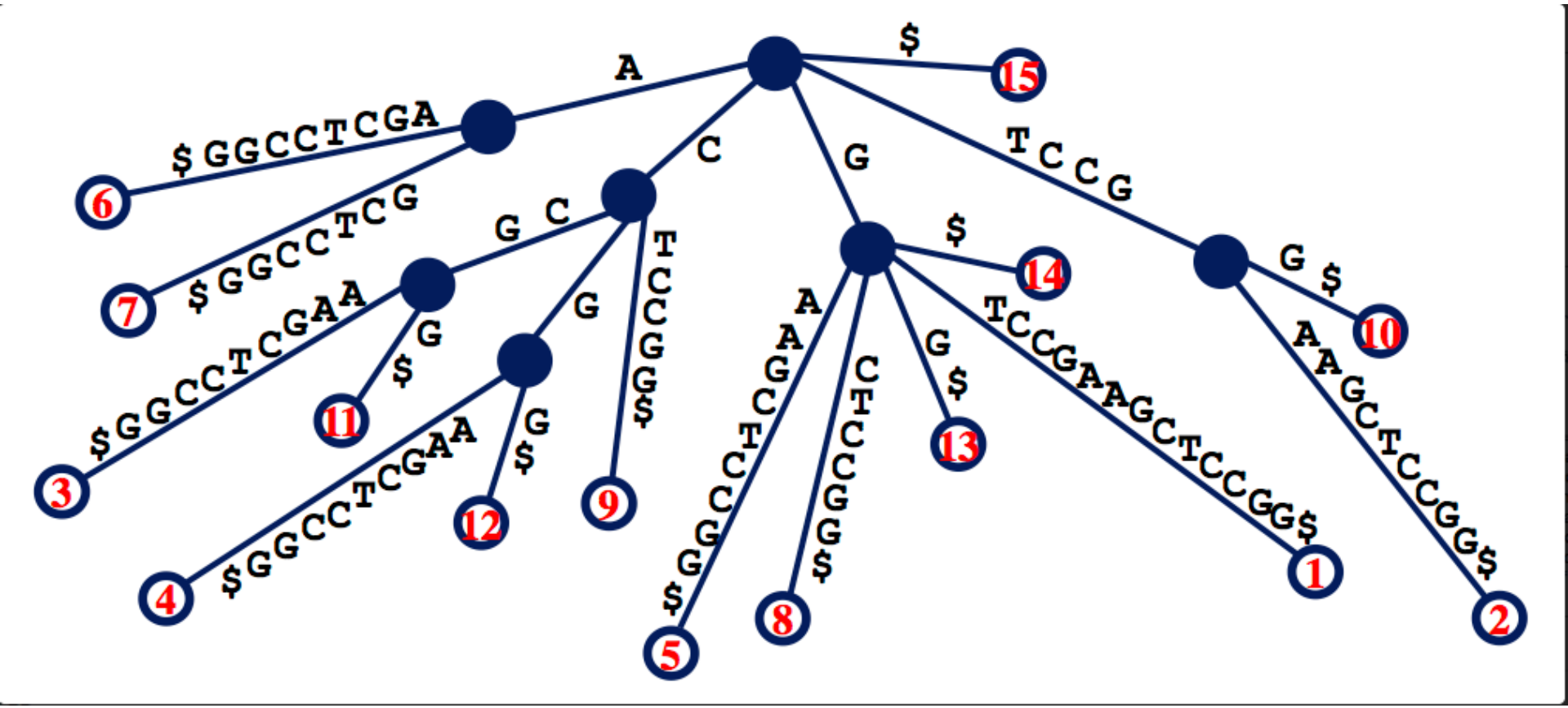
Sequence	Count	PValue	Obs/Exp Max	Max Obs/Exp Position
ACCGAAC	35	1.0615131E-6	34.067673	33
ACCGGAC	30	1.4503141E-5	34.06767	33
ACCGGAA	55	3.092282E-11	34.06767	33
GACCGGT	20	0.0027499169	34.06767	32
GACCGGA	95	0.0	34.06767	32

Using BWA

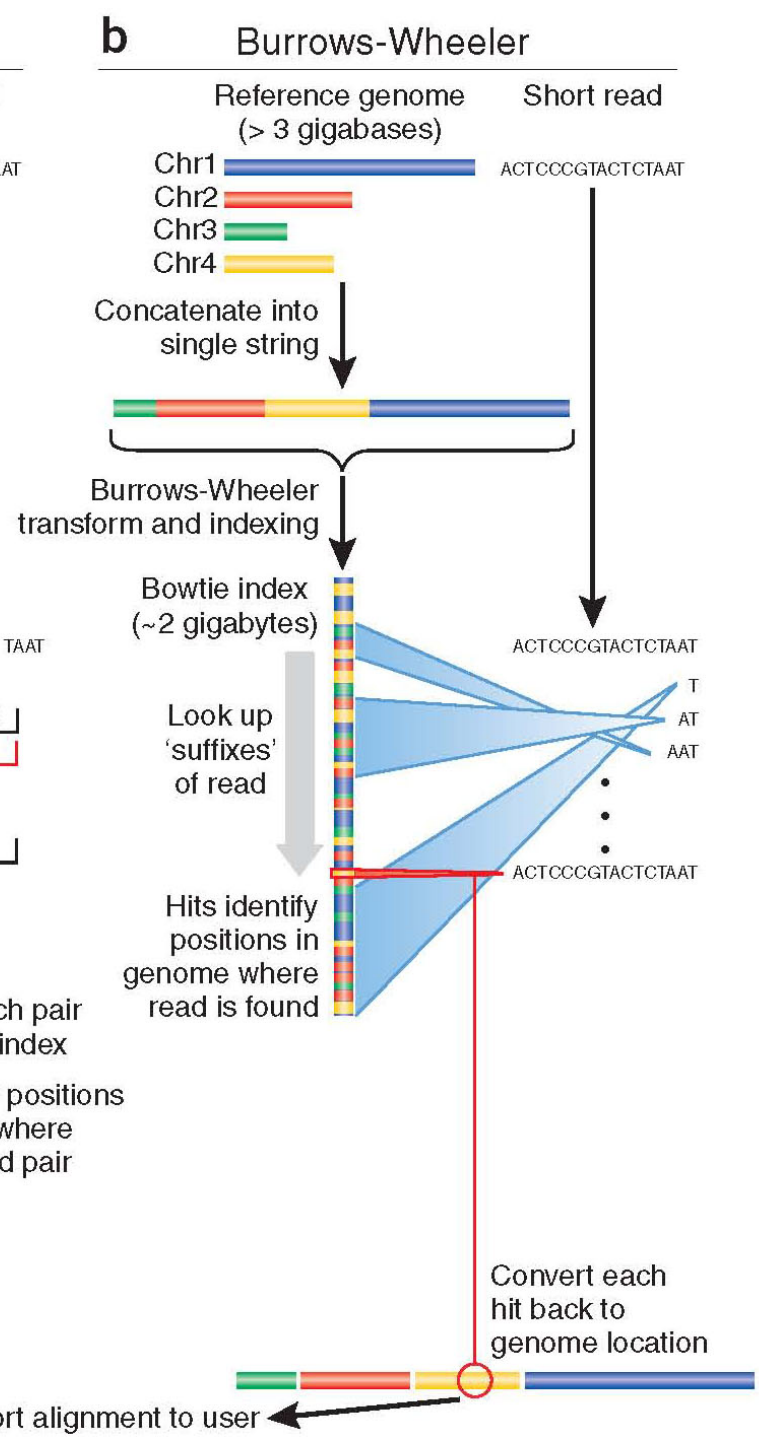
ALIGNMENT

Suffix Array

GTCCGAAGCTCCGG$



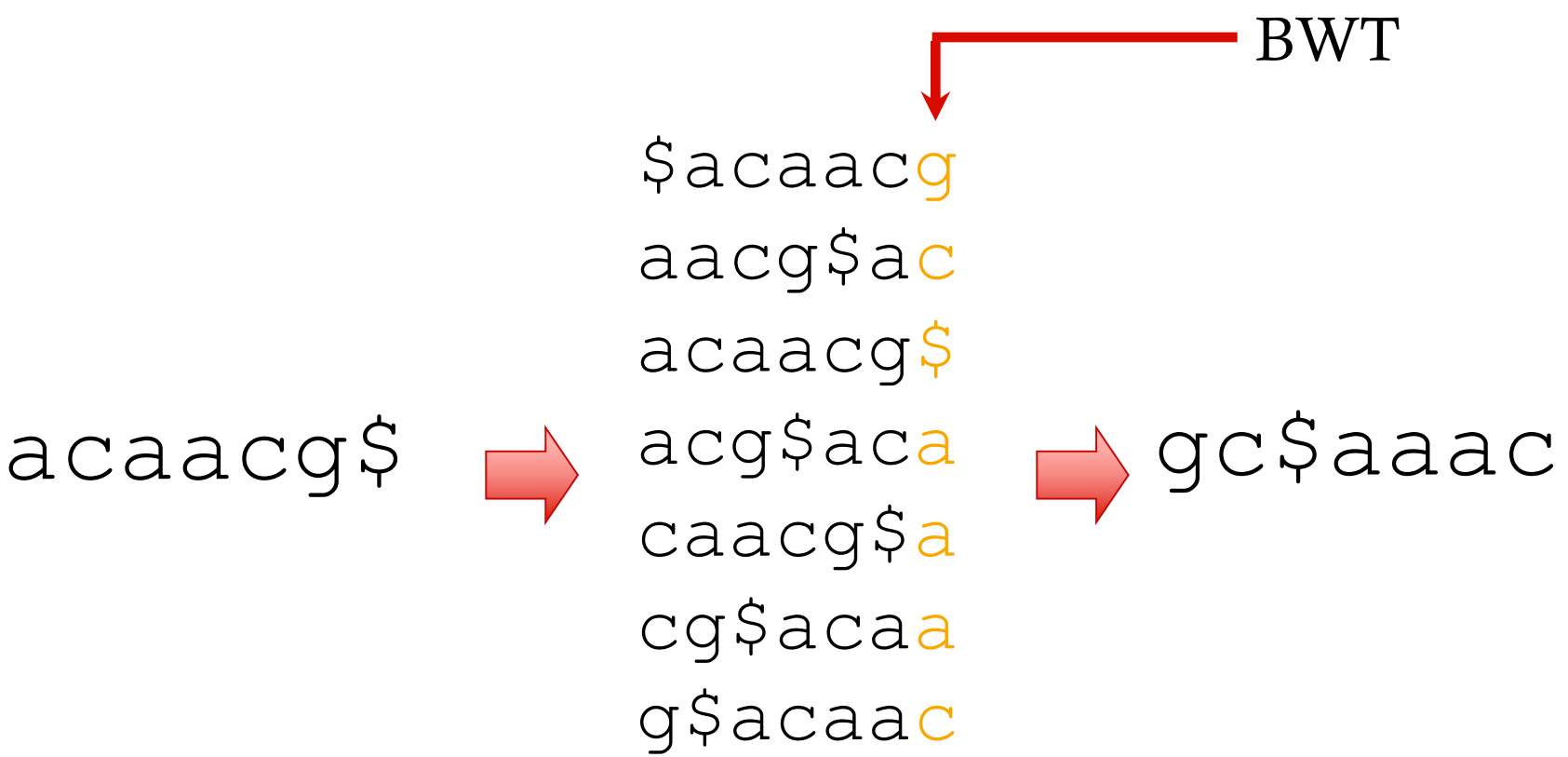
Fast but too much memory requirement for the whole genome



Burrows-Wheeler

- Store entire reference genome.
- Align tag base by base from the end.
- When tag is traversed, all active locations are reported.
- If no match is found, then back up and try a substitution.

Burrows-Wheeler Transform (BWT)



Burrows-Wheeler Matrix

$acaacg

aacg$ac

acaacg$

acg$aca

caacg$a

cg$acaa

g$acaac

Burrows-Wheeler Matrix

	$acaacg
3	aacg$ac
1	acaacg$
4	acg$aca
2	caacg$a
5	cg$acaa
6	g$acaac

Burrows-Wheeler Transform

Langmead et al (2009)

BWA

```
Program: bwa (alignment via Burrows-Wheeler transformation)
Version: 0.7.9a-r786
Contact: Heng Li <lh3@sanger.ac.uk>

Usage:   bwa <command> [options]

Command: index         index sequences in the FASTA format
         mem           BWA-MEM algorithm
         fastmap       identify super-maximal exact matches
         pemerge       merge overlapping paired ends (EXPERIMENTAL)
         aln           gapped/ungapped alignment
         samse         generate alignment (single ended)
         sampe         generate alignment (paired ended)
         bwasw         BWA-SW for long queries

         fa2pac        convert FASTA to PAC format
         pac2bwt       generate BWT from PAC
         pac2bwtgen    alternative algorithm for generating BWT
         bwtupdate     update .bwt to the new format
         bwt2sa        generate SA from BWT and Occ

Note: To use BWA, you need to first index the genome with `bwa index'.
      There are three alignment algorithms in BWA: `mem', `bwasw', and
      `aln/samse/sampe'. If you are not sure which to use, try `bwa mem'
      first. Please `man ./bwa.1' for the manual.
```

