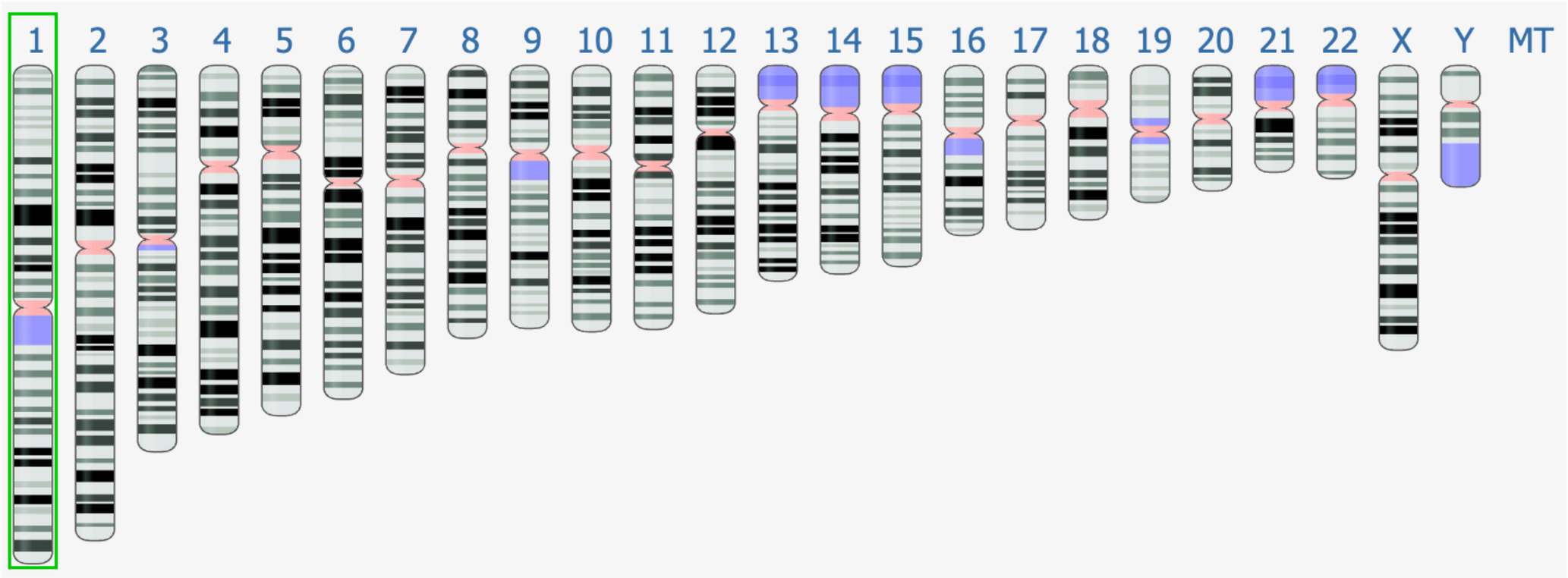


Genomics105

BIOL647
Digital Biology

Rodolfo Aramayo

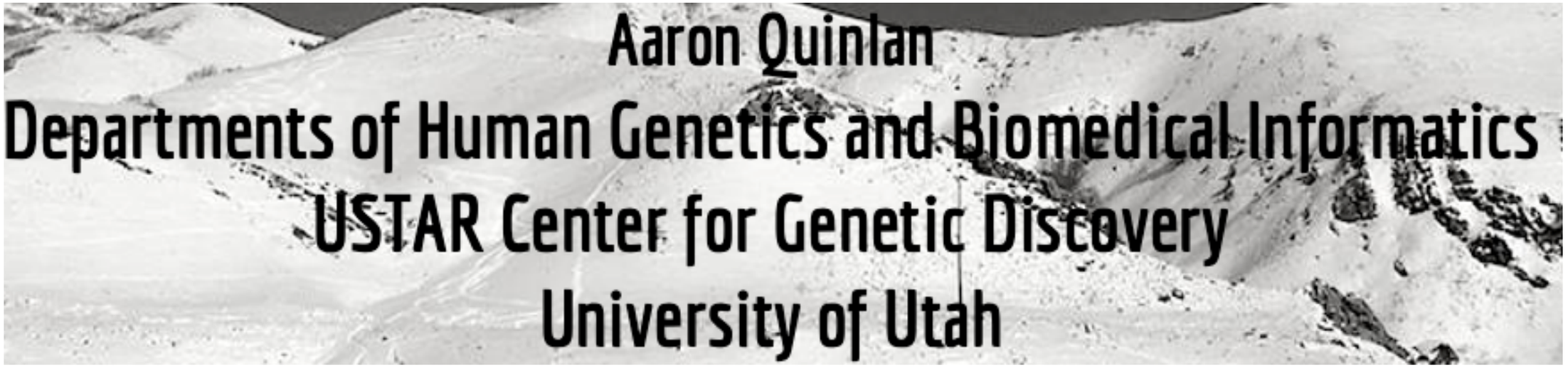
Genomics105

Mapping and Alignment 101

Ben Langmead

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Department of Computer Science

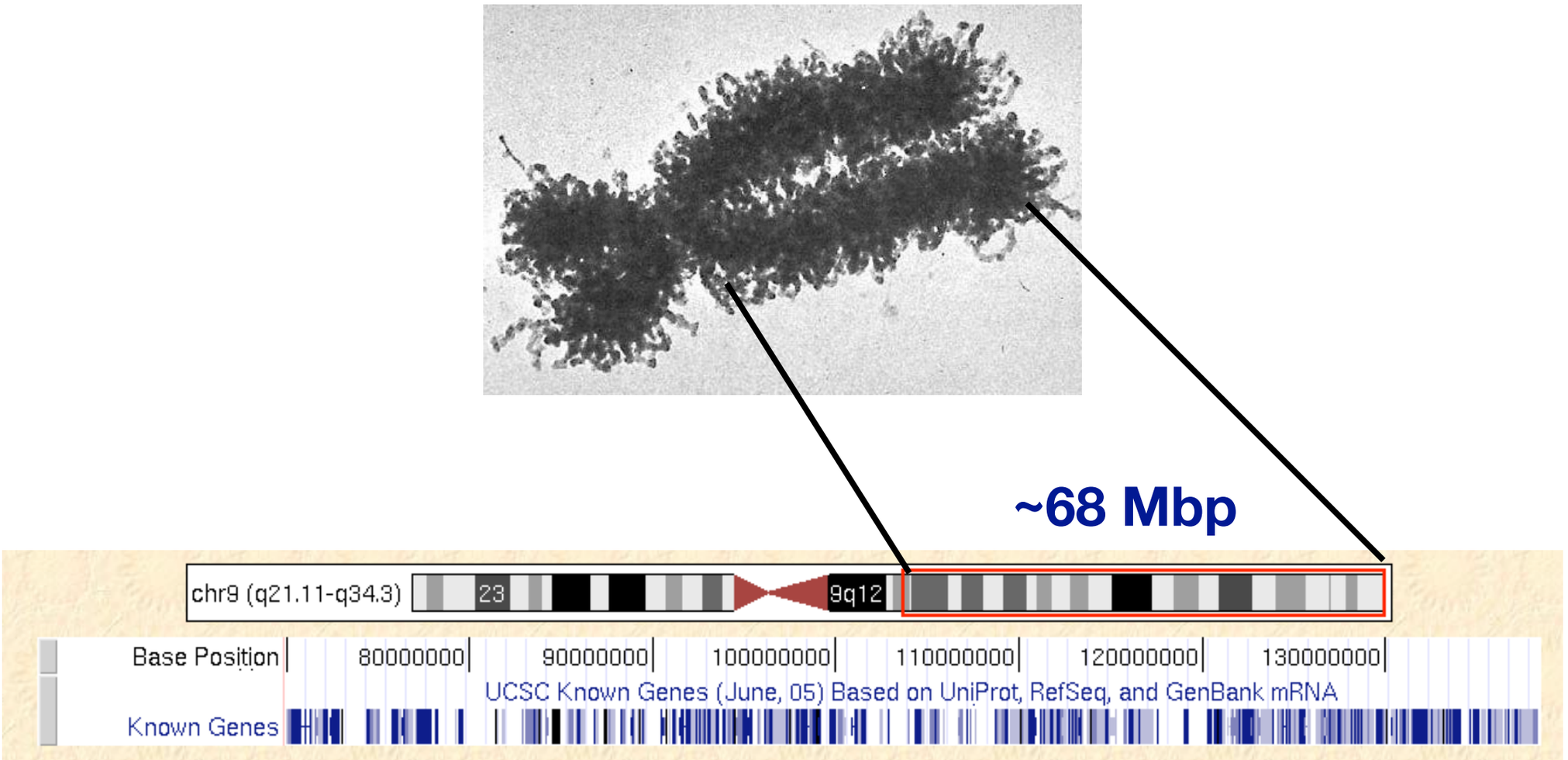


Genomics105

Mapping and Alignment 101

The Human Genome

What is the gene distribution?



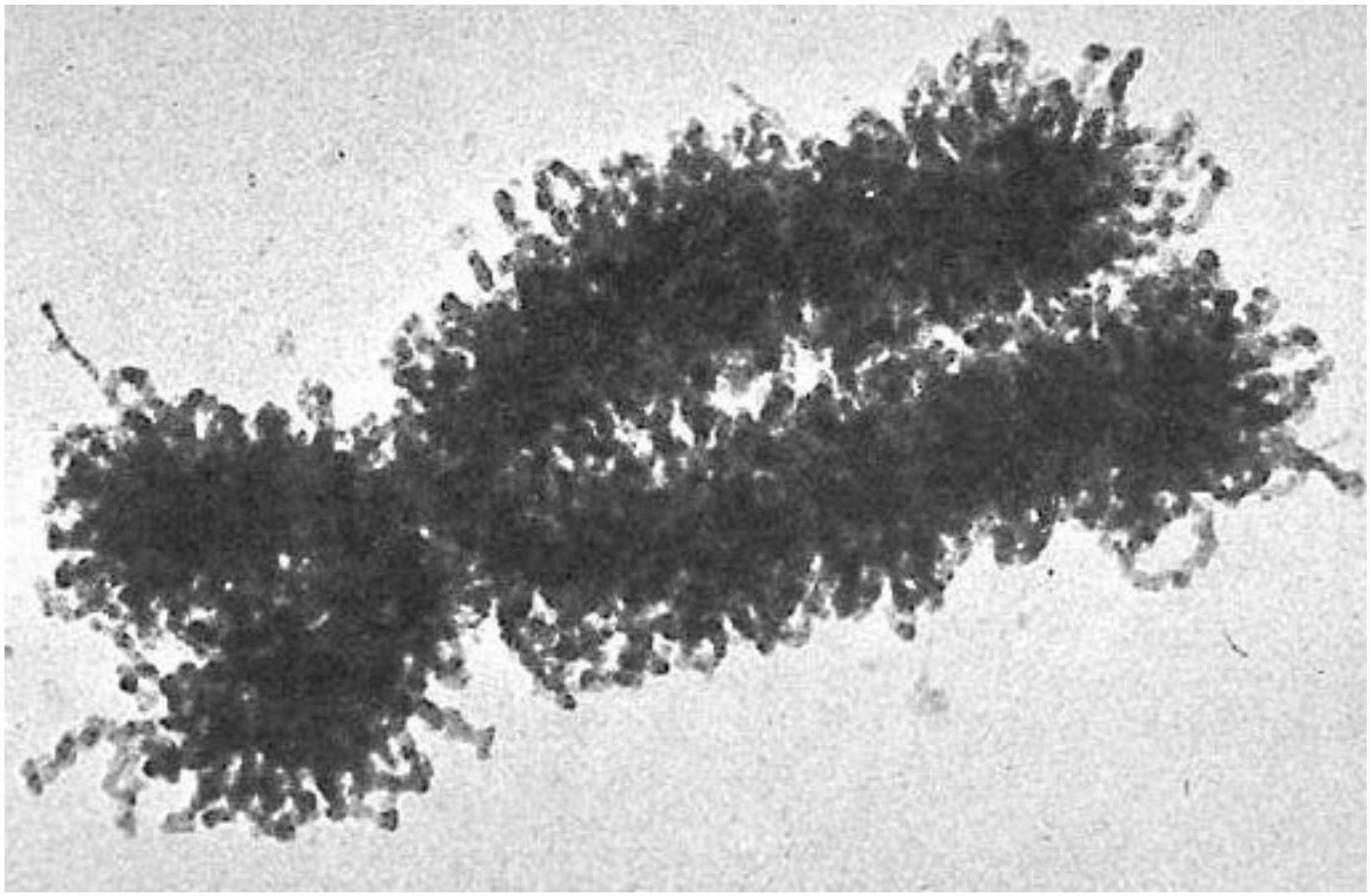
UCSC Genome Browser with VISTA tracks on Human May 2004 Assembly

Genomics105

Mapping and Alignment 101

The Human Genome

The Human Genome is Full of Repetitive Elements!

