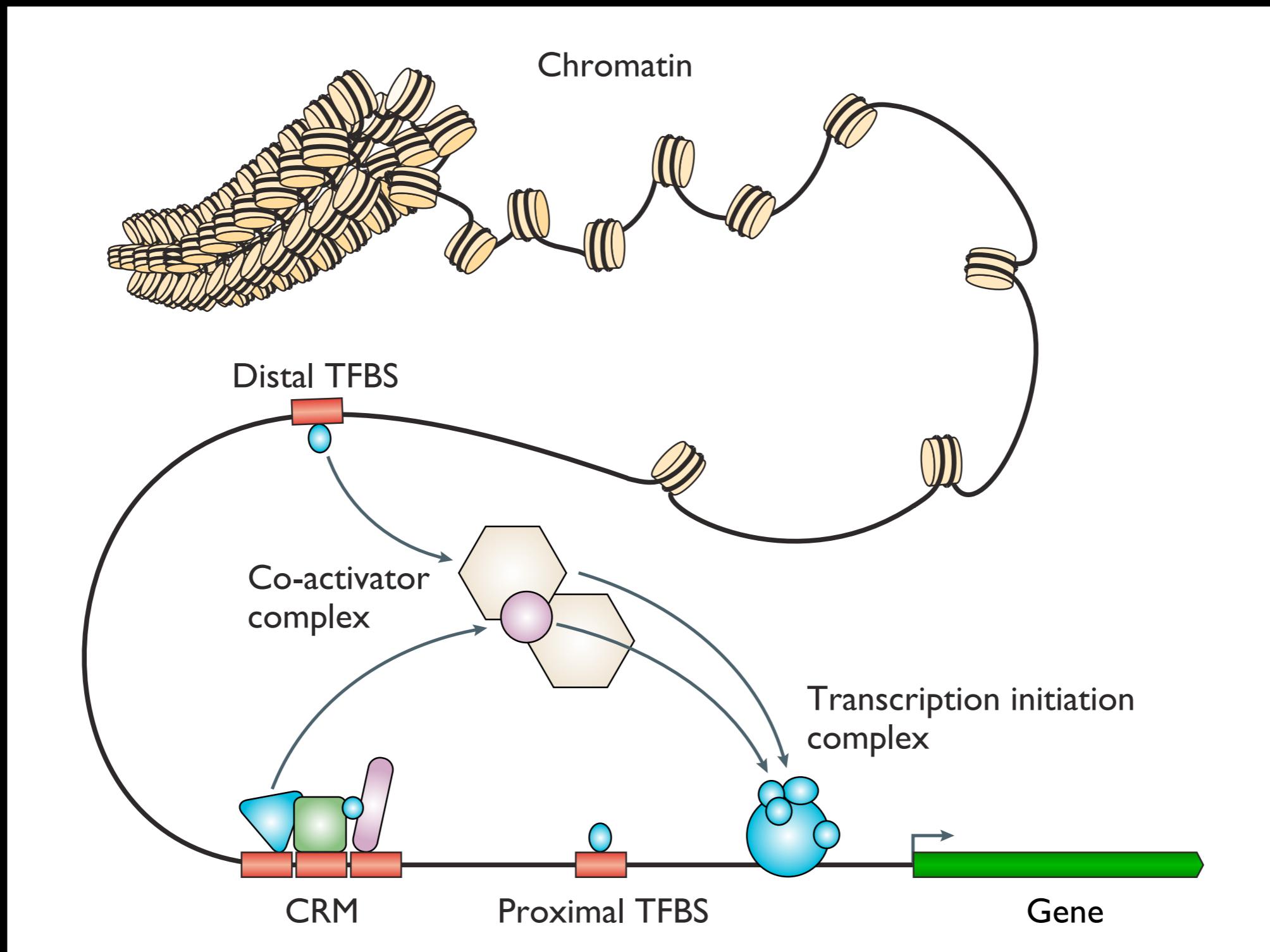


“Regulatory genomics”

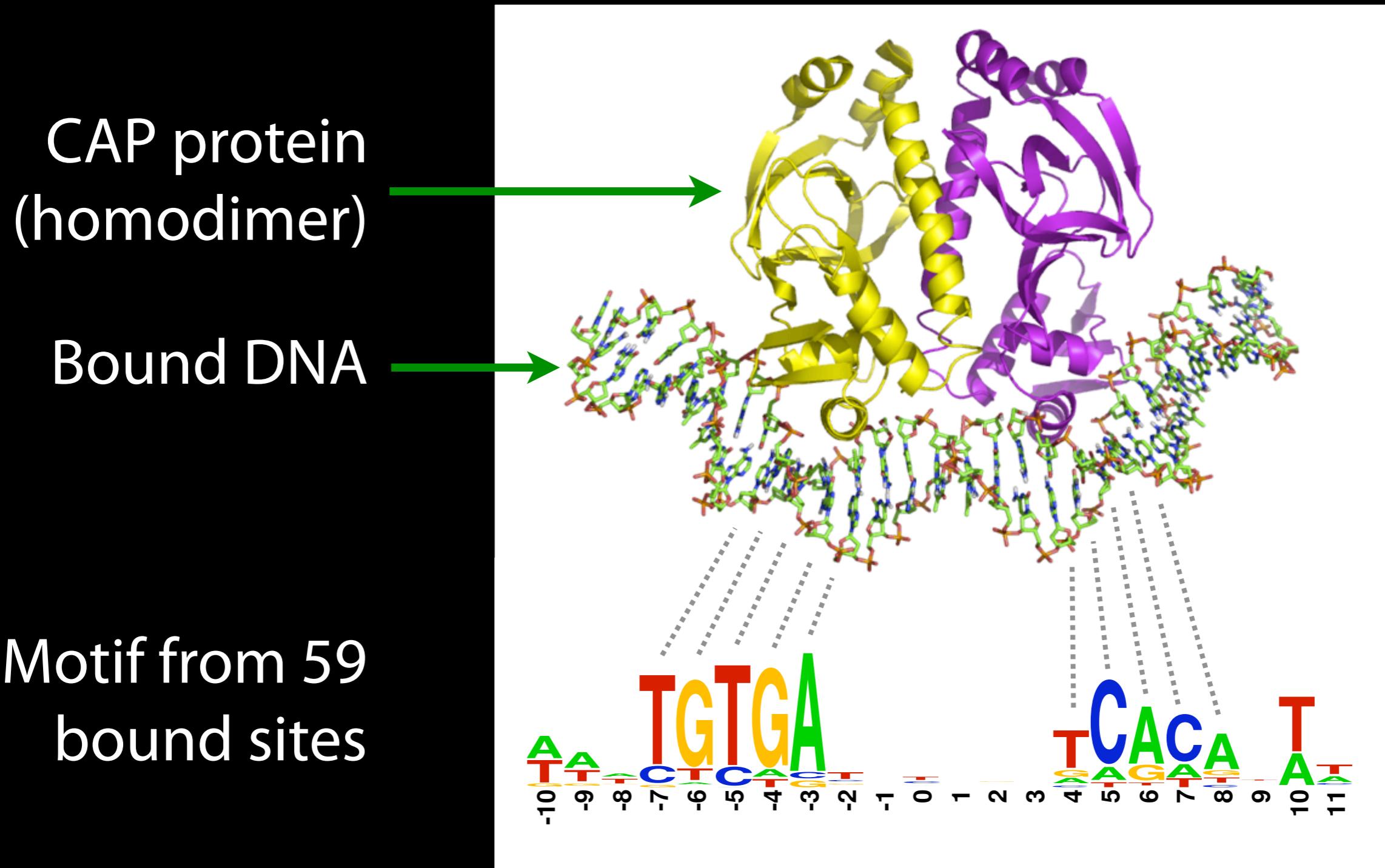
# Regulation of gene transcription



Wasserman and Sandelin. *Nature Review Genetics*. 2004.

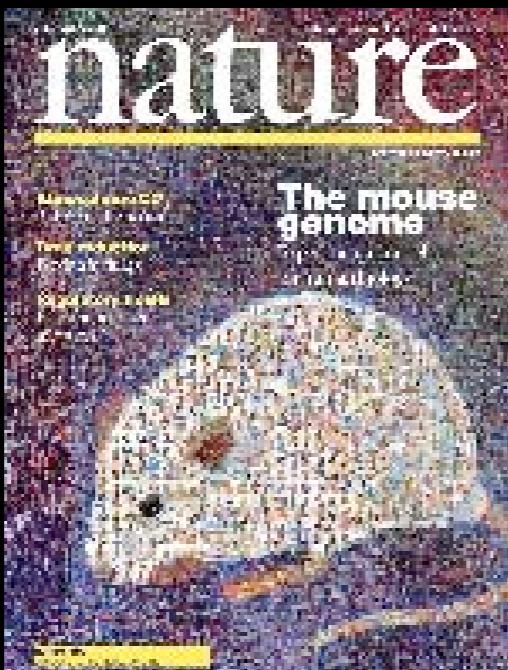
**If only it were even *that* simple...**

# Sequence specific binding yields constraint

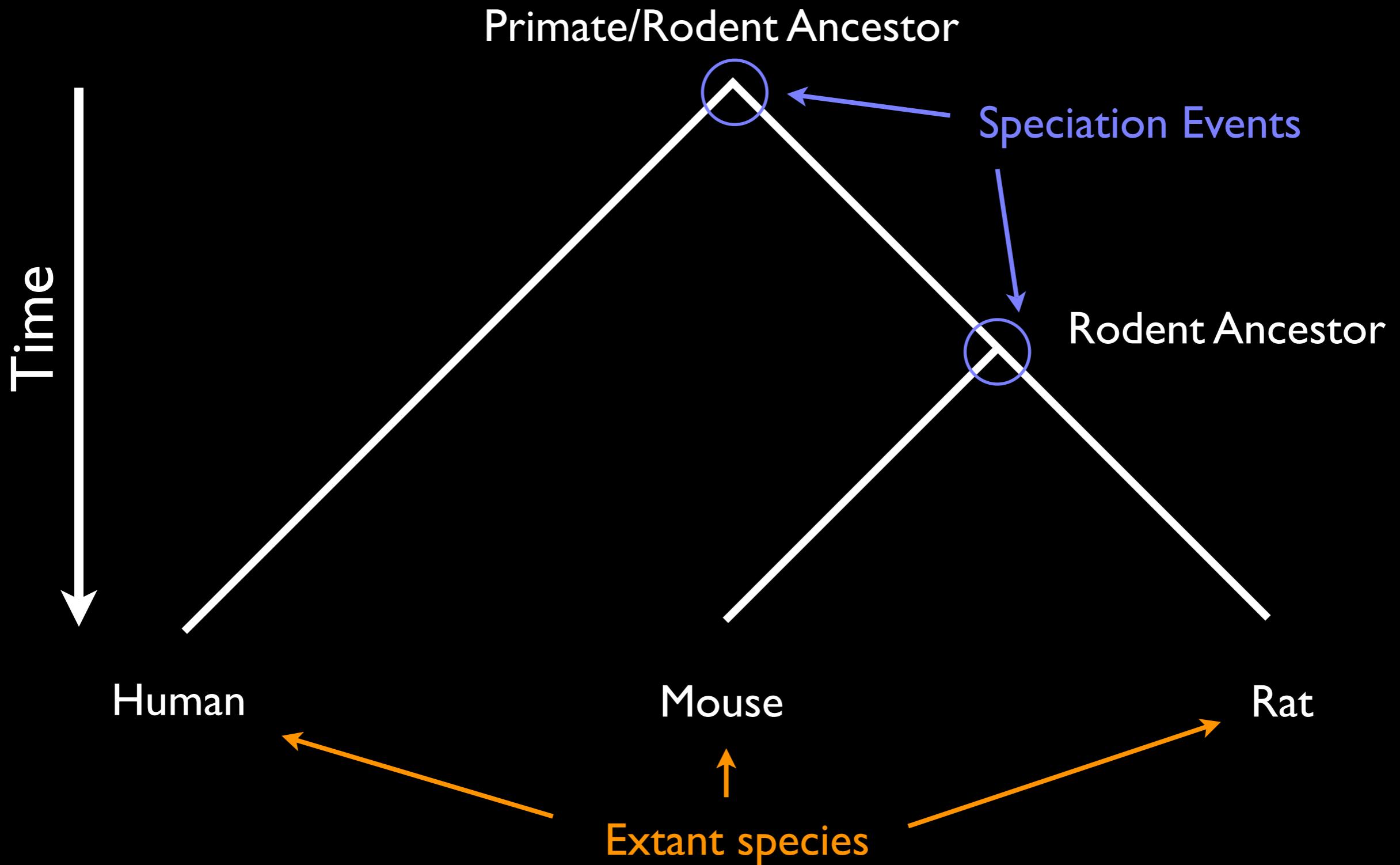


Structure: Schultz et al. *Science*. 1991, Logo software: Crooks et al. *Genome Research*. 2004

