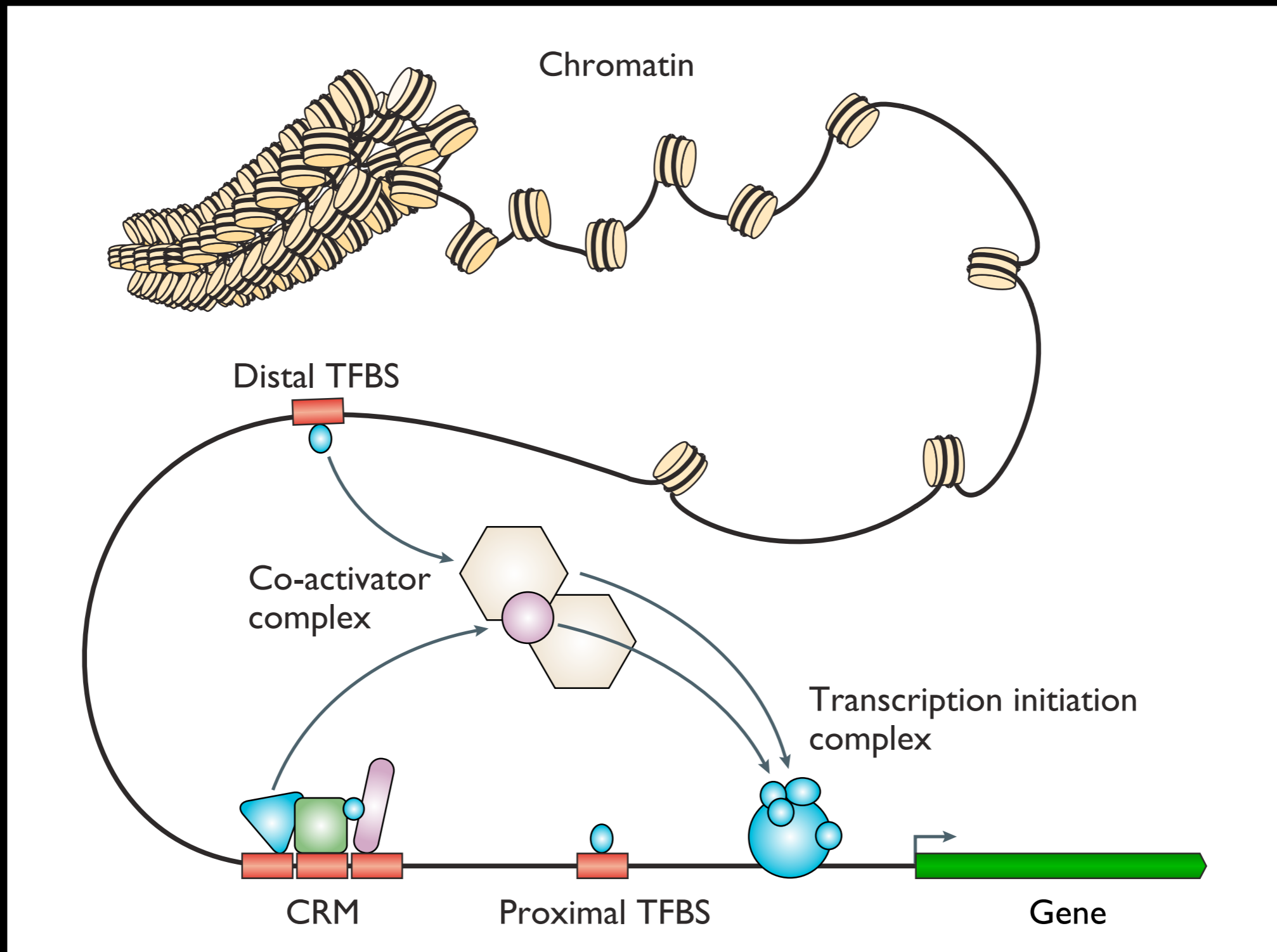


“Regulatory genomics”

Regulation of gene transcription



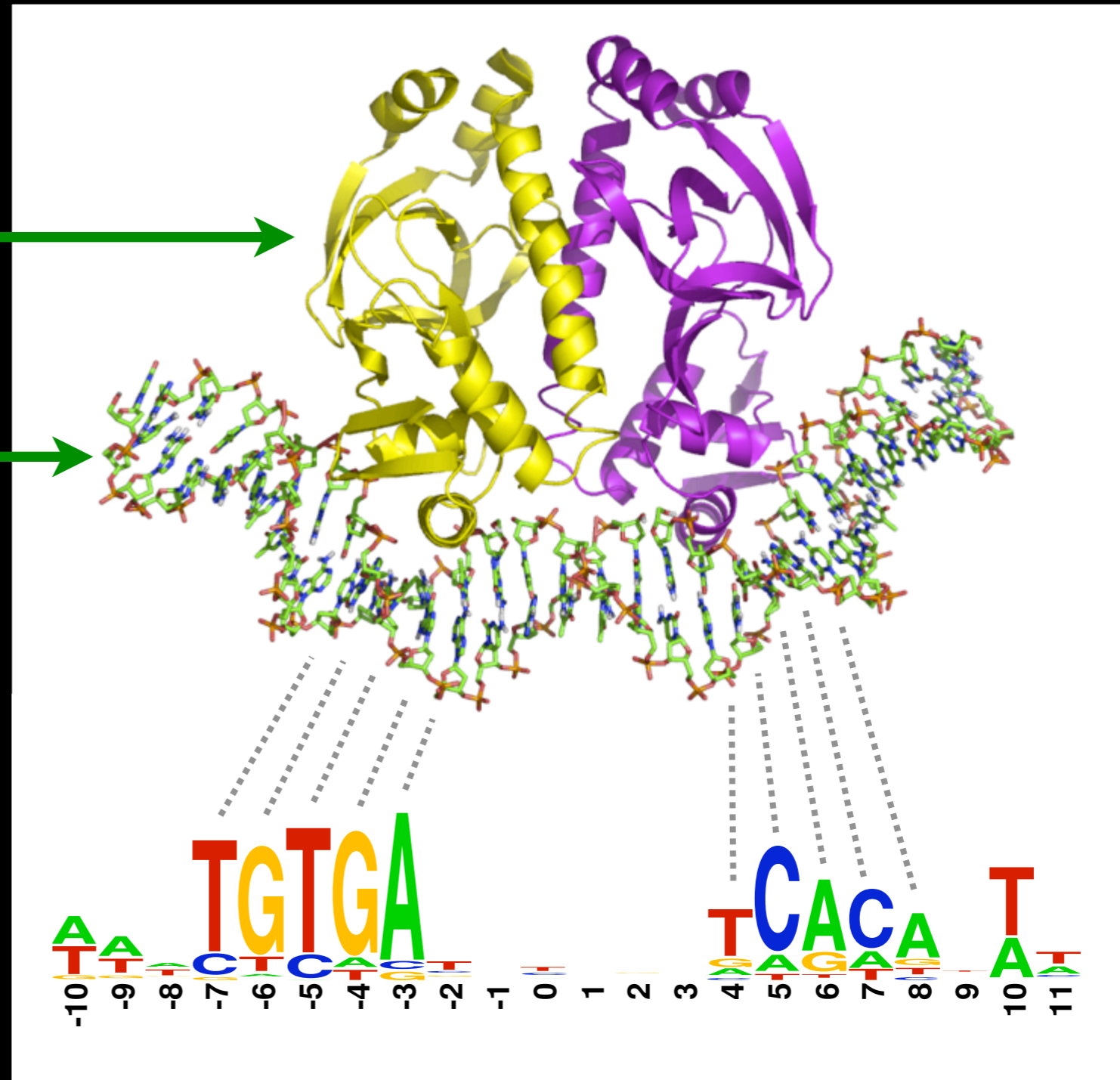
If only it were even *that* simple...

Sequence specific binding yields constraint

CAP protein
(homodimer)

Bound DNA

Motif from 59
bound sites

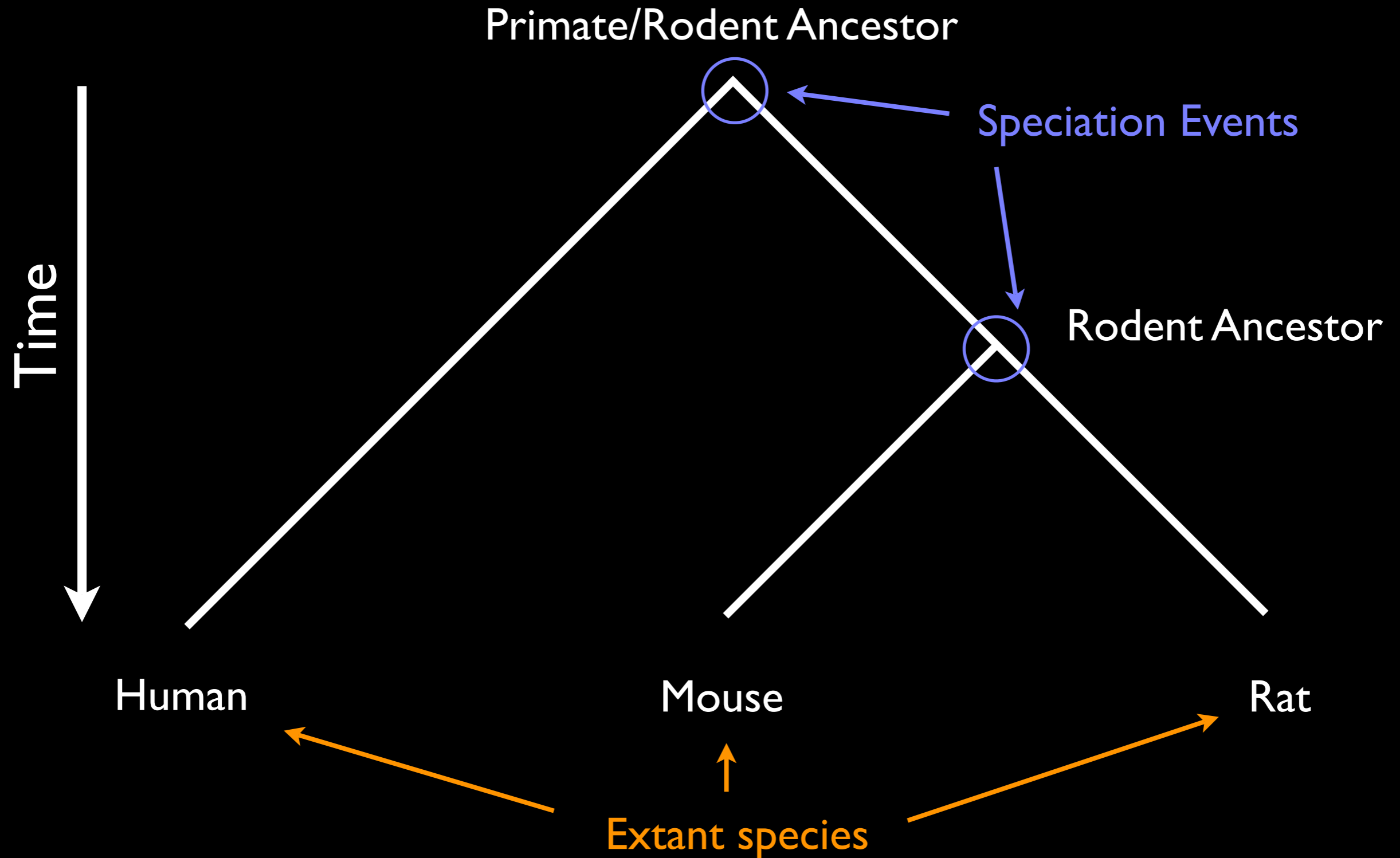


The genomic era...

