MSU Next Generation Sequencing – Summer 2012

ChIP-Seq Studies

István Albert

Bioinformatics Consulting Center

Penn State

What's New! Hot off the press!

BMC Bioinformatics

IMPACT FACTOR 3.03

Search BMC Bioinformatics

Home | Articles | Authors | Reviewers | About this journal | My BMC Bioinformatics

Software

Open acce

BioWord: a sequence manipulation suite for Microsoft Word

Laura J Anzaldi, Daniel Muñoz-Fernández and **Ivan Erill**

For all author emails, please log on.

BMC Bioinformatics 2012, **13**:124 doi:10.1186/1471-2105-13-124

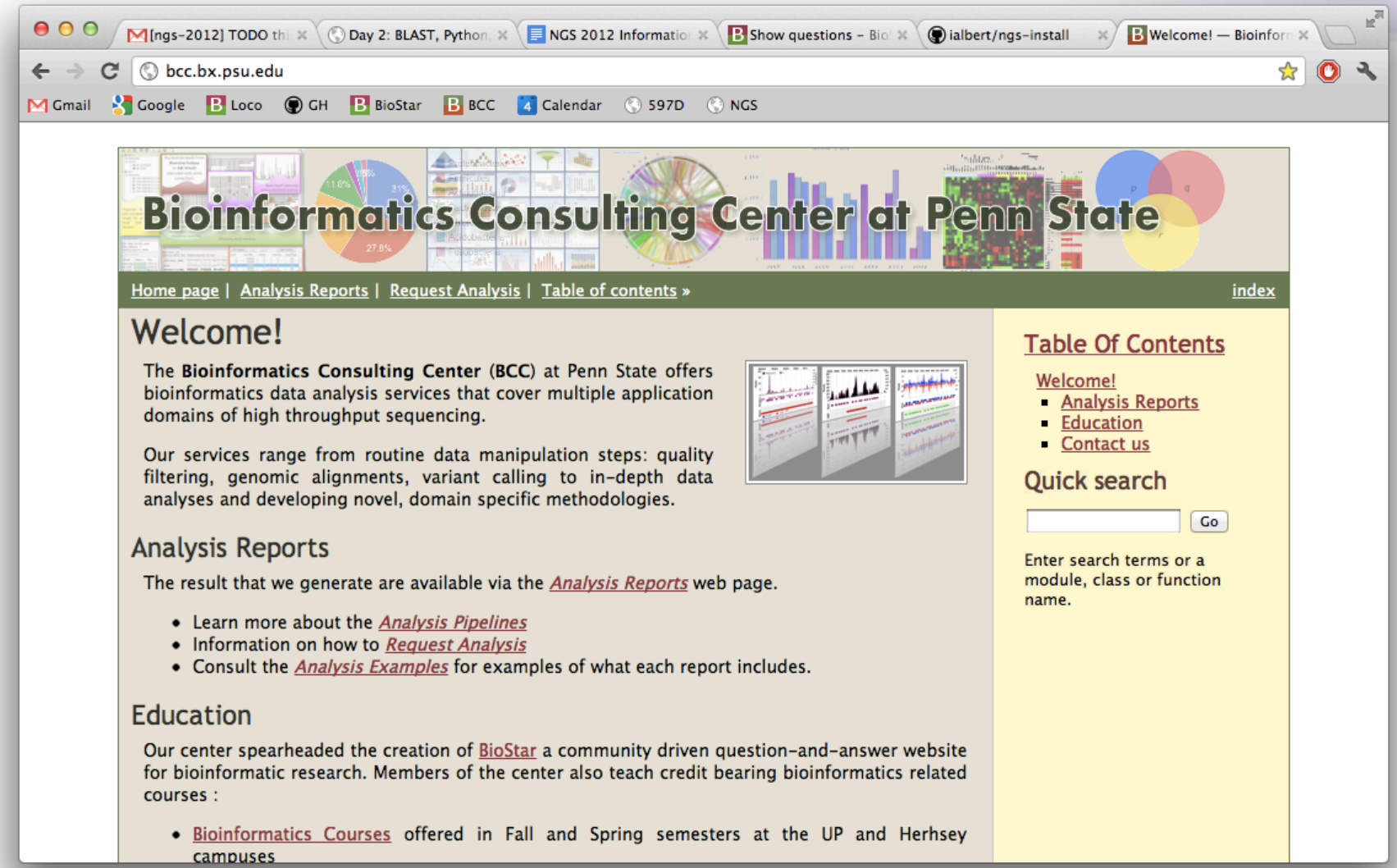
Published: 7 June 2012

Abstract (provisional)

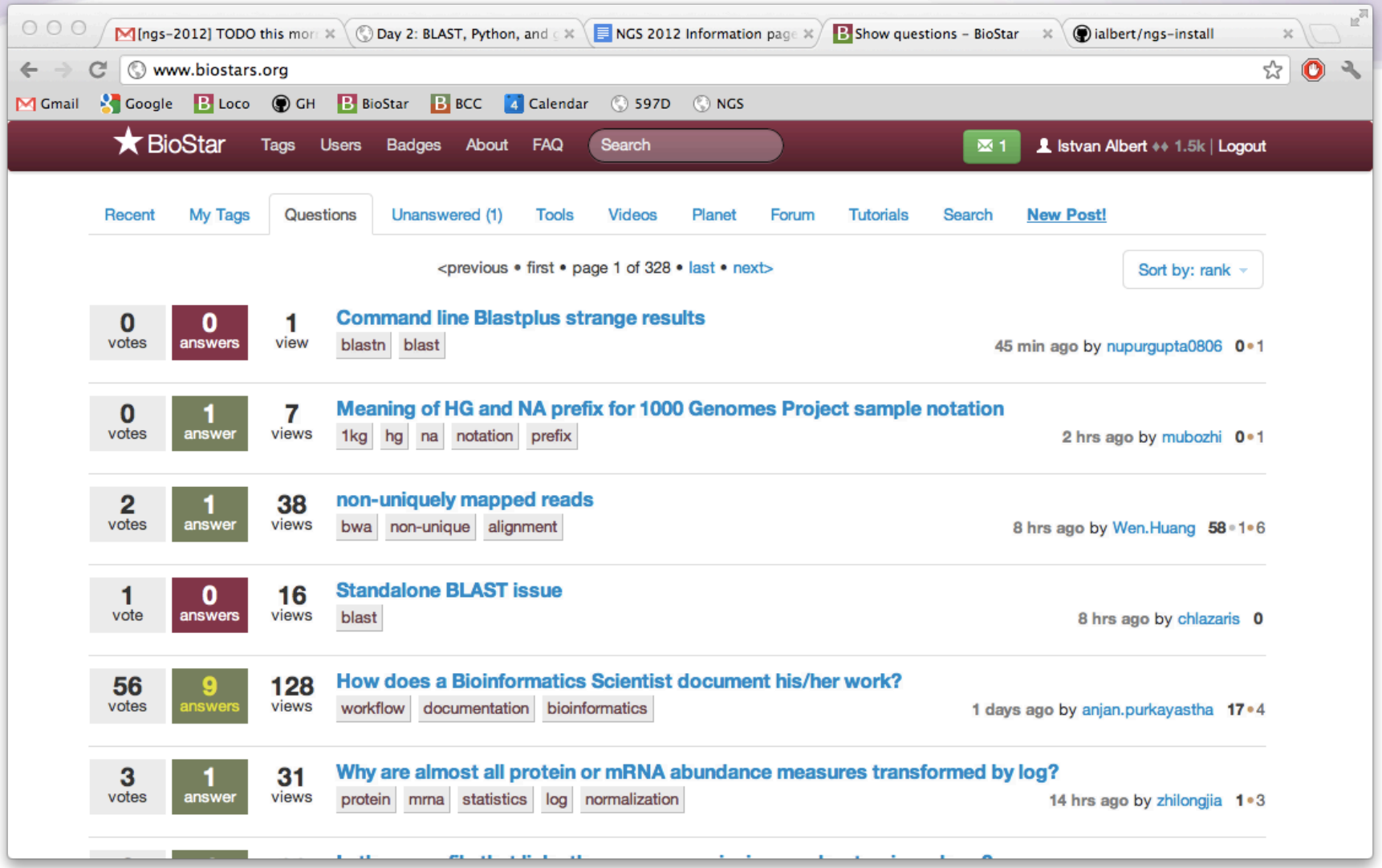
What's New! Hot off the press!

PSU: Bioinformatics Consulting Center

http://bcc.bx.psu.edu

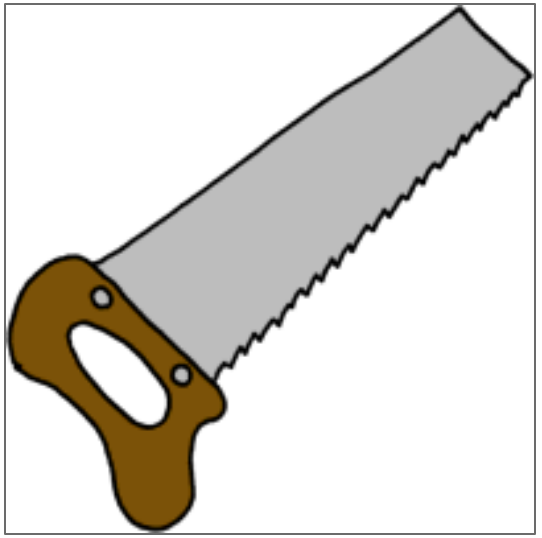


BioStar: **http://www.biostars.org**



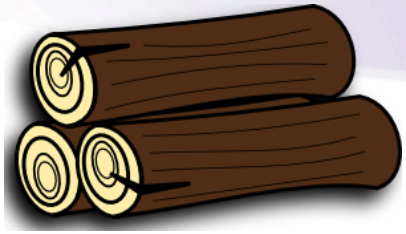
Bioinformatics tools

VS



Bioinformatician's Toolbelt

- Lumber → data



- hammer, saw, pliers → (mapper, samtools, bedtools)

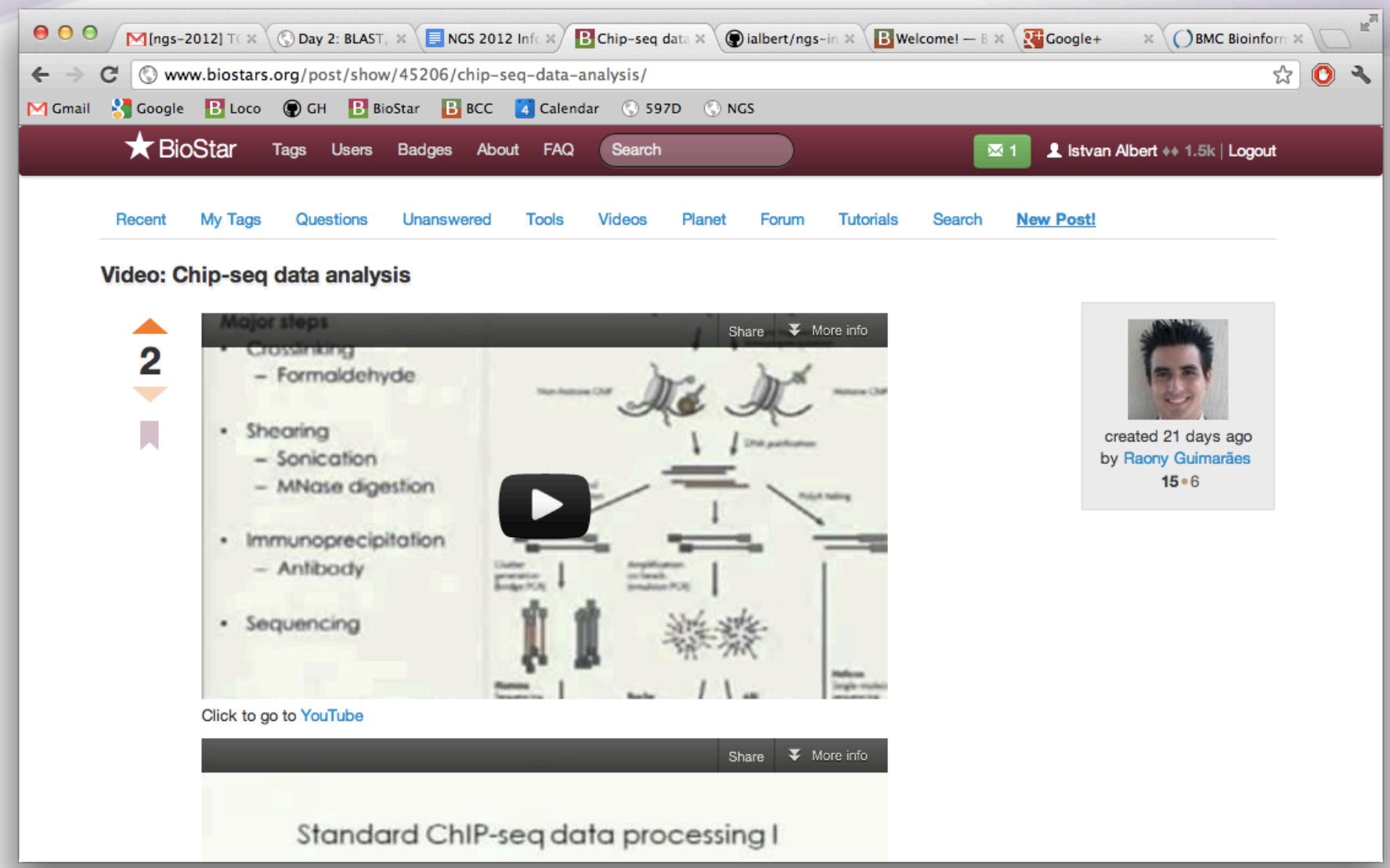
- Glue, nails, screws, ductape → scripts → shell, python, perl, awk

You can do just about anything with these!

You don't need specialized tools for most tasks!