# The genomic era... Comparative

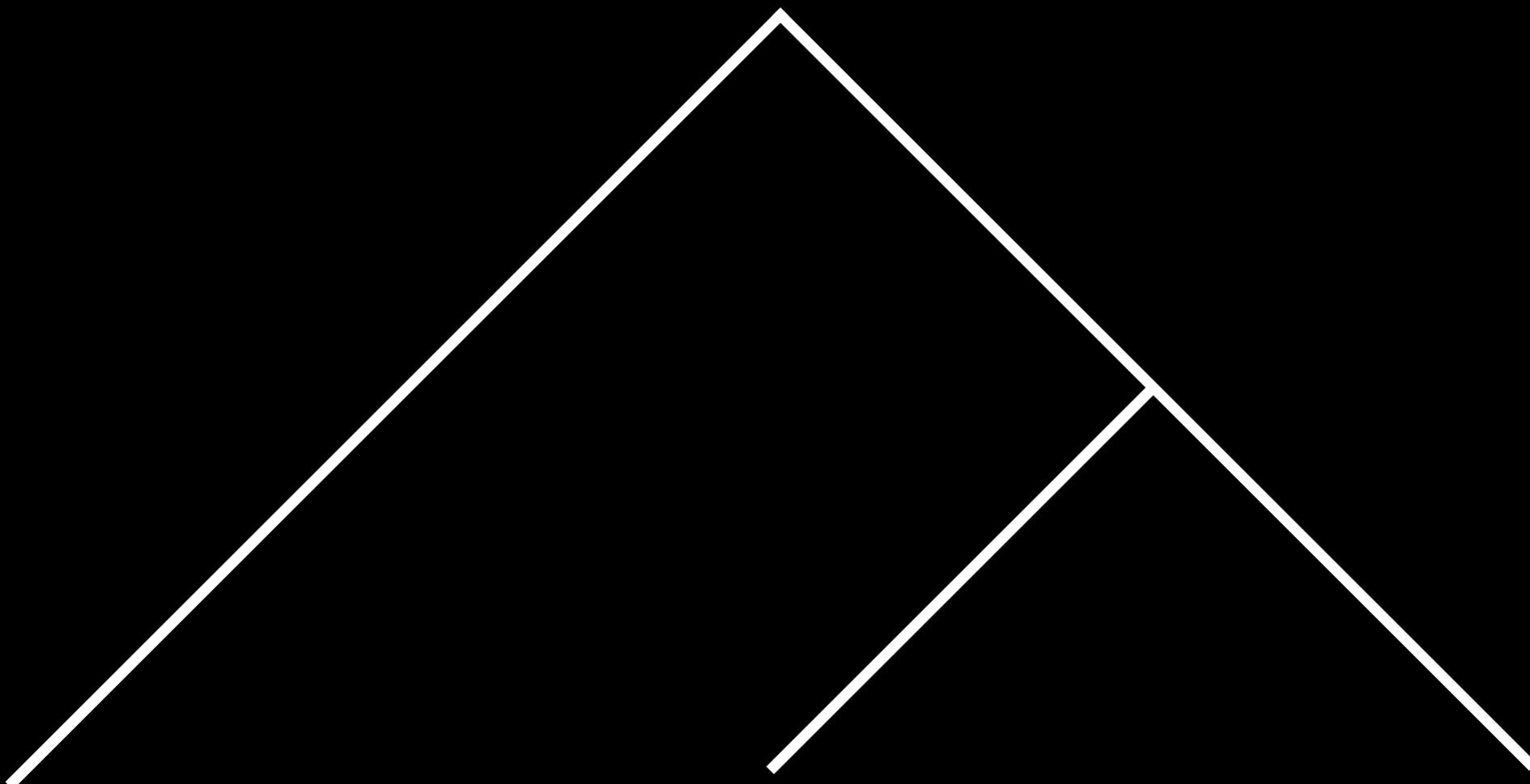


# Evolution of functional elements



# Evolution of functional elements

CTCCCCAGCTGCC

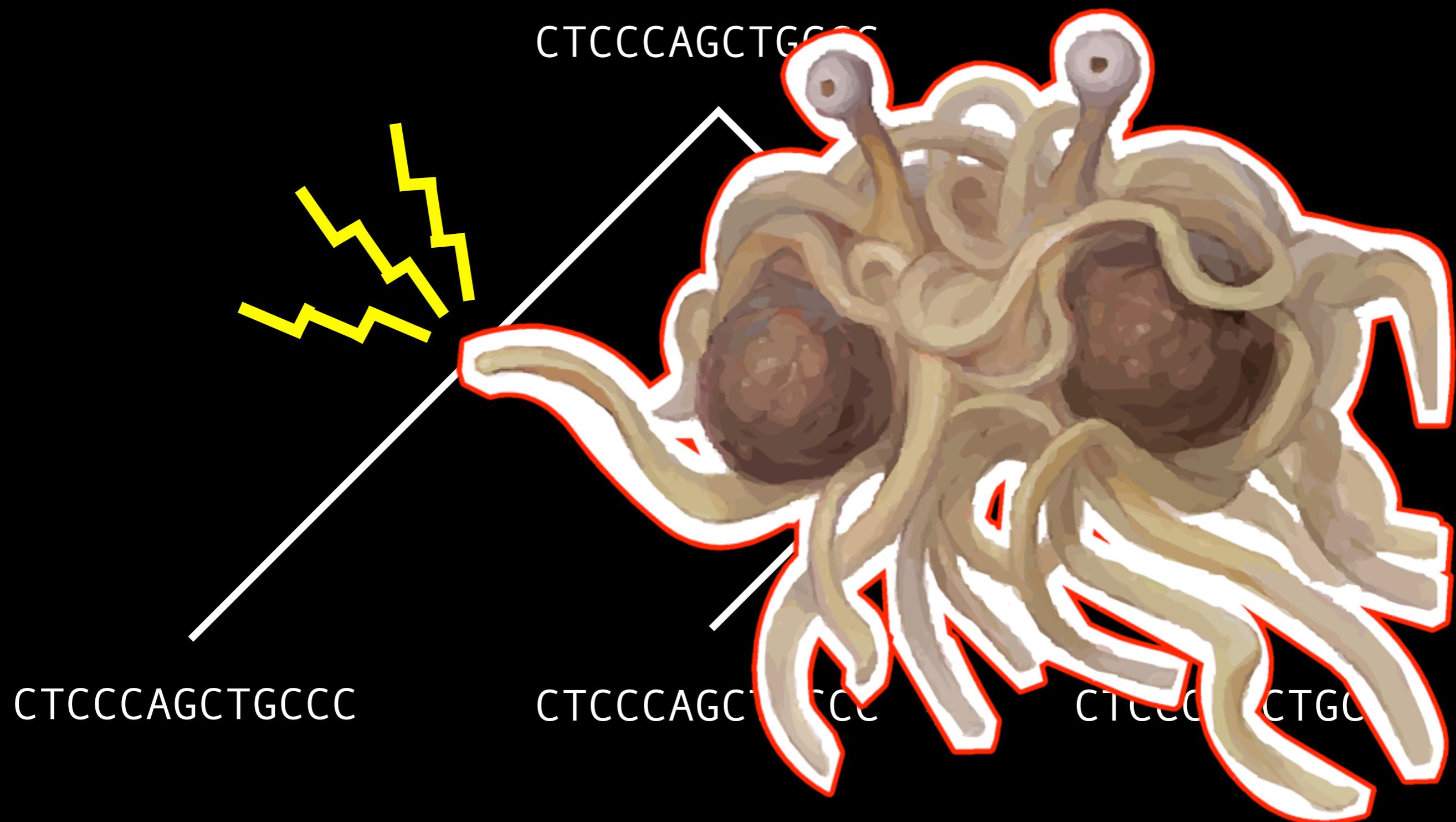


CTCCCCAGCTGCC

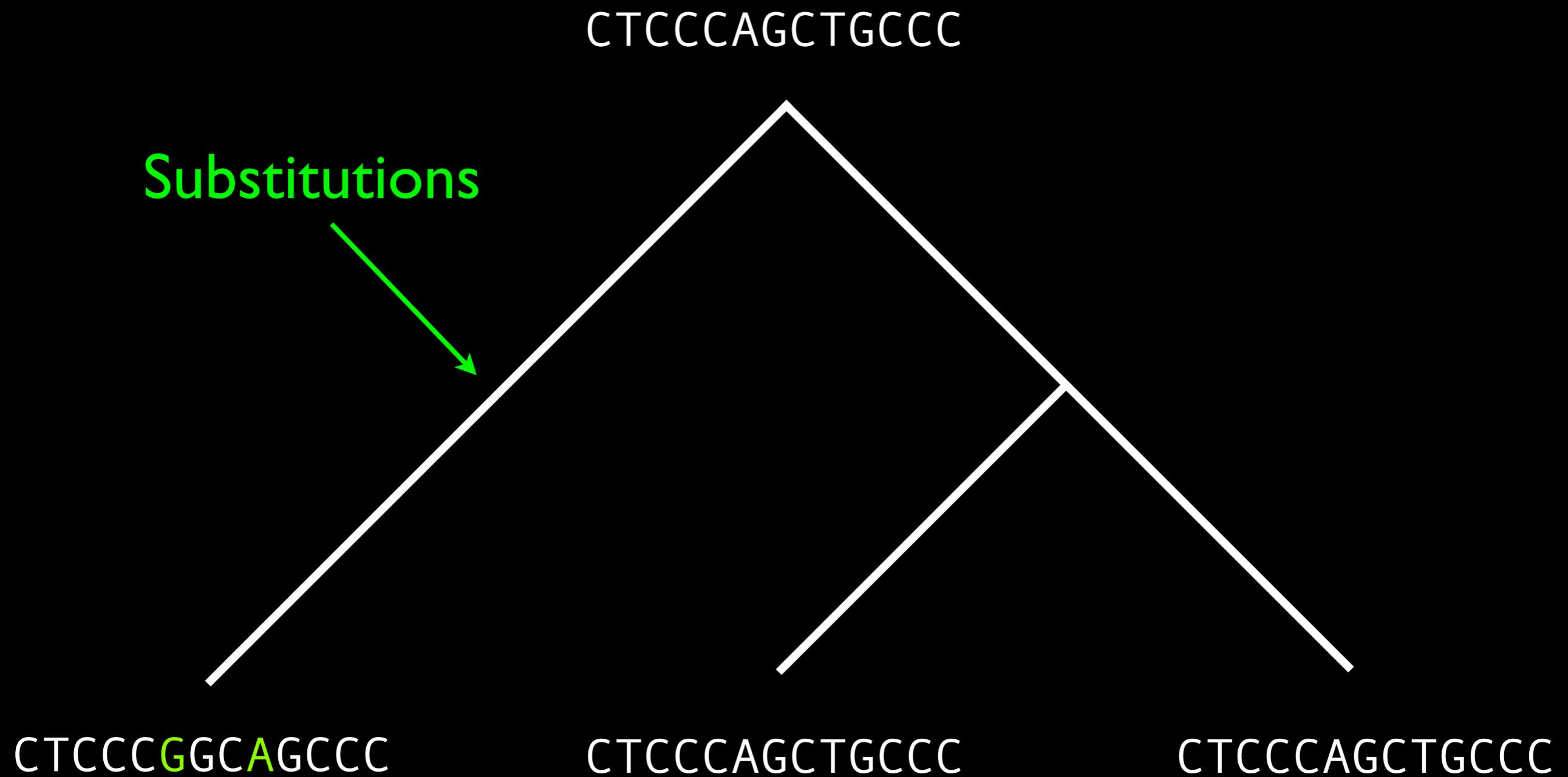
CTCCCCAGCTGCC

CTCCCCAGCTGCC

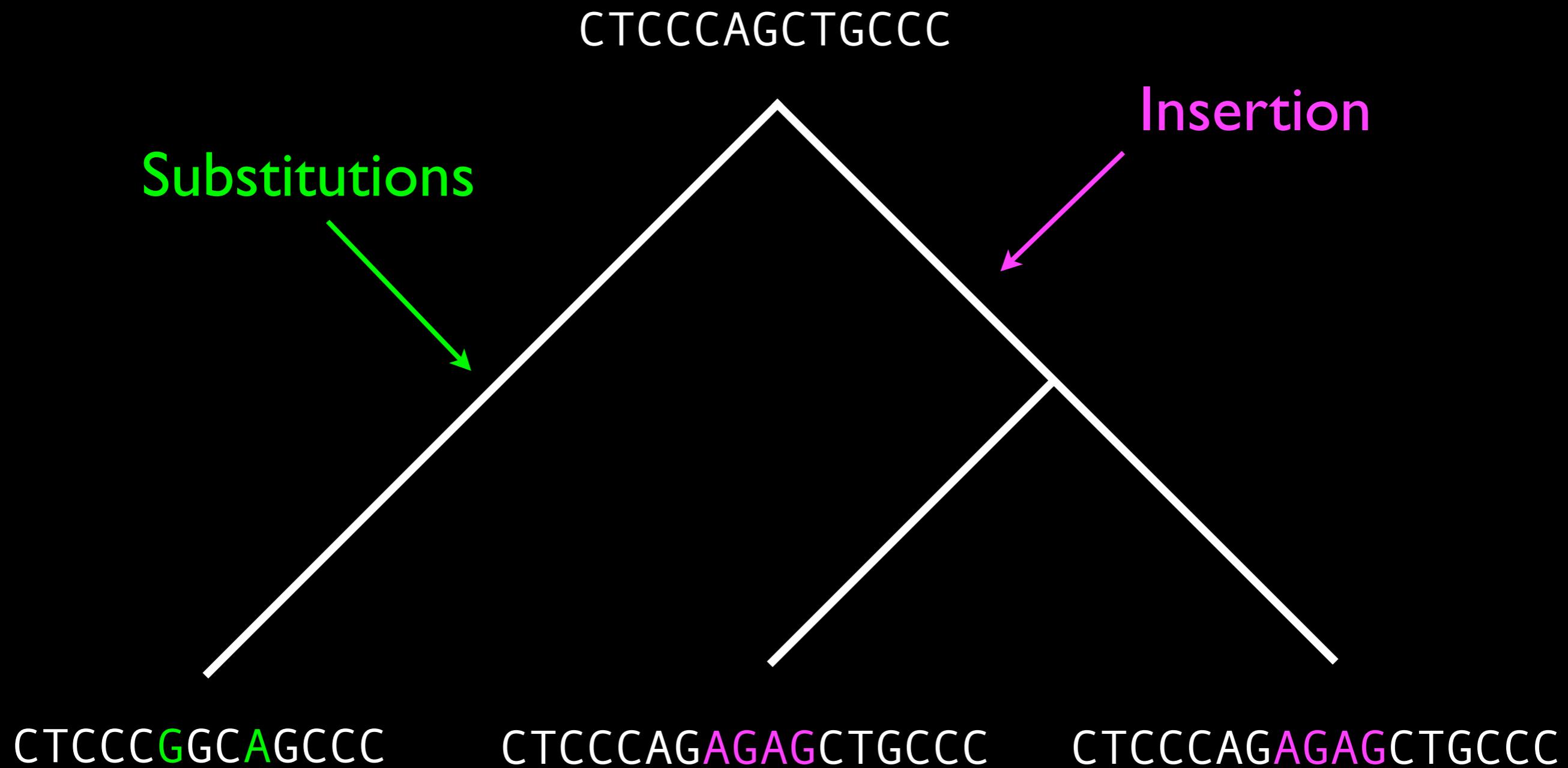
# Evolution of functional elements



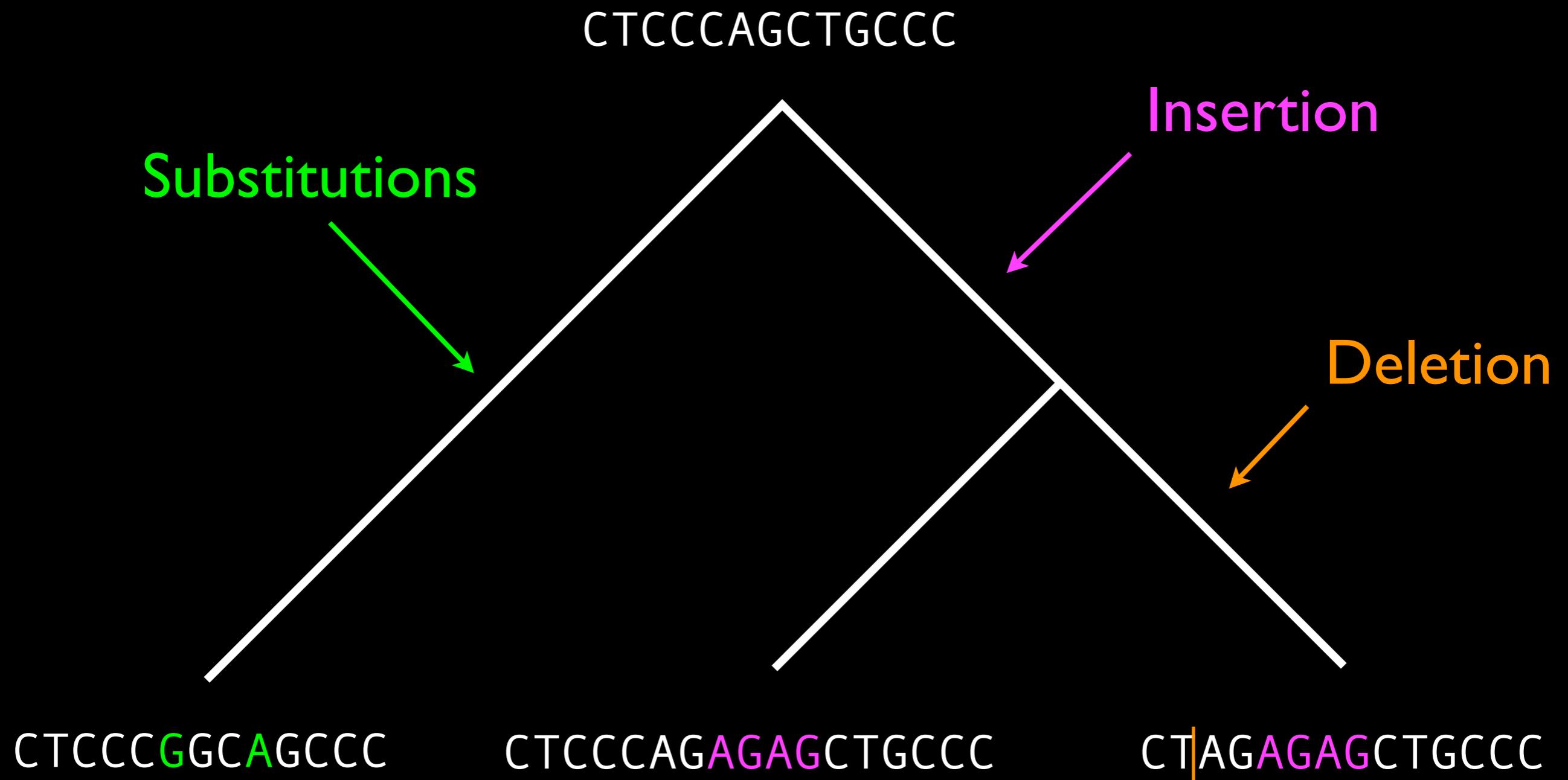
# Evolution of functional elements



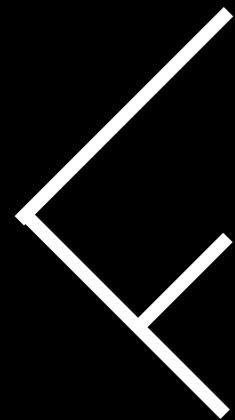
# Evolution of functional elements



# Evolution of functional elements

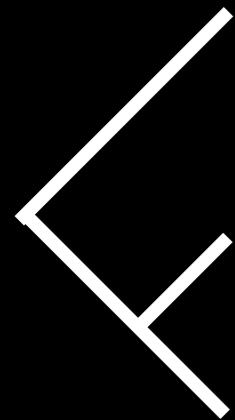


# Sequence alignment



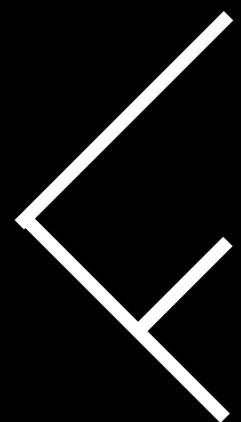
CTCCCCGGCAGCCC  
CTCCCCAGAGAGCTGCC  
CTAGAGAGCTGCC

# Sequence alignment



CTCCCCGG- - -CAGGCC  
CTCCCCAGAGAGCTGCC  
CT- - -AGAGAGCTGCC

# Sequence alignment



Substitutions

CTCCCCGG - - - CAGCCC

CTCCCCAGAGAGCTGCC

CT - - - AGAGAGCTGCC

Deletion

Insertion

# Evolutionary constraint

- After speciation different changes occur in each lineage
- Events occur randomly, but **selection** determines if events are tolerated
- Constraint due to function may prevent certain changes, resulting in a **different pattern of change in functional regions**

# ESPERR

(Evolutionary and Sequence Pattern Extraction  
through Reduced Representation)

# ESPERR: Learning strong and weak signals in genomic sequence alignments to identify functional elements

James Taylor,<sup>1</sup> Svitlana Tyekucheva, David C. King, Ross C. Hardison, Webb Miller, and Francesca Chiaromonte<sup>1</sup>

*Center for Comparative Genomics and Bioinformatics, The Pennsylvania State University, University Park, Pennsylvania 16802, USA*

Genomic sequence signals—such as base composition, presence of particular motifs, or evolutionary constraint—have been used effectively to identify functional elements. However, approaches based only on specific signals known to correlate with function can be quite limiting. When training data are available, application of computational learning algorithms to multispecies alignments has the potential to capture broader and more informative sequence and evolutionary patterns that better characterize a class of elements. However, effective exploitation of patterns in multispecies alignments is impeded by the vast number of possible alignment columns and by a limited understanding of which particular strings of columns may characterize a given class. We have developed a computational method, called ESPERR (*evolutionary and sequence pattern extraction through reduced representations*), which uses training examples to learn encodings of multispecies alignments into reduced forms tailored for the prediction of chosen classes of functional elements. ESPERR produces a greatly improved Regulatory Potential score, which can discriminate regulatory regions from neutral sites with excellent accuracy (~94%). This score captures strong signals (GC content and conservation), as well as subtler signals (with small contributions from many different alignment patterns) that characterize the regulatory elements in our training set. ESPERR is also effective for predicting other classes of functional elements, as we show for DNasel hypersensitive sites and highly conserved regions with developmental enhancer activity. Our software, training data, and genome-wide predictions are available from our Web site (<http://www.bx.psu.edu/projects/esperr>).

[Supplemental material is available online at [www.genome.org](http://www.genome.org).]

Identification of functional elements within genome sequences often relies on specific characteristic signals, typically based on known biological examples. For instance, prediction of protein-coding exons and genes relies on knowledge of the genetic code and splicing signals. These predictions can be improved by incorporating evolutionary information from orthologous regions of other species through sequence alignments. In particular, in-

most ubiquitous promoters, and (3) evolutionary patterns, particularly a high level of interspecies conservation, which should characterize functional regions under purifying selection.

While each of these signals is associated with some *cis*-regulatory modules, all of them have limitations (Tompa et al. 2005). Motif-based approaches can have high specificity, particularly when using a stringent consensus sequence, but when the

# A different approach

- Don't assume a database of known binding motifs
- Don't assume strict conservation of the important sequence signals
- Instead, use alignments of **validated examples** to learn sequence and evolutionary patterns that characterize a class of elements

# Objective

Find a mapping from alignment columns into a smaller alphabet that maintains the “right” information for some classification problem

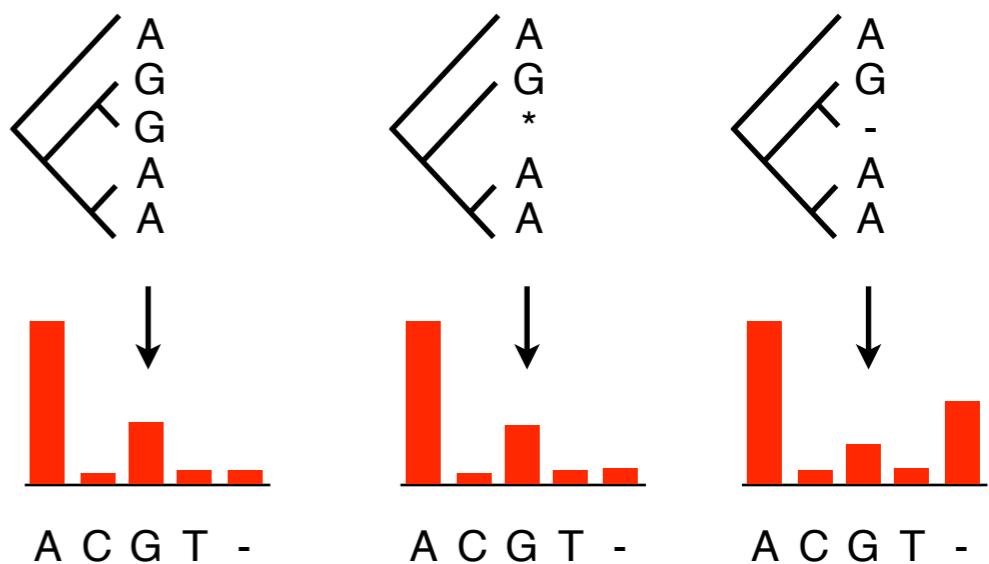
CTCCCCAGCTGCCAGTGCCTGCCTCTTT  
CTCCTAGCTG-CCAGCATCTCCCGTTTT  
CTCCCCAGCTGCCCTGCGCCTCCTCTTTTT



131110213211023211211313333

# Ancestral probability distribution

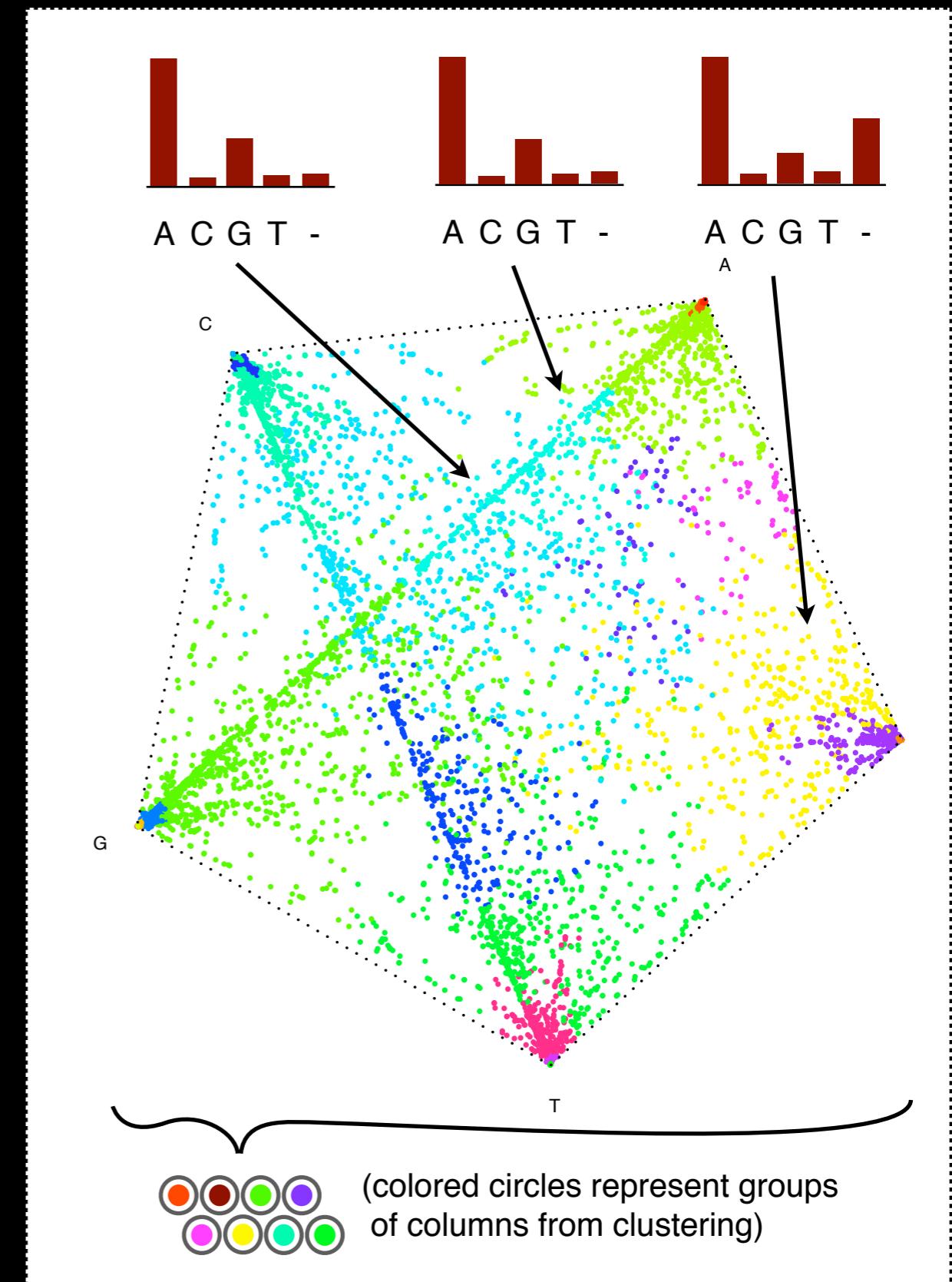
Map each possible column of a multiple alignment to a probability distribution of the nucleotide in that position in the common ancestor.



# Clustering spatially and distributionally

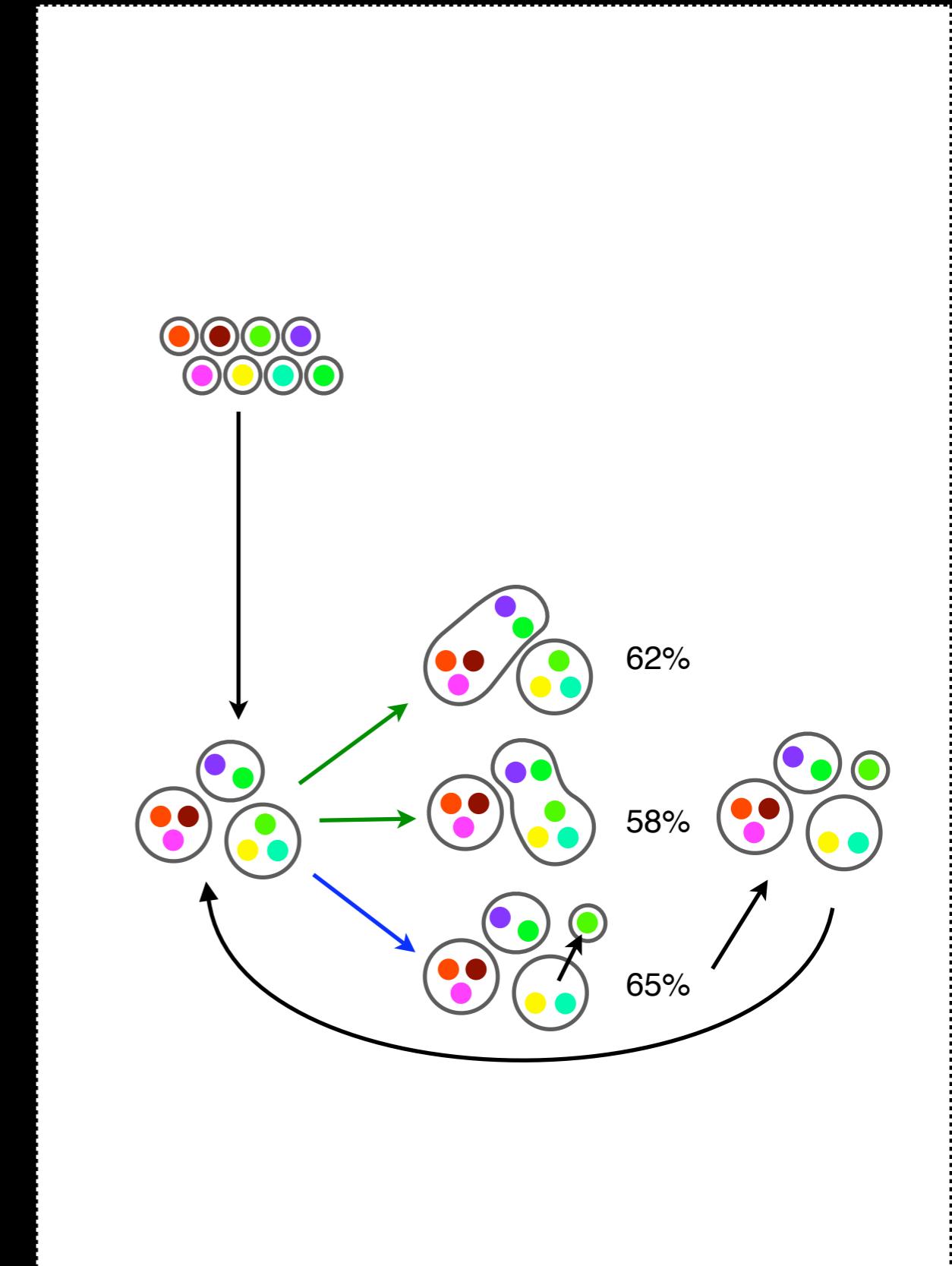
Consider the observed column frequencies as a discrete distribution over the probability simplex, and find a distribution on a smaller number of points that preserves:

- spatial structure: merge only neighboring points
- distributional structure: select mergers that maximize mutual information