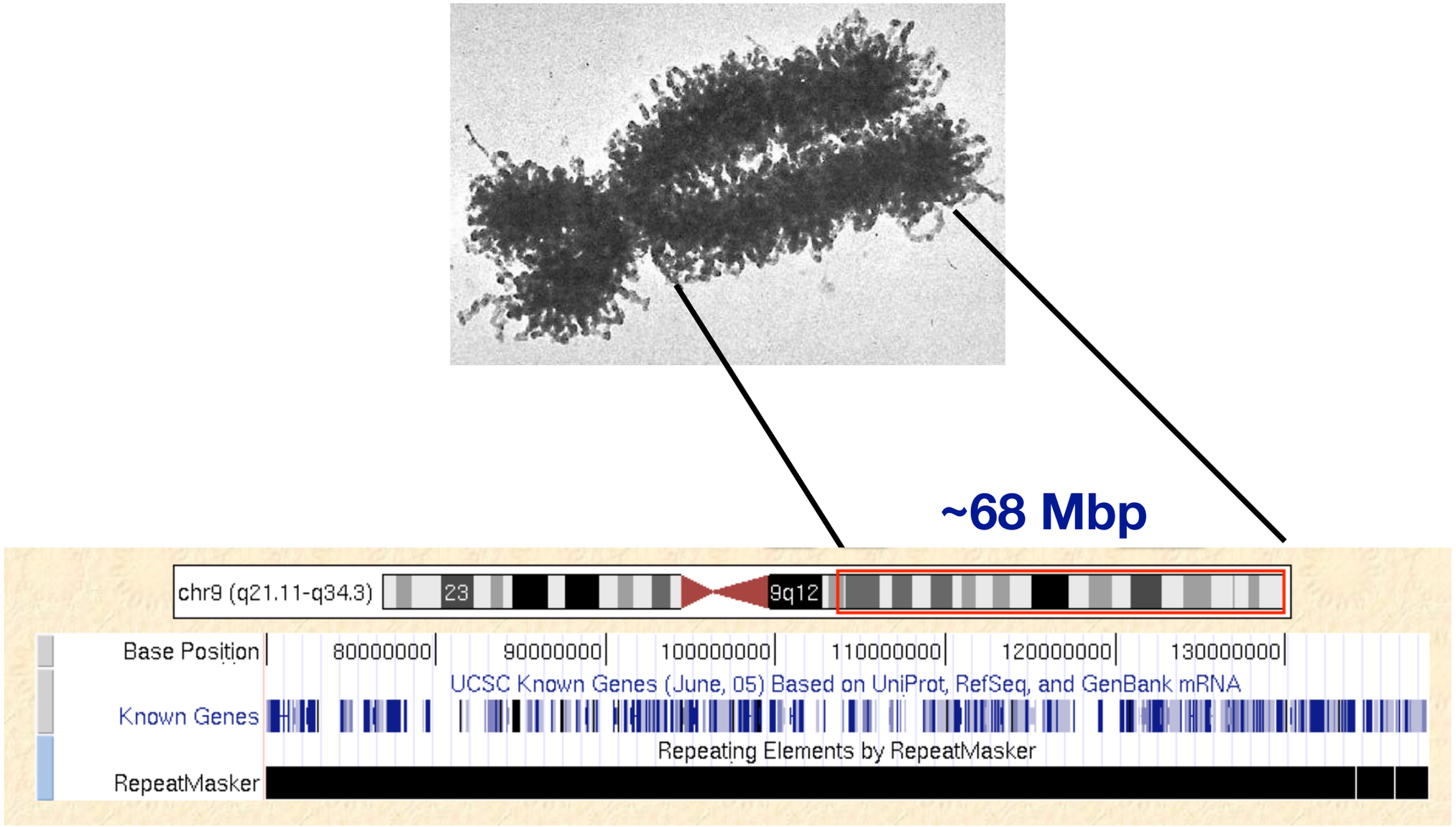


Genomics105

Mapping and Alignment 101

The Human Genome

The Human Genome is Full of Repetitive Elements!



UCSC Genome Browser with VISTA tracks on Human May 2004 Assembly

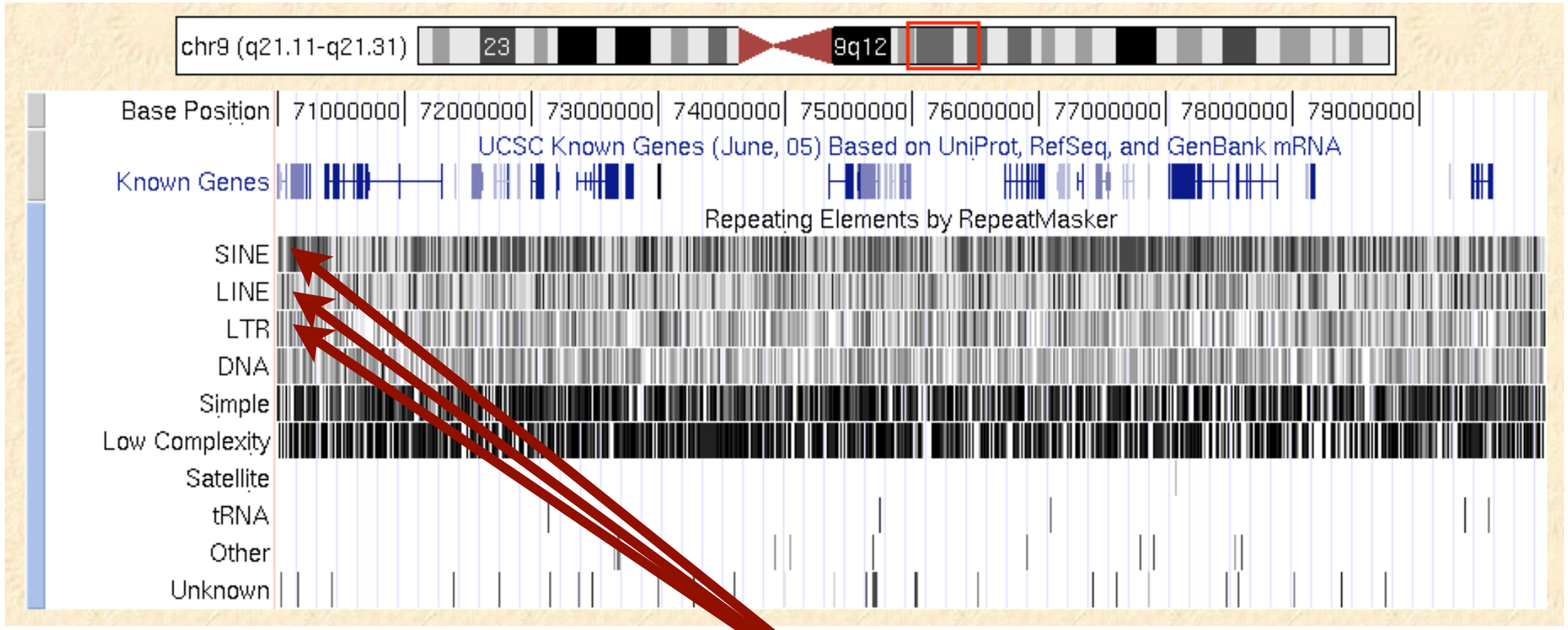
Genomics105

Mapping and Alignment 101

The Human Genome

The Human Genome is Full of Repetitive Elements!

~10 Mbp



Transposable Elements!

UCSC Genome Browser with VISTA tracks on Human May 2004 Assembly

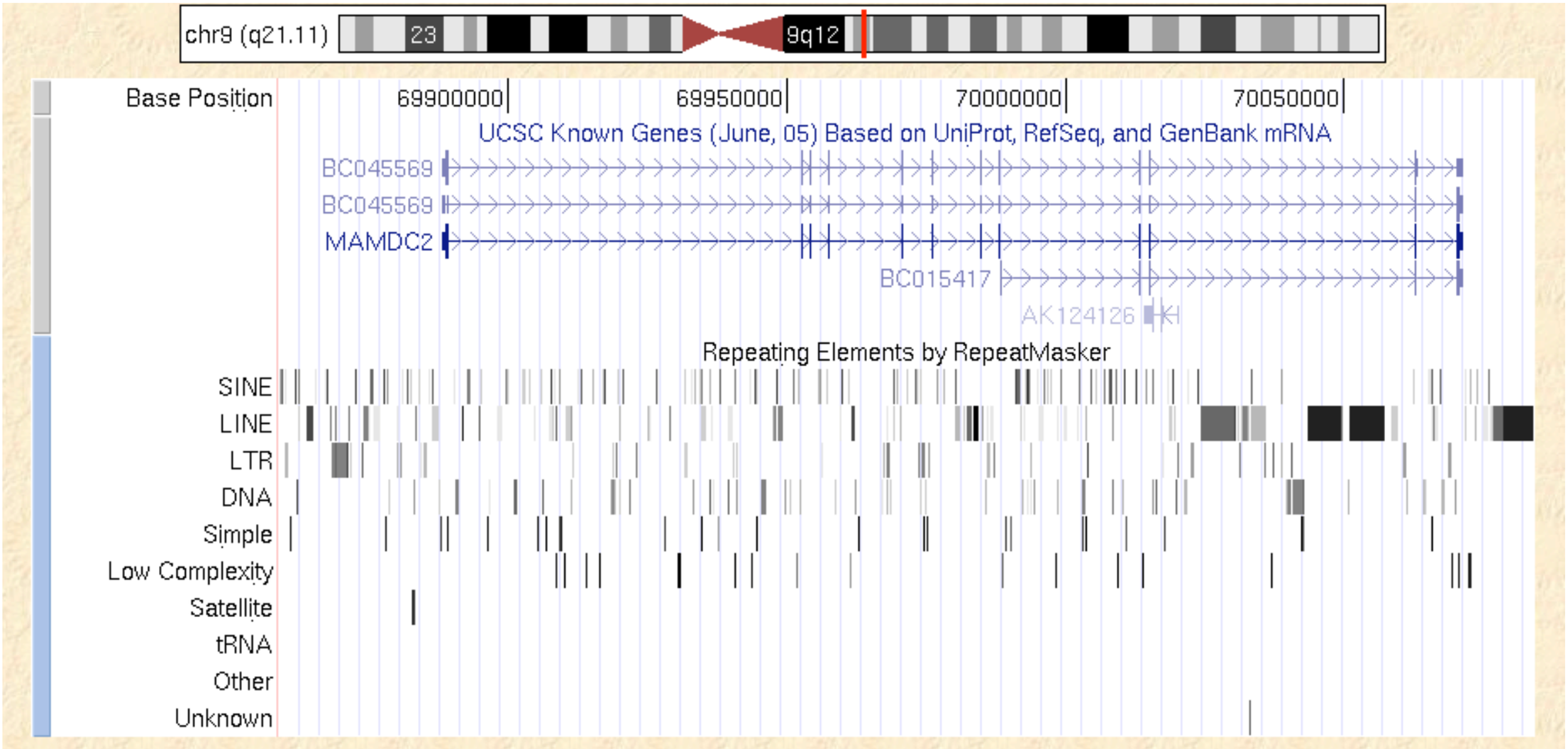
Genomics105

Mapping and Alignment 101

The Human Genome

Selfish Genetic Elements Live within our own Genes!

Zooming in ~45,000X to ~225 kbp



UCSC Genome Browser with VISTA tracks on Human May 2004 Assembly

Genomics105

Mapping and Alignment 101

The problem

- ***The human genome is big and complex***
- ***Sequencers can produce 1 billion reads / run***
- ***Sequencers make mistakes - Frequently***
- ***Aligning Reads to Genomes takes time***
- ***Alignment Shortcuts lead to artifacts***
- ***Alignment is more an Art than a Science***

Genomics105

Mapping and Alignment 101

Alignment is central to most genomic research

FASTQ format

Align to genome

Quality control

Find peaks

Quantify transcripts

Find SVs

.

.

.

etc.

