2013 - BMMB 597D: Analyzing Next Generation Sequencing Data

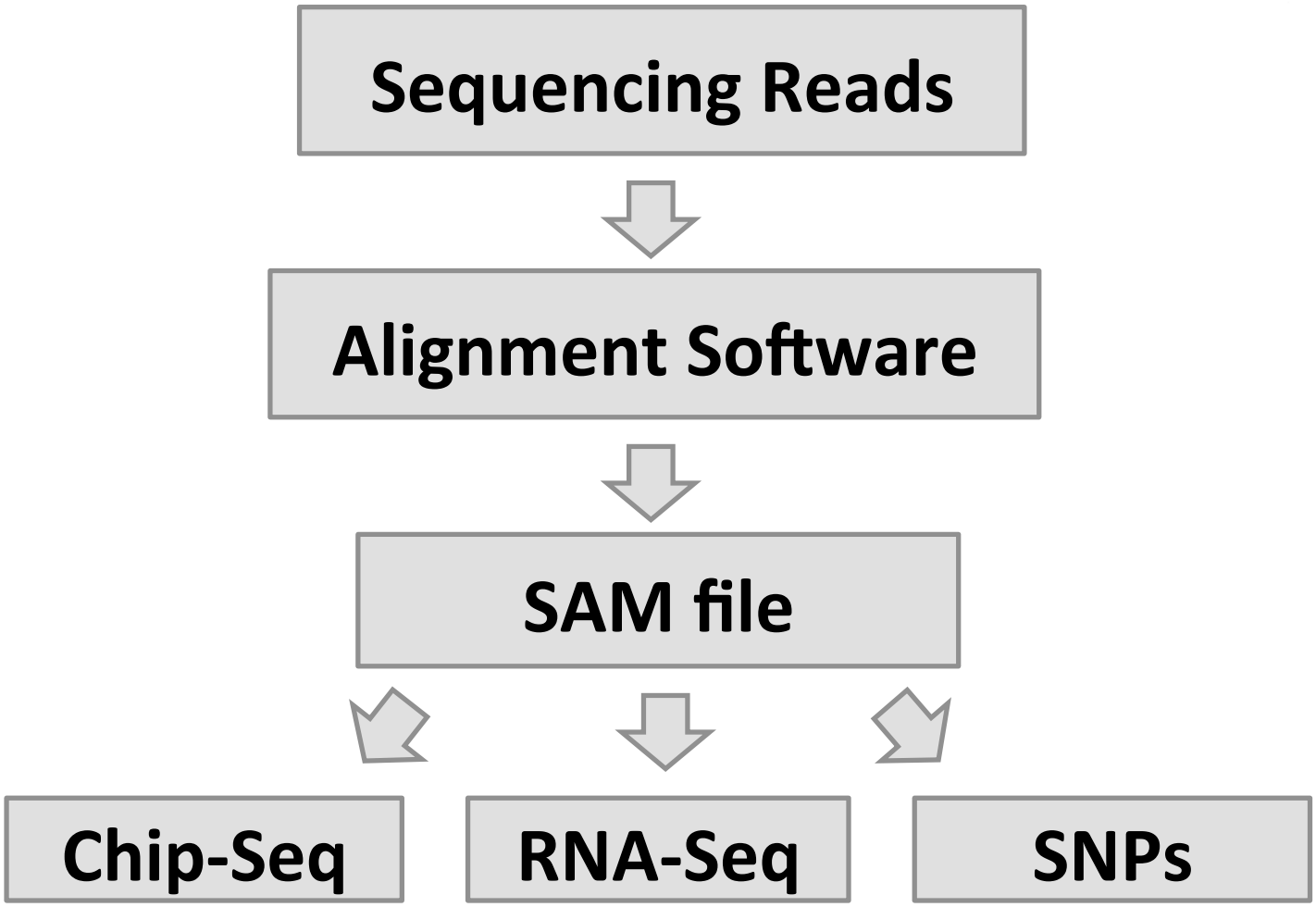
Week 8, Lecture 15