Common sense and a good understanding of what each does

Great ChIP-Seq video series see the **Videos** tab



Todays' goal

- Understand the principles of ChIP-Seq analysis
- Run through an entire mapping procedure – "view from the trenches"
- We provide scripts to download the data
- Characterize the results by their density around other genomic features
- Compare results by different peak callers

Next big Topic: **Protein - DNA** interactions **ChIP-Chip** and **ChIP-Seq** studies

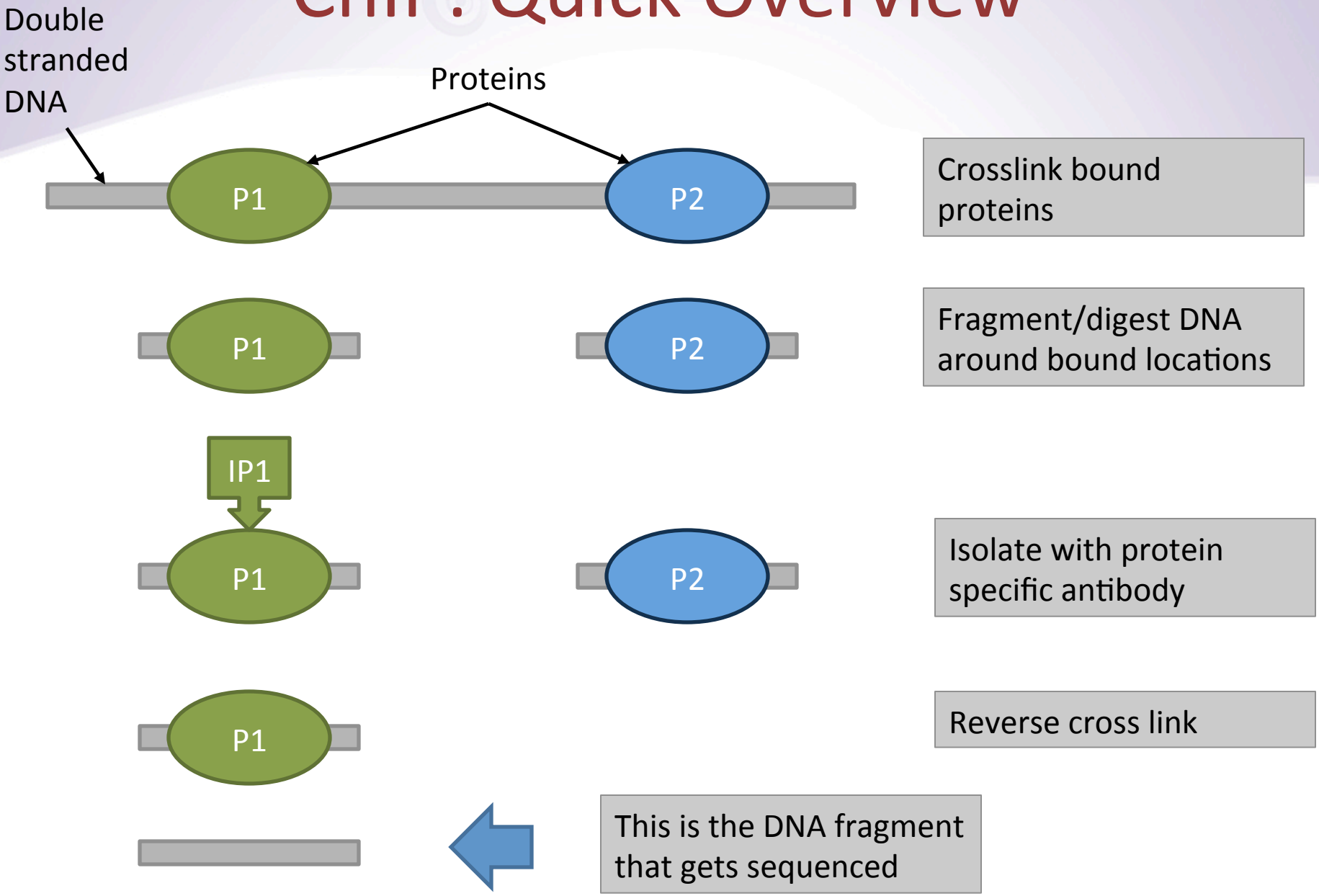
- **ChIP** → Chromatin Immuno-Precipitation (used during sample preparation)
- **Chip** → microarray technology to detect bound genomic locations
- **Seq** → high throughput sequencing to detect bound genomic locations

Chromatin Immuno-Precipitation

It is a well know methodology to detect:

- transcription factor binding
- polymerase binding
- chromatin structure and modifications
- etc…

ChIP: Quick Overview



Sample origins

Understanding the sample preparation is essential for analysis

- **WGS** whole genome sequencing (shotgun) → random DNA fragments covering the entire genome
- **Chip-Seq** → DNA fragments covering bound locations in the genome

(we will keep adding to this list as we cover new techniques)

The ChIP output

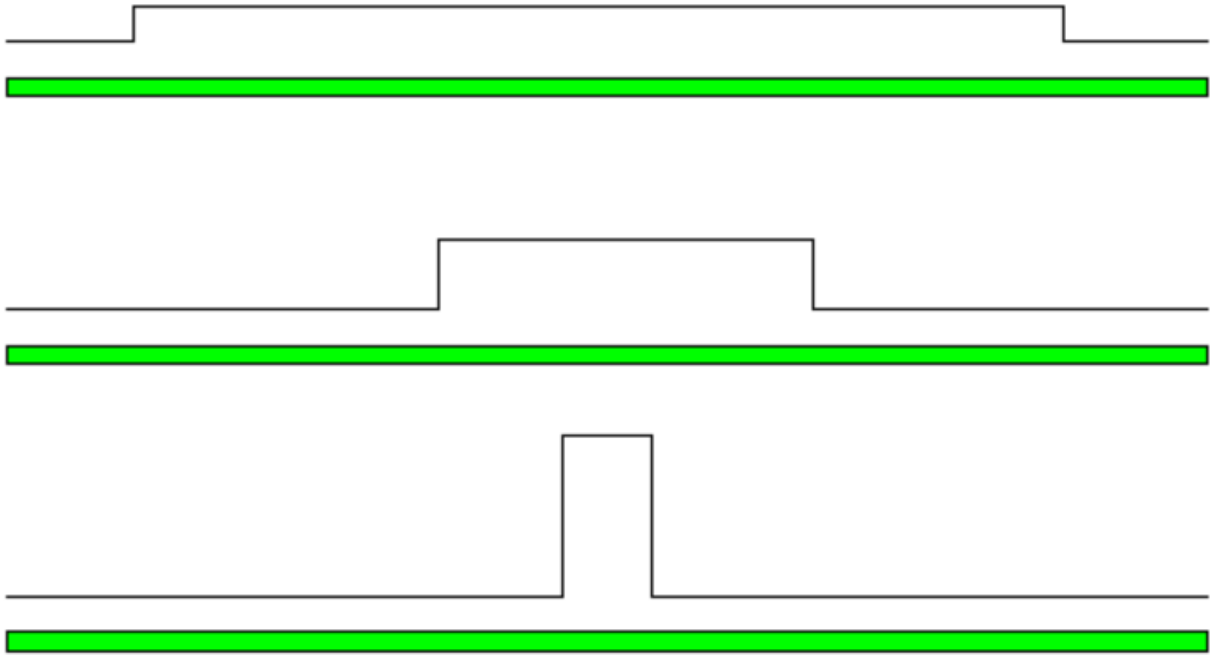
- a DNA sample **enriched*** for fragments associated with the events under study
- We are measuring an ensemble of cells that may be in different states
- Coverage depends on the number of sites, efficiency of the IP (precipitation) step.
- Accuracy depends on **fragmentation** strategy: sonication, MNASE digestion, lambda-nuclease digestion

Note: Lots of **other DNA fragments** can make it through!

Chip-Seq →High throughput sequencing

- Fragments are sequenced
- Aligned against genome (Bowtie is a good choice for Chip-Seq)
  (Note: today at 4pm we have a presentation in the Berg auditorium by the PI directing the Bowtie development!)
- The output is in the form of the intervals (start/end) where each read matches the genome

Type of events of interest



Long-range

(e.g., histone modifications)

Mid-range

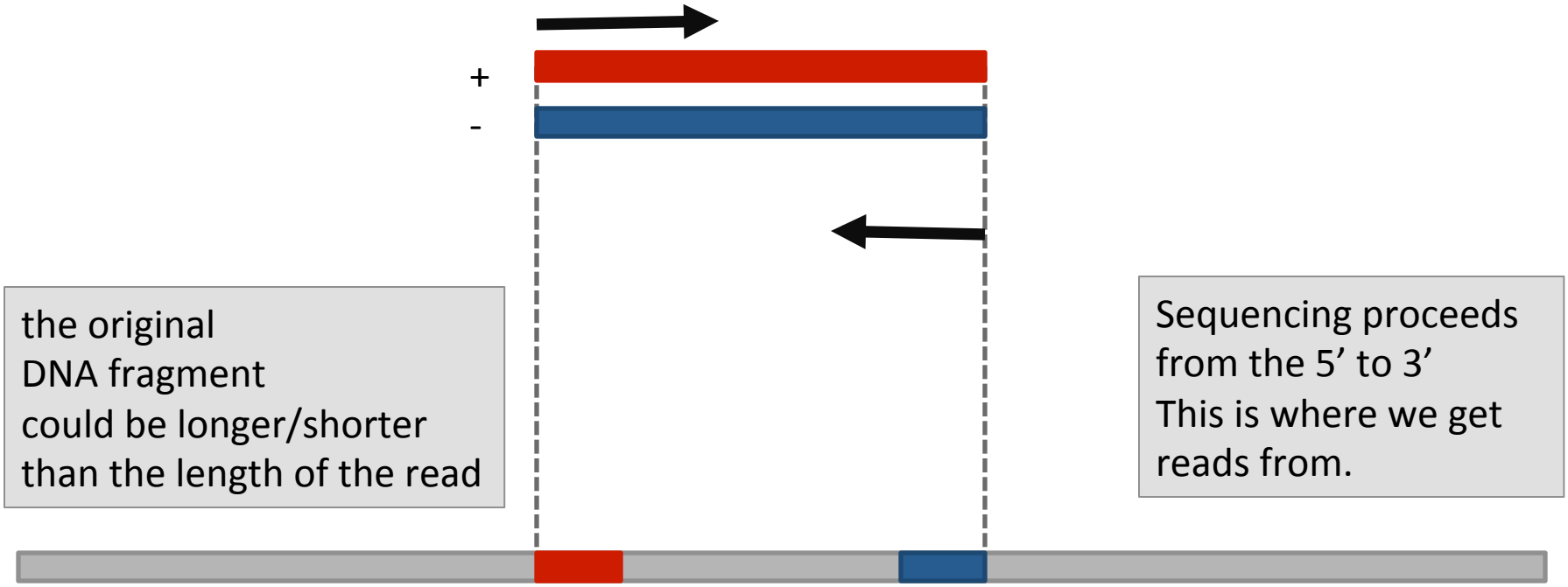
(e.g. polymerase binding)

Punctate

(e.g. TF binding)

Single End sequencing

Original two stranded DNA fragment

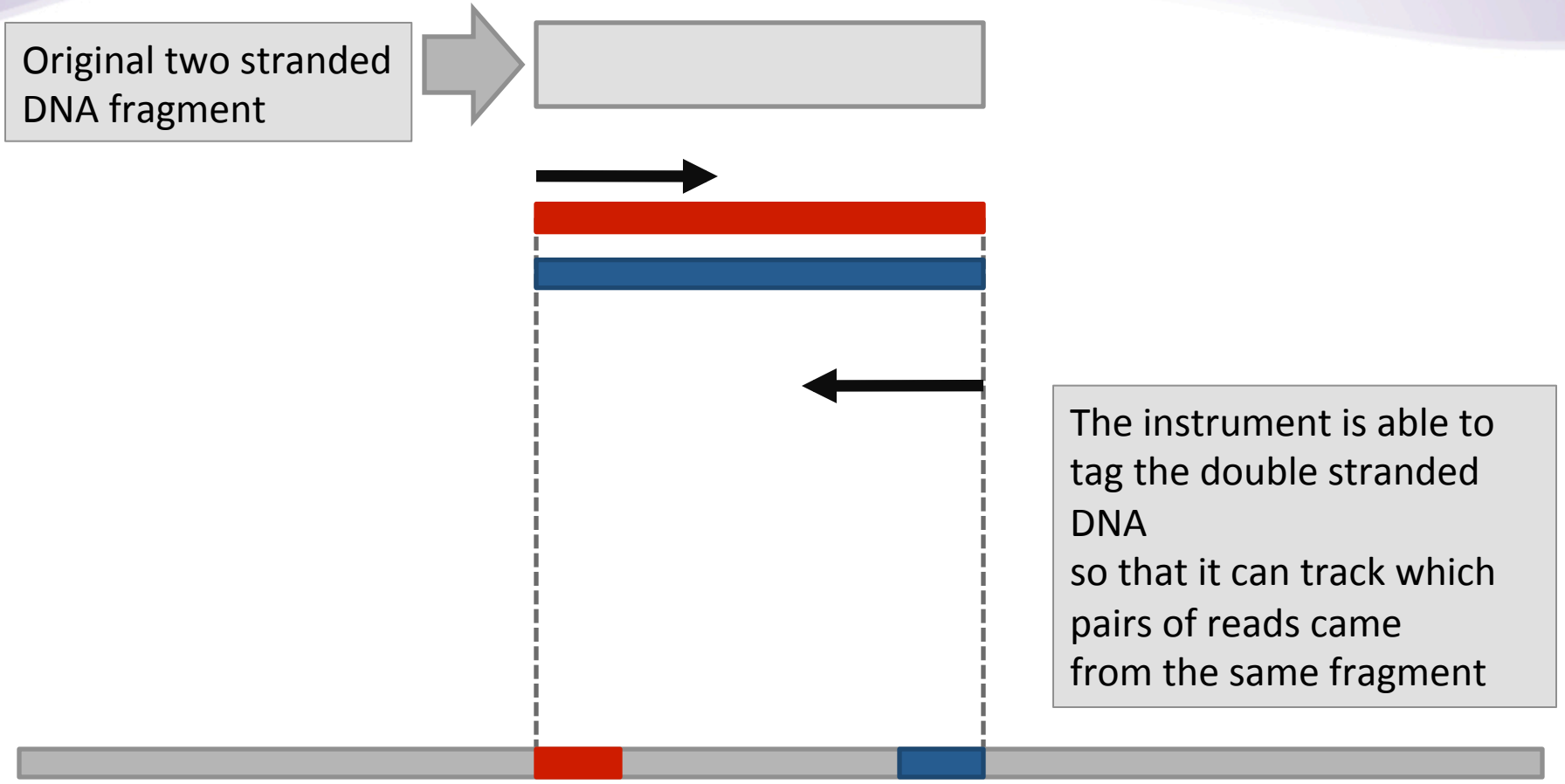


the original DNA fragment could be longer/shorter than the length of the read

Sequencing proceeds from the 5' to 3' This is where we get reads from.