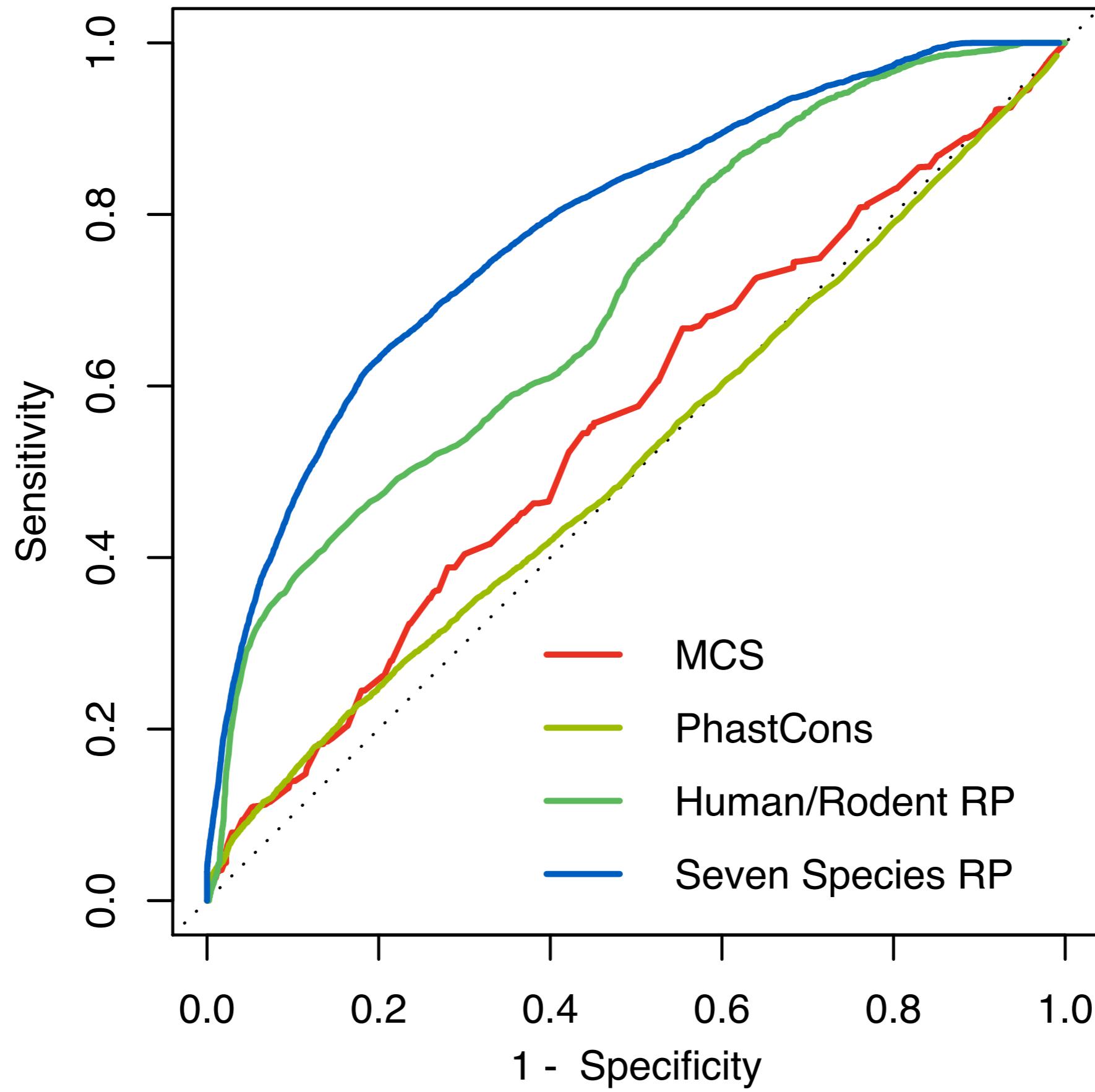


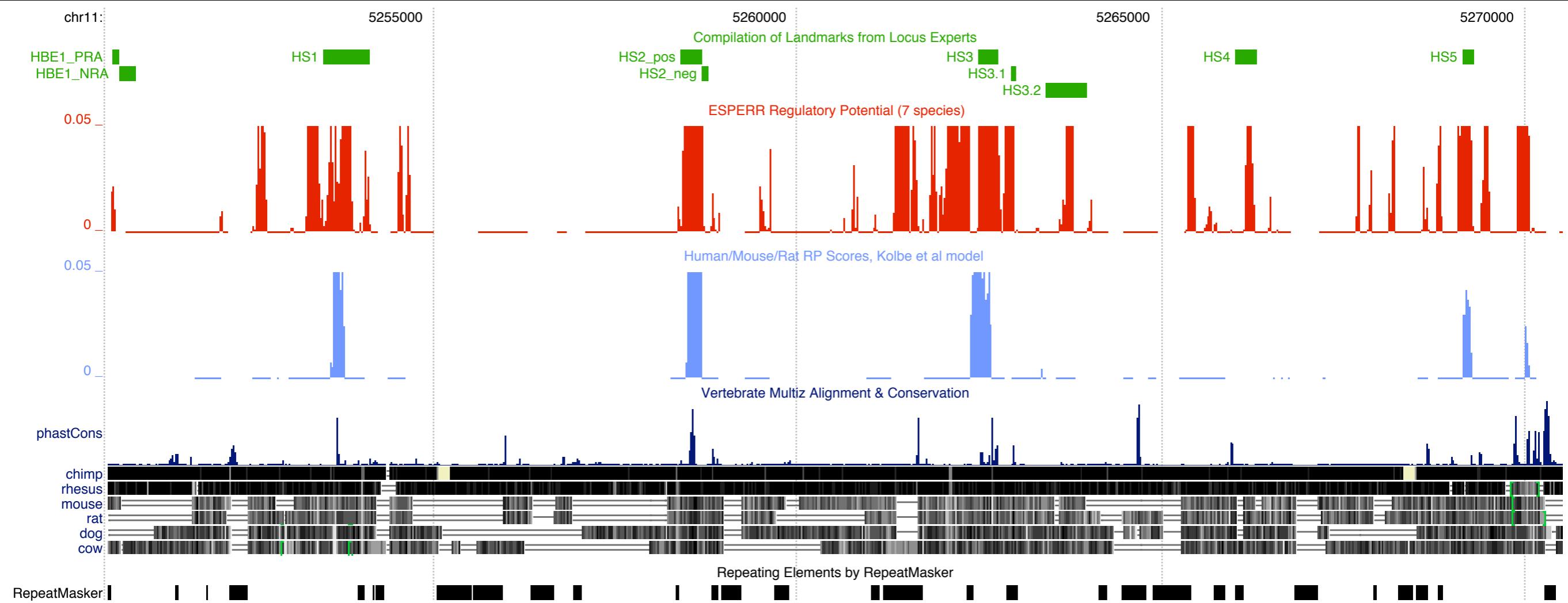
# Searching for encodings

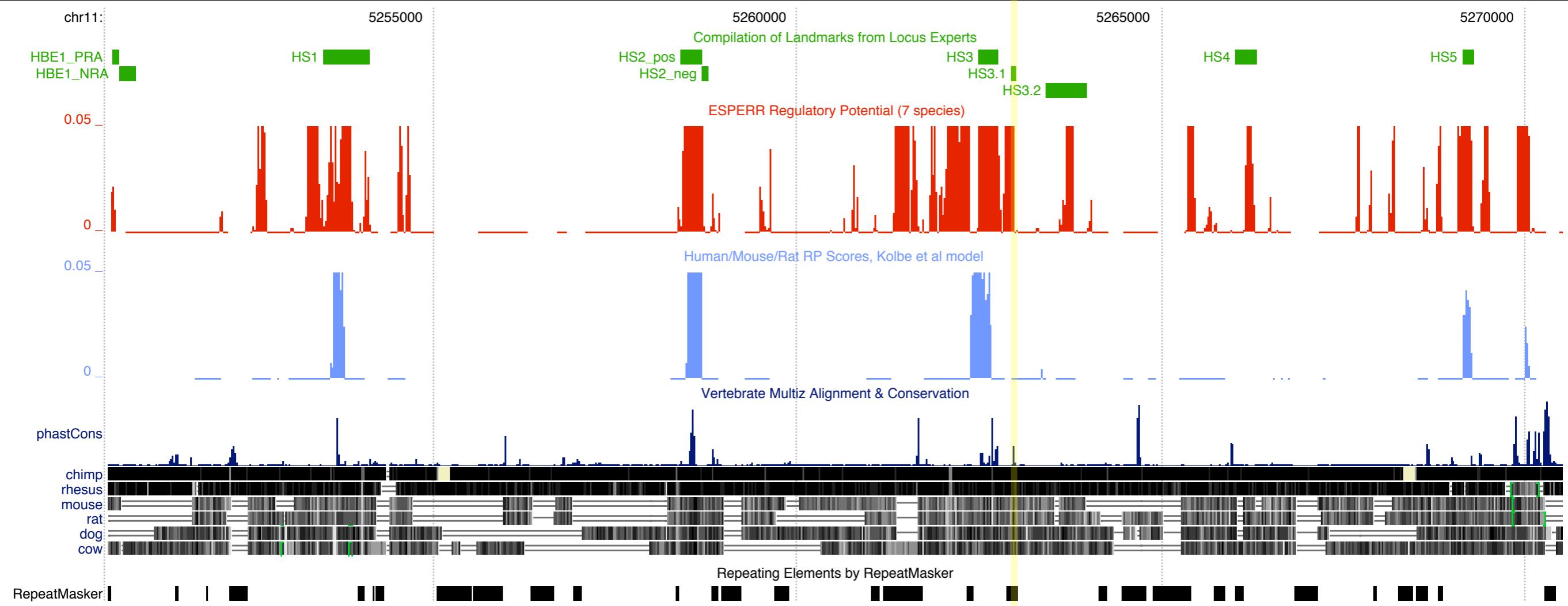
- Random / heuristic search through space of possible encodings

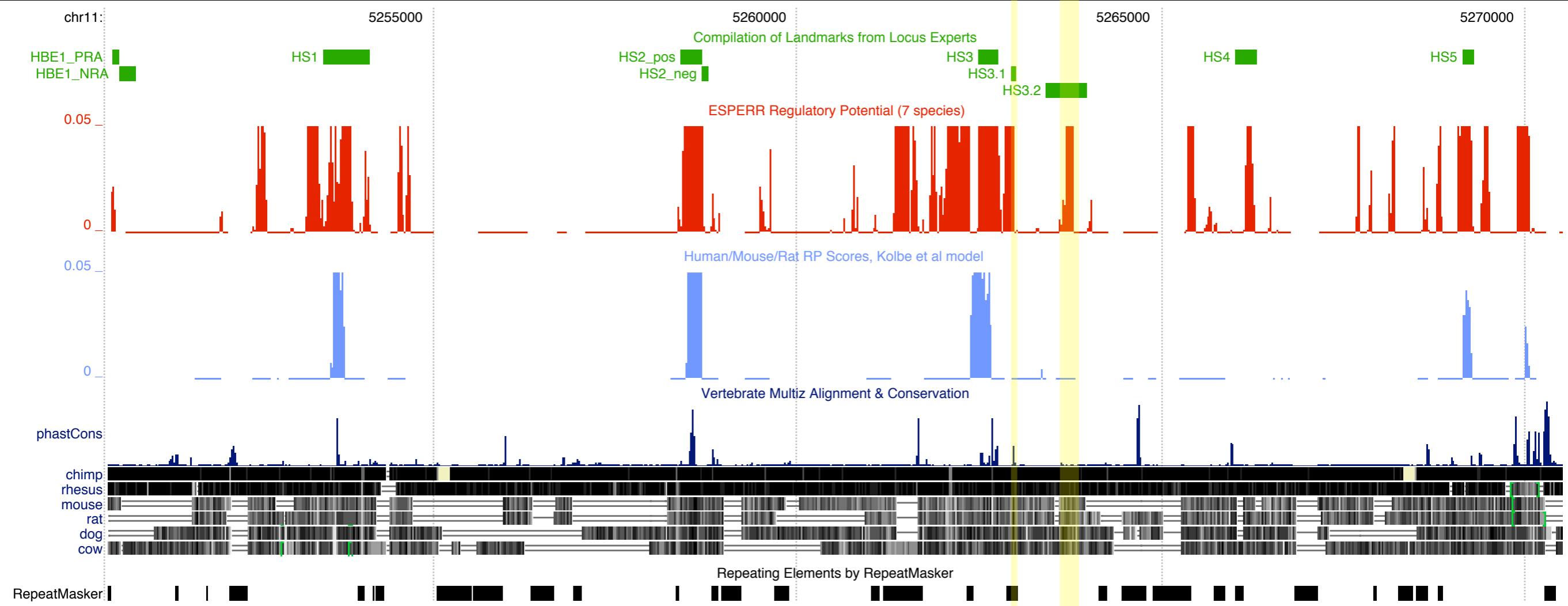


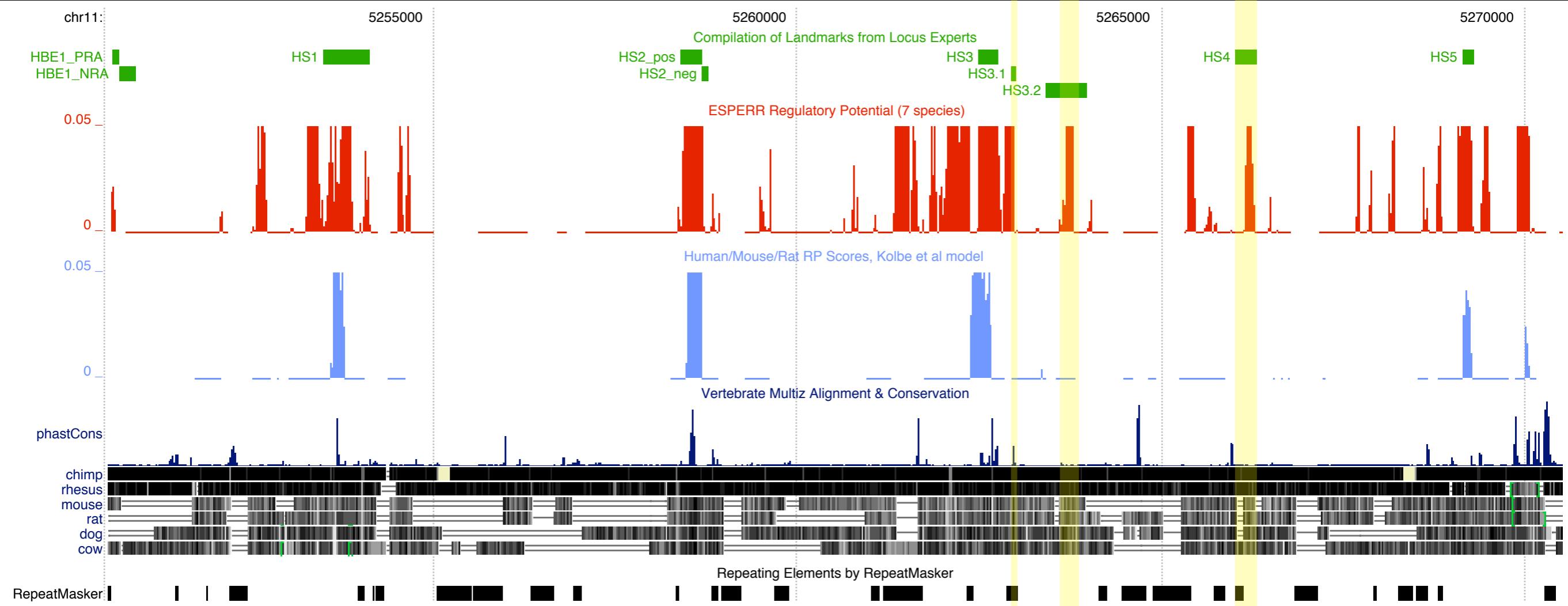
# Some validation







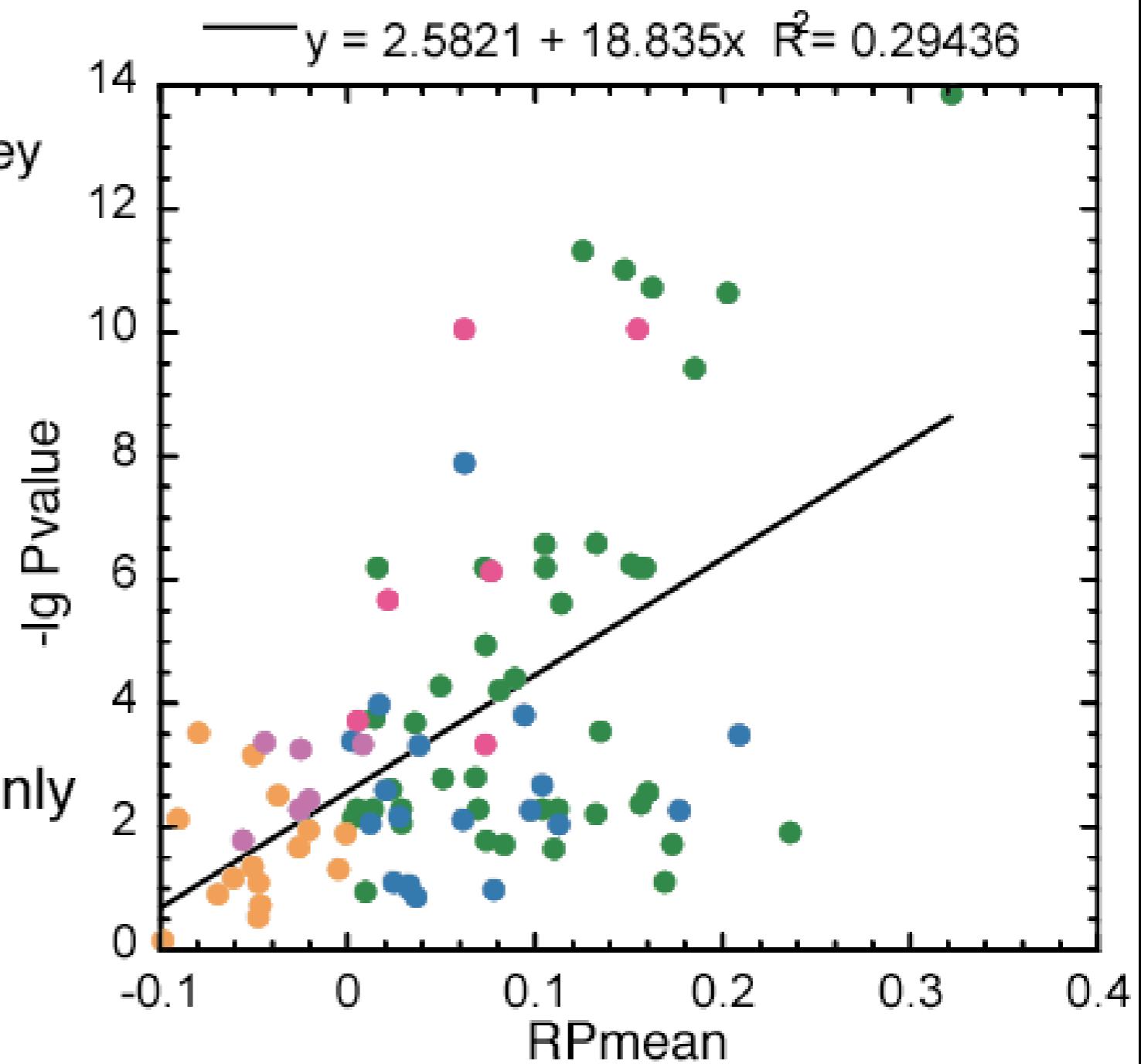




# Enhancer activity correlates with RP score

Pvalue: derived from Mann-Whitney test for tested transient or stable (min for 7 days) assays.

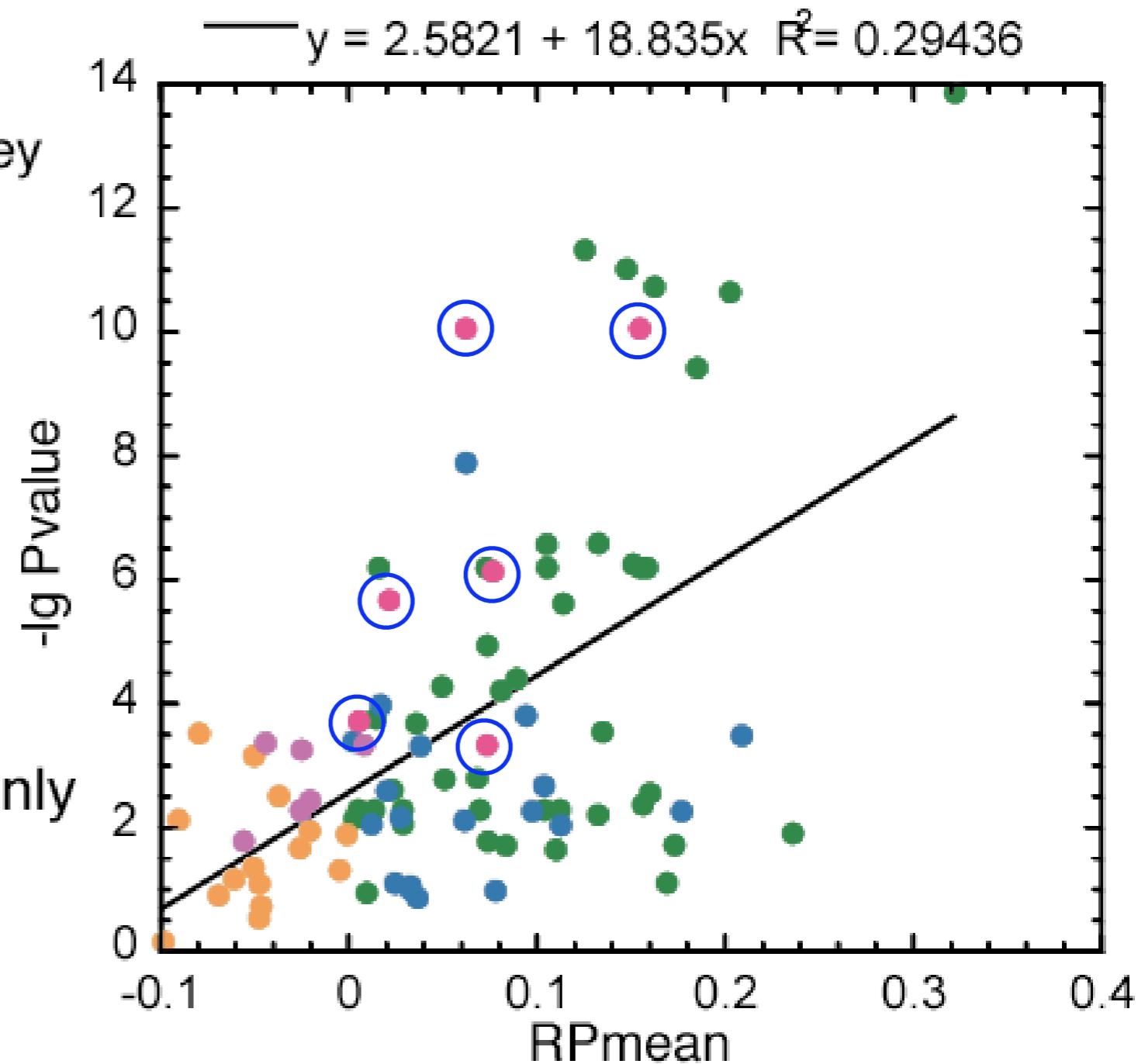
- preCRMcc
- preCRMcnc
- preNeutral
- NegRPw/ccGATA1
- PosiRPw/GATA1mouseonly



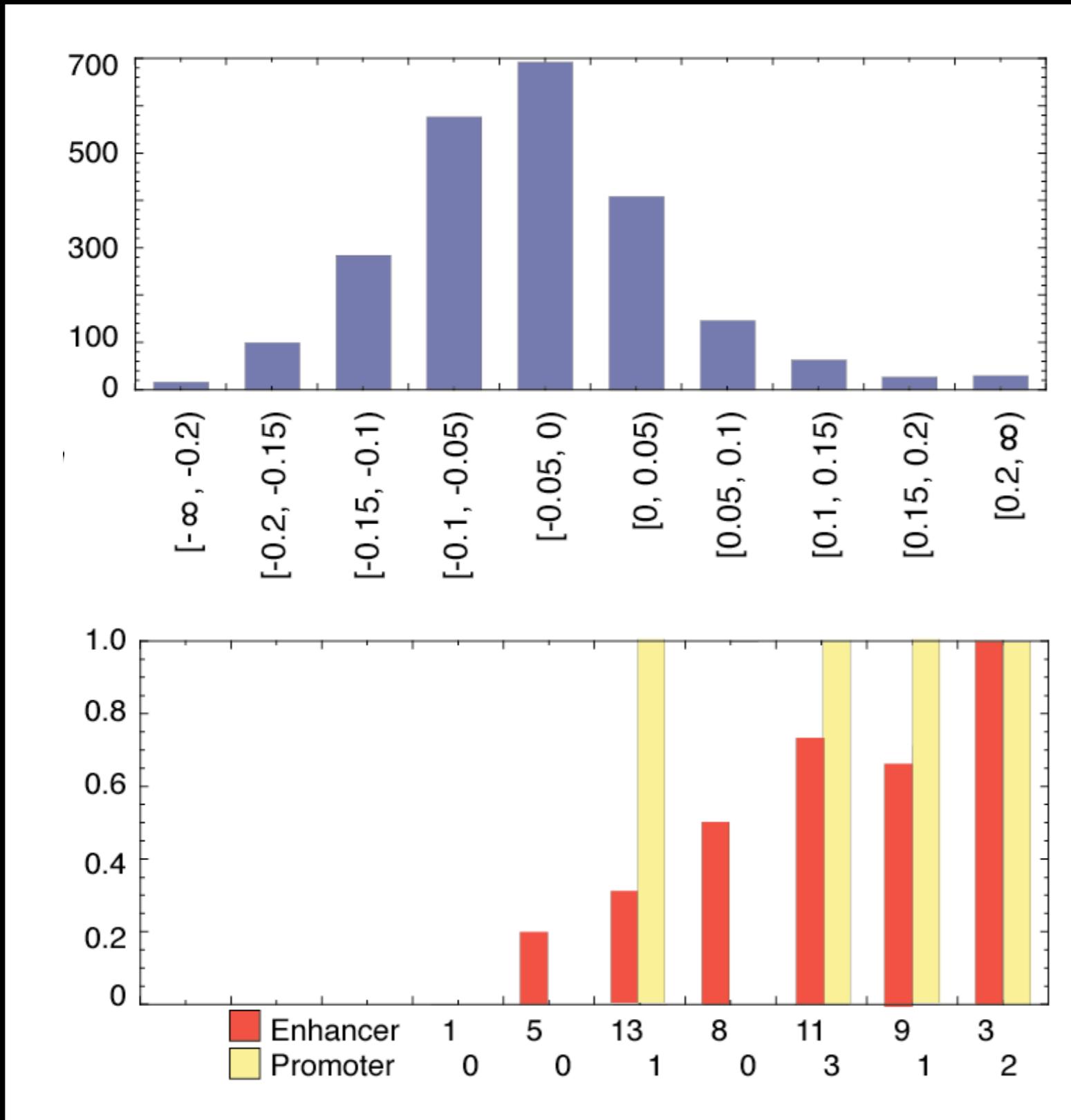
# Enhancer activity correlates with RP score

Pvalue: derived from Mann-Whitney test for tested transient or stable (min for 7 days) assays.

- preCRMcc
- preCRMcnc
- preNeutral
- NegRPw/ccGATA1
- PosiRPw/GATA1mouseonly



# Higher RP scores yield better validation rates



RP distribution across  
loci of interest

Validation rate of test  
loci falling in the RP  
score bin



# Galaxy

(<http://g2.bx.psu.edu>)

# Biological data explosion

- Genome sequences and alignments
- Large scale genotyping and resequencing
- Gene expression and other high throughput functional assays
- “Meta genomics”

# Genomic data management successes

- Data warehouses and query interfaces
  - NCBI
  - UCSC Table Browser
  - Biomart
- Data visualization
  - UCSC Genome browser
  - Ensembl
  - GBrowse

# Many computational methods

- An enormous number of methods / application note papers are being published
  - Usually with some kind of working implementation!
- But what about interfaces? Are these methods accessible to data producers?

# Developing interfaces: Scenario 1

- Developer simply provides scripts or programs with a (usually non-standard) command line interface
  - Experimentalist hires a grad student who hacks it together with Excel / some perl script / manual labor
  - ...or just re-implements the method from scratch with all new bugs

# Developing interfaces: Scenario 2

- Developer builds an interface to their tool that is usable without computational expertise
  - Requires more maintenance, more work to move to new platforms
  - Most of the effort in building interfaces is highly repetitive, substantial waste of developers time
  - Even with a good interface, the tool is not integrated with other tools and datasources, still wasting effort moving data around manually, converting, et cetera.