István Albert

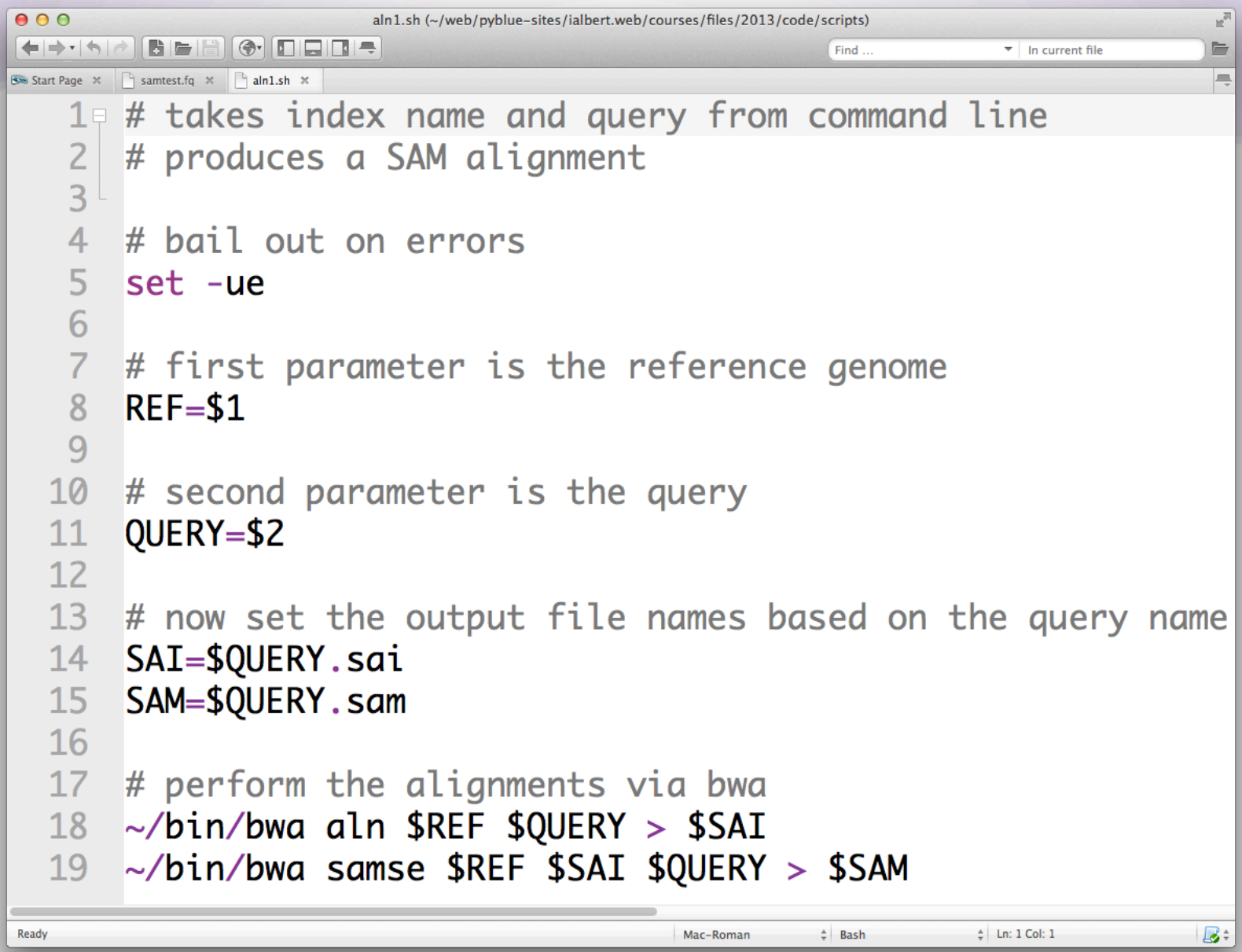
Biochemistry and Molecular Biology
and Bioinformatics Consulting Center