Comparative

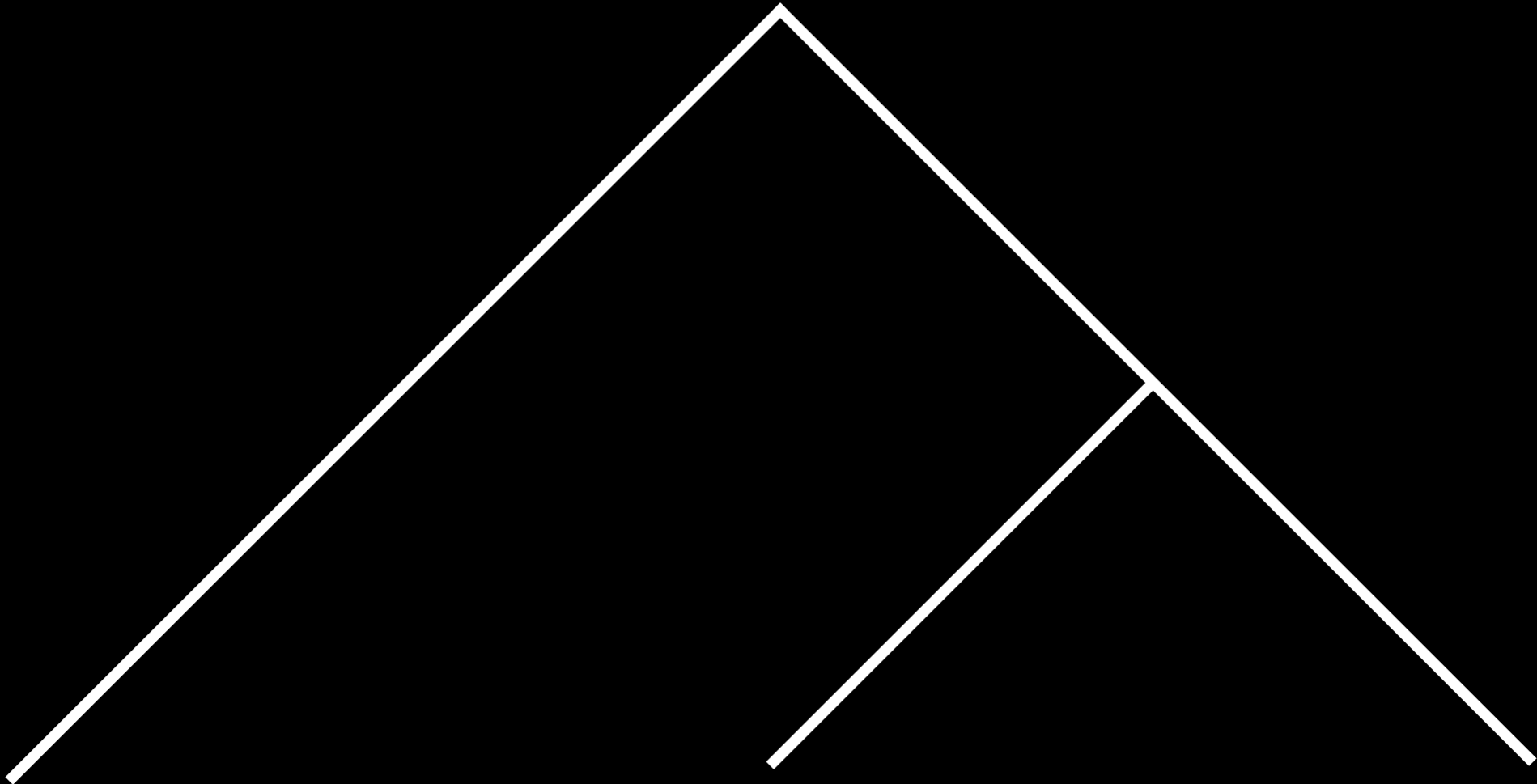


Evolution of functional elements



Evolution of functional elements

CTCCCAGCTGCCC

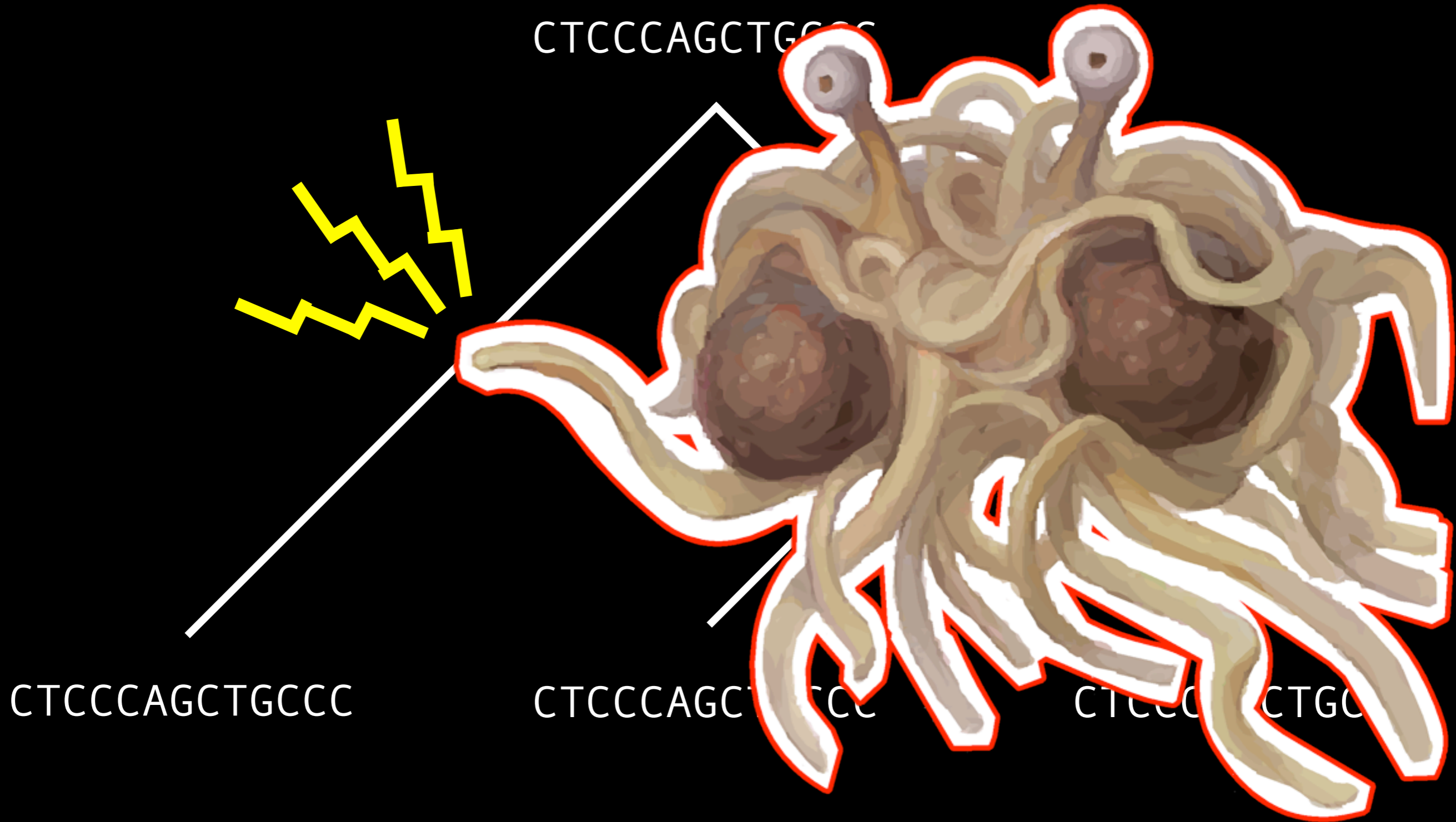


CTCCCAGCTGCCC

CTCCCAGCTGCCC

CTCCCAGCTGCCC

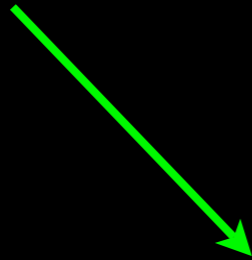
Evolution of functional elements



Evolution of functional elements

CTCCCAGCTGCCC

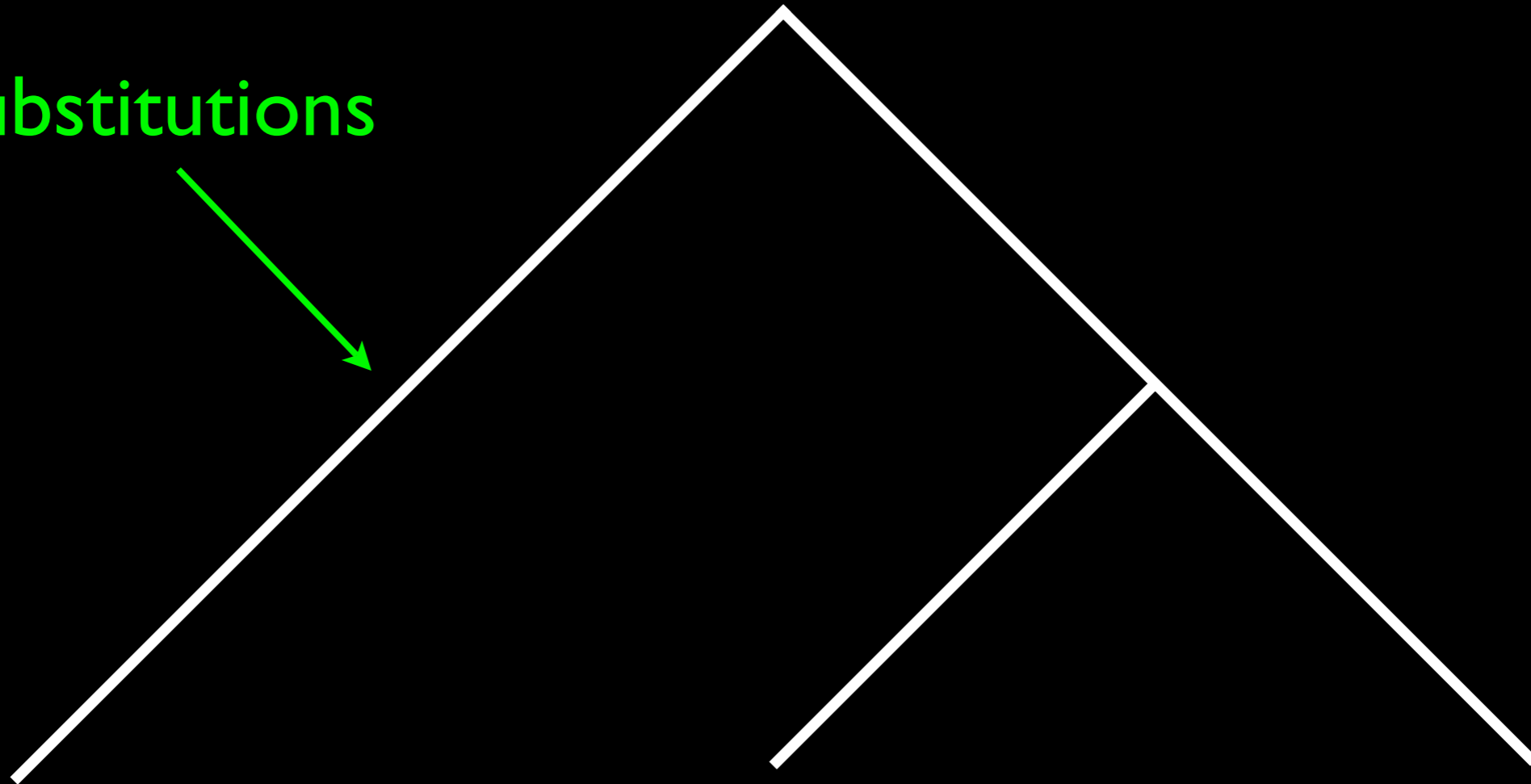
Substitutions



CTCCC**G**GC**A**GCCC

CTCCCAGCTGCCC

CTCCCAGCTGCCC



Evolution of functional elements

CTCCCAGCTGCCC

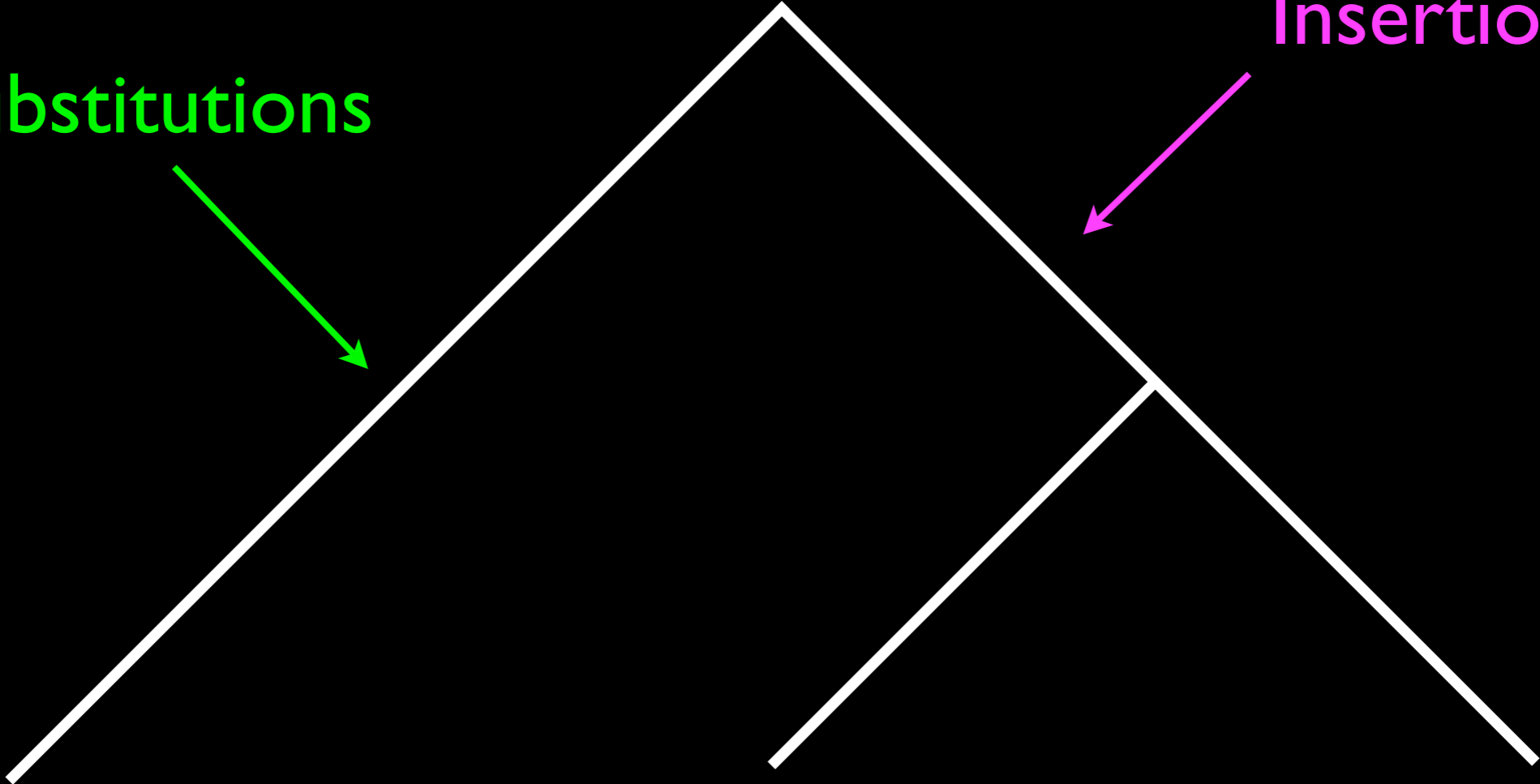
Substitutions

Insertion

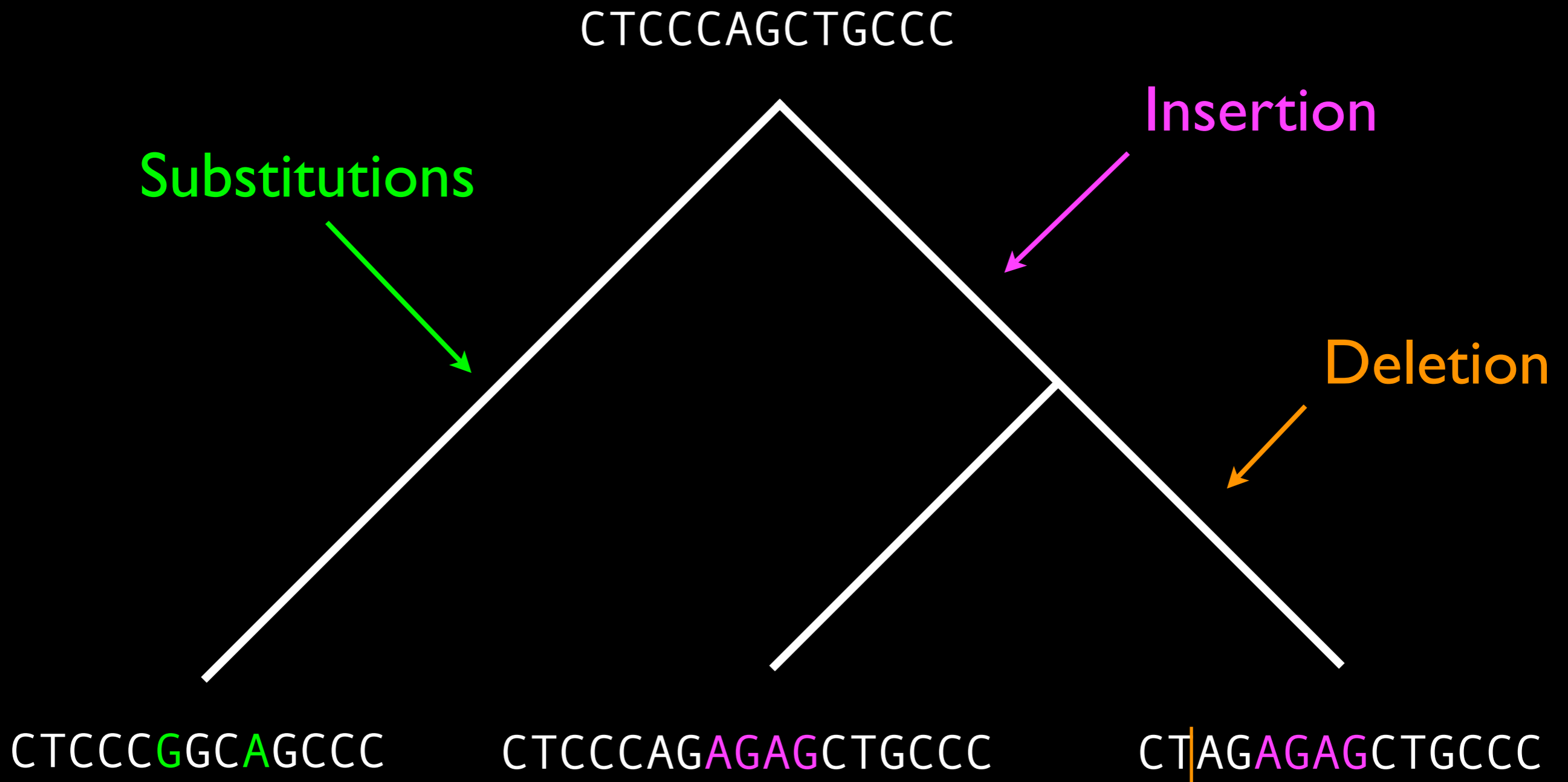
CTCCCGGCAGCCC

CTCCCAGAGAGCTGCCC

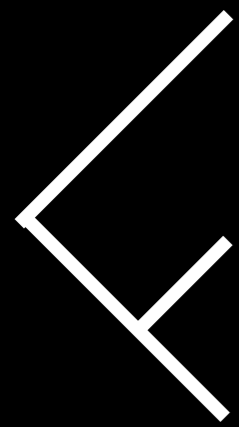
CTCCCAGAGAGCTGCCC



Evolution of functional elements



Sequence alignment