BWA mem

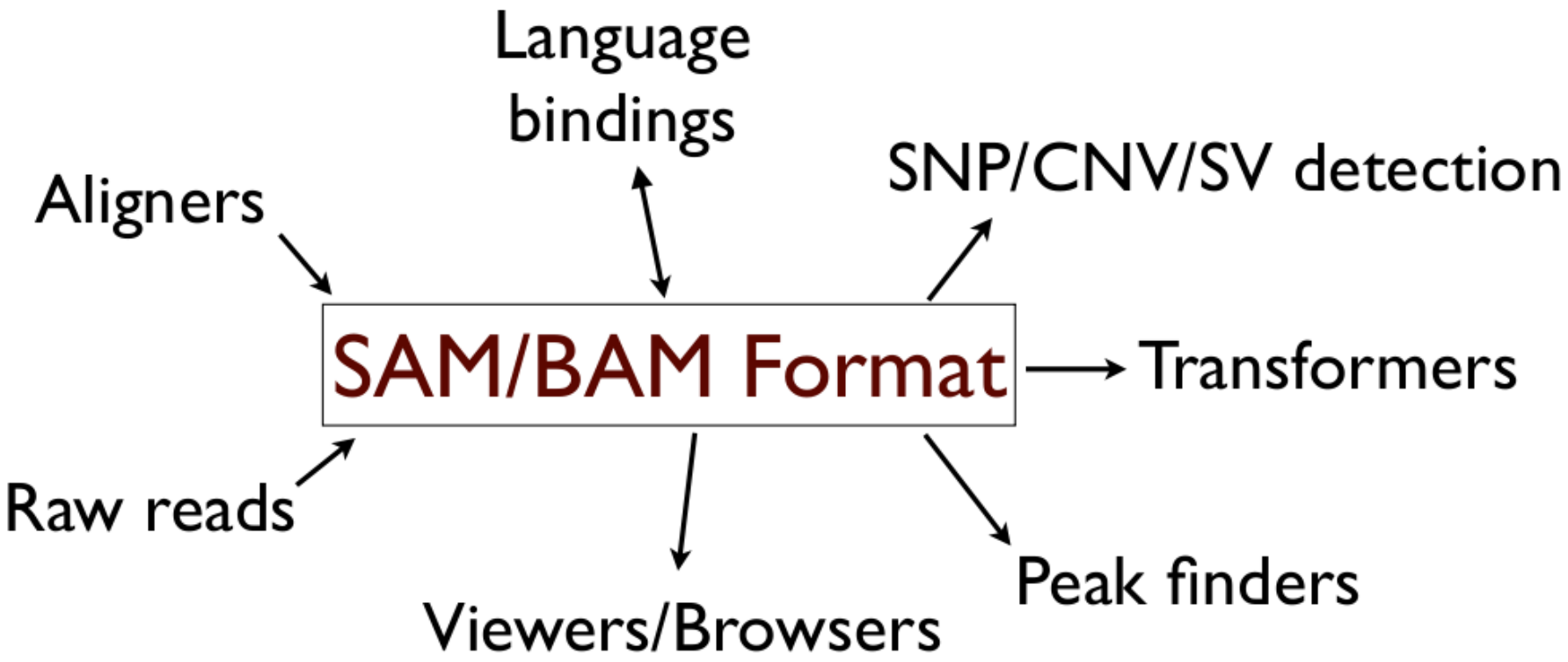
```
Usage: bwa mem [options] <idxbase> <in1.fq> [in2.fq]

Algorithm options:

       -t INT        number of threads [1]
       -k INT        minimum seed length [19]
       -w INT        band width for banded alignment [100]
       -d INT        off-diagonal X-dropoff [100]
       -r FLOAT      look for internal seeds inside a seed longer than {-k} * FLOAT [1.5]
       -c INT        skip seeds with more than INT occurrences [500]
       -D FLOAT      drop chains shorter than FLOAT fraction of the longest overlapping chain [0.50]
       -W INT        discard a chain if seeded bases shorter than INT [0]
       -m INT        perform at most INT rounds of mate rescues for each read [50]
       -S            skip mate rescue
       -P            skip pairing; mate rescue performed unless -S also in use
       -e            discard full-length exact matches
       -A INT        score for a sequence match, which scales options -TdBOELU unless overridden [1]
       -B INT        penalty for a mismatch [4]
       -O INT[,INT]  gap open penalties for deletions and insertions [6,6]
       -E INT[,INT]  gap extension penalty; a gap of size k cost '{-O} + {-E}*k' [1,1]
       -L INT[,INT]  penalty for 5'- and 3'-end clipping [5,5]
       -U INT        penalty for an unpaired read pair [17]
       -x STR        read type. Setting -x changes multiple parameters unless overriden [null]
                     pacbio: -k17 -W40 -r10 -A2 -B5 -O2 -E1 -L0
                     pbread: -k13 -W40 -c1000 -r10 -A2 -B5 -O2 -E1 -N25 -FeaD.001
```

BWA mem

```
Input/output options:

       -p            first query file consists of interleaved paired-end sequences
       -R STR        read group header line such as '@RG\tID:foo\tSM:bar' [null]

       -v INT        verbose level: 1=error, 2=warning, 3=message, 4+=debugging [3]
       -T INT        minimum score to output [30]
       -h INT        if there are <INT hits with score >80% of the max score, output all in XA [5]
       -a            output all alignments for SE or unpaired PE
       -C            append FASTA/FASTQ comment to SAM output
       -Y            use soft clipping for supplementary alignments
       -M            mark shorter split hits as secondary

       -I FLOAT[,FLOAT[,INT[,INT]]]
                     specify the mean, standard deviation (10% of the mean if absent), max
                     (4 sigma from the mean if absent) and min of the insert size distribution.
                     FR orientation only. [inferred]
```

SAM/BAM FORMAT

Utility


Language bindings
Aligners
SNP/CNV/SV detection
SAM/BAM Format
Transformers
Raw reads
Viewers/Browsers
Peak finders

BAM format

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	`[!-?A-~]{1,255}`	Query template NAME
2	FLAG	Int	`[0,2^16-1]`	bitwise FLAG
3	RNAME	String	`\*|[!-()+-<>-~][!-~]*`	Reference sequence NAME
4	POS	Int	`[0,2^29-1]`	1-based leftmost mapping POSition
5	MAPQ	Int	`[0,2^8-1]`	MAPping Quality
6	CIGAR	String	`\*|([0-9]+[MIDNSHPX=])+`	CIGAR string
7	RNEXT	String	`\*|=|[!-()+-<>-~][!-~]*`	Ref. name of the mate/next segment
8	PNEXT	Int	`[0,2^29-1]`	Position of the mate/next segment
9	TLEN	Int	`[-2^29+1,2^29-1]`	observed Template LENgth
10	SEQ	String	`\*|[A-Za-z=.]+`	segment SEQuence
11	QUAL	String	`[!-~]+`	ASCII of Phred-scaled base QUALity+33

Bioinformatics. 2009 Aug 15;25(16):2078-9. Epub 2009 Jun 8. The Sequence Alignment/Map format and SAMtools.

BAM example (field 1-9)

DER4KXP1:174:D15C1ACXX:1:2207:13442:83865	163	chr1	761784	37	100M	=	761817	133	
DER4KXP1:174:D15C1ACXX:1:1107:14909:28891	163	chr1	761801	60	100M	=	762070	369	
DER4KXP1:174:D15C1ACXX:1:1216:6353:47474	163	chr1	761803	29	100M	=	762147	444	
DER4KXP1:174:D15C1ACXX:1:1301:6227:77514	163	chr1	761808	29	100M	=	762134	426	
DER4KXP1:174:D15C1ACXX:1:1313:17938:49828	99	chr1	761810	46	100M	=	762100	390	
DER4KXP1:174:D15C1ACXX:1:2207:13442:83865	83	chr1	761817	29	100M	=	761784	-133	
DER4KXP1:174:D15C1ACXX:1:2311:10645:84244	99	chr1	761824	29	100M	=	762081	357	
DER4KXP1:174:D15C1ACXX:1:1201:14518:8618	163	chr1	761833	36	100M	=	762058	325	
DER4KXP1:174:D15C1ACXX:1:2208:10379:46594	99	chr1	761840	29	100M	=	762052	312	
DER4KXP1:174:D15C1ACXX:1:2306:9949:25192	99	chr1	761851	29	100M	=	762112	361	
DER4KXP1:174:D15C1ACXX:1:1102:14903:72066	99	chr1	761862	29	100M	=	762134	372	
DER4KXP1:174:D15C1ACXX:1:2301:17227:60156	99	chr1	761862	60	100M	=	762075	313	
DER4KXP1:174:D15C1ACXX:1:1314:10931:20595	163	chr1	761867	0	91M1I8M	=	762055	288	
DER4KXP1:174:D15C1ACXX:1:2111:2062:12110	99	chr1	761869	60	100M	=	762053	284	
DER4KXP1:174:D15C1ACXX:1:2113:9776:45228	163	chr1	761870	60	100M	=	762020	250	
DER4KXP1:174:D15C1ACXX:1:2301:16409:71094	163	chr1	761871	60	100M	=	762157	386	
DER4KXP1:174:D15C1ACXX:1:1210:12439:97466	163	chr1	761872	0	86M1I13M	=	762052	280	
DER4KXP1:174:D15C1ACXX:1:2202:9605:74870	163	chr1	761874	0	84M1I15M	=	762069	295	
DER4KXP1:174:D15C1ACXX:1:1106:16383:81731	99	chr1	761880	60	100M	=	762017	237	
DER4KXP1:174:D15C1ACXX:1:1110:17641:12224	163	chr1	761880	0	78M1I21M	=	762137	357	
DER4KXP1:174:D15C1ACXX:1:2214:13641:15965	163	chr1	761880	0	78M1I21M	=	762056	276	

CIGAR string

Op	BAM	Description
M	0	alignment match (can be a sequence match or mismatch)
I	1	insertion to the reference
D	2	deletion from the reference
N	3	skipped region from the reference
S	4	soft clipping (clipped sequences present in SEQ)
H	5	hard clipping (clipped sequences NOT present in SEQ)
P	6	padding (silent deletion from padded reference)
=	7	sequence match
X	8	sequence mismatch

Before alignment

```
RefPos:     1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
Reference:  C  C  A  T  A  C  T  G  A  A  C  T  G  A  C  T  A  A  C
Read: ACTAGAATGGCT
```

After alignment

```
RefPos:     1  2  3  4  5  6  7     8  9 10 11 12 13 14 15 16 17 18 19
Reference:  C  C  A  T  A  C  T     G  A  A  C  T  G  A  C  T  A  A  C
Read:                  A  C  T  A  G  A  A     T  G  G  C  T
```

POS: 5
CIGAR: 3M1I3M1D5M

BAM example

```
Coor     12345678901234  5678901234567890123456789012345
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

+r001/1        TTAGATAAAGGATA*CTG
+r002         aaaAGATAA*GGATA
+r003       gcctaAGCTAA
+r004                       ATAGCT..............TCAGC
-r003                              ttagctTAGGC
-r001/2                                        CAGCGCCAT
```

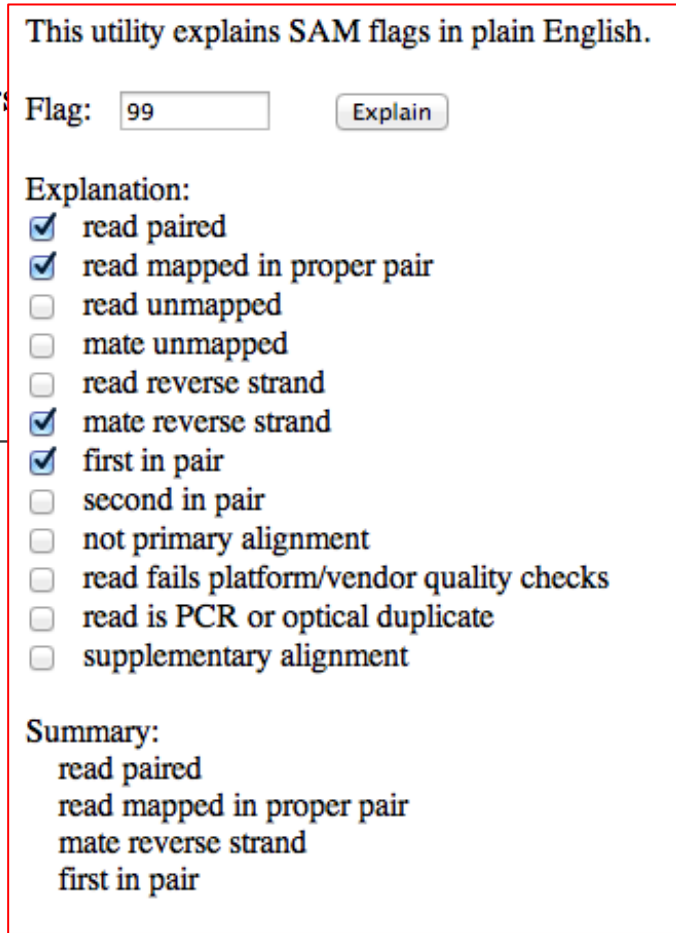
The corresponding SAM format is:

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

SAM FLAGS

Bit	Description
0x1	template having multiple segments in sequencing
0x2	each segment properly aligned according to the aligner
0x4	segment unmapped
0x8	next segment in the template unmapped
0x10	SEQ being reverse complemented
0x20	SEQ of the next segment in the template being revers
0x40	the first segment in the template
0x80	the last segment in the template
0x100	secondary alignment
0x200	not passing quality controls
0x400	PCR or optical duplicate
0x800	supplementary alignment

http://picard.sourceforge.net/explain-flags.html



Extra Info (custom)

AS	i	Alignment score generated by aligner
BC	Z	Barcode sequence, with any quality scores stored in the QT tag.
BQ	Z	Offset to base alignment quality (BAQ), of the same length as the read sequence. At the i-th read base, $\mathrm{BAQ}_i = Q_i - (\mathrm{BQ}_i - 64)$ where Q_i is the i-th base quality.
CC	Z	Reference name of the next hit; '=' for the same chromosome
CM	i	Edit distance between the color sequence and the color reference (see also NM)
CO	Z	Free-text comments
CP	i	Leftmost coordinate of the next hit
CQ	Z	Color read quality on the original strand of the read. Same encoding as QUAL; same length as CS.
CS	Z	Color read sequence on the original strand of the read. The primer base must be included.
CT	Z	Complete read annotation tag, used for consensus annotation dummy features.[5]
E2	Z	The 2nd most likely base calls. Same encoding and same length as QUAL.
FI	i	The index of segment in the template.
FS	Z	Segment suffix.
FZ	B,S	Flow signal intensities on the original strand of the read, stored as (uint16_t) round(value * 100.0).
LB	Z	Library. Value to be consistent with the header RG-LB tag if @RG is present.
H0	i	Number of perfect hits
H1	i	Number of 1-difference hits (see also NM)
H2	i	Number of 2-difference hits
HI	i	Query hit index, indicating the alignment record is the i-th one stored in SAM
IH	i	Number of stored alignments in SAM that contains the query in the current record
MC	Z	CIGAR string for mate/next segment
MD	Z	String for mismatching positions. *Regex*: [0-9]+(([A-Z]|\^[A-Z]+)[0-9]+)*[6]

```
Reference: ...CTTCTATTATCCTT...  M       =/X          MD
     Read:    CTTCTATTATCCTT     14M     14=          14        // example 1
     Read:    CTTATATTATCCTT     14M     3=1X10=      3C10      // example 2
     Read:    CTTATATTGGCCTT     14M     3=1X4=2X4=   3C4AT4
     Read:    CTTCTATTGGCCTT     14M     8=2X4=       8AT4
     Read:    TTTATATTATCCTG     14M     1X12=1X      0C12T0
```

BAM/SAM format

- flexible : compatible with multiple alignment programs
- simple: easy to generate and convert
- compact in file size
- works on a stream : low memory footprint
- Indexable for efficiency
- SAM is human readable

SAM/BAM manipulation

SAMTOOLS & PICARD

SAMTOOLS

- Manipulate alignments in the BAM format
  - imports from and exports to the SAM (Sequence Alignment/Map) format
  - sorting
  - merging
  - indexing
  - retrieve reads in any regions
- Works on a stream. Commands combined with Unix pipes.
- Able to open a BAM on a remote FTP or HTTP server

http://www.htslib.org

SAMTOOLS examples

```
Program: samtools (Tools for alignments in the SAM format)
Version: 0.1.19-44428cd

Usage:   samtools <command> [options]

Command: view         SAM<->BAM conversion
         sort         sort alignment file
         mpileup      multi-way pileup
         depth        compute the depth
         faidx        index/extract FASTA
         tview        text alignment viewer
         index        index alignment
         idxstats     BAM index stats (r595 or later)
         fixmate      fix mate information
         flagstat     simple stats
         calmd        recalculate MD/NM tags and '=' bases
         merge        merge sorted alignments
         rmdup        remove PCR duplicates
         reheader     replace BAM header
         cat          concatenate BAMs
         bedcov       read depth per BED region
         targetcut    cut fosmid regions (for fosmid pool only)
         phase        phase heterozygotes
         bamshuf      shuffle and group alignments by name
```

http://www.htslib.org

SAMTOOLS view

Usage: samtools view [options] <in.bam>|<in.sam> [region1 [...]]