Penn State

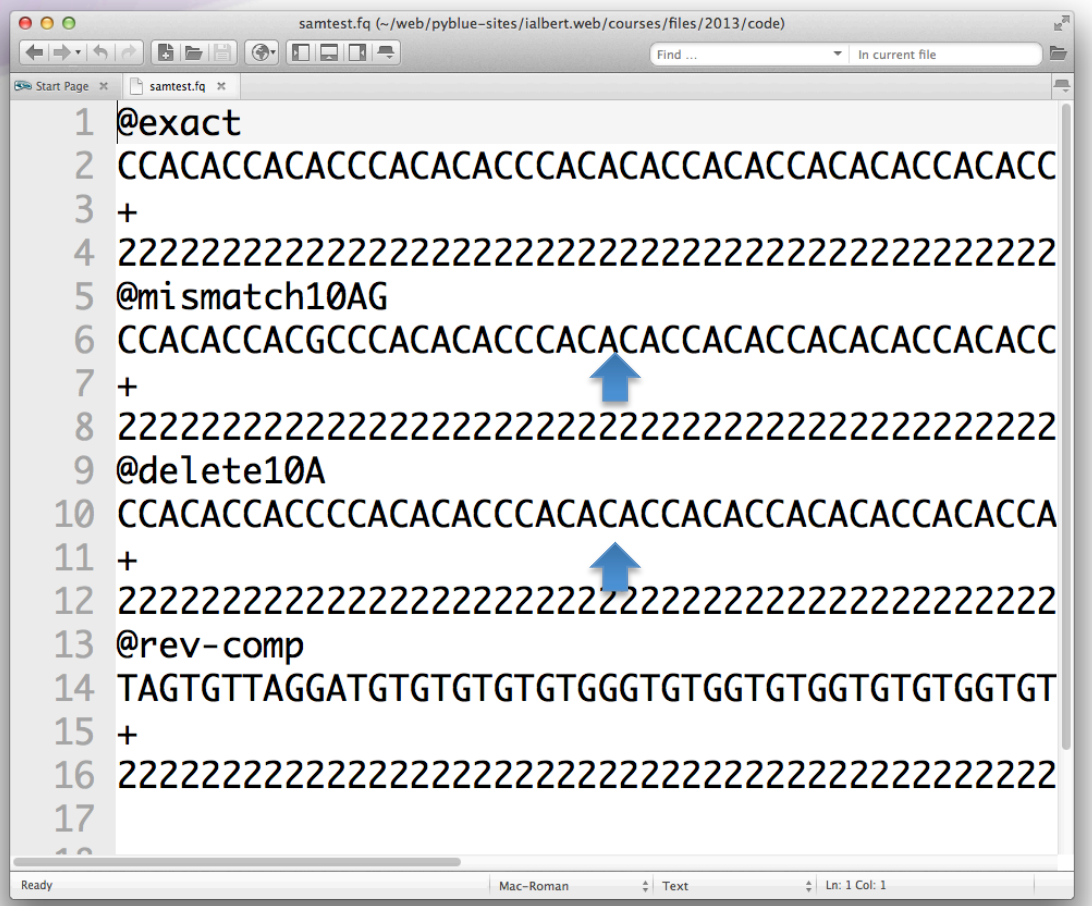
The majority of data analyses split off after generating a SAM file