# Integration

- The primary problem is how do we integrate tools and datasources
  - Give tools a usable and *common* interface
  - Facilitate building complex analysis that use multiple data sources and tools
  - Make it easy to work with large datasets and long running analysis

# Galaxy

# What is **Galaxy**?

- An open-source framework for integrating various computational tools and databases into a cohesive workspace
- A web-based service we (Penn State) provide, integrating many popular tools and resources for comparative genomics
- A completely self-contained Python application for building your own **Galaxy** style sites

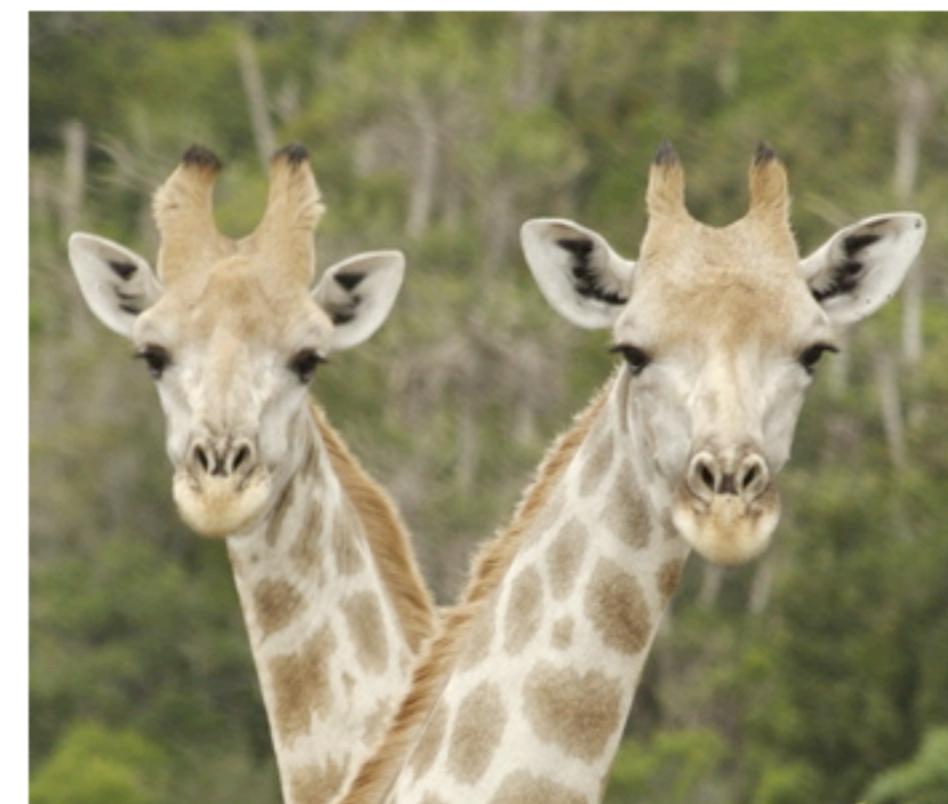
# Galaxy's web user interface

**Tools****Get Data****Get ENCODE Data****ENCODE Tools****Text Manipulation****Filter and Sort****Join, Subtract and Group****Convert Formats****Extract Features****Fetch Sequences****Fetch Alignments****Get Genomic Scores****Operate on Genomic Intervals****Statistics****Graph/Display Data****EMBOSS****HYPHY****Galaxy at ISMB/ECCB2007 in Vienna (July 21-25)**

- July 19 10:00am | [The Galaxy Framework for Computational Biology Tool Integration](#)
- July 25 10:15am Room L | [GALAXY: a simple web application for the analysis of enormous datasets](#)
- July 25 11:10am Room L | [Effortless integration of tools into simple, scalable, multiuser, pythonic framework](#)

**Two Galaxy sessions at the 57th Annual Meeting of the American Society for Human Genetics (October 23-27)!**

The first session will be designed for biomedical researchers and will concentrate on analyses of genomic and disease association data. The second session will be organized as a hands-on software development workshop for bioinformaticians and computational biologists. For more information [click here](#).

**Unsequenced Genomes of the World | July 2007****Giraffe (Giraffa camelopardalis) | Eastern Cape, SAR**

**Why use Galaxy?** It is not a database or a browser. It is an analysis medium that enables multiple tools to be applied to existing data in a simple unified way. Click [here](#) to learn more. Don't like to read manuals? Then [watch](#) a movie (QuickTime required). Here are some reasons why users love it:

**History ([options](#))**[refresh](#) | [collapse all](#)

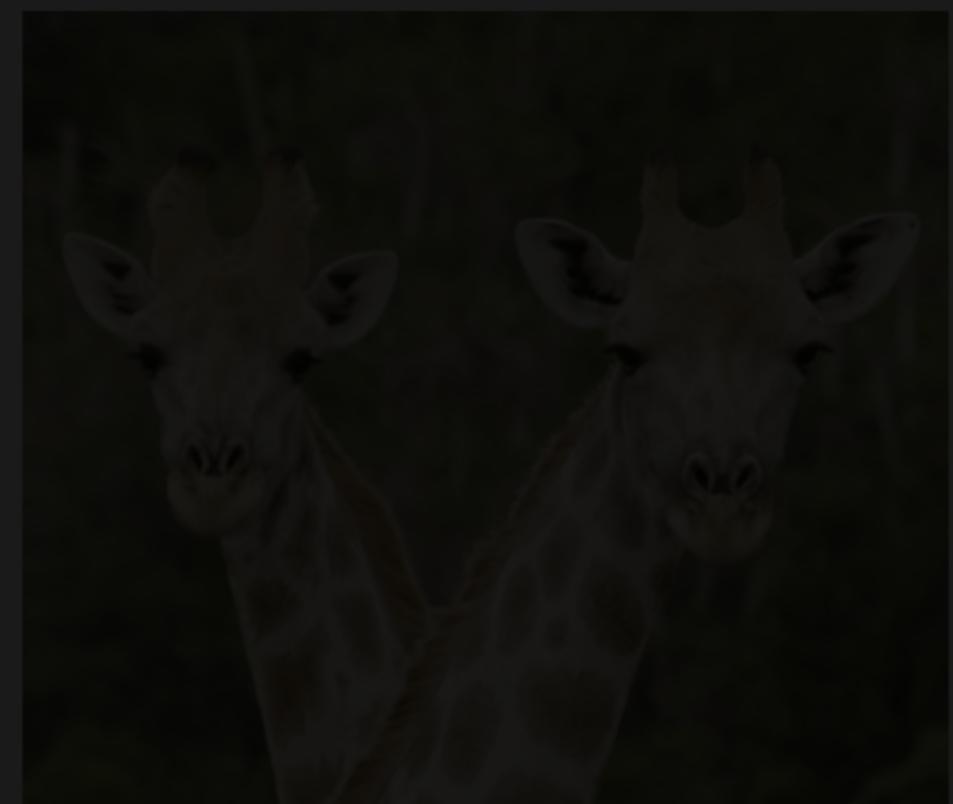
**i** Your history is empty. Click 'Get Data' on the left pane to start

**Tools**[\*\*Get Data\*\*](#)[\*\*Get ENCODE Data\*\*](#)[\*\*ENCODE Tools\*\*](#)[\*\*Text Manipulation\*\*](#)[\*\*Filter and Sort\*\*](#)[\*\*Join, Subtract and Group\*\*](#)[\*\*Convert Formats\*\*](#)[\*\*Extract Features\*\*](#)[\*\*Fetch Sequences\*\*](#)[\*\*Fetch Alignments\*\*](#)[\*\*Get Genomic Scores\*\*](#)[\*\*Operate on Genomic Intervals\*\*](#)[\*\*Statistics\*\*](#)[\*\*Graph/Display Data\*\*](#)[\*\*EMBOSS\*\*](#)[\*\*HYPHY\*\*](#)**Galaxy at ISMB/ECCB2007 in Vienna (July 21-25)**

- July 19 10:00am | [The Galaxy Framework for Computational Biology Tool Integration](#)
- July 25 10:15am Room L | [GALAXY: a simple web application for the analysis of enormous datasets](#)
- July 25 11:10am Room L | [Effortless integration of tools into simple, scalable, multiuser, pythonic framework](#)

**Two Galaxy sessions at the 57th Annual Meeting of the American Society for Human Genetics (October 23-27)!**

The first session will be designed for biomedical researchers and will concentrate on analyses of genomic and disease association data. The second session will be organized as a hands-on software development workshop for bioinformaticians and computational biologists. For more information [click here](#).

**Unsequenced Genomes of the World | July 2007****Giraffe (Giraffa camelopardalis) | Eastern Cape, SAR**

**Why use Galaxy?** It is not a database or a browser. It is an analysis medium that enables multiple tools to be applied to existing data in a simple unified way. Click [here](#) to learn more. Don't like to read manuals? Then [watch](#) a movie (QuickTime required). Here are some reasons why users love it:

**History (options)**[refresh](#) | [collapse all](#)

**i** Your history is empty. Click 'Get Data' on the left pane to start

**Tools****Get Data**

- [Upload File](#) from your computer
- [UCSC Main](#) table browser
- [UCSC Archaea](#) table browser
- [Get Microbial Data](#)
- [BioMart](#) Central server

**Get ENCODE Data****ENCODE Tools****Text Manipulation****Filter and Sort****Join, Subtract and Group****Convert Formats****Extract Features****Fetch Sequences****Fetch Alignments****Get Genomic Scores****Operate on Genomic Intervals****Statistics****Graph/Display Data****EMBOSS****HYPHY****Galaxy at ISMB/ECCB2007 in Vienna (July 21-25)**

- July 19 10:00am | [The Galaxy Framework for Computational Biology Tool Integration](#)
- July 25 10:15am Room L | [GALAXY: a simple web application for the analysis of enormous datasets](#)
- July 25 11:10am Room L | [Effortless integration of tools into simple, scalable, multiuser, pythonic framework](#)

**Two Galaxy sessions at the 57th Annual Meeting of the American Society for Human Genetics (October 23-27)!**

The first session will be designed for biomedical researchers and will concentrate on analyses of genomic and disease association data. The second session will be organized as a hands-on software development workshop for bioinformaticians and computational biologists. For more information [click here](#).

**Unsequenced Genomes of the World | July 2007****Giraffe (Giraffa camelopardalis) | Eastern Cape, SAR**

**Why use Galaxy?** It is not a database or a browser. It is an analysis medium that enables multiple tools to be applied to existing data in a simple unified way. Click [here](#) to learn more. Don't like to read manuals? Then [watch](#) a movie (QuickTime required). Here are some reasons why users love it:

**History (options)**[refresh](#) | [collapse all](#)

**i** Your history is empty. Click 'Get Data' on the left pane to start

**Tools****Get Data****Get ENCODE Data****ENCODE Tools****Text Manipulation****Filter and Sort****Join, Subtract and Group****Convert Formats****Extract Features****Fetch Sequences****Fetch Alignments****Get Genomic Scores****Operate on Genomic Intervals**

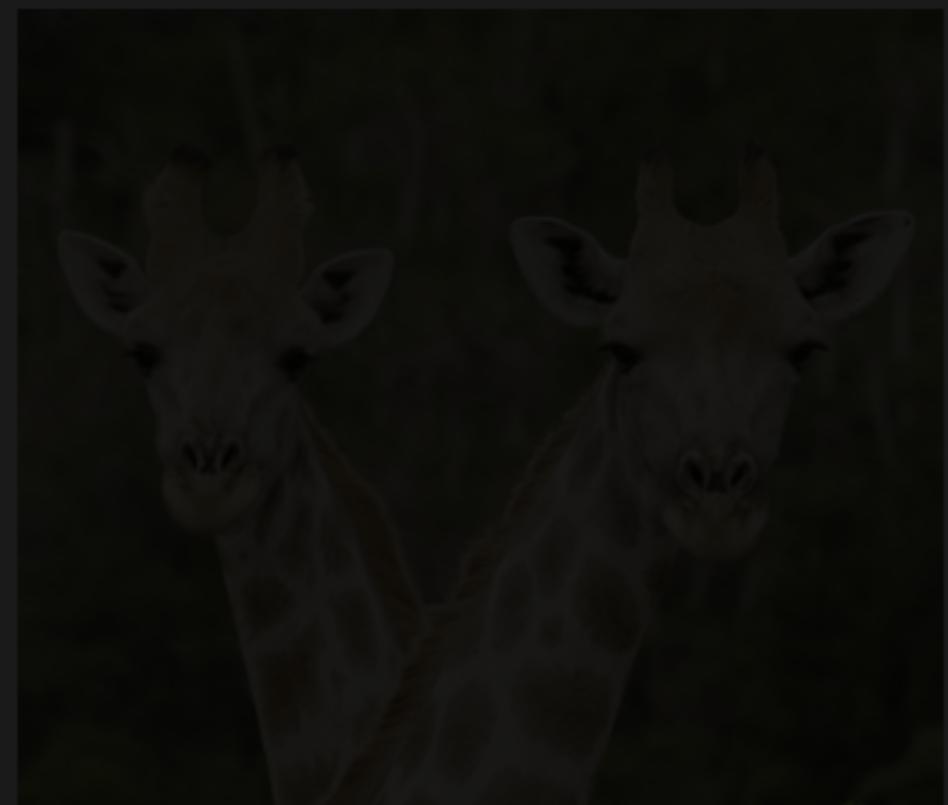
- [Intersect](#) the intervals of two queries
- [Subtract](#) the intervals of two queries
- [Merge](#) the overlapping intervals of a query
- [Concatenate](#) two queries into one query
- [Base Coverage](#) of all intervals
- [Coverage](#) of a set of intervals on second set of intervals
- [Complement](#) intervals of a query
- [Cluster](#) the intervals of a query
- [Join](#) the intervals of two queries side-by-side
- [Get flanks](#) returns flanking region/s for every gene

**Statistics****Graph/Display Data****EMBOSS****HYPHY****Galaxy at ISMB/ECCB2007 in Vienna (July 21-25)**

- July 19 10:00am | [The Galaxy Framework for Computational Biology Tool Integration](#)
- July 25 10:15am Room L | [GALAXY: a simple web application for the analysis of enormous datasets](#)
- July 25 11:10am Room L | [Effortless integration of tools into simple, scalable, multiuser, pythonic framework](#)

**Two Galaxy sessions at the 57th Annual Meeting of the American Society for Human Genetics (October 23-27)!**

The first session will be designed for biomedical researchers and will concentrate on analyses of genomic and disease association data. The second session will be organized as a hands-on software development workshop for bioinformaticians and computational biologists. For more information [click here](#).

**Unsequenced Genomes of the World | July 2007****Giraffe (Giraffa camelopardalis) | Eastern Cape, SAR**

**Why use Galaxy?** It is not a database or a browser. It is an analysis medium that enables multiple tools to be applied to existing data in a simple unified way. Click [here](#) to learn more. Don't like to read manuals? Then [watch](#) a movie (QuickTime required). Here are some reasons why users love it:

**History (options)**[refresh](#) | [collapse all](#)

**i** Your history is empty. Click 'Get Data' on the left pane to start

**Tools****Get Data**

- [Upload File](#) from your computer
- [UCSC Main](#) table browser
- [UCSC Archaea](#) table browser
- [Get Microbial Data](#)
- [BioMart](#) Central server

**Get ENCODE Data****ENCODE Tools****Text Manipulation****Filter and Sort****Join, Subtract and Group****Convert Formats****Extract Features****Fetch Sequences****Fetch Alignments****Get Genomic Scores****Operate on Genomic Intervals****Statistics****Graph/Display Data****EMBOSS****HYPHY****Galaxy at ISMB/ECCB2007 in Vienna (July 21-25)**

- July 19 10:00am | [The Galaxy Framework for Computational Biology Tool Integration](#)
- July 25 10:15am Room L | [GALAXY: a simple web application for the analysis of enormous datasets](#)
- July 25 11:10am Room L | [Effortless integration of tools into simple, scalable, multiuser, pythonic framework](#)

**Two Galaxy sessions at the 57th Annual Meeting of the American Society for Human Genetics (October 23-27)!**

The first session will be designed for biomedical researchers and will concentrate on analyses of genomic and disease association data. The second session will be organized as a hands-on software development workshop for bioinformaticians and computational biologists. For more information [click here](#).

**Unsequenced Genomes of the World | July 2007****Giraffe (Giraffa camelopardalis) | Eastern Cape, SAR**

**Why use Galaxy?** It is not a database or a browser. It is an analysis medium that enables multiple tools to be applied to existing data in a simple unified way. Click [here](#) to learn more. Don't like to read manuals? Then [watch](#) a movie (QuickTime required). Here are some reasons why users love it:

**History (options)**[refresh](#) | [collapse all](#)

**i** Your history is empty. Click 'Get Data' on the left pane to start

**Tools****Get Data**

- [Upload File](#) from your computer
- [UCSC Main](#) table browser
- [UCSC Archaea](#) table browser
- [Get Microbial Data](#)
- [BioMart Central server](#)

**Get ENCODE Data****ENCODE Tools****Text Manipulation****Filter and Sort****Join, Subtract and Group****Convert Formats****Extract Features****Fetch Sequences****Fetch Alignments****Get Genomic Scores****Operate on Genomic Intervals****Statistics****Graph/Display Data****EMBOSS****HYPHY****Upload File**

File:

[Browse...](#)

URL/Text:

Here you may specify a list of URLs (one per line) or paste the contents of a file.

Convert spaces to tabs:

 Yes

Use this option if you are entering intervals by hand.

File Format:

[Auto-detect](#)

BED or Interval? See help below

Genome:

[unspecified \(?\)](#)[Execute](#)**Auto-detect**

The system will attempt to detect AXT, FASTA, Gff, HTML, LAV, Maf, Wiggle, BED and Interval (BED with headers) formats. Other formats will be set to generic text files. If your file is not detected properly as one of the known formats, it most likely means that it has some format problems (e.g., different number of columns on different rows). You can still coerce the system to set your data to the format you think it should be (please send us a note if you see a case when a valid format is not detected).

**BED**

- Tab delimited format (tabular)
- Does not require header line
- Contains 3 required fields:
  - chrom - The name of the chromosome (e.g. chr3, chrY, chr2\_random) or contig (e.g. ctgY1).
  - chromStart - The starting position of the feature in the chromosome or contig. The first base in a chromosome is numbered 0.
  - chromEnd - The ending position of the feature in the chromosome or contig. 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.

**History (options)**[refresh](#) | [collapse all](#)

**i** Your history is empty. Click 'Get Data' on the left pane to start

**Tools****Get Data**

- [Upload File](#) from your computer
- [UCSC Main](#) table browser
- [UCSC Archaea](#) table browser
- [Get Microbial Data](#)
- [BioMart](#) Central server

**Get ENCODE Data****ENCODE Tools****Text Manipulation****Filter and Sort****Join, Subtract and Group****Convert Formats****Extract Features****Fetch Sequences****Fetch Alignments****Get Genomic Scores****Operate on Genomic Intervals****Statistics****Graph/Display Data****EMBOSS****HYPHY****Upload File**

File:

[Browse...](#)

URL/Text:

```
chr1 147971133 148471133  
ENr231  
chr2 51570355 52070355  
ENr112  
chr2 118010803 118510803  
ENr121
```



Here you may specify a list of URLs (one per line) or paste the contents of a file.

Convert spaces to tabs:



Use this option if you are entering intervals by hand.

File Format:

[Auto-detect](#)

BED or Interval? See help below

Genome:

[Human May 2004 \(hg17\)](#)[Execute](#)**Auto-detect**

The system will attempt to detect AXT, FASTA, Gff, HTML, LAV, Maf, Wiggle, BED and Interval (BED with headers) formats. Other formats will be set to generic text files. If your file is not detected properly as one of the known formats, it most likely means that it has some format problems (e.g., different number of columns on different rows). You can still coerce the system to set your data to the format you think it should be (please send us a note if you see a case when a valid format is not detected).

**BED**

- Tab delimited format (tabular)
- Does not require header line
- Contains 3 required fields:
  - chrom - The name of the chromosome (e.g. chr3, chrY, chr2\_random) or contig (e.g. ctgY1).
  - chromStart - The starting position of the feature in the chromosome or contig. The first base in a chromosome is numbered 0.
  - chromEnd - The ending position of the feature in the chromosome or contig. 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.

**History (options)**[refresh](#) | [collapse all](#)

**i** Your history is empty. Click 'Get Data' on the left pane to start



The following job has been successfully added to the queue:

**3: Pasted Entry**

You can check the status of queued jobs and view the resulting data by refreshing the **History** pane. When the job has been run the status will change from 'running' to 'finished' if completed successfully or 'error' if problems were encountered.

**Tools****Get Data**

- [Upload File](#) from your computer
- [UCSC Main](#) table browser
- [UCSC Archaea](#) table browser
- [Get Microbial Data](#)
- [BioMart Central](#) server

**Get ENCODE Data****ENCODE Tools****Text Manipulation****Filter and Sort****Join, Subtract and Group****Convert Formats****Extract Features****Fetch Sequences****Fetch Alignments****Get Genomic Scores****Operate on Genomic Intervals****Statistics****Graph/Display Data****EMBOSS****HYPHY**

The following job has been successfully added to the queue:

**3: Pasted Entry**

You can check the status of queued jobs and view the resulting data by refreshing the **History** pane. When the job has been run the status will change from 'running' to 'finished' if completed successfully or 'error' if problems were encountered.

**History (options)**

[refresh](#) | [collapse all](#)

**3: Pasted Entry**

**Tools****Get Data**

- [Upload File](#) from your computer
- [UCSC Main](#) table browser
- [UCSC Archaea](#) table browser
- [Get Microbial Data](#)
- [BioMart Central](#) server

**Get ENCODE Data****ENCODE Tools****Text Manipulation****Filter and Sort****Join, Subtract and Group****Convert Formats****Extract Features****Fetch Sequences****Fetch Alignments****Get Genomic Scores****Operate on Genomic Intervals****Statistics****Graph/Display Data****EMBOSS****HYPHY**

The following job has been successfully added to the queue:

**3: Pasted Entry**

You can check the status of queued jobs and view the resulting data by refreshing the **History** pane. When the job has been run the status will change from 'running' to 'finished' if completed successfully or 'error' if problems were encountered.

**History (options)**

[refresh](#) | [collapse all](#)

**3: Pasted Entry**

44 regions, format: bed, database: hg17

Info: pasted entry

[save](#) | [display at UCSC main test](#)

1 2 3 4

chr1	147971133	148471133	ENr231
chr2	51570355	52070355	ENr112
chr2	118010803	118510803	ENr121
chr2	220102850	220602850	ENr331
chr2	234273824	234773888	ENr131
chr4	118604258	119104258	ENr113

**Tools****Get Data**

- [Upload File](#) from your computer
- [UCSC Main](#) table browser
- [UCSC Archaea](#) table browser
- [Get Microbial Data](#)
- [BioMart Central](#) server

**Get ENCODE Data****ENCODE Tools****Text Manipulation****Filter and Sort****Join, Subtract and Group****Convert Formats****Extract Features****Fetch Sequences****Fetch Alignments****Get Genomic Scores****Operate on Genomic Intervals****Statistics****Graph/Display Data****EMBOSS****HYPHY**

The following job has been successfully added to the queue:

**3: Pasted Entry**

You can check the status of queued jobs and view the resulting data by refreshing the **History** pane. When the job has been run the status will change from 'running' to 'finished' if completed successfully or 'error' if problems were encountered.

**History (options)**

[refresh](#) | [collapse all](#)

**3: Pasted Entry**

**Tools****Get Data**

- [Upload File](#) from your computer
- [UCSC Main](#) table browser
- [UCSC Archaea](#) table browser
- [Get Microbial Data](#)
- [BioMart Central](#) server

**Get ENCODE Data****ENCODE Tools****Text Manipulation****Filter and Sort****Join, Subtract and Group****Convert Formats****Extract Features****Fetch Sequences****Fetch Alignments****Get Genomic Scores****Operate on Genomic Intervals****Statistics****Graph/Display Data****EMBOSS****HYPHY**

The following job has been successfully added to the queue:

**4: UCSC Main**

You can check the status of queued jobs and view the resulting data by refreshing the **History** pane. When the job has been run the status will change from 'running' to 'finished' if completed successfully or 'error' if problems were encountered.

**History (options)**

[refreshing in 9 sec](#) | [collapse all](#)

**4: UCSC Main****3: Pasted Entry**

[Tools](#)[Get Data](#)[Get ENCODE Data](#)[ENCODE Tools](#)[Text Manipulation](#)[Filter and Sort](#)[Join, Subtract and Group](#)[Convert Formats](#)[Extract Features](#)[Fetch Sequences](#)[Fetch Alignments](#)[Get Genomic Scores](#)[Operate on Genomic Intervals](#)

- [Intersect](#) the intervals of two queries
- [Subtract](#) the intervals of two queries
- [Merge](#) the overlapping intervals of a query
- [Concatenate](#) two queries into one query
- [Base Coverage](#) of all intervals
- [Coverage](#) of a set of intervals on second set of intervals
- [Complement](#) intervals of a query
- [Cluster](#) the intervals of a query
- [Join](#) the intervals of two queries side-by-side
- [Get flanks](#) returns flanking region/s for every gene

[Statistics](#)[Graph/Display Data](#)[EMBOSS](#)[HYPHY](#)

The following job has been successfully added to the queue:

**5: Intersect on data 3 and data 4**

You can check the status of queued jobs and view the resulting data by refreshing the [History](#) pane. When the job has been run the status will change from 'running' to 'finished' if completed successfully or 'error' if problems were encountered.