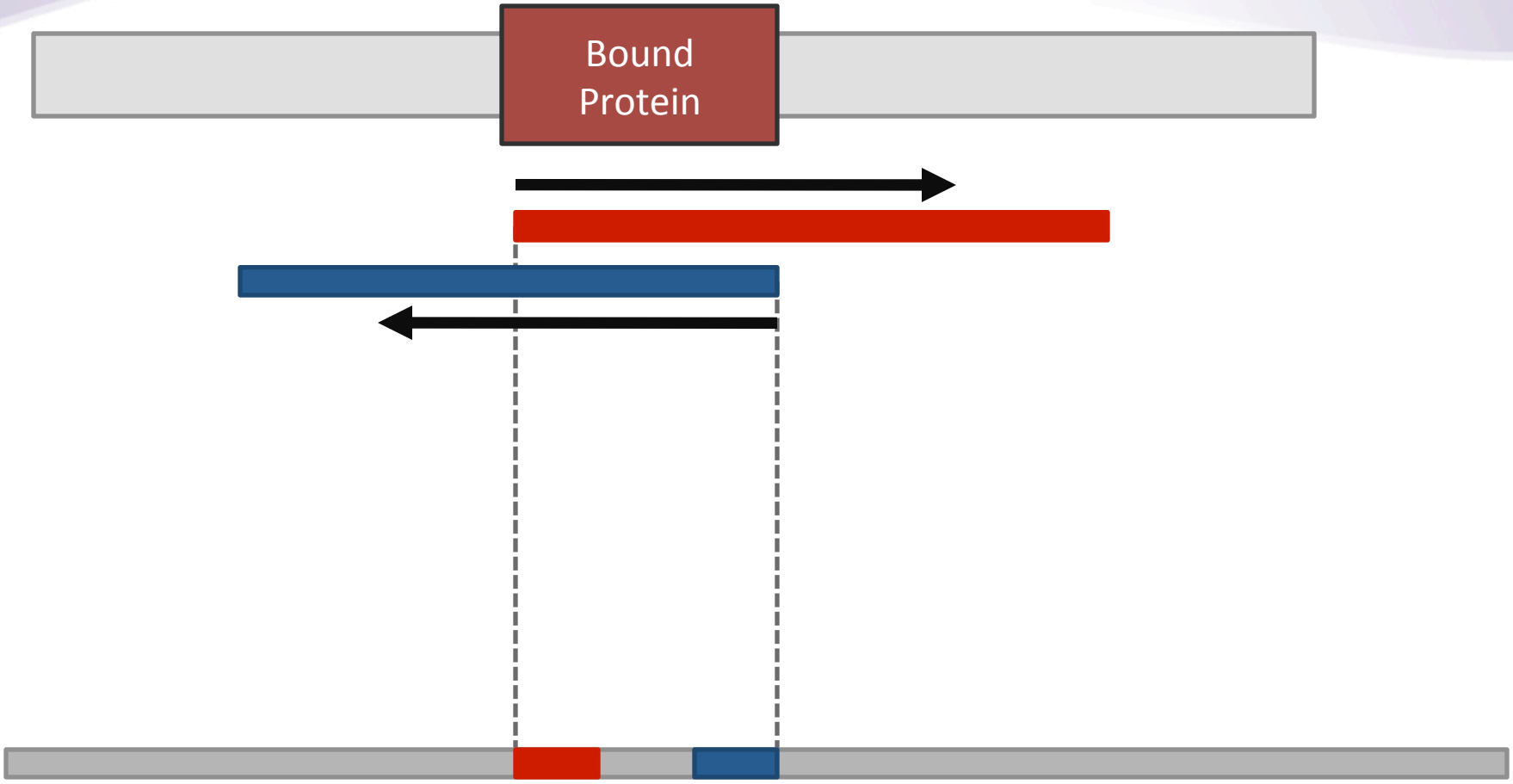
Single end read: we don't know which left border is paired up with which right border.

Paired End sequencing



Data will comes in pairs of right-left border
We **know the width of each fragment**!

Other techniques: Chip-Exo



The reads are longer than the bound location – allows one to measure short-punctate locations. Other sample preparation techniques are always been developed. It is essential to know the sample preparation.

Alignment to the genome

After alignment we get genomic intervals for each read, minimally:

chrom, start, end, strand

The fragment 5' end locations will then

the **start** coordinate for the + strand

the **end** coordinate for the – strand

Other considerations

- For single end sequencing **each fragment** may correspond to **0**, **1** or **2** reads. If it has **two reads** we don't know which two formed the fragment

- For paired and sequencing each fragment corresponds to **0, 1** or **2** reads. (1 no mate). We know which **two reads** correspond to one another.

Peak Calling

- Process of finding the locations enriched due to events of interest

We will need to define

- **Peak Region** - contiguous set of basepairs that belong to a peak
- **Enrichment Level** - read-based measure of supporting evidence

Peak calling: base pair level measurements

- Number of fragments overlapping that position (need to go from reads to fragments)

- Number of reads (fragment ends, midpoints) reported at that position (possibly, taking strandedness into account)

  Variation: **kernel-smoothed read density**. This is closer to overlap approach.

  (we will cover the peak calling in next lecture)

Data/script location

Collected all into a single installer, get the repository:

```
git clone git://github.com/ialbert/kbs-install.git
```

Then switch to the kbs directory and un the install-chipseq.sh script.

```
cd kbs-install
sh install-chipseq.sh
```

Running the install script

Look inside the script to see what it does

more install-chipseq.sh

Then run it

sh install-chipseq.sh

Downloads a series of files and programs that will be compiled on your system. It will create the following directory structure:

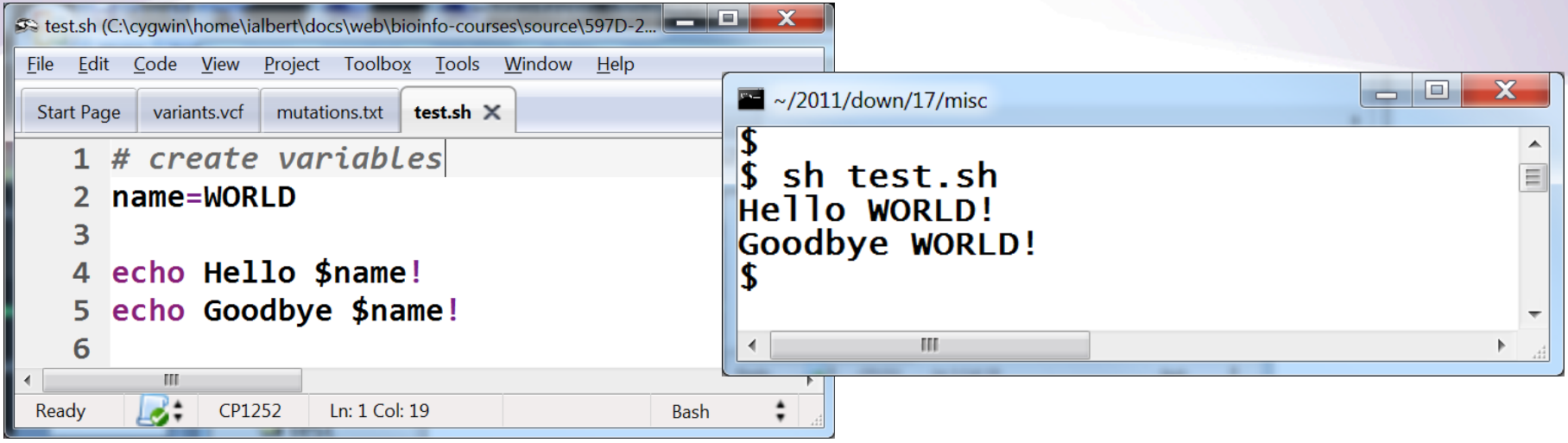
~/work/data
~/work/src
~/work/refs

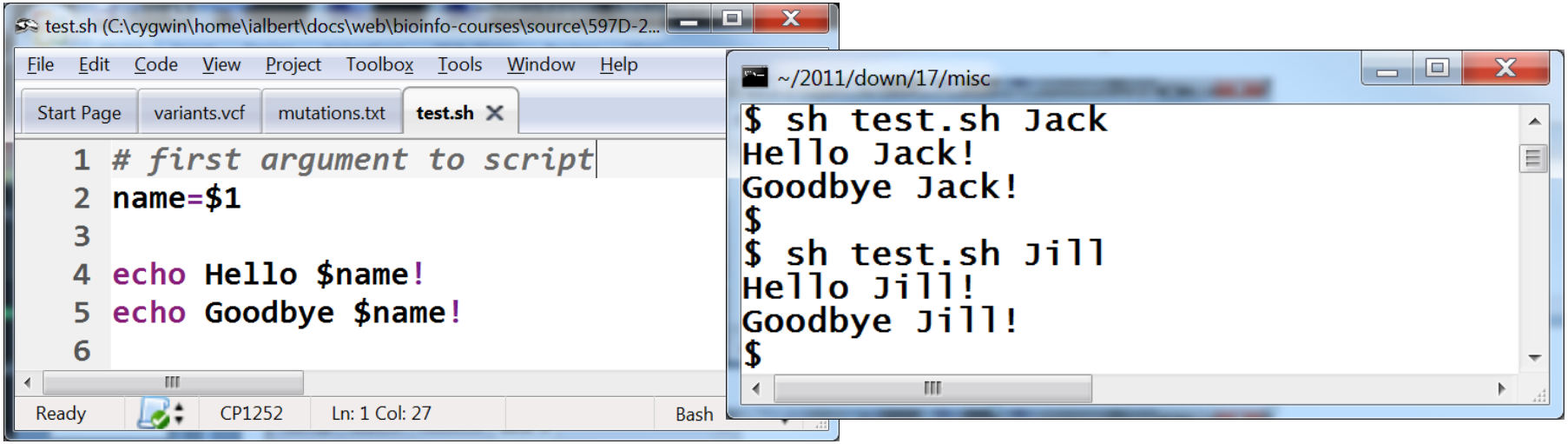
Practical exercise

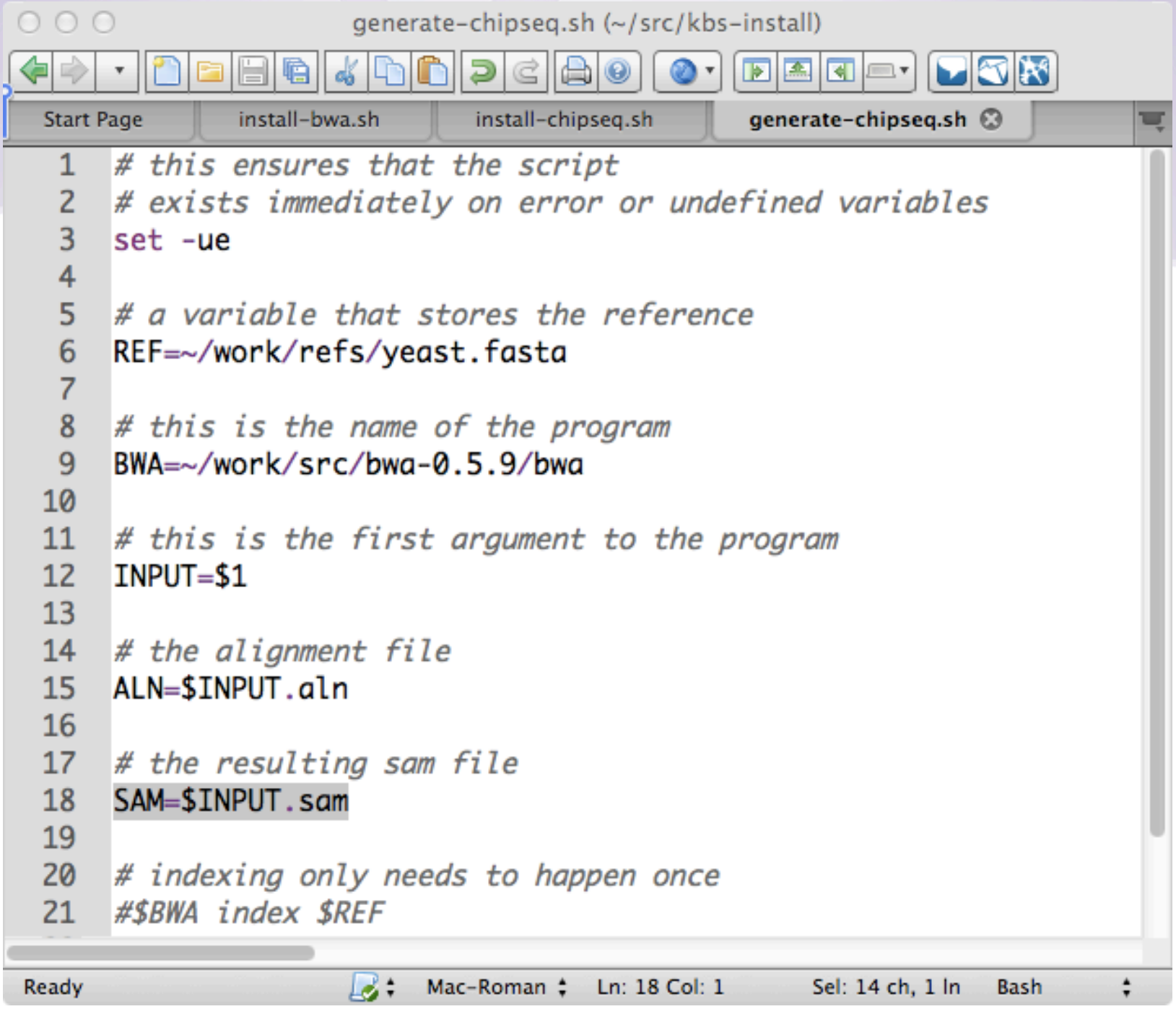
- We have two datasets → the same binding factor was simulated as if it had a short or a long footprint (**long.fq**, **short.fq**)