Create a script that creates an alignment



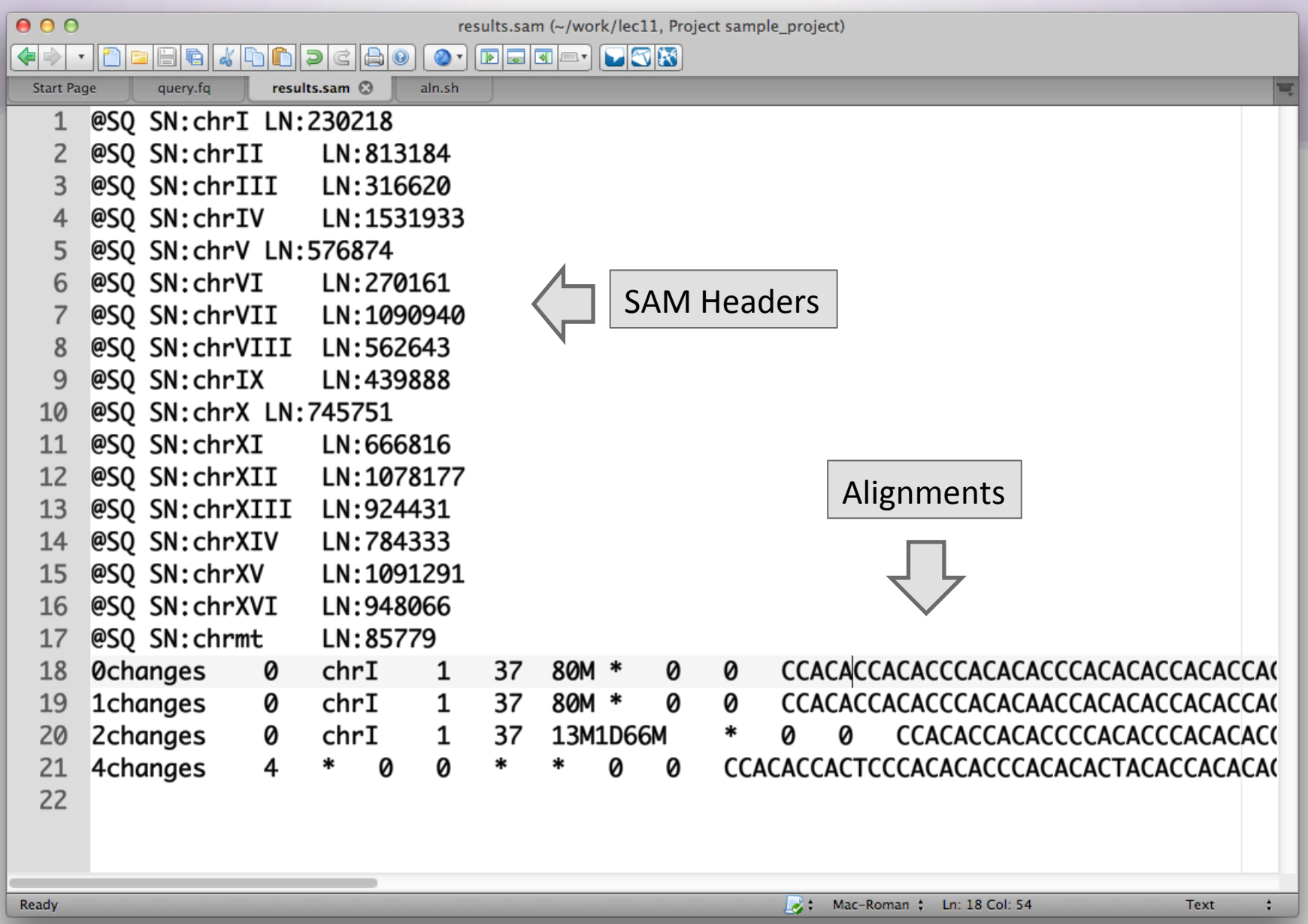
A test query file



The first line from the yeast genome – with a twist:

- 1st record is exact match
- 2nd record has a mismatch at position 10
- 3rd record has a deletion at position 10
- 4th record is the reverse complement of record 1