**History (options)**[refreshing in 8 sec](#) | [collapse all](#)**5: Intersect on data 3 and data 4**[edit](#) / [delete](#)**4: UCSC Main on Human: knownGene (genome)**[edit](#) / [delete](#)**3: Pasted Entry**[edit](#) / [delete](#)

# Integrating tools into Galaxy

## Tools

### Get Data

- [Upload File](#) from your computer
- [UCSC Main](#) table browser
- [UCSC Archaea](#) table browser
- [Get Microbial Data](#)
- [BioMart Central server](#)

### Get ENCODE Data

#### ENCODE Tools

#### Text Manipulation

#### Filter and Sort

#### Join, Subtract and Group

#### Convert Formats

#### Extract Features

#### Fetch Sequences

#### Fetch Alignments

#### Get Genomic Scores

#### Operate on Genomic Intervals

#### Statistics

#### Graph/Display Data

#### EMBOSS

#### HYPHY

## Tools

### Get Data

- [Upload File](#) from your computer
- [UCSC Main](#) table browser
- [UCSC Archaea](#) table browser
- [Get Microbial Data](#)
- [BioMart](#) Central server

### Get ENCODE Data

#### ENCODE Tools

#### Text Manipulation

#### Filter and Sort

#### Join, Subtract and Group

#### Convert Formats

#### Extract Features

#### Fetch Sequences

#### Fetch Alignments

#### Get Genomic Scores

#### Operate on Genomic Intervals

#### Statistics

#### Graph/Display Data

#### EMBOSS

#### HYPHY

tool\_conf.xml

```
1 <?xml version="1.0"?>
2 <toolbox>
3   <section name="Get Data" id="getext">
4     <tool file="data_source/upload.xml"/>
5     <tool file="data_source/ucsc_tablebrowser.xml" />
6     <tool file="data_source/ucsc_tablebrowser_archaea.xml" />
7     <tool file="data_source/microbial_import.xml" />
8     <tool file="data_source/biomart.xml" />
9   </section>
10  <section name="Get ENCODE Data" id="encode">
11    <tool file="data_source/encode_import_chromatin_and_chromosomes.xml"/>
12    <tool file="data_source/encode_import_genes_and_transcripts.xml"/>
13    <tool file="data_source/encode_import_transcription_regulation.xml"/>
14    <tool file="data_source/encode_import_all_latest_datasets.xml" />
15    <tool file="data_source/encode_import_gencode.xml" />
16  </section>
17  <section name="ENCODE Tools" id="EncodeTools">
18    <tool file="encode/gencode_partition.xml" />
19  </section>
20  <section name="Text Manipulation" id="textutil">
21    <tool file="filters/fixedValueColumn.xml" />
22    <tool file="stats/column_maker.xml" />
23    <tool file="filters/catWrapper.xml" />
24    <tool file="filters/condense_characters.xml" />
25    <tool file="filters/convert_characters.xml" />
26    <tool file="filters/CreateInterval.xml" />
27    <tool file="filters/cutWrapper.xml" />
28    <tool file="filters/pasteWrapper.xml" />
29    <tool file="filters/remove_beginning.xml" />
30    <tool file="filters/headWrapper.xml" />
31    <tool file="filters/tailWrapper.xml" />
32  </section>
33  <section name="Filter and Sort" id="filter">
34    <tool file="stats/filtering.xml" />
35    <tool file="filters/sorter.xml" />
36    <tool file="filters/grep.xml" />
37  </section>
38  <section name="Join, Subtract and Group" id="group">
39    <tool file="filters/joiner.xml" />
40    <tool file="filters/compare.xml"/>
41    <tool file="new_operations/subtract_query.xml"/>
42    <tool file="stats/grouping.xml" />
43  </section>
```

Line: 12 Column: 71 XML Soft Tabs: 2 -

## Tools

### Get Data

- [Upload File](#) from your computer
- [UCSC Main](#) table browser
- [UCSC Archaea](#) table browser
- [Get Microbial Data](#)
- [BioMart](#) Central server

### Get ENCODE Data

### ENCODE Tools

### Text Manipulation

### Filter and Sort

### Join, Subtract and Group

### Convert Formats

### Extract Features

### Fetch Sequences

### Fetch Alignments

### Get Genomic Scores

### Operate on Genomic Intervals

### Statistics

### Graph/Display Data

### EMBOSS

### HYPHY

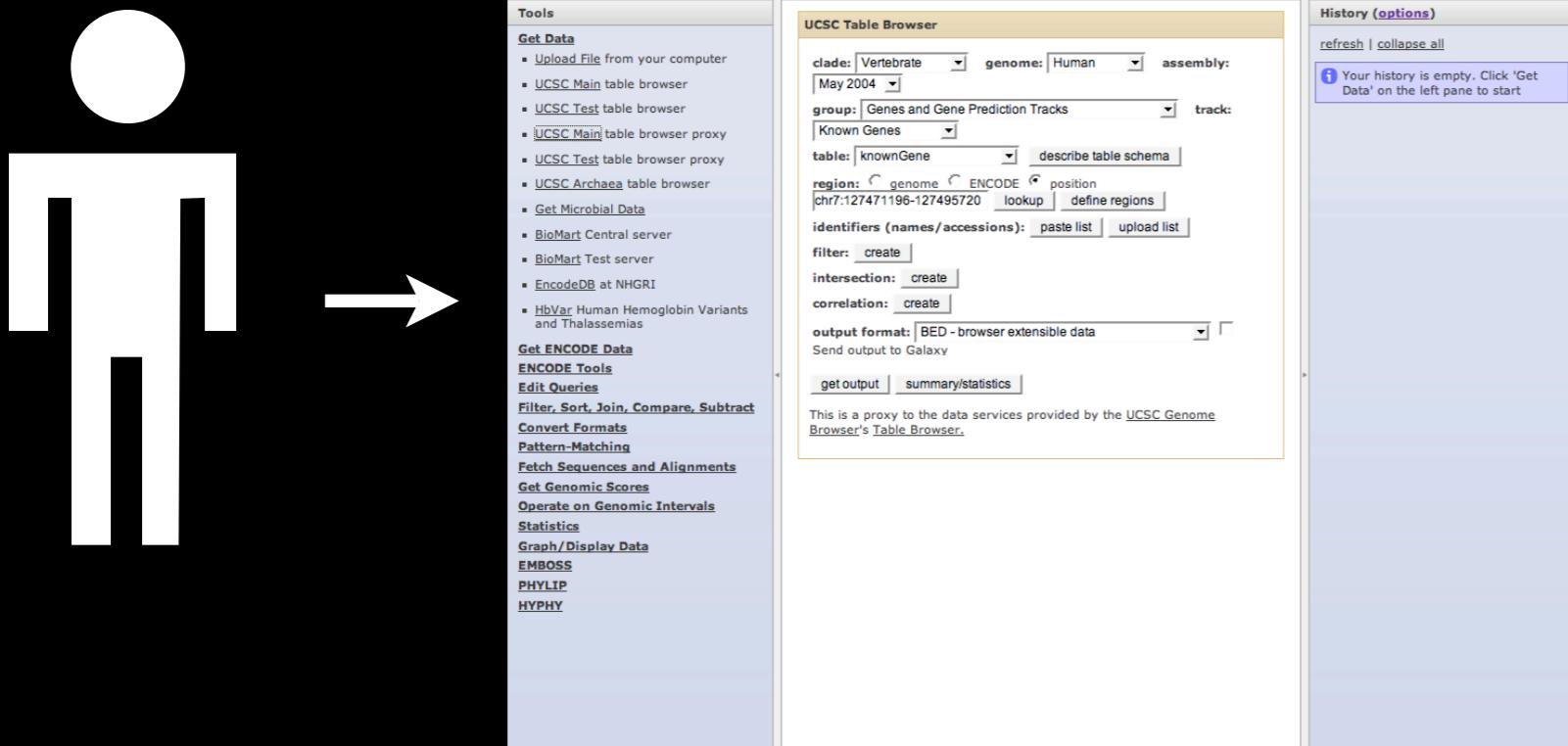
tool\_conf.xml

```
1 <?xml version="1.0"?>
2 <toolbox>
3   <section name="Get Data" id="getext">
4     <tool file="data_source/upload.xml"/>
5     <tool file="data_source/ucsc_tablebrowser.xml" />
6     <tool file="data_source/ucsc_tablebrowser_archaea.xml" />
7     <tool file="data_source/microbial_import.xml" />
8     <tool file="data_source/biomart.xml" />
9   </section>
10  <section name="Get ENCODE Data" id="encode">
11    <tool file="data_source/encode_import_chromatin_and_chromosomes.xml"/>
12    <tool file="data_source/encode_import_genes_and_transcripts.xml"/>
13    <tool file="data_source/encode_import_transcription_regulation.xml"/>
14    <tool file="data_source/encode_import_all_latest_datasets.xml" />
15    <tool file="data_source/encode_import_gencode.xml" />
16  </section>
17  <section name="ENCODE Tools" id="EncodeTools">
18    <tool file="encode/gencode_partition.xml" />
19  </section>
20  <section name="Text Manipulation" id="textutil">
21    <tool file="filters/fixedValueColumn.xml" />
22    <tool file="stats/column_maker.xml" />
23    <tool file="filters/catWrapper.xml" />
24    <tool file="filters/condense_characters.xml" />
25    <tool file="filters/convert_characters.xml" />
26    <tool file="filters/CreateInterval.xml" />
27    <tool file="filters/cutWrapper.xml" />
28    <tool file="filters/pasteWrapper.xml" />
29    <tool file="filters/remove_beginning.xml" />
30    <tool file="filters/headWrapper.xml" />
31    <tool file="filters/tailWrapper.xml" />
32  </section>
33  <section name="Filter and Sort" id="filter">
34    <tool file="stats/filtering.xml" />
35    <tool file="filters/sorter.xml" />
36    <tool file="filters/grep.xml" />
37  </section>
38  <section name="Join, Subtract and Group" id="group">
39    <tool file="filters/joiner.xml" />
40    <tool file="filters/compare.xml"/>
41    <tool file="new_operations/subtract_query.xml"/>
42    <tool file="stats/grouping.xml" />
43  </section>
```

Line: 12 Column: 71 XML Soft Tabs: 2 —

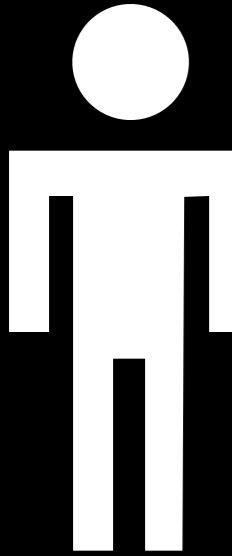
How **Galaxy** integrates existing  
web-based tools

# Proxy based tools



User makes request to Galaxy

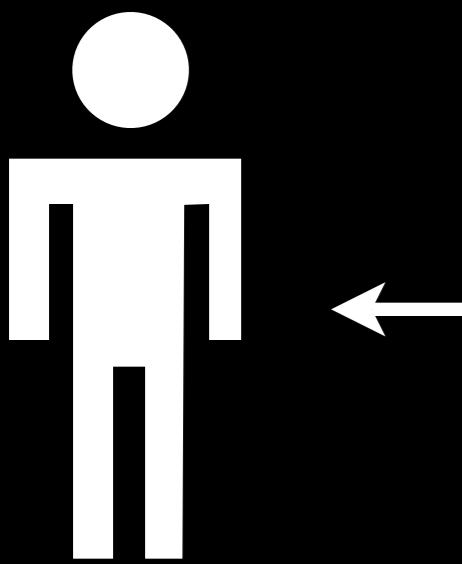
# Proxy based tools



The screenshot shows the Galaxy web interface. On the left, there's a sidebar with various tools like 'Get Data', 'Get ENCODE Data', 'ENCODE Tools', etc. The main area is titled 'UCSC Table Browser' and contains fields for 'clade' (Vertebrate), 'genome' (Human), 'assembly' (May 2004), 'group' (Genes and Gene Prediction Tracks), 'track' (Known Genes), 'table' (knownGene), and 'region' (position). A message in the history pane says 'Your history is empty. Click 'Get Data' on the left pane to start'. To the right, there's a larger panel for the 'Table Browser' with similar controls and a detailed description of its use.

Galaxy delegates request to external site

# Proxy based tools

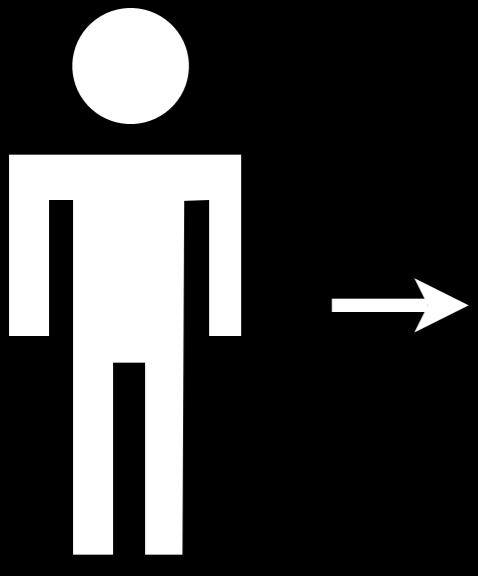


The image displays two side-by-side screenshots of the Galaxy web interface. On the left, a screenshot of the 'UCSC Table Browser' tool shows a search form for genomic data. On the right, a screenshot of the 'Table Browser' tool shows a similar search form. Both tools have a sidebar on the left containing various genomic analysis tools like 'Get Data', 'Get ENCODE Data', and 'Edit Queries'. A large white arrow points from the stick figure towards the Galaxy interface.

External site generates response

- If data, Galaxy determines type, processes, and adds to ‘history’
- Otherwise, return response to user

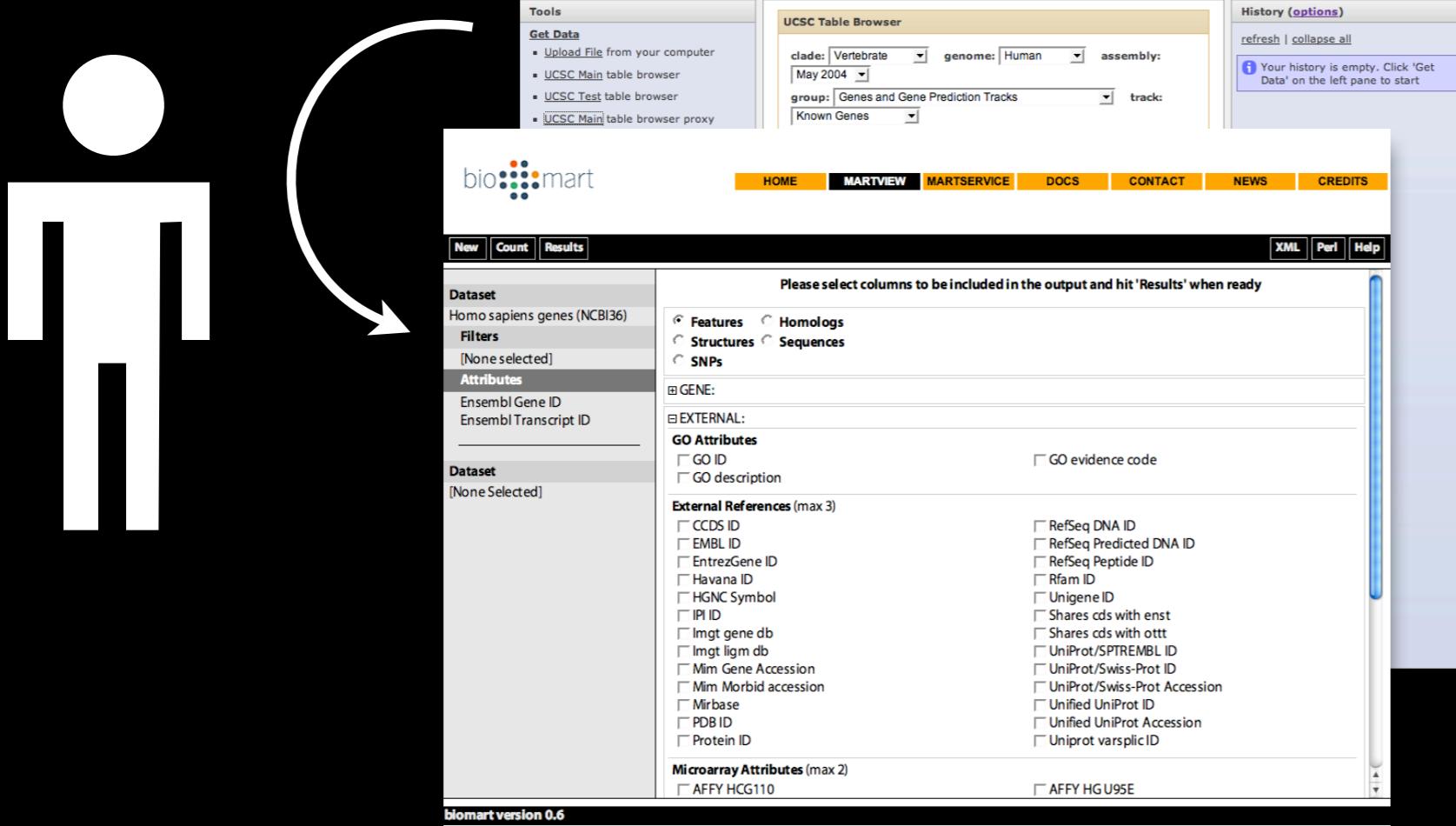
# External tools



The screenshot shows the Galaxy web interface. On the left, there is a sidebar titled "Tools" with several sections: "Get Data" (including "Upload File from your computer", "UCSC Main table browser", etc.), "Get ENCODE Data", "ENCODE Tools", "Edit Queries", "Filter, Sort, Join, Compare, Subtract", "Convert Formats", "Pattern-Matching", "Fetch Sequences and Alignments", "Get Genomic Scores", "Operate on Genomic Intervals", "Statistics", "Graph/Display Data", "EMBOSS", "PHYLIP", and "HYPHY". The main area is titled "UCSC Table Browser" and contains various input fields: "clade: Vertebrate", "genome: Human", "assembly: May 2004", "group: Genes and Gene Prediction Tracks", "track: Known Genes", "table: knownGene", "region: genome (radio button selected)", "position: chr7:127471196-127495720", "identifiers (names/accessions): paste list / upload list", "filter: create", "intersection: create", "correlation: create", "output format: BED - browser extensible data", and "Send output to Galaxy" with "get output" and "summary/statistics" buttons. A note at the bottom states: "This is a proxy to the data services provided by the UCSC Genome Browser's Table Browser." The top right shows the user is logged in as "james@bx.psu.edu" with links for "manage" and "logout". The top bar also includes links for "Info: report bugs | wiki | screencasts" and "History (options)" which says "Your history is empty. Click 'Get Data' on the left pane to start".

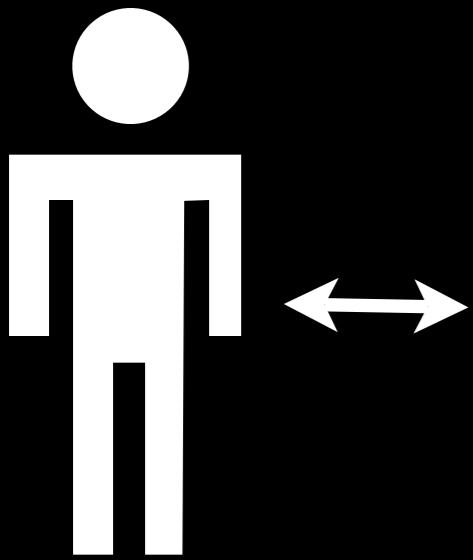
User makes request to Galaxy

# External tools



Galaxy sends user directly to external site  
with extra URL data

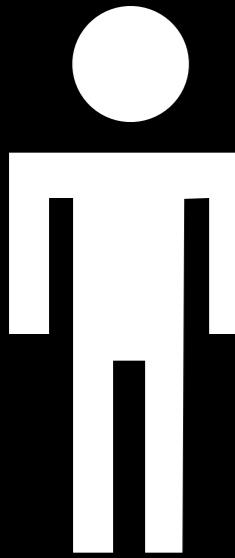
# External tools



The screenshot shows two side-by-side web interfaces. On the left is the Galaxy web interface, specifically the UCSC Table Browser. It has a sidebar with 'Tools' and 'Get Data' sections, and a main panel for 'UCSC Table Browser' with dropdown menus for 'clade', 'genome', 'assembly', 'group', and 'track'. On the right is the bioMart search interface. It has a header with 'HOME', 'MARTVIEW', 'MARTSERVICE', 'DOCS', 'CONTACT', 'NEWS', and 'CREDITS'. A sidebar on the left lists datasets like 'Homo sapiens genes (NCBI36)' and filters like 'Ensembl Gene ID'. The main panel contains a form for selecting columns to include in the output, with sections for 'Features', 'Homologs', 'Structures', 'Sequences', and 'SNPs'. It also includes sections for 'GENE', 'EXTERNAL', 'GO Attributes', 'External References (max 3)', and 'Microarray Attributes (max 2)'. At the bottom of the bioMart interface, it says 'biomart version 0.6'.

User interacts directly with external site

# External tools



The screenshot shows the Galaxy web interface. On the left, there's a sidebar with a 'bio' logo and sections for 'Tools' (including 'Get Data', 'UCSC Table Browser', 'BioMart', etc.) and 'Dataset' (with 'Homo sapiens' selected). The main area is titled 'UCSC Table Browser' and shows form fields for 'clade' (Vertebrate), 'genome' (Human), 'assembly' (May 2004), 'group' (Genes and Gene Prediction Tracks), 'track' (Known Genes), 'table' (knownGene), 'region' (chr7:127471196-127495720), and 'output format' (BED - browser extensible data). Below the form is a note: 'This is a proxy to the data services provided by the UCSC Genome Browser's Table Browser.' At the bottom, there are sections for 'Microarray Attributes' (checkboxes for Mim Gene Accession, Mim Morbid accession, Mirbase, PDB ID, Protein ID, UniProt/Swiss-Prot ID, UniProt/Swiss-Prot Accession, Unified UniProt ID, Unified UniProt Accession, Uniprot varsplicID, and AFFY HG U95E) and 'biomart version 0.6'.

When data is generated the user is sent back to Galaxy. Data can be fetched immediately, or wait for notification from the external site

**Tools****Get Data**

- [Upload File](#) from your computer
- [UCSC Main](#) table browser
- [UCSC Archaea](#) table browser
- [Get Microbial Data](#)
- [BioMart Central server](#)

**Get ENCODE Data****ENCODE Tools****Text Manipulation****Filter and Sort****Join, Subtract and Group****Convert Formats****Extract Features****Fetch Sequences****Fetch Alignments****Get Genomic Scores****Operate on Genomic Intervals****Statistics****Graph/Display Data****EMBOSS****HYPHY****Home   Genomes   Genome Browser   Blat   Tables   Gene Sorter   PCR   Session   FAQ   Help****Table Browser**

Use this program to retrieve the data associated with a track in text format, to calculate intersections between tracks, and to retrieve DNA sequence covered by a track. For help in using this application see [Using the Table Browser](#) for a description of the controls in this form, the [User's Guide](#) for general information and sample queries, and the OpenHelix Table Browser [tutorial](#) for a narrated presentation of the software features and usage. For more complex queries, you may want to use [Galaxy](#) or our [public MySQL server](#). Refer to the [Credits](#) page for the list of contributors and usage restrictions associated with these data.

clade:  genome:  assembly:

group:  track:

table:

region:  genome  ENCODE  position

identifiers (names/acccessions):

filter:

intersection:

correlation:

output format:   Send output to [Galaxy](#)

output file:  (leave blank to keep output in browser)

file type returned:  plain text  gzip compressed

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

**Using the Table Browser**

This section provides brief line-by-line descriptions of the Table Browser controls. For more information on using this program, see the [Table Browser User's Guide](#).

**Tools****Get Data**

- [Upload File](#) from your computer
- [UCSC Main](#) table browser
- [UCSC Archaea](#) table browser
- [Get Microbial Data](#)
- [BioMart Central server](#)

**Get ENCODE Data****ENCODE Tools****Text Manipulation****Filter and Sort****Join, Subtract and Group****Convert Formats****Extract Features****Fetch Sequences****Fetch Alignments****Get Genomic Scores****Operate on Genomic Intervals****Statistics****Graph/Display Data****EMBOSS****HYPHY****Home   Genomes   Genome Browser   Blat   Tables   Gene Sorter   PCR   Session   FAQ   Help****Output knownGene as BED** **Include [custom track](#) header:**

name=tb\_knownGene

description=table browser query on knownGene

visibility=pack ▾

url=

**Create one BED record per:** Whole Gene Upstream by 200 bases Exons plus 0 bases at each end Introns plus 0 bases at each end 5' UTR Exons Coding Exons 3' UTR Exons Downstream by 200 bases

Note: if a feature is close to the beginning or end of a chromosome and upstream/downstream bases are added, they may be truncated in order to avoid extending past the edge of the chromosome.