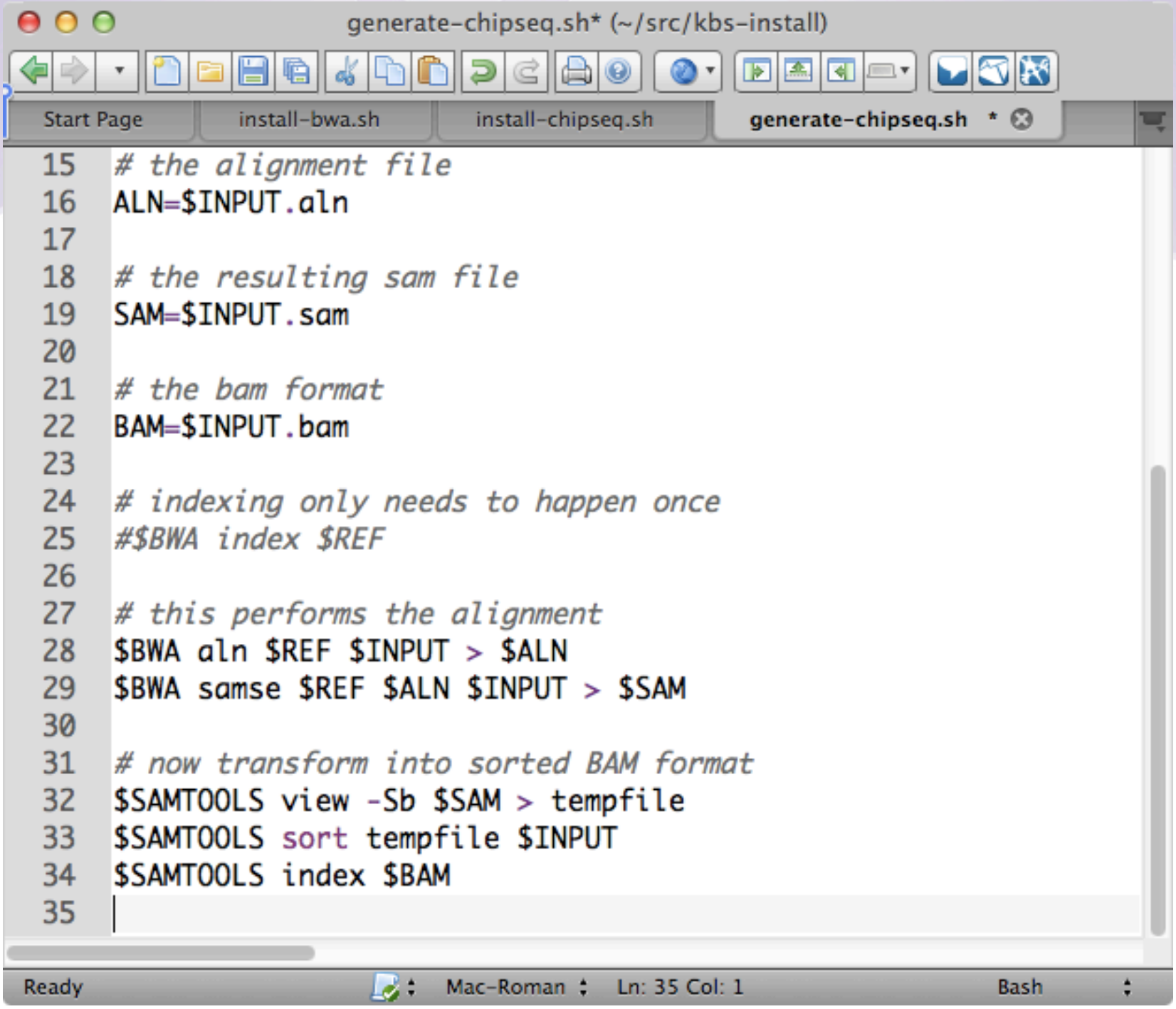
- We will visualize and investigate these: detect bound locations, binding size, peaks etc

Script automation






generate-chipseq.sh (~/src/kbs-install)
Start Page
install-bwa.sh
install-chipseq.sh
generate-chipseq.sh
1 # this ensures that the script
2 # exists immediately on error or undefined variables
3 set -ue
4
5 # a variable that stores the reference
6 REF=~/work/refs/yeast.fasta
7
8 # this is the name of the program
9 BWA=~/work/src/bwa-0.5.9/bwa
10
11 # this is the first argument to the program
12 INPUT=$1
13
14 # the alignment file
15 ALN=$INPUT.aln
16
17 # the resulting sam file
18 SAM=$INPUT.sam
19
20 # indexing only needs to happen once
21 #$BWA index $REF
Ready
Mac-Roman
Ln: 18 Col: 1
Sel: 14 ch, 1 ln
Bash


generate-chipseq.sh* (~/src/kbs-install)
Start Page
install-bwa.sh
install-chipseq.sh
generate-chipseq.sh *
15 # the alignment file
16 ALN=$INPUT.aln
17
18 # the resulting sam file
19 SAM=$INPUT.sam
20
21 # the bam format
22 BAM=$INPUT.bam
23
24 # indexing only needs to happen once
25 #$BWA index $REF
26
27 # this performs the alignment
28 $BWA aln $REF $INPUT > $ALN
29 $BWA samse $REF $ALN $INPUT > $SAM
30
31 # now transform into sorted BAM format
32 $SAMTOOLS view -Sb $SAM > tempfile
33 $SAMTOOLS sort tempfile $INPUT
34 $SAMTOOLS index $BAM
35
Ready
Mac-Roman
Ln: 35 Col: 1
Bash

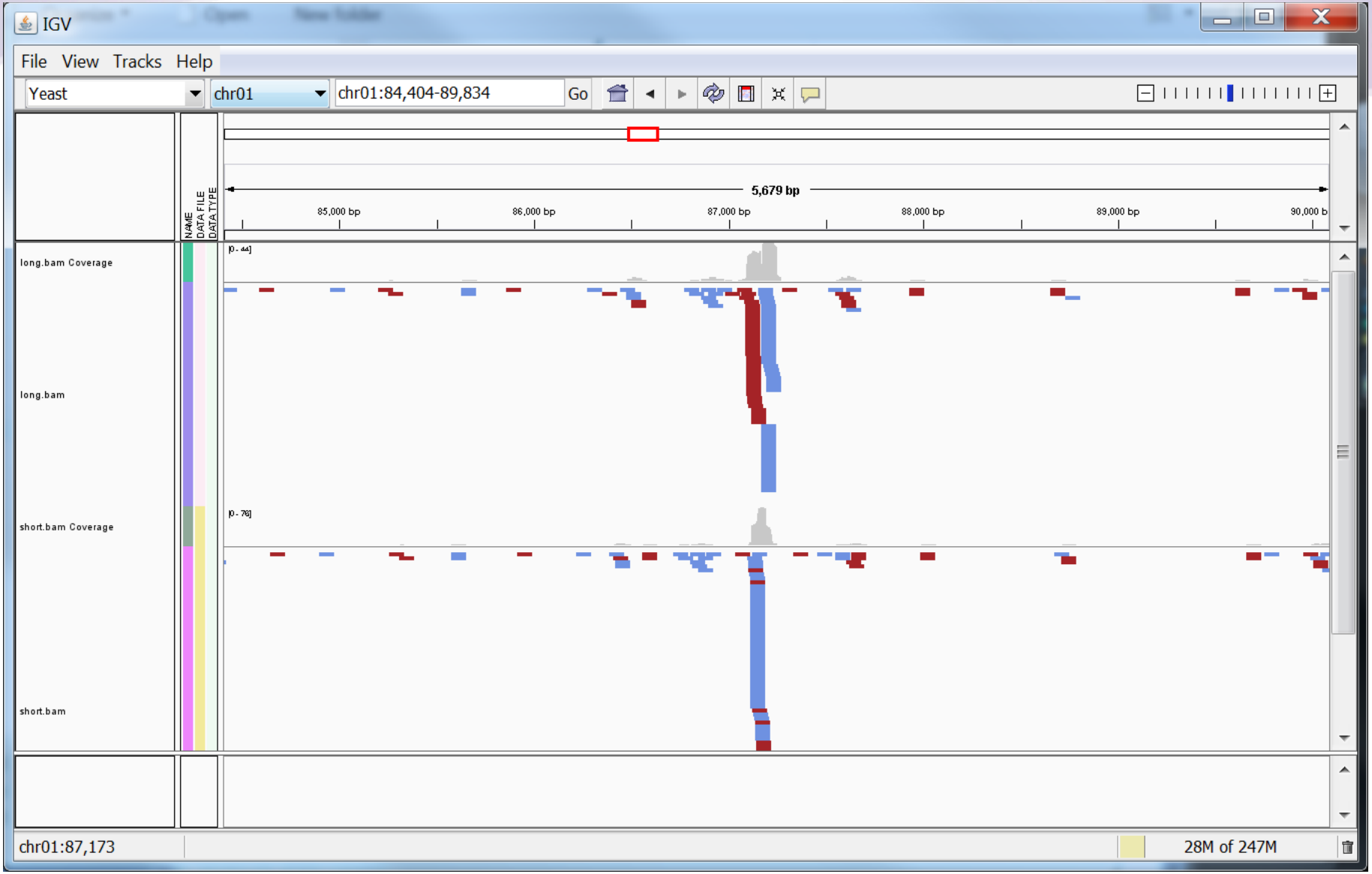
Run the analysis for both files

```
sh ~/kbs-install/generate-chipseq.sh data/short.fq
```

```
sh ~/kbs-install/generate-chipseq.sh data/long.fq
```

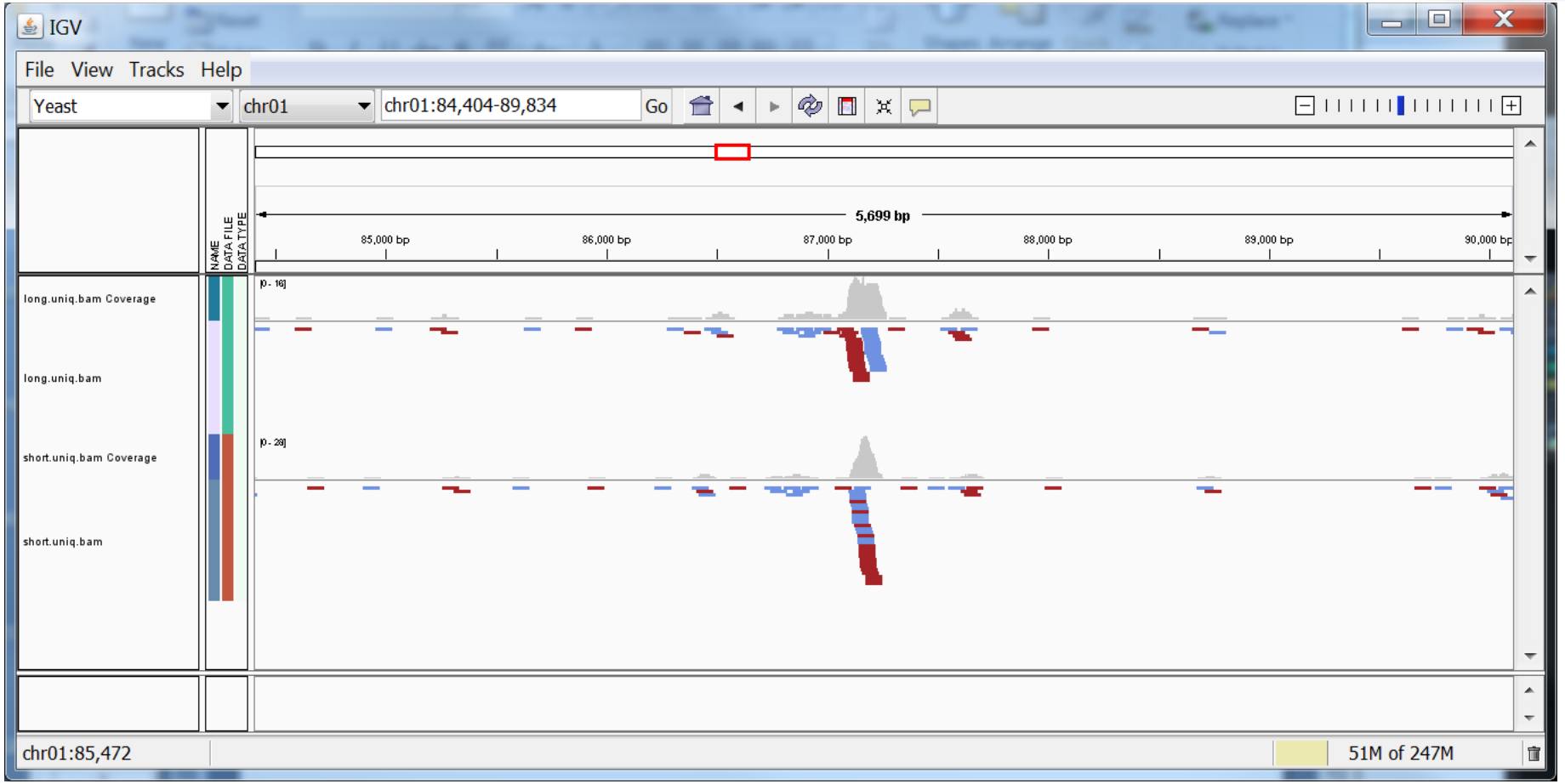
You have to pass the right path to the script and the right path of the file

Visualize in IGV



After de-duplication

(add the necessary steps to your pipeline and rerun)