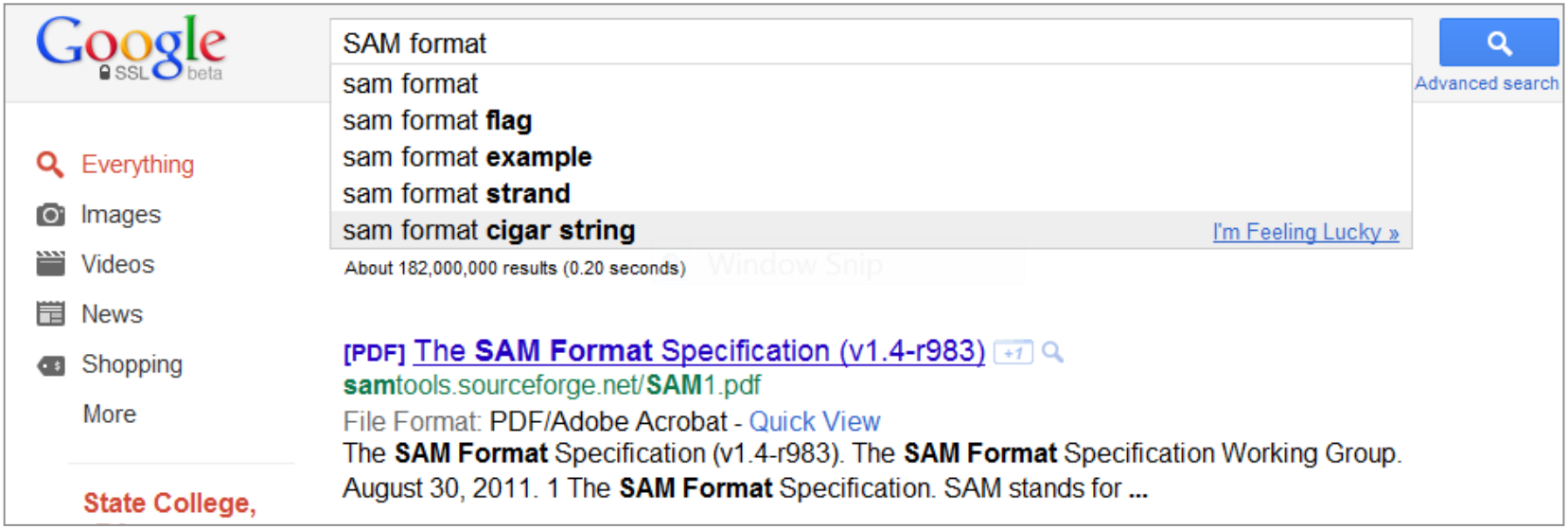
The structure of the SAM file



SAM format: tabular text format

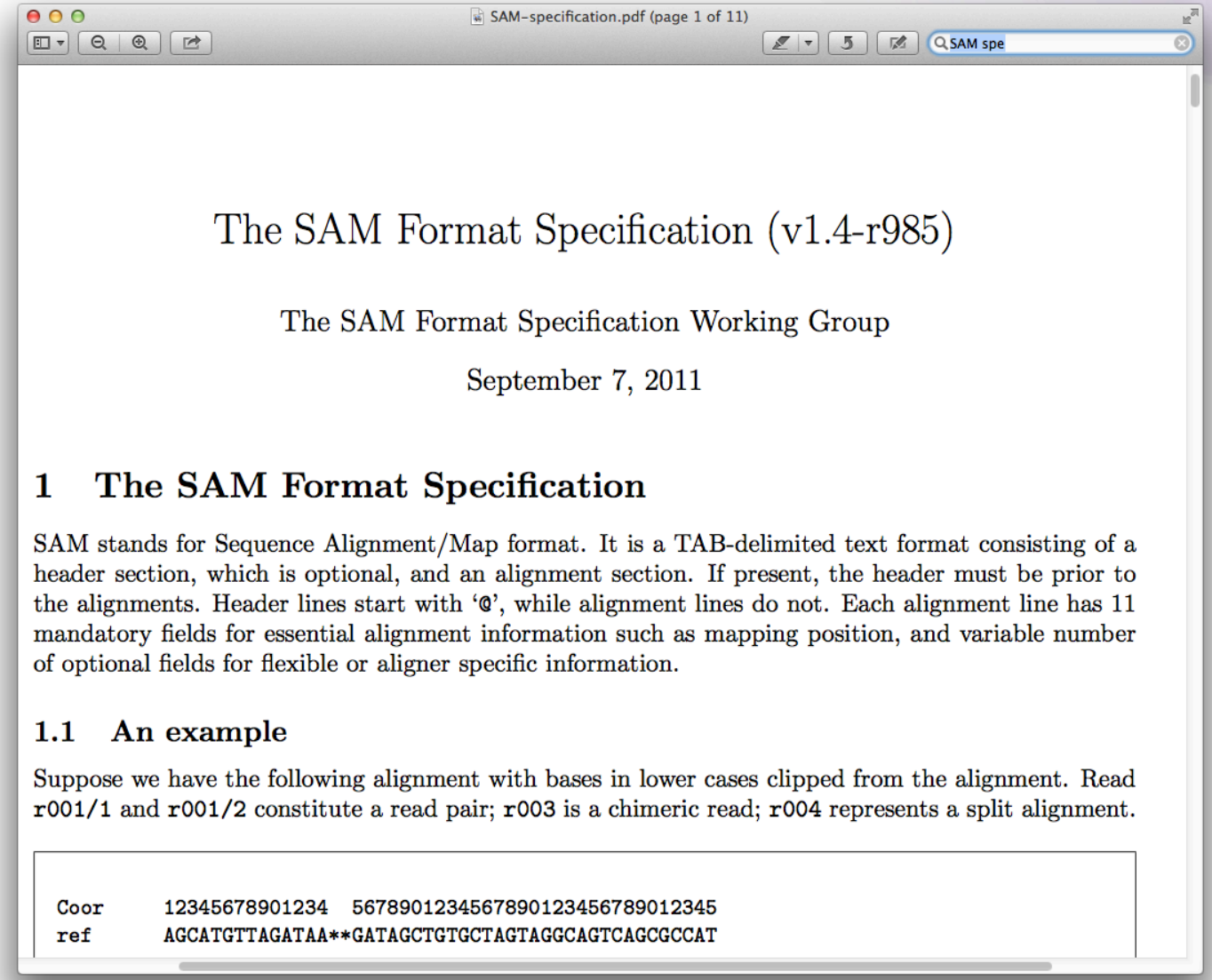
Published as
The Sequence Alignment/Map format and SAMtools by **Heng Li** et al
Bioinformatics 25, Volume 25, Issue 16, 2009

"A TAB-delimited text format consisting of a header section, which is optional, and an alignment section."



