SVs and Genome Arithmetic

Michael Schatz

Feb 27, 2018

Lecture 9: Applied Comparative Genomics

Assignment 4: Due Thursday March 1

Assignment 4: Read mapping and variant calling

Assignment Date: Thursday, Feb. 22, 2018
Due Date: Thursday, Mar. 1, 2018 @ 11:59pm

Assignment Overview

In this assignment, you will align reads to a reference genome to call SNPs and short indels. Then, you will perform an experiment to empirically determine the "mappability" of a genomic region. Finally, you will investigate some empirical behavior of the binomial test for heterozygous variant calling.
As a reminder, any questions about the assignment should be posted to Piazza. Don't forget to read the **Resources** section at the bottom of the page!

Question 1. Small Variant Analysis [XX pts]

Download chromosome 22 from build 38 of the human genome from here:
http://hgdownload.cse.ucsc.edu/goldenPath/hg38/chromosomes/chr22.fa.gz

Download the read set from here:
http://schatzlab.cshl.edu/data/teaching/sample.tgz

For this question, you may find this tutorial helpful:
http://clavius.bc.edu/~erik/CSHL-advanced-sequencing/freebayes-tutorial.html

- 1a. How many reads align to the reference? How many reads did not align? How many aligned reads had a mate that did not align (AKA singletons)? Count each read in a pair separately.
  [Hint: Build the index using `bowtie2-build`, align reads using `bowtie2`, analyze with `samtools flagstat`.]
- 1b. How many reads are mapped to the reverse strand? Count each read in a pair separately.
  [Hint: Find out what SAM flags mean here and use samtools view.]
- 1c. How many high-quality (QUAL > 20) single nucleotide and indel variants does the sample have? Of the high-quality SNPs, what is the transition / transversion ratio? Of the indels, how many are insertions and how many are deletions?
  [Hint: Identify variants using `freebayes` - sort the SAM file first. Filter using `bcftools filter`, and summarize using `bcftools stats`.]
- 1d. Does the sample have any nonsense or missense mutations?
  [Hint: try the Variant Effect Predictor using the `Gencode basic transcripts`.]

Question 2. Read Mapping Uncertainty [XX pts]

For the region chr22:21000000-22000000 of the reference sequence for chromosome 22, extract every substring of length 35. Format the substrings as a FASTA file and use read names that indicate the origin. (No need to construct quality values or read pairs: use bowtie2 with `-f` and `-U` respectively). Make a new index and align these "reads" to chr22:21000000-22000000.
[Hint: On the command line or in a script, load the sequence once and extract substrings in a loop.]

- 2a. How many reads align more than one time to the reference? How many reads did not align?

Genotyping Theory

Heterozygous variant (3/7)

Homozygous variant (6/6)

```
                                                    GGTATAC…
…CCATAG     TGTGCGCCC    CGGAAATTT CGGTATAC
…CCAT    CTATGTGCG      TCGGAAATT  CGGTATAC
…CCAT GGCTATGTG     CTATCGGAAA    GCGGCATA
…CCA AGGCTATAT     CCTATCGGA    TTGCGGTA   C…
…CCA AGGCTATAT  GCCCTATCG     TTTGCGGT    C…
…CC  AGGCTATAT  GCCCTATCG  AAATTTGC    ATAC…
…CC TAGGCTATA GCGCCCTA     AAATTTGC GTATAC…
```

Subject

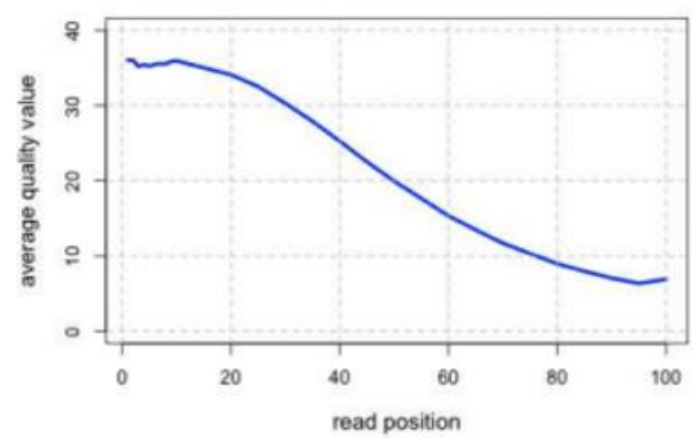
Reference

```
…CCATAGGCTATATGCGCCCTATCGGCAATTTGCGGTATAC…
```

Error or Het (1/7)?

- If there were no sequencing errors, identifying SNPs would be very easy: any time a read disagrees with the reference, it must be a variant!

- Sequencing instruments make mistakes
  - Quality of read decreases over the read length



- A single read differing from the reference is probably just an error, but it becomes more likely to be real as we see it multiple times

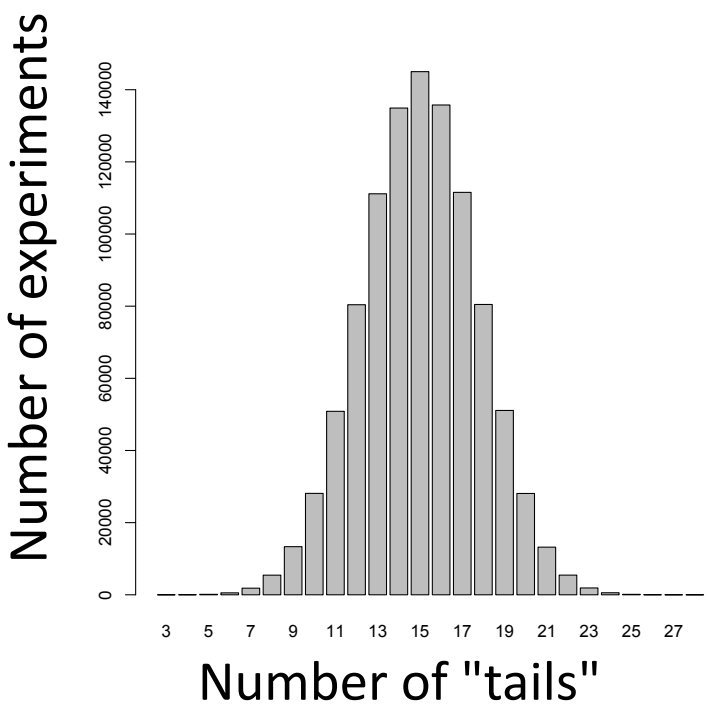
The Binomial Distribution: Adventures in Coin Flipping

P(heads) = 0.5

P(tails) = 0.5

Aaron Quinlan

So, with 30 tosses (reads), we are much more likely to see an even mix of alternate and reference alleles at a heterozygous locus in a genome

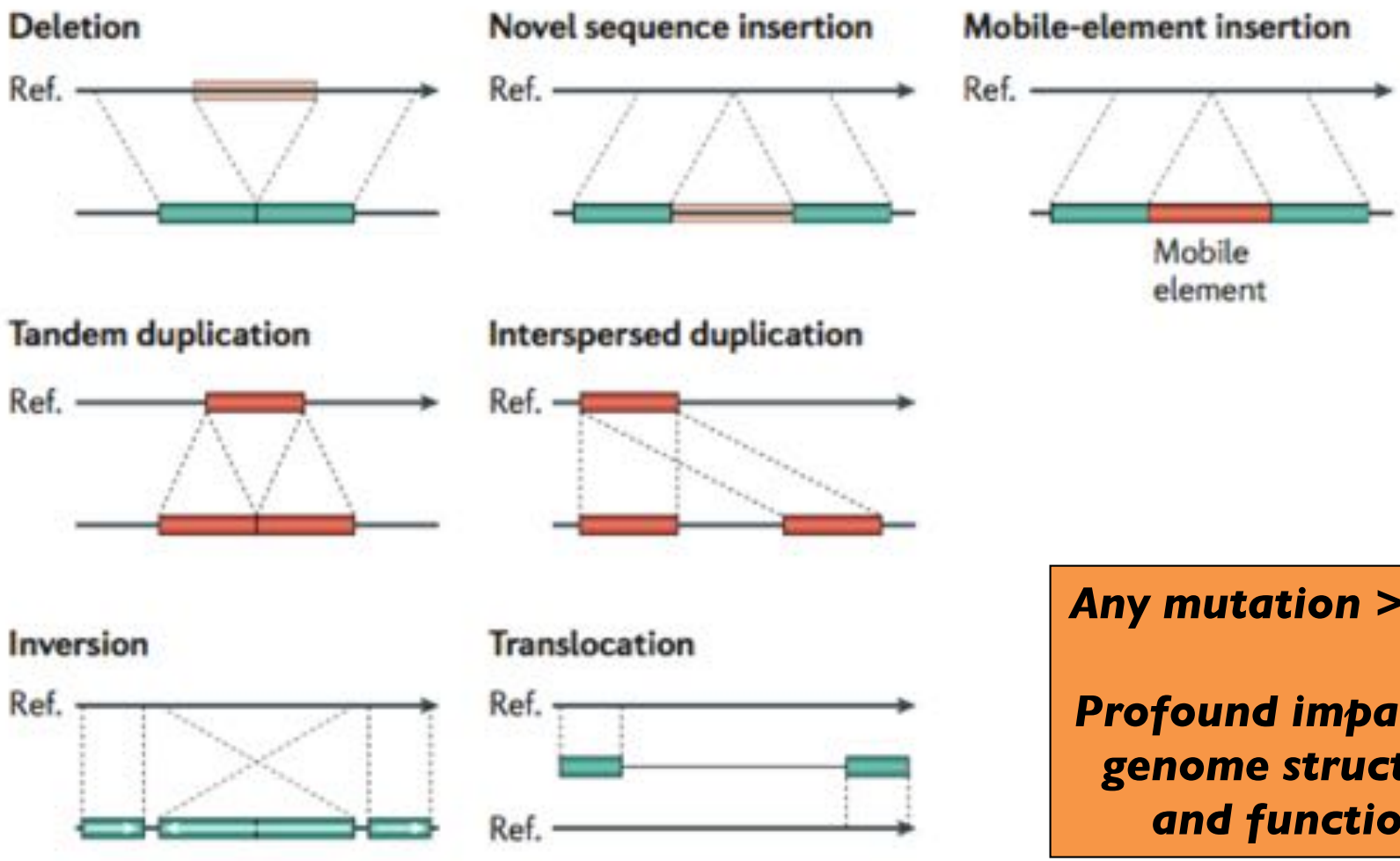


This is why <u>at least</u> a "30X" (30 fold sequence coverage) genome is recommended: it confers sufficient power to distinguish heterozygous alleles and from mere sequencing errors