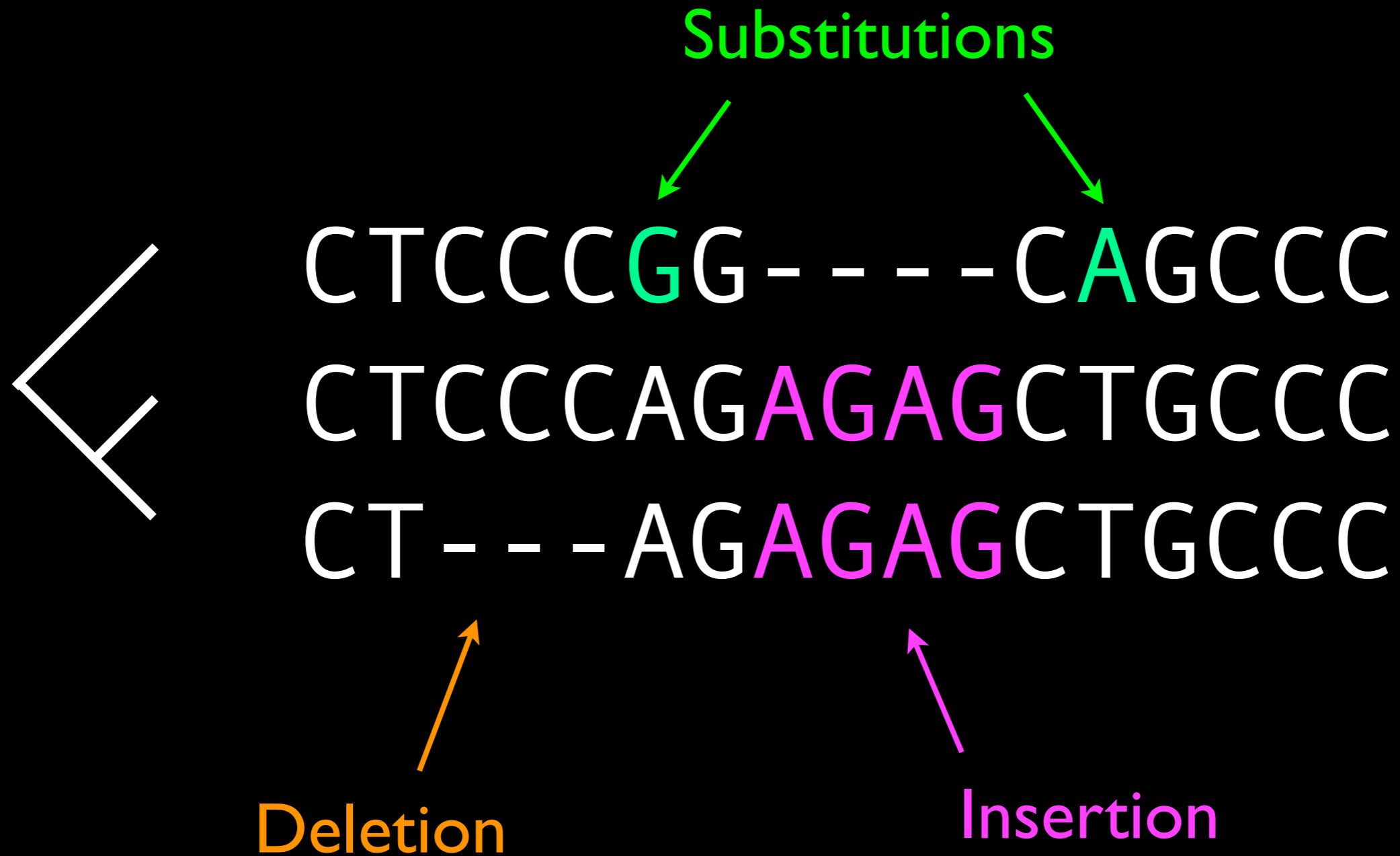
CTCCCGGCAGCCC
CTCCCAGAGAGGCTGCCC
CTAGAGAGGCTGCCC

Sequence alignment



CTCCC GG - - - - CAGCCC
CTCCACAGAGAGCTGCCC
CT - - - AGAGAGCTGCCC

Sequence alignment



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,¹ Svitlana Tyekucheva, David C. King, Ross C. Hardison, Webb Miller, and Francesca Chiaromonte¹

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 sequences 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 DNaseI 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.]

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

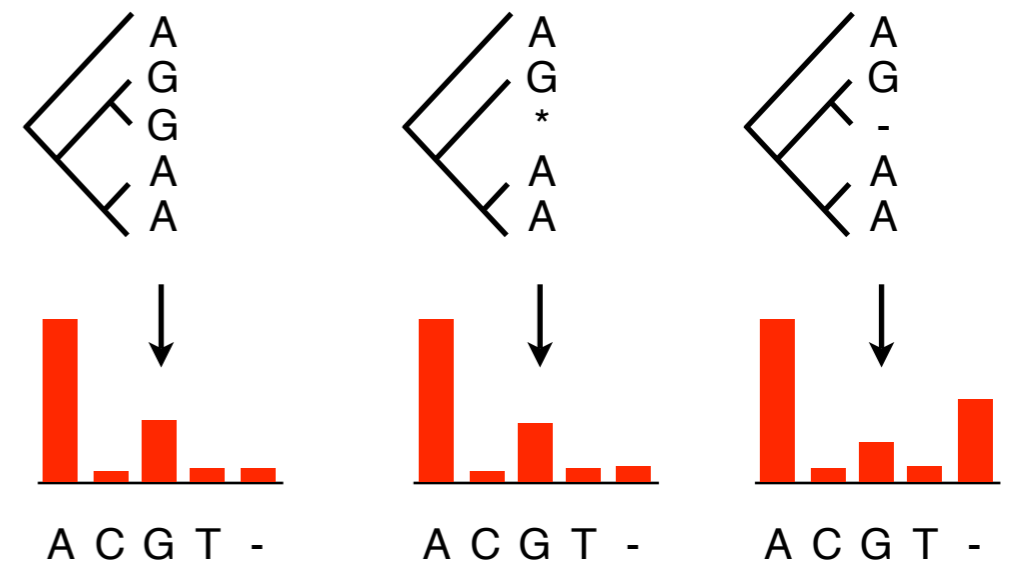
```
CTCCCAGCTGCCCAGTGCCGCCTCTTTTTT  
CTCCTAGCTG-CCAGCATCTCCCGTTTTT  
CTCCCAGCTGCCCTGCGCCTCCTCTTTTTT
```



```
13111021321110232112113133333
```