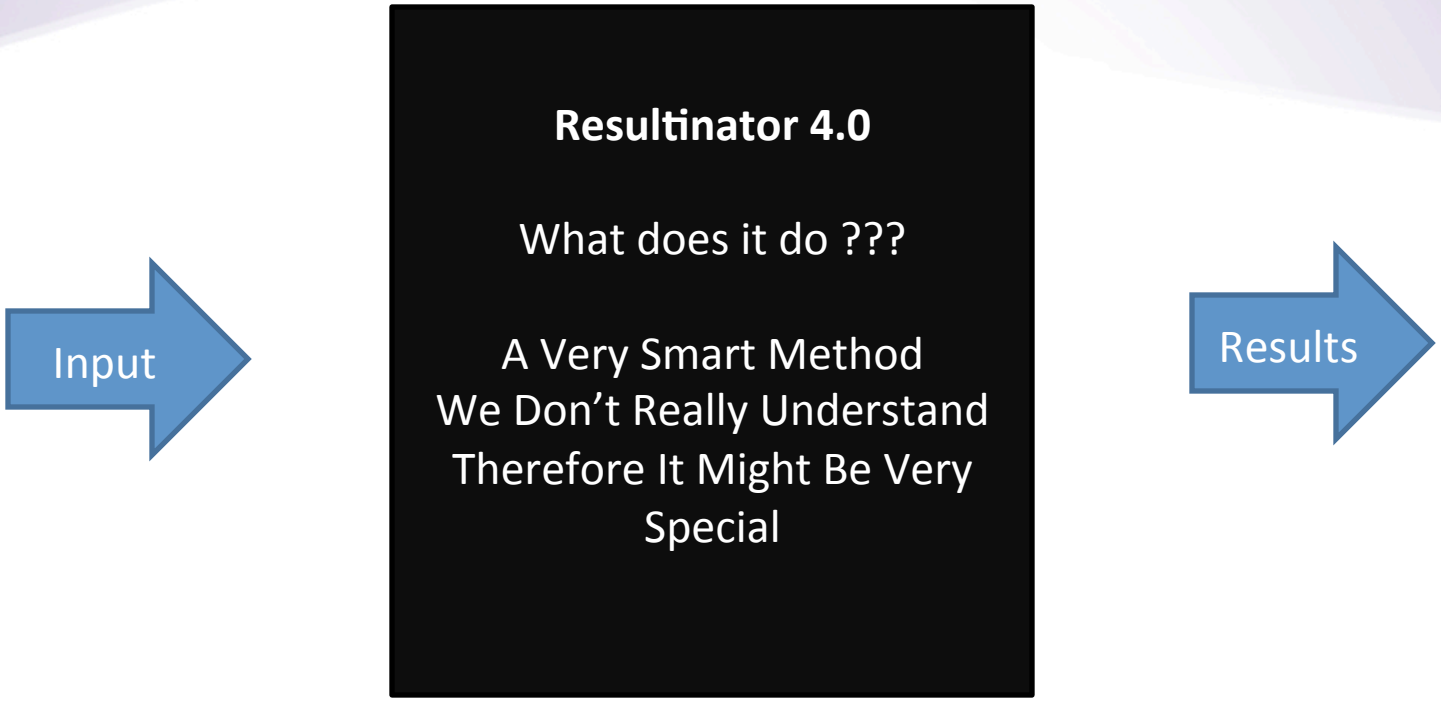


Back to Chip-Seq

Identify bound locations then

1. tabulate the observations → protein X binds at locations Y
2. compare between samples to detect different level of occupancy
3. extract sequences for binding locations and find motifs
4. study common attributes of genomic features nearby bound locations

Beware of Black Boxes



- The **more sophisticated** a method the **less likely** is that we understand its limitations and applicability !
- In general always **do a simple analysis first** – do common sense, straightforward comparisons – you'll be repeatedly surprised just how far you can get that way
- Then move to more complicated approaches

Visualization can be counterintuitive

Read extension/trimming

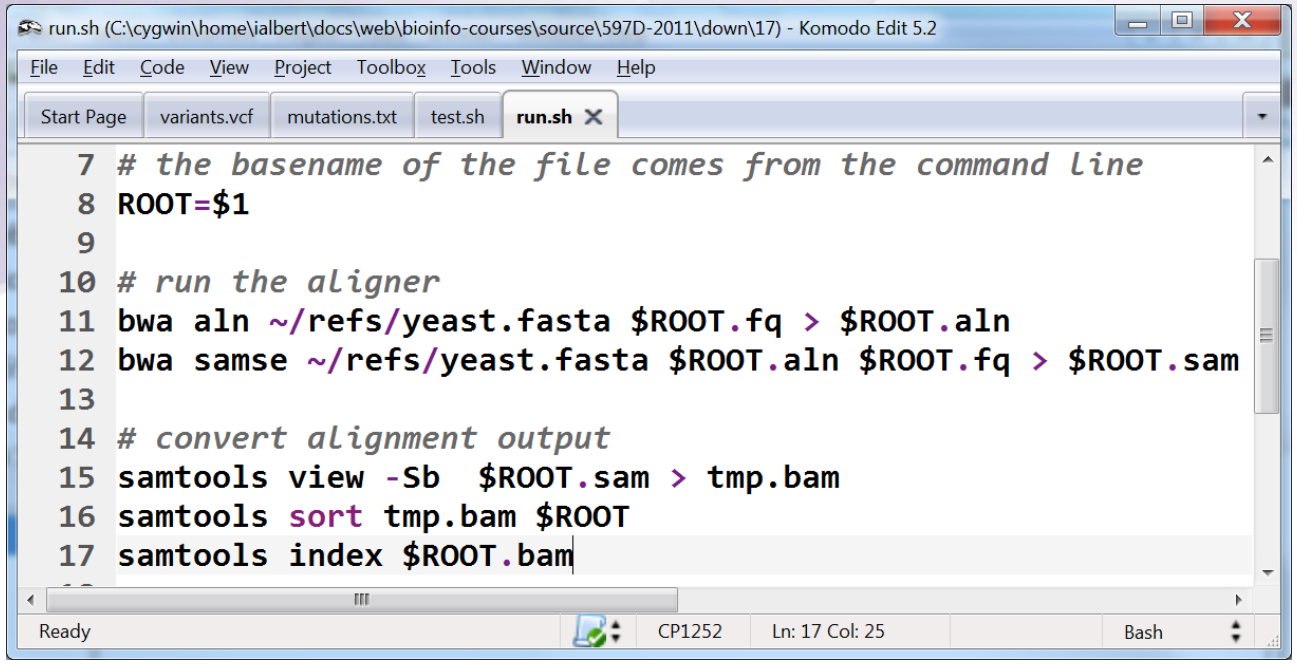
- Extend to full fragment:
  - operate on the actual fragments that were under study
  - we need to know how far to extend

- Trim to 5' end:
  - use the two ends as markers of the binding location
  - twice as much data to keep track of (+/-) borders

In both cases since strands are sequenced independently we can treat the data as technical replicates → allows us to evaluate errors/noise

Get to know you data → compute the coverage

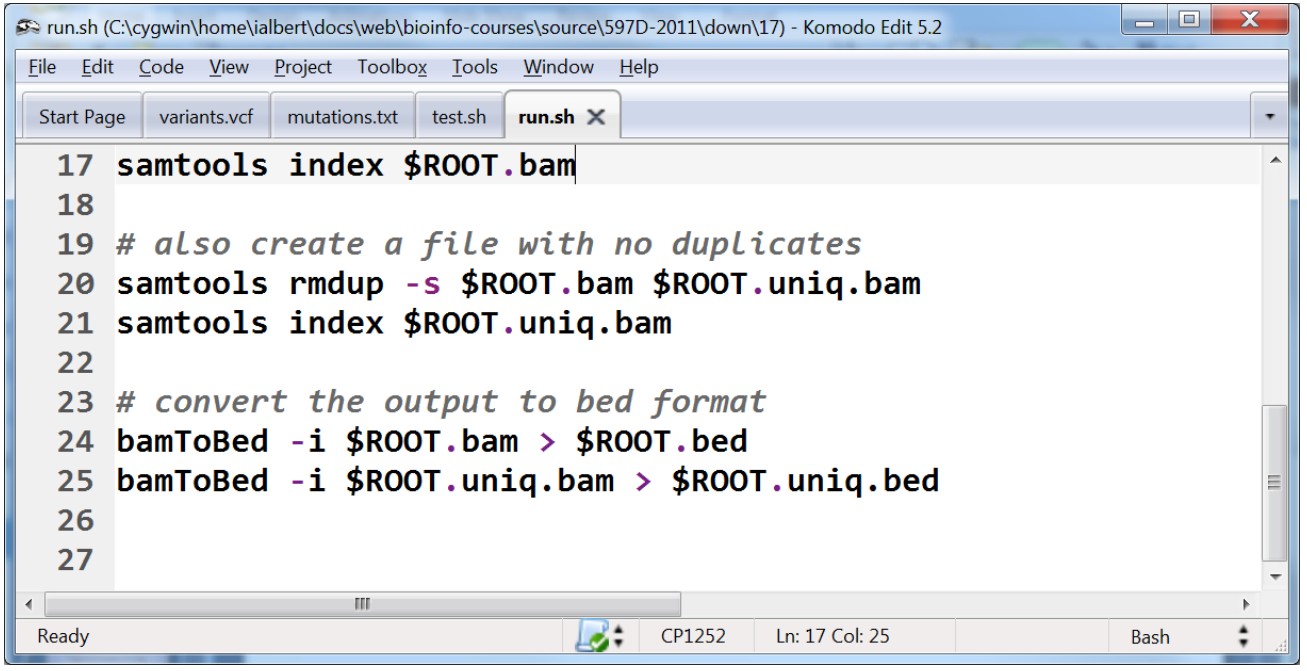
- Convert to wiggle format (WIG) → useful and required for some tools
- Convert to coverage → we can query it with tabix
- We will continue editing our pipeline from lecture 17



All steps in one file allows us to repeat the analysis

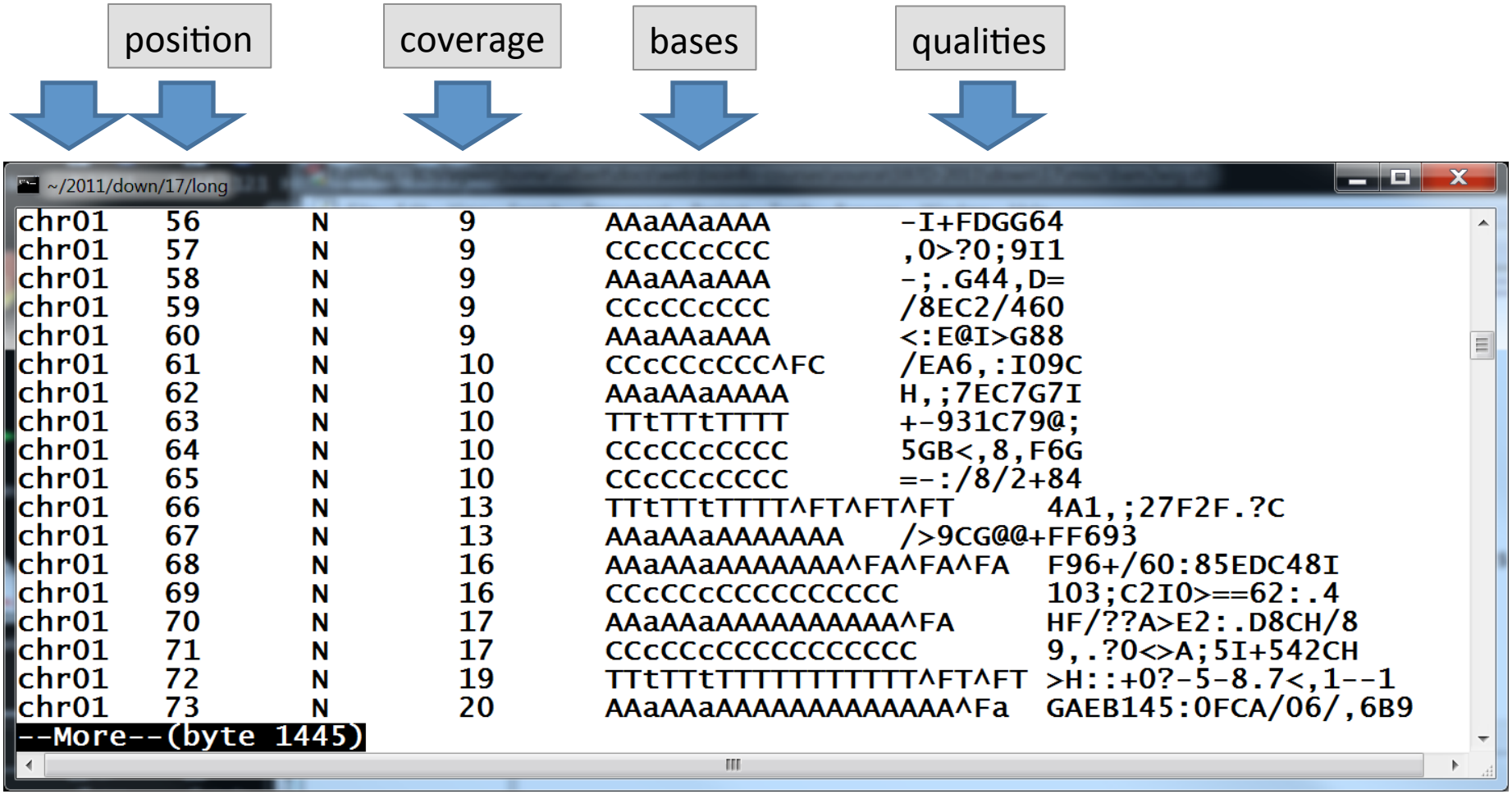
Don't copy it!