P(3/30 het) <?> P(3/30 err)

Part 1: What about indels & structural variants

Structural Variations

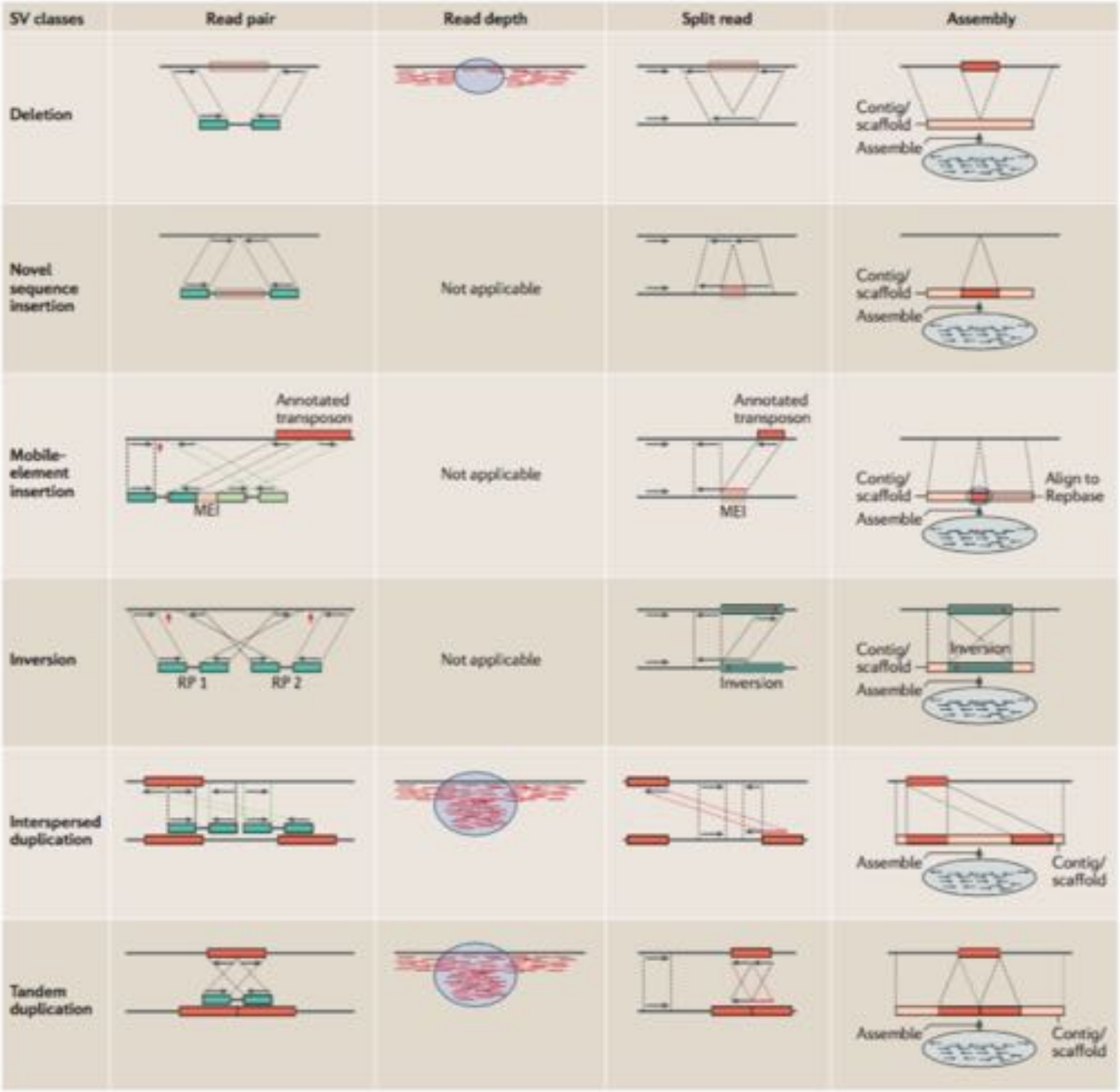


Any mutation >50bp

Profound impact on genome structure and function

Genome structural variation discovery and genotyping
Alkan, C, Coe, BP, Eichler, EE (2011) *Nature Reviews Genetics*. May;12(5):363-76. doi: 10.1038/nrg2958.

Structural Variation Sequence Signatures



Similarity metrics

- Hamming distance
  - Count the number of substitutions to transform one string into another

```
MIKESCHATZ
||X||XXXX|
MICESHATZZ
     5
```

- Edit distance
  - The minimum number of substitutions, insertions, or deletions to transform one string into another

```
MIKESCHAT-Z
||X||X|||X|
MICES-HATZZ
     3
```

Reverse Engineering Edit Distance

D(MIKESCHATZ, MICESHATZZ) = ?

Imagine we already have the optimal alignment of the strings, the last column can only be 1 of 3 options:

```
…M    …I    …D
…Z    …-    …Z
…Z    …Z    …-
```

The optimal alignment of last two columns is then 1 of 9 possibilities

```
…MM  …IM  …DM    …MI  …II  …DI    …MD  …ID  …DD
…TZ  …-Z  …TZ    …Z-  …--  …Z-    …TZ  …-Z  …TZ
…ZZ  …ZZ  …-Z    …TZ  …TZ  …-Z    …Z-  …Z-  …--
```

The optimal alignment of the last three columns is then 1 of 27 possibilities…

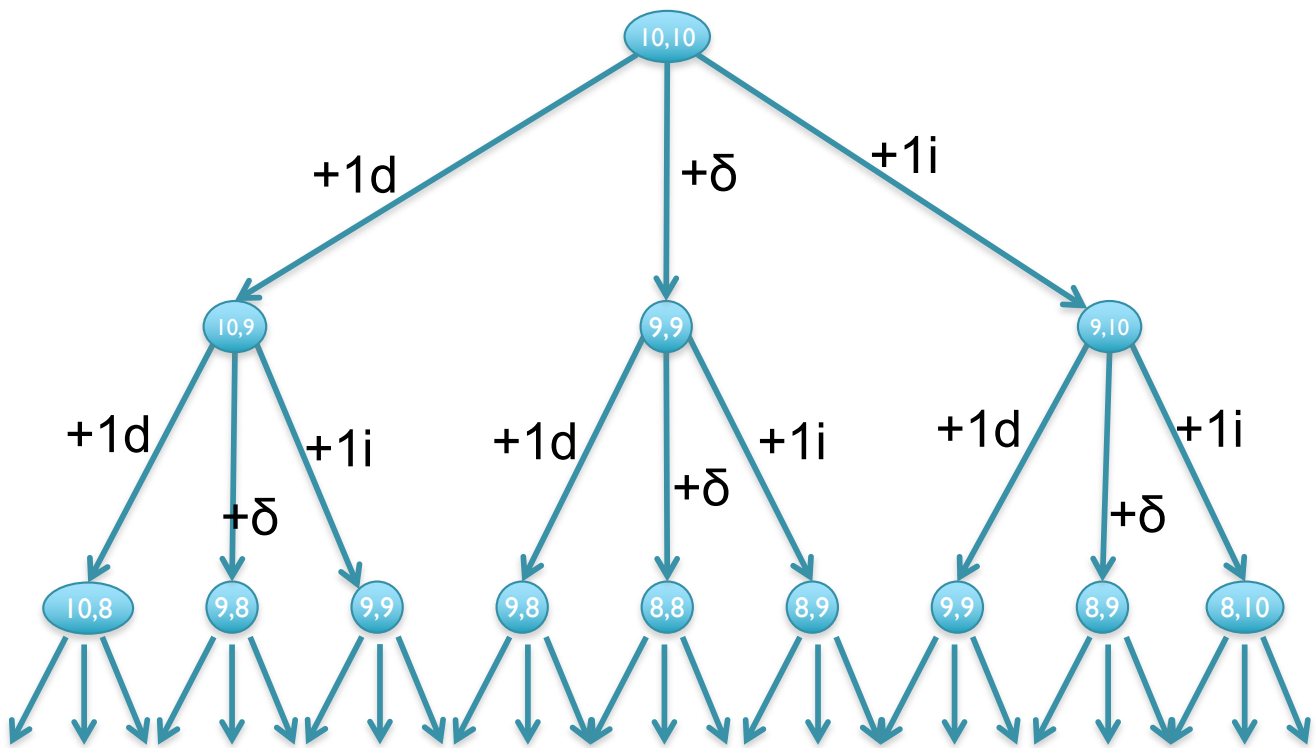
```
…M…    …I…    …D…
…X…    …-…    …X…
…Y…    …Y…    …-…
```

Eventually spell out every possible sequence of {I,M,D}

Recursive solution

- Computation of D is a recursive process.
  - At each step, we only allow matches, substitutions, and indels
  - D(i,j) in terms of D(i',j') for i' ≤ i and j' ≤ j.