Ancestral probability distribution

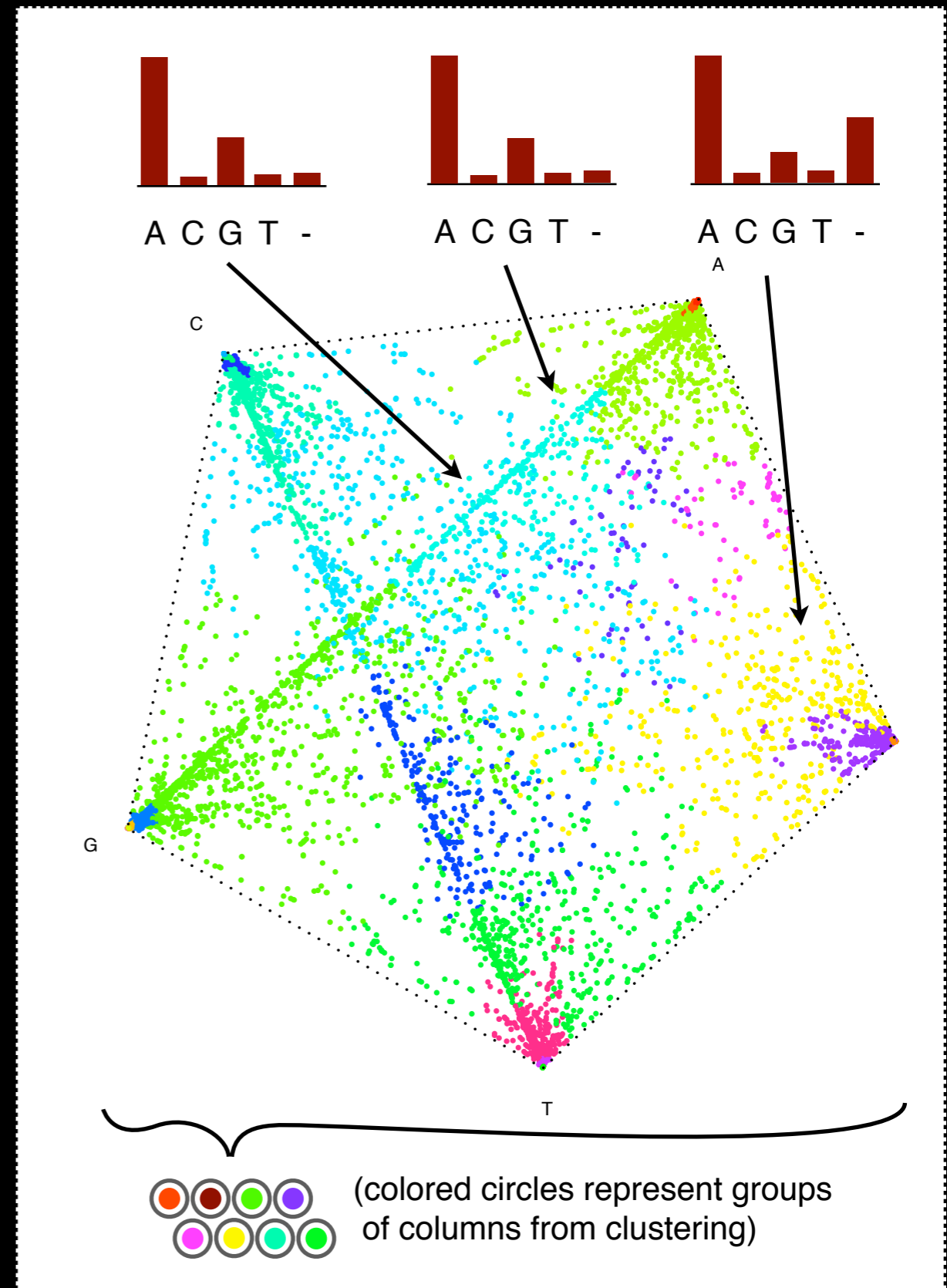
Map each possible column of a multiple alignment to a probability distribution of the nucleotide in that position in the common ancestor.



Clustering spatially and distributionally

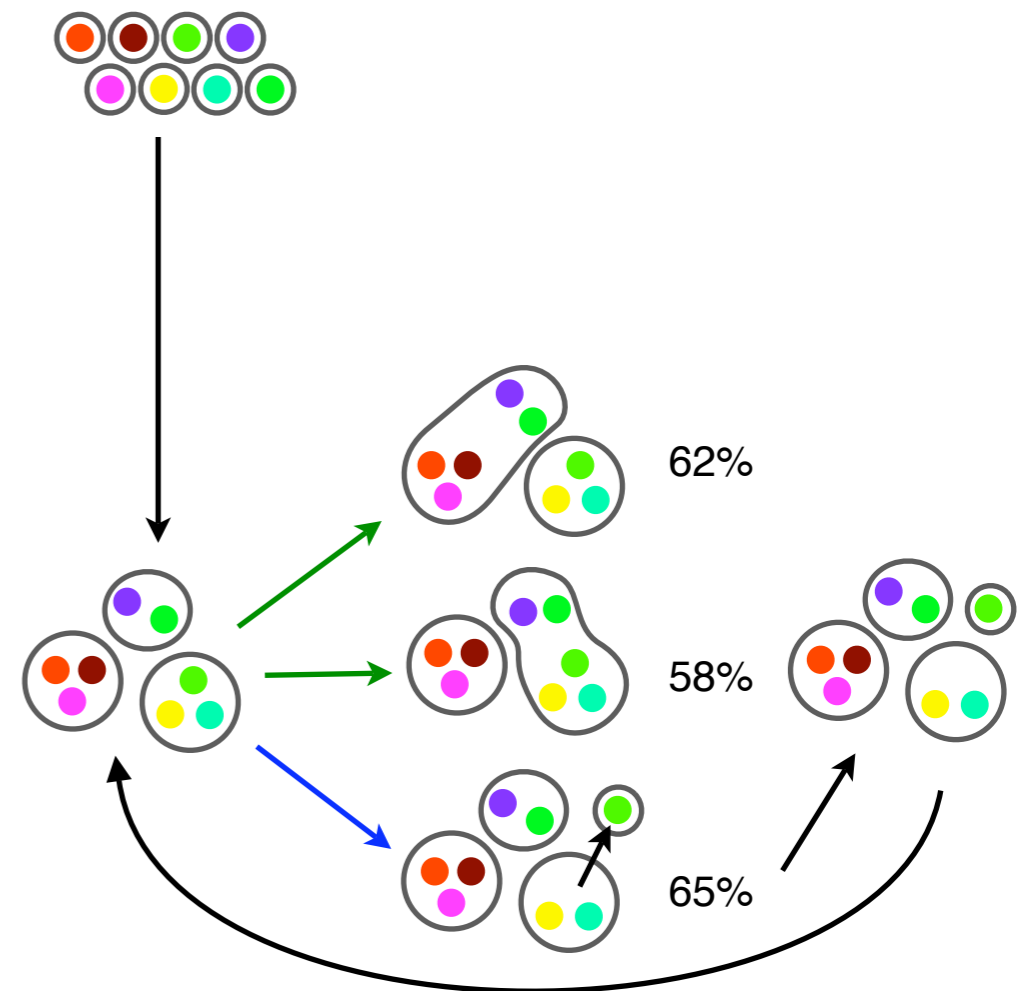
Consider the observed column frequencies as a discrete distribution over the probability simplex, and find a distribution on a smaller number of points that preserves:

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

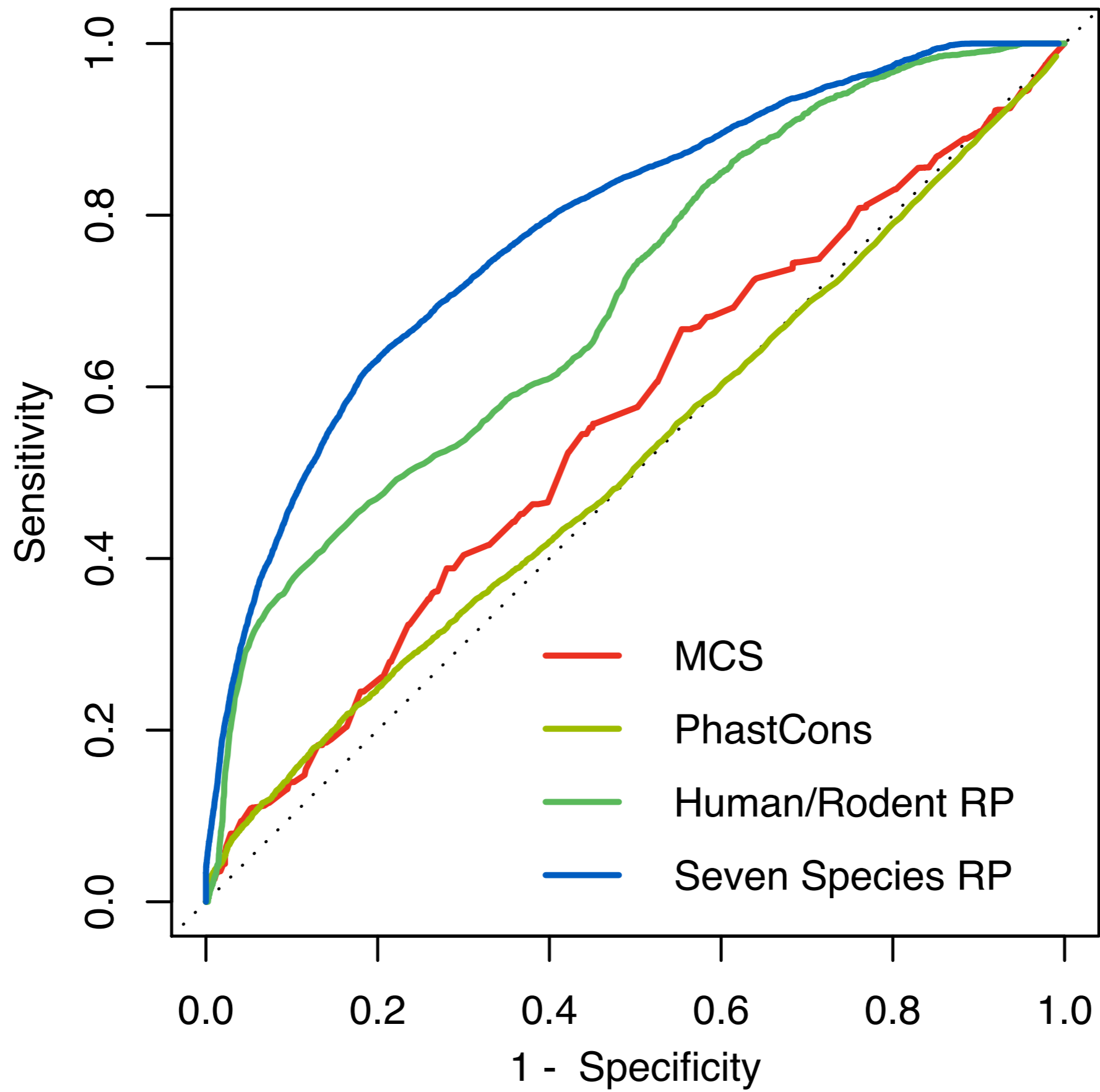


Searching for encodings

- Random / heuristic search through space of possible encodings



Some validation



chr11: 5255000 5260000 5265000 5270000

HBE1_PRA HBE1_NRA HS1 HS2_pos HS2_neg HS3 HS3.1 HS3.2 HS4 HS5

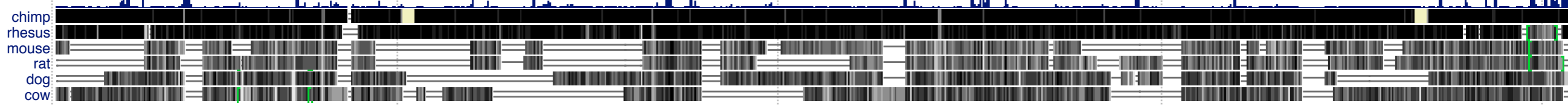
Compilation of Landmarks from Locus Experts

ESPERR Regulatory Potential (7 species)

Human/Mouse/Rat RP Scores, Kolbe et al model

Vertebrate Multiz Alignment & Conservation

phastCons

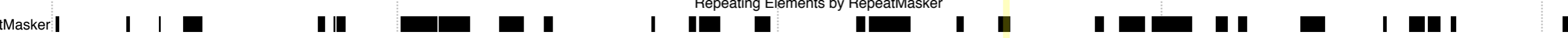
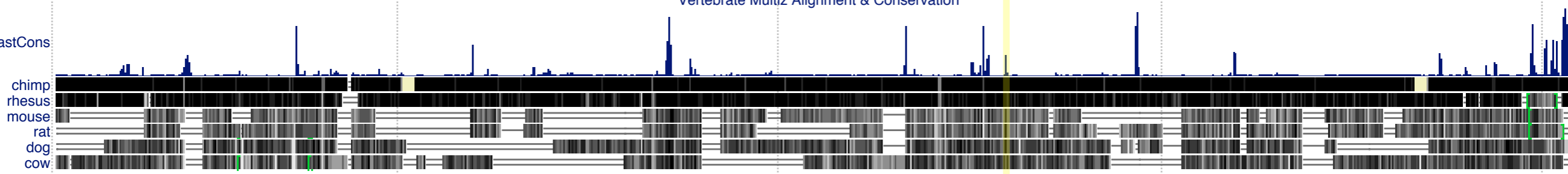
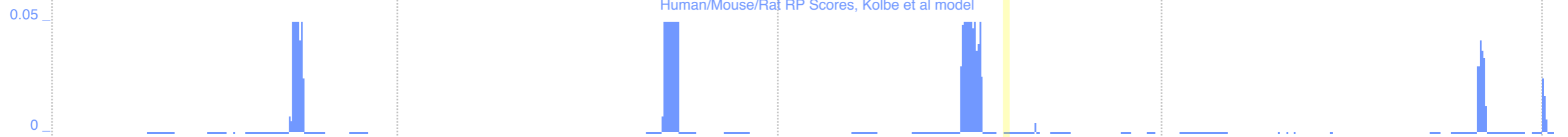
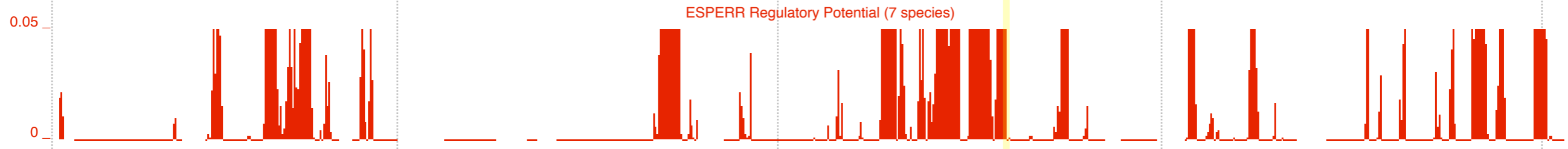