Options: -b output BAM

-h print header for the SAM output

-H print header only (no alignments)

-S input is SAM

-u uncompressed BAM output (force -b)

-1 fast compression (force -b)

-x output FLAG in HEX (samtools-C specific)

-X output FLAG in string (samtools-C specific)

-c print only the count of matching records

-B collapse the backward CIGAR operation

-@ INT number of BAM compression threads [0]

-L FILE output alignments overlapping the input BED FILE [null]

-t FILE list of reference names and lengths (force -S) [null]

-T FILE reference sequence file (force -S) [null]

-o FILE output file name [stdout]

-R FILE list of read groups to be outputted [null]

-f INT required flag, 0 for unset [0]

-F INT filtering flag, 0 for unset [0]

-q INT minimum mapping quality [0]

-l STR only output reads in library STR [null]

-r STR only output reads in read group STR [null]

-s FLOAT fraction of templates to subsample; integer part as seed [-1]

-? longer help

http://www.htslib.org

SAMTOOLS Pileup

Chrom Ref

Position

Coverage

Read bases

Qualities

Picard – the other tool box

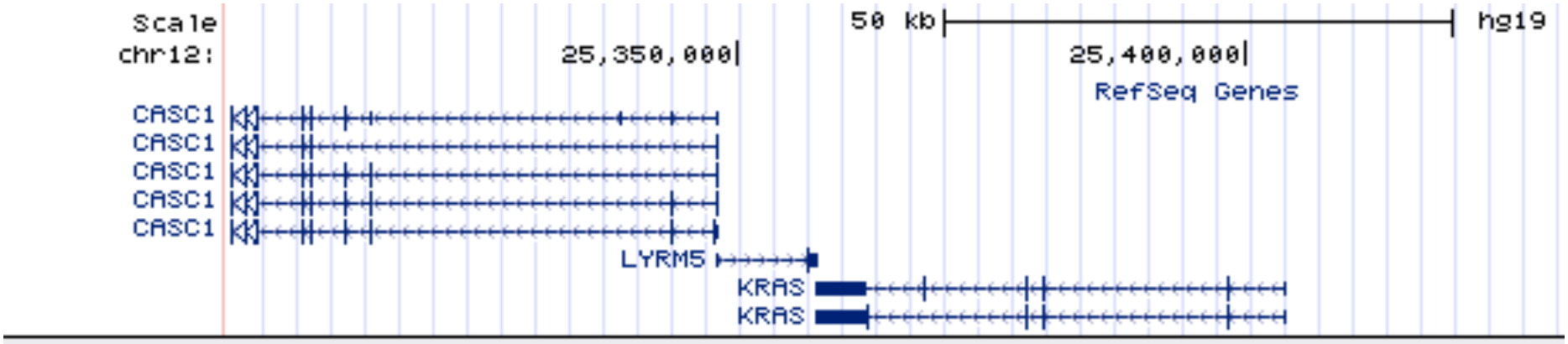
- AddCommentsToBam
- AddOrReplaceReadGroups
- BamToBfq
- BamIndexStats
- BuildBamIndex
- CalculateHsMetrics
- CleanSam
- CollectAlignmentSummaryMetrics
- CollectBaseDistributionByCycle
- CollectGcBiasMetrics
- CollectInsertSizeMetrics
- CollectMultipleMetrics
- CollectTargetedPcrMetrics
- CollectRnaSeqMetrics
- CollectWgsMetrics
- CompareSAMs
- CreateSequenceDictionary
- DownsampleSam
- ExtractIlluminaBarcodes
- EstimateLibraryComplexity
- FastqToSam
- FifoBuffer
- FilterSamReads
- FixMateInformation
- GatherBamFiles
- IlluminaBasecallsToFastq
- IlluminaBasecallsToSam
- CheckIlluminaDirectory
- IntervalListTools
- MakeSitesOnlyVcf
- MarkDuplicates
- MeanQualityByCycle
- MergeBamAlignment
- MergeSamFiles
- MergeVcfs
- NormalizeFasta
- ExtractSequences
- QualityScoreDistribution
- ReorderSam
- ReplaceSamHeader
- RevertSam
- RevertOriginalBaseQualitiesAndAddMateCigar
- SamFormatConverter
- SamToFastq
- SortSam
- VcfFormatConverter
- MarkIlluminaAdapters
- SplitVcfs
- ValidateSamFile
- ViewSam

http://picard.sourceforge.net/

Interval Manipulation

BED AND BEDTOOLS

BED track


Scale
chr12:
25,350,000
50 kb
25,400,000
hg19
RefSeq Genes
CASC1
CASC1
CASC1
CASC1
CASC1
LYRM5
KRAS
KRAS

BED format

- Header (space separated) [optional]
- Chr [mandatory]
- Start [mandatory]
- Stop [mandatory]
- Other (optional)
  - Name
  - Strand
  - Color
  - Type

A BAM alignment can be condensed into a BED file: Information loss !

BEDTOOLS

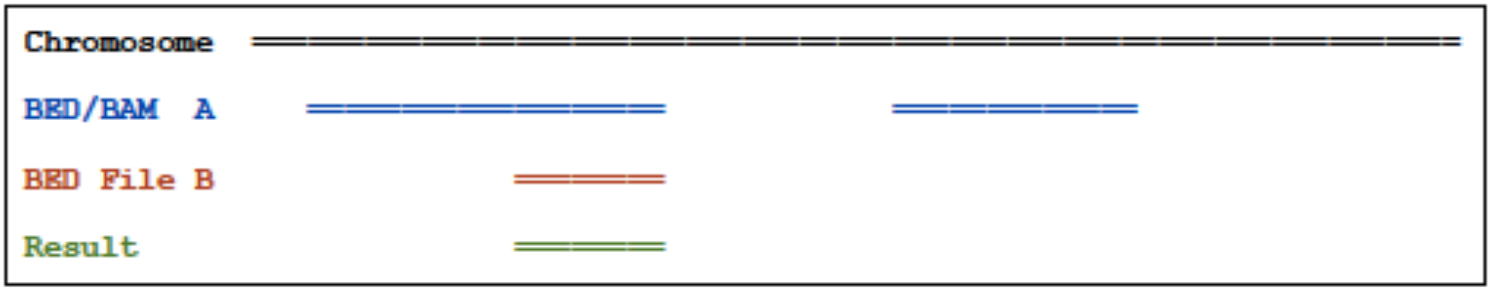
Utility	Description
intersectBed	Returns overlapping features between two BED/GFF/VCF files. *Also supports BAM format as input and output.*
windowBed	Returns overlapping features between two BED/GFF/VCF files within a "window". *Also supports BAM format as input and output.*
closestBed	Returns the closest feature to each entry in a BED/GFF/VCF file.
coverageBed	Summarizes the depth and breadth of coverage of features in one BED/GFF file (e.g., aligned reads) relative to another (e.g., user-defined windows). *Also supports BAM format as input and output.*
genomeCoverageBed	Histogram or a "per base" report of genome coverage. *Also supports BAM format as input and output.*
pairToBed	Returns overlaps between a BEDPE file and a regular BED/GFF/VCF file. *Also supports BAM format as input and output.*
pairToPair	Returns overlaps between two BEDPE files.
bamToBed	Converts BAM alignments to BED and BEDPE formats. *Also supports BAM format as input and output.*
bed12ToBam	Converts BED/GFF/VCF features (both blocked and unblocked) to BAM format.
bedToIgv	Creates a batch script to create IGV images at each interval defined in a BED/GFF/VCF file.
bed12ToBed6	Splits BED12 features into discrete BED6 features.
subtractBed	Removes the portion of an interval that is overlapped by another feature.
mergeBed	Merges overlapping features into a single feature.
fastaFromBed	Creates FASTA sequences from BED/GFF intervals.
maskFastaFromBed	Masks a FASTA file based upon BED/GFF coordinates.
shuffleBed	Permutes the locations of features within a genome.
slopBed	Adjusts features by a requested number of base pairs.
sortBed	Sorts BED/GFF files in useful ways.
linksBed	Creates an HTML links from a BED/GFF file.
complementBed	Returns intervals not spanned by features in a BED/GFF file.
overlap	Computes the amount of overlap (positive values) or distance (negative values) between genome features and reports the result at the end of the same line.
groupBy	Summarizes a dataset column based upon common column groupings. Akin to the SQL "group by" command.
unionBedGraphs	Combines multiple BedGraph files into a single file, allowing coverage/other comparisons between them.
annotateBed	Annotates one BED/VCF/GFF file with overlaps from many others.

IntersectBed

Usage: `$ intersectBed [OPTIONS] [-a <BED/GFF/VCF> || -abam <BAM>] -b <BED/GFF/VCF>`

Option	Description
-a	BED/GFF/VCF file A. Each feature in A is compared to B in search of overlaps. Use "stdin" if passing A with a UNIX pipe.
-b	BED/GFF/VCF file B. Use "stdin" if passing B with a UNIX pipe.
-abam	BAM file A. Each BAM alignment in A is compared to B in search of overlaps. Use "stdin" if passing A with a UNIX pipe: For example: `samtools view -b <BAM> | intersectBed -abam stdin -b genes.bed`
-ubam	Write uncompressed BAM output. The default is write compressed BAM output.
-bed	When using BAM input (-abam), write output as BED. The default is to write output in BAM when using -abam. For example: `intersectBed -abam reads.bam -b genes.bed -bed`
-wa	Write the original entry in A for each overlap.
-wb	Write the original entry in B for each overlap. Useful for knowing *what* A overlaps. **Restricted by -f and -r.**
-wo	Write the original A and B entries plus the number of base pairs of overlap between the two features. Only A features with overlap are reported. **Restricted by -f and -r.**
-wao	Write the original A and B entries plus the number of base pairs of overlap between the two features. However, A features w/o overlap are also reported with a NULL B feature and overlap = 0. **Restricted by -f and -r.**
-u	Write original A entry once if any overlaps found in B. In other words, just report the fact at least one overlap was found in B. **Restricted by -f and -r.**
-c	For each entry in A, report the number of hits in B while restricting to -f. Reports 0 for A entries that have no overlap with B. **Restricted by -f and -r.**
-v	Only report those entries in A that have **no overlap** in B. **Restricted by -f and -r.**
-f	Minimum overlap required as a fraction of A. Default is 1E-9 (i.e. 1bp).
-r	Require that the fraction of overlap be **reciprocal** for A and B. In other words, if -f is 0.90 and -r is used, this requires that B overlap at least 90% of A and that A **also** overlaps at least 90% of B.
-s	Force "strandedness". That is, only report hits in B that overlap A on the **same** strand. By default, overlaps are reported without respect to strand.
-split	Treat "split" BAM (i.e., having an "N" CIGAR operation) or BED12 entries as distinct BED intervals.

Intersect BED



For example:

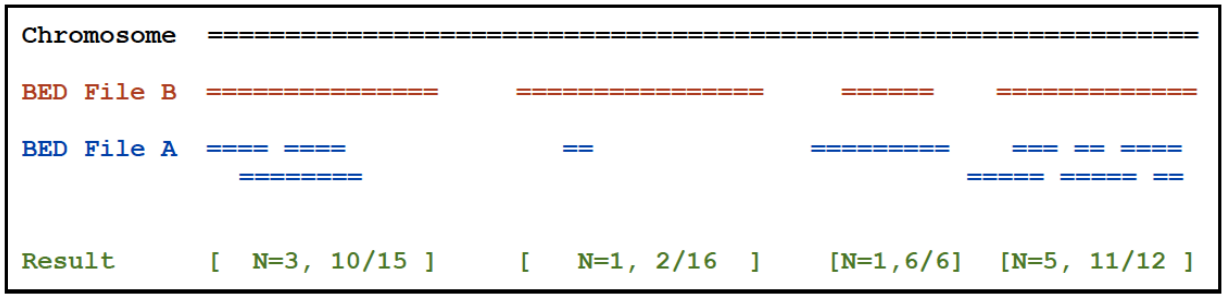
```
$ cat A.bed
chr1  100   200
chr1  1000  2000

$ cat B.bed
chr1  150   250

$ intersectBed -a A.bed -b B.bed
chr1  150   200
```

coverageBed

Usage: `$ coverageBed [OPTIONS] -a <BED/GFF/VCF> -b <BED/GFF/VCF>`

Option	Description
-abam	BAM file A. Each BAM alignment in A is compared to B in search of overlaps. Use "stdin" if passing A with a UNIX pipe: For example: `samtools view -b <BAM> | intersectBed -abam stdin -b genes.bed`
-s	Force strandedness. That is, only features in A are only counted towards coverage in B if they are the same strand. *By default, this is disabled and coverage is counted without respect to strand.*
-hist	Report a histogram of coverage for each feature in B as well as a summary histogram for _all_ features in B. Output (tab delimited) after each feature in B: 1) depth 2) # bases at depth 3) size of B 4) % of B at depth
-d	Report the depth at each position in each B feature. Positions reported are one based. Each position and depth follow the complete B feature.
-split	Treat "split" BAM or BED12 entries as distinct BED intervals when computing coverage. For BAM files, this uses the CIGAR "N" and "D" operations to infer the blocks for computing coverage. For BED12 files, this uses the BlockCount, BlockStarts, and BlockEnds fields (i.e., columns 10,11,12).