[Send query to Galaxy](#)[Cancel](#)

**Tools****Get Data**

- [Upload File](#) from your computer
- [UCSC Main](#) table browser
- [UCSC Archaea](#) table browser
- [Get Microbial Data](#)
- [BioMart](#) Central server

**Get ENCODE Data****ENCODE Tools**[Text Manipulation](#)[Filter and Sort](#)[Join, Subtract and Group](#)[Convert Formats](#)[Extract Features](#)[Fetch Sequences](#)[Fetch Alignments](#)[Get Genomic Scores](#)[Operate on Genomic Intervals](#)[Statistics](#)[Graph/Display Data](#)[EMBOSS](#)[HYPHY](#)

The following job has been successfully added to the queue:

**6: UCSC Main**

You can check the status of queued jobs and view the resulting data by refreshing the **History** pane. When the job has been run the status will change from 'running' to 'finished' if completed successfully or 'error' if problems were encountered.

**History (options)**

[refreshing in 8 sec](#) | [collapse all](#)

**6: UCSC Main on Human:** [@](#) / [X](#)  
[knownGene \(genome\)](#)

**5: Intersect on data 3 and data 4** [@](#) / [X](#)

**4: UCSC Main on Human:** [@](#) / [X](#)  
[knownGene \(genome\)](#)

**3: Pasted Entry** [@](#) / [X](#)

**Tools****Get Data**

- [Upload File](#) from your computer
- [UCSC Main](#) table browser
- [UCSC Archaea](#) table browser
- [Get Microbial Data](#)
- [BioMart Central server](#)

**Get ENCODE Data****ENCODE Tools****Text Manipulation****Filter and Sort****Join, Subtract and Group****Convert Formats****Extract Features****Fetch Sequences****Fetch Alignments****Get Genomic Scores****Operate on Genomic Intervals****Statistics****Graph/Display Data****EMBOSS****HYPHY**

The bioMart interface is shown, featuring a logo at the top left and a navigation bar with tabs: HOME, MARTVIEW (which is selected), MARTSERVICE, DOCS, and CON. Below the navigation bar are buttons for New, Count, Results, XML, Perl, and Help. A dropdown menu labeled "Dataset" shows "[None selected]" and "- CHOOSE DATABASE -". At the bottom, it says "biomart version 0.6".

**Tools****Get Data**

- [Upload File](#) from your computer
- [UCSC Main](#) table browser
- [UCSC Archaea](#) table browser
- [Get Microbial Data](#)
- [BioMart Central](#) server

**Get ENCODE Data****ENCODE Tools****Text Manipulation****Filter and Sort****Join, Subtract and Group****Convert Formats****Extract Features****Fetch Sequences****Fetch Alignments****Get Genomic Scores****Operate on Genomic Intervals****Statistics****Graph/Display Data****EMBOSS****HYPHY**

**bio**mart****

HOME MARTVIEW MARTSERVICE DOCS COM

New Count Results XML Perl Help

Dataset: Homo sapiens genes (NCBI36)

Filters: [None selected]

Attributes: Ensembl Gene ID, Ensembl Transcript ID, Transcript Start (bp), Transcript End (bp), Chromosome Name

Export all results to: Galaxy TSV Unique results only Go

Email notification to:

View: 10 rows as TSV Unique results only

Ensembl Gene ID	Ensembl Transcript ID	Transcript Start (bp)	Transcript End (bp)	Chromosome Name
ENSG00000184895	ENST00000383070	2714896	2715740	Y
ENSG00000184895	ENST00000327563	2715030	2715644	Y
ENSG00000129824	ENST00000322114	2769527	2794997	Y
ENSG00000129824	ENST00000250784	2769623	2794995	Y
ENSG00000067646	ENST00000383052	2863322	2910547	Y
ENSG00000067646	ENST00000155093	2863546	2909891	Y
ENSG00000176679	ENST00000383049	3507096	3508082	Y
ENSG00000176679	ENST00000321217	3507126	3508080	Y
ENSG00000099715	ENST00000333703	4928267	5033485	Y
ENSG00000099715	ENST00000362095	4928267	5033485	Y

bio**mart** version 0.6

**Tools****Get Data**

- [Upload File](#) from your computer
- [UCSC Main](#) table browser
- [UCSC Archaea](#) table browser
- [Get Microbial Data](#)
- [BioMart Central server](#)

**Get ENCODE Data****ENCODE Tools****Text Manipulation****Filter and Sort****Join, Subtract and Group****Convert Formats****Extract Features****Fetch Sequences****Fetch Alignments****Get Genomic Scores****Operate on Genomic Intervals****Statistics****Graph/Display Data****EMBOSS****HYPHY**

The following job has been successfully added to the queue:

**7: BioMart**

You can check the status of queued jobs and view the resulting data by refreshing the **History** pane. When the job has been run the status will change from 'running' to 'finished' if completed successfully or 'error' if problems were encountered.

**History (options)**

[refreshing in 8 sec](#) | [collapse all](#)

**7: BioMart** **6: UCSC Main on Human: knownGene (genome)** **5: Intersect on data 3 and data 4** **4: UCSC Main on Human: knownGene (genome)** **3: Pasted Entry**

**Tools****Get Data**

- [Upload File](#) from your computer
- [UCSC Main](#) table browser
- [UCSC Archaea](#) table browser
- [Get Microbial Data](#)
- [BioMart](#) Central server

**Get ENCODE Data****ENCODE Tools****Text Manipulation****Filter and Sort****Join, Subtract and Group****Convert Formats****Extract Features****Fetch Sequences****Fetch Alignments****Get Genomic Scores****Operate on Genomic Intervals****Statistics****Graph/Display Data****EMBOSS****HYPHY**

The following job has been successfully added to the queue:

**7: BioMart**

You can check the status of queued jobs and view the resulting data by refreshing the **History** pane. When the job has been run the status will change from 'running' to 'finished' if completed successfully or 'error' if problems were encountered.

**History (options)**

[refresh](#) | [collapse all](#)

**7: Homo sapiens genes (NCBI36)**

52,160 lines, format: tabular,  
database: ?

Info: Homo sapiens genes (NCBI36)  
[save](#)

**1****2**

Ensembl Gene ID	Ensembl Transcript ID
ENSG00000184895	ENST00000383070
ENSG00000184895	ENST00000327563
ENSG00000129824	ENST00000322114
ENSG00000129824	ENST00000250784
ENSG00000067646	ENST00000383052

**6: UCSC Main on Human: knownGene (genome)** **5: Intersect on data 3 and data 4** **4: UCSC Main on Human: knownGene (genome)** **3: Pasted Entry**

# How Galaxy integrates existing command line tools

**Tools****Get Data****Get ENCODE Data****ENCODE Tools****Text Manipulation****Filter and Sort****Join, Subtract and Group****Convert Formats****Extract Features****Fetch Sequences****Fetch Alignments****Get Genomic Scores****Operate on Genomic Intervals**

- Intersect the intervals of two queries
- Subtract the intervals of two queries
- Merge the overlapping intervals of a query
- Concatenate two queries into one query
- Base Coverage of all intervals
- Coverage of a set of intervals on second set of intervals
- Complement intervals of a query
- Cluster the intervals of a query
- Join the intervals of two queries side-by-side
- Get flanks returns flanking region/s for every gene

**Statistics****Graph/Display Data****EMBOSS****HYPHY****Cluster**

Cluster intervals of: 6: UCSC Main on Human: knownGene

max distance between intervals: 1  
(bp)

min number of intervals per cluster: 2

Return type: Merge clusters into single intervals

**Execute**

**i TIP:** If your query does not appear in the pulldown menu -> it is not in interval format. Use "edit attributes" to set chromosome, start, end, and strand columns

**Screencasts!**

See Galaxy Interval Operation [Screencasts](#) (right click to open this link in another window).

**Syntax**

- **Maximum distance** is greatest distance in base pairs allowed between intervals that will be considered "clustered". **Negative** values for distance are allowed, and are useful for clustering intervals that overlap.
- **Minimum intervals per cluster** allow a threshold to be set on the minimum number of intervals to be considered a cluster. Any area with less than this minimum will not be included in the output.
- **Merge clusters into single intervals** outputs intervals that span the entire cluster.
- **Find cluster intervals; preserve comments and order** filters out non-cluster intervals while maintaining the original ordering and comments in the file.
- **Find cluster intervals; output grouped by clusters** filters out non-cluster intervals, but outputs the cluster intervals so that they are grouped together. Comments and original ordering in the file are lost.

**Example****History (options)**[refresh](#) | [collapse all](#)**7: Homo sapiens genes (NCBI36)** [edit](#) / [X](#)**6: UCSC Main on Human: knownGene (genome)** [edit](#) / [X](#)**5: Intersect on data 3 and data 4** [edit](#) / [X](#)**4: UCSC Main on Human: knownGene (genome)** [edit](#) / [X](#)**3: Pasted Entry** [edit](#) / [X](#)

## Cluster

Cluster intervals of: 6: UCSC Main on Human: knownGene

max distance between intervals: 1 (bp)

min number of intervals per cluster: 2

Return type: Merge clusters into single intervals

**Execute**

**TIP:** If your query does not appear in the pulldown menu -> it is not in interval format. Use "edit attributes" to set chromosome, start, end, and strand columns

## Screencasts!

See Galaxy Interval Operation [Screencasts](#) (right click to open this link in another window).

## Syntax

- **Maximum distance** is greatest distance in base pairs allowed between intervals that will be considered "clustered". **Negative** values for distance are allowed, and are useful for clustering intervals that overlap.
- **Minimum intervals per cluster** allow a threshold to be set on the minimum number of intervals to be considered a cluster. Any area with less than this minimum will not be included in the output.
- **Merge clusters into single intervals** outputs intervals that span the entire cluster.
- **Find cluster intervals; preserve comments and order** filters out non-cluster intervals while maintaining the original ordering and comments in the file.
- **Find cluster intervals; output grouped by clusters** filters out non-cluster intervals, but outputs the cluster intervals so that they are grouped together. Comments and original ordering in the file are lost.

## Example



```
cluster.xml
1 <tool id="gops_cluster_1" name="Cluster">
2   <description>[[Cluster]] the intervals of a query</description>
3   <command interpreter="python2.4">
4     gops_cluster.py $input1 $output -1 $input1_chromCol,$input1_startC
5       -d $distance -m $minregions -o $returntype
6   </command>
7   <inputs>
8     <param format="interval" name="input1" type="data">
9       <label>Cluster intervals of</label>
10      </param>
11     <param name="distance" size="5" type="integer" value="1" help="(bp)">
12       <label>max distance between intervals</label>
13      </param>
14     <param name="minregions" size="5" type="integer" value="2">
15       <label>min number of intervals per cluster</label>
16      </param>
17     <param name="returntype" type="select" label="Return type">
18       <option value="1">Merge clusters into single intervals</option>
19       <option value="2">Find cluster intervals; preserve comments and
20         <option value="3">Find cluster intervals; output grouped by clus
21         <option value="4">Find the smallest interval in each cluster</op
22         <option value="5">Find the largest interval in each cluster</opt
23     </param>
24   </inputs>
25   <help>
26   .. class:: infomark
27
28 **TIP:** If your query does not appear in the pulldown menu -> it is n
29
30 -----
31
32 **Screencasts!**
33
34 See Galaxy Interval Operation Screencasts (right click to open this l
35
36 ...
37 ... _Screencasts: http://www.bx.psu.edu/cgi-bin/trac.cgi/GopsDesc
38
39 ...
40
41 **Syntax**
42
43 - **Maximum distance** is greatest distance in base pairs allowed betw
44 - **Minimum intervals per cluster** allow a threshold to be set on the
45 - **Merge clusters into single intervals** outputs intervals that span
46 - **Find cluster intervals; preserve comments and order** filters out
47 - **Find cluster intervals; output grouped by clusters** filters out n
```

**Cluster**

Cluster intervals of: 6: UCSC Main on Human: knownGene

max distance between intervals: 1 (bp)

min number of intervals per cluster: 2

Return type: Merge clusters into single intervals

**Execute**

**TIP:** If your query does not appear in the pulldown menu -> it is not in interval format. Use "edit attributes" to set chromosome, start, end, and strand columns

## Screencasts!

See Galaxy Interval Operation [Screencasts](#) (right click to open this link in another window).

## Syntax

- **Maximum distance** is greatest distance in base pairs allowed between intervals that will be considered "clustered". **Negative** values for distance are allowed, and are useful for clustering intervals that overlap.
- **Minimum intervals per cluster** allow a threshold to be set on the minimum number of intervals to be considered a cluster. Any area with less than this minimum will not be included in the output.
- **Merge clusters into single intervals** outputs intervals that span the entire cluster.
- **Find cluster intervals; preserve comments and order** filters out non-cluster intervals while maintaining the original ordering and comments in the file.
- **Find cluster intervals; output grouped by clusters** filters out non-cluster intervals, but outputs the cluster intervals so that they are grouped together. Comments and original ordering in the file are lost.

## Example



```

cluster.xml
1 <tool id="gops_cluster_1" name="Cluster">
2   <description>[[Cluster]] the intervals of a query</description>
3   <command interpreter="python2.4">
4     gops_cluster.py $input1 $output -1 $input1_chromCol,$input1_startC
5       -d $distance -m $minregions -o $returntype
6   </command>
7   <inputs>
8     <param format="interval" name="input1" type="data">
9       <label>Cluster intervals of</label>
10      </param>
11     <param name="distance" size="5" type="integer" value="1" help="(bp)">
12       <label>max distance between intervals</label>
13     </param>
14     <param name="minregions" size="5" type="integer" value="2">
15       <label>min number of intervals per cluster</label>
16     </param>
17     <param name="returntype" type="select" label="Return type">
18       <option value="1">Merge clusters into single intervals</option>
19       <option value="2">Find cluster intervals; preserve comments and
20         <option value="3">Find cluster intervals; output grouped by clus
21         <option value="4">Find the smallest interval in each cluster</op
22         <option value="5">Find the largest interval in each cluster</opt
23     </param>
24   </inputs>
25   <help>
26   .. class:: infomark
27
28 **TIP:** If your query does not appear in the pulldown menu -> it is n
29
30 -----
31
32 **Screencasts!**
33
34 See Galaxy Interval Operation Screencasts (right click to open this l
35
36 ...
37 ... _Screencasts: http://www.bx.psu.edu/cgi-bin/trac.cgi/wiki/GopsDesc
38 ...
39
40 **Syntax**
41
42 - **Maximum distance** is greatest distance in base pairs allowed betw
43 - **Minimum intervals per cluster** allow a threshold to be set on the
44 - **Merge clusters into single intervals** outputs intervals that span
45 - **Find cluster intervals; preserve comments and order** filters out

```

# HTML inputs generated from abstract parameter description

**Cluster**

Cluster intervals of: 6: UCSC Main on Human: knownGene

max distance between intervals: 1 (bp)

min number of intervals per cluster: 2

Return type: Merge clusters into single intervals

**Execute**

**TIP:** If your query does not appear in the pulldown menu -> it is not in interval format. Use "edit attributes" to set chromosome, start, end, and strand columns

## Screencasts!

See Galaxy Interval Operation [Screencasts](#) (right click to open this link in another window).

## Syntax

- **Maximum distance** is greatest distance in base pairs allowed between intervals that will be considered "clustered". **Negative** values for distance are allowed, and are useful for clustering intervals that overlap.
- **Minimum intervals per cluster** allow a threshold to be set on the minimum number of intervals to be considered a cluster. Any area with less than this minimum will not be included in the output.
- **Merge clusters into single intervals** outputs intervals that span the entire cluster.
- **Find cluster intervals; preserve comments and order** filters out non-cluster intervals while maintaining the original ordering and comments in the file.
- **Find cluster intervals; output grouped by clusters** filters out non-cluster intervals, but outputs the cluster intervals so that they are grouped together. Comments and original ordering in the file are lost.

## Example



```
cluster.xml
1 <tool id="gops_cluster_1" name="Cluster">
2   <description>[[Cluster]] the intervals of a query</description>
3   <command interpreter="python2.4">
4     gops_cluster.py $input1 $output -1 $input1_chromCol,$input1_startC
5       -d $distance -m $minregions -o $returntype
6   </command>
7   <inputs>
8     <param format="interval" name="input1" type="data">
9       <label>Cluster intervals of</label>
10      </param>
11     <param name="distance" size="5" type="integer" value="1" help="(bp)">
12       <label>max distance between intervals</label>
13     </param>
14     <param name="minregions" size="5" type="integer" value="2">
15       <label>min number of intervals per cluster</label>
16     </param>
17     <param name="returntype" type="select" label="Return type">
18       <option value="1">Merge clusters into single intervals</option>
19       <option value="2">Find cluster intervals; preserve comments and
20         <option value="3">Find cluster intervals; output grouped by clus
21         <option value="4">Find the smallest interval in each cluster</op
22         <option value="5">Find the largest interval in each cluster</opt
23     </param>
24   </inputs>
25   <help>
26   .. class:: infomark
27
28   **TIP:** If your query does not appear in the pulldown menu -> it is n
29
30   -----
31
32   **Screencasts!**
33
34   See Galaxy Interval Operation Screencasts (right click to open this l
35
36   .. _Screencasts: http://www.bx.psu.edu/cgi-bin/trac.cgi/wiki/GopsDesc
37
38   -----
39
40   **Syntax**
41
42   - **Maximum distance** is greatest distance in base pairs allowed betw
43   - **Minimum intervals per cluster** allow a threshold to be set on the
44   - **Merge clusters into single intervals** outputs intervals that span
45   - **Find cluster intervals; preserve comments and order** filters out
```

# HTML inputs generated from abstract parameter description

**Cluster**

Cluster intervals of: 6: UCSC Main on Human: knownGene

max distance between intervals: 1 (bp)

min number of intervals per cluster: 2

Return type: Merge clusters into single intervals

**Execute**

**TIP:** If your query does not appear in the pulldown menu -> it is not in interval format. Use "edit attributes" to set chromosome, start, end, and strand columns

## Screencasts!

See Galaxy Interval Operation [Screencasts](#) (right click to open this link in another window).

## Syntax

- **Maximum distance** is greatest distance in base pairs allowed between intervals that will be considered "clustered". **Negative** values for distance are allowed, and are useful for clustering intervals that overlap.
- **Minimum intervals per cluster** allow a threshold to be set on the minimum number of intervals to be considered a cluster. Any area with less than this minimum will not be included in the output.
- **Merge clusters into single intervals** outputs intervals that span the entire cluster.
- **Find cluster intervals; preserve comments and order** filters out non-cluster intervals while maintaining the original ordering and comments in the file.
- **Find cluster intervals; output grouped by clusters** filters out non-cluster intervals, but outputs the cluster intervals so that they are grouped together. Comments and original ordering in the file are lost.

## Example



```
cluster.xml
1 <tool id="gops_cluster_1" name="Cluster">
2   <description>[[Cluster]] the intervals of a query</description>
3   <command interpreter="python2.4">
4     gops_cluster.py $input1 $output -1 $input1_chromCol,$input1_startC
5       -d $distance -m $minregions -o $returntype
6   </command>
7   <inputs>
8     <param format="interval" name="input1" type="data">
9       <label>Cluster intervals of</label>
10      </param>
11     <param name="distance" size="5" type="integer" value="1" help="(bp)">
12       <label>max distance between intervals</label>
13      </param>
14     <param name="minregions" size="5" type="integer" value="2">
15       <label>min number of intervals per cluster</label>
16      </param>
17     <param name="returntype" type="select" label="Return type">
18       <option value="1">Merge clusters into single intervals</option>
19       <option value="2">Find cluster intervals; preserve comments and
20         <option value="3">Find cluster intervals; output grouped by clus
21         <option value="4">Find the smallest interval in each cluster</op
22         <option value="5">Find the largest interval in each cluster</opt
23       </param>
24   </inputs>
25   <help>
26   .. class:: infomark
27
28   **TIP:** If your query does not appear in the pulldown menu -> it is n
29
30   -----
31
32   **Screencasts!**
33
34
35   See Galaxy Interval Operation Screencasts (right click to open this l
36
37   ... _Screencasts: http://www.bx.psu.edu/cgi-bin/trac.cgi/wiki/GopsDesc
38
39   -----
40
41   **Syntax**
42
43   - **Maximum distance** is greatest distance in base pairs allowed betw
44   - **Minimum intervals per cluster** allow a threshold to be set on the
45   - **Merge clusters into single intervals** outputs intervals that span
46   - **Find cluster intervals; preserve comments and order** filters out
```

# HTML inputs generated from abstract parameter description

**Cluster**

Cluster intervals of: 6: UCSC Main on Human: knownGene

max distance between intervals: 1 (bp)

min number of intervals per cluster: 2

Return type: Merge clusters into single intervals

**Execute**

**TIP:** If your query does not appear in the pulldown menu -> it is not in interval format. Use "edit attributes" to set chromosome, start, end, and strand columns

## Screencasts!

See Galaxy Interval Operation [Screencasts](#) (right click to open this link in another window).

## Syntax

- **Maximum distance** is greatest distance in base pairs allowed between intervals that will be considered "clustered". **Negative** values for distance are allowed, and are useful for clustering intervals that overlap.
- **Minimum intervals per cluster** allow a threshold to be set on the minimum number of intervals to be considered a cluster. Any area with less than this minimum will not be included in the output.
- **Merge clusters into single intervals** outputs intervals that span the entire cluster.
- **Find cluster intervals; preserve comments and order** filters out non-cluster intervals while maintaining the original ordering and comments in the file.
- **Find cluster intervals; output grouped by clusters** filters out non-cluster intervals, but outputs the cluster intervals so that they are grouped together. Comments and original ordering in the file are lost.

## Example



```

cluster.xml
1 <tool id="gops_cluster_1" name="Cluster">
2   <description>[[Cluster]] the intervals of a query</description>
3   <command interpreter="python2.4">
4     gops_cluster.py $input1 $output -1 $input1_chromCol,$input1_startC
5       -d $distance -m $minregions -o $returntype
6   </command>
7   <inputs>
8     <param format="interval" name="input1" type="data">
9       <label>Cluster intervals of</label>
10      </param>
11     <param name="distance" size="5" type="integer" value="1" help="(bp)">
12       <label>max distance between intervals</label>
13      </param>
14     <param name="minregions" size="5" type="integer" value="2">
15       <label>min number of intervals per cluster</label>
16      </param>
17     <param name="returntype" type="select" label="Return type">
18       <option value="1">Merge clusters into single intervals</option>
19       <option value="2">Find cluster intervals; preserve comments and
20         <option value="3">Find cluster intervals; output grouped by clus
21         <option value="4">Find the smallest interval in each cluster</op
22         <option value="5">Find the largest interval in each cluster</opt
23     </param>
24   </inputs>
25   <help>
26   .. class:: infomark
27
28 **TIP:** If your query does not appear in the pulldown menu -> it is n
29
30 -----
31
32 **Screencasts!**
33
34 See Galaxy Interval Operation Screencasts (right click to open this l
35
36 ...
37 ... _Screencasts: http://www.bx.psu.edu/cgi-bin/trac.cgi/wiki/GopsDesc
38 ...
39
40 **Syntax**
41
42 - **Maximum distance** is greatest distance in base pairs allowed betw
43 - **Minimum intervals per cluster** allow a threshold to be set on the
44 - **Merge clusters into single intervals** outputs intervals that span
45 - **Find cluster intervals; preserve comments and order** filters out

```

# HTML inputs generated from abstract parameter description

# Tool help generated from a simple text format

Cluster intervals of: 6: UCSC Main on Human: knownGene

max distance between intervals: 1 (bp)

min number of intervals per cluster: 2

Return type: Merge clusters into single intervals

Execute

**TIP:** If your query does not appear in the pulldown menu -> it is not in interval format. Use "edit attributes" to set chromosome, start, end, and strand columns

## Screencasts!

See Galaxy Interval Operation [Screencasts](#) (right click to open this link in another window).

## Syntax

- **Maximum distance** is greatest distance in base pairs allowed between intervals that will be considered "clustered". **Negative** values for distance are allowed, and are useful for clustering intervals that overlap.
- **Minimum intervals per cluster** allow a threshold to be set on the minimum number of intervals to be considered a cluster. Any area with less than this minimum will not be included in the output.
- **Merge clusters into single intervals** outputs intervals that span the entire cluster.
- **Find cluster intervals; preserve comments and order** filters out non-cluster intervals while maintaining the original ordering and comments in the file.
- **Find cluster intervals; output grouped by clusters** filters out non-cluster intervals, but outputs the cluster intervals so that they are grouped together. Comments and original ordering in the file are lost.