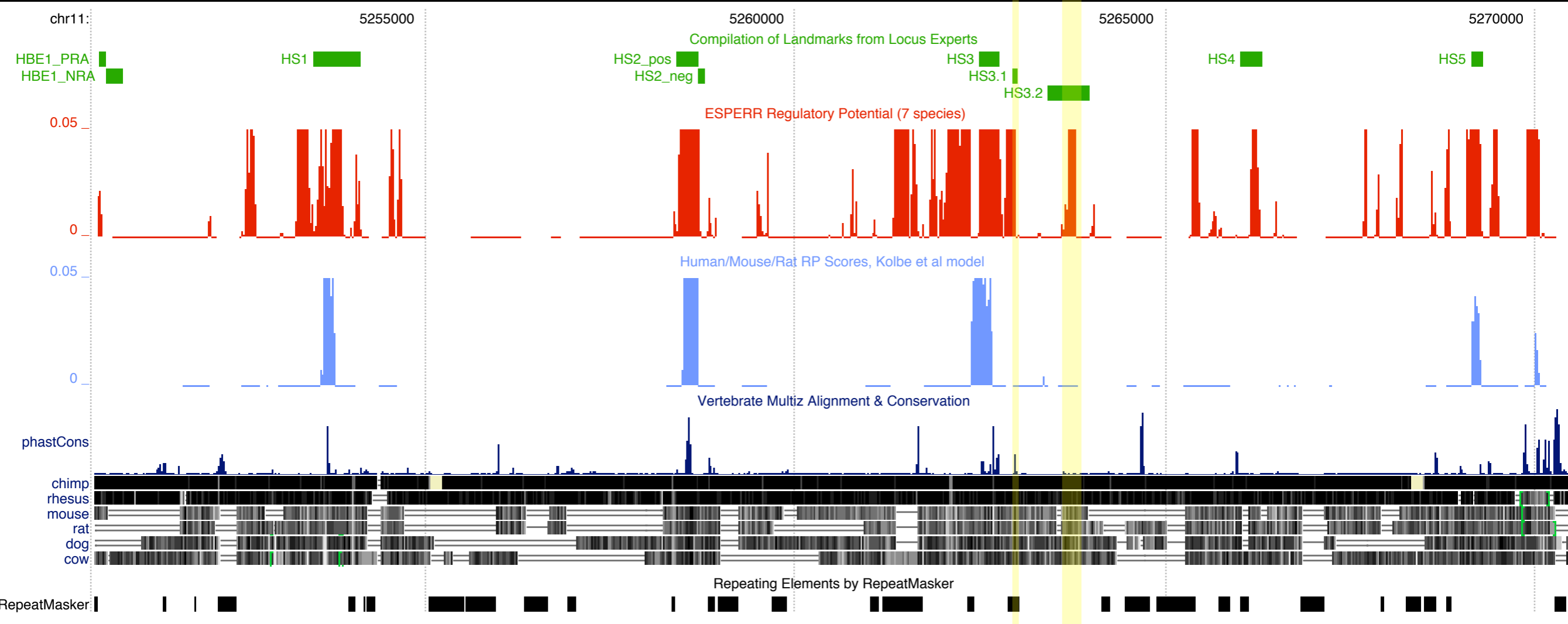
Repeating Elements by RepeatMasker

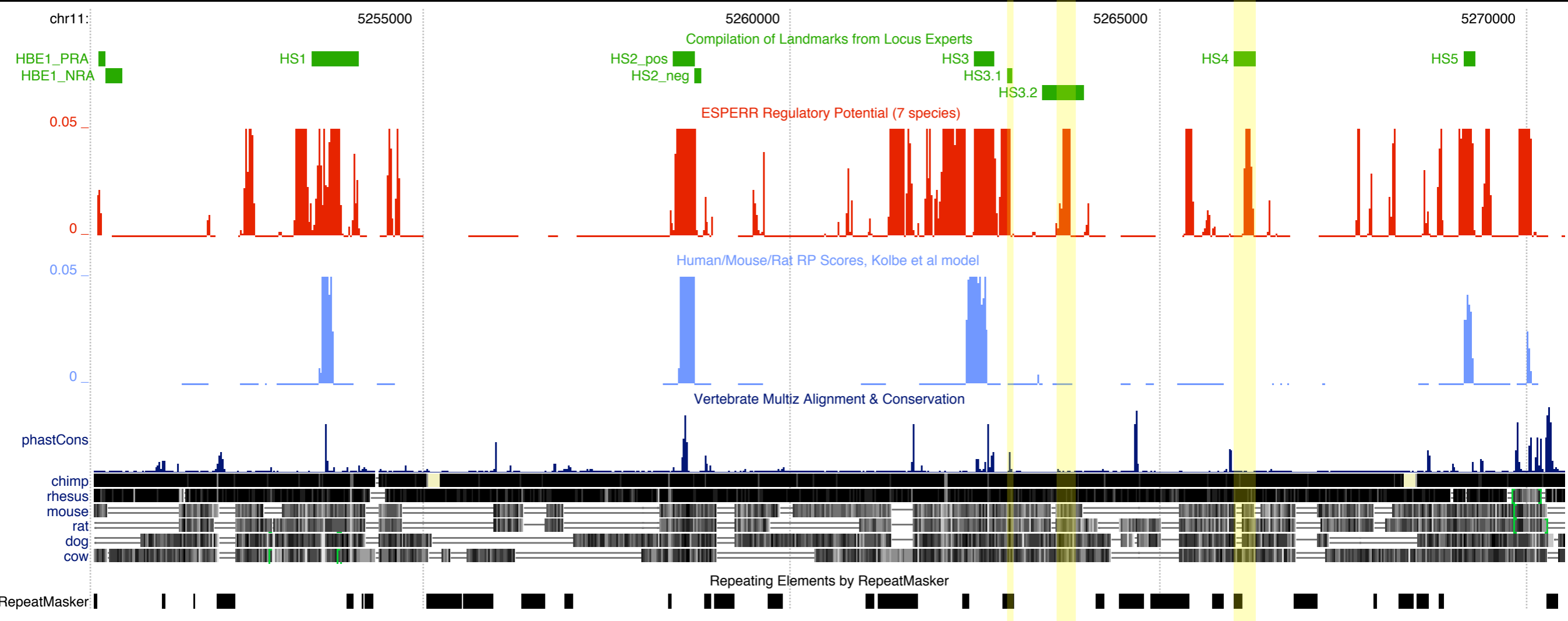


chr11: 5255000 5260000 5265000 5270000

HBE1_PRA HBE1_NRA HS1 HS2_pos HS2_neg HS3 HS3.1 HS3.2 HS4 HS5



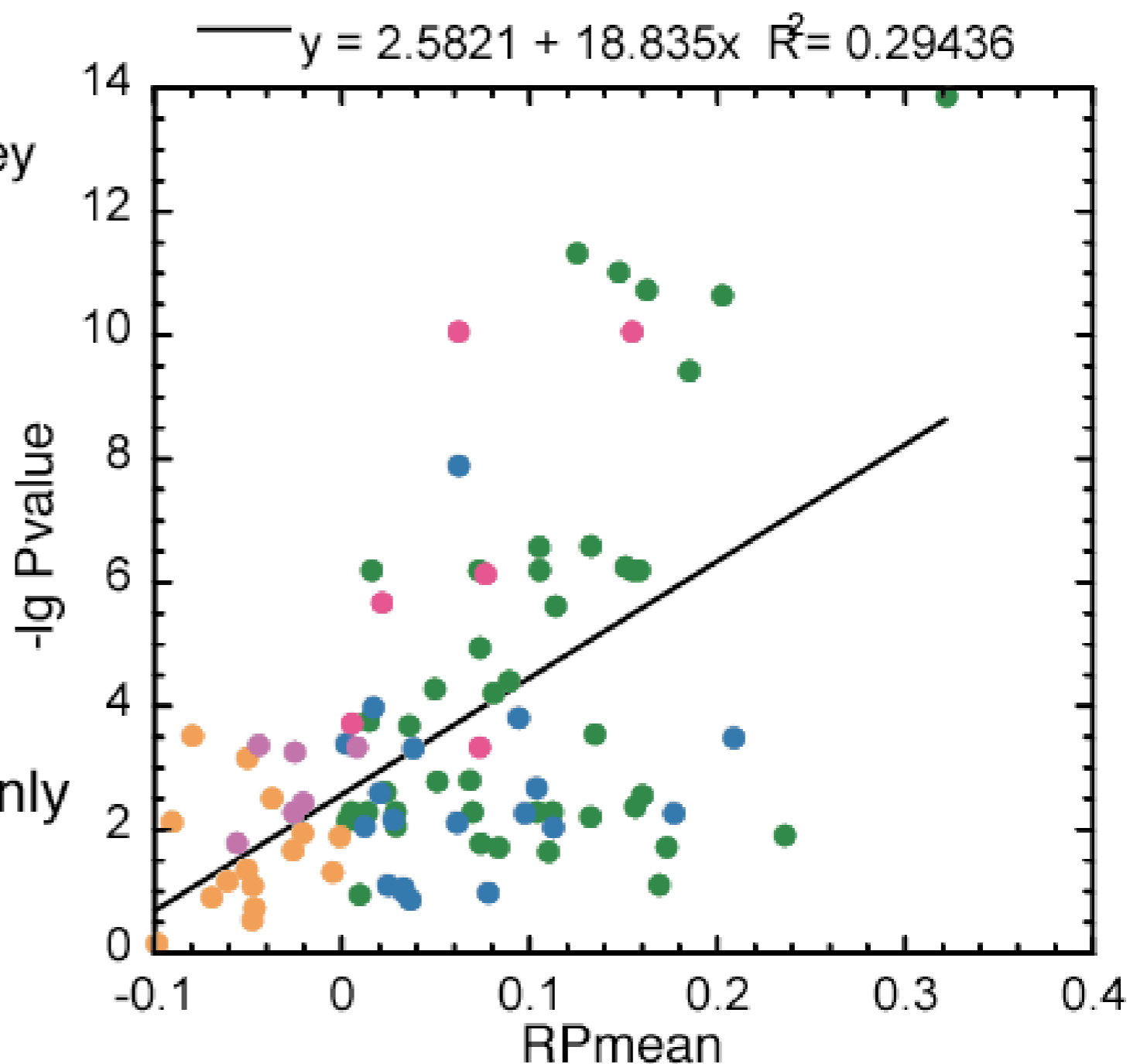




Enhancer activity correlates with RP score

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

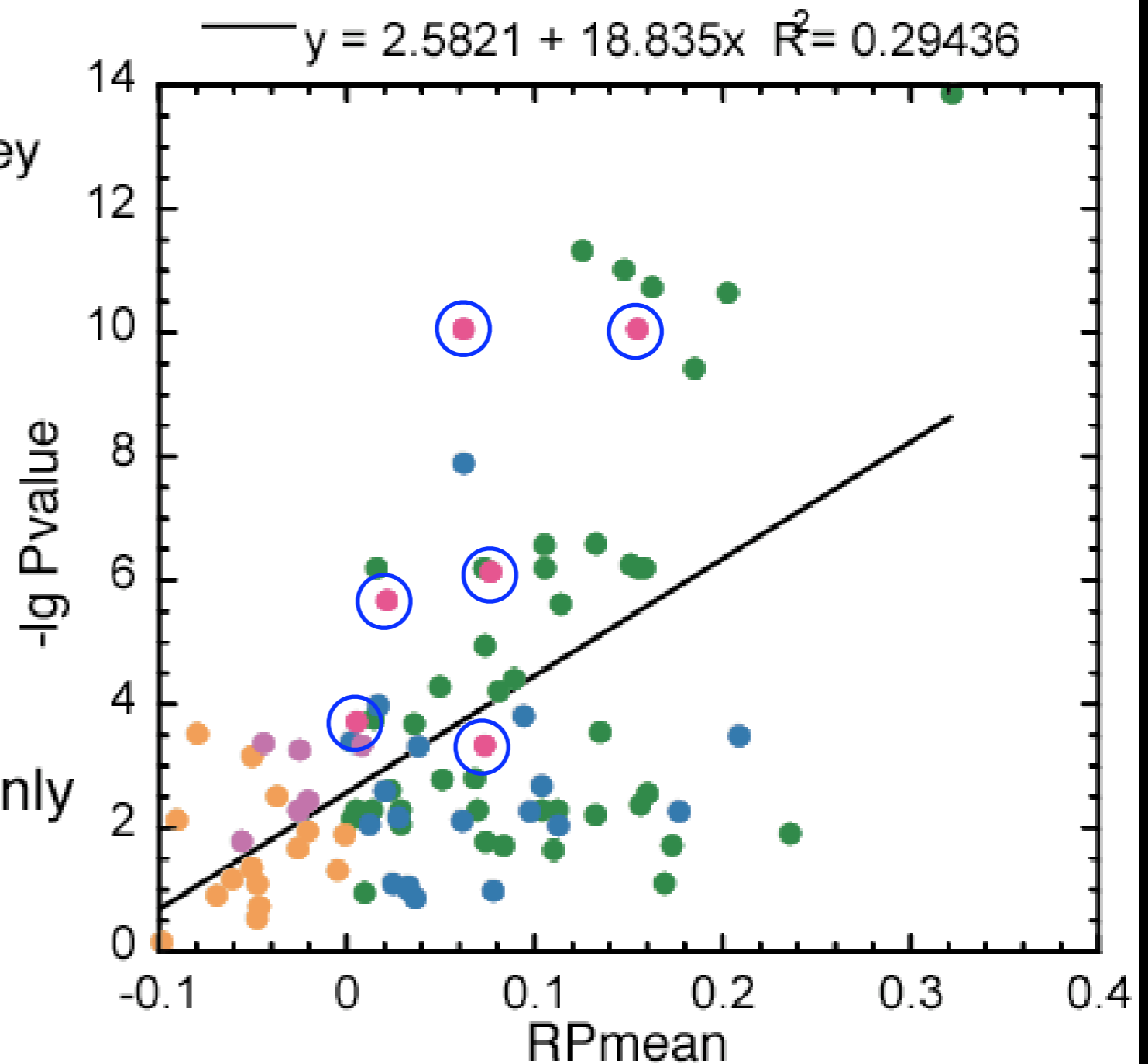
- preCRMcc
- preCRMcnc
- preNeutral
- NegRPw/ccGATA1
- PosiRPw/GATA1mouseonly



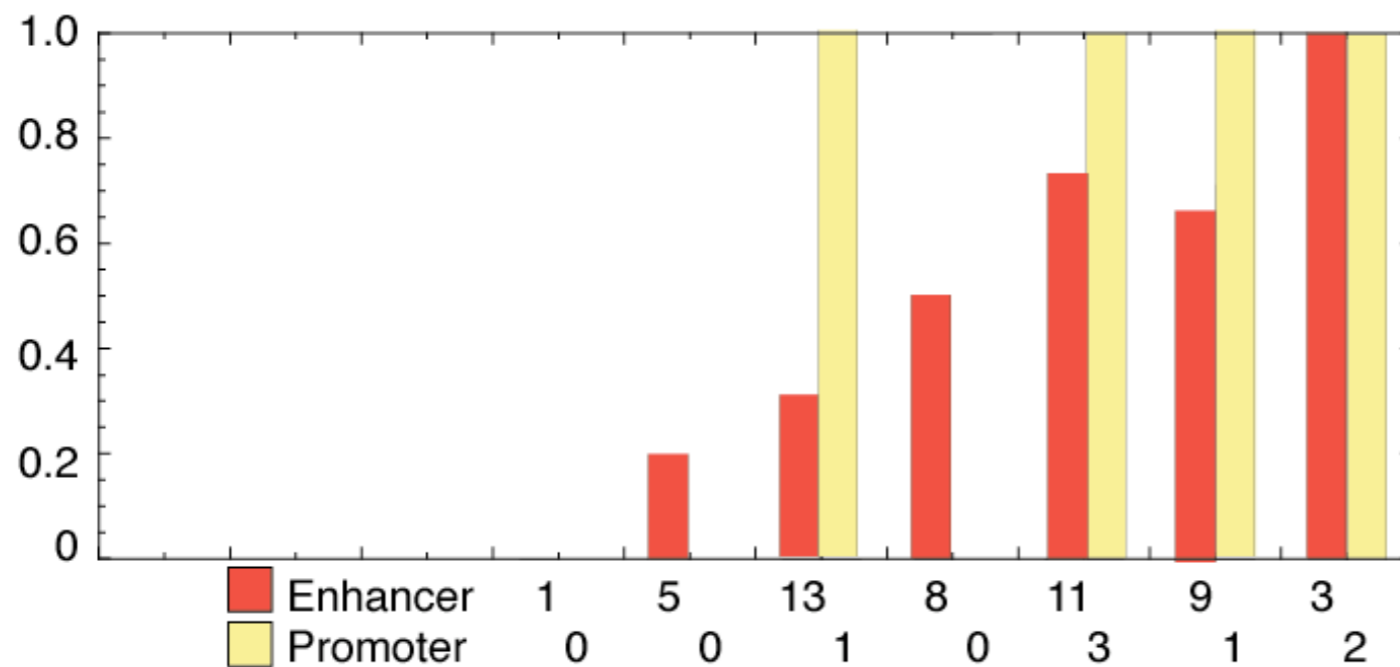
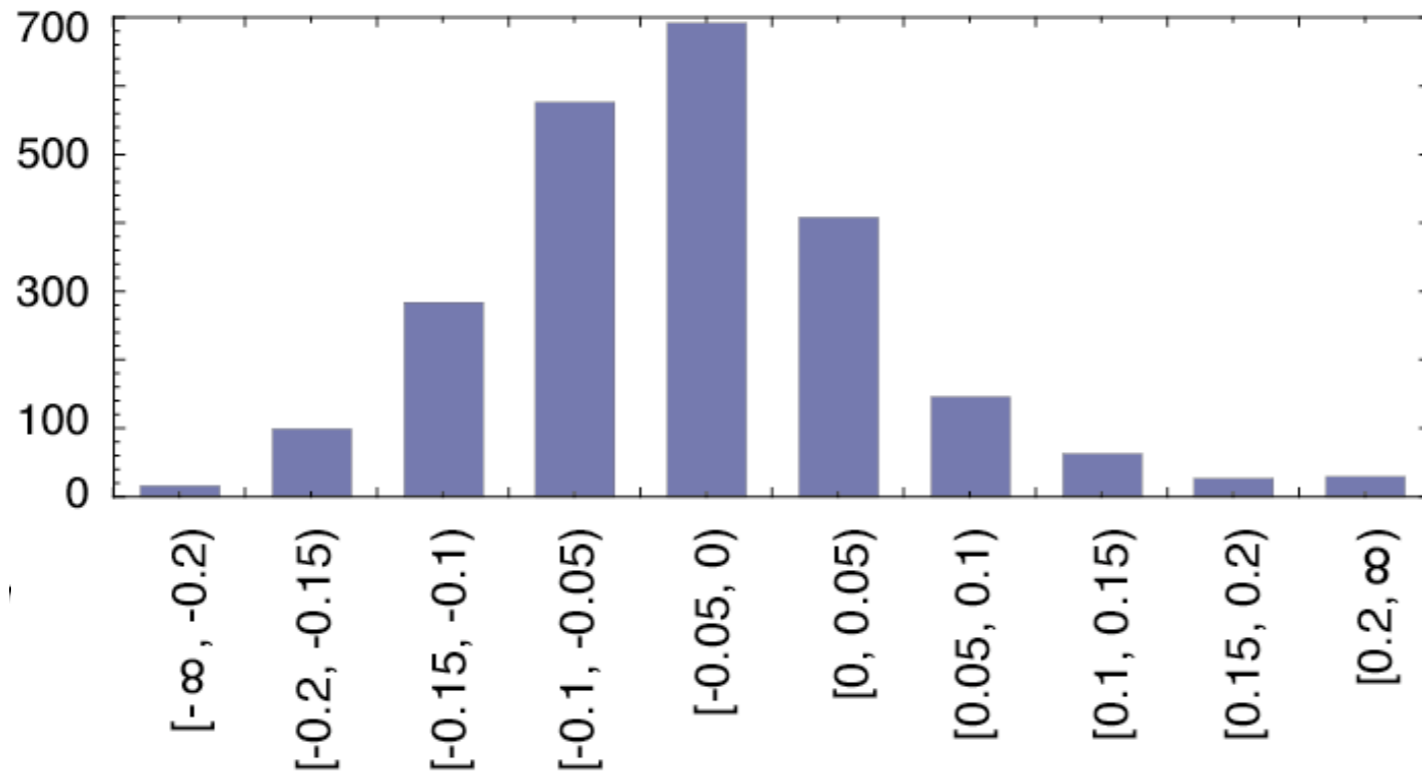
Enhancer activity correlates with RP score

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

- preCRMcc
- preCRMcnc
- preNeutral
- NegRPw/ccGATA1
- PosiRPw/GATA1 mouseonly



Higher RP scores yield better validation rates



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 2007Giraffe (*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 2007Giraffe (*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](#)

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](#)

 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](#)

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](#)

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:

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:
BED or Interval? See help below

Genome:


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](#)

 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:

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: Yes
Use this option if you are entering intervals by hand.

File Format:
BED or Interval? See help below

Genome:

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](#)

 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**

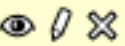
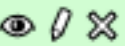
The following job has been successfully added to the queue:

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



4: UCSC Main on Human: knownGene (genome)



3: Pasted Entry



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="gettext">
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