D(MIKESCHATZ, MICESHATZZ) = min{D(MIKESCHATZ, MICESHATZ) + I,
D(MIKESCHAT, MICESHATZ) + I,
D(MIKESCHAT, MICESHATZ) +δ(Z, Z)}



[What is the running time?]

Dynamic Programming

- We could code this as a recursive function call...

  ...with an exponential number of function evaluations

- There are only (n+1)x(m+1) pairs i and j
  - We are evaluating D(i,j) multiple times

- Compute D(i,j) bottom up.
  - Start with smallest (i,j) = (1,1).
  - Store the intermediate results in a table.
    - Compute D(i,j) *after* D(i-1,j), D(i,j-1), and D(i-1,j-1)

Recurrence Relation for D

Find the edit distance (minimum number of sub, ins, del operations) to convert one string into another

- Base conditions:

  $D(i,0) = i$, for all $i = 0,\ldots,n$

  $D(0,j) = j$, for all $j = 0,\ldots,m$

- For $i > 0$, $j > 0$:

$$
\begin{aligned}
D(i,j) = \min \{ & \\
& D(i-1,j) + 1, && \text{// align 0 from S, 1 from T} \\
& D(i,j-1) + 1, && \text{// align 1 from S, 0 from T} \\
& D(i-1,j-1) + \delta(S(i),T(j)) && \text{// align 1+1 chars} \\
\} &
\end{aligned}
$$

[Why do we want the min?]

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1										
I	2										
C	3										
E	4										
S	5										
H	6										
A	7										
T	8										
Z	9										
Z	10										

[What does the initialization mean?]

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0									
I	2										
C	3										
E	4										
S	5										
H	6										
A	7										
T	8										
Z	9										
Z	10										

$$D[M,M] = \min\{D[M, \emptyset] + 1,\ D[\emptyset,M]+1,\ D[\emptyset, \emptyset]+\delta(M,M)\}$$

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1								
I	2										
C	3										
E	4										
S	5										
H	6										
A	7										
T	8										
Z	9										
Z	10										

$$D[MI,M] = \min\{D[MI, \emptyset]+1,\ D[M,M]+1,\ D[M, \emptyset]+\delta(I,M)\}$$

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2							
I	2										
C	3										
E	4										
S	5										
H	6										
A	7										
T	8										
Z	9										
Z	10										

D[MIK,M] = min{D[MIK, ∅]+1, D[MI,M]+1, D[MI,]+δ(K,M)}

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2	3						
I	2										
C	3										
E	4										
S	5										
H	6										
A	7										
T	8										
Z	9										
Z	10										

D[MIKE,M] = min{D[MIKE,]+1, D[MIK,M]+1, D[MIK,]+δ(E,M)}

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2	3	4	5	6	7	8	9
I	2										
C	3										
E	4										
S	5										
H	6										
A	7										
T	8										
Z	9										
Z	10										

D[MIKESCHATZ,M] = 9

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2	3	4	5	6	7	8	9
I	2	1									
C	3										
E	4										
S	5										
H	6										
A	7										
T	8										
Z	9										
Z	10										

$$D[M,MI] = \min\{D[M,M]+1,\ D[MI,\ \emptyset]+1,\ D[\emptyset,M]+\delta(M,I)\}$$

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2	3	4	5	6	7	8	9
I	2	1	0								
C	3										
E	4										
S	5										
H	6										
A	7										
T	8										
Z	9										
Z	10										

$$D[MI,MI] = \min\{D[MI,M]+1,\ D[M,\ MI]+1,\ D[M,M]+\delta(I,I)\}$$

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2	3	4	5	6	7	8	9
I	2	1	0	1							
C	3										
E	4										
S	5										
H	6										
A	7										
T	8										
Z	9										
Z	10										

$$D[MIK,MI] = \min\{D[MIK,M]+1,\ D[MI,\ MI]+1,\ D[MI,M]+\delta(K,I)\}$$

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2	3	4	5	6	7	8	9
I	2	1	0	1	2	3	4	5	6	7	8
C	3										
E	4										
S	5										
H	6										
A	7										
T	8										
Z	9										
Z	10										

D[MIKESCHATZ,MI] = 8

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2	3	4	5	6	7	8	9
I	2	1	0	1	2	3	4	5	6	7	8
C	3	2	1	1							
E	4										
S	5										
H	6										
A	7										
T	8										
Z	9										
Z	10										

D[MIK,MIC] = 1

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2	3	4	5	6	7	8	9
I	2	1	0	1	2	3	4	5	6	7	8
C	3	2	1	1	2	3	3	4	5	6	7
E	4										
S	5										
H	6										
A	7										
T	8										
Z	9										
Z	10										

D[MIKESCHATZ,MIC] = 7

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2	3	4	5	6	7	8	9
I	2	1	0	1	2	3	4	5	6	7	8
C	3	2	1	1	2	3	3	4	5	6	7
E	4	3	2	2	1	2	3	4	5	6	7
S	5										
H	6										
A	7										
T	8										
Z	9										
Z	10										

D[MIKESCHATZ,MICE] = 7

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2	3	4	5	6	7	8	9
I	2	1	0	1	2	3	4	5	6	7	8
C	3	2	1	1	2	3	3	4	5	6	7
E	4	3	2	2	1	2	3	4	5	6	7
S	5	4	3	3	2	1	2	3	4	5	6
H	6	5	4	4	3	2	2	2	3	4	5
A	7	6	5	5	4	3	3	3	2	3	4
T	8	7	6	6	5	4	4	4	3	2	3
Z	9	8	7	7	6	5	5	5	4	3	2
Z	10	9	8	8	7	6	6	6	5	4	3

D[MIKESCHATZ,MICESHATZZ] = 3

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2	3	4	5	6	7	8	9
I	2	1	0	1	2	3	4	5	6	7	8
C	3	2	1	1	2	3	3	4	5	6	7
E	4	3	2	2	1	2	3	4	5	6	7
S	5	4	3	3	2	1	2	3	4	5	6
H	6	5	4	4	3	2	2	2	3	4	5
A	7	6	5	5	4	3	3	3	2	3	4
T	8	7	6	6	5	4	4	4	3	2	3
Z	9	8	7	7	6	5	5	5	4	3	2
Z	10	9	8	8	7	6	6	6	5	4	3

D[MIKESCHATZ,MICESHATZZ] = 3

Distance is 3, but how?

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2	3	4	5	6	7	8	9
I	2	1	0	1	2	3	4	5	6	7	8
C	3	2	1	1	2	3	3	4	5	6	7
E	4	3	2	2	1	2	3	4	5	6	7
S	5	4	3	3	2	1	2	3	4	5	6
H	6	5	4	4	3	2	2	2	3	4	5
A	7	6	5	5	4	3	3	3	2	3	4
T	8	7	6	6	5	4	4	4	3	2	3
Z	9	8	7	7	6	5	5	5	4	3	2
Z	10	9	8	8	7	6	6	6	5	4	3

D[MIKESCHATZ,MICESHATZZ] = 3

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2	3	4	5	6	7	8	9
I	2	1	0	1	2	3	4	5	6	7	8
C	3	2	1	1	2	3	3	4	5	6	7
E	4	3	2	2	1	2	3	4	5	6	7
S	5	4	3	3	2	1	2	3	4	5	6
H	6	5	4	4	3	2	2	2	3	4	5
A	7	6	5	5	4	3	3	3	2	3	4
T	8	7	6	6	5	4	4	4	3	2	3
Z	9	8	7	7	6	5	5	5	4	3	2
Z	10	9	8	8	7	6	6	6	5	4	3

Line up chars

D[MIKESCHATZ,MICESHATZZ] = 3

Z
Z

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2	3	4	5	6	7	8	9
I	2	1	0	1	2	3	4	5	6	7	8
C	3	2	1	1	2	3	3	4	5	6	7
E	4	3	2	2	1	2	3	4	5	6	7
S	5	4	3	3	2	1	2	3	4	5	6
H	6	5	4	4	3	2	2	2	3	4	5
A	7	6	5	5	4	3	3	3	2	3	4
T	8	7	6	6	5	4	4	4	3	2	3
Z	9	8	7	7	6	5	5	5	4	3	2
Z	10	9	8	8	7	6	6	6	5	4	3

Gap in top string

D[MIKESCHATZ,MICESHATZZ] = 3

-Z
ZZ

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2	3	4	5	6	7	8	9
I	2	1	0	1	2	3	4	5	6	7	8
C	3	2	1	1	2	3	3	4	5	6	7
E	4	3	2	2	1	2	3	4	5	6	7
S	5	4	3	3	2	1	2	3	4	5	6
H	6	5	4	4	3	2	2	2	3	4	5
A	7	6	5	5	4	3	3	3	2	3	4
T	8	7	6	6	5	4	4	4	3	2	3
Z	9	8	7	7	6	5	5	5	4	3	2
Z	10	9	8	8	7	6	6	6	5	4	3