coverageBed



After each interval in B, **coverageBed** will report:

1) The number of features in A that overlapped (by at least one base pair) the B interval.
2) The number of bases in B that had non-zero coverage from features in A.
3) The length of the entry in B.
4) The fraction of bases in B that had non-zero coverage from features in A.

```
$ cat A.bed
chr1  10    20
chr1  20    30
chr1  30    40
chr1  100   200

$ cat B.bed
chr1  0     100
chr1  100   200
chr2  0     100

$ coverageBed -a A.bed -b B.bed
chr1  0     100   3     30    100   0.3000000
chr1  100   200   1     100   100   1.0000000
chr2  0     100   0     0     100   0.0000000
```

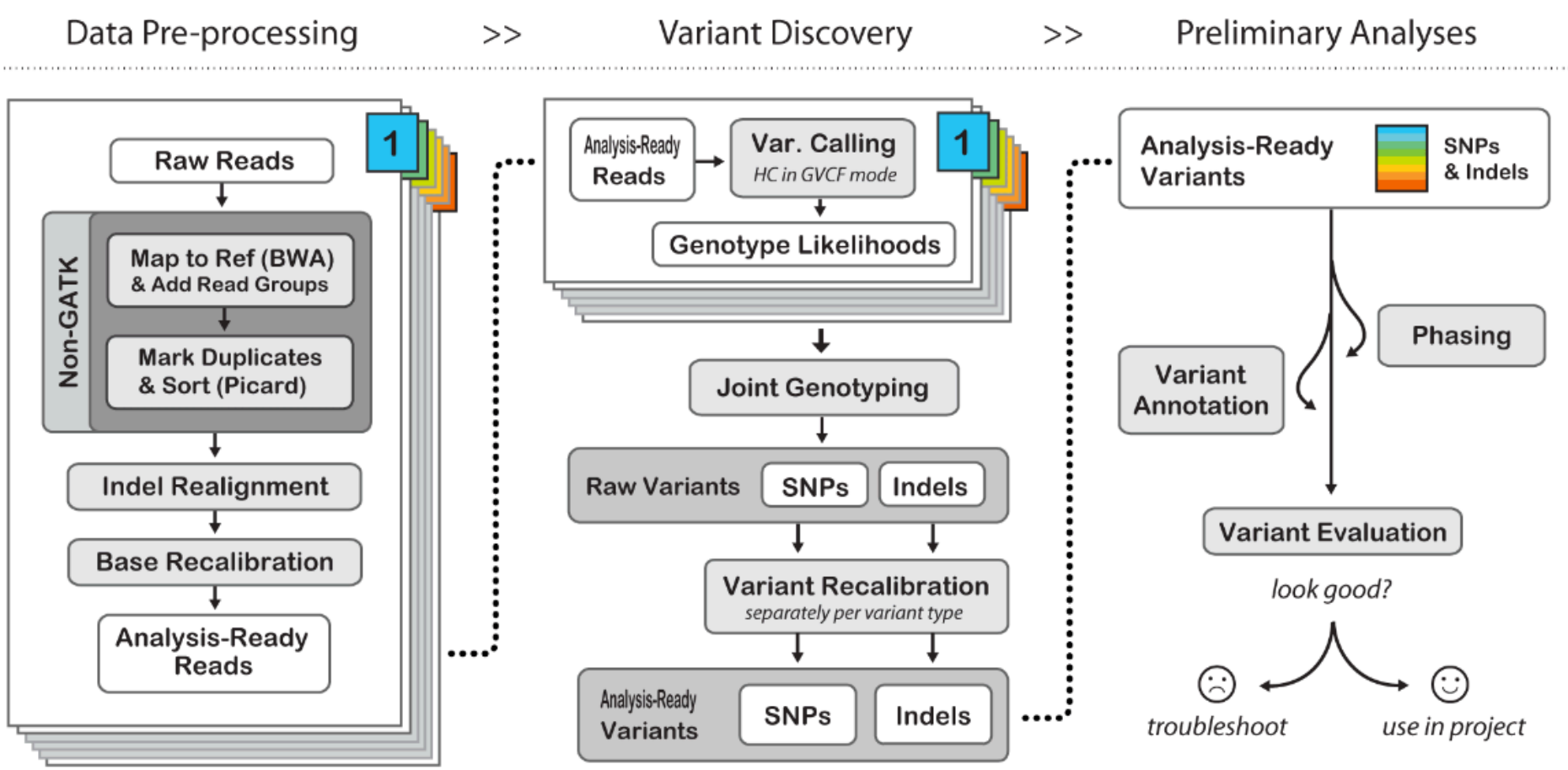
coverageBed

```
$ cat A.bed
chr1  10    20    a1    1    -
chr1  20    30    a2    1    -
chr1  30    40    a3    1    -
chr1  100   200   a4    1    +

$ cat B.bed
chr1  0     100   b1    1    +
chr1  100   200   b2    1    -
chr2  0     100   b3    1    +

$ coverageBed -a A.bed -b B.bed -hist
chr1  0     100   b1    1    +    0    70    100   0.7000000
chr1  0     100   b1    1    +    1    30    100   0.3000000
chr1  100   200   b2    1    -    1    100   100   1.0000000
chr2  0     100   b3    1    +    0    100   100   1.0000000
all   0     170   300   0.5666667
all   1     130   300   0.4333333
```

BAM postprocessing

GATK

GATK Best Practice



Diagnostics and Quality Control Tools

Name	Summary
AnalyzeCovariates	Tool to analyze and evaluate base recalibration tables.
BaseCoverageDistribution	Simple walker to plot the coverage distribution per base
CallableLoci	Emits a data file containing information about callable, uncallable, poorly mapped, and other parts of the genome
CheckPileup	Compare GATK's internal pileup to a reference Samtools pileup
CompareCallableLoci	Test routine for new VariantContext object
CountBases	Walks over the input data set, calculating the number of bases seen for diagnostic purposes.
CountIntervals	Count contiguous regions in an interval list.
CountLoci	Walks over the input data set, calculating the total number of covered loci for diagnostic purposes.
CountMales	Walks over the input data set, calculating the number of reads seen from male samples for diagnostic purposes.
CountRODs	Prints out counts of the number of reference ordered data objects encountered.
CountRODsByRef	Prints out counts of the number of reference ordered data objects encountered along the reference.
CountReadEvents	Walks over the input data set, counting the number of read events (from the CIGAR operator)
CountReads	Walks over the input data set, calculating the number of reads seen for diagnostic purposes.
CountTerminusEvent	Walks over the input data set, counting the number of reads ending in insertions/deletions or soft-clips
CoveredByNSamplesSites	Print intervals file with all the variant sites for which most of the samples have good coverage
DepthOfCoverage	Assess sequence coverage by a wide array of metrics, partitioned by sample, read group, or library
DiagnoseTargets	Analyzes coverage distribution and validates read mates for a given interval and sample.
DiffObjects	A generic engine for comparing tree-structured objects
ErrorRatePerCycle	Compute the read error rate per position
FastaStats	Calculate basic statistics about the reference sequence itself
FindCoveredIntervals	Outputs a list of intervals that are covered above a given threshold.
FlagStat	A reimplementation of the 'samtools flagstat' subcommand in the GATK
GCContentByInterval	Walks along reference and calculates the GC content for each interval.
Pileup	Emulates the samtools pileup command to print aligned reads
PrintRODs	Prints out all of the RODs in the input data set.
QCRef	Quality control for the reference fasta
QualifyMissingIntervals	Walks along reference and calculates a few metrics for each interval.
ReadClippingStats	Read clipping statistics for all reads.
ReadGroupProperties	Emits a GATKReport containing read group, sample, library, platform, center, sequencing data, paired end status, simple read type name (e.g.
ReadLengthDistribution	Outputs the read lengths of all the reads in a file.
SimulateReadsForVariants	Generates simulated reads for variants

Sequence Data Processing Tools

Name	Summary
BaseRecalibrator	First pass of the base quality score recalibration -- Generates recalibration table based on various user-specified covariates (such as read group, reported quality score, machine cycle, and nucleotide context).
ClipReads	Read clipping based on quality, position or sequence matching
IndelRealigner	Performs local realignment of reads to correct misalignments due to the presence of indels.
LeftAlignIndels	Left-aligns indels from reads in a bam file.
PrintReads	Renders, in SAM/BAM format, all reads from the input data set in the order in which they appear in the input file.
ReadAdaptorTrimmer	Utility tool to blindly strip base adaptors.
RealignerTargetCreator	Emits intervals for the Local Indel Realigner to target for realignment.
SplitNCigarReads	Splits reads that contain Ns in their cigar string (e.g.
SplitSamFile	Divides the input data set into separate BAM files, one for each sample in the input data set.

Variant Discovery Tools

Name	Summary
ApplyRecalibration	Applies cuts to the input vcf file (by adding filter lines) to achieve the desired novel truth sensitivity levels which were specified during VariantRecalibration
BeagleOutputToVCF	Takes files produced by Beagle imputation engine and creates a vcf with modified annotations.
GenotypeGVCFs	Genotypes any number of gVCF files that were produced by the Haplotype Caller into a single joint VCF file.
HaplotypeCaller	Call SNPs and indels simultaneously via local re-assembly of haplotypes in an active region.
PhaseByTransmission	Computes the most likely genotype combination and phases trios and parent/child pairs
ProduceBeagleInput	Converts the input VCF into a format accepted by the Beagle imputation/analysis program.
ReadBackedPhasing	Walks along all variant ROD loci, caching a user-defined window of VariantContext sites, and then finishes phasing them when they go out of range (using upstream and downstream reads).
UnifiedGenotyper	A variant caller which unifies the approaches of several disparate callers -- Works for single-sample and multi-sample data.
VariantRecalibrator	Create a Gaussian mixture model by looking at the annotations values over a high quality subset of the input call set and then evaluate all input variants.
VariantsToBeagleUnphased	Produces an input file to Beagle imputation engine, listing unphased, hard-called genotypes for a single sample in input variant file.

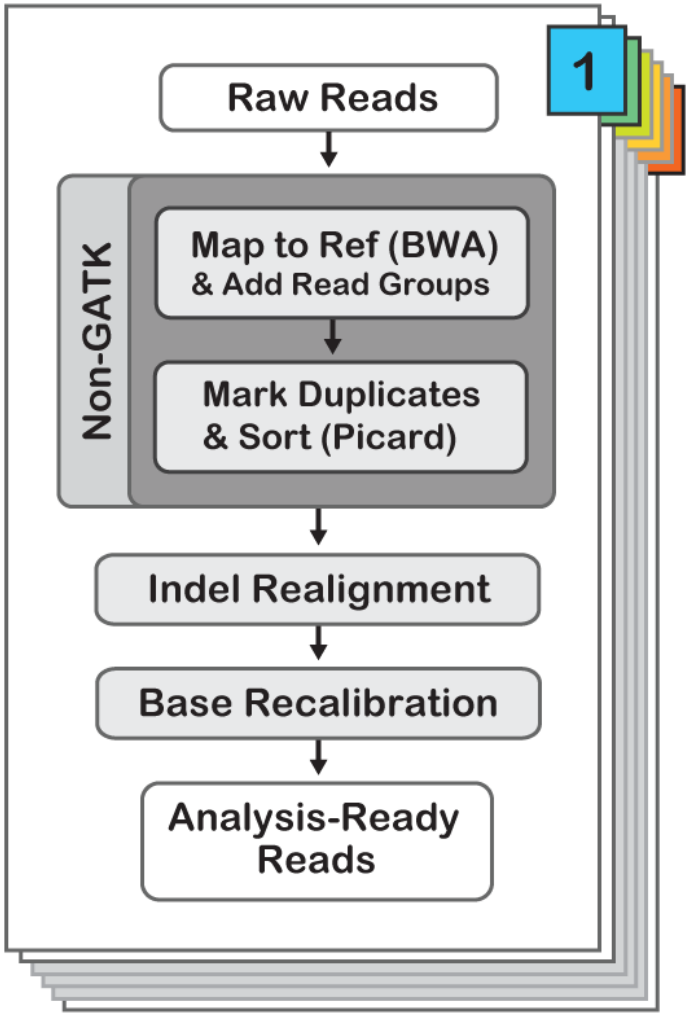
Variant Evaluation and Manipulation Tools

Name	Summary
CalculateGenotypePosteriors	Calculates genotype posterior likelihoods given panel data
CatVariants	Concatenates VCF files of non-overlapped genome intervals, all with the same set of samples
CombineGVCFs	Combines any number of gVCF files that were produced by the Haplotype Caller into a single joint gVCF file.
CombineVariants	Combines VCF records from different sources.
FilterLiftedVariants	Filters a lifted-over VCF file for ref bases that have been changed.
GenotypeConcordance	Genotype concordance (per-sample and aggregate counts and frequencies, NRD/NRS and site allele overlaps) between two callsets
HaplotypeResolver	Haplotype-based resolution of variants in 2 different eval files.
LeftAlignAndTrimVariants	Left-aligns indels from a variants file.
LiftoverVariants	Lifts a VCF file over from one build to another.
RandomlySplitVariants	Takes a VCF file, randomly splits variants into two different sets, and outputs 2 new VCFs with the results.
RegenotypeVariants	Regenotypes the variants from a VCF.
SelectHeaders	Selects headers from a VCF source.
SelectVariants	Selects variants from a VCF source.
VariantAnnotator	Annotates variant calls with context information.
VariantEval	General-purpose tool for variant evaluation (% in dbSNP, genotype concordance, Ti/Tv ratios, and a lot more)
VariantFiltration	Filters variant calls using a number of user-selectable, parameterizable criteria.
VariantsToAllelicPrimitives	Takes alleles from a variants file and breaks them up (if possible) into more basic/primitive alleles.
VariantsToBinaryPed	Converts a VCF file to a binary plink Ped file (.bed/.bim/.fam)
VariantsToTable	Emits specific fields from a VCF file to a tab-deliminated table
VariantsToVCF	Converts variants from other file formats to VCF format.