Build it one step at a time! Then comment out the previous step

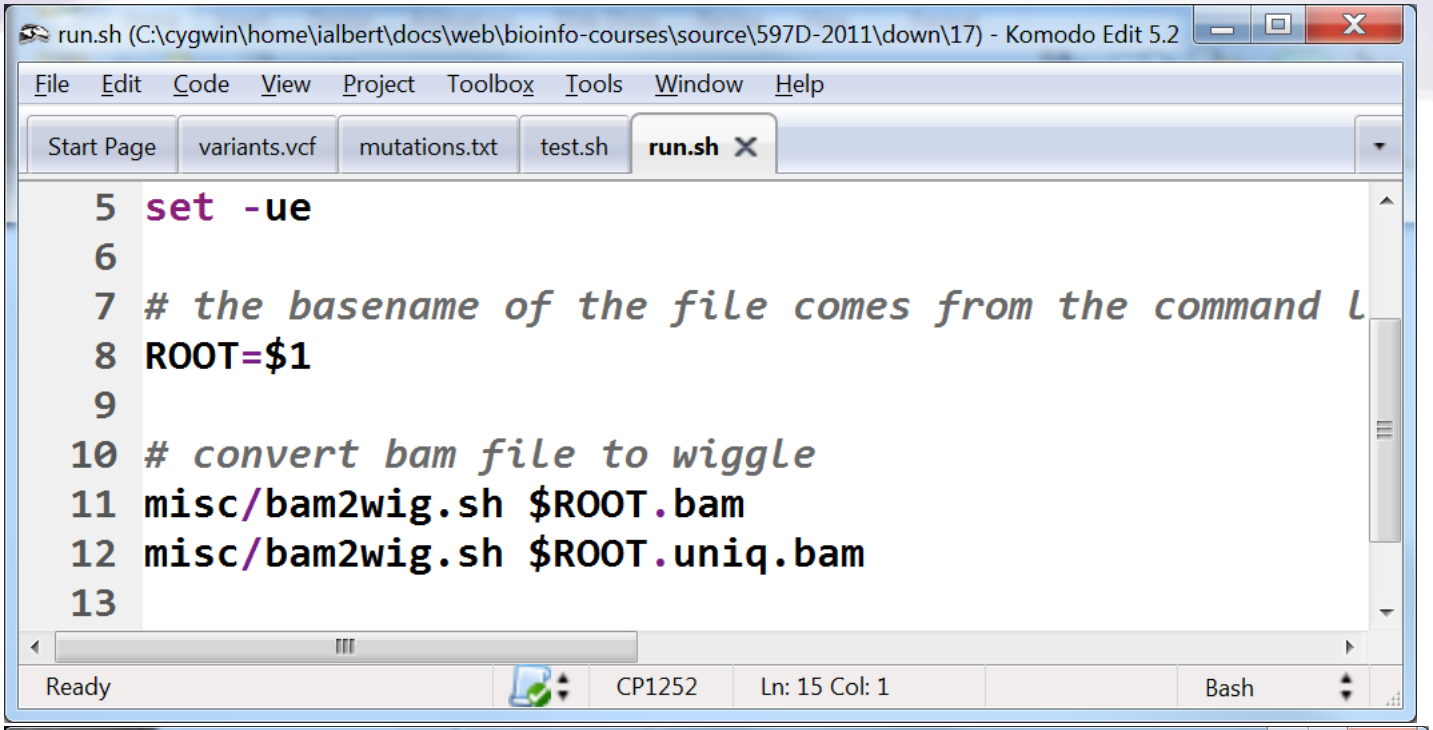


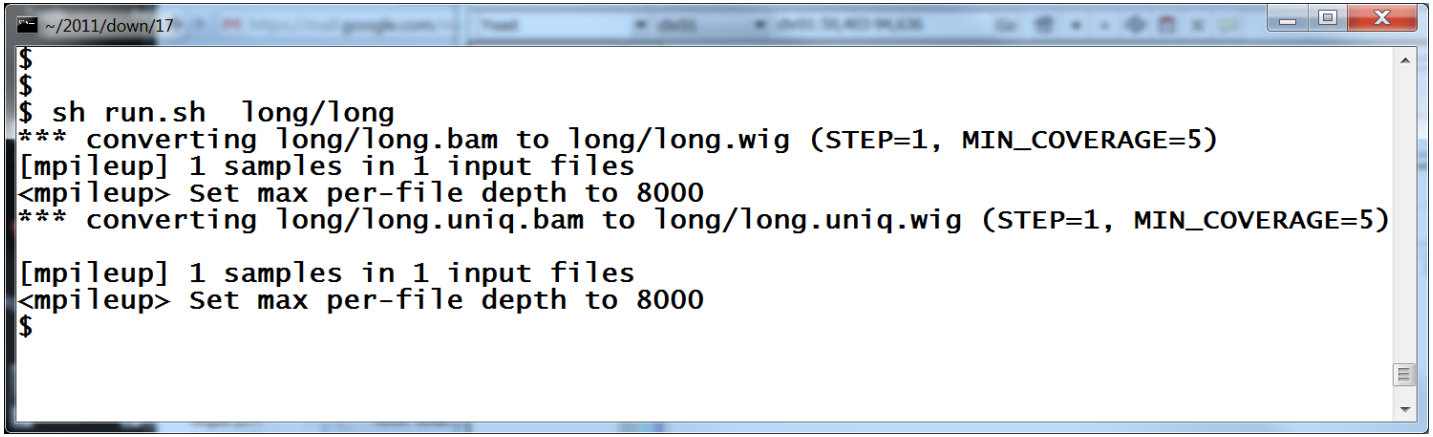
Samtools mpileup output

samtools mpileup long.bam

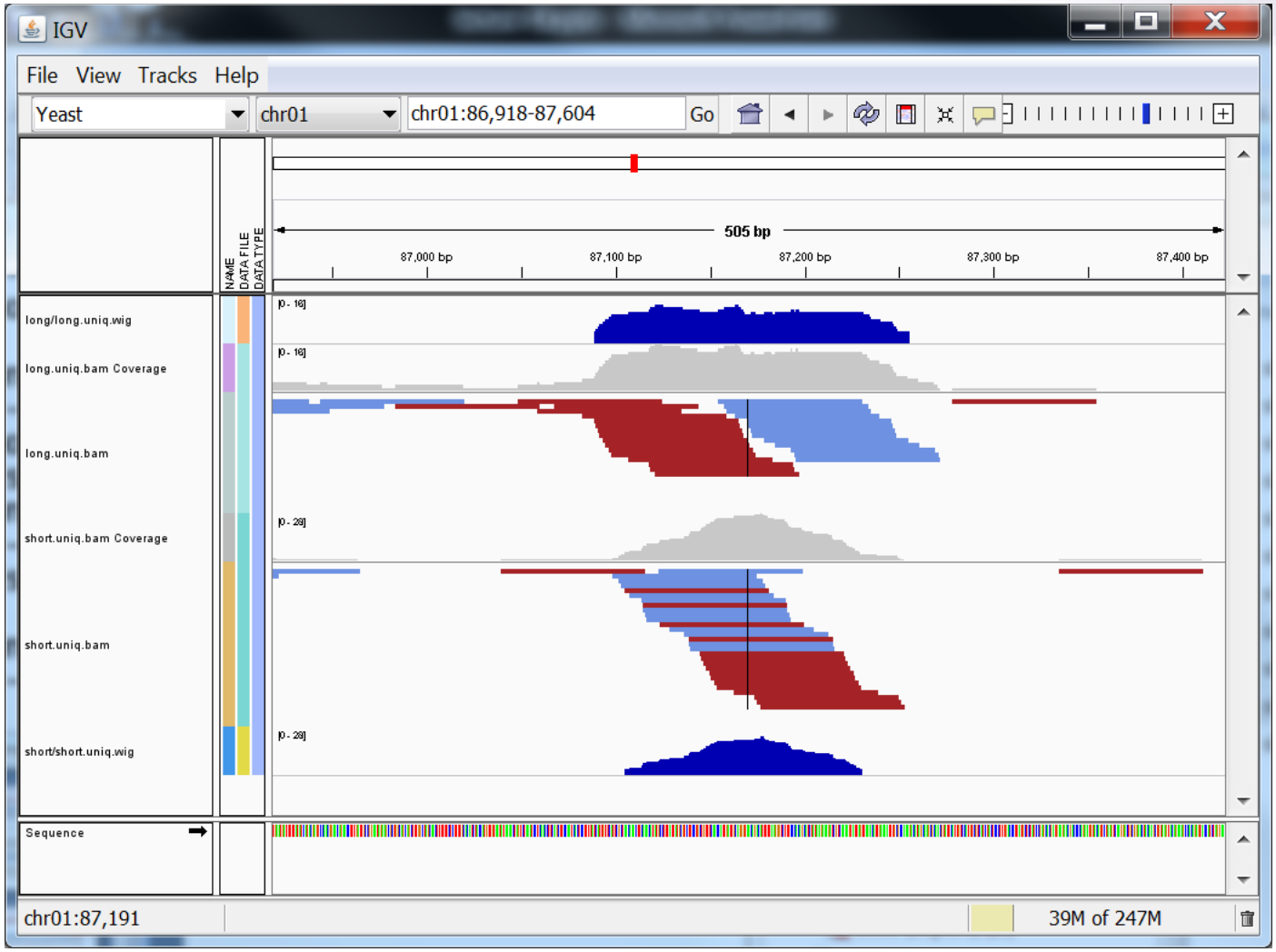


Lets transform the BAM file to wiggle





Visualizing the filtered wiggle file



Peak calling tools

- Very large selection of tools and techniques:

  ERANGE, FindPeaks, MACS, QuEST, CisGenome, SISSRS, USeq, PeakSeq, SPP, ChIPSeqR, GLITR, ChIPDiff, T-PIC, BayesPeak, MOSAiCS, CCAT, CSAR,

Featured today (historically precedes many of the tools above)

GeneTrack—a genomic data processing and visualization framework

Istvan Albert, Shinichiro Wachi, Cizhong Jiang and B. Franklin Pugh

Bioinformatics, 2008

The nomenclature is a bit hazy

- The term “peaks”, “signal”, and “enriched regions” are used interchangeably → leads to a lot of confusion

Proper definition would be

- Peaks → regions with a well defined shape, a midpoint and a spread that describe one or more measurements

- Enrichment → selecting those peaks that show statistically significant properties when compared to background/control

GeneTrack

- It performs error corrections and peak predictions
- Visualizes data as tracks in a ‘browser’ interface.
- Also a integrated into our own LIMS: Laboratory Data Management System LionDB3: with projects, users, data sharing etc.

http://genetrack.googlecode.com the LIMS is at http://liondb3.atlas.bx.psu.edu/

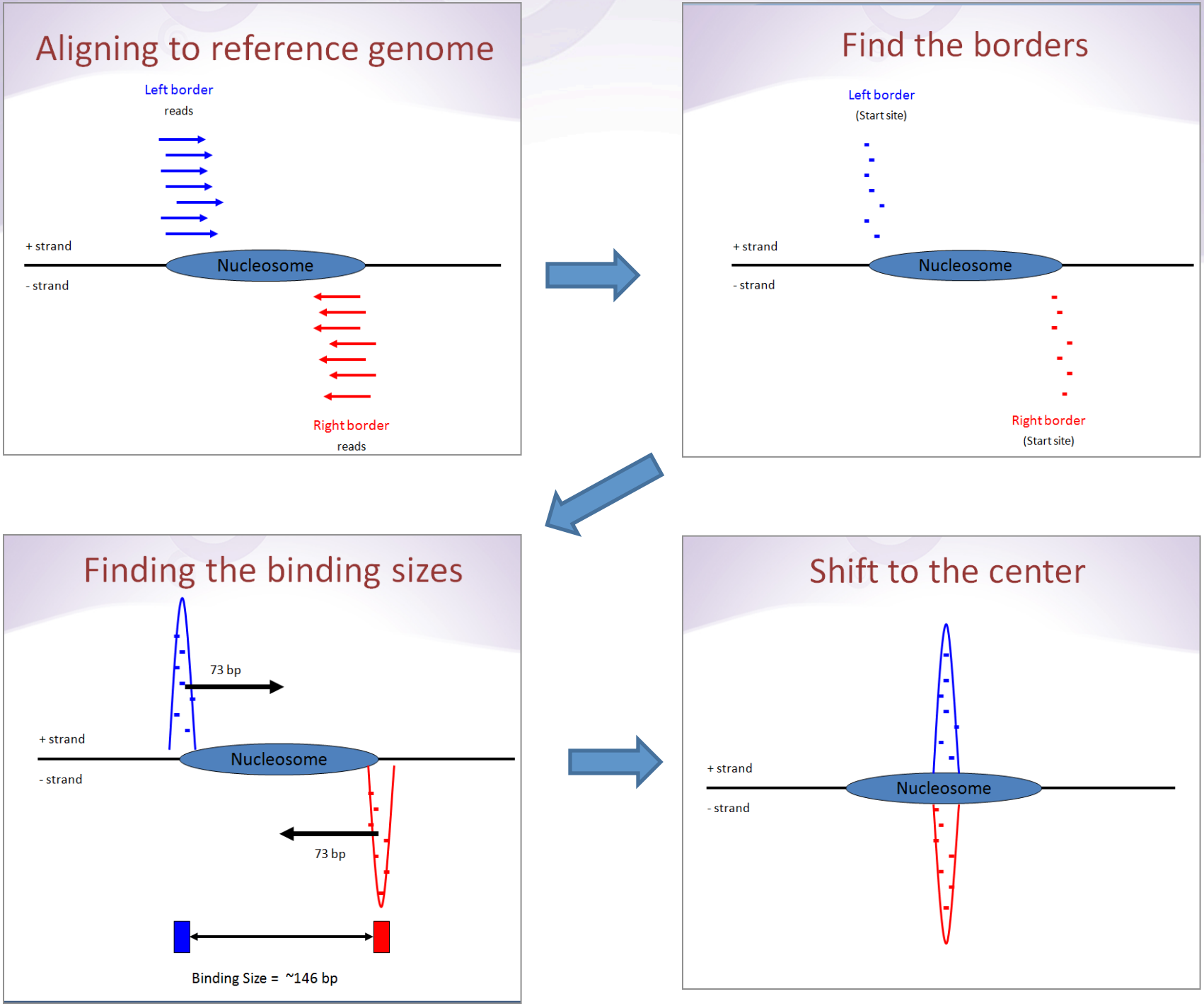
GeneTrack Features

Unique features:

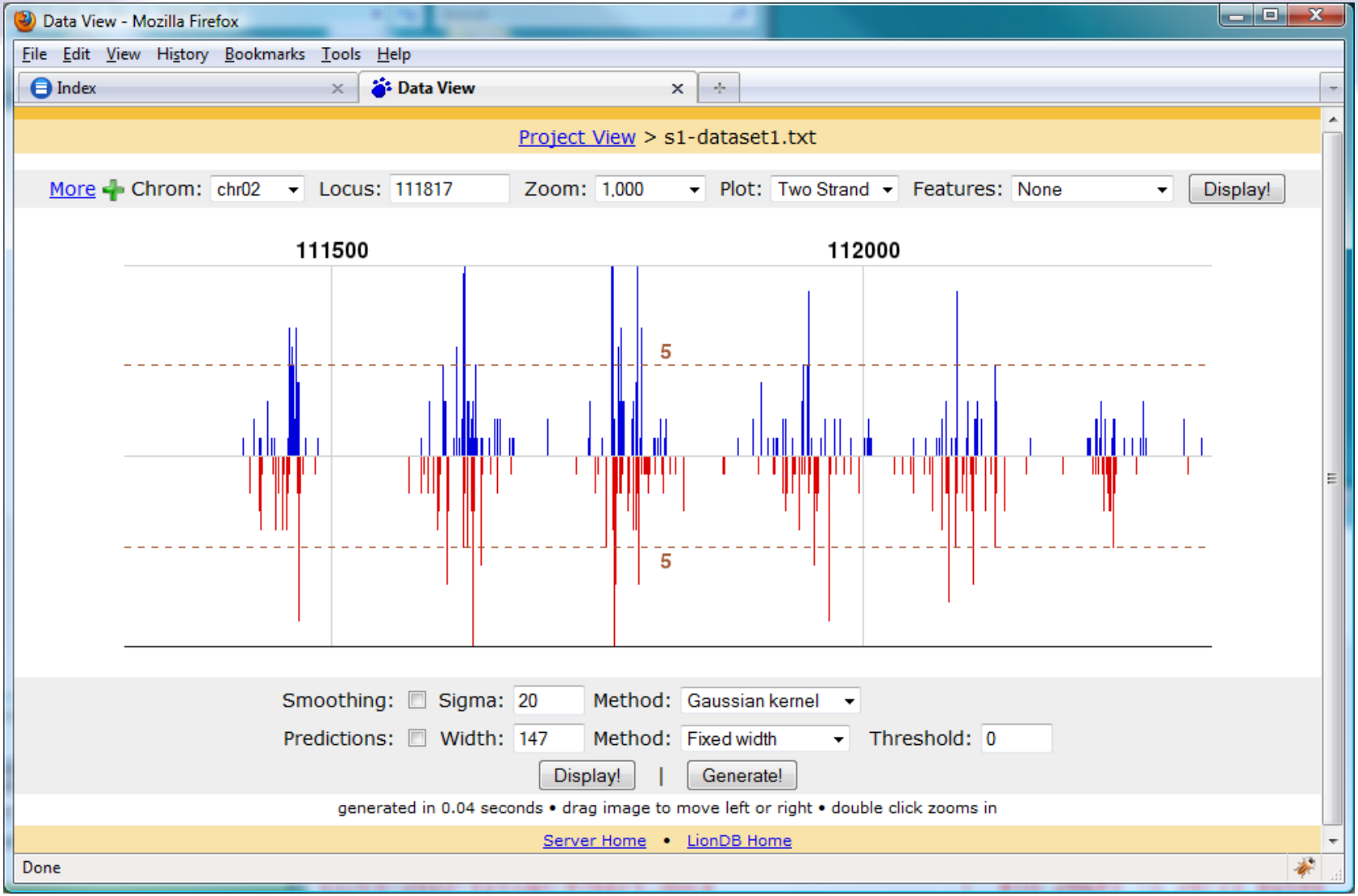
- Peak detection and visualization at the same time.
- User sees the effect of the peak prediction parameters right away

Much criticized missing component

- Lacks the statistical modeling of the background (expected frequencies)

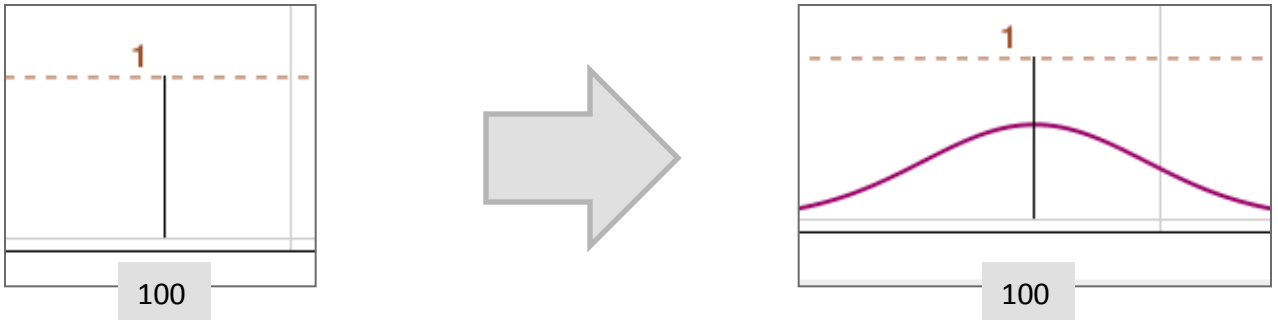

Aligning to reference genome
Left border
reads
+ strand
Nucleosome
- strand
Right border
reads
Find the borders
Left border
(Start site)
+ strand
Nucleosome
- strand
Right border
(Start site)
Finding the binding sizes
73 bp
+ strand
Nucleosome
- strand
73 bp
Binding Size = ~146 bp
Shift to the center
+ strand
Nucleosome
- strand

GeneTrack: reads on both strands



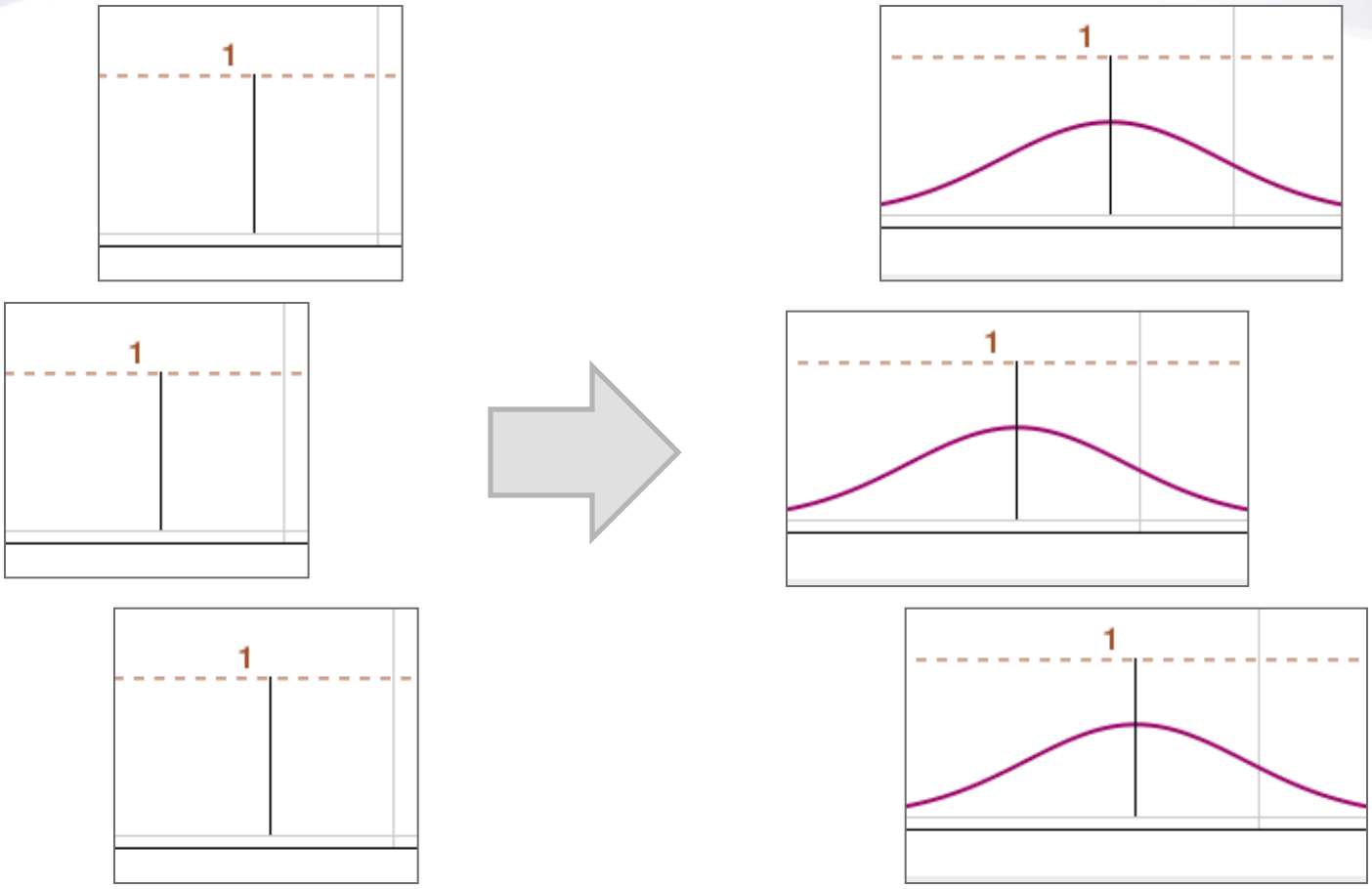
Error Model

- Replace every read with the probability of the read falling onto the observed location.



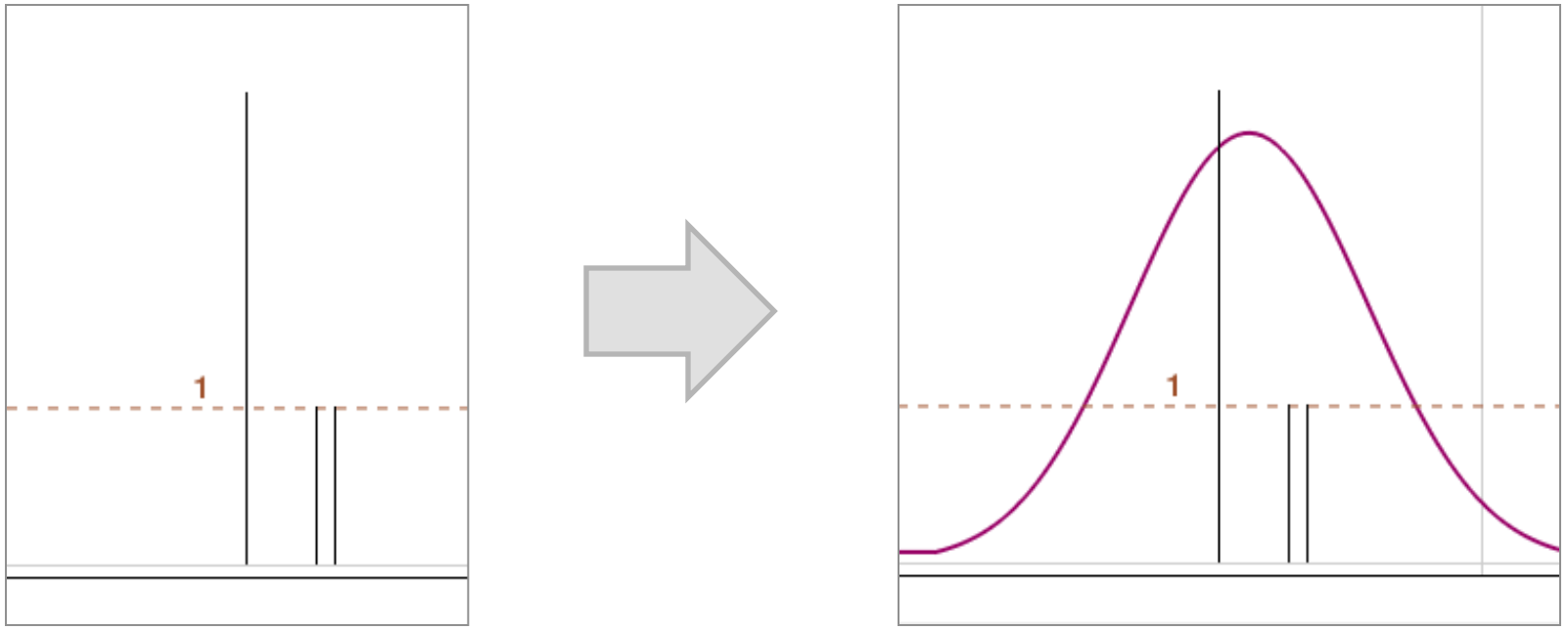
Each single coordinate will be replaced with a “smudge” of values (hundreds of them).

Sum up your “smudges”