D[MIKESCHATZ,MICESHATZZ] = 3

```
T-Z
TZZ
```

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2	3	4	5	6	7	8	9
I	2	1	0	1	2	3	4	5	6	7	8
C	3	2	1	1	2	3	3	4	5	6	7
E	4	3	2	2	1	2	3	4	5	6	7
S	5	4	3	3	2	1	2	3	4	5	6
H	6	5	4	4	3	2	2	2	3	4	5
A	7	6	5	5	4	3	3	3	2	3	4
T	8	7	6	6	5	4	4	4	3	2	3
Z	9	8	7	7	6	5	5	5	4	3	2
Z	10	9	8	8	7	6	6	6	5	4	3

D[MIKESCHATZ,MICESHATZZ] = 3

```
AT-Z
ATZZ
```

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2	3	4	5	6	7	8	9
I	2	1	0	1	2	3	4	5	6	7	8
C	3	2	1	1	2	3	3	4	5	6	7
E	4	3	2	2	1	2	3	4	5	6	7
S	5	4	3	3	2	1	2	3	4	5	6
H	6	5	4	4	3	2	2	2	3	4	5
A	7	6	5	5	4	3	3	3	2	3	4
T	8	7	6	6	5	4	4	4	3	2	3
Z	9	8	7	7	6	5	5	5	4	3	2
Z	10	9	8	8	7	6	6	6	5	4	3

D[MIKESCHATZ,MICESHATZZ] = 3

```
HAT-Z
HATZZ
```

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2	3	4	5	6	7	8	9
I	2	1	0	1	2	3	4	5	6	7	8
C	3	2	1	1	2	3	3	4	5	6	7
E	4	3	2	2	1	2	3	4	5	6	7
S	5	4	3	3	2	1	2	3	4	5	6
H	6	5	4	4	3	2	2	2	3	4	5
A	7	6	5	5	4	3	3	3	2	3	4
T	8	7	6	6	5	4	4	4	3	2	3
Z	9	8	7	7	6	5	5	5	4	3	2
Z	10	9	8	8	7	6	6	6	5	4	3

Gap in bottom string

D[MIKESCHATZ,MICESHATZZ] = 3

```
CHAT-Z
-HATZZ
```

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2	3	4	5	6	7	8	9
I	2	1	0	1	2	3	4	5	6	7	8
C	3	2	1	1	2	3	3	4	5	6	7
E	4	3	2	2	1	2	3	4	5	6	7
S	5	4	3	3	2	1	2	3	4	5	6
H	6	5	4	4	3	2	2	2	3	4	5
A	7	6	5	5	4	3	3	3	2	3	4
T	8	7	6	6	5	4	4	4	3	2	3
Z	9	8	7	7	6	5	5	5	4	3	2
Z	10	9	8	8	7	6	6	6	5	4	3

D[MIKESCHATZ,MICESHATZZ] = 3

```
SCHAT-Z
S-HATZZ
```

Dynamic Programming Matrix

		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2	3	4	5	6	7	8	9
I	2	1	0	1	2	3	4	5	6	7	8
C	3	2	1	1	2	3	3	4	5	6	7
E	4	3	2	2	1	2	3	4	5	6	7
S	5	4	3	3	2	1	2	3	4	5	6
H	6	5	4	4	3	2	2	2	3	4	5
A	7	6	5	5	4	3	3	3	2	3	4
T	8	7	6	6	5	4	4	4	3	2	3
Z	9	8	7	7	6	5	5	5	4	3	2
Z	10	9	8	8	7	6	6	6	5	4	3

Just line up mis-matches

D[MIKESCHATZ,MICESHATZZ] = 3

```
KESCHAT-Z
CES-HATZZ
```

Dynamic Programming Matrix

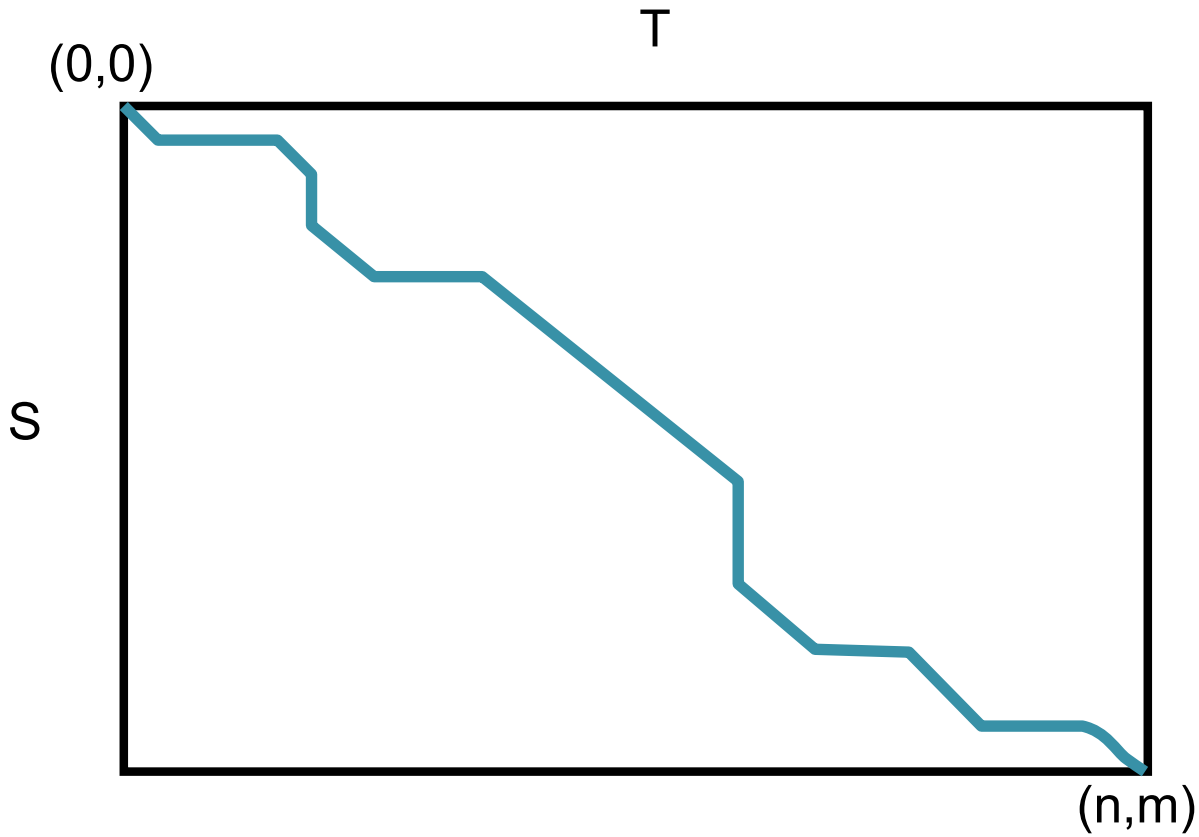
		M	I	K	E	S	C	H	A	T	Z
	0	1	2	3	4	5	6	7	8	9	10
M	1	0	1	2	3	4	5	6	7	8	9
I	2	1	0	1	2	3	4	5	6	7	8
C	3	2	1	1	2	3	3	4	5	6	7
E	4	3	2	2	1	2	3	4	5	6	7
S	5	4	3	3	2	1	2	3	4	5	6
H	6	5	4	4	3	2	2	2	3	4	5
A	7	6	5	5	4	3	3	3	2	3	4
T	8	7	6	6	5	4	4	4	3	2	3
Z	9	8	7	7	6	5	5	5	4	3	2
Z	10	9	8	8	7	6	6	6	5	4	3

D[MIKESCHATZ,MICESHATZZ] = 3

```
MIKESCHAT-Z
MICES-HATZZ
```

Hooray!

Global Alignment Schematic



- A high quality alignment will stay close to the diagonal
  - If we are only interested in high quality alignments, we can skip filling in cells that can't possibly lead to a high quality alignment
  - Find the global alignment with at most edit distance d: O(2dn)

Sequence Similarity

- Similarity score generalizes edit distance
  - Certain mutations are much more likely than others
    - Hydrophilic -> Hydrophillic much more likely than Hydrophillic -> Hydrophobic
  - BLOSSUM62
    - Empirically measure substitution rates among proteins that are 62% identical
    - Positive score: more likely than chance, Negative score: less likely

	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	4																			
Arg	-1	5																		
Asn	-2	0	6																	
Asp	-2	-2	1	6																
Cys	0	-3	-3	-3	9															
Gln	-1	1	0	0	-3	5														
Glu	-1	0	0	2	-4	2	5													
Gly	0	-2	0	-1	-3	-2	-2	6												
His	-2	0	1	-1	-3	0	0	-2	8											
Ile	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
Leu	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
Lys	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5								
Met	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
Phe	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
Pro	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
Ser	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
Thr	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
Trp	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Tyr	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
Val	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

Edit Distance and Global Similarity

$$
\begin{aligned}
D(i,j) = \min \{ & \\
& D(i-1,j) + 1, \\
& D(i,j-1) + 1, \\
& D(i-1,j-1) + \delta(S(i),T(j)) \\
\} &
\end{aligned}
$$

s = 4x4 or 20x20 scoring matrix

$$
\begin{aligned}
S(i,j) = \max \{ & \\
& S(i-1,j) - 1, \\
& S(i,j-1) - 1, \\
& S(i-1,j-1) + s(S(i),T(j)) \\
\} &
\end{aligned}
$$

[Why max?]