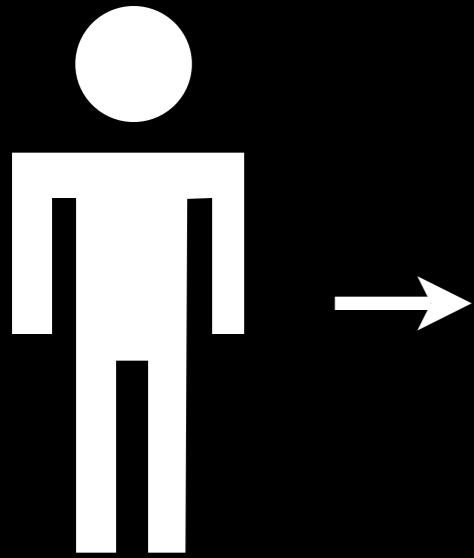
HYPHY

```
tool_conf.xml
1 <?xml version="1.0"?>
2 <toolbox>
3   <section name="Get Data" id="gettext">
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



Galaxy

Info: [report bugs](#) | [wiki](#) | [screencasts](#) Logged in as james@bx.psu.edu: [manage](#) | [logout](#)

Tools

- Get Data**
 - Upload File from your computer
 - UCSC Main table browser
 - UCSC Test table browser
 - UCSC Main table browser proxy
 - UCSC Test table browser proxy
 - UCSC Archaea table browser
 - Get Microbial Data
 - BioMart Central server
 - BioMart Test server
 - EncodeDB at NHGRI
 - HbVar: Human Hemoglobin Variants and Thalassemias
- 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**

UCSC Table Browser

clade: Vertebrate genome: Human assembly: May 2004

group: Genes and Gene Prediction Tracks track: Known Genes

table: knownGene describe table schema

region: genome ENCODE position
chr7:127471196-127495720 lookup define regions

identifiers (names/accessions): paste list upload list

filter: create

intersection: create

correlation: create

output format: BED - browser extensible data
Send output to Galaxy

get output summary/statistics

This is a proxy to the data services provided by the UCSC Genome Browser's Table Browser.

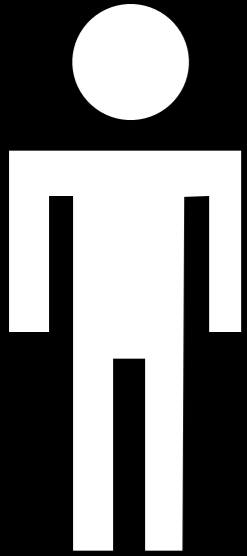
History (options)

refresh | collapse all

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

User makes request to Galaxy

Proxy based tools



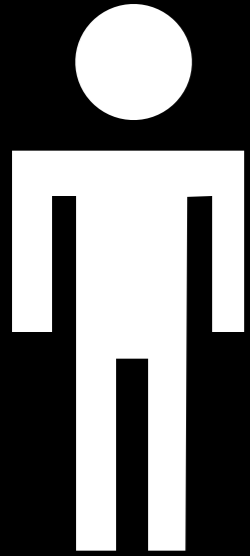
The screenshot shows the Galaxy web interface. The top navigation bar includes 'Galaxy', 'Info: report bugs | wiki | screencasts', and 'Logged in as james@bx.psu.edu: manage | logout'. The main content area is divided into three panels. The left panel, titled 'Tools', lists various data sources and analysis tools. The middle panel, titled 'UCSC Table Browser', contains the following configuration fields: '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'. The right panel, titled 'History (options)', shows a message: 'Your history is empty. Click "Get Data" on the left pane to start'.



The screenshot shows the external UCSC Table Browser web interface. The top navigation bar includes 'Home Genomes Genome Browser Blat Tables Gene Sorter PCR Session FAQ Help'. The main content area is titled 'Table Browser' and contains the following configuration fields: '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 configuration fields, there is a section titled 'Using the Table Browser' which provides a brief line-by-line description of the controls. The controls listed are: 'clade: Specifies which clade the organism is in.', 'genome: Specifies which organism data to use.', and 'assembly: Specifies which version of the organism's genome sequence to use.'

Galaxy delegates request to external site

Proxy based tools



Galaxy

Info: [report bugs](#) | [wiki](#) | [screencasts](#) Logged in as james@bx.psu.edu: [manage](#) | [logout](#)

Tools

- Get Data
 - Upload File from your computer
 - UCSC Main table browser
 - UCSC Test table browser
 - UCSC Main table browser proxy
 - UCSC Test table browser proxy
 - UCSC Archaea table browser
 - Get Microbial Data
 - BioMart Central server
 - BioMart Test server
 - EncodeDB at NHGRI
 - HbVar Human Hemoglobin Variants and Thalassemias
- 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

UCSC Table Browser

clade: Vertebrate genome: Human assembly: May 2004

group: Genes and Gene Prediction Tracks track: Known Genes

table: knownGene describe table schema

region: genome ENCODE position chr7:127471196-127495720 lookup define regions

identifiers (names/accessions): paste list upload list

filter: create

intersection: create

correlation: create

output format: BED - browser extensible data Send output to Galaxy

get output summary/statistics

This is a proxy to the data services provided by the UCSC Genome Browser's Table Browser.

History (options)

refresh collapse all

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

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: Vertebrate genome: Human assembly: May 2004

group: Genes and Gene Prediction Tracks track: Known Genes

table: knownGene describe table schema

region: genome ENCODE position chr7:127471196-127495720 lookup define regions

identifiers (names/accessions): paste list upload list

filter: create

intersection: create

correlation: create

output format: BED - browser extensible data Send output to Galaxy

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

file type returned: plain text gzip compressed

get output summary/statistics

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

Using the Table Browser

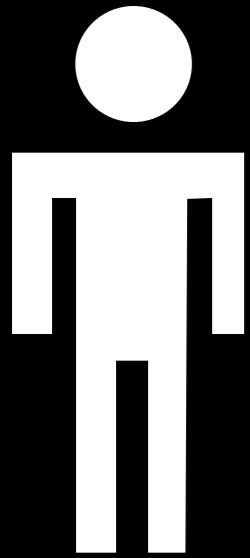
This section provides brief line-by-line descriptions of the Table Browser controls. For more information on using this program, see the [Table Browser User's Guide](#).

- clade:** Specifies which clade the organism is in.
- genome:** Specifies which organism data to use.
- assembly:** Specifies which version of the organism's genome sequence to use.

External site generates response

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

External tools



Galaxy Info: [report bugs](#) | [wiki](#) | [screencasts](#) Logged in as james@bx.psu.edu: [manage](#) | [logout](#)

Tools

- Get Data**
 - Upload File from your computer
 - UCSC Main table browser
 - UCSC Test table browser
 - UCSC Main table browser proxy
 - UCSC Test table browser proxy
 - UCSC Archaea table browser
 - Get Microbial Data
 - BioMart Central server
 - BioMart Test server
 - EncodeDB at NHGRI
 - HbVar: Human Hemoglobin Variants and Thalassemias
- 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**

UCSC Table Browser

clade: Vertebrate genome: Human assembly: May 2004

group: Genes and Gene Prediction Tracks track: Known Genes

table: knownGene describe table schema

region: genome ENCODE position
chr7:127471196-127495720 lookup define regions

identifiers (names/accessions): paste list upload list

filter: create

intersection: create

correlation: create

output format: BED - browser extensible data
Send output to Galaxy

get output summary/statistics

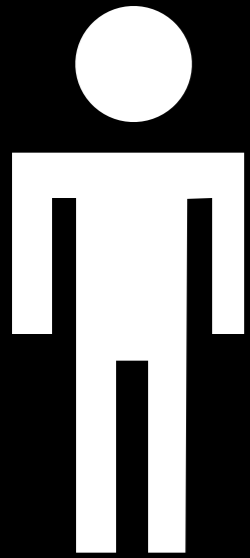
This is a proxy to the data services provided by the UCSC Genome Browser's Table Browser.

History (options)
refresh | collapse all

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

User makes request to Galaxy

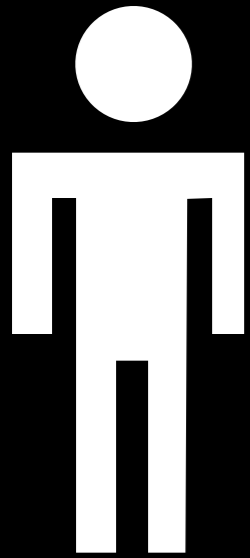
External tools



The screenshot displays the Galaxy web interface. At the top, the 'Galaxy' logo is visible along with navigation links for 'report bugs', 'wiki', and 'screencasts'. The user is logged in as 'james@bx.psu.edu'. The main content area shows the 'UCSC Table Browser' tool with the following configuration: clade: Vertebrate, genome: Human, assembly: May 2004, group: Genes and Gene Prediction Tracks, and track: Known Genes. A 'Tools' sidebar on the left lists options like 'Upload File from your computer' and 'UCSC Main table browser'. A 'History' panel on the right shows a message: 'Your history is empty. Click 'Get Data' on the left pane to start'. Overlaid on this is the 'bioMart' interface, which prompts the user to 'Please select columns to be included in the output and hit 'Results' when ready'. The bioMart interface includes a 'Dataset' section (Homo sapiens genes (NCBI36)), 'Filters' (None selected), and 'Attributes' (Ensembl Gene ID, Ensembl Transcript ID). The main selection area is divided into 'Features', 'Homologs', 'Structures', 'Sequences', and 'SNPs'. Under 'EXTERNAL:', there are sections for 'GO Attributes' (GO ID, GO description, GO evidence code), 'External References (max 3)' (CCDS ID, EMBL ID, EntrezGene ID, Havana ID, HGNC Symbol, IPI ID, limgt gene db, limgt ligm db, Mim Gene Accession, Mim Morbid accession, Mirbase, PDB ID, Protein ID, RefSeq DNA ID, RefSeq Predicted DNA ID, RefSeq Peptide ID, Rfam ID, Unigene ID, Shares cds with enst, Shares cds with ott, UniProt/SPTREMBL ID, UniProt/Swiss-Prot ID, UniProt/Swiss-Prot Accession, Unified UniProt ID, Unified UniProt Accession, Uniprot varsplicID), and 'Microarray Attributes (max 2)' (AFFY HCG110, AFFY HGU95E). The bioMart version is 0.6.

Galaxy sends user directly to external site with extra URL data

External tools



Galaxy Info: [report bugs](#) | [wiki](#) | [screencasts](#) Logged in as james@bx.psu.edu: [manage](#) | [logout](#)

Tools

- Get Data
 - Upload File from your computer
 - UCSC Main table browser
 - UCSC Test table browser
 - UCSC Main table browser proxy

UCSC Table Browser

clade: Vertebrate genome: Human assembly: May 2004

group: Genes and Gene Prediction Tracks track: Known Genes

History (options)
refresh | collapse all
Your history is empty. Click 'Get Data' on the left pane to start

bio::mart

HOME MARTVIEW MARTSERVICE DOCS CONTACT NEWS CREDITS

New Count Results XML Perl Help

Please select columns to be included in the output and hit 'Results' when ready

Features Homologs
 Structures Sequences
 SNPs

GENE:

EXTERNAL:

GO Attributes

GO ID GO evidence code
 GO description

External References (max 3)

CCDS ID RefSeq DNA ID
 EMBL ID RefSeq Predicted DNA ID
 EntrezGene ID RefSeq Peptide ID
 Havana ID Rfam ID
 HGNC Symbol Unigene ID
 IPI ID Shares cds with enst
 limgt gene db Shares cds with ott
 limgt ligm db UniProt/SPTREMBL ID
 Mim Gene Accession UniProt/Swiss-Prot ID
 Mim Morbid accession UniProt/Swiss-Prot Accession
 Mirbase Unified UniProt ID
 PDB ID Unified UniProt Accession
 Protein ID Uniprot varsplicID

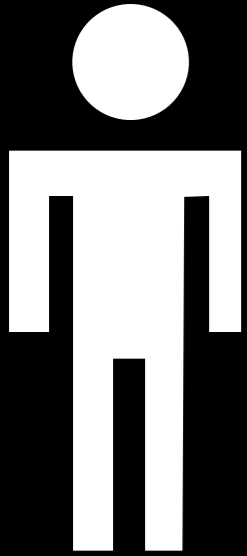
Microarray Attributes (max 2)

AFFY HCG110 AFFY HGU95E

biomart version 0.6

User interacts directly with external site

External tools



The screenshot shows the Galaxy web interface with the UCSC Table Browser tool. The tool is configured with the following settings:

- clade: Vertebrate
- genome: Human
- assembly: May 2004
- group: Genes and Gene Prediction Tracks
- track: Known Genes
- table: knownGene
- region: chr7:127471196-127495720
- output format: BED - browser extensible data

The interface also shows a list of tools on the left, including 'Get Data', 'UCSC Main table browser', 'UCSC Test table browser', 'UCSC Main table browser proxy', 'UCSC Test table browser proxy', 'UCSC Archaea table browser', 'Get Microbial Data', 'BioMart Central server', 'BioMart Test server', 'EncodeDB at NHGRI', 'HbVar Human Hemoglobin Variants and Thalassemias', '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', 'PHYLP', and 'HYPHY'. The bottom of the interface shows 'biomart version 0.6' and a list of attributes including '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', 'Microarray Attributes (max 2)', 'AFFY HCG110', and 'AFFY HGU95E'.

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

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/accessions):

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= description= visibility= ▾url= **Create one BED record per:**

- Whole Gene
- Upstream by bases
- Exons plus bases at each end
- Introns plus bases at each end
- 5' UTR Exons
- Coding Exons
- 3' UTR Exons
- Downstream by 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.

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: 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



HOME

MARTVIEW

MARTSERVICE

DOCS

CON

New

Count

Results

XML

Perl

Help

Dataset

[None selected]

- CHOOSE DATABASE -

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

MARTVIEW

MARTSERVICE

DOCS

CON

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

Dataset

[None Selected]

Export all results to

Galaxy

TSV

 Unique results only

Go

Email notification to

to

View

10

rows as TSV

 Unique results only

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

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**

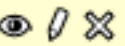
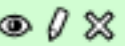
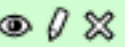
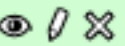
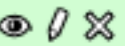
The following job has been successfully added to the queue:

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](#)

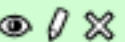
7: Homo sapiens genes (NCBI36)

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

Info: Homo sapiens genes (NCBI36)

[save](#)

1	2
Ensembl Gene ID	Ensembl Transcri
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:

max distance between intervals: (bp)

min number of intervals per cluster:

Return type:

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 \(NCBI36\)](#)   
- [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](#)   

Cluster

Cluster intervals of:

max distance between intervals: (bp)

min number of intervals per cluster:

Return type:

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



```

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</opt
22      <option value="5">Find the largest interval in each cluster</opt
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: 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
47  - Find cluster intervals; output grouped by clusters filters out n

```

Line: 87 Column: 8 XML Soft Tabs: 2

Cluster

Cluster intervals of:

max distance between intervals: (bp)

min number of intervals per cluster:

Return type:

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 -l $input1_chromCol,$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</opt
22      <option value="5">Find the largest interval in each cluster</opt
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: 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:

max distance between intervals: (bp)

min number of intervals per cluster:

Return type:

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 -l $input1_chromCol,$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</opt
22     <option value="5">Find the largest interval in each cluster</opt
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: 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:

max distance between intervals: (bp)

min number of intervals per cluster:

Return type:

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 -l $input1_chromCol,$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</opt
22      <option value="5">Find the largest interval in each cluster</opt
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  **Screencasts!**
33
34  See Galaxy Interval Operation Screencasts (right click to open this l
35  .. _Screencasts: http://www.bx.psu.edu/cgi-bin/trac.cgi/wiki/GopsDesc
36  -----
37
38  **Syntax**
39
40
41  .. class:: infomark
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:

max distance between intervals: (bp)

min number of intervals per cluster:

Return type:

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 -l $input1_chromCol,$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
27  .. class:: infomark
28
29  **TIP:** If your query does not appear in the pulldown menu -> it is n
30  -----
31
32  **Screencasts!**
33
34  See Galaxy Interval Operation Screencasts_ (right click to open this l
35  .. _Screencasts: http://www.bx.psu.edu/cgi-bin/trac.cgi/wiki/GopsDesc
36  -----
37
38  **Syntax**
39
40
41  .. **Maximum distance** is greatest distance in base pairs allowed betw
42  .. **Minimum intervals per cluster** allow a threshold to be set on the
43  .. **Merge clusters into single intervals** outputs intervals that span
44  .. **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 -l $input1_chromCol,$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
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: 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
47 - **Find cluster intervals; output grouped by clusters** filters out n
```