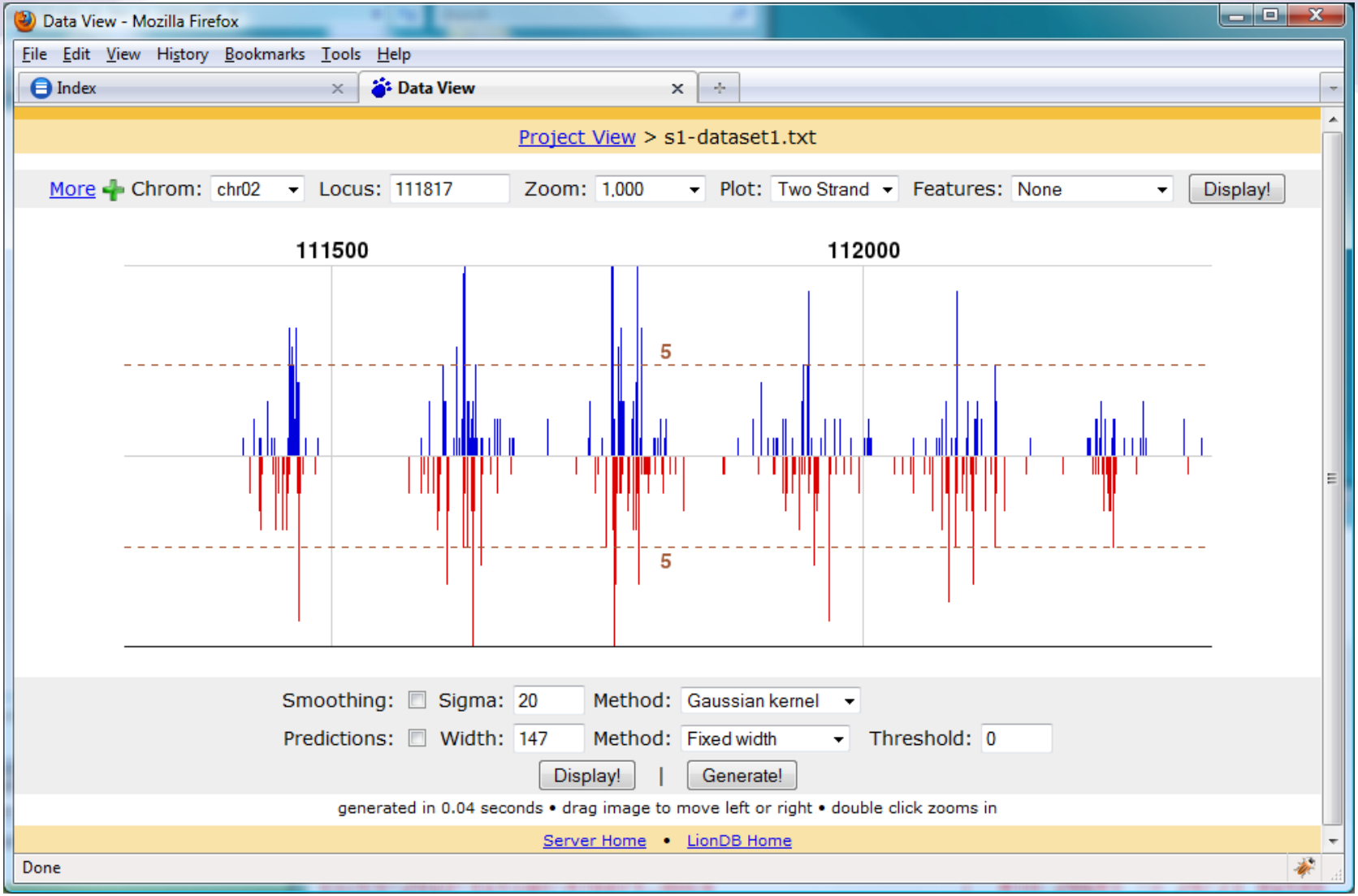


Central Limit Theorem guarantees another normal function

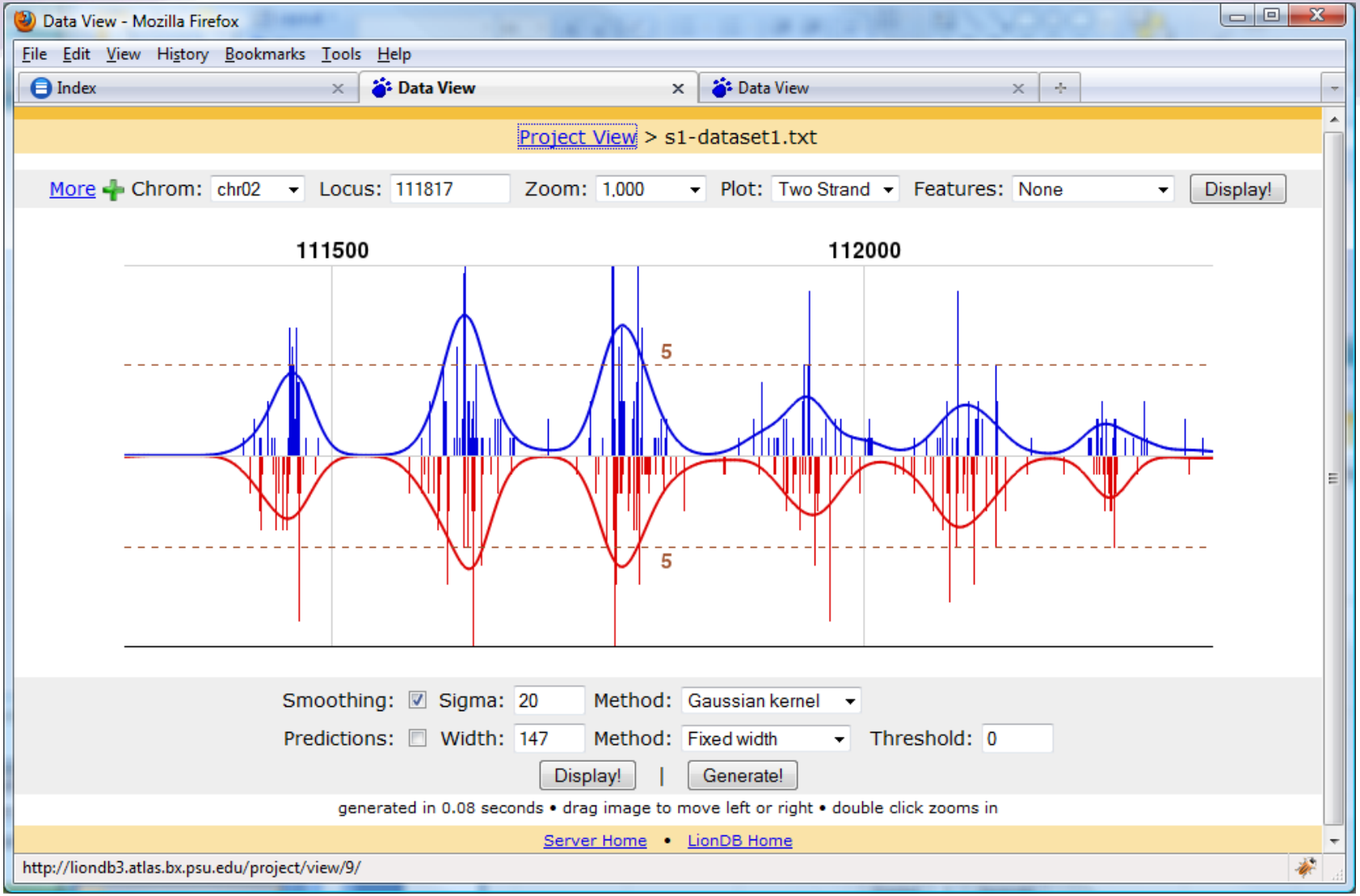
Summing up multiple errors



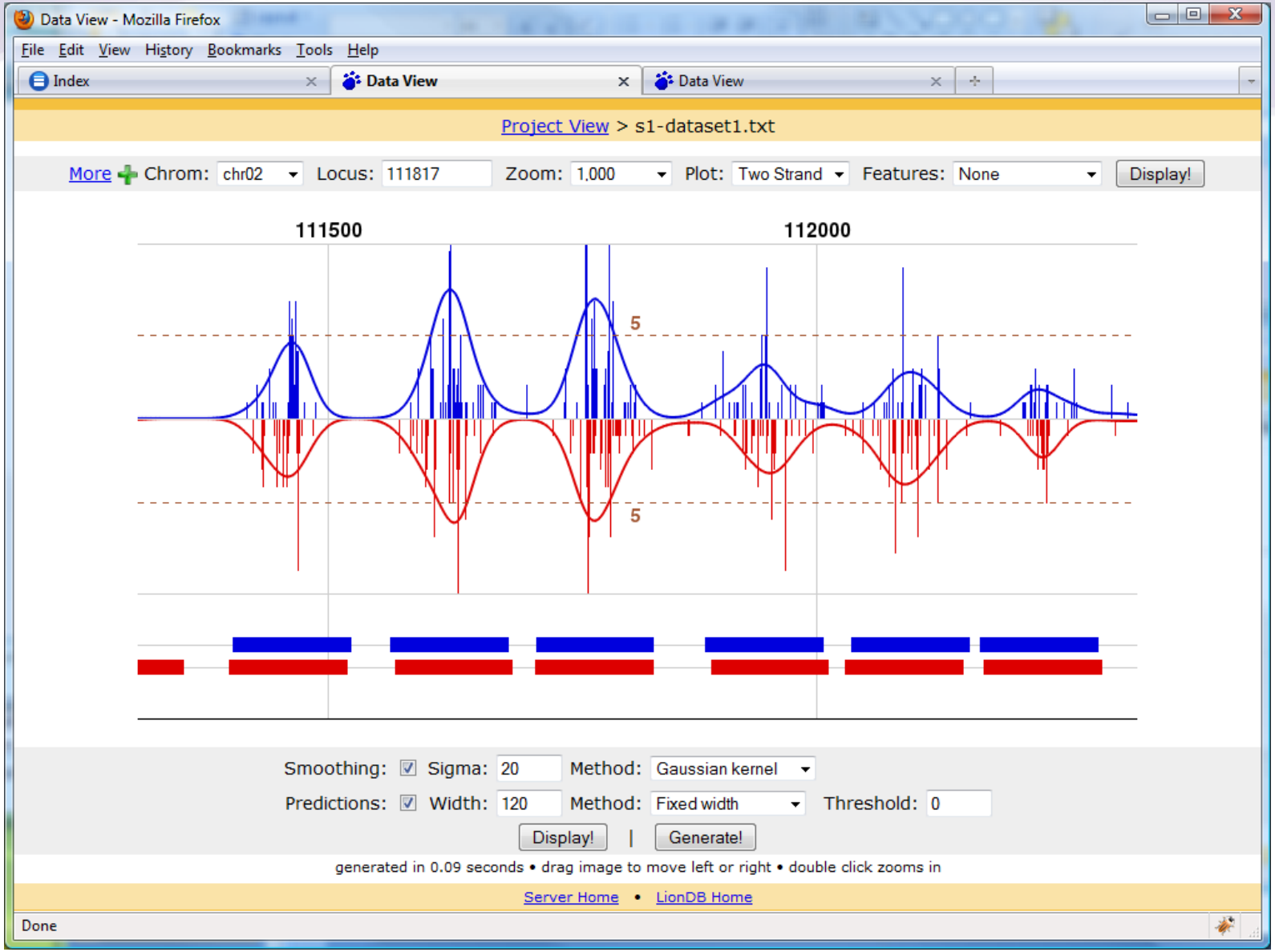
GeneTrack: revisit the reads



GeneTrack: with smoothing



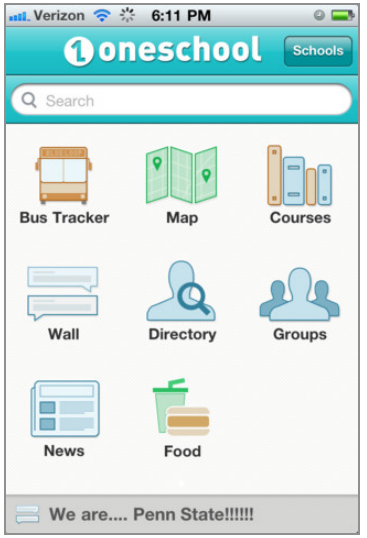
GeneTrack also does peak prediction



Displayed relative to genomic features

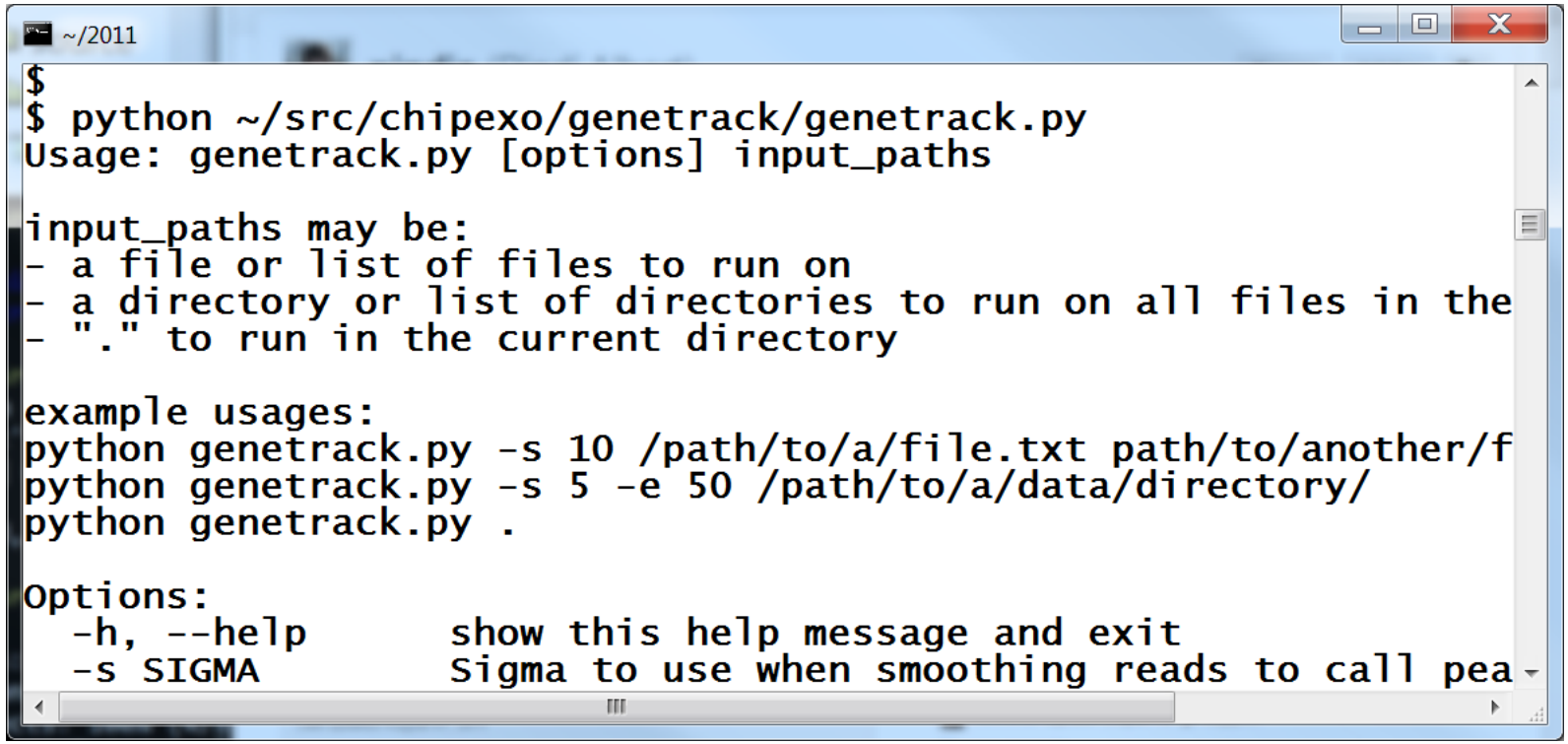
Command line GeneTrack

- Author Pindi Albert (freshman at PSU)
- The functionality of GeneTrack website ported to a command line tool
- Also the author of many cool stuff (not science related alas):



Clone the **chipexo** and **bioawk**

- Clone my "**fork**" of it rather than the original since I made some changes for this class



Running the GeneTrack peak caller