Resource: SAM specification

11 required column + optional fields



The SAM Format Specification (v1.4-r985)

The SAM Format Specification Working Group

September 7, 2011

1 The SAM Format Specification

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.

1.1 An example

Suppose we have the following alignment with bases in lower cases 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
```

An aligment consist of 11 tab delimited columns

1.4 The alignment section: mandatory fields

Each alignment line has 11 mandatory fields. These fields always appear in the same order and must be present, but their values can be '0' or '*' (depending on the field) if the corresponding information is unavailable. The following table gives an overview of the mandatory fields in the SAM 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

Column 1 and 2: QNAME and FLAG (Query name and bitwise flags)

QNAME: the name of the query sequence

2. FLAG: bitwise FLAG. Each bit is explained in the following table:

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 reversed
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

- Bit 0x4 is the only reliable place to tell whether the segment is unmapped. If 0x4 is set, no assumptions can be made about RNAME, POS, CIGAR, MAPQ, bits 0x2, 0x10 and 0x100 and the bit 0x20 of the next segment in the template.

Column 2: FLAG
the bitwise representation

1 = 00000001 → paired end read
2 = 00000010 → mapped as proper pair
4 = 00000100 → unmappable read
8 = 00001000 → read mate unmapped
16 = 00010000 → read mapped on reverse strand

The flag **11** → **1** + **2** + **8** = **0001011** (conditions 1, 2 and 8 satisfied)

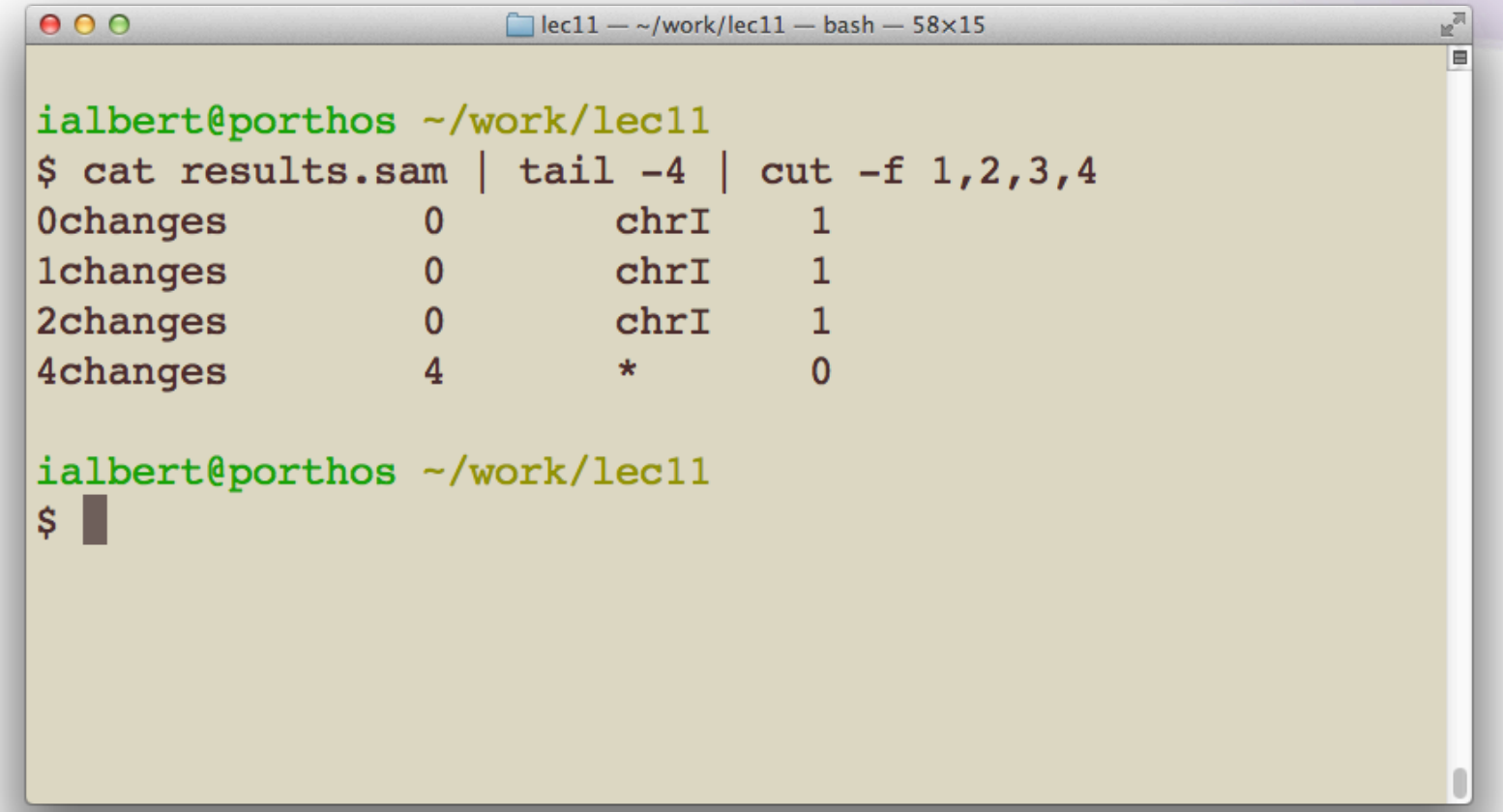
It is used to save space – but it does make things a bit more difficult.

Usually very few flags are needed in practice – 0, 4, 16 are the most generic ones