Variant Annotations

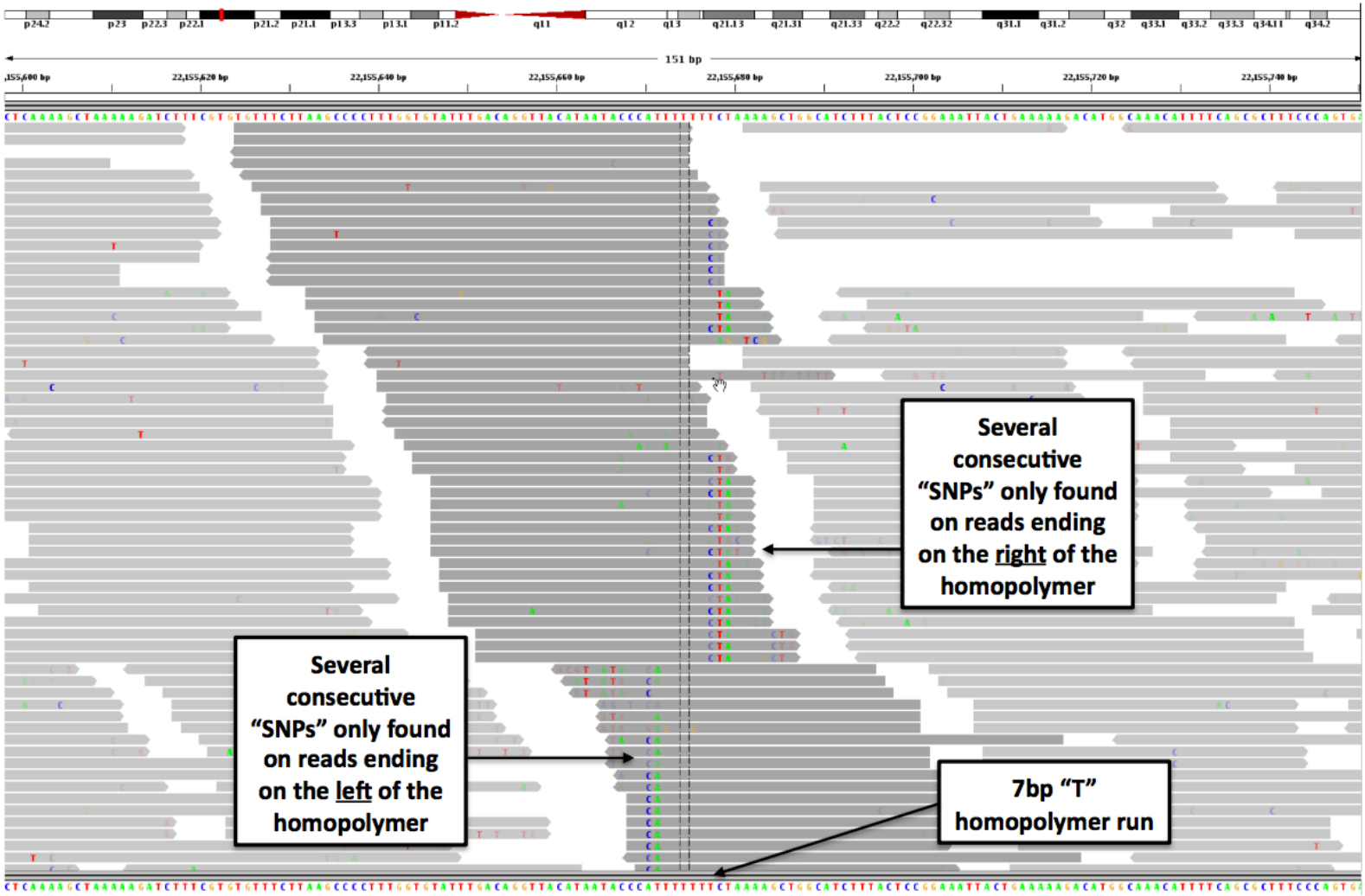
Annotations available to VariantAnnotator and the variant callers (some restrictions apply)

Name	Summary
AlleleBalance	Allele balance across all samples
AlleleBalanceBySample	Allele balance per sample
BaseCounts	Count of A, C, G, T bases across all samples
BaseQualityRankSumTest	U-based z-approximation from the Mann-Whitney Rank Sum Test for base qualities
ChromosomeCounts	Allele counts and frequency for each ALT allele and total number of alleles in called genotypes
ClippingRankSumTest	U-based z-approximation from the Mann-Whitney Rank Sum Test for reads with clipped bases
Coverage	Total (unfiltered) depth over all samples.
DepthPerAlleleBySample	The depth of coverage of each allele per sample
DepthPerSampleHC	The depth of coverage for informative reads for each sample.
FisherStrand	Phred-scaled p-value using Fisher's Exact Test to detect strand bias
GCContent	GC content of the reference around the given site
GenotypeSummaries	Created by rpoplin on 4/5/14.
HaplotypeScore	Consistency of the site with two (and only two) segregating haplotypes.
HardyWeinberg	Hardy-Weinberg test for disequilibrium
HomopolymerRun	Largest contiguous homopolymer run of the variant allele
InbreedingCoeff	Likelihood-based (using PL field) test for the inbreeding among samples.
LikelihoodRankSumTest	U-based z-approximation from the Mann-Whitney Rank Sum Test contrasting the likelihoods of reads to their most likely haplotypes.
LowMQ	Triplet annotation: fraction of MAQP == 0, MAPQ < 10, and count of all mapped reads
MVLikelihoodRatio	Likelihood of being a Mendelian Violation
MappingQualityRankSumTest	U-based z-approximation from the Mann-Whitney Rank Sum Test for mapping qualities
MappingQualityZero	Total count across all samples of mapping quality zero reads
MappingQualityZeroBySample	Count for each sample of mapping quality zero reads
NBaseCount	The number of N bases, counting only SOLiD data
PossibleDeNovo	Tags variants with called genotypes that support the existence of a de novo mutation in at least one of the given families
QualByDepth	Variant confidence (from the QUAL field) / unfiltered depth of non-reference samples.
RMSMappingQuality	Root Mean Square of the mapping quality of the reads across all samples.
ReadPosRankSumTest	U-based z-approximation from the Mann-Whitney Rank Sum Test for the distance from the end of the read for reads with the alternate allele
SampleList	List all of the polymorphic samples.
SnpEff	A set of genomic annotations based on the output of the SnpEff variant effect predictor tool
SpanningDeletions	Fraction of reads containing spanning deletions at this site
StrandBiasBySample	Per-sample component statistics which comprise the Fisher's Exact Test to detect strand bias User: rpoplin Date: 8/28/13
StrandOddsRatio	Symmetric Odds Ratio to detect strand bias
TandemRepeatAnnotator	Annotates variants that are composed of tandem repeats
TransmissionDisequilibriumTest	Wittkowski transmission disequilibrium test
VariantType	Assigns a roughly correct category of the variant type (SNP, MNP, insertion, deletion, etc.)


1
Raw Reads
Non-GATK
Map to Ref (BWA)
& Add Read Groups
Mark Duplicates
& Sort (Picard)
Indel Realignment
Base Recalibration
Analysis-Ready
Reads

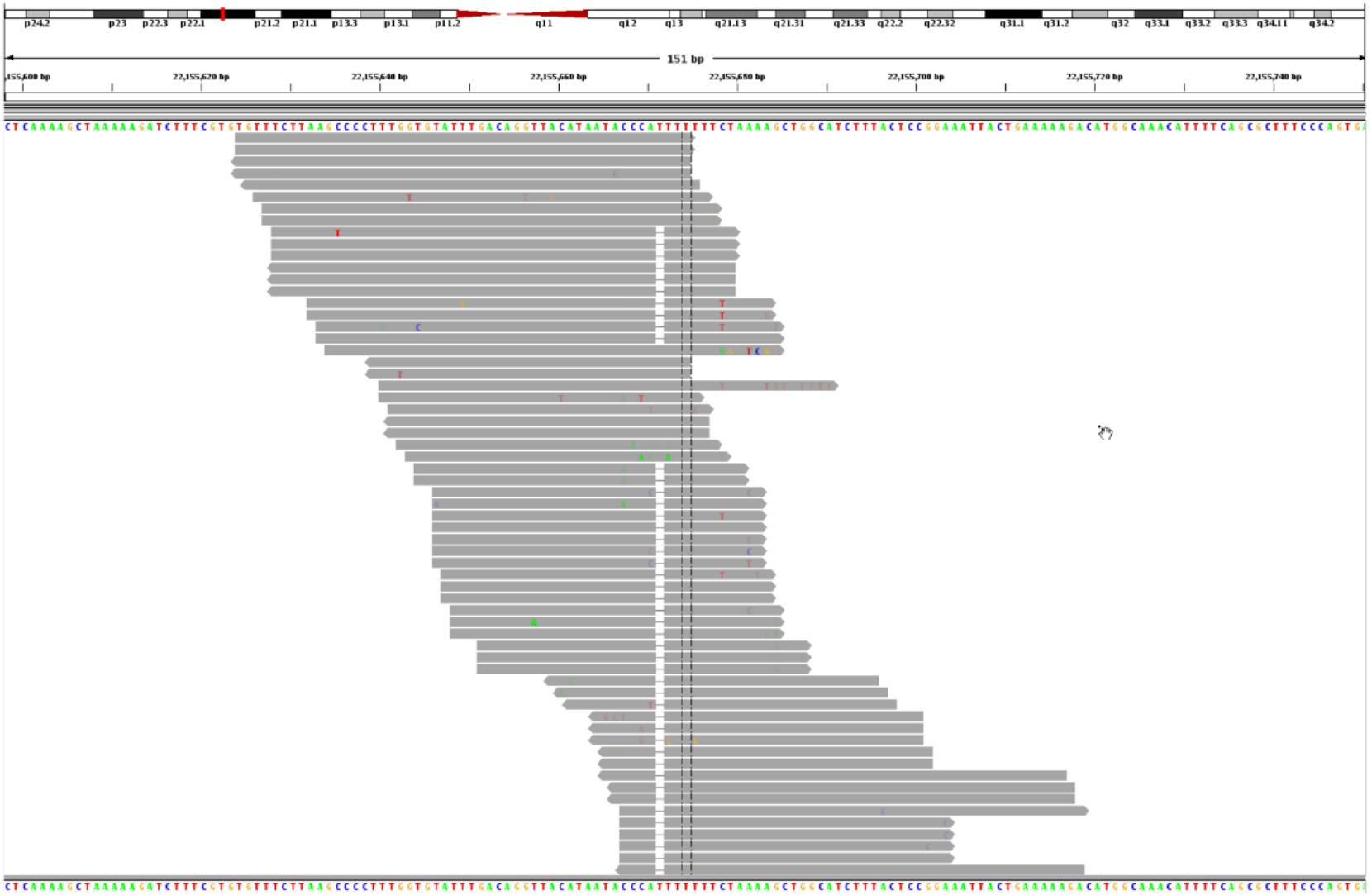
INDEL Realignment

- The problem



INDEL Realignment

- The problem

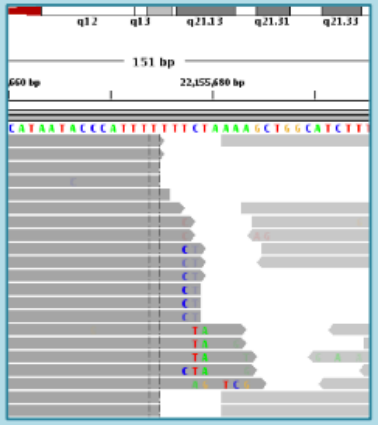


INDEL Realignment

- The solution

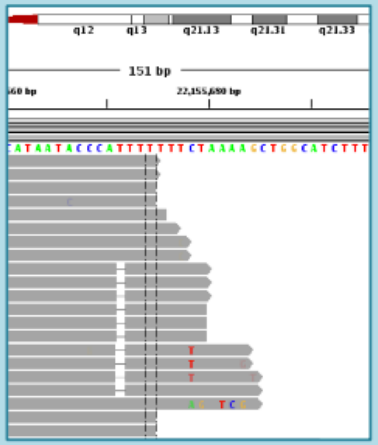
- Identify what regions need to be realigned