Analysis, Interpretation, Discovery

Phase 0

Phase1

Phase 2-100

https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

Once we obtained FASTQ Files

- ***Need to find a home for every read in the file***

- ***Must get the alignment just right. Else problems***

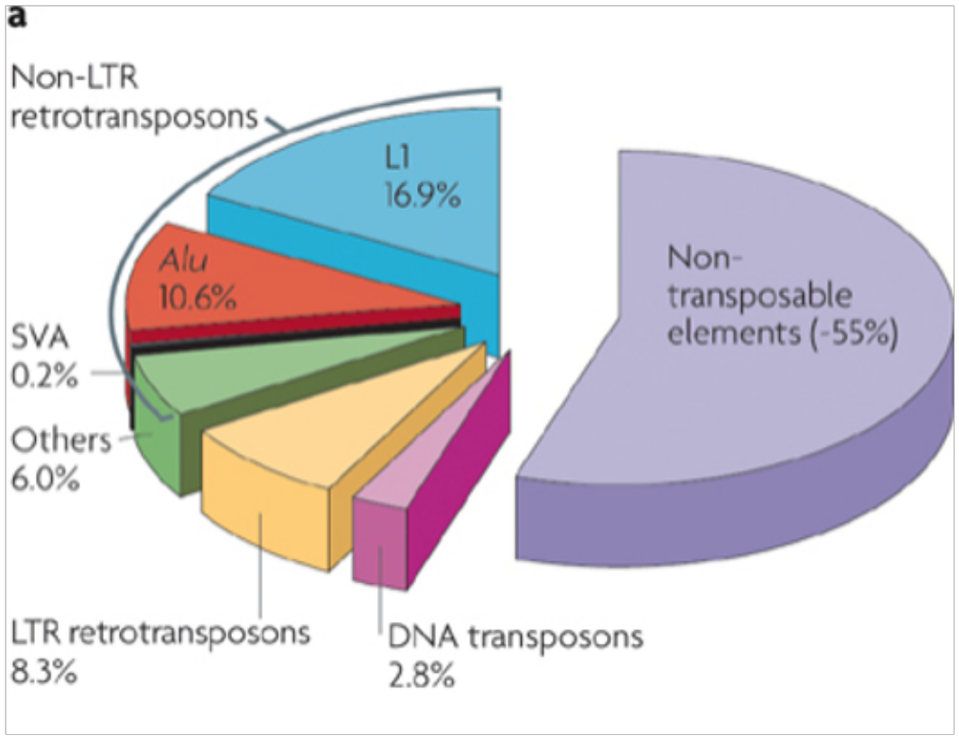
- ***Must choose the right tool for the experiment***

https://github.com/quinlan-lab/applied-computational-genomics

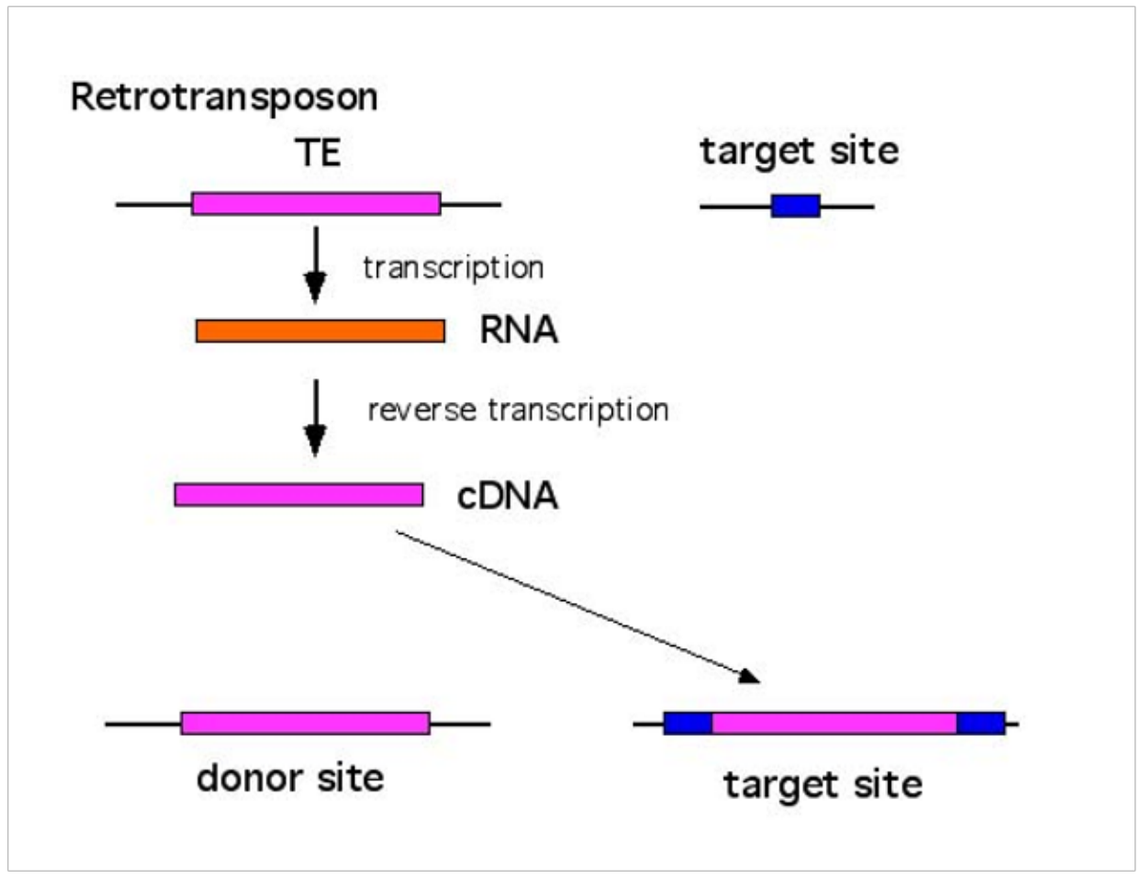
Genomics105

Mapping and Alignment 101

Problem: Half of the human genome is comprised of repeats



McClintock's "jumping genes" in maize



Retrotransposons use a "copy/paste" mechanism
DNA transposons use a "cut/paste" mechanism

http://www.nature.com/nrg/journal/v10/n10/pdf/nrg2640.pdf

Genomics105

Mapping and Alignment 101

Problem: Half of the human genome is comprised of repeats

(first bit of human chromosome 1)

```
taaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaa
accctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaac
cctaacccaaccctaaccctaaccctaaccctaaccctaaccctaaccctaacccc
taaccctaaccctaaccctaaccctaacctaaccctaaccctaaccctaaccctaa
ccccccctaaccctaaccctaaccctaaccccctaaccctaaccctaaaccc
ccctaaaccctaaccctaaccctaaccctaaccctaaccccaaccccaac
cccaaccccaaccccaaccccaaccctaaccccctaaccctaaccctaacc
ctaccctaaccctaaccctaaccctaaccctaaccctaaccccctaacccc
taaccctaaccctaaccctaaccctaaccctaaccctaaccccctaaccct
aaccctaaccctaaccctcgcggtaccctcagccggcccgcccgcccggg
tctgacctgaggagaactgtgctccgccttcagagtaccaccgaaatctg
tgcagaggacaacgcagctccgccctcgcggtgctctccgggtctgtgct
gaggagaacgcaactccgccggcgcaggcgcagagaggcgcgccgcgccg
gcgcaggcgcagacacatgctagcgcgtcggggtggaggcgtggcgcagg
cgcagagaggcgcgccgcgccggcgcaggcgcagagacacatgctaccgc
gtccaggggtggaggcgtggcgcaggcgcagagaggcgcaccgcgccggc
gcaggcgcagagacacatgctagcgcgtccaggggtggaggcgtggcgca
ggcgcagagacgcaagcctacgggcgggggttgggggggcgtgtgttgca
ggagcaaagtcgcacggcgccgggctggggcgggggggagggtggcgccgt
gcacgcgcagaaactcacgtcacggtggcgcggcgcagagacgggtagaa
```

https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

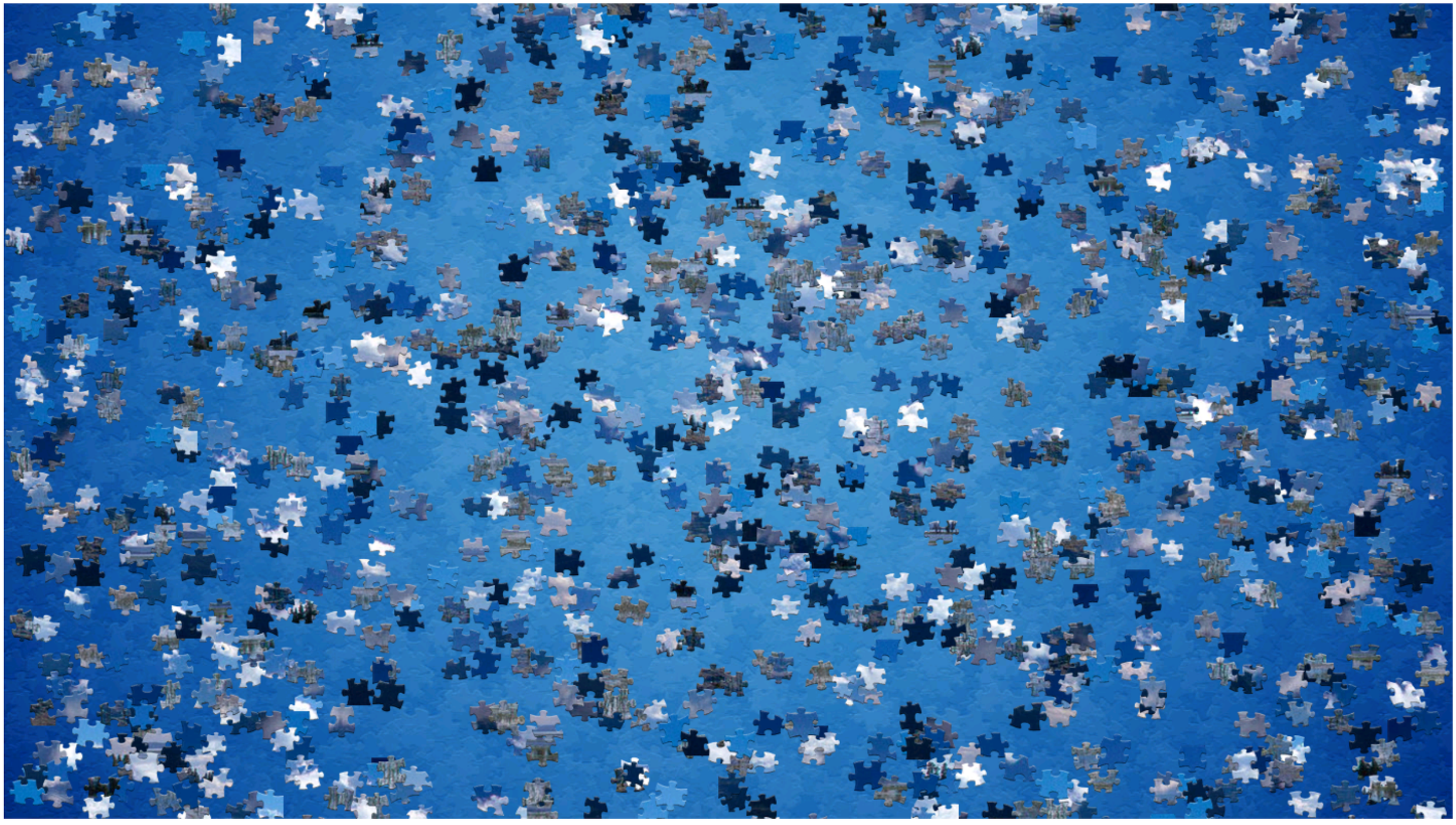
Problem: Half of the human genome is comprised of repeats

https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

Problem: Half of the human genome is comprised of repeats

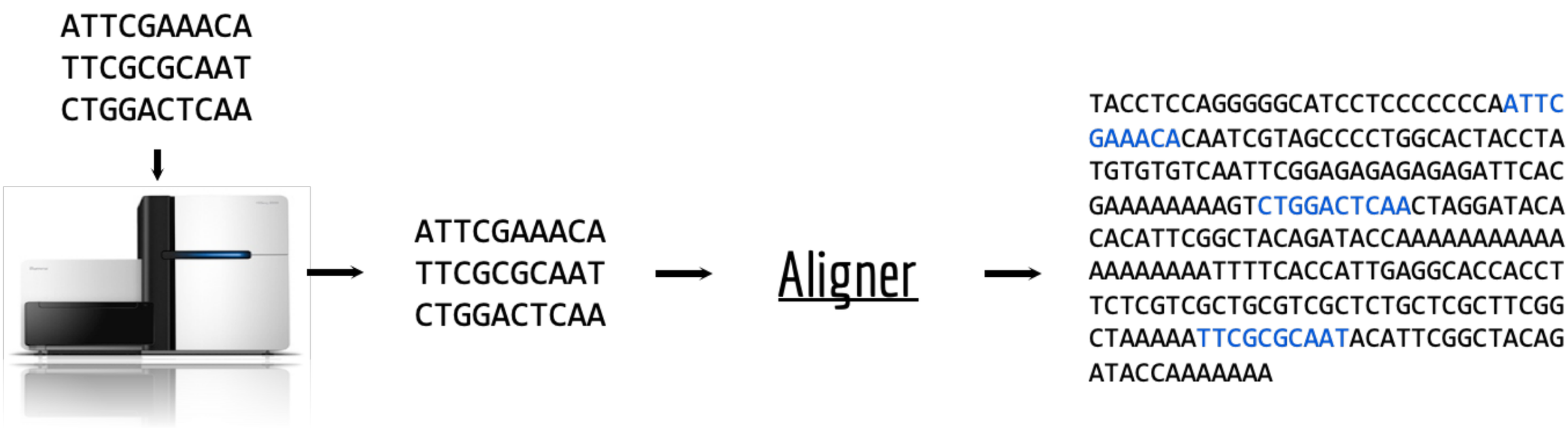


https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

Best case scenario: an error-free sequencing technology



Computers are rather good at finding ***exact*** matches.
Think Google.