If you need to construct a more complex flag search for explain SAM flags:

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

Columns 3, 4: RNAME and POS Reference and Position



Column 4 POS: **1-based leftmost mapping POSition of the first matching base**.

Very important to remember later when we need to find the 5' end (the actual start)

Column 5: MAPQ - Mapping Quality

- Phred score, identical to the quality measure in the fastq file. quality **Q**, probability **P**:

$$P = 10 \wedge (-Q / 10.0)$$

If **Q=30**, **P=1/1000** → on average, one of out 1000 alignments will be wrong

As good as this sounds it is not easy to compute such a quality.

Details of the mapping quality computation – hard to find good answers

- Tool specific – there is no standard of what it should be
- The repeat structure of the reference. Reads falling in repetitive regions usually get very low mapping quality.
- The base quality of the read. Low quality means the observed read sequence is possibly wrong, and wrong sequence may lead to a wrong alignment.
- The sensitivity of the alignment algorithm. The true hit is more likely to be missed by an algorithm with low sensitivity, which also causes mapping errors.
- Paired end or not. Reads mapped in pairs are more likely to be correct.

(*from the MAQ manual*)

BWA specific high scores

A read alignment with a mapping quality 30 or above usually implies

- The overall base quality of the read is good.
- The best alignment has few mismatches.
- The read has few or just one `good' hit on the reference, which means the current alignment is still the best even if one or two bases are actually mutations or sequencing errors.

BWA specific low scores

Surprisingly difficult to track down the exact behavior

- Q=0 → if a read can be aligned equally well to multiple positions, BWA will randomly pick one position and give it a mapping quality zero.