  → **RealignerTargetCreator**



- Perform the actual realignment

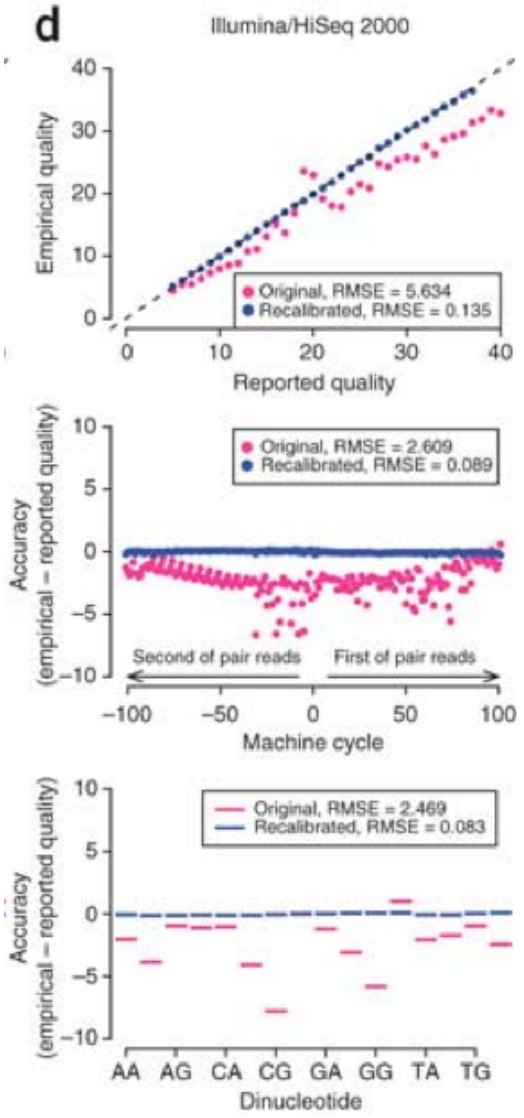
  → **IndelRealigner**



Base Quality Recalibration

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

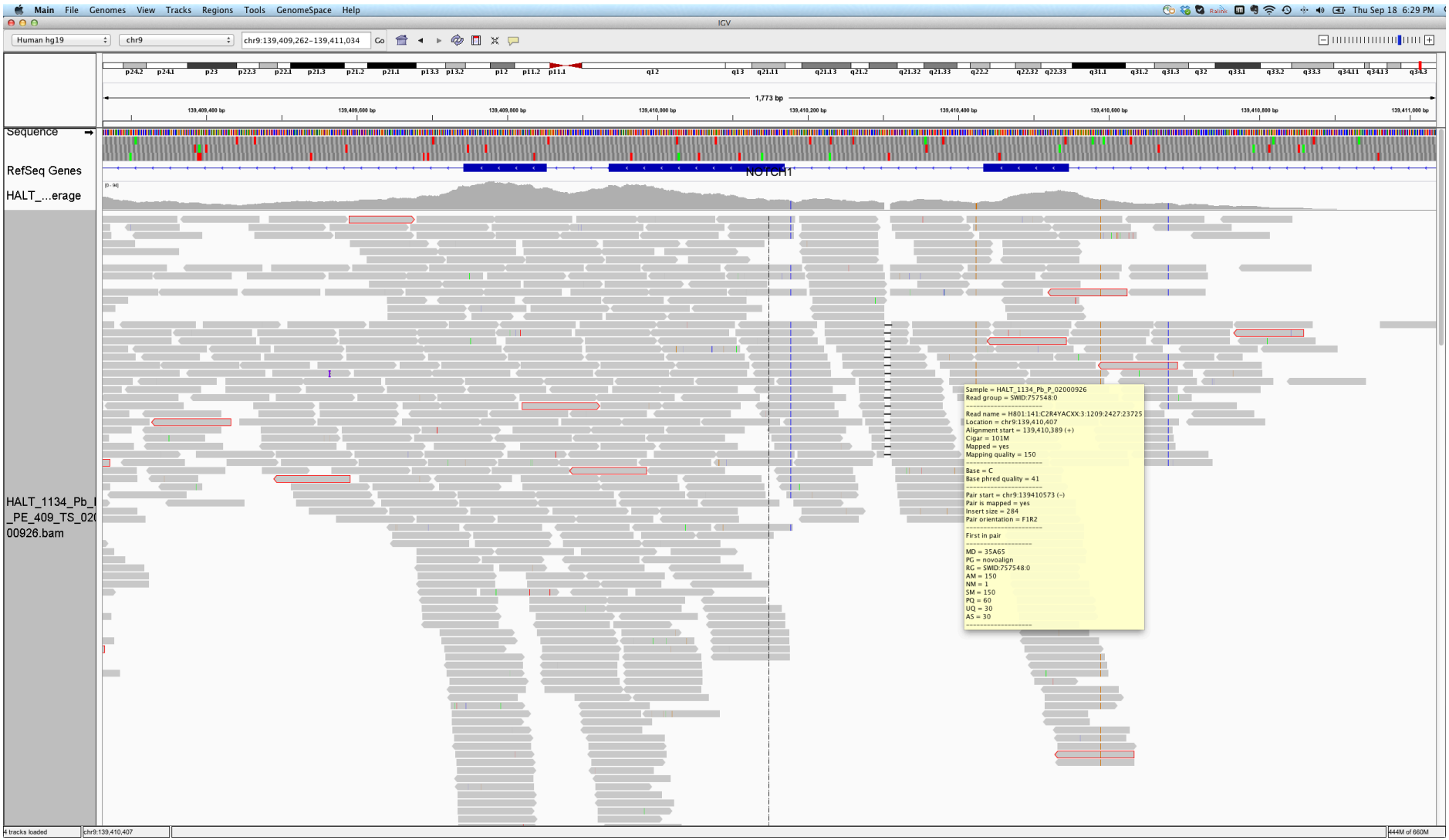
Phred Quality Score	Probability of incorrect base call	Base call accuracy
10	1 in 10	90 %
20	1 in 100	99 %
30	1 in 1000	99.9 %
40	1 in 10000	99.99 %
50	1 in 100000	99.999 %



DePristo et al (2011)

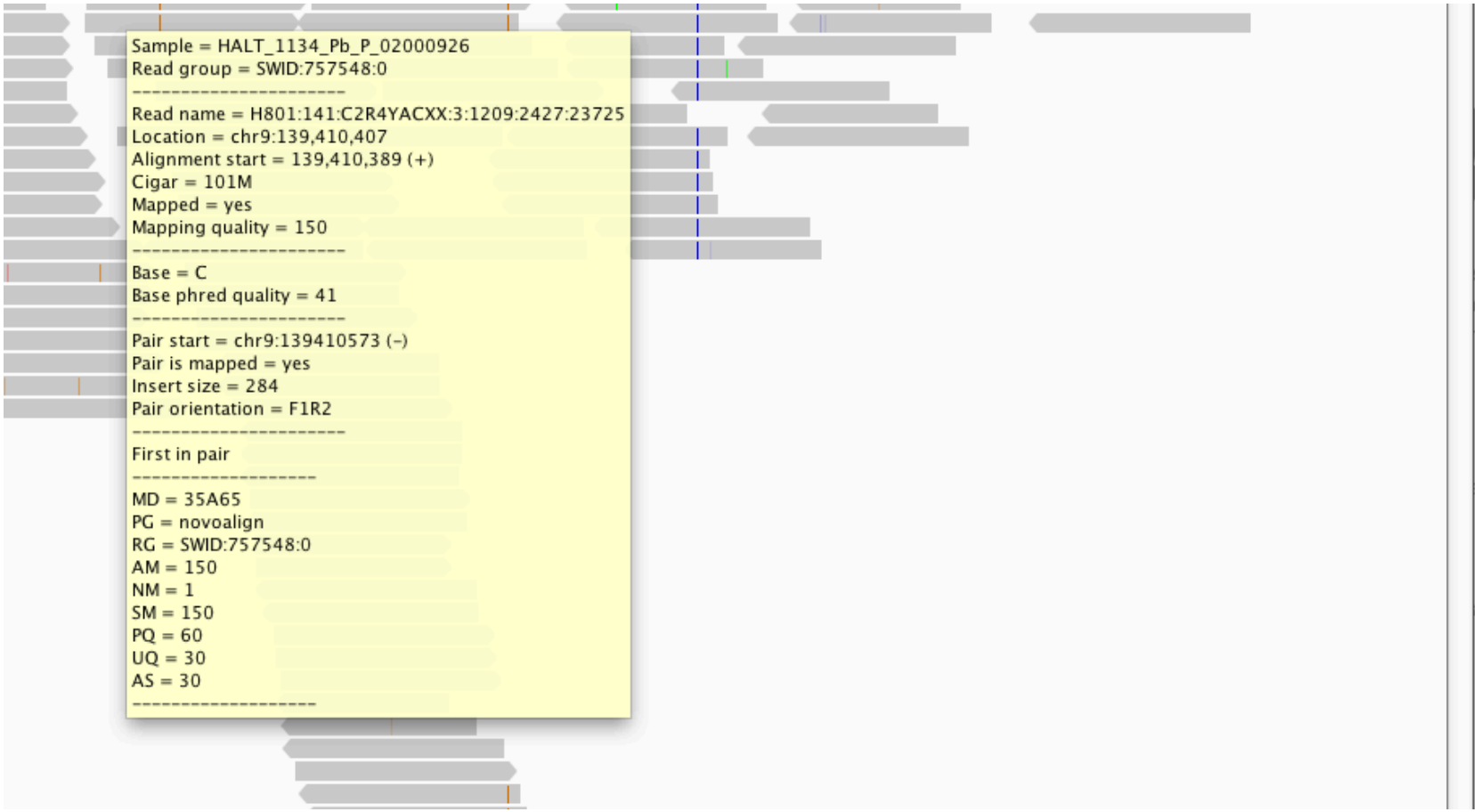
ALIGNMENT VISUALISATION

Integrative Genome Viewer



http://www.broadinstitute.org/software/igv/

Integrative Genome Viewer



http://www.broadinstitute.org/software/igv/

GATK/VARSCAN

VARIANT CALLING

Variant Calling

- Bayesian Statistics

  P(genotype|data)  P(data|genotype).P(genotype)

  – P(genotype) : prior probability for the genotype
  – P(data|genotype): likelihood for observed (called) allele

- Likelihood P(data|genotype): What's known to affect base calling:
  - Error rate increases as cycle numbers increase
  - Error rate depends on substitution type
  - Error rate depends on local sequence

GATK Callers

- UnifiedGenotyper

  Call SNPs and indels separately by considering each variant locus independently

- HaplotypeCaller

  Call SNPs, indels, and some SVs simultaneously by performing a local *de-novo* assembly

Somatic Variant Calling

- Compare Normal and Tumor directly

	Reference-supporting	Variant-supporting
Tumor Reads:	*tumor_reads1*	*tumor_reads2*
Normal Reads:	*normal_reads1*	*normal_reads2*

Fisher Exact test

- Results
  - Germline: Present in both
  - Somatic: Present in Tumor only
  - Loss of Heterozygosity: Missed Het in the tumor

Kobolt et al. (2012)

Heuristic Filters

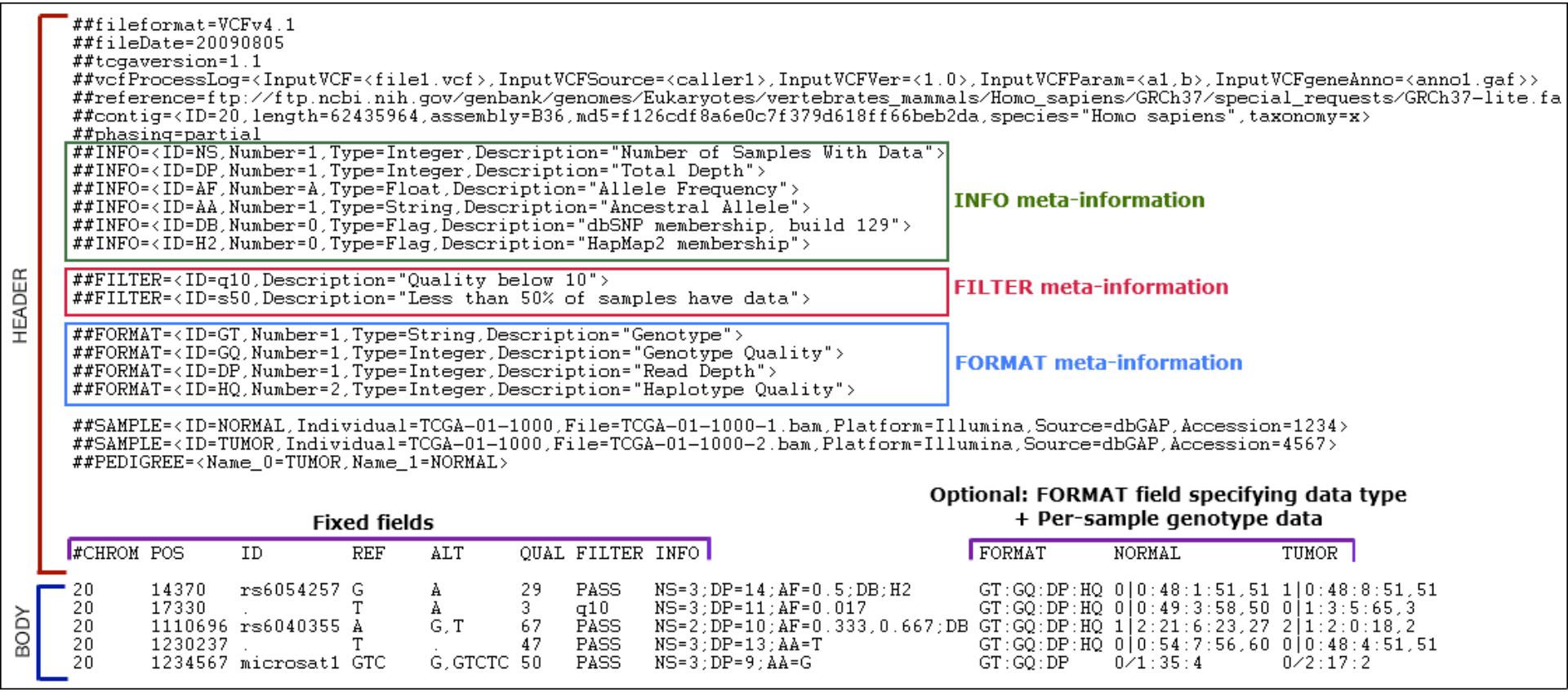
Table 1. Empirically derived filtering parameters for putative somatic mutations

Parameter	Description	Requirement
Read position	Average variant position in supporting reads, relative to read length	Between 10 and 90
Strandedness	Fraction of supporting reads from the forward strand	Between 1%–99%
Variant reads	Total number of reads supporting the variant	At least four
Variant frequency	Variant allele frequency inferred from read counts	At least 5%
Distance to 3′	Average distance to effective 3′ end of variant position in supporting reads	At least 20
Homopolymer	Number of bases in a flanking homopolymer matching one allele	Less than five
Map quality difference	Difference in average mapping quality between reference and variant reads	Less than 30
Read length difference	Difference in average trimmed read length between reference and variant reads	Less than 25
MMQS difference	Difference in average mismatch quality sum between variant and reference reads	Less than 100