https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

Reality check: Errors happen...Frequently

ATTCGAAACA

ATTTGAAACA → Aligner →

```
TACCTCCAGGGGGCATCCTCCCCCCCAATTC
GAAACACAATCGTAGCCCCTGGCACTACCTA
TGTGTGTCAATTCGGAGAGAGAGAGATTGGA
AACAAAAAAGTGCTACAGATACCACTAGGAT
ACACACATTCGGCTACAGATACCAAAAAAAA
AAAAAAAAAAATTTTCACCATTGAGGCACCA
CCTTCTCGTCGCTGCGTCGCTCTGCTCGCGG
CTAAAAAATTAGAAACAACATTCGGCTACAG
ATACCAAAATTT
```

"Fuzzy" matching is much more computationally expensive.
Think Google's "Did you mean..."

https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

Sequence mapping versus alignment

- ***Mapping**: (quickly) find the best possible loci to which a sequence could be aligned*

- ***Alignment**: for each locus to which a sequence can be mapped, determine the optimal base by base alignment of the query sequence to the reference sequence*

https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

Hash-based mapping:

Step1: hash/index the genome

Toy genome
(16 bp)

CATGGTCATTGGTTCC

https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

Hash-based mapping:

Step1: hash/index the genome

CATGGTCATTGGTTCC

k = 3

Kmer/Hash	Genome Positions
CAT	1

Genomics105

Mapping and Alignment 101

Hash-based mapping:

Step1: hash/index the genome

CATGGTCATTGGTTCC

k = 3

Kmer/Hash	Genome Positions
CAT	1
ATG	2

https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

Hash-based mapping:

Step1: hash/index the genome

CATGGTCATTGGTTCC

k = 3

Kmer/Hash	Genome Positions
CAT	1
ATG	2
TGG	3

https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

Hash-based mapping:

Step1: hash/index the genome

CATGGTCATTGGTTCC

k = 3

Kmer/Hash	Genome Positions
CAT	1
ATG	2
TGG	3
GGT	4

https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

Hash-based mapping:

Step1: hash/index the genome

CATGGTCATTGGTTCC

k = 3

Kmer/Hash	Genome Positions
CAT	1
ATG	2
TGG	3
GGT	4
GTC	5

https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

Hash-based mapping:

Step1: hash/index the genome

CATGGTCATTGGTTCC

k = 3

Kmer/Hash	Genome Positions
CAT	1
ATG	2
TGG	3
GGT	4
GTC	5
TCA	6

https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

Hash-based mapping:

Step1: hash/index the genome

CATGGTCATTGGTTCC

k = 3

Kmer/Hash	Genome Positions
CAT	1,7
ATG	2
TGG	3
GGT	4
GTC	5
TCA	6

https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

Hash-based mapping:

Step1: hash/index the genome

CATGGTCATTGGTTCC

k = 3	Kmer/Hash	Genome Positions
	CAT	1,7
	ATG	2
	TGG	3,10
	GGT	4,11
	GTC	5
	TCA	6
	ATT	8
	TTG	9
	GTT	12
	TTC	13
	TCC	14

Complete hash/kmer index of our toy genome (forward strand only)

https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

Hash-based mapping:

Step2: use the index to map (i.e., find alignment locations) reads

Toy genome CATGGTCATTGGTTCC

→ Read TGGTCA

Kmer/Hash	Genome Positions
CAT	1,7
ATG	2
TGG	3,10
GGT	4,11
GTC	5
TCA	6
ATT	8
TTG	9
GTT	12
TTC	13
TCC	14

kmer index is used to quickly find candidate alignment locations in genome.

https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

Hash-based mapping:

Step2: use the index to map (i.e., find alignment locations) reads

Toy genome CATGGTCATTGGTTCC

→ Read TGGTCA

Kmer/Hash	Genome Positions
CAT	1,7
ATG	2
TGG	3,10
GGT	4,11
GTC	5
TCA	6
ATT	8
TTG	9
GTT	12
TTC	13
TCC	14

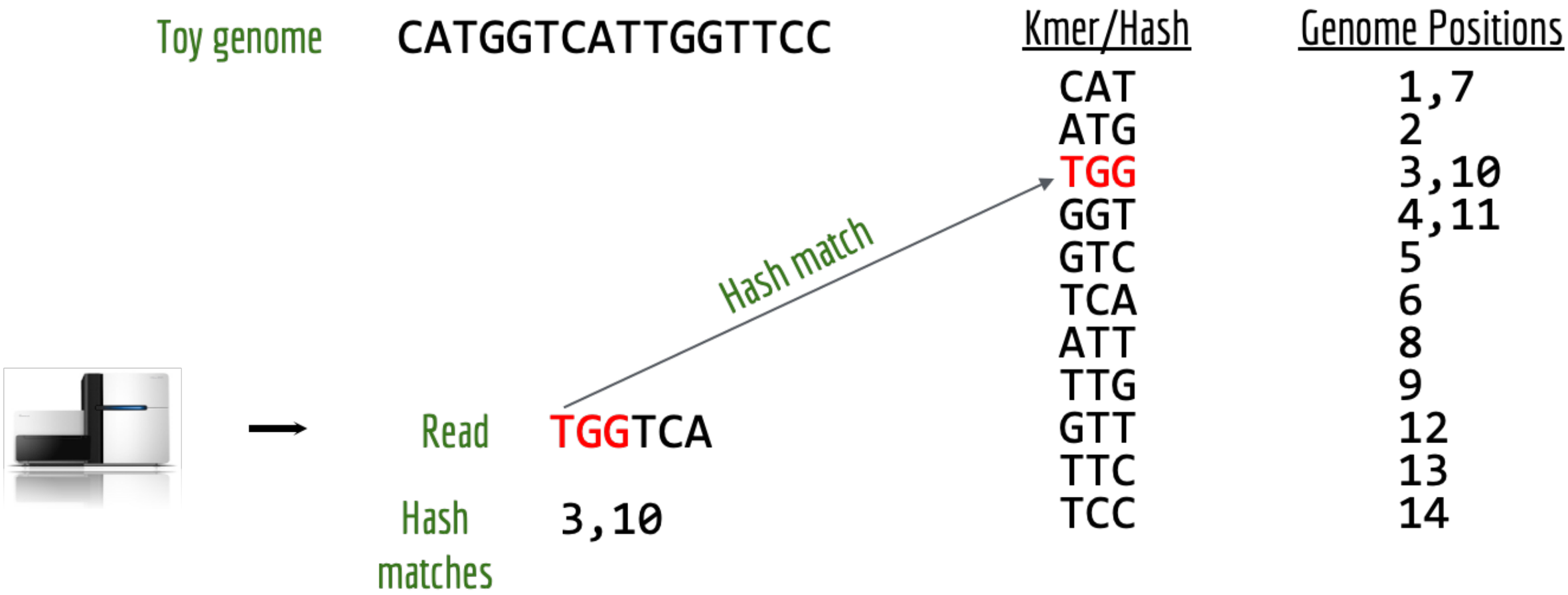
https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

Hash-based mapping:

Step2: use the index to map (i.e., find alignment locations) reads



Genomics105

Mapping and Alignment 101

Hash-based mapping:

Step2: use the index to map (i.e., find alignment locations) reads

Toy genome CATGGTCATTGGTTCC

Read TGGTCA

Hash matches 3,10,6

Kmer/Hash	Genome Positions
CAT	1,7
ATG	2
TGG	3,10
GGT	4,11
GTC	5
TCA	6
ATT	8
TTG	9
GTT	12
TTC	13
TCC	14

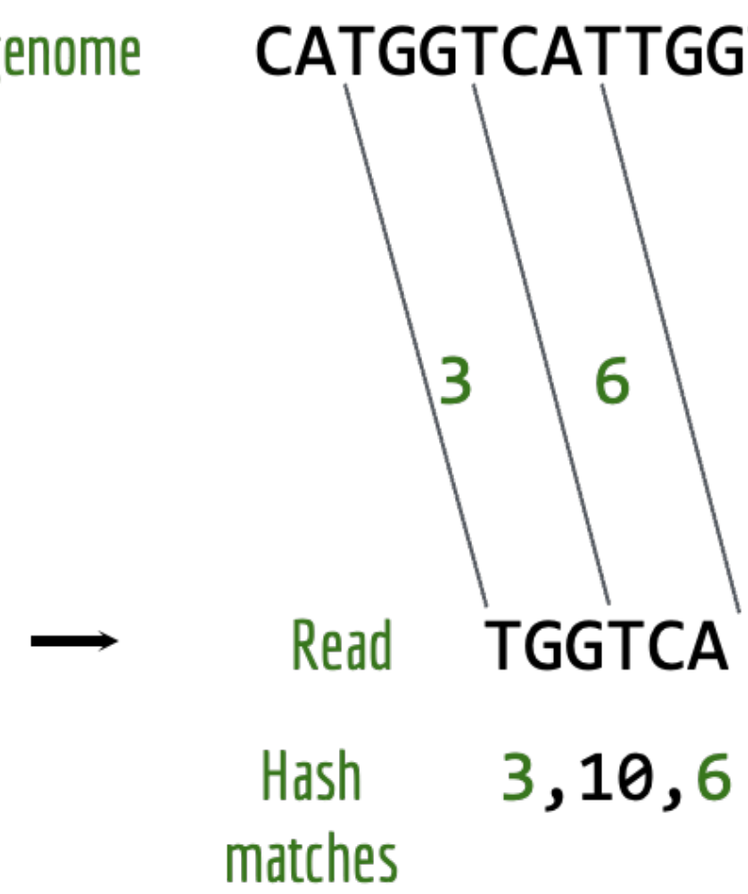
https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

Hash-based mapping:

Step2: use the index to map (i.e., find alignment locations) reads



Kmer/Hash	Genome Positions
CAT	1,7
ATG	2
TGG	3,10
GGT	4,11
GTC	5
TCA	6
ATT	8
TTG	9
GTT	12
TTC	13
TCC	14

https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

Okay, that was a bit easy because the read and the reference exactly matched

What about if there is a sequencing error or a genetic variant in the read?

Genomics105

Mapping and Alignment 101

Hash-based mapping:

Step2: use the index to map (i.e., find alignment locations) reads

Toy genome CATGGTCATTGGTTCC

→ Read TGGTCT

Kmer/Hash	Genome Positions
CAT	1,7
ATG	2
TGG	3,10
GGT	4,11
GTC	5
TCA	6
ATT	8
TTG	9
GTT	12
TTC	13
TCC	14

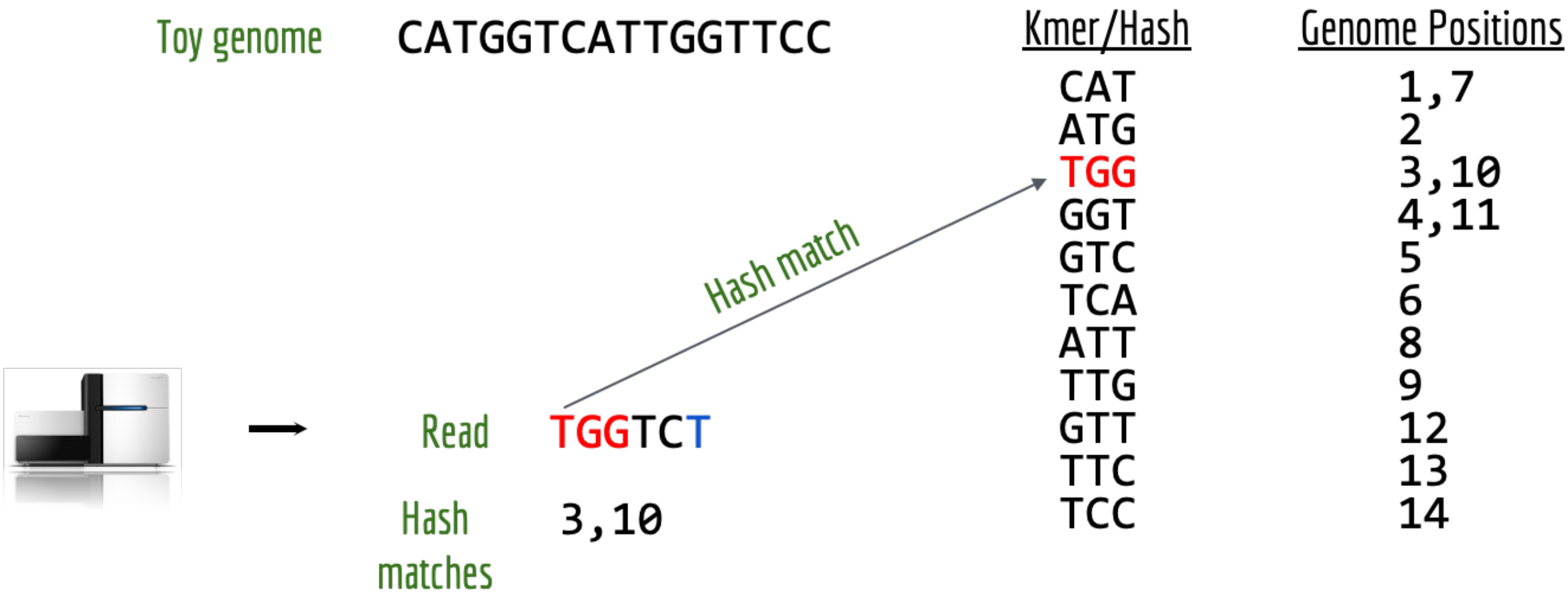
kmer index is used to quickly find candidate alignment locations in genome.