Local vs. Global Alignment (cont' d)

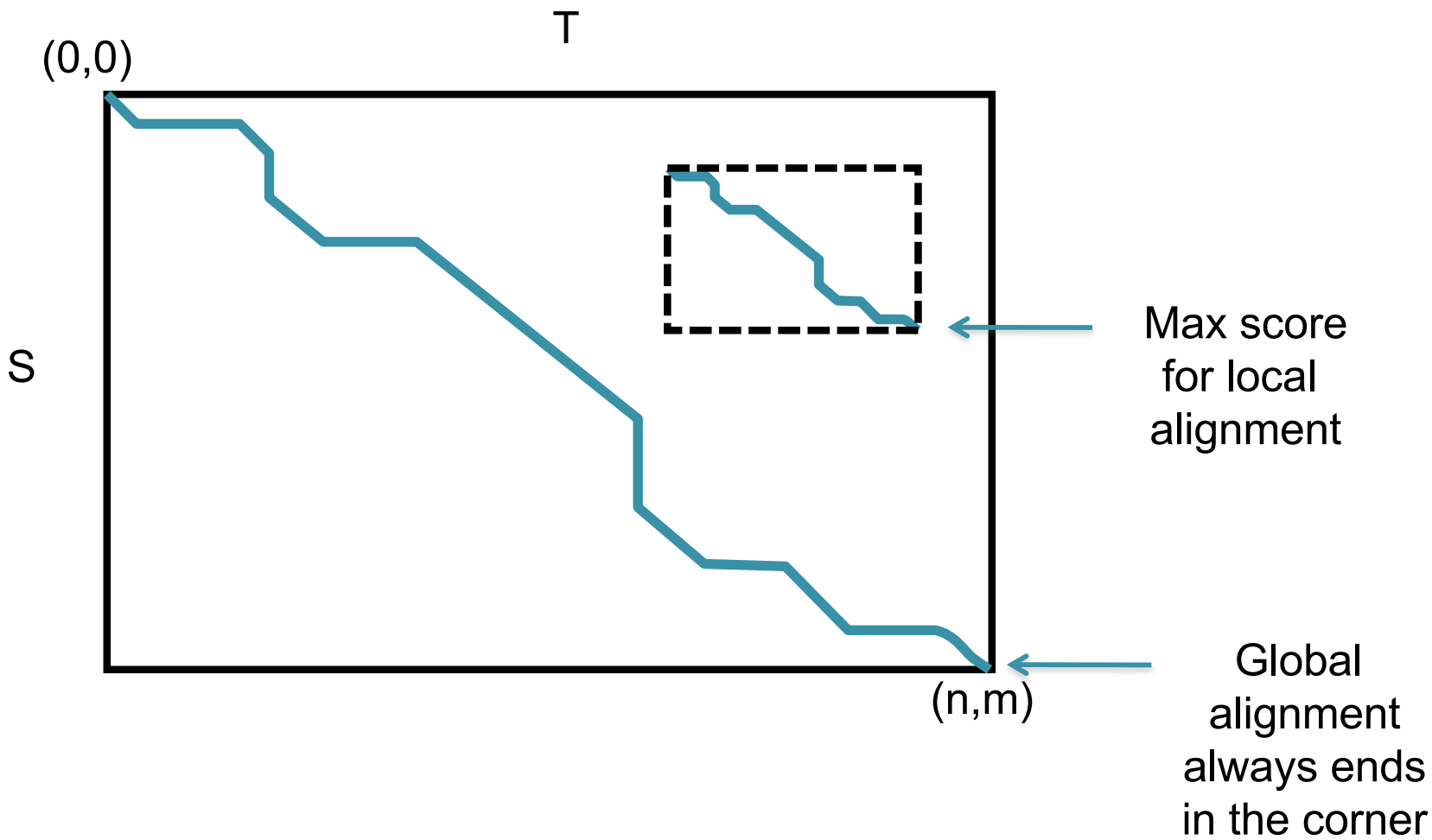
- Global Alignment

```
--T—-CC-C-AGT—-TATGT-CAGGGGACACG—A-GCATGCAGA-GAC
  |  || |  ||  | | | |||    || | | |  | ||||   |
AATTGCCGCC-GTCGT-T-TTCAG----CA-GTTATG—T-CAGAT--C
```

- Local Alignment—better alignment to find conserved segment

```
                tccCAGTTATGTCAGggacacgagcatgcagagac
                   ||||||||||||
aattgccgccgtcgttttcagCAGTTATGTCAGatc
```

Global vs Local Alignment Schematic



Nathan Edwards

The Local Alignment Recurrence

- The largest value of $s_{i,j}$ over the whole edit graph is the score of the best local alignment.

- The recurrence:

$$s_{i,j} = max \begin{cases} 0 \\ s_{i-1,j-1} + \delta(v_i, w_j) \\ s_{i-1,j} + \delta(v_i, -) \\ s_{i,j-1} + \delta(-, w_j) \end{cases}$$

Power of ZERO: there is only this change from the original recurrence of a Global Alignment - since there is only one "free ride" edge entering into every vertex

Affine Gap Penalties

- In nature, a series of *k* indels often come as a single event rather than a series of *k* single nucleotide events:

```
ATA___GC        ATAG_GC
ATATTGC         AT_GTGC
```

This is more likely.

Normal scoring would give the same score for both alignments

This is less likely.

Accounting for Gaps

- *Gaps*- contiguous sequence of spaces in one of the rows

- Score for a gap of length x is: $-(\rho + \sigma x)$
  where $\rho > 0$ is the gap opening penalty
  ρ will be large relative to gap extension penalty σ

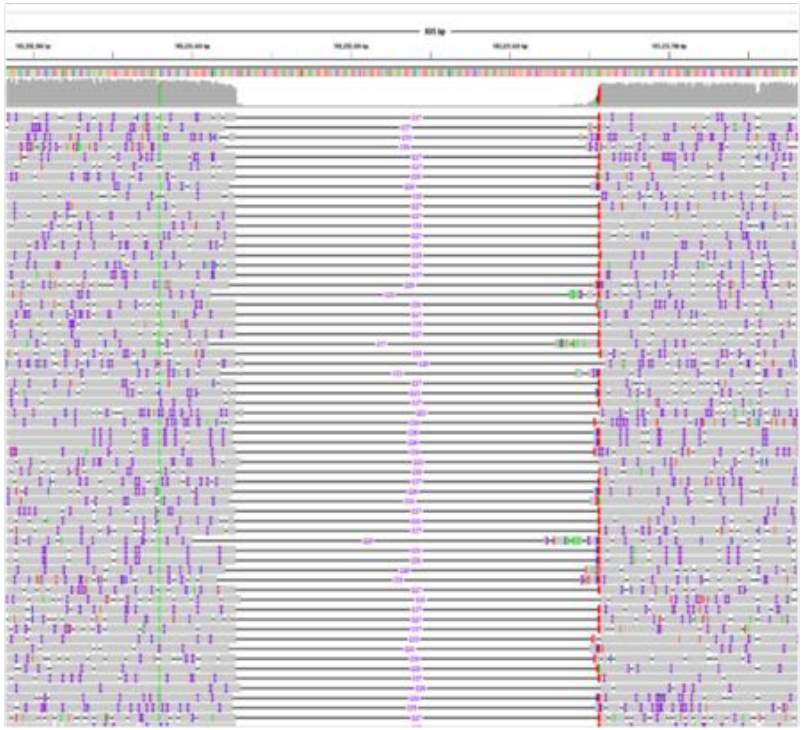
  – Gap of length 1: $-(\rho + \sigma) = -6$
  – Gap of length 2: $-(\rho + \sigma 2) = -7$
  – Gap of length 3: $-(\rho + \sigma 3) = -8$

- Smith-Waterman-Gotoh incorporates affine gap penalties without increasing the running time O(mn)
  – Uses parallel matrices for considering gap openings and gap extensions at every step

NGMLR + Sniffles

BWA-MEM:

NGMLR:



NGMLR: Convex gap penalty to balance frequent small sequencing errors with larger SVs
Sniffles: Scan within and between split reads to accurately find SVs (Ins, Del, Dup, Inv, Trans)
Mendelian concordance >95%, experimental validation also very high

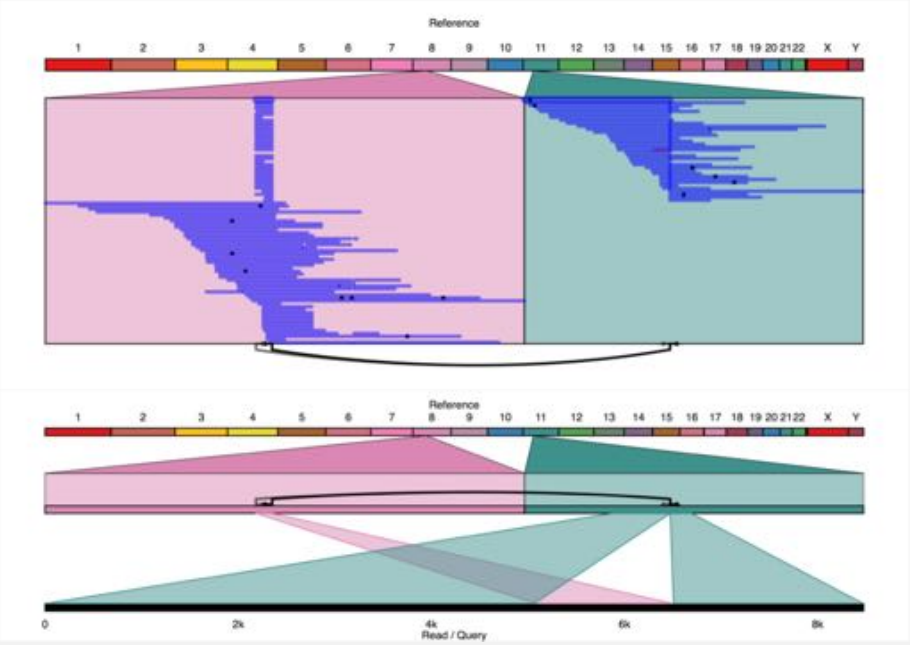
Accurate detection of complex structural variations using single molecule sequencing
Sedlazeck, Rescheneder et al (2018) *Nature Methods. In Press*