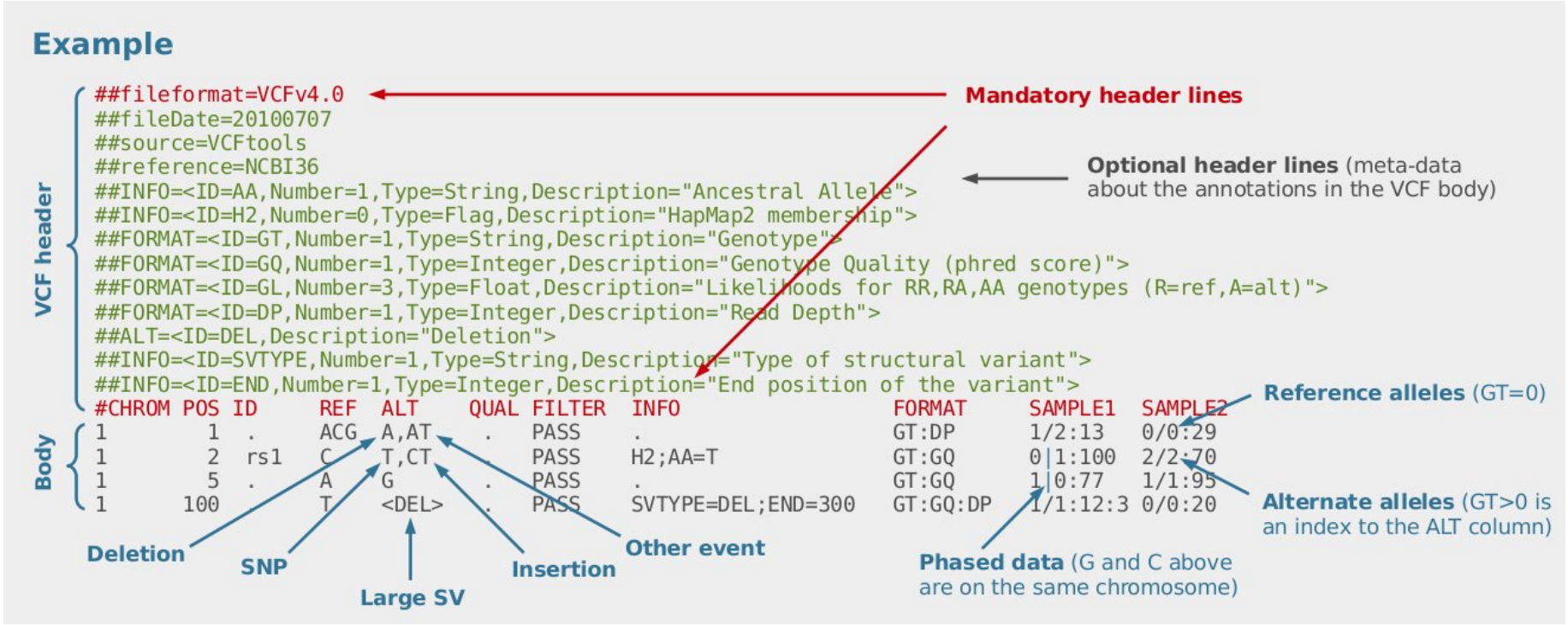
VCF / VCFTOOLS

VARIANT ANALYSIS

VCF format


HEADER
##fileformat=VCFv4.1
##fileDate=20090805
##tcgaversion=1.1
##vcfProcessLog=<InputVCF=<file1.vcf>,InputVCFSource=<caller1>,InputVCFVer=<1.0>,InputVCFParam=<a1,b>,InputVCFgeneAnno=<anno1.gaf>>
##reference=ftp://ftp.ncbi.nih.gov/genbank/genomes/Eukaryotes/vertebrates_mammals/Homo_sapiens/GRCh37/special_requests/GRCh37-lite.fa
##contig=<ID=20,length=62435964,assembly=B36,md5=f126cdf8a6e0c7f379d618ff66beb2da,species="Homo sapiens",taxonomy=x>
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
INFO meta-information
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
FILTER meta-information
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
FORMAT meta-information
##SAMPLE=<ID=NORMAL,Individual=TCGA-01-1000,File=TCGA-01-1000-1.bam,Platform=Illumina,Source=dbGAP,Accession=1234>
##SAMPLE=<ID=TUMOR,Individual=TCGA-01-1000,File=TCGA-01-1000-2.bam,Platform=Illumina,Source=dbGAP,Accession=4567>
##PEDIGREE=<Name_0=TUMOR,Name_1=NORMAL>
Fixed fields
Optional: FORMAT field specifying data type
+ Per-sample genotype data
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT NORMAL TUMOR
BODY
20 14370 rs6054257 G A 29 PASS NS=3;DP=14;AF=0.5;DB;H2 GT:GQ:DP:HQ 0|0:48:1:51,51 1|0:48:8:51,51
20 17330 . T A 3 q10 NS=3;DP=11;AF=0.017 GT:GQ:DP:HQ 0|0:49:3:58,50 0|1:3:5:65,3
20 1110696 rs6040355 A G,T 67 PASS NS=2;DP=10;AF=0.333,0.667;DB GT:GQ:DP:HQ 1|2:21:6:23,27 2|1:2:0:18,2
20 1230237 . T . 47 PASS NS=3;DP=13;AA=T GT:GQ:DP:HQ 0|0:54:7:56,60 0|0:48:4:51,51
20 1234567 microsat1 GTC G,GTCTC 50 PASS NS=3;DP=9;AA=G GT:GQ:DP 0/1:35:4 0/2:17:2

VCF format


Example
VCF header
Body
##fileformat=VCFv4.0
##fileDate=20100707
##source=VCFtools
##reference=NCBI36
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality (phred score)">
##FORMAT=<ID=GL,Number=3,Type=Float,Description="Likelihoods for RR,RA,AA genotypes (R=ref,A=alt)">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##ALT=<ID=DEL,Description="Deletion">
##INFO=<ID=SVTYPE,Number=1,Type=String,Description="Type of structural variant">
##INFO=<ID=END,Number=1,Type=Integer,Description="End position of the variant">
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT SAMPLE1 SAMPLE2
1 1 . ACG A,AT . PASS . GT:DP 1/2:13 0/0:29
1 2 rs1 C T,CT . PASS H2;AA=T GT:GQ 0|1:100 2/2:70
1 5 . A G . PASS . GT:GQ 1|0:77 1/1:95
1 100 T <DEL> . PASS SVTYPE=DEL;END=300 GT:GQ:DP 1/1:12:3 0/0:20
Mandatory header lines
Optional header lines (meta-data about the annotations in the VCF body)
Reference alleles (GT=0)
Alternate alleles (GT>0 is an index to the ALT column)
Deletion
SNP
Large SV
Insertion
Other event
Phased data (G and C above are on the same chromosome)

VCFLib

comparison

- Generate **haplotype-aware intersections** (vcfintersect -i), **unions** (vcfintersect -u), and **complements** (vcfintersect -v -i).
- **Overlay-merge** multiple VCF files together, using provided order as precedence (vcfoverlay).
- **Combine** multiple VCF files together, handling samples when alternate allele descriptions are identical (vcfcombine).
- **Validate** the integrity and identity of the VCF by verifying that the VCF record's REF matches a given reference file (vcfcheck).

format conversion

- Convert a VCF file into a per-allele or per-genotype **tab-separated (.tsv)** file (vcf2tsv).
- Store a VCF file in an **SQLite3** database (vcf2sqlite.py).
- Make a **BED file** from the intervals in a VCF file (vcf2bed.py).

filtering and subsetting

- **Filter** variants and genotypes using arbitrary expressions based on values in the INFO and sample fields (vcffilter).
- **Randomly sample** a subset of records from a VCF file, given a rate (vcfrandomsample).
- **Select variants** of a certain type (vcfsnps, vcfbiallelic, vcfindels, vcfcomplex, etc.)

annotation

- **Annotate** one VCF file with fields from the INFO column of another, based on position (vcfaddinfo, vcfintersect).
- Incorporate annotations or targets provided by a *BED* file (vcfannotate, vcfintersect).
- Examine **genotype correspondence** between two VCF files by annotating samples in one file with genotypes from another (vcfannotategenotypes).
- Annotate variants with the **distance** to the nearest variant (vcfdistance).
- Count the number of alternate alleles represented in samples at each variant record (vcfaltcount).
- **Subset INFO fields** to decrease file size and processing time (vcfkeepinfo).
- Lighten up VCF files by keeping only a **subset of per-sample information** (vcfkeepgeno).
- **Numerically index** alleles in a VCF file (vcfindex).

VCFLib

samples

- Quickly obtain the **list of samples** in a given VCF file (vcfsamplenames).
- **Remove samples** from a VCF file (vcfkeepsamples, vcfremovesamples).

ordering

- **Sort variants** by genome coordinate (vcfstreamsort).
- **Remove duplicate** variants in vcfstreamsort'ed files according to their REF and ALT fields (vcfuniq).

variant representation

- **Break multiallelic** records into multiple records (vcfbreakmulti), retaining allele-specific INFO fields.
- **Combine overlapping biallelic** records into a single record (vcfcreatemulti).
- **Decompose complex variants** into a canonical SNP and indel representation (vcfallelicprimitives), generating phased genotypes for available samples.
- **Reconstitute complex variants** provided a phased VCF with samples (vcfgeno2haplo).
- **Left-align indel and complex variants** (vcfleftalign).

genotype manipulation

- **Set genotypes** in a VCF file provided genotype likelihoods in the GL field (vcfglxgt).
- Establish putative **somatic variants** using reported differences between germline and somatic samples (vcfsamplediff).
- Remove samples for which the reported genotype (GT) and observation counts disagree (AO, RO) (vcfremoveaberrantgenotypes).

VCFLib

interpretation and classification of variants

- Obtain aggregate **statistics** about VCF files (vcfstats).
- Print the **receiver-operating characteristic (ROC)** of one VCF given a truth set (vcfroc).
- Annotate VCF records with the **Shannon entropy** of flanking sequence (vcfentropy).
- Calculate the heterozygosity rate (vcfhetcount).
- Generate potential **primers** from VCF records (vcfprimers), to check for genome uniqueness.
- Convert the numerical represenation of genotypes provided by the GT field to a **human-readable genotype format** (vcfgenotypes).
- Observe how different alignment parameters, including context and entropy-dependent ones, influence **variant classification and interpretation** (vcfremap).
- **Classify variants** by annotations in the INFO field using a self-organizing map (vcfsom); **re-estimate their quality** given known variants.

BCFTOOLS

LIST OF COMMANDS

For a full list of available commands, run **bcftools** without arguments. For a full list of available options, run **bcftools** *COMMAND* without arguments.

LIST OF COMMANDS

- **annotate** .. edit VCF files, add or remove annotations
- **call** .. SNP/indel calling (former "view")
- **cnv** .. Copy Number Variation caller
- **concat** .. concatenate VCF/BCF files from the same set of samples
- **consensus** .. create consensus sequence by applying VCF variants
- **convert** .. convert VCF/BCF to other formats and back
- **csq** .. haplotype aware consequence caller
- **filter** .. filter VCF/BCF files using fixed thresholds
- **gtcheck** .. check sample concordance, detect sample swaps and contamination
- **index** .. index VCF/BCF
- **isec** .. intersections of VCF/BCF files
- **merge** .. merge VCF/BCF files files from non-overlapping sample sets
- **mpileup** .. multi-way pileup producing genotype likelihoods
- **norm** .. normalize indels
- **plugin** .. run user-defined plugin
- **polysomy** .. detect contaminations and whole-chromosome aberrations
- **query** .. transform VCF/BCF into user-defined formats
- **reheader** .. modify VCF/BCF header, change sample names
- **roh** .. identify runs of homo/auto-zygosity
- **stats** .. produce VCF/BCF stats (former vcfcheck)
- **view** .. subset, filter and convert VCF and BCF files

VARIANT ANNOTATION

Variant Annotation Tools

- SNPEff (Broad)
  - http://snpeff.sourceforge.net
- VEP (Sanger)
  - http://www.ensembl.org/info/docs/tools/vep/index.html
- ANNOVAR
  - http://www.openbioinformatics.org/annovar/
- Variant Tools (database / Manipulation)
  - http://varianttools.sourceforge.net

McCarty et al: http://genomemedicine.com/content/6/3/26

ANNOVAR

exonic_variant_function

Value	Default precedence	Explanation
exonic	1	variant overlaps a coding exon
splicing	1	variant is within 2-bp of a splicing junction (use -splicing_threshold to change this)
ncRNA	2	variant overlaps a transcript without coding annotation in the gene definition (see Notes below for more explanation)
UTR5	3	variant overlaps a 5' untranslated region
UTR3	3	variant overlaps a 3' untranslated region
intronic	4	variant overlaps an intron
upstream	5	variant overlaps 1-kb region upstream of transcription start site
downstream	5	variant overlaps 1-kb region downtream of transcription end site (use -neargene to change this)
intergenic	6	variant is in intergenic region

Annotation	Precedence	Explanation
frameshift insertion	1	an insertion of one or more nucleotides that cause frameshift changes in protein coding seque
frameshift deletion	2	a deletion of one or more nucleotides that cause frameshift changes in protein coding sequenc
frameshift block substitution	3	a block substitution of one or more nucleotides that cause frameshift changes in protein codin
stopgain	4	a nonsynonymous SNV, frameshift insertion/deletion, nonframeshift insertion/deletion or block variant will not be counted as "stopgain"!
stoploss	5	a nonsynonymous SNV, frameshift insertion/deletion, nonframeshift insertion/deletion or block
nonframeshift insertion	6	an insertion of 3 or multiples of 3 nucleotides that do not cause frameshift changes in protein
nonframeshift deletion	7	a deletion of 3 or mutliples of 3 nucleotides that do not cause frameshift changes in protein co
nonframeshift block substitution	8	a block substitution of one or more nucleotides that do not cause frameshift changes in protei
nonsynonymous SNV	9	a single nucleotide change that cause an amino acid change
synonymous SNV	10	a single nucleotide change that does not cause an amino acid change
unknown	11	unknown function (due to various errors in the gene structure definition in the database file)

Same transcripts, different software:ANNOVAR and VEP annotations for exonic variants

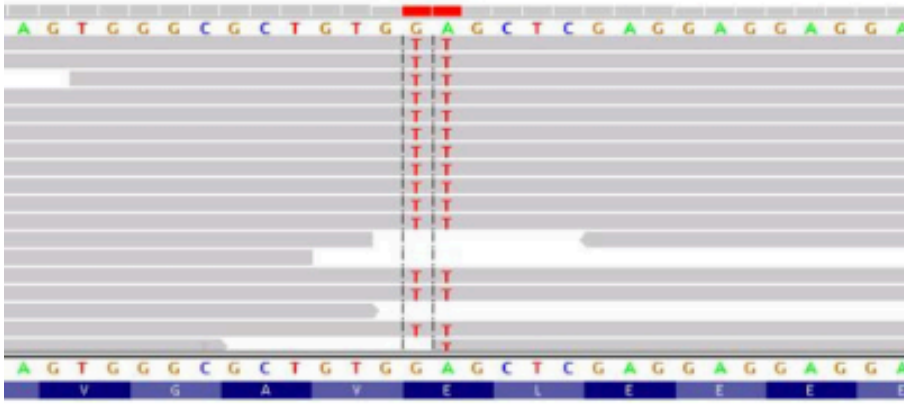
	ANV + VEP	ANV	VEP	Exact match	Category match	ANV match rate (%)	VEP match rate (%)	Overall category match rate (%)	Overall exact match rate (%)
LOF total	104,915	77,527	96,761	68,284	69,373	88.08	70.57	66.12	65.09
Frameshift	19,021	15,822	16,685	13,486	-	85.24	80.83	-	70.90
Stop gained	16,758	14,960	16,146	14,348	-	95.91	88.86	-	85.62
Stop lost	1,113	906	1,077	870	-	96.03	80.78	-	78.17
All splicing	69,112	45,839	62,853	39,580	-	86.35	62.97	-	57.27
MISSENSE total	350,806	324,242	347,752	318,056	321,188	98.09	91.46	91.56	90.66
Inframe indel	9,455	8,650	6,600	5,795	-	66.99	87.80	-	61.29
Missense	343,284	315,592	339,953	312,261	-	98.94	91.85	-	90.96
Initiator codon	1,199	0	1,199	0	-	-	0.00	-	0.00
SYNONYMOUS and OTHER CODING total	182,120	172,463	175,483	165,643	165,826	96.05	94.39	91.05	90.95
Synonymous	181,873	172,463	175,053	165,643	-	96.05	94.62	-	91.08
Stop retained	203	0	203	0	-	-	0.00	-	0.00
Other coding	227	0	227	0	-	-	0.00	-	0.00
ALL LOF	104,915	77,527	96,761	68,284	69,373	88.08	70.57	66.12	65.09
ALL LOF and MISSENSE	455,721	401,769	444,513	386,340	390,561	96.16	86.91	85.70	84.78
ALL EXONIC	637,841	574,232	619,996	551,983	556,387	96.13	89.03	87.23	86.54