Genomics105

Mapping and Alignment 101

Hash-based mapping:

Step2: use the index to map (i.e., find alignment locations) reads



Genomics105

Mapping and Alignment 101

Hash-based mapping:

Step2: use the index to map (i.e., find alignment locations) reads

Toy genome CATGGTCATTGGTTCC

Read TGGTCT ?

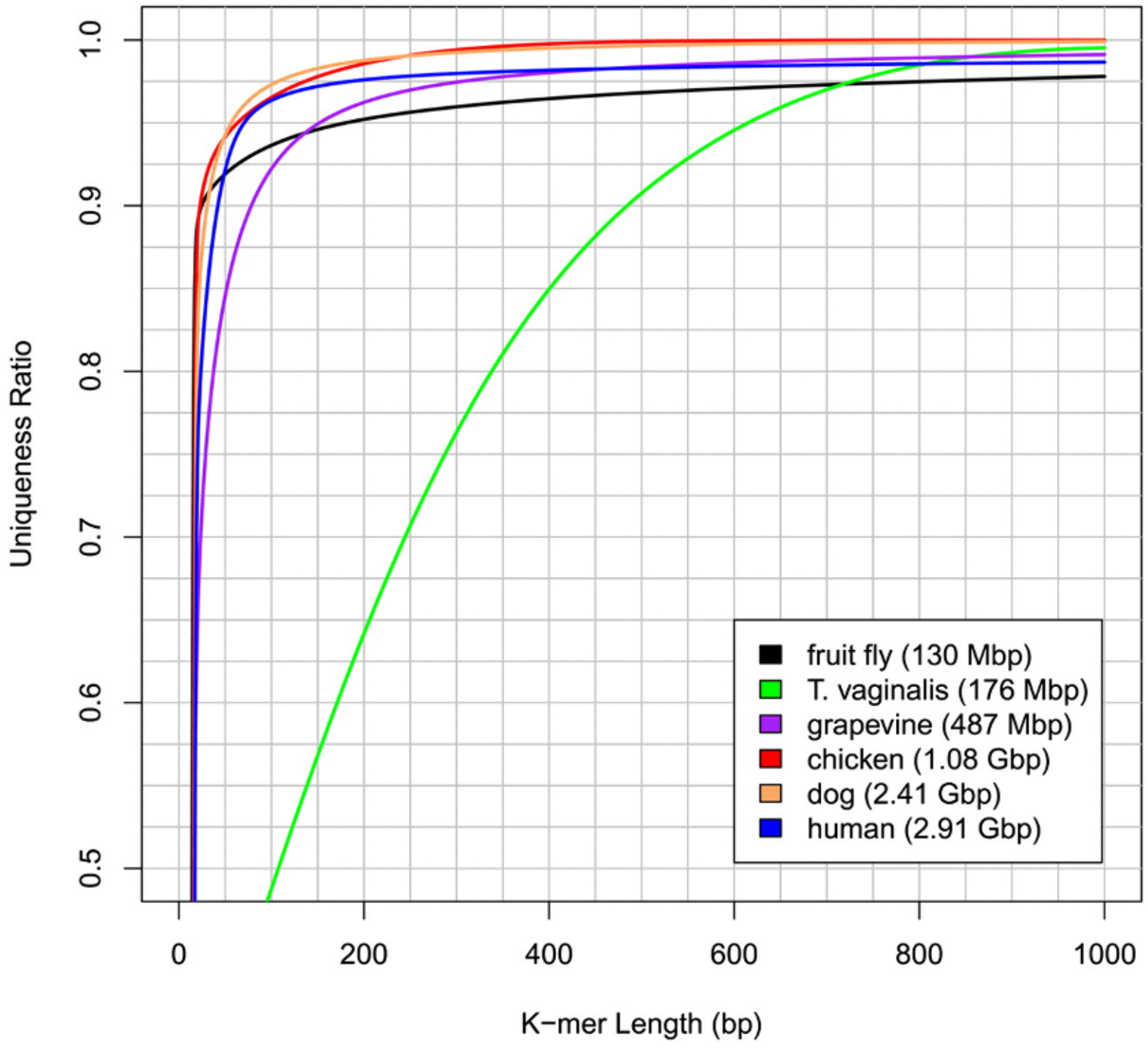
Hash matches 3,10

Kmer/Hash	Genome Positions
CAT	1,7
ATG	2
TGG	3,10
GGT	4,11
GTC	5
TCA	6
ATT	8
TTG	9
GTT	12
TTC	13
TCC	14

Genomics105

Mapping and Alignment 101

Mapping quality (MAPQ)



http://genome.cshlp.org/content/20/9/1165

Genomics105

FASTQ Format

Remember? - Phred quality score calculation

$$Q = -10*log10(Perr)$$

Error probability	log10(Per)	Phred quality score
1	0	0
0.1	-1	10
0.01	-2	20
0.001	-3	30
0.0001	-4	40

Genomics105

Mapping and Alignment 101

Mapping quality (MAPQ)

MAPQ = -10*log10(Pmap_loc_wrong)

Error probability	log10(Per)	Phred quality score
1	0	0
0.1	-1	10
0.01	-2	20
0.001	-3	30
0.0001	-4	40

Genomics105

Mapping and Alignment 101

Edit distance

How many edits (changes) must be made to a word or kmer to make it match (align) to another word or kmer?

CURLED
HURLED → Edit distance = 1. Substitute C for H

SHORT
SHO-T → Edit distance = 1. Delete R

TGTTACGG
GGTTGACTA ?

TG-TT-ACGG
-GGTTGACTA

Edit distance = 5

Genomics105

Mapping and Alignment 101

SAM format

Overview

- In the dark ages, sequence aligners used disparate output formats - Painful
- 1000 Genomes Project sought to standardize - Standards are good
- The result is imperfect, but it's a huge improvement
- Strengths of the SAM and BAM formats
    - Compressed: less disk hungry
    - Indexed: fast viewing, slicing, etc.
    - Single-end and paired-end
    - Relatively simple to produce
    - Good toolkits available

Genomics105

Mapping and Alignment 101

SAM format

a text-based standard for representing sequence alignments

SAM stands for Sequence Alignment/Map format

It is a TAB-delimited text format consisting of a header section, which is optional, and an alignment section. If present, the header must be prior to the alignments. Header lines start with '@', while alignment lines do not. Each alignment line has 11 mandatory fields for essential alignment information such as mapping position, and variable number of optional fields for flexible or aligner specific information

Genomics105

Mapping and Alignment 101

SAM format

Example

Suppose we have the following alignment with bases in lowercase clipped from the alignment. Read r001/1 and r001/2 constitute a read pair; r003 is a chimeric read; r004 represents a split alignment.

```
Coor     12345678901234  5678901234567890123456789012345
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

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

Genomics105

Mapping and Alignment 101

SAM format

Example

The corresponding SAM format is:

```
@HD VN:1.6 SO:coordinate
@SQ SN:ref LN:45
r001    99 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002     0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003     0 ref  9 30 5S6M       *  0   0 GCCTAAGCTAA       * SA:Z:ref,29,-,6H5M,17,0;
r004     0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003  2064 ref 29 17 6H5M       *  0   0 TAGGC             * SA:Z:ref,9,+,5S6M,30,1;
r001   147 ref 37 30 9M         =  7 -39 CAGCGGCAT         * NM:i:1
```

The values in the FLAG column correspond to bitwise flags as follows: 99 = 0x63: first/next is reverse-complemented/ properly aligned/multiple segments; 0: no flags set, thus a mapped single segment; 2064 = 0x810: supplementary/reversecomplemented; 147 = 0x93: last (second of a pair)/reverse-complemented/properly aligned/ multiple segments.

Genomics105

Mapping and Alignment 101

SAM format

Example

```
Coor    12345678901234  5678901234567890123456789012345
ref     AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT
+r001/1       TTAGATAAAGGATA*CTG
-r001/2                                     CAGCGGCAT

@HD VN:1.6 SO:coordinate
@SQ SN:ref LN:45
r001 99  ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r001 147 ref 37 30 9M        =  7 -39 CAGCGGCAT         * NM:i:1
```

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	`[!-?A-~]{1,254}`	Query template NAME
2	FLAG	Int	$[0, 2^{16} - 1]$	bitwise FLAG
3	RNAME	String	`\*|[:rname:^*=][:rname:]*`	Reference sequence NAME[11]
4	POS	Int	$[0, 2^{31} - 1]$	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	`\*|=|[:rname:^*=][:rname:]*`	Reference name of the mate/next read
8	PNEXT	Int	$[0, 2^{31} - 1]$	Position of the mate/next read
9	TLEN	Int	$[-2^{31} + 1, 2^{31} - 1]$	observed Template LENgth
10	SEQ	String	`\*|[A-Za-z=.]+`	segment SEQuence
11	QUAL	String	`[!-~]+`	ASCII of Phred-scaled base QUALity+33

Mandatory Fields in the SAM Format

Genomics105

Mapping and Alignment 101

SAM format

Example

```
Coor     12345678901234  5678901234567890123456789012345
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT
+r001/1        TTAGATAAAGGATA*CTG
-r001/2                                       CAGCGGCAT