SVs in a typical healthy human

Sniffles calls

	All SVs (50bp+)	Large SVs (10kbp+)
Deletions	7,389	164
Duplications	1,284	139
Insertions	8,382	4
Inversions	229	116
Translocations	170	170
All	***17,454***	***593***

Translocation in Ribbon

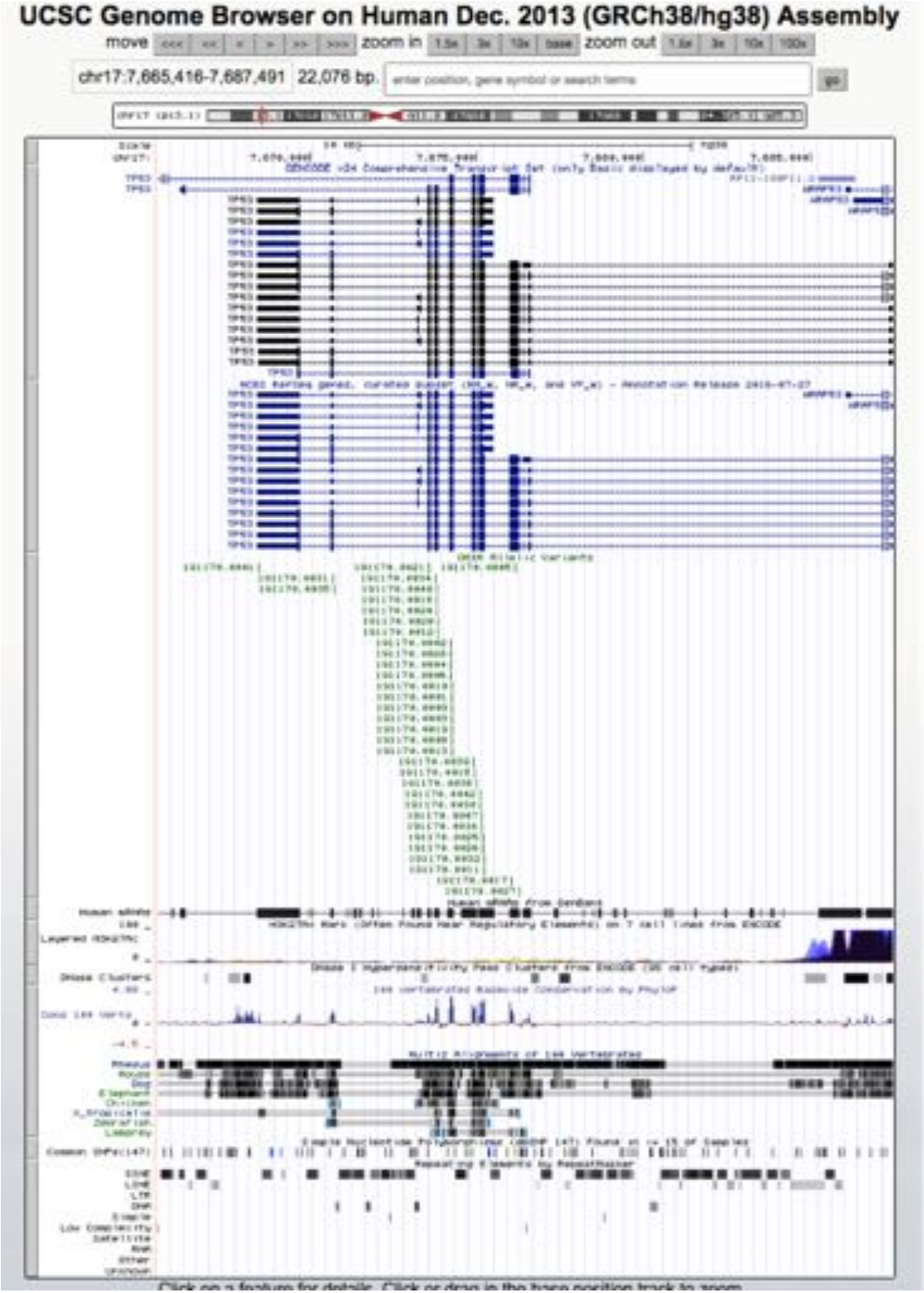


Ribbon: Visualizing complex genome alignments and structural variation
Nattestad et al. (2016) *bioRxiv* doi: http://dx.doi.org/10.1101/082123

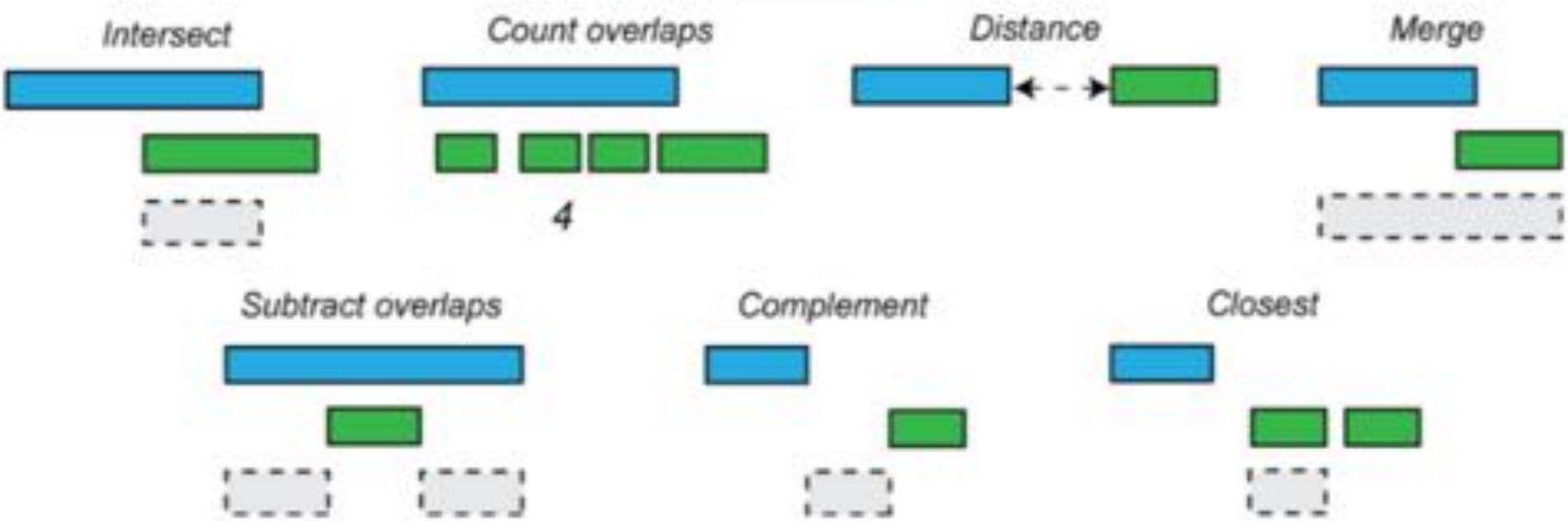
Part 2:
Genome Arithmetic

What are genome intervals?

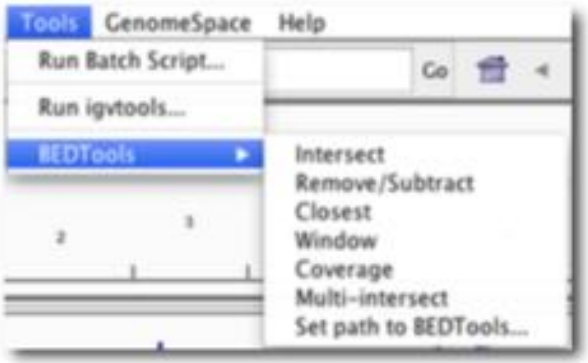
- Genetic variation:
  - SNPs: 1bp
  - Indels: 1-50bp
  - SVs: >50bp
- Genes:
  - exons, introns, UTRs, promoters
- Conservation
- Transposons
- Origins of replication
- TF binding sites
- CpG islands
- Segmental duplications
- Sequence alignments
- Chromatin annotations
- Gene expression data
- …
- ***Your own observations and data: put them into context!***



BEDTools to the rescue!