McCarty et al: http://genomemedicine.com/content/6/3/26

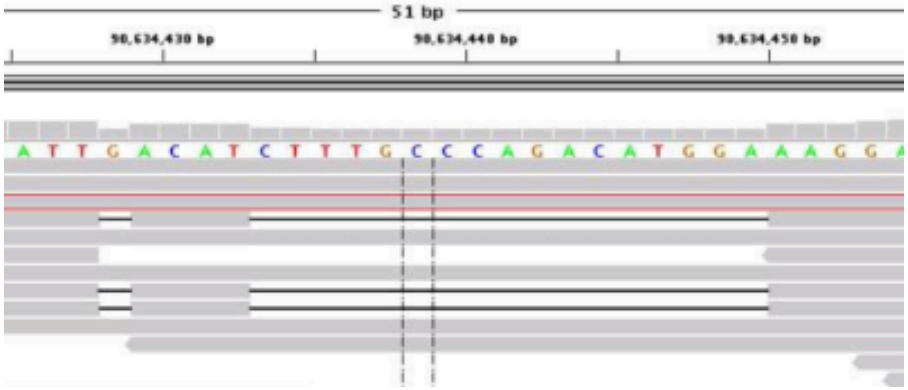
Annotation Challenges

- Compensating mutations
  DNP or TNP need to be annotated together

  Phase of Indels on the same haplotype



- Partial effects
  Mutation affecting one transcript only
- Same gene
  Unknown Loss of Function mutations affecting known cancer genes



Adapted from MacArthur et al, 2012

Useful Annotations

- dbSNP
- dbNSFP
  - SIFT
  - MutationTaster
- COSMIC
- ESP data
- 1000G data
- ExAC

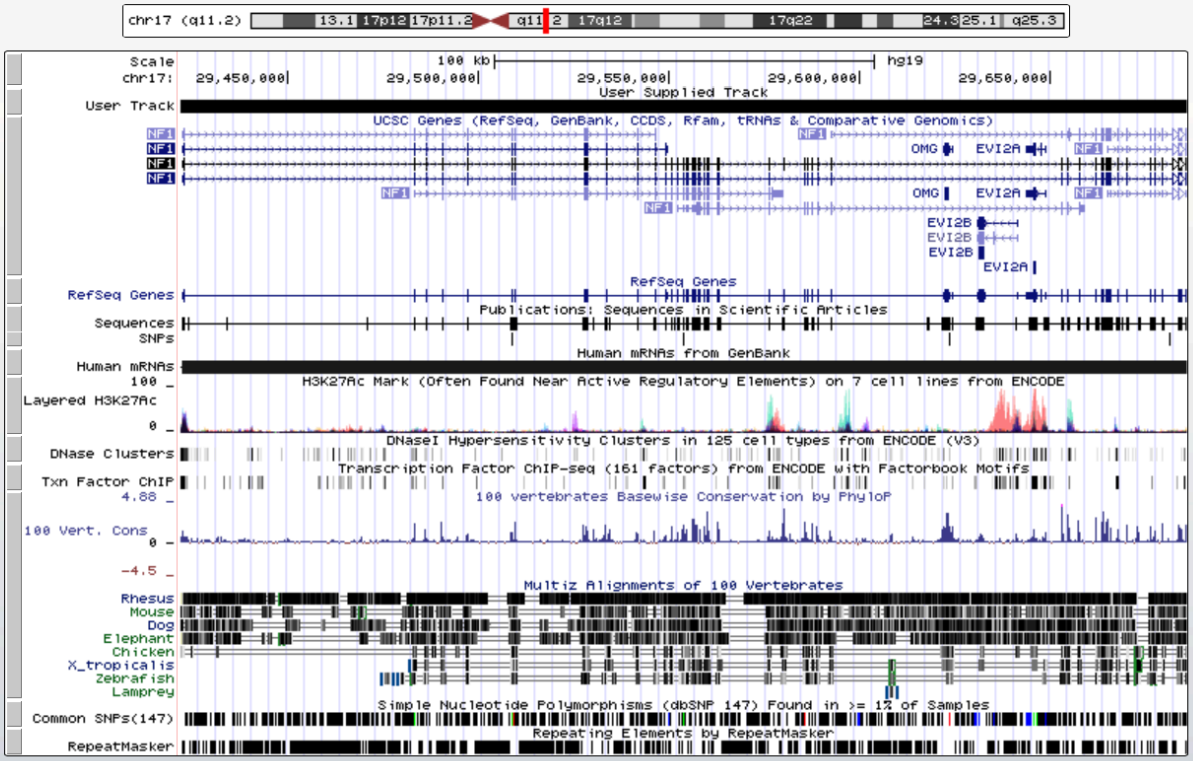
GENOME BROWSER

Genomes | Genome Browser | Tools | Mirrors | Downloads | My Data | View | Help | About Us

UCSC Genome Browser on Human Feb. 2009 (GRCh37/hg19) Assembly

move <<< << < > >> >>> zoom in 1.5x 3x 10x base zoom out 1.5x 3x 10x 100x

chr17:29,421,745-29,685,765 264,021 bp. enter position, gene symbol or search terms go



move start < 2.0 >

Click on a feature for details. Click or drag in the base position track to zoom in. Click side bars for track options. Drag side bars or labels up or down to reorder tracks. Drag tracks left or right to new position. Press "?" for keyboard shortcuts.

move end < 2.0 >

track search | default tracks | default order | hide all | manage custom tracks | track hubs | configure | multi-region | reverse | resize | refresh

My Data | View | Help

- My Sessions s s
- Public Sessions p s
- Track Hubs t h
- Custom Tracks c t

Downloads | My Data

- Genome Data
- Source Code
- Genome Browser Store
- Utilities
- FTP
- MySQL Access

Tools | Mirrors | Downlo

- Blat t b
- Table Browser t t
- Variant Annotation Integrator
- Data Integrator
- Gene Sorter
- Genome Graphs
- In-Silico PCR t i
- LiftOver
- VisiGene
- Other Utilities

Expression

refresh

GTEx hide	Affy Exon Array hide	Affy GNF1H hide	Affy RNA Loc hide	Affy U133 hide	Affy U133Plus2 hide
Affy U95 hide	Allen Brain hide	Burge RNA-seq hide	CSHL Small RNA-seq hide	ENC Exon Array... hide	ENC ProtGeno... hide
ENC RNA-seq... hide	GIS RNA PET hide	GNF Atlas 2 hide	GWIPS-viz Riboseq hide	Illumina WG-6 hide	PeptideAtlas hide
qPCR Primers hide	RIKEN CAGE Loc hide	Sestan Brain hide			

Regulation

refresh

ENCODE Regulation... show	CD34 DnaseI hide	CpG Islands... hide	ENC Chromatin... hide	ENC DNA Methyl... hide	ENC DNase/FAIRE... hide
ENC Histone... hide	ENC RNA Binding... hide	ENC TF Binding... hide	FSU Repli-chip hide	Genome Segments hide	NKI Nuc Lamina... hide
ORegAnno hide	Stanf Nucleosome hide	SUNY SwitchGear hide	SwitchGear TSS hide	TFBS Conserved hide	TS miRNA sites hide
UCSF Brain Methyl hide	UMMS Brain Hist hide	UW Repli-seq hide	Vista Enhancers hide		

Comparative Genomics

refresh

Conservation full	Cons 46-Way hide	Cons Indels MmCf hide	Evo Cpg hide	GERP hide	phastBias gBGC hide
Primate Chain/Net hide	Placental Chain/Net hide	Vertebrate Chain/Net hide			

ENC DNase/FAIRE Super-track Settings

ENCODE Open Chromatin by DNaseI HS and FAIRE Tracks (▲All Regulation tracks)

Display mode: hide Submit

All

☑	dense	Master DNaseI HS	DNaseI Hypersensitive Site Master List (125 cell types) from ENCODE/Analysis
☑	dense	Uniform DNaseI HS	DNaseI Hypersensitivity Uniform Peaks from ENCODE/Analysis *ENCODE Jan 2011 Freeze (Sept 2012 Analysis Pubs)*
☑	dense	Open Chrom Synth	DNaseI/FAIRE/ChIP Synthesis from ENCODE/OpenChrom(Duke/UNC/UTA) *ENCODE Mar 2012 Freeze*
☑	dense	Duke DNaseI HS	Open Chromatin by DNaseI HS from ENCODE/OpenChrom(Duke University) *ENCODE July 2012 Freeze*
☑	dense	UNC FAIRE	Open Chromatin by FAIRE from ENCODE/OpenChrom(UNC Chapel Hill) *ENCODE July 2012 Freeze*
☑	dense	UW DNaseI DGF	DNaseI Digital Genomic Footprinting from ENCODE/University of Washington *ENCODE July 2012 Freeze*
☑	dense	UW DNaseI HS	DNaseI Hypersensitivity by Digital DNaseI from ENCODE/University of Washington *ENCODE July 2012 Freeze*

hide
✓dense
squish
pack
full

Description

These tracks display evidence of open chromatin in ENCODE cell types. Open chromatin describes segments of DNA that are unpacked and accessible to the regulatory factors, enzymes, and smaller molecules in the cell. This is in contrast to closed chromatin, which is packed and inaccessible. Transcriptionally-active chromatin tends to be more open, while condensed, densely-packed chromatin tends to be silent.

Open chromatin was identified using complementary methods including: DNaseI hypersensitivity (HS), Formaldehyde-Assisted Isolation of Regulatory Elements (FAIRE), and chromatin immunoprecipitation (ChIP) for select regulatory factors.

DNaseI HS: DNaseI is an enzyme that has long been used to map general chromatin accessibility, and DNaseI "hyperaccessibility" or "hypersensitivity" is a feature of active cis-regulatory sequences. The use of this method has led to the discovery of functional regulatory elements that include enhancers, silencers, insulators, promotors, locus control regions and novel elements. DNaseI hypersensitivity signifies chromatin accessibility following binding of trans-acting factors in place of a canonical nucleosome.

Step 1. Format the data set
Formulate your data set as a tab-separated file using one of the formats supported by the Genome Browser. Annotation data can be in standard GFF format or in a format designed specifically for the Human Genome Project or UCSC Genome Browser, including bedGraph, GTF, PSL, BED, bigBed, WIG, bigGenePred, bigMaf, bigChain, bigPsl, bigWig, BAM, CRAM, VCF, MAF, BED detail, Personal Genome SNP, broadPeak, narrowPeak, and microarray (BED15). GFF and GTF files *must* be tab-delimited rather than space-delimited to display correctly. Chromosome references must be of the form *chrN* (the parsing of chromosome names ***is*** case-sensitive). You may include more than one data set in your annotation file; these need not be in the same format.

Step 2. Define the Genome Browser display characteristics
Add one or more optional browser lines to the beginning of your formatted data file to configure the overall display of the Genome Browser when it initially shows your annotation data. Browser lines allow you to configure such things as the genome position that the Genome Browser will initially open to, the width of the display, and the configuration of the other annotation tracks that are shown (or hidden) in the initial display. NOTE: If the browser position is not explicitly set in the annotation file, the initial display will default to the position setting most recently used by the user, which may not be an appropriate position for viewing the annotation track.

Step 3. Define the annotation track display characteristics
Following the browser lines--and immediately preceding the formatted data--add a track line to define the display attributes for your annotation data set. Track lines enable you to define annotation track characteristics such as the name, description, colors, initial display mode, use score, etc. The track type=<*track_type*> attribute is required for some tracks. If you have included more than one data set in your annotation file, insert a track line at the beginning of each new set of data.

Example 1:
Here is an example of a simple annotation file that contains a list of chromosome coordinates.

```
browser position chr22:20100000-20100900
track name=coords description="Chromosome coordinates list" visibility=2
chr22   20100000 20100100
chr22   20100011 20100200
chr22   20100215 20100400
chr22   20100350 20100500
chr22   20100700 20100800
chr22   20100700 20100900
```

Click here to view this track in the Genome Browser.

Table Browser

Use this program to retrieve the data associated with a track in text format, to calculate intersections between tracks, and to retrieve DNA sequence covered by a track. For help in using this application see Using the Table Browser for a description of the controls in this form, the User's Guide for general information and sample queries, and the OpenHelix Table Browser tutorial for a narrated presentation of the software features and usage. For more complex queries, you may want to use Galaxy or our public MySQL server. To examine the biological function of your set through annotation enrichments, send the data to GREAT. Send data to GenomeSpace for use with diverse computational tools. Refer to the Credits page for the list of contributors and usage restrictions associated with these data. All tables can be downloaded in their entirety from the Sequence and Annotation Downloads page.

clade: Mammal **genome:** Human **assembly:** Feb. 2009 (GRCh37/hg19)

group: Genes and Gene Predictions **track:** RefSeq Genes manage custom tracks track hubs

table: refGene describe table schema

region: genome ENCODE Pilot regions position chr17:29421745-29685765 lookup define regions

identifiers (names/accessions): paste list upload list

filter: edit clear

intersection: create

correlation: create

output format: BED - browser extensible data Send output to Galaxy GREAT GenomeSpace

output file: RefSeq.10kup.bed (leave blank to keep output in browser)

file type returned: plain text gzip compressed

get output summary/statistics

To reset **all** user cart settings (including custom tracks), click here.

Summary and Resources

- Formats
  - FASTQ
  - SAM/BAM (CRAM)
  - VCF
- Alignment and tools
  - BWA
  - IGV
  - Samtools
- Post-Alignment Processing and QC
  - GATK IndelRealigner
  - GATK BaseRecalibrator
  - Picard MarkDuplicate
  - Picard HS-Metrics
  - Samtools
- Variant Calling
  - GATK UnifiedGenotyper
  - GATK HaplotypeCaller
  - FreeBayes
  - VarScan (somatic)
  - Mutect (somatic)
- Variant Annotations
  - SNPEff
  - ANNOVAR
  - Vtools
  - VEP
- Intervals and Tracks
  - UCSC Genome Browser
  - BEDtools
  - UNIX