## Example



```
3 <command interpreter="python2.4">
4   gops_cluster.py $input1 $output -1 $input1_chromCol,$input1_startC
5   -d $distance -m $minregions -o $returntype
6 </command>
7 <inputs>
8   <param format="interval" name="input1" type="data">
9     <label>Cluster intervals of</label>
10    </param>
11   <param name="distance" size="5" type="integer" value="1" help="(bp)">
12     <label>max distance between intervals</label>
13   </param>
14   <param name="minregions" size="5" type="integer" value="2">
15     <label>min number of intervals per cluster</label>
16   </param>
17   <param name="returntype" type="select" label="Return type">
18     <option value="1">Merge clusters into single intervals</option>
19     <option value="2">Find cluster intervals; preserve comments and
20       order</option>
21     <option value="3">Find cluster intervals; output grouped by clusters</option>
22     <option value="4">Find the smallest interval in each cluster</option>
23     <option value="5">Find the largest interval in each cluster</option>
24   </param>
25 </inputs>
26 <help>
27 .. class:: infomark
28
29 **TIP:** If your query does not appear in the pulldown menu -> it is not in interval format. Use "edit attributes" to set chromosome, start, end, and strand columns
30
31 -----
32
33 **Screencasts!**
34
35 See Galaxy Interval Operation Screencasts (right click to open this link in another window).
36
37 .. _Screencasts: http://www.bx.psu.edu/cgi-bin/trac.cgi/wiki/GopsDescriptor
38
39 -----
40
41 **Syntax**
42
43 - **Maximum distance** is greatest distance in base pairs allowed between intervals that will be considered "clustered".
44 - **Minimum intervals per cluster** allow a threshold to be set on the minimum number of intervals to be considered a cluster.
45 - **Merge clusters into single intervals** outputs intervals that span the entire cluster.
46 - **Find cluster intervals; preserve comments and order** filters out non-cluster intervals while maintaining the original ordering and comments in the file.
47 - **Find cluster intervals; output grouped by clusters** filters out non-cluster intervals, but outputs the cluster intervals so that they are grouped together. Comments and original ordering in the file are lost.
```

One or more errors were found in the input you provided. The specific errors are marked below.

## Cluster

Cluster intervals of: 6: UCSC Main on Human: knownGene ▾

max distance between intervals: seven

An integer is required

min number of intervals per cluster:

2

Return type: Merge clusters into single intervals ▾

Execute

**TIP:** If your query does not appear in the pulldown menu -> it is not in interval format. Use "edit attributes" to set chromosome, start, end, and strand columns

## Screencasts!

See Galaxy Interval Operation [Screencasts](#) (right click to open this link in another window).

## Syntax

- **Maximum distance** is greatest distance in base pairs allowed between intervals that will be considered "clustered". **Negative** values for distance are allowed, and are useful for clustering intervals that overlap.
- **Minimum intervals per cluster** allow a threshold to be set on the minimum number of intervals to be considered a cluster. Any area with less than this minimum will not be included in the output.
- **Merge clusters into single intervals** outputs intervals that span the entire cluster.
- **Find cluster intervals; preserve comments and order** filters out non-cluster intervals while maintaining the original ordering and comments in the file.
- **Find cluster intervals; output grouped by clusters** filters out non-cluster intervals, but outputs the cluster intervals so that they are grouped together. Comments and original ordering in the file are lost.

```
cluster.xml
1 <tool id="gops_cluster_1" name="Cluster">
2   <description>[[Cluster]] the intervals of a query</description>
3   <command interpreter="python2.4">
4     gops_cluster.py $input1 $output -1 $input1_chromCol,$input1_startC
5       -d $distance -m $minregions -o $returntype
6   </command>
7   <inputs>
8     <param format="interval" name="input1" type="data">
9       <label>Cluster intervals of</label>
10      </param>
11     <param name="distance" size="5" type="integer" value="1" help="(bp
12       <label>max distance between intervals</label>
13      </param>
14     <param name="minregions" size="5" type="integer" value="2">
15       <label>min number of intervals per cluster</label>
16      </param>
17     <param name="returntype" type="select" label="Return type">
18       <option value="1">Merge clusters into single intervals</option>
19       <option value="2">Find cluster intervals; preserve comments and
20       <option value="3">Find cluster intervals; output grouped by clus
21       <option value="4">Find the smallest interval in each cluster</op
22       <option value="5">Find the largest interval in each cluster</op
23     </param>
24   </inputs>
25   <help>
26
27   .. class:: infomark
28
29   **TIP:** If your query does not appear in the pulldown menu -> it is n
30
31   -----
32
33   **Screencasts!**
34
35   See Galaxy Interval Operation Screencasts (right click to open this l
36
37   .. _Screencasts: 

Automatic input validation based on type, or more...


```



```
1 <tool id="gops_cluster_1" name="Cluster">
2   <description>[[Cluster]] the intervals of a query</description>
3   <command interpreter="python2.4">
4     gops_cluster.py $input1 $output -1 $input1_chromCol,$input1_startCol,$input1_endCol
5       -d $distance -m $minregions -o $returntype
6   </command>
7   <inputs>
8     <param format="interval" name="input1" type="data">
9       <label>Cluster intervals of</label>
10      </param>
11     <param name="distance" size="5" type="integer" value="1" help="(bp)">
12       <label>max distance between intervals</label>
13      </param>
14     <param name="minregions" size="5" type="integer" value="2">
15       <label>min number of intervals per cluster</label>
16      </param>
17     <param name="returntype" type="select" label="Return type">
18       <option value="1">Merge clusters into single intervals</option>
19       <option value="2">Find cluster intervals; preserve comments and order</option>
20       <option value="3">Find cluster intervals; output grouped by clusters</option>
21       <option value="4">Find the smallest interval in each cluster</option>
22       <option value="5">Find the largest interval in each cluster</option>
23     </param>
24   </inputs>
25   <help>
26
27 .. class:: infomark
28
29 **TIP:** If your query does not appear in the pulldown menu -> it is not in interval fo
30
31 -----
32
33 **Screencasts**
34
35 See Galaxy Interval Operation Screencasts (right click to open this link in another wi
36
37 .. _Screencasts: http://www.bx.psu.edu/cgi-bin/trac.cgi/wiki/GopsDesc
38
39 -----
40
41 **Syntax**
42
43 - **Maximum distance** is greatest distance in base pairs allowed between intervals tha
44 - **Minimum intervals per cluster** allow a threshold to be set on the minimum number o
45 - **Merge clusters into single intervals** outputs intervals that span the entire cluster
46 - **Find cluster intervals; preserve comments and order** filters out non-cluster interva
47 - **Find cluster intervals; output grouped by clusters** filters out non-cluster interv
```

# Template for generating } command line from parameter values

```
cluster.xml
1 <tool id="gops_cluster_1" name="Cluster">
2   <description>[[Cluster]] the intervals of a query</description>
3   <command interpreter="python2.4">
4     gops_cluster.py $input1 $output -1 $input1_chromCol,$input1_startCol,$input1_endCol
5       -d $distance -m $minregions -o $returntype
6   </command>
7   <inputs>
8     <param format="interval" name="input1" type="data">
9       <label>Cluster intervals of</label>
10      </param>
11     <param name="distance" size="5" type="integer" value="1" help="(bp)">
12       <label>max distance between intervals</label>
13      </param>
14     <param name="minregions" size="5" type="integer" value="2">
15       <label>min number of intervals per cluster</label>
16      </param>
17     <param name="returntype" type="select" label="Return type">
18       <option value="1">Merge clusters into single intervals</option>
19       <option value="2">Find cluster intervals; preserve comments and order</option>
20       <option value="3">Find cluster intervals; output grouped by clusters</option>
21       <option value="4">Find the smallest interval in each cluster</option>
22       <option value="5">Find the largest interval in each cluster</option>
23     </param>
24   </inputs>
25   <help>
26
27 .. class:: infomark
28
29 **TIP:** If your query does not appear in the pulldown menu -> it is not in interval fo
30
31 -----
32
33 **Screencasts**
34
35 See Galaxy Interval Operation Screencasts (right click to open this link in another wi
36
37 .. _Screencasts: http://www.bx.psu.edu/cgi-bin/trac.cgi/wiki/GopsDesc
38
39 -----
40
41 **Syntax**
42
43 - **Maximum distance** is greatest distance in base pairs allowed between intervals tha
44 - **Minimum intervals per cluster** allow a threshold to be set on the minimum number o
45 - **Merge clusters into single intervals** outputs intervals that span the entire cluster
46 - **Find cluster intervals; preserve comments and order** filters out non-cluster interva
47 - **Find cluster intervals; output grouped by clusters** filters out non-cluster interv
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

} Output datasets  
generated by the tool

```
cluster.xml
41 **Syntax**
42
43 - **Maximum distance** is greatest distance in base pairs allowed between intervals that
44 - **Minimum intervals per cluster** allow a threshold to be set on the minimum number of
45 - **Merge clusters into single intervals** outputs intervals that span the entire cluster
46 - **Find cluster intervals; preserve comments and order** filters out non-cluster intervals
47 - **Find cluster intervals; output grouped by clusters** filters out non-cluster intervals
48
49 -----
50
51 **Example**
52
53 .. image:: ../static/operation_icons/gops_cluster.gif
54
55 </help>
56
57 <outputs>
58   <data format="input" name="output" metadata_source="input1" />
59 </outputs>
60 <code file="operation_filter.py">
61   <hook exec_after_process="exec_after_cluster" />
62 </code>
63 <tests>
64   <test>
65     <param name="input1" value="1.bed" />
66     <param name="distance" value="1" />
67     <param name="minregions" value="2" />
68     <param name="returntype" value="1" />
69     <output name="output" file="gops-cluster-1.dat" />
70   </test>
71   <test>
72     <param name="input1" value="1.bed" />
73     <param name="distance" value="1" />
74     <param name="minregions" value="2" />
75     <param name="returntype" value="2" />
76     <output name="output" file="gops-cluster-2.dat" />
77   </test>
78   <test>
79     <param name="input1" value="1.bed" />
80     <param name="distance" value="1" />
81     <param name="minregions" value="2" />
82     <param name="returntype" value="3" />
83     <output name="output" file="gops-cluster-3.dat" />
84   </test>
85 </tests>
86
87 </tool>
```

cluster.xml

```
41 **Syntax**
42
43 - **Maximum distance** is greatest distance in base pairs allowed between intervals that
44 - **Minimum intervals per cluster** allow a threshold to be set on the minimum number of
45 - **Merge clusters into single intervals** outputs intervals that span the entire cluster
46 - **Find cluster intervals; preserve comments and order** filters out non-cluster intervals
47 - **Find cluster intervals; output grouped by clusters** filters out non-cluster intervals
48
49 -----
50
51 **Example**
52
53 .. image:: ../static/operation_icons/gops_cluster.gif
54
55 </help>
56
57 <outputs>
58   <data format="input" name="output" metadata_source="input1" />
59 </outputs>
60 <code file="operation_filter.py">
61   <hook exec_after_process="exec_after_cluster" />
62 </code>
63 <tests>
64   <test>
65     <param name="input1" value="1.bed" />
66     <param name="distance" value="1" />
67     <param name="minregions" value="2" />
68     <param name="returntype" value="1" />
69     <output name="output" file="gops-cluster-1.dat" />
70   </test>
71   <test>
72     <param name="input1" value="1.bed" />
73     <param name="distance" value="1" />
74     <param name="minregions" value="2" />
75     <param name="returntype" value="2" />
76     <output name="output" file="gops-cluster-2.dat" />
77   </test>
78   <test>
79     <param name="input1" value="1.bed" />
80     <param name="distance" value="1" />
81     <param name="minregions" value="2" />
82     <param name="returntype" value="3" />
83     <output name="output" file="gops-cluster-3.dat" />
84   </test>
85 </tests>
86
87 </tool>
```

} Special actions to be run  
before / after execution

cluster.xml

```
41 **Syntax**
42
43 - **Maximum distance** is greatest distance in base pairs allowed between intervals that
44 - **Minimum intervals per cluster** allow a threshold to be set on the minimum number of
45 - **Merge clusters into single intervals** outputs intervals that span the entire cluster
46 - **Find cluster intervals; preserve comments and order** filters out non-cluster intervals
47 - **Find cluster intervals; output grouped by clusters** filters out non-cluster intervals
48
49 -----
50
51 **Example**
52
53 .. image:: ../static/operation_icons/gops_cluster.gif
54
55 </help>
56
57 <outputs>
58   <data format="input" name="output" metadata_source="input1" />
59 </outputs>
60 <code file="operation_filter.py">
61   <hook exec_after_process="exec_after_cluster" />
62 </code>
63 <tests>
64   <test>
65     <param name="input1" value="1.bed" />
66     <param name="distance" value="1" />
67     <param name="minregions" value="2" />
68     <param name="returntype" value="1" />
69     <output name="output" file="gops-cluster-1.dat" />
70   </test>
71   <test>
72     <param name="input1" value="1.bed" />
73     <param name="distance" value="1" />
74     <param name="minregions" value="2" />
75     <param name="returntype" value="2" />
76     <output name="output" file="gops-cluster-2.dat" />
77   </test>
78   <test>
79     <param name="input1" value="1.bed" />
80     <param name="distance" value="1" />
81     <param name="minregions" value="2" />
82     <param name="returntype" value="3" />
83     <output name="output" file="gops-cluster-3.dat" />
84   </test>
85 </tests>
86
87 </tool>
```

Functional tests to be run  
with the “full stack” in  
place

```
Default Default Default
henduck% sh run_functional_tests.sh -id gops_cluster_1
'run_functional_tests.sh help'          for help
Architecture appears to be darwin-i386
python path: /Users/james/projects/galaxy/code/galaxy-trunk/lib:/Users/james/projects/galaxy/code/galaxy-trunk/modules:/Users/james/projects/galaxy/code/galaxy-trunk/eggs:/Users/james/projects/galaxy/code/galaxy-trunk/arch/darwin-i386/lib/python:eggs/NoseHTML-0.2-py2.4.egg
Operate on Genomic Intervals > Cluster > Test-1 ... ok
Operate on Genomic Intervals > Cluster > Test-2 ... ok
Operate on Genomic Intervals > Cluster > Test-3 ... ok

-----
Ran 3 tests in 38.260s
OK
henduck%
```

Running functional tests for a specific tool on the command line

**ID:** functional.test\_toolbox.GeneratedToolTestCase\_gops\_cluster\_1.test\_tool

**Description:** Operate on Genomic Intervals > Cluster > Test-1

**Status:** failure

**Output:** [...](#)

```
galaxy.datatypes.registry WARNING 2007-07-16 18:55:30,380 unknown extension in data factory None
```

**Exception:** [...](#)

```
Traceback (most recent call last):
```

```
  File "/Library/Frameworks/Python.framework/Versions/2.4//lib/python2.4/unittest.py", line 260, in run
    testMethod()

  File "/Users/james/projects/galaxy/code/galaxy-trunk/test/functional/test_toolbox.py", line 48, in test_tool
    self.do_it()

  File "/Users/james/projects/galaxy/code/galaxy-trunk/test/functional/test_toolbox.py", line 37, in do_it
    self.check_data( file )

  File "/Users/james/projects/galaxy/code/galaxy-trunk/test/base/twilltestcase.py", line 239, in check_data
    raise AssertionError( errmsg )
```

```
AssertionError: Data at history id 1 does not match expected, diff:
```

```
--- local_file
+++ history_data
@@ -1,4 +1,65 @@
-chr6 108640045 108640151
-chr6 108642394 108642519
-chr6 108650846 108650942
-chr6 108688656 108688818
+chr1 147962192 147962580 CCDS989.1_cds_0_0_chr1_147962193_r 0 -
+chr1 147984545 147984630 CCDS990.1_cds_0_0_chr1_147984546_f 0 +
+chr1 148078400 148078582 CCDS993.1_cds_0_0_chr1_148078401_r 0 -
+chr1 148185136 148185276 CCDS996.1_cds_0_0_chr1_148185137_f 0 +
```

Test results, on command line and as HTML report

Dealing with more complex  
interface needs

**Tools****Get Data**[Get ENCODE Data](#)[ENCODE Tools](#)[Text Manipulation](#)[Filter and Sort](#)[Join, Subtract and Group](#)[Convert Formats](#)[Extract Features](#)[Fetch Sequences](#)[Fetch Alignments](#)[Get Genomic Scores](#)[Operate on Genomic Intervals](#)[Statistics](#)[Graph/Display Data](#)

- [Histogram](#) of a numeric columns
- [Scatterplot](#) of two numeric columns
- [XY Plot](#)
- [GMAJ](#) Multiple Alignment Viewer
- [Build custom track](#) for UCSC genome browser

[EMBOSS](#)[HYPHY](#)**Build custom track****Tracks****Track 1**

Dataset:

6: UCSC Main on Human: knownGene ▾

name:

User Track

description:

User Suppli

Color:

Black ▾

Visibility:

Dense ▾

[Remove Track 1](#)**Track 2**

Dataset:

4: UCSC Main on Human: knownGene ▾

name:

User Track

description:

User Suppli

Color:

Magenta ▾

Visibility:

Full ▾

[Remove Track 2](#)[Add new Track](#)[Execute](#)**Info**

This tool displays the selected datasets with their custom track attributes (if any) in the UCSC genome browser.

This tool allows you to set the **Color** and **Visibility** attributes and you can edit the **Name** attribute of the dataset by clicking on "**edit attributes**" button (pencil icon) next to the dataset name in the history panel.

Please note that the primary dataset in step 1 of the tool sets the database build for the datasets in following steps. For example, if your first dataset belongs to hg18, you will only be able to select hg18 associated datasets on the next step.

**History (options)**[refresh](#) | [collapse all](#)**7: Homo sapiens genes (NCBI36)** / **6: UCSC Main on Human: knownGene (genome)** / **5: Intersect on data 3 and data 4** / **4: UCSC Main on Human: knownGene (genome)** / **3: Pasted Entry** /

### Build custom track

**Tracks**

**Track 1**

Dataset: 6: UCSC Main on Human: knownGene

name: User Track

description: User Suppli

Color: Black

Visibility: Dense

[Remove Track 1](#)

**Track 2**

Dataset: 4: UCSC Main on Human: knownGene

name: User Track

description: User Suppli

Color: Magenta

Visibility: Full

[Remove Track 2](#)

[Add new Track](#)

[Execute](#)

### Info

This tool displays the selected datasets with their custom track attributes (if any) in the UCSC genome browser.

This tool allows you to set the **Color** and **Visibility** attributes and you can edit the **Name** attribute of the dataset by clicking on "edit attributes" button (pencil icon) next to the dataset name in the history panel.

Please note that the primary dataset in step 1 of the tool sets the database build for the datasets in following steps. For example, if your first dataset belongs to hg18, you will only be able to select hg18 associated datasets on the next step.

```

build_ucsc_custom_track.xml

20 <inputs>
21   <repeat name="tracks" title="Track">
22     <param name="input" type="data" format="interval,wig" label="D
23     <param name="name" type="text" size="15" value="User Track">
24       <validator type="length" max="15"/>
25     </param>
26     <param name="description" type="text" value="User Supplied Tra
27       <validator type="length" max="60"/>
28     </param>
29     <param label="Color" name="color" type="select">
30       <option selected="yes" value="0-0-0">Black</option>
31       <option value="255-0-0">Red</option>
32       <option value="0-255-0">Green</option>
33       <option value="0-0-255">Blue</option>
34       <option value="255-0-255">Magenta</option>
35       <option value="0-255-255">Cyan</option>
36       <option value="255-215-0">Gold</option>
37       <option value="160-32-240">Purple</option>
38       <option value="255-140-0">Orange</option>
39       <option value="255-20-147">Pink</option>
40       <option value="92-51-23">Dark Chocolate</option>
41       <option value="85-107-47">Olive green</option>
42     </param>
43     <param label="Visibility" name="visibility" type="select">
44       <option selected="yes" value="1">Dense</option>
45       <option value="2">Full</option>
46       <option value="3">Pack</option>
47       <option value="4">Squish</option>
48       <option value="0">Hide</option>
49     </param>
50   </repeat>
51 </inputs>
52 <outputs>
53   <data format="customtrack" name="out_file1" />
54 </outputs>
55 <!--
56 <tests>
57   <test>
58     <param name="primary" value="customTrack1.bed" />
59     <param name="primary_color" value="0-0-0" />
60     <param name="primary_visib" value="1" />
61     <param name="primary_name" value="customTrack1.bed" />
62     <param name="newdata" value="customTrack2.bed" />
63     <param name="status" value="1" />
64     <param name="Color" value="255-0-0" />
65     <param name="Visibility" value="2" />
66     <param name="other names" value="customTrack2.bed" />
-->
```

Line: 87 Column: 8 XML Soft Tabs: 2

**Build custom track**

**Tracks**

**Track 1**

Dataset: 6: UCSC Main on Human: knownGene

name: User Track

description: User Suppli

Color: Black

Visibility: Dense

**Remove Track 1**

**Track 2**

Dataset: 4: UCSC Main on Human: knownGene

name: User Track

description: User Suppli

Color: Magenta

Visibility: Full

**Remove Track 2**

**Add new Track**

**Execute**

#### Info

This tool displays the selected datasets with their custom track attributes (if any) in the UCSC genome browser.

This tool allows you to set the **Color** and **Visibility** attributes and you can edit the **Name** attribute of the dataset by clicking on "edit attributes" button (pencil icon) next to the dataset name in the history panel.

Please note that the primary dataset in step 1 of the tool sets the database build for the datasets in following steps. For example, if your first dataset belongs to hg18, you will only be able to select hg18 associated datasets on the next step.