- Q=25 → the edit distance equals mismatches and is greater than zero

Column 6: CIGAR

- CIGAR = Compact Idiosyncratic Gapped Alignment Report

6. CIGAR: CIGAR string. The CIGAR operations are given in the following table (set '*' if unavailable):

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

- H can only be present as the first and/or last operation.
- S may only have H operations between them and the ends of the CIGAR string.
- For mRNA-to-genome alignment, an N operation represents an intron. For other types of alignments, the interpretation of N is not defined.

Columns 7, 8, 9: RNEXT, PNEXT, TLEN (used in paired end read sequencing)

- **RNEXT**: the name of the pair
- **PNEXT**: the position of the pair
- **TLEN**: the distance between the leftmost positions of the pairs

We will discuss these in more detail later.

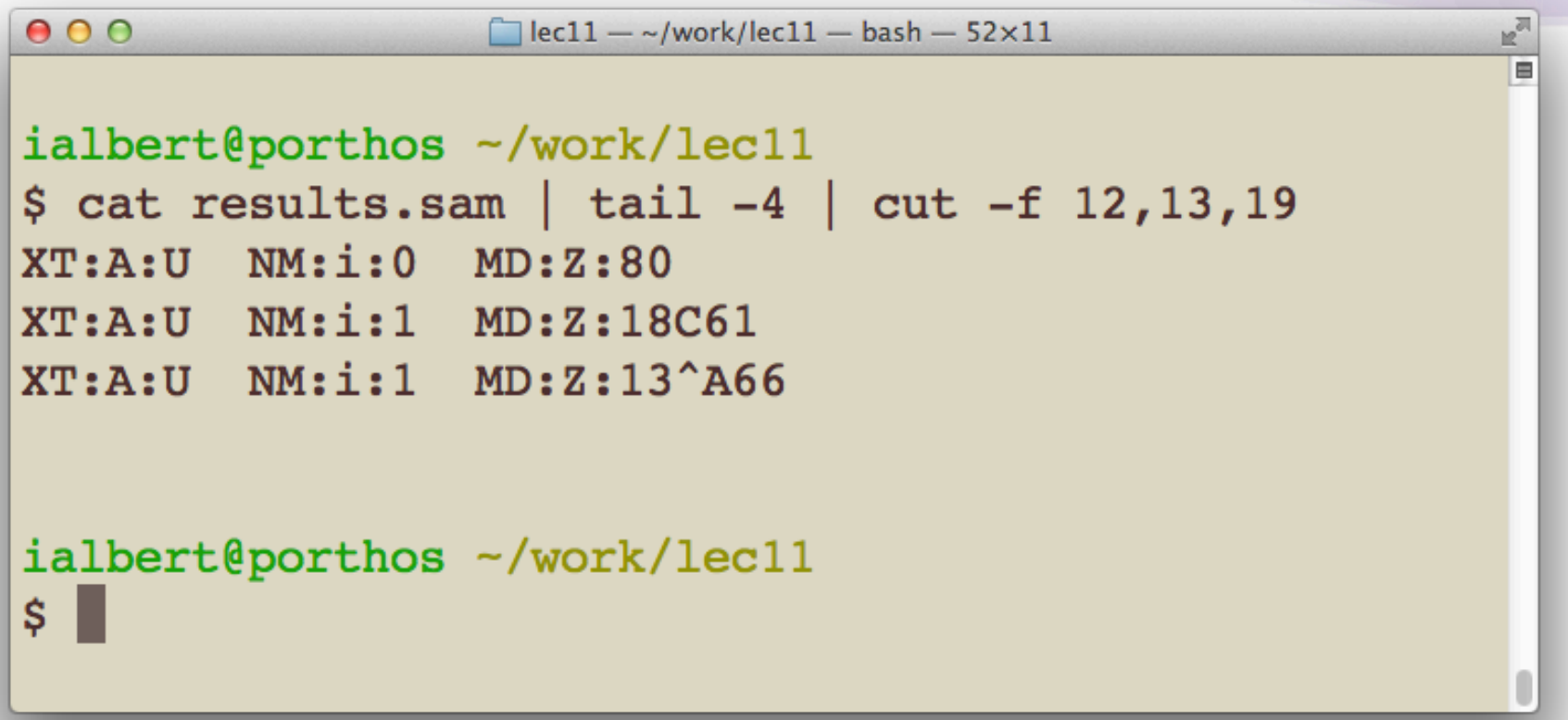
These can show the position of reads that are distant – allow us to infer genomic variations

Column 10, 11

- SEQ: the query sequence
- QUAL: the phred encoded quality sequence

SEQ may contain the original sequence or the segment it was aligned to. Not all tools do the same thing.

Column 12 and beyond



Specific information about the alignment process that the tools was able to establish.
more details in later lectures

Homework 15

Generate a SAM file using dataset **lect-15.fq.gz** as query and the yeast genome as reference then answer the following questions:

1. How many reads are in the data?

2. How many reads are unmapped?

3. How may different types of quality scores can you observe? (hint: cut, sort, uniq)?

4. How many different CIGAR string do you see? What are the 5 most common CIGAR strings?

5. How many reads align to the reverse strand?

6. How many reads have a MAPQ (mapping quality) of 0 and what does that value mean in a SAM file from BWA?