@HD VN:1.6 SO:coordinate
@SQ SN:ref LN:45
r001 99  ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r001 147 ref 37 30 9M         =  7 -39 CAGCGGCAT         * NM:i:1
```

Bit		Description
1	0x1	template having multiple segments in sequencing
2	0x2	each segment properly aligned according to the aligner
4	0x4	segment unmapped
8	0x8	next segment in the template unmapped
16	0x10	SEQ being reverse complemented
32	0x20	SEQ of the next segment in the template being reverse complemented
64	0x40	the first segment in the template
128	0x80	the last segment in the template
256	0x100	secondary alignment
512	0x200	not passing filters, such as platform/vendor quality controls
1024	0x400	PCR or optical duplicate
2048	0x800	supplementary alignment

Genomics105

Mapping and Alignment 101

Recall: Edit distance

How many edits (changes) must be made to a word or kmer to make it match (align) to another word or kmer?

CURLED
HURLED → Edit distance = 1. Substitute C for H

SHORT
SHO-T → Edit distance = 1. Delete R

TGTTACGG
GGTTGACTA ?

TG-TT-ACGG
-GGTTGACTA

Edit distance = 5

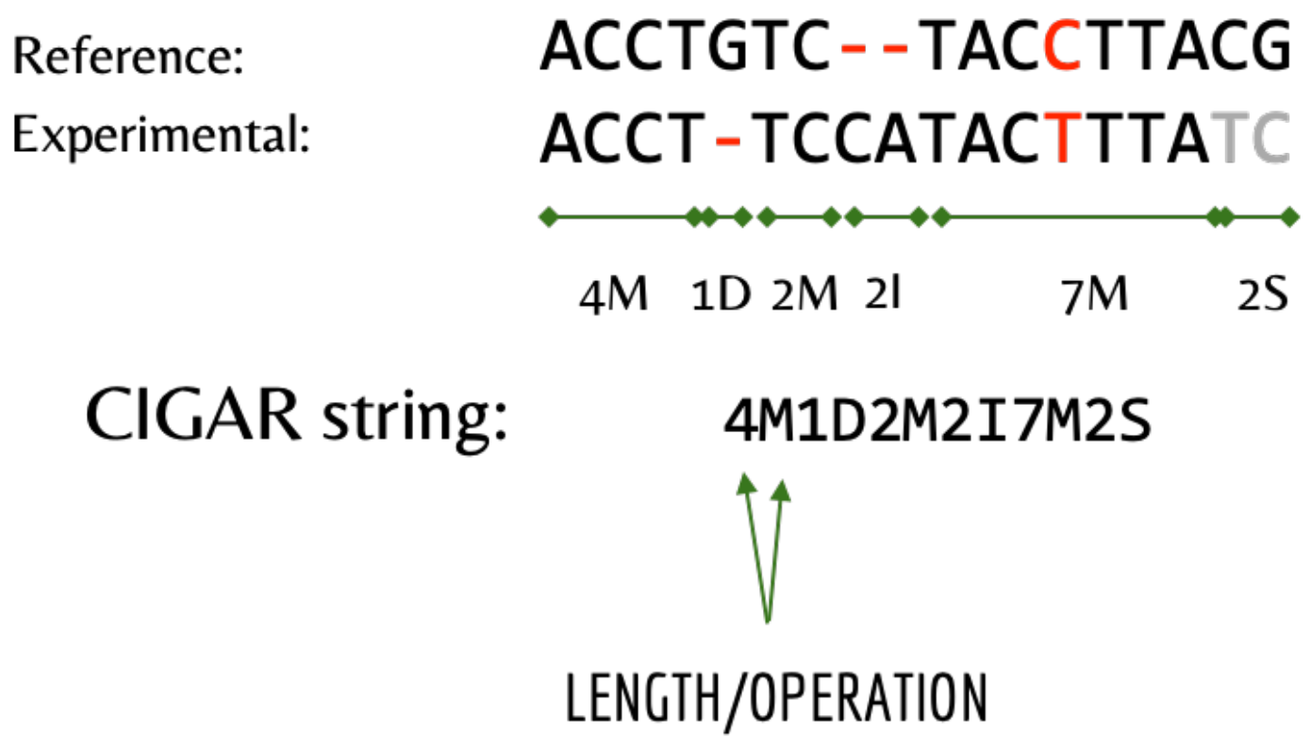
https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

The CIGAR string: encode the details of the alignment

Operation	Meaning
M	Match*
D	Deletion w.r.t. reference
I	Insertion w.r.t. reference
N	Split or spliced alignment
S	Soft-clipping
H	Hard-clipping
P	Padding



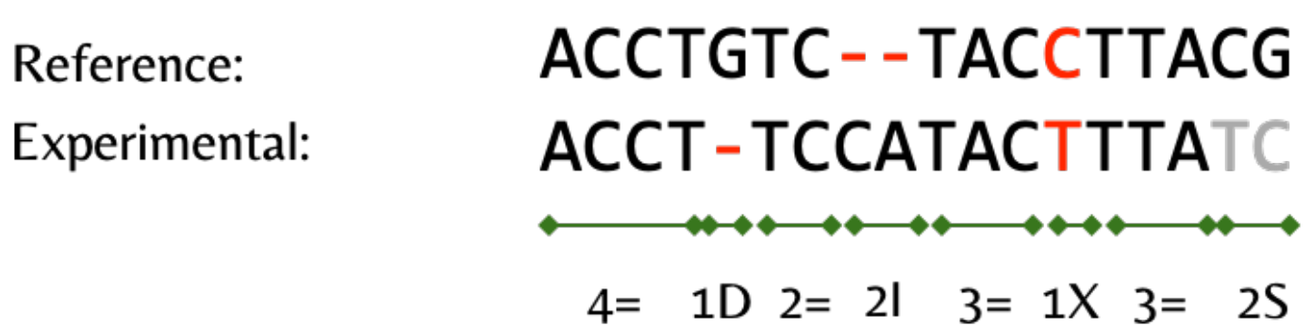
https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

The extended CIGAR string: M become = and X

Operation	Meaning
=	Exact match
X	Mismatch
D	Deletion w.r.t. reference
I	Insertion w.r.t. reference
N	Split or spliced alignment
S	Soft-clipping
H	Hard-clipping
P	Padding



CIGAR string: 4=1D2=2I3=1X3=2S

https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

The FLAG column

Sequence ID	FLAG	CHROM	POS
ST-E00223:32:H5J57CCXX:6:2123:15189:52872	97	1	10001
ST-E00223:46:HG7V5CCXX:2:1116:12601:22862	1123	1	10006
ST-E00223:32:H5J57CCXX:5:2208:10074:43308	99	1	10008
ST-E00223:46:HG7V5CCXX:5:2119:12936:64896	99	1	10013
ST-E00223:32:H5J57CCXX:1:1205:17290:54577	99	1	10019
ST-E00223:32:H5J57CCXX:6:1115:16844:11013	81	1	10026
ST-E00223:32:H5J57CCXX:7:2113:18935:32356	99	1	10032
ST-E00223:46:HG7V5CCXX:6:2117:3082:44239	99	1	10040
ST-E00223:46:HG7V5CCXX:5:2213:10744:58813	163	1	10074
ST-E00223:32:H5J57CCXX:4:1220:14651:8868	99	1	10086

Genomics105

Mapping and Alignment 101

The FLAG score

base2	base10	base16	Meaning	Applies to:
00000000001	1	0x0001	The read originated from a paired sequencing molecule	Both
00000000010	2	0x0002	The read is mapped in a **proper** pair	Pairs only
00000000100	4	0x0004	The query sequence itself is unmapped	Both
00000001000	8	0x0008	The query's mate is unmapped	Pairs only
00000010000	16	0x0010	Strand of the query (0 for forward; 1 for reverse strand)	Both
00000100000	32	0x0020	Strand of the query's mate	Pairs only
00001000000	64	0x0040	The query is the first read in the pair	Pairs only
00010000000	128	0x0080	The read is the second read in the pair	Pairs only
00100000000	256	0x0100	The alignment is not primary	Both
01000000000	512	0x0200	The read fails platform/vendor quality checks	Both
10000000000	1024	0x0400	The read is either a PCR duplicate or an optical duplicate	Both

Genomics105

Mapping and Alignment 101

ST-E00223:32:H5J57CCXX:4:1220:14651:8868 99 1 10086

26+25+21+20 = 64+32+2+1 = 99

base2	base10	base16	Meaning	Applies to:
00000000001	1	0x0001	The read originated from a paired sequencing molecule	Both
00000000010	2	0x0002	The read is mapped in a **proper** pair	Pairs only
00000000100	4	0x0004	The query sequence itself is unmapped	Both
00000001000	8	0x0008	The query's mate is unmapped	Pairs only
00000010000	16	0x0010	Strand of the query (0 for forward; 1 for reverse strand)	Both
00000100000	32	0x0020	Strand of the query's mate	Pairs only
00001000000	64	0x0040	The query is the first read in the pair	Pairs only
00010000000	128	0x0080	The read is the second read in the pair	Pairs only
00100000000	256	0x0100	The alignment is not primary	Both