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

Cluster

Cluster intervals of:

max distance between intervals: *An integer is required*

min number of intervals per cluster:

Return type:

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</opt
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: 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
```

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

```
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 -l $input1_chromCol,$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  **Screenshots!**
34
35  See Galaxy Interval Operation Screenshots_ (right click to open this link in another wi
36
37  .. _Screenshots: 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 clust
46  - **Find cluster intervals; preserve comments and order** filters out non-cluster inter
47  - **Find cluster intervals; output grouped by clusters** filters out non-cluster interv
48
49
```

```
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 -l $input1_chromCol,$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 clust
46  - Find cluster intervals; preserve comments and order filters out non-cluster inter
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
```

} Template for generating
command line from
parameter values

```
cluster.xml
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 clust
46 - **Find cluster intervals; preserve comments and order** filters out non-cluster inter
47 - **Find cluster intervals; output grouped by clusters** filters out non-cluster interv
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>
```

} Output datasets
generated by the tool

```
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 clust
46 - **Find cluster intervals; preserve comments and order** filters out non-cluster inter
47 - **Find cluster intervals; output grouped by clusters** filters out non-cluster interv
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 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 clust
46 - **Find cluster intervals; preserve comments and order** filters out non-cluster inter
47 - **Find cluster intervals; output grouped by clusters** filters out non-cluster interv
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/modul
es:/Users/james/projects/galaxy/code/galaxy-trunk/eggs:/Users/james/projects/galaxy/code/galaxy-trunk/arch/darwin-i3
86/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: name: description: Color: Visibility:

Track 2

Dataset: name: description: Color: Visibility:



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" />
```

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

```
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" />
```

Repeating sets of parameters

```
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_file1"
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:

Series 2

Dataset:

Column for x axis:

Column for y axis:


Series Type:




Point Type:

Point Color:




Point Scale:

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   

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:

Series 2

Dataset:
Column for x axis:
Column for y axis:
Series Type:
Point Type:
Point Color:
Point Scale:

```
xy_plot.xml
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>
43           <option value="7">Yellow</option>
44           <option value="8">Gray</option>
45         </param>
46         <param name="lwd" type="float" label="Line Width" value="" size="30"/>
47       </when>

```

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:

Series 2

Dataset:

Column for x axis:

Column for y axis:

Series Type:

Point Type:

Point Color:

Point Scale:

```

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>
43         <option value="7">Yellow</option>
44         <option value="8">Gray</option>
45       </param>
46       <param name="lwd" type="float" label="Line Width" val
47     </when>
48     <when value="points">
49       <param name="pch" type="select" label="Point Type">
50         <option value="1">Circle (hollow)</option>
51         <option value="2">Triangle (hollow)</option>
52         <option value="3">Cross</option>
53         <option value="4">Diamond (hollow)</option>
54         <option value="15">Square (filled)</option>
55         <option value="16">Circle (filled)</option>
56         <option value="17">Triangle (filled)</option>
57     </param>

```

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:

Series 2

Dataset:

Column for x axis:

Column for y axis:

Series Type:

Point Type:

Point Color:

Point Scale:

```

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>
43         <option value="7">Yellow</option>
44         <option value="8">Gray</option>
45       </param>
46       <param name="lwd" type="float" label="Line Width" value="1.0"/>
47     </when>
48     <when value="points">
49       <param name="pch" type="select" label="Point Type">
50         <option value="1">Circle (hollow)</option>
51         <option value="2">Triangle (hollow)</option>
52         <option value="3">Cross</option>
53         <option value="4">Diamond (hollow)</option>
54         <option value="15">Square (filled)</option>
55         <option value="16">Circle (filled)</option>
56         <option value="17">Triangle (filled)</option>

```

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

```
xy_plot.xml
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 Info: report bugs | wiki | screencasts Account: create | login

Specialty tools

- aggregate_score Aggregate genomic scores
- alignability Alignability
- liftOver local liftover
- blat blat

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

Tools

aggregate_score

refresh | collapse all

Bed file: 2: alignibility on data 1

Data:

- esperr:hg18
- esperr:mm8
- esperr:hg17
- esperr:mm7
- esperr:red_mm8
- esperr:tmp
- phastcons:mm8_chr7

Execute

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	51570355	52070355	ENr112
chr2	118010803	118510803	ENr121
chr2	228102850	228602850	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

Tools

Gencode Partition

File to Partition: no data has the proper type

Execute

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:

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

refresh | collapse all

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

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_chr7(+):26907301-26907310|hg17_0
GTGGGAGGT
>panTro1_chr6(+):28037319-28037328|panTro1_0
GTGGGAGGT
>mm5_chr6(+):52104022-52104031|mm5_0
GTGGGAGGT
>rn3_chr4(+):80734395-80734404|rn3_0
GTGGGAGGT
>canFam1_chr14(+):42826409-42826418|canFam1_0
GTGGGAGGT

>hg17_chr7(+):26907310-26907326|hg17_1
AGTCAGAGTCTGAG
>panTro1_chr6(+):28037328-28037344|panTro1_1
AGTCAGAGTCTGAG
>mm5_chr6(+):52104031-52104047|mm5_1
AGTCAGAGTCTGAG
>rn3_chr4(+):80734404-80734420|rn3_1
AGTCAGAGTCTGAG
>canFam1_chr14(+):42826418-42826434|canFam1_1
AGTCAGAGTCTGAG

>hg17_chr7(+):26907326-26907338|hg17_2
GTAGAGACCCC
>panTro1_chr6(+):28037344-28037356|panTro1_2
GTAGAGACCCC
>mm5_chr6(+):52104047-52104059|mm5_2
GTAGAGATGCC
>rn3_chr4(+):80734420-80734432|rn3_2
GTAGAGATGCC
>canFam1_chr14(+):42826434-42826446|canFam1_2
GTAGAGACCCC

>hg17_chr7(+):26907338-26907654|hg17_3
GGGGAGGAAACGAGGGCGAGAGCTGGACTTCTGAGGAT---TCCTGGCCCTTCGCT---CGTTCCTGG---CGGGGTGGC
>panTro1_chr6(+):28037356-28037672|panTro1_3
GGGGAGGAAACGAGGGCGAGAGCTGGACTTCTGAGGAT---TCCTGGCCCTTCGCT---CGTTCCTGG---CGGGGTGGC
>mm5_chr6(+):52104059-52104375|mm5_3
GGAGAGGGGCACTGGCGAGGGGCTGATTCTCAGATGAT---TCTTCGGTTTCTCAT---CGCTGCCAGG---AGGAGTGGC
>rn3_chr4(+):80734432-80734748|rn3_3
GGAGAGGGGCACTGGCGAGGGGCTGATTCTCAGATGAT---TCTTCAGTTTCTCAT---CGCTGCCAGG---AGGAGTGGC
>canFam1_chr14(+):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

Get Data

Rgenetics

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

Tools

Clean:

Genotype file: /usr/local/galaxy/data/rg/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

Syntax

- Genotype data** is the input pedfile chosen from available library files
- New name** is the name to use for the filtered output file
- Missfrac threshold: subjects** is the threshold for missingness by subject. Subjects with more than this fraction missing will be excluded from the import
- Missfrac threshold: markers** is the threshold for missingness by marker. Markers with more than this fraction missing will be excluded from the import
- MaxMendel Individuals** Mendel error fraction above which to exclude subjects with more than the specified fraction of mendelian errors in transmission (for family data only)
- MaxMendel Families** Mendel error fraction above which to exclude families with more than the specified fraction of mendelian errors in transmission (for family data only)
- HWE** is the threshold for HWE test p values below which the marker will not be imported. Set this to -1 and all markers will be imported regardless of HWE p value
- MAF** is the threshold for minor allele frequency - SNPs with lower MAF will be excluded

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 chip

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: TDI_TDI.xls

2: camp2007RAWqc_QC_report.html

116.6 kb, format: html, database: 2
Info: ## Rgenetics v0.0.1, april 2007 /: http://rgenetics.org Galaxy Tools plink-CaCo.py started at 24/04/2007 14:14:34 spawning /usr/local/bin/R /usr/local/bin/R --vanilla --slave --args camp2007RAWqc /home/rerla/galaxy_dist/database/files /static/rg < /usr/loc

HTML #116 (116.6 kb)

There are 4 secondary datasets.

1: camp2007RAWqc_marker.xls

2: camp2007RAWqc_QC_report.log

3: camp2007RAWqc_nhet.xls

4: camp2007RAWqc_subject.xls

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 extract 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

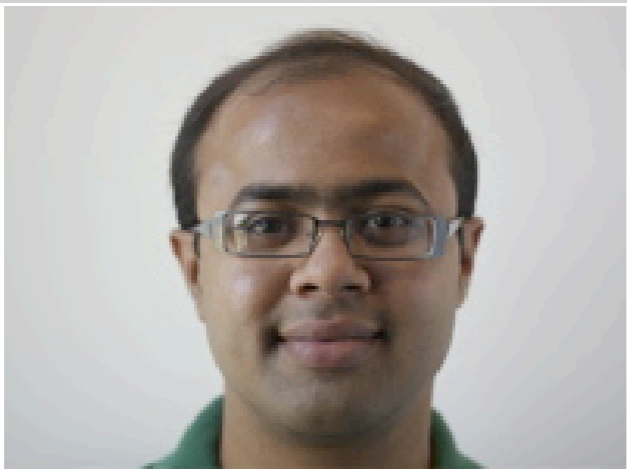
PSU production configuration

- 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"

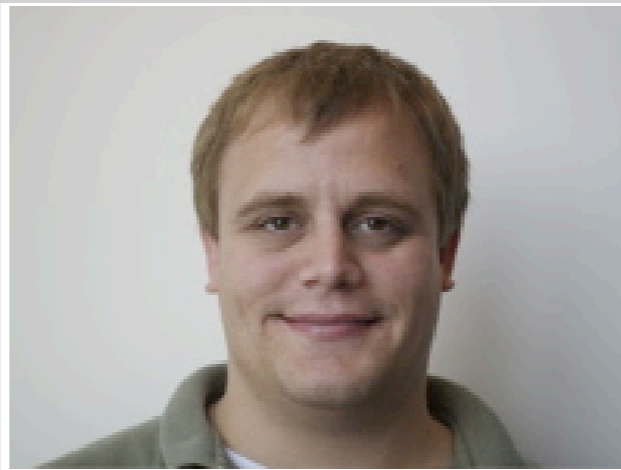
Acknowledgements

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

The core **Galaxy** development team



Guru Ananda



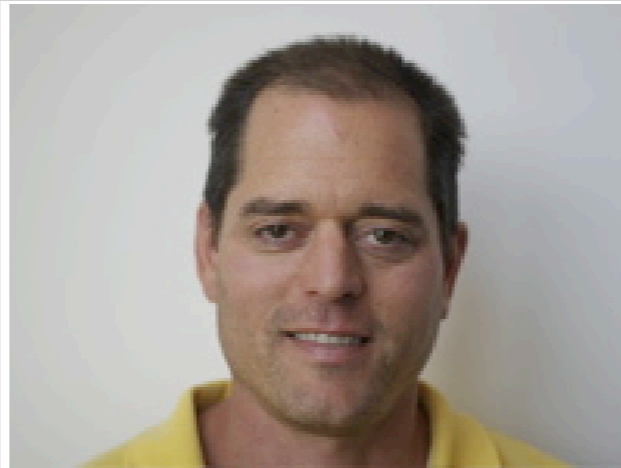
Dan Blankenberg



Nate Coraor



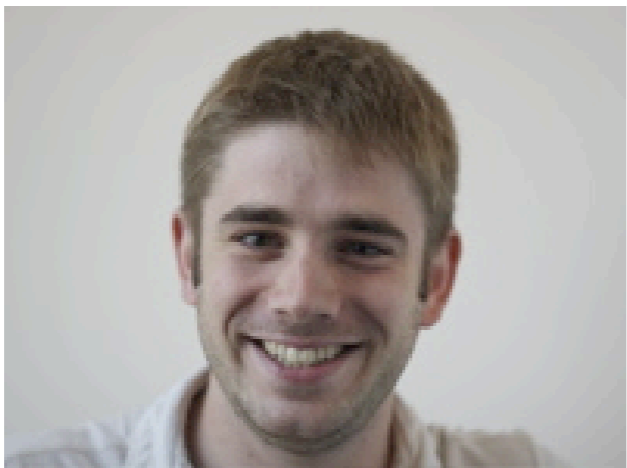
Jianbin He



Greg von Kuster



Anton Nekrutenko



Ian Schenck



James Taylor



Yi Zhang