Getting & Using BEDTools

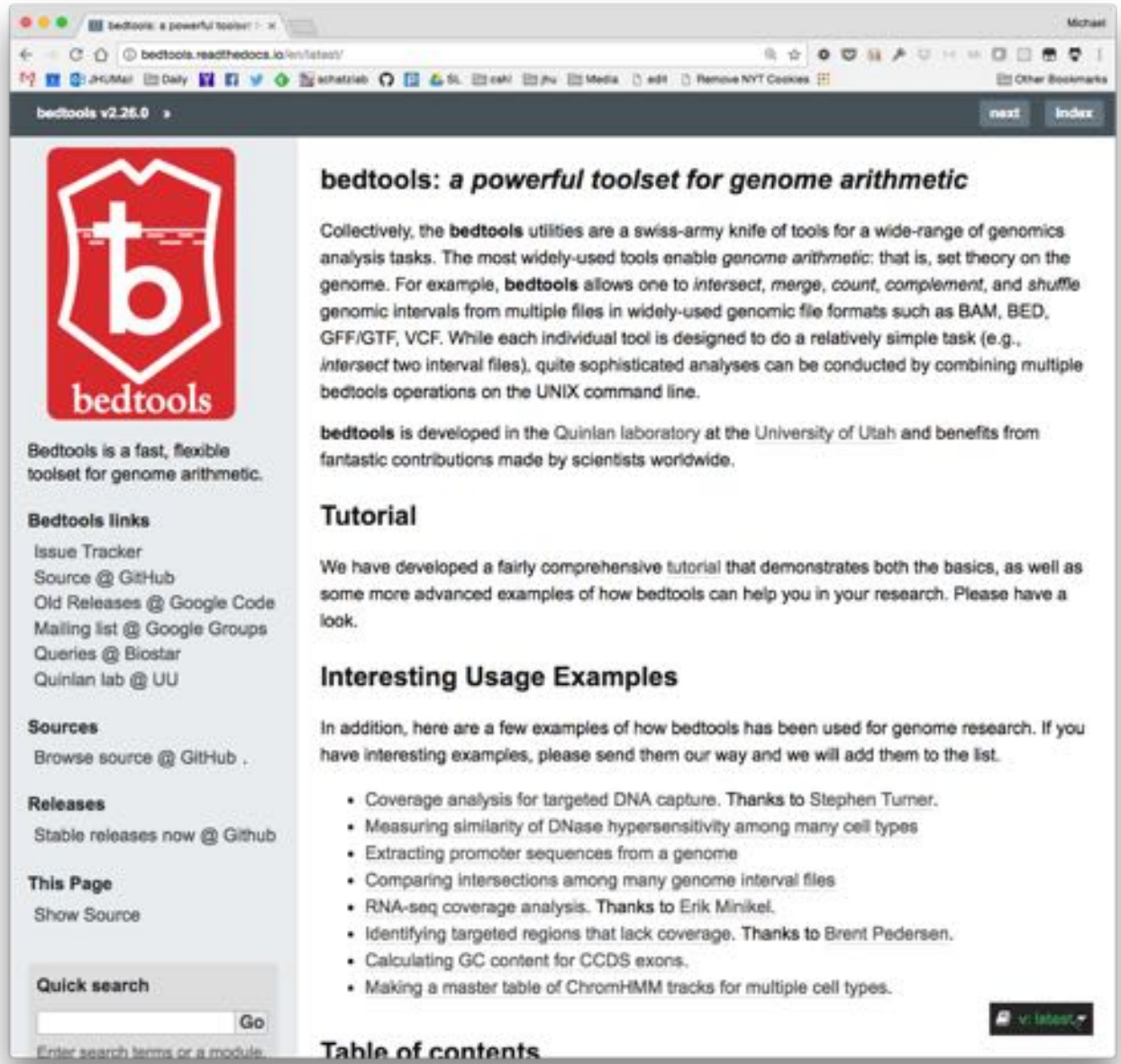


Integrated into **IGV**

BEDTools

- Intersect BAM alignments with intervals in another files
- Count intervals in one file overlapping intervals in another file
- Create a histogram of genome coverage
- Create a BedGraph of genome coverage
- Convert from BAM to BED
- Merge BedGraph files
- Intersect multiple sorted BED files

In **Galaxy** Toolshed



Extensive Documentation and Examples

Genomic Coordinates

What are coordinates of "TAC" in GATTACA?

1-based coordinates

- Base 4 through 6: [4,6] "closed"
- Base 4 through 7: [4,7) "half-open"
- 3 bases starting at base 4: [4, +3]

```
GATTACA
1234567
```

0-based coordinates

- Position 3 through 5: [3,5] "closed"
- Position 3 through 6: [3,6) "half-open"
- 3 bases starting at position 3: [3, +3]

```
GATTACA
0123456
```

Genomic Conventions

1-based coordinates

- BLAST/MUMmer alignments
- Ensembl Genome Browser
- SAM, VCF, GFF and Wiggle

```
GATTACA
1234567
```

0-based coordinates

- BAM, BCFv2, BED, and PSL
- UCSC Genome Browser
- C/C++, Perl, Python, Java

```
GATTACA
0123456
```

Always double check the manual!
You will get this wrong someday ☹

BED Format

BED (Browser Extensible Data) format provides a flexible way to define intervals.

The first three required BED fields are:

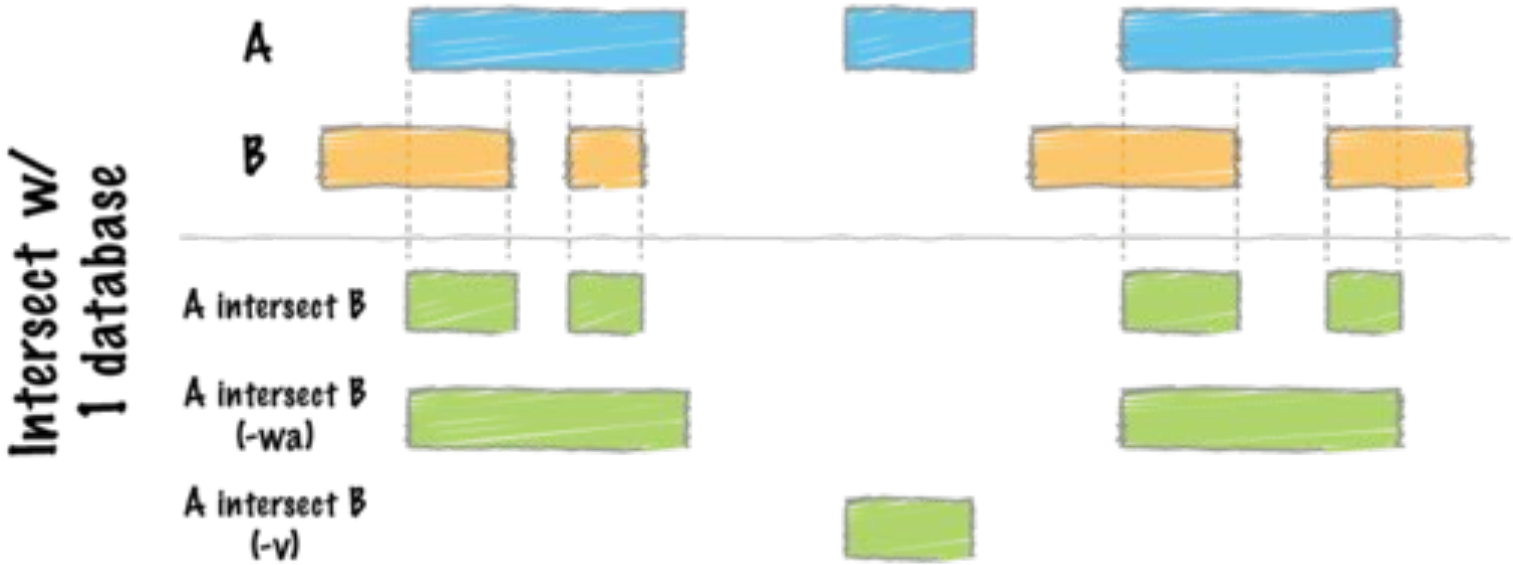
1. chrom — The name of the chromosome (e.g. chr3, chrY, chr2_random) or scaffold (e.g. scaffold10671).
2. chromStart — The starting position of the feature in the chromosome or scaffold. The first base in a sequence is numbered 0.
3. chromEnd — The ending position of the feature in the chromosome or scaffold.
   The chromEnd base is not included in the display of the feature. For example, the first 100 bases of a chromosome are defined as chromStart=0, chromEnd=100, and span the bases numbered 0-99.

The 9 additional optional BED fields are:

1. name - Defines the name of the BED line
2. score - A score between 0 and 1000
3. strand - Defines the strand. Either "." (=no strand) or "+" or "-".
4. thickStart - The starting position at which the feature is drawn thickly
5. thickEnd - The ending position at which the feature is drawn thickly (for example the stop codon in gene displays).
6. itemRgb - An RGB value of the form R,G,B (e.g. 255,0,0).
7. blockCount - The number of blocks (exons) in the BED line.
8. blockSizes - A comma-separated list of the block sizes. The number of items in this list should correspond to blockCount.
9. blockStarts - A comma-separated list of block starts. All of the blockStart positions should be calculated relative to chromStart. The number of items in this list should correspond to blockCount.

```
## genes.bed has: chrom, txStart, txEnd, name, num_exons, and strand
$ head -n4 genes.bed
chr1    134212701    134230065    Nuak2    8     +
chr1    134212701    134230065    Nuak2    7     +
chr1    33510655     33726603     Prim2,   14    -
chr1    25124320     25886552     Bai3,    31    -
```

BEDTools Intersect



What exons are hit by SVs?