```

29 <inputs>
30   <repeat name="tracks" title="Track">
31     <param name="input" type="data" format="interval,wig" label="D
32     <param name="name" type="text" size="15" value="User Track">
33       <validator type="length" max="15"/>
34     </param>
35     <param name="description" type="text" value="User Supplied Tra
36       <validator type="length" max="60"/>
37     </param>
38     <param label="Color" name="color" type="select">
39       <option selected="yes" value="0-0-0">Black</option>
40       <option value="255-0-0">Red</option>
41       <option value="0-255-0">Green</option>
42       <option value="0-0-255">Blue</option>
43       <option value="255-0-255">Magenta</option>
44       <option value="0-255-255">Cyan</option>
45       <option value="255-215-0">Gold</option>
46       <option value="160-32-240">Purple</option>
47       <option value="255-140-0">Orange</option>
48       <option value="255-20-147">Pink</option>
49       <option value="92-51-23">Dark Chocolate</option>
50       <option value="85-107-47">Olive green</option>
51     </param>
52     <param label="Visibility" name="visibility" type="select">
53       <option selected="yes" value="1">Dense</option>
54       <option value="2">Full</option>
55       <option value="3">Pack</option>
56       <option value="4">Squish</option>
57       <option value="0">Hide</option>
58     </param>
59   </repeat>
60   <outputs>
61     <data format="customtrack" name="out_file1" />
62   </outputs>
63   <!--
64   <tests>
65     <test>
66       <param name="primary" value="customTrack1.bed" />
67       <param name="primary_color" value="0-0-0" />
68       <param name="primary_visib" value="1" />
69       <param name="primary_name" value="customTrack1.bed" />
70       <param name="newdata" value="customTrack2.bed" />
71       <param name="status" value="1" />
72       <param name="Color" value="255-0-0" />
73     </test>
74   </tests>
75 -->
76 
```

# Repeating sets of parameters

### build\_ucsc\_custom\_track.xml

```
1 <tool id="build_ucsc_custom_track_1" name="Build custom track">
2   <description>for UCSC genome browser</description>
3   <command interpreter="python2.4">
4     build_ucsc_custom_track.py
5     "$out_file"
6     #for $t in $tracks
7     "${t.input.file_name}"
8     "${t.input.ext}"
9     #if ${t.input.ext} == "interval"
10    ${t.input.metadata.chromCol},${t.input.metadata.startCol},${t.input.metadata.endCol},${t.input.metadata.strandCol}
11  #else
12    "NA"
13  #end if
14  "${t.name}"
15  "${t.description}"
16  "${t.color}"
17  "${t.visibility}"
18  #end for
19 </command>
20 <inputs>
21   <repeat name="tracks" title="Track">
22     <param name="input" type="data" format="interval,wig" label="Dataset"/>
23     <param name="name" type="text" size="15" value="User Track">
24       <validator type="length" max="15"/>
25     </param>
26     <param name="description" type="text" value="User Supplied Track (from Galaxy)">
27       <validator type="length" max="60"/>
28     </param>
29     <param label="Color" name="color" type="select">
30       <option selected="yes" value="0-0-0">Black</option>
31       <option value="255-0-0">Red</option>
32       <option value="0-255-0">Green</option>
33       <option value="0-0-255">Blue</option>
34       <option value="255-0-255">Magenta</option>
35       <option value="0-255-255">Cyan</option>
36       <option value="255-215-0">Gold</option>
37       <option value="160-32-240">Purple</option>
38       <option value="255-140-0">Orange</option>
39       <option value="255-20-147">Pink</option>
40       <option value="92-51-23">Dark Chocolate</option>
41       <option value="85-107-47">Olive green</option>
42     </param>
```

Template language for building complex command lines

**Tools****Get Data**[Get ENCODE Data](#)[ENCODE Tools](#)[Text Manipulation](#)[Filter and Sort](#)[Join, Subtract and Group](#)[Convert Formats](#)[Extract Features](#)[Fetch Sequences](#)[Fetch Alignments](#)[Get Genomic Scores](#)[Operate on Genomic Intervals](#)[Statistics](#)[Graph/Display Data](#)

- [Histogram](#) of a numeric columns
- [Scatterplot](#) of two numeric columns
- [XY Plot](#)
- [GMAJ](#) Multiple Alignment Viewer
- [Build custom track](#) for UCSC genome browser

[EMBOSS](#)[HYPHY](#)**XY Plot**Plot Title: Label for x axis: Label for y axis: **Series****Series 1**Dataset: Column for x axis: Column for y axis: Series Type: Line Type: Line Color: Line Width: [Remove Series 1](#)**Series 2**Dataset: Column for x axis: Column for y axis: Series Type: Point Type: Point Color: Point Scale: [Remove Series 2](#)[Add new Series](#)[Execute](#)**History (options)**[refresh](#) | [collapse all](#)**7: Homo sapiens genes (NCBI36)** [\(edit\)](#) [X](#)**6: UCSC Main on Human: knownGene (genome)** [\(edit\)](#) [X](#)**5: Intersect on data 3 and data 4** [\(edit\)](#) [X](#)**4: UCSC Main on Human: knownGene (genome)** [\(edit\)](#) [X](#)**3: Pasted Entry** [\(edit\)](#) [X](#)

**XY Plot**

Plot Title: Sample Plot

Label for x axis: Distance

Label for y axis: Count

---

**Series**

**Series 1**

Dataset: 5: Intersect on data 3 and data 4

Column for x axis: 1

Column for y axis: 2

Series Type: Line

Line Type: Solid

Line Color: Black

Line Width: 1.0

[Remove Series 1](#)

**Series 2**

Dataset: 7: Homo sapiens genes (NCBI36)

Column for x axis: 1

Column for y axis: 1

Series Type: Points

Point Type: Circle (hollow)

Point Color: Black

Point Scale: 1.0

[Remove Series 2](#)

[Add new Series](#)

[Execute](#)

xy\_plot.xml

```

<inputs>
  <param name="main" type="text"
         value="" size="30"
         label="Plot Title"/>
  <param name="xlab" type="text"
         value="" size="30"
         label="Label for x axis"/>
  <param name="ylab" type="text"
         value="" size="30"
         label="Label for y axis"/>
  <repeat name="series" title="Series">
    <param name="input"
          type="data" format="tabular"
          label="Dataset"/>
    <param name="xcol" type="integer"
          value="1" size="30"
          label="Column for x axis"/>
    <param name="ycol" type="integer"
          value="1" size="30"
          label="Column for y axis"/>
    <conditional name="series_type">
      <param name="type" type="select" label="Series Type">
        <option value="line" selected="true">Line</option>
        <option value="points">Points</option>
      </param>
      <when value="line">
        <param name="lty" type="select" label="Line Type">
          <option value="1">Solid</option>
          <option value="2">Dashed</option>
          <option value="3">Dotted</option>
        </param>
        <param name="col" type="select" label="Line Color">
          <option value="1">Black</option>
          <option value="2">Red</option>
          <option value="3">Green</option>
          <option value="4">Blue</option>
          <option value="5">Cyan</option>
          <option value="6">Magenta</option>
          <option value="7">Yellow</option>
          <option value="8">Gray</option>
        </param>
        <param name="lwd" type="float" label="Line Width" value="1.0"/>
      </when>
    </conditional>
  </repeat>
</inputs>
```

Line: 146 Column: 1 XML Soft Tabs: 2

**XY Plot**

Plot Title: Sample Plot

Label for x axis: Distance

Label for y axis: Count

**Series**

**Series 1**

Dataset: 5: Intersect on data 3 and data 4

Column for x axis: 1

Column for y axis: 2

Series Type: Line

Line Type: Solid

Line Color: Black

Line Width: 1.0

[Remove Series 1](#)

**Series 2**

Dataset: 7: Homo sapiens genes (NCBI36)

Column for x axis: 1

Column for y axis: 1

Series Type: Points

Point Type: Circle (hollow)

Point Color: Black

Point Scale: 1.0

[Remove Series 2](#)

[Add new Series](#)

[Execute](#)

```

xy_plot.xml
<repeat name="series" title="Series">
  <param name="input" type="data" format="tabular" label="Dataset"/>
  <param name="xcol" type="integer" value="1" size="30" label="Column for x axis"/>
  <param name="ycol" type="integer" value="1" size="30" label="Column for y axis"/>
  <conditional name="series_type">
    <param name="type" type="select" label="Series Type">
      <option value="line" selected="true">Line</option>
      <option value="points">Points</option>
    </param>
    <when value="line">
      <param name="lty" type="select" label="Line Type">
        <option value="1">Solid</option>
        <option value="2">Dashed</option>
        <option value="3">Dotted</option>
      </param>
      <param name="col" type="select" label="Line Color">
        <option value="1">Black</option>
        <option value="2">Red</option>
        <option value="3">Green</option>
        <option value="4">Blue</option>
        <option value="5">Cyan</option>
        <option value="6">Magenta</option>
        <option value="7">Yellow</option>
        <option value="8">Gray</option>
      </param>
      <param name="lwd" type="float" label="Line Width" value="1.0" />
    </when>
    <when value="points">
      <param name="pch" type="select" label="Point Type">
        <option value="1">Circle (hollow)</option>
        <option value="2">Triangle (hollow)</option>
        <option value="3">Cross</option>
        <option value="4">Diamond (hollow)</option>
        <option value="15">Square (filled)</option>
        <option value="16">Circle (filled)</option>
        <option value="17">Triangle (filled)</option>
      </param>
    </when>
  </conditional>
</repeat>

```

Line: 146 Column: 1 XML Soft Tabs: 2

**XY Plot**

Plot Title: Sample Plot

Label for x axis: Distance

Label for y axis: Count

---

**Series**

**Series 1**

Dataset: 5: Intersect on data 3 and data 4

Column for x axis: 1

Column for y axis: 2

Series Type: Line

Line Type: Solid

Line Color: Black

Line Width: 1.0

[Remove Series 1](#)

**Series 2**

Dataset: 7: Homo sapiens genes (NCBI36)

Column for x axis: 1

Column for y axis: 1

Series Type: Points

Point Type: Circle (hollow)

Point Color: Black

Point Scale: 1.0

[Remove Series 2](#)

[Add new Series](#)

[Execute](#)

```

    <repeat name="series" title="Series">
      <param name="input" type="data" format="tabular" label="Dataset"/>
      <param name="xcol" type="integer" value="1" size="30" label="Column for x axis"/>
      <param name="ycol" type="integer" value="1" size="30" label="Column for y axis"/>
    <conditional name="series_type">
      <param name="type" type="select" label="Series Type">
        <option value="line" selected="true">Line</option>
        <option value="points">Points</option>
      </param>
      <when value="line">
        <param name="lty" type="select" label="Line Type">
          <option value="1">Solid</option>
          <option value="2">Dashed</option>
          <option value="3">Dotted</option>
        </param>
        <param name="col" type="select" label="Line Color">
          <option value="1">Black</option>
          <option value="2">Red</option>
          <option value="3">Green</option>
          <option value="4">Blue</option>
          <option value="5">Cyan</option>
          <option value="6">Magenta</option>
          <option value="7">Yellow</option>
          <option value="8">Gray</option>
        </param>
        <param name="lwd" type="float" label="Line Width" value="1.0" />
      </when>
      <when value="points">
        <param name="pch" type="select" label="Point Type">
          <option value="1">Circle (hollow)</option>
          <option value="2">Triangle (hollow)</option>
          <option value="3">Cross</option>
          <option value="4">Diamond (hollow)</option>
          <option value="15">Square (filled)</option>
          <option value="16">Circle (filled)</option>
          <option value="17">Triangle (filled)</option>
        </param>
      </when>
    </conditional>
  </repeat>

```

Conditional groups, grouping constructs can be nested

```
xy_plot.xml
1 <tool id="XY_Plot_1" name="XY Plot">
2   <description> of two numeric columns</description>
3   <command interpreter="bash">r_wrapper.sh $script_file</command>
4
5   <inputs>
6     <param name="main" type="text"
7       value="" size="30"
8       label="Plot Title"/>
9     <param name="xlab" type="text"
10      value="" size="30"
11      label="Label for x axis"/>
12     <param name="ylab" type="text"
13      value="" size="30"
14      label="Label for y axis"/>
15   <repeat name="series" title="Series">
16     <param name="input"
17       type="data" format="tabular"
18       label="Dataset"/>
19     <param name="xcol" type="integer"
20       value="1" size="30"
21       label="Column for x axis"/>
22     <param name="ycol" type="integer"
23       value="1" size="30"
24       label="Column for y axis"/>
25     <conditional name="series_type">
26       <param name="type" type="select" label="Series Type">
27         <option value="line" selected="true">Line</option>
28         <option value="points">Points</option>
29       </param>
30       <when value="line">
31         <param name="lty" type="select" label="Line Type">
32           <option value="1">Solid</option>
33           <option value="2">Dashed</option>
34           <option value="3">Dotted</option>
35         </param>
36         <param name="col" type="select" label="Line Color">
37           <option value="1">Black</option>
38           <option value="2">Red</option>
39           <option value="3">Green</option>
40           <option value="4">Blue</option>
41           <option value="5">Cyan</option>
42           <option value="6">Magenta</option>
```

Command line tool expects a configuration file

```

70      </conditional>
71    </repeat>
72  </inputs>
73
74  <configfiles>
75    <configfile name="script_file">
76      ## Setup R error handling to go to stderr
77      options( show.error.messages=F,
78                error = function () { cat( geterrmessage(), file=stderr() ); q( "no", 1, F ) } )
79      ## Determine range of all series in the plot
80      xrange = c( NULL, NULL )
81      yrange = c( NULL, NULL )
82      #for $i, $s in enumerate( $series )
83      s${i} = read.table( "${s.input.file_name}" )
84      x${i} = s${i}[, ${s.xcol}]
85      y${i} = s${i}[, ${s.ycol}]
86      xrange = range( x${i}, xrange )
87      yrange = range( y${i}, yrange )
88    #end for
89    ## Open output PDF file
90    pdf( "${out_file1}" )
91    ## Dummy plot for axis / labels
92    plot( NULL, type="n", xlim=xrange, ylim=yrange, main="${main}", xlab="${xlab}", ylab="${ylab}" )
93    ## Plot each series
94    #for $i, $s in enumerate( $series )
95    #if ${s.series_type['type']} == "line"
96      lines( x${i}, y${i}, lty=${s.series_type.lty}, lwd=${s.series_type.lwd}, col=${s.series_type.col} )
97    #elif ${s.series_type.type} == "points"
98      points( x${i}, y${i}, pch=${s.series_type.pch}, cex=${s.series_type.cex}, col=${s.series_type.col} )
99    #end if
100   #end for
101   ## Close the PDF file
102   devname = dev.off()
103   </configfile>
104 </configfiles>
105
106 <outputs>
107   <data format="pdf" name="out_file1" />
108 </outputs>
109
110 <help>
111 .. class:: infomark

```

Configuration file is generated based on user input

# Job execution in **Galaxy**

# Flexible execution environment

- Dependencies between jobs handled by “JobManager” within **Galaxy**.
  - Either in-process with the web application, or a separate process managing a queue to which multiple front-ends submit

# Flexible execution environment

- Once jobs are ready, submitted to a “JobRunner”
  - Runners are pluggable
  - Can have multiple runners, and jobs to different runners depending on capabilities
- Current implementations:
  - Local runner executing a limited number of local processes
  - PBS runner dispatches to a cluster of worker nodes
  - Pluggable queueing policies

# Core tools

# Genomic interval analysis

- Set-like operations on intervals, base-level and interval level
  - Merge, intersect, subtract...
  - Interval clustering
- Relational-like operations
  - Join, group
- Data structures and high-level Python interfaces to all operations available as part of “bx-python”

# Genomic alignment analysis

- Extracting features of interest from pairwise and multiple genome wide alignments
  - Dealing with gene / transcript structure
- Filtering alignments in many ways
- Tools for indexing alignments, fast random access, and all operations available in “bx-python”

# Phylogenomic tools

- Built on top of HyPhy (<http://hyphy.org>)
  - Phylogenetic tree reconstruction
  - Selection detection
  - Hypothesis testing
  - Relative rate tests
  - Detecting recombination

# Statistical genetics

- Built with RGenetics (<http://rgenetics.org>)
  - Experiment design (including power and sample size calculations)
  - Quality control and filtering
  - Exploration of and adjustment for population substructure
  - Linkage disequilibrium visualization, data reduction based on LD (tag SNP identification)
  - Inference for pedigree and unrelated subject data

# Metagenomics

- Mapping “reads” onto protein databases
  - Need to figure out how to do this a lot faster!
- Visualizing
  - On the global phylogeny (MEGAN style)
- ...?

*Deeper customization of Galaxy*

# Galaxy

**Specialty tools**

- aggregate\_score Aggregate genomic scores
- alignability Alignability
- liftOver local liftover
- blast blast

**Motifs**

**Database**

**Get Data**

**Get ENCODE Data**

**ENCODE Tools**

**Edit Queries**

**Filter, Sort, Join and Compare**

**Convert Formats**

**Fetch Sequences and Alignments**

**Alignment Viewers**

**Get Genomic Scores**

**Operate on Genomic Intervals**

**Statistics**

**Graph Data**

**EMBOSS**

**PHYLIP**

**PAML**

**HYPHY**

Info: report bugs | wiki | screencasts Account: create | login

aggregate\_score

Bed file: 2: alignability on data 1

Data:

- esperr:hg18
- esperr:mm8
- esperr:hg17
- esperr:mm7
- esperr:red\_mm8
- esperr:tmp
- phastcons:mm8\_chr7

Execute

refresh | collapse all

2: alignability on data 1

UCSC Table Browser on Human: encodeRegions (genome)

44 regions, format: bed, database: hg17

Info: UCSC Table Browser on Human: encodeRegions (genome) save | display at UCSC\_main test

1	2	3	4
chr1	147971133	148471133	ENr231
chr2	5157835	5207835	ENr12
chr2	1188010803	118510803	ENr121
chr2	220102850	220602850	ENr331
chr2	234273824	234773888	ENr131
chr4	118604258	119104258	ENr113

History options...

# Galaxy

Info: report bugs | wiki | screencasts Logged in as james@bx.psu.edu: manage | logout

Get Data

Get ENCODE Data

**ENCODE Tools**

- Extract MAF blocks from locally cached alignments
- Gencode Partition partition an interval file
- Random Intervals create a random set of intervals

**Edit Queries**

**Filter, Sort, Join, Compare, Subtract**

**Convert Formats**

**Pattern-Matching**

**Fetch Sequences and Alignments**

**Get Genomic Scores**

**Operate on Genomic Intervals**

**Statistics**

**Graph/Display Data**

**EMBOSS**

**PHYLIP**

**HYPHY**

File to Partition: no data has the proper type

Execute

refresh | collapse all

Your history is empty. Click 'Get Data' on the left pane to start

For detailed information about partitioning, click [here](#).

Datasets are partitioned according to the protocol below:

A partition scheme has been defined that is similar to what has previously been done with TARs/TRANSFRAGS such that any feature can be classified as falling into one of the following 6 categories:

1. Coding -- coding exons defined from the GENCODE experimentally verified coding set (coding in any transcript)
2. SUTR -- 5' UTR exons defined from the GENCODE experimentally verified coding set (5' UTR in some transcript but never coding in any other)
3. 3UTR -- 3' UTR exons defined from the GENCODE experimentally verified coding set (3' UTR in some transcript but never coding in any other)
4. Intronic Proximal -- intronic and no more than 5kb away from an exon.
5. Intergenic Proximal -- between genes and no more than 5kb away from an exon.
6. Intronic Distal -- intronic and greater than 5kb away from an exon.
7. Intergenic Distal -- between genes and greater than 5kb away from an exon.

Note: Features overlapping more than one partition will take the identity of the lower-numbered partition.

Tools

History options...

# Galaxy

Info: report bugs | wiki | screencasts Logged in as james@bx.psu.edu: manage | logout

Tools

**Get Data**

**Get ENCODE Data**

**ENCODE Tools**

**Edit Queries**

**Filter, Sort, Join, Compare, Subtract**

**Convert Formats**

**Pattern-Matching**

**Fetch Sequences and Alignments**

**Get Genomic Scores**

**Operate on Genomic Intervals**

**Statistics**

**Graph/Display Data**

**EMBOSS**

**PHYLIP**

**HYPHY**

- Branch Lengths Estimation
- Neighbor Joining Tree Builder

Branch Lengths

Fasta file: no data has the proper type

Tree Definition: For example: ((hg17,panTro1),(mm5,rn3),canFam1)

Substitution Model: F81

Base Frequencies: Nucleotide frequencies collected from the d

Execute

This tool takes a single or multiple FASTA alignment file and estimates branch lengths using HYPHY, a maximum likelihood analyses package.

For the tree definition, you only need to specify the species build names. For example, you could use the tree ((hg17,panTro1),(mm5,rn3),canFam1), if your FASTA file looks like this:

```
>hg17 Chr(+) : 26907301-26907310|hg17_0
GTGGGAGCTT
>panTro1 Chr(6+) : 28837319-28837328|panTro1_0
GTGGGAGCTT
>mm5 Chr(6+) : 52104822-52104831|mm5_0
>rn3 Chr(4+) : 88734395-88734484|rн3_0
GTGGGAGCTT
>canFam1 Chr(14+) : 42826409-42826418|canFam1_0
GTGGGAGCTT

>hg17 Chr(+) : 26907326-26907326|hg17_1
AGTCAGATGTCAG
>panTro1 Chr(6+) : 28837328-28837344|panTro1_1
AGTCAGATGTCAG
>mm5 Chr(6+) : 52104821-52104847|mm5_1
AGTCAGATGTCAG
>rn3 Chr(4+) : 88734484-88734428|rн3_1
AGTCAGATGTCAG
>canFam1 Chr(14+) : 42826418-42826434|canFam1_1
AGTCAGATGTCAG

>hg17 Chr(+) : 26907326-26907338|hg17_2
GTAGAGACCC
>panTro1 Chr(6+) : 28837344-28837356|panTro1_2
GTAGAGACCC
>mm5 Chr(6+) : 52104847-52104859|mm5_2
GTAGAGACCC
>rn3 Chr(4+) : 88734428-88734432|rн3_2
GTAGAGACCC
>canFam1 Chr(14+) : 42826434-42826446|canFam1_2
GTAGAGACCC

>hg17 Chr(+) : 26907338-26907341|hg17_3
GGGGAGGAAAGCAAGGGGAAGACCTGACTCTTGAGAT---TCTTCGGCCCTCTCGT----CGTTTCTGG----CGGGGGTGGC
>panTro1 Chr(6+) : 28837356-28837672|panTro1_3
GGGGAGGAAAGCAAGGGGAAGACCTGACTCTTGAGAT---TCTTCGGCCCTCTCGT----CGTTTCTGG----CGGGGGTGGC
>mm5 Chr(6+) : 52104859-52104875|mm5_3
>rn3 Chr(4+) : 88734432-88734478|rн3_3
GGGGAGGAAAGCAAGGGGAAGACCTGAGAT---TCTTCAGTTTCTCAT----CGCTGCCAGG----AGGAGTGGC
>canFam1 Chr(14+) : 42826446-42826762|canFam1_3
```

History options...

# Galaxy at the Channing Laboratory

Info: report bugs | wiki | screencasts Logged in as ross.lazarus@gmail.com: manage | logout

Tools

**Get Data**

**Rgenetics**

- GenomeGraphs UCSC instant track
- Clean: SNP data for QC
- QC reports: Genotype plots and details
- CaseControl Statistical tests
- TDT: Statistical tests
- GLM for genotypes GLM Statistical models and tests
- Flat for family genotypes Pbat Statistical models and tests
- Pbat for family genotypes Pbat Statistical models and tests

**Edit Queries**

**Filter, Sort, Join and Compare**

**Convert Formats**

**Fetch Sequences and Alignments**

**Alignment Viewers**

**Get Genomic Scores**

**Operate on Genomic Intervals**

**Statistics**

**Graph Data**

**EMBOSS**

**PHYLIP**

**PAML**

**HYPHY**

**Get ENCODE Data**

**ENCODE Tools**

Clean:

Genotype file: /usr/local/galaxy/data/g/plinkbed/camp2007

Cleaned file name: campClean

Max.Missfrac: subjects: 0.05

Max.Missfrac: markers: 0.1

Max.Mendelfrac: Individuals: 0.05

Max.Mendelfrac: Families: 0.05

Smallest HWE p value: -1

SmallestMAF: 0.01

Execute

refresh | collapse all

40: Histogram on data 15

34: myClean.log

24: Import ped/map

20: CaseControl

15: http://meme/~rerla/prefev\_pc\_all\_nocov\_PRE

12: test\_QC\_report.html

4: Sort on data 3

3: TDT\_TDT.xls

2: camp2007RAWqc\_QC\_report.html

11:6 kB

There are 4 secondary datasets.

1: camp2007RAWqc\_marker.xls

2: camp2007RAWqc\_QC\_report.log

3: camp2007RAWqc\_nhet.xls

4: camp2007RAWqc\_subject.xls

Summary

This tool imports genotype data from a high throughput snp platform into Galaxy. Data in linkage (ped and map files) format are currently importable. Others will be added. Filters are available to remove markers below a specific minor allele frequency, or above a specific level of missingness, and to remove subjects using similar criteria. Subjects and markers for family data can be filtered by proportions of Mendelian errors in observed transmission. Use the QC reporting tool to generate a comprehensive series of reports for quality control.

Originally designed and written for family based data from the CAMP Illumina run

History options...

# Galaxy web interface is easily customized / branded

# Datatypes

- Datatypes supported by a Galaxy instance can be configured at runtime
- Declarative definition of “metadata”
  - Easy way to define custom metadata
  - Automatically generated editing interfaces (similar to tool interfaces)
- Actions on datatypes (displaying at external sites, format conversion) all pluggable
- Nothing “genomics” specific hardcoded!

# Reuse and reproducibility

# Sharing histories

- A history in **Galaxy** is a complete record of a complex analysis
  - Histories in **Galaxy** can be easily be shared
  - A shared history is always a copy (the original analysis is always retained)
  - All of the details of any analysis can thus be inspected, rerun, ...

# Workflows

- A series on analysis steps involving the invocation of multiple tools can be stored and reused
- Parameters within the workflow can be set in the workflow, or when the workflow is invoked (like any other tool)
- Support for repetitive invocation of tools and workflows, and aggregation of results
- Saving and sharing of workflows, reproducible!

# Workflow construction

- Explicit workflow construction and editing
- Workflow construction by example
  - Users will continue to build analysis as they do now, and will be able to extraction portions of their histories as reusable workflows
  - Will work for most existing histories! (we've been saving the right data all along)

# Some Technical Details

# Under the hood

- Python 2.4, though some dependencies use CPython specific extensions (database access, tools)
- WSGI Web framework: PythonPaste, Routes, WebHelpers, Beaker, Cheetah, ...
- SQLAlchemy for database abstraction
- ♥ jQuery

# Out of the box configuration

- Just checkout from subversion and run!
- All dependencies packaged as eggs
- Pure python HTTP server included  
(`paste.httpserver`)
- Embedded database (`sqlite`)
- Datasets stored on local filesystem
- Jobs run locally

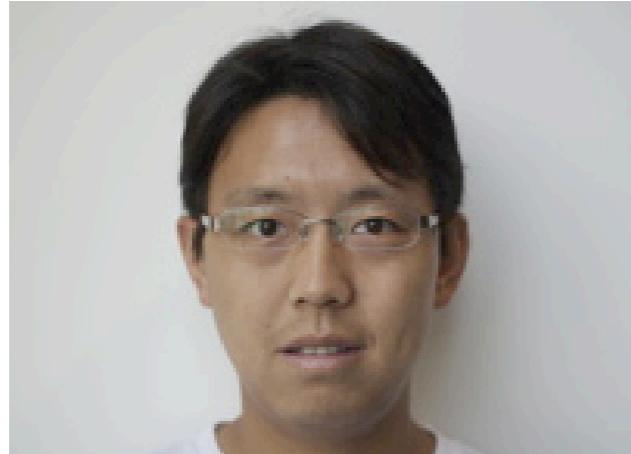
# PSU production configuration

- Deployed behind Apache using mod\_proxy
  - Python threads do not scale across CPUs, we use both forking and threading similar to Apache's worker MPM
- PostgreSQL
- Jobs dispatched to a PBS cluster using “pbs-python”

# Acknowledgements

- **Galaxy** collaborators:
  - Ross Lazarus, Sergei Kosakovsky Pond
  - UCSC Genome Browser team
  - Biomart team
  - National Science Foundation

# The core Galaxy development team

		
Guru Ananda	Dan Blankenberg	Nate Coraor
		
Jianbin He	Greg von Kuster	Anton Nekrutenko
		
Ian Schenck	James Taylor	Yi Zhang