01000000000	512	0x0200	The read fails platform/vendor quality checks	Both
10000000000	1024	0x0400	The read is either a PCR duplicate or an optical duplicate	Both

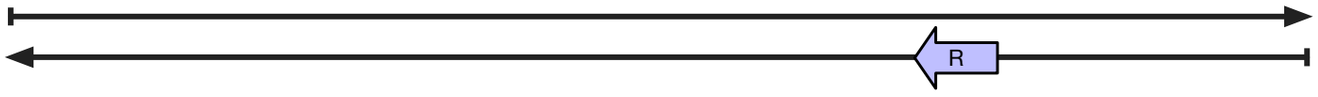
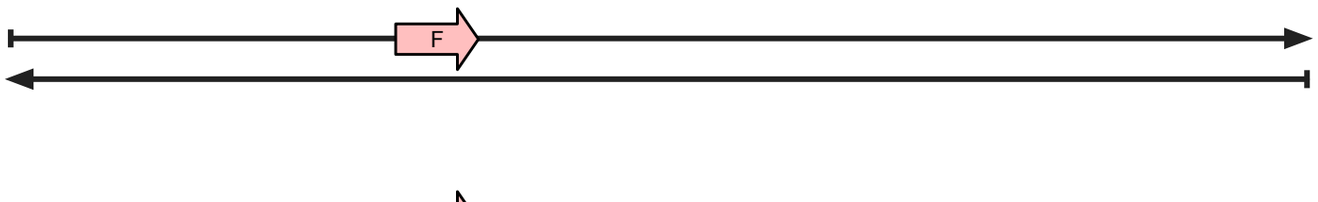
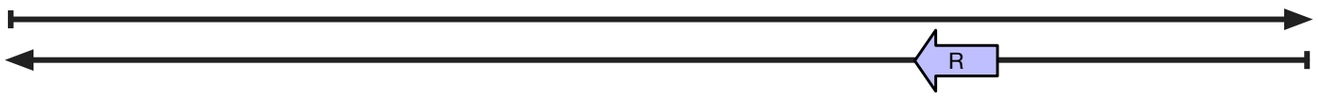
https://github.com/quinlan-lab/applied-computational-genomics

Genomics105

Mapping and Alignment 101

SAM format

Example

SAM Record #	Strand	FLAG Values	Pair-End Reads
01	Reverse (Query)	83	 R1
02	Forward (Mate)	163	 R2
03	Forward (Query)	99	 R1
04	Reverse (Mate)	147	 R2

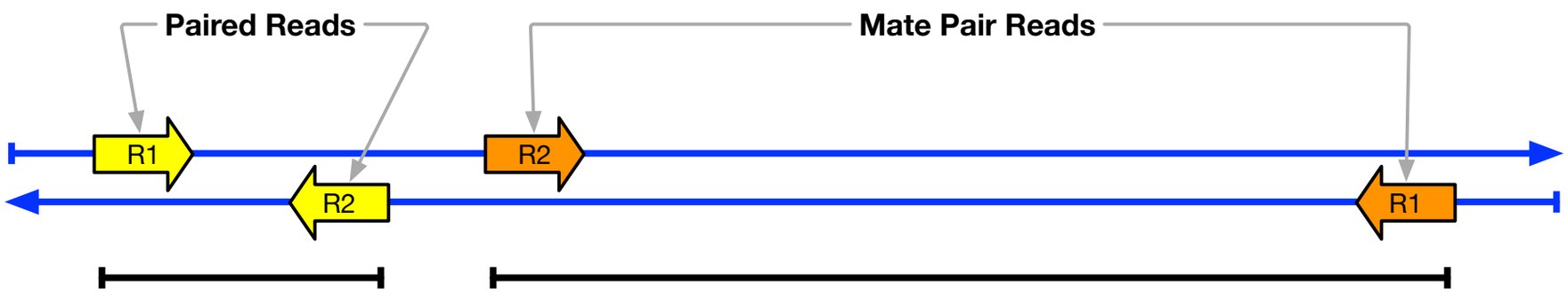
Bit		Description
1	0x1	template having multiple segments in sequencing
2	0x2	each segment properly aligned according to the aligner
4	0x4	segment unmapped
8	0x8	next segment in the template unmapped
16	0x10	SEQ being reverse complemented
32	0x20	SEQ of the next segment in the template being reverse complemented
64	0x40	the first segment in the template
128	0x80	the last segment in the template
256	0x100	secondary alignment
512	0x200	not passing filters, such as platform/vendor quality controls
1024	0x400	PCR or optical duplicate
2048	0x800	supplementary alignment

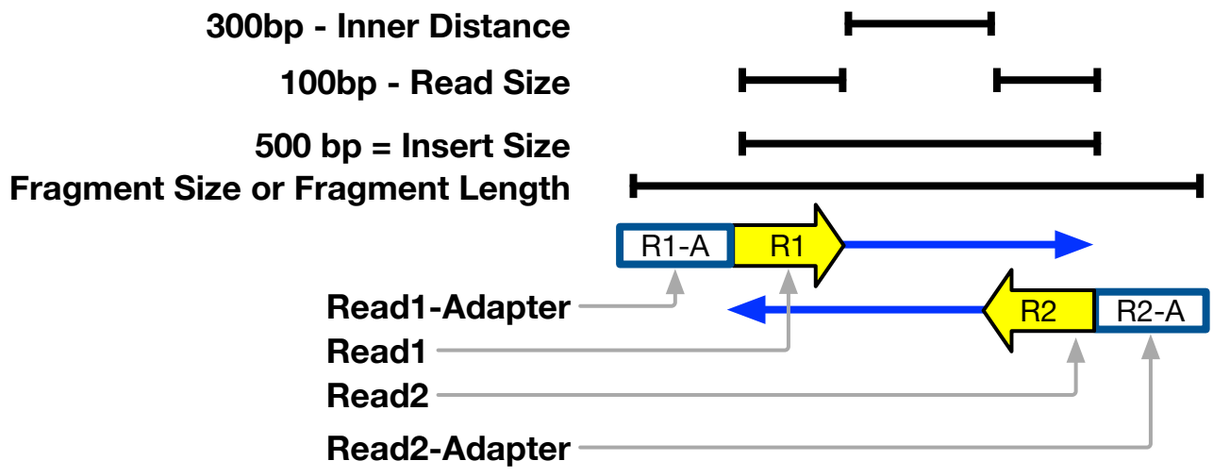
Decoding SAM Flags

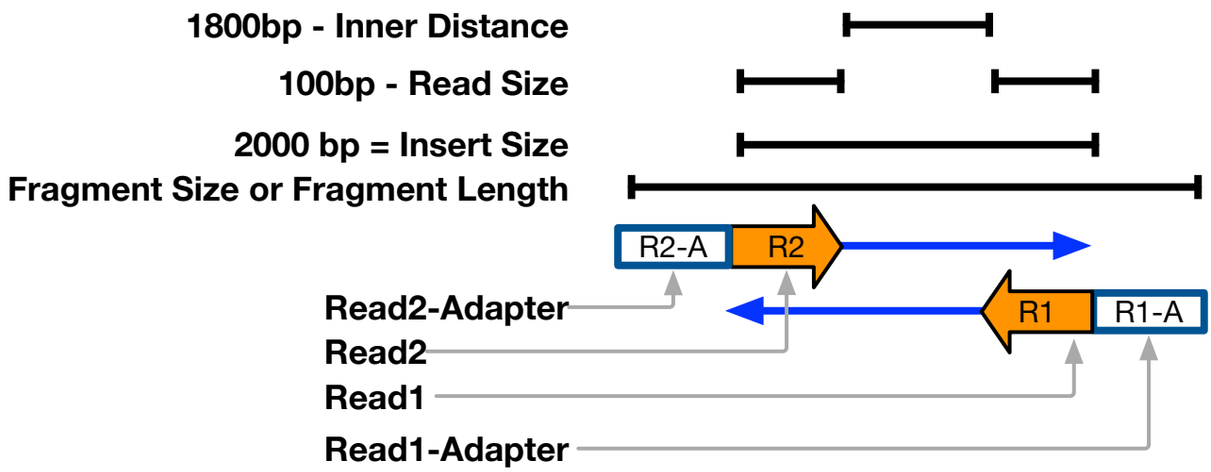
Genomics105

Mapping and Alignment 101

FIRST FUNDAMENTAL CONCEPT OF READ MAPPING







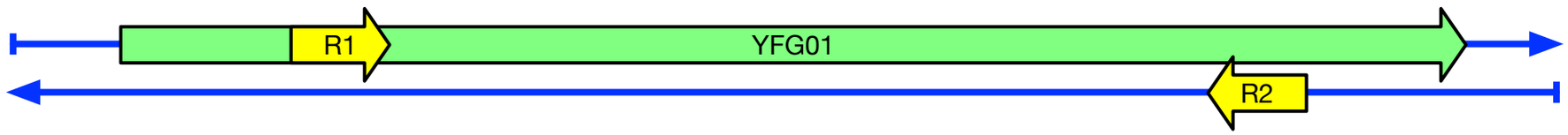
Genomics105

Mapping and Alignment 101

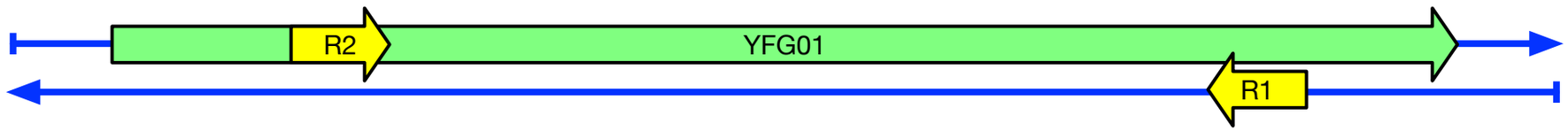
SECOND FUNDAMENTAL CONCEPT OF READ MAPPING

Do Not Confuse R1/R2 with Forward/Reverse, Watson/Crick, and/or Coding/Template Strands

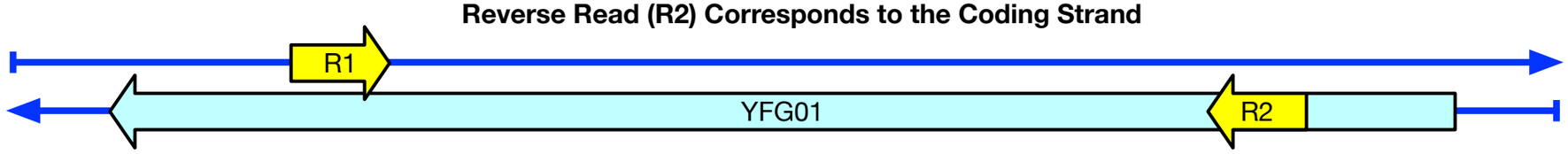
YFG01 **Gene Lives in Watson Strand**
Forward Read (R1) Corresponds to the Coding Strand
Reverse Read (R2) Corresponds to the Template Strand



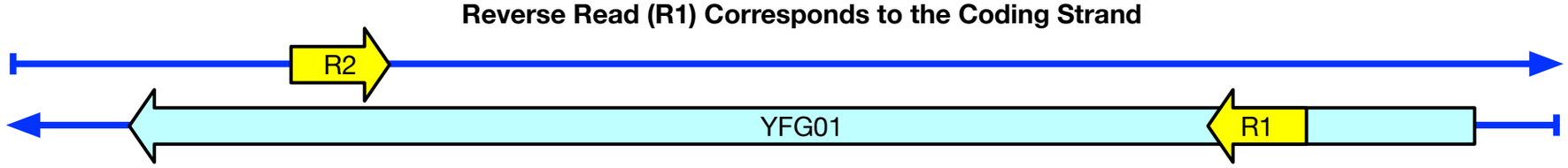
YFG01 **Gene Lives in Watson Strand**
Forward Read (R2) Corresponds to the Coding Strand
Reverse Read (R1) Corresponds to the Template Strand



YFG01 **Gene Lives in Crick Strand**
Forward Read (R1) Corresponds to the Template Strand
Reverse Read (R2) Corresponds to the Coding Strand



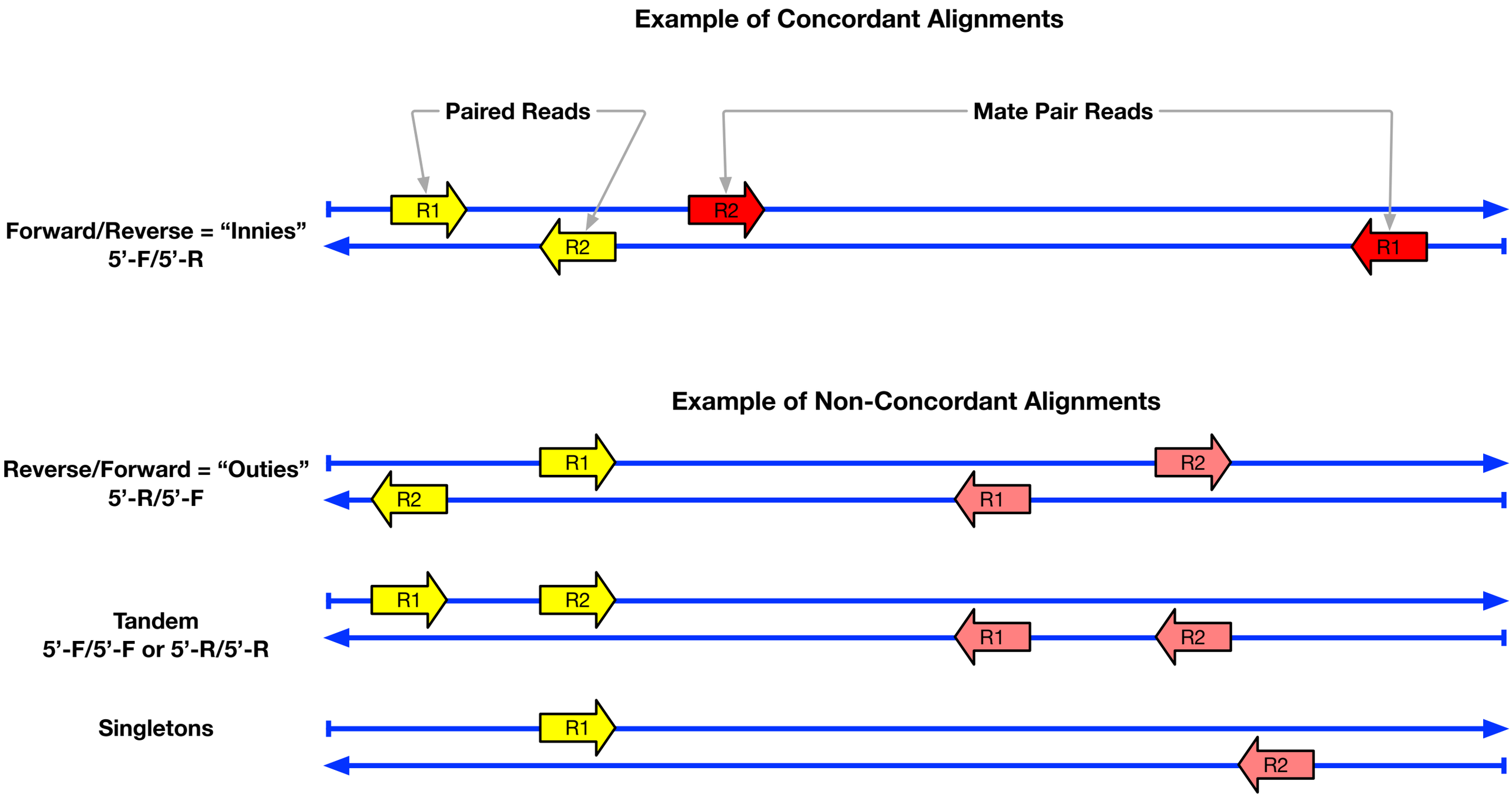
YFG01 **Gene Lives in Crick Strand**
Forward Read (R2) Corresponds to the Template Strand
Reverse Read (R1) Corresponds to the Coding Strand



Genomics105

Mapping and Alignment 101

EXAMPLES OF CONCORDANT AND NON-CONCORDANT ALIGNMENTS

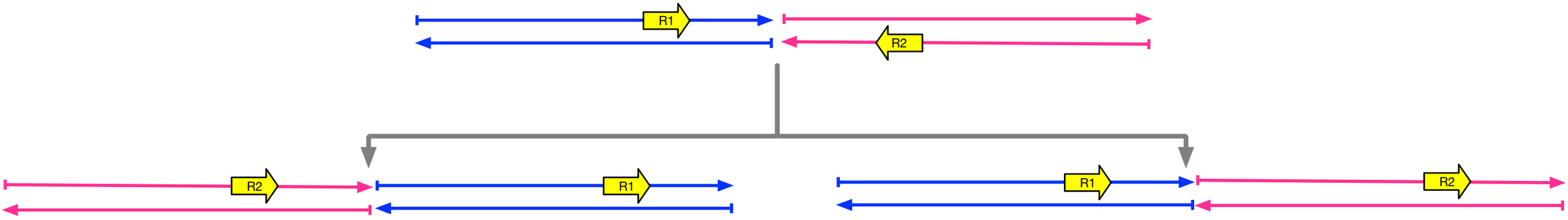


Genomics105

Mapping and Alignment 101

EXAMPLES OF CONCORDANT AND NON-CONCORDANT ALIGNMENTS

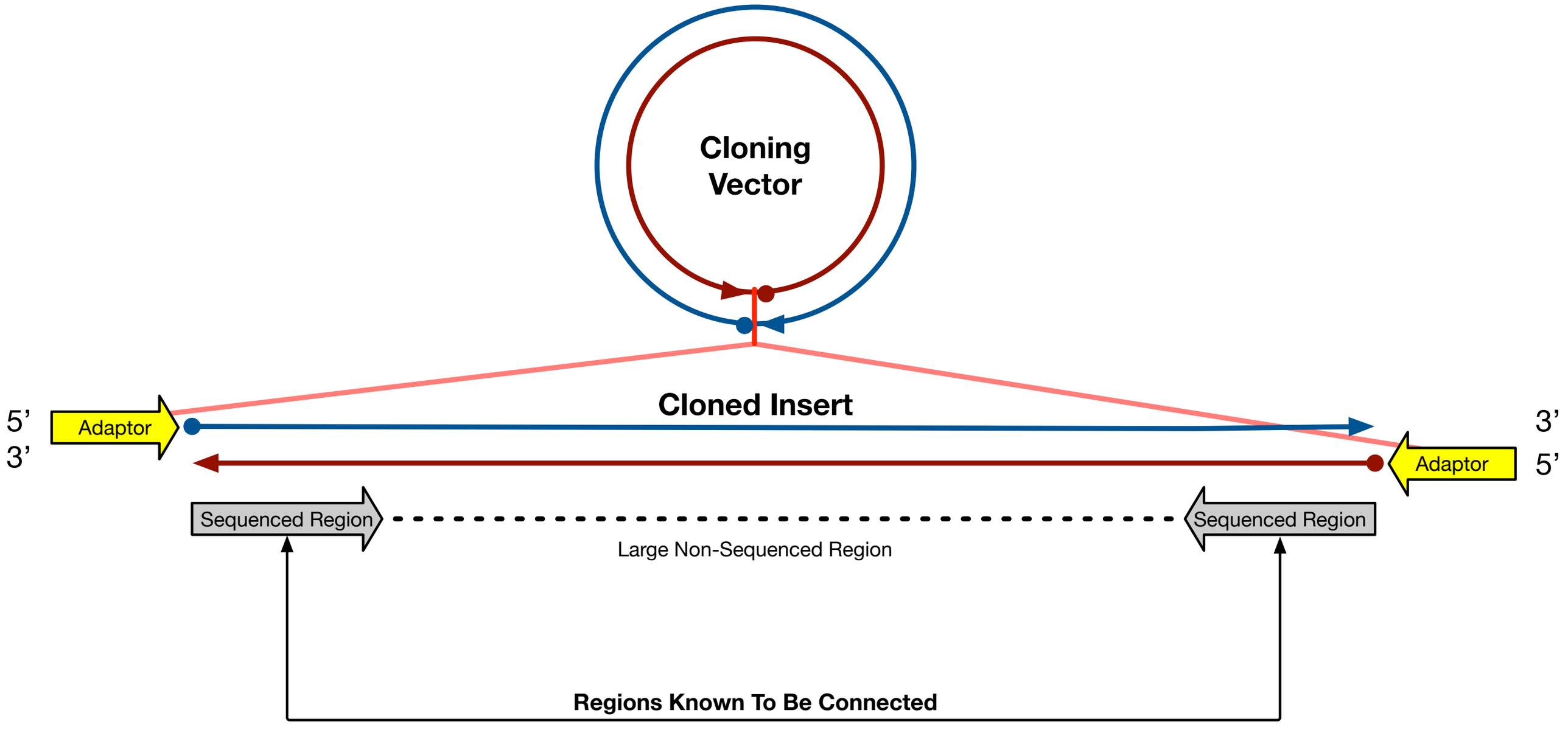
Example of a Generation of a Non-Concordant Alignment



Genomics105

Mapping and Alignment 101

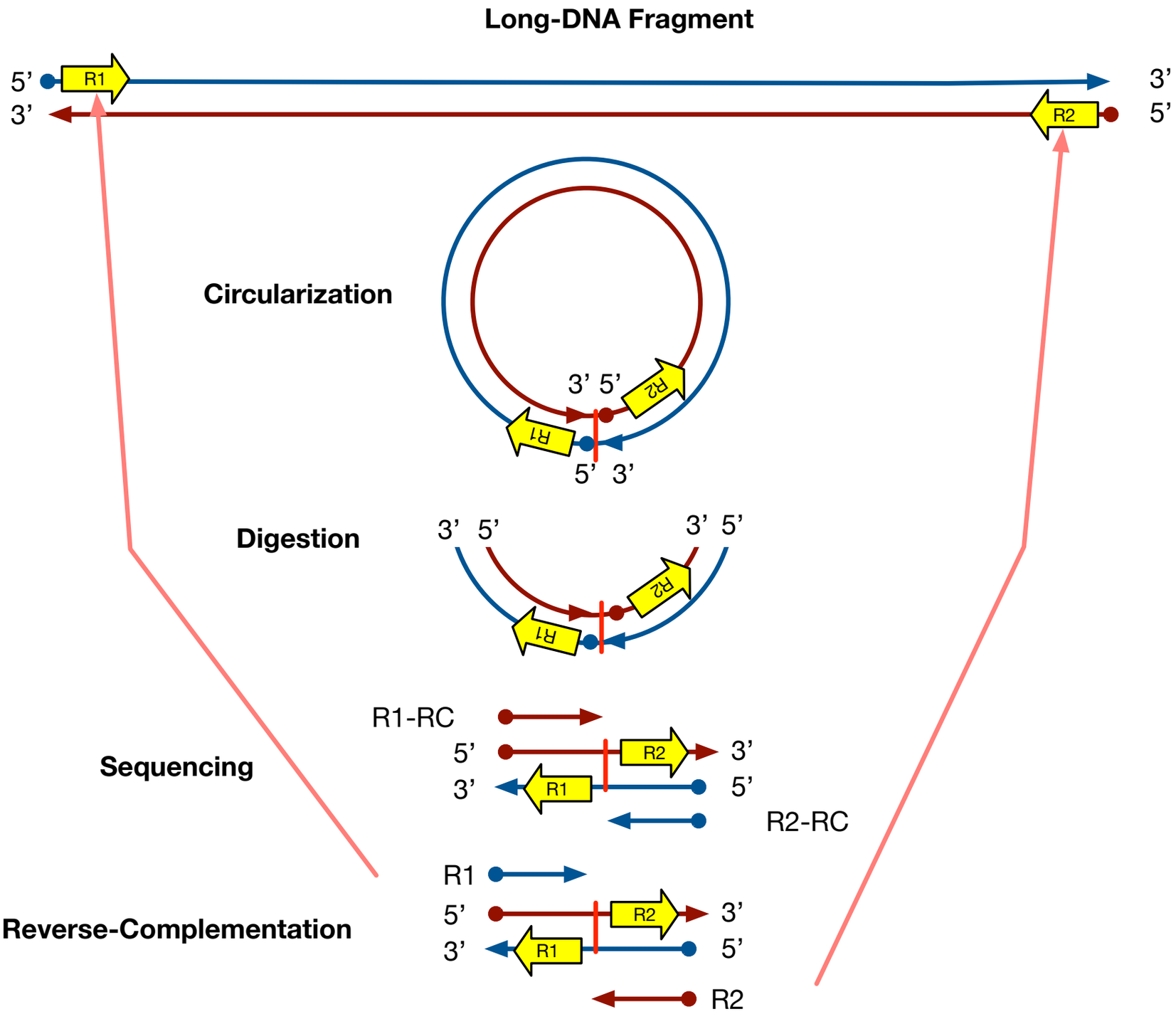
MATE-PAIRS 101



Genomics105

Mapping and Alignment 101

MATE-PAIRS 101



Genomics105

Mapping and Alignment 101

USING MATE PAIRS: EXAMPLE01

Mate pair sequencing for the detection of chromosomal aberrations in patients with intellectual disability and congenital malformations

Sarah Vergult [1], Ellen Van Binsbergen [2], Tom Sante [1], Silke Nowak [1], Olivier Vanakker [1], Kathleen Claes [1], Bruce Poppe [1], Nathalie Van der Aa [3], Markus J van Roosmalen [2], Karen Duran [2], Masoumeh Tavakoli-Yaraki [2], Marielle Swinkels [2], Marie-José van den Boogaard [2], Mieke van Haelst [2], Filip Roelens [4], Frank Speleman [1], Edwin Cuppen [2], Geert Mortier [5], Wigard P Kloosterman [2], Björn Menten [1]

Genomics105

Mapping and Alignment 101

USING MATE PAIRS: EXAMPLE02

Mayo Clinic launches first-in-world mate-pair sequencing test that locates "breakpoints" of chromosome rearrangements

This novel chromosome test finds the "breaks" in gene rearrangements that other tests can't.

Genomics105

Mapping and Alignment 101

Mapping Qualities When Considering Base Calling Errors

- **Modified from the Mapping qualities Blog from David Tangs Blog**
- **To model base calling errors we can use the Binomial distribution; if we expect there to be 1 base calling error in 100 bps, We can calculate the probability of an error for a read of 25 nt as such using R**