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

$ cat B.bed
chr1  15  20

$ bedtools intersect -a A.bed -b B.bed -wa
chr1  10   20
```

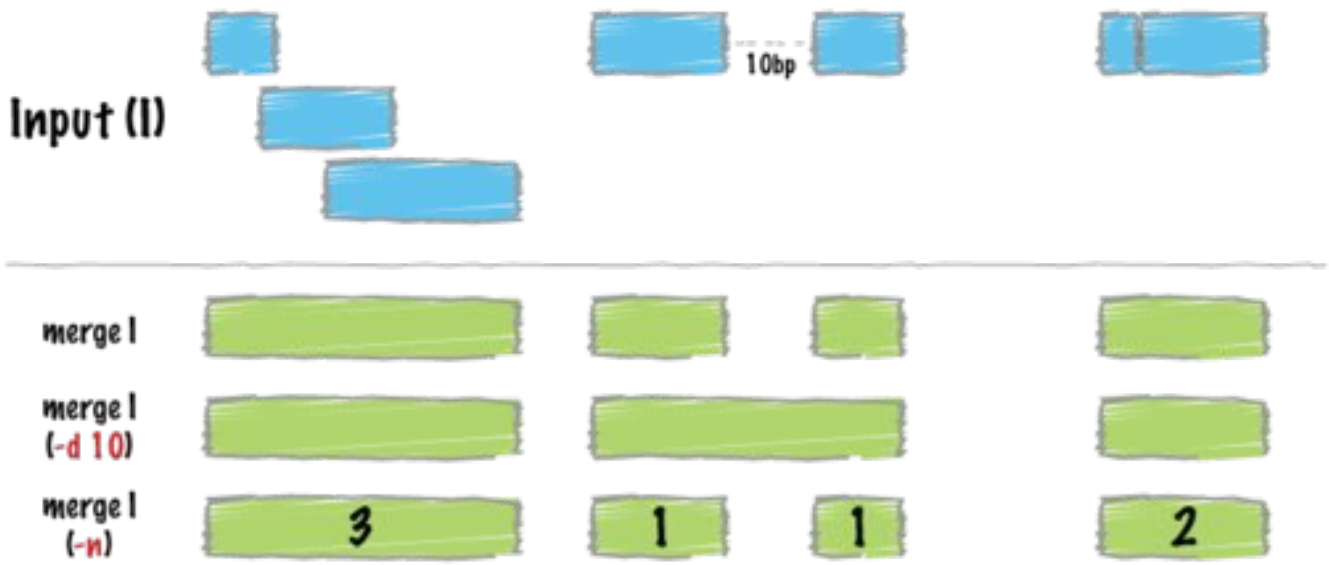
What parts of exons are hit by SVs?

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

$ cat B.bed
chr1  15   20

$ bedtools intersect -a A.bed -b B.bed
chr1  15   20
```

BEDTools Merge



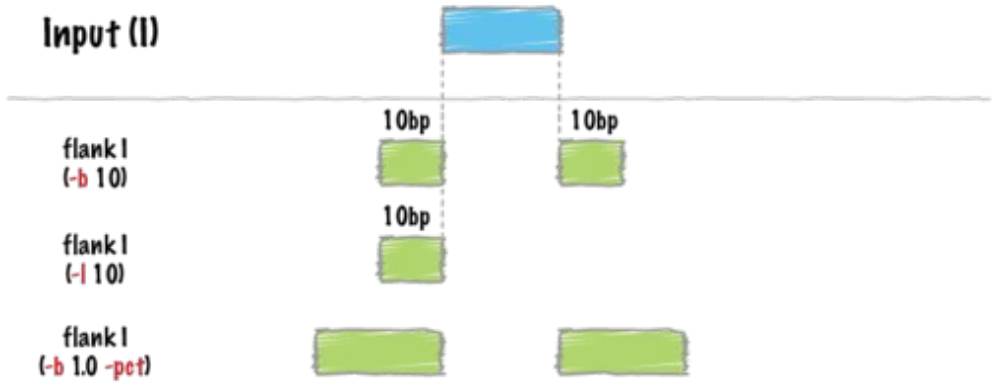
What parts of the genome are exonic?

```
bedtools merge -i exons.bed | head -n 20
chr1    11873   12227
chr1    12612   12721
chr1    13220   14829
chr1    14969   15038
chr1    15795   15947
chr1    16606   16765
chr1    16857   17055
```

Note input must be sorted!

```
sort -k1,1 -k2,2n foo.bed > foo.sort.bed
```

BEDTools Flank & getfasta



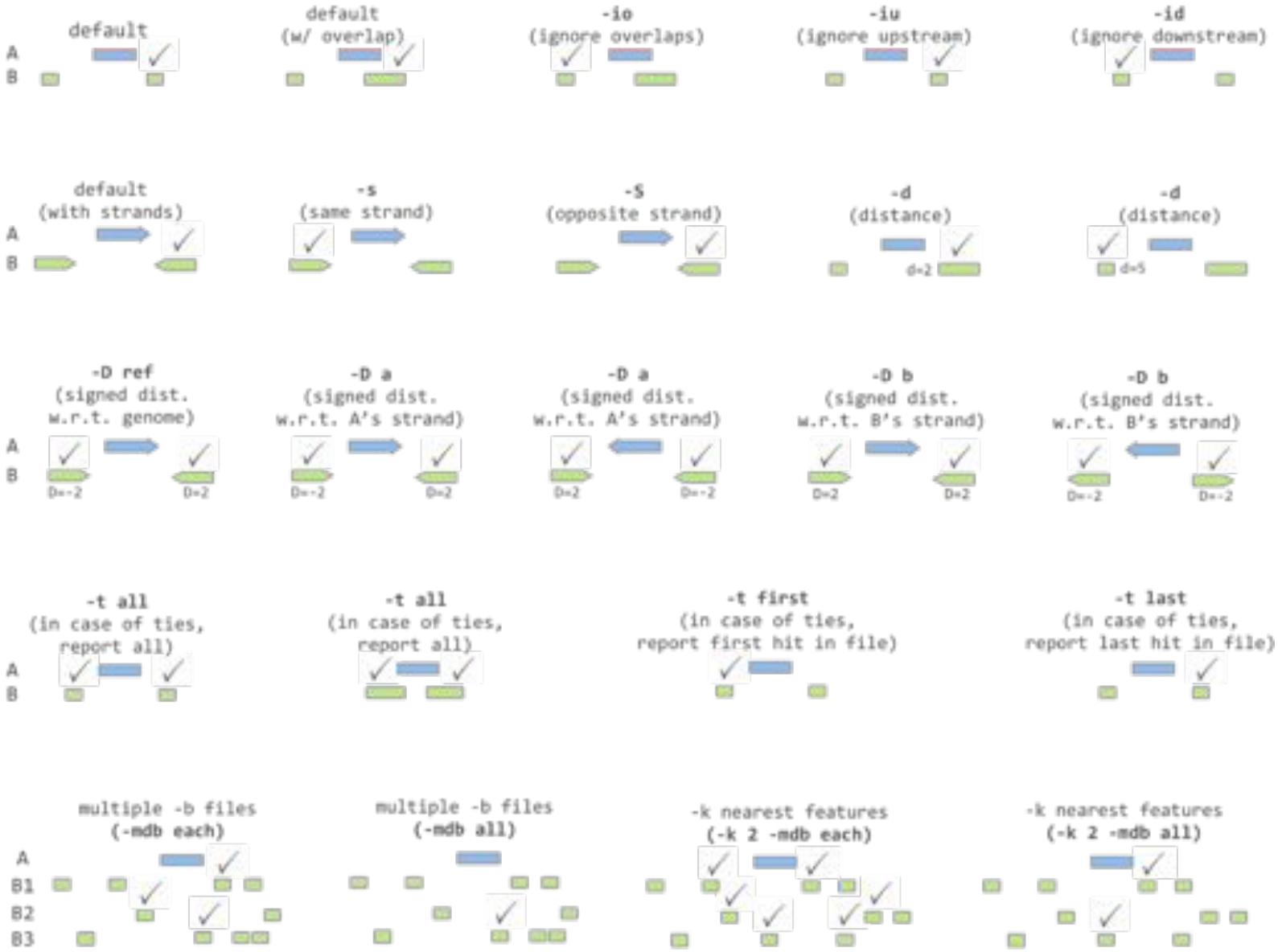

```
## genes.bed has: chrom, txStart, txEnd, name, num_exons, and strand
$ head -n4 genes.bed
chr1    134212701    134230065    Nuak2    8     +
chr1    134212701    134230065    Nuak2    7     +
chr1    33510655     33726603     Prim2,   14    -
chr1    25124320     25886552     Bai3,    31    -

## Identify promoter regions (2kbp upstream)
$ bedtools flank -i genes.bed -g mm9.chromsizes -l 2000 -r 0 -s > genes.2kb.promoters.bed

## Show promoter coordinates
$ head genes.2kb.promoters.bed
chr1    134210701    134212701    Nuak2     8    +
chr1    134210701    134212701    Nuak2     7    +
chr1    33726603     33728603     Prim2,    14    -
chr1    25886552     25888552     Bai3,     31    -

## Extract the sequences
$ bedtools getfasta -fi mm9.fa -bed genes.2kb.promoters.bed -fo genes.2kb.promoters.bed.fa
```

BEDTools Closest



What is the gene closest to this SNP or this enhancer?

BEDTools commands

```
annotate
bamtobed
bamtofastq
bed12tobed6
bedpetobam
bedtobam
closest
cluster
complement
coverage
expand
flank
fisher
genomecov
getfasta
groupby
groupby
igv
intersect
jaccard
links
makewindows
map
maskfasta
merge
multicov
multiinter
nuc
overlap
pairtobed
pairtopair
random
reldist
shift
shuffle
slop
sort
subtract
tag
unionbedg
window
```

http://bedtools.readthedocs.io/en/latest/content/bedtools-suite.html