```
R

#Probability of success (1 error in 100 bases) = 0.99
#Number of trials (each base is a trial) = 25
#no base calling errors, i.e. 25 successes

dbinom(x = 25, size=25, prob=0.99)
[1] 0.7778214

#one base calling error, i.e. 24 successes
dbinom(x = 24, size=25, prob=0.99)
[1] 0.1964195

#two base calling errors, i.e. 23 successes
dbinom(x = 23, size=25, prob=0.99)
[1] 0.02380843
```

Genomics105

Mapping and Alignment 101

Mapping Qualities When Considering Base Calling Errors

- If we expect 1 base calling error in 100 bps, the probability of making two base calling errors in 25 bps is quite low.
- Calculating the posterior probability that the best alignment is actually correct in R

```
#the posterior probability that the best alignment is correct
p = 0.99
dbinom(x=25,size=25,prob=p)/(dbinom(x=25,size=25,prob=p)+(5*dbinom(x=24,size=25,prob=p)))
[1] 0.4419643
```

- In reality base calling is much more accurate than 1 error in 100 bases, which is a Phred quality score of 20. If we changed the base calling error rate to 1 in 1000 (Phred score of 30):

```
p = 0.999
dbinom(x=25,size=25,prob=p)/(dbinom(x=25,size=25,prob=p)+(5*dbinom(x=24,size=25,prob=p)))
[1] 0.88879
```

- Then the posterior probability that the best alignment is correct improves to 0.88879. Using a base calling error rate of 1 in 10000 (Phred score of 40):

```
p = 0.9999
dbinom(x=25,size=25,prob=p)/(dbinom(x=25,size=25,prob=p)+(5*dbinom(x=24,size=25,prob=p)))
[1] 0.9876531
```

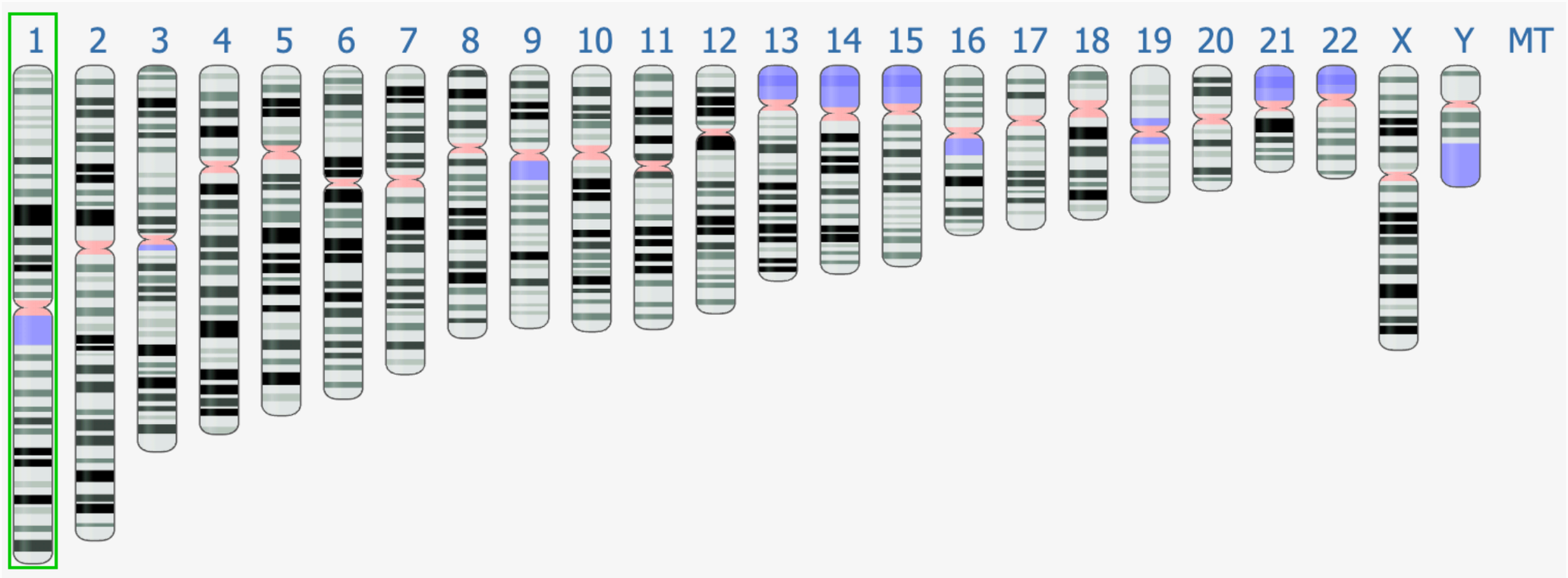
- This improves the probability to 0.9876531, which is a ~0.012 probability that the alignment is incorrect, which is around the same ball park to the BWA mapping quality of 16, which is a 0.025 probability that the alignment is incorrect

Genomics105

Mapping and Alignment 101

Sequence Alignment Software

Aligner	Approach	Applications	Availability
BWA-mem	Burrows-Wheeler	DNA, SE, PE, SV	open-source
Bowtie2	Burrows-Wheeler	DNA, SE, PE, SV	open-source
Novoalign	hash-based	DNA, SE, PE	free for academic use
TopHat	Burrows-Wheeler	RNA-seq	open-source
STAR	hash-based (reads)	RNA-seq	open-source
GSNAP	hash-based (reads)	RNA-seq	open-source



Genomics105

BIOL647
Digital Biology

Rodolfo Aramayo