The Generic Model Organism Database project

Scott Cain

scott@scottcain.net

Mastodon: scottcain@genomic.social

Research Professor (also, GMOD project coordinator emeritus)

Penn State

October 24, 2024

About me

- PhD in (Bio)Chemical Engineering in the late 90s
- Decided at the end to call myself a bioinformatician
- Taught myself Perl (and C, and Java, and JavaScript, and Python)
- Worked at a few bio-related startups
- For 22 years, worked for Lincoln Stein (at CSHL then OICR) on
  - GMOD (Coordinator)
  - WormBase (Developer)
  - Alliance of Genome Resources (NIH funded MODs) (Group leader)
  - Project manager for a few bio-portal projects (Cancer and Covid)
- “Moved” to Penn State last month, working on Galaxy

Intro to GMOD

- Started in 2001, funded by the NIH to reduce redundant software development efforts at NIH-funded MODs (Rat Genome Database, Mouse Genome Informatics, WormBase, FlyBase, Saccharomyces Genome Database + plus Gramene (rice) and TAIR (arabidopsis) later)
- Funded 2 developers at each MOD to work together as a group to identify projects that could be made generic.
- Also funded a few meetings a year which resulted in building a community that continues to this day.

GMOD projects that have stood the test of time

- GBrowse (widely used genome browser, end of lifed a few years ago)
- CMap (genomic comparative mapping tool–limped along for a long time after the developer left, and some ideas have come along to replace it)
- JBrowse (a JavaScript genome browser, even more widely used)
- Apollo (initially a Java application for graphical feature editing, then a web-based tool based on JBrowse)
- Intermine (a data querying tool)
- Chado (a database schema)
- Tripal (a MOD web service built on Drupal)
- MAKER (a gene prediction tool)
- Galaxy (a web based computation pipelining tool)

…and some that didn’t

- PubSearch (a reference tracking system)
- FlashGViewer (a flash-based genome browser)
- A “generic gene page” project (though Alliance of Genome Resources has kind of moved in this direction)

But the point is, we tried a lot of things, and a fair number of those were quite successful.

Types of projects we typically do

- Annotation creation/curation (both automated and manual)
- Data storage (databases and “queriers”)
- Visualization (browsers, whole websites)

What we don’t do:

- Algorithms! (ie, no gene predictors)

Genome Project Overview

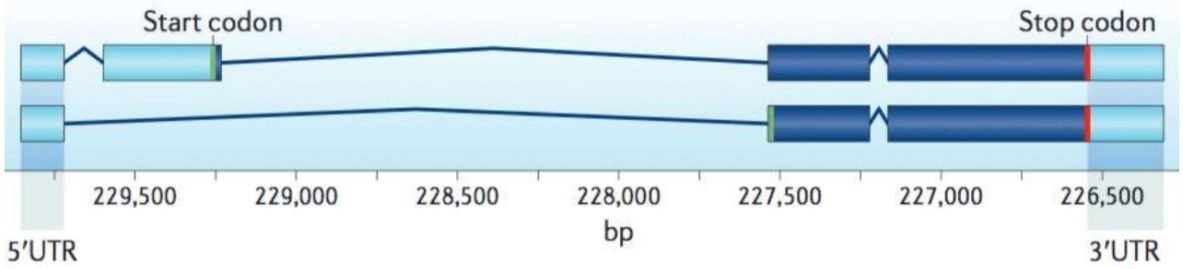
Genome Project Overview

Genome Project Overview

Genome Project Overview

What is an Annotation?

Structural

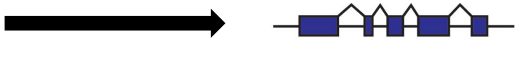


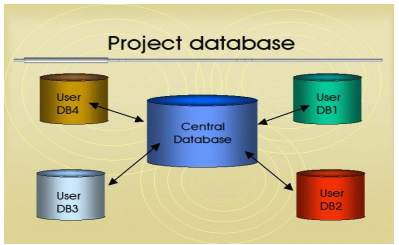
Functional

Function	cAMP-dependent and sulfonylurea-sensitive anion transporter. Key gatekeeper influencing intracellular cholesterol transport.
Subcellular location	Membrane; Multi-pass membrane protein Ref.13 Ref.14.
Domain	Multifunctional polypeptide with two homologous halves, each containing a hydrophobic membrane-anchoring domain and an ATP binding cassette (ABC) domain.

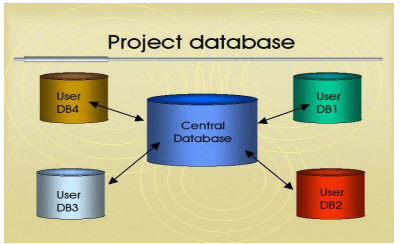
Genome Project Overview

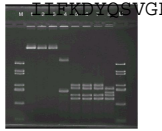
Genome Project Overview

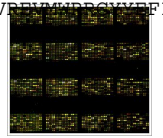


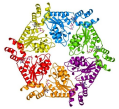


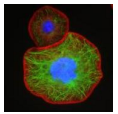
Genome Project Overview



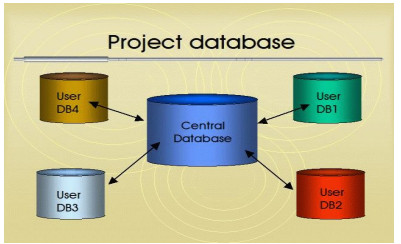


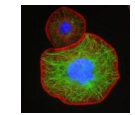


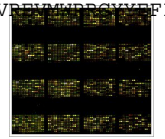


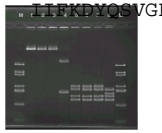


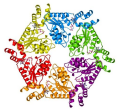
Genome Project Overview











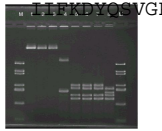
```
>Smg5
MEVTFSSGGSSNASSECAIDGGTNRCRGL
EPNNGTCILSQEVKDLYRSLYTASKQLDD
AKRNVQSVGQLFQHEIEEKRSLLVQLCKQ
IIFKDYQSVGKKVDFVMTPDGVYFIAFV
```

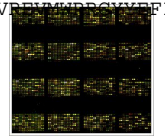
SUCCESS

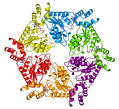
Genome Project Overview

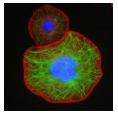
Incorrect annotations poison every experiment that uses them!!

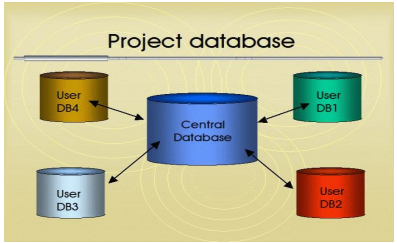
```
>Smg5
MEVTFSSGGSSNASSECAIDGGTNRCRGL
EPNNGTCILSQEVKDLYRSLYTASKQLDD
AKRNVQSVGQLFQHEIEEKRSLLVQLCKQ
IIFKDYQSVGKKVDFVMTPPGVYSFIAFV
```











MAKER

an annotation pipeline and genome-database management tool for second-generation genome projects

GMOD

Easy-to-use by design

Easy-to-use by design

User Requirements:	Can be run by a single individual with little bioinformatics experience
System	

Easy-to-use by design

User Requirements:	Can be run by a single individual with little bioinformatics experience
System Requirements:	Can run on laptop or desktop computers (running Linux or Mac OS X)

Easy-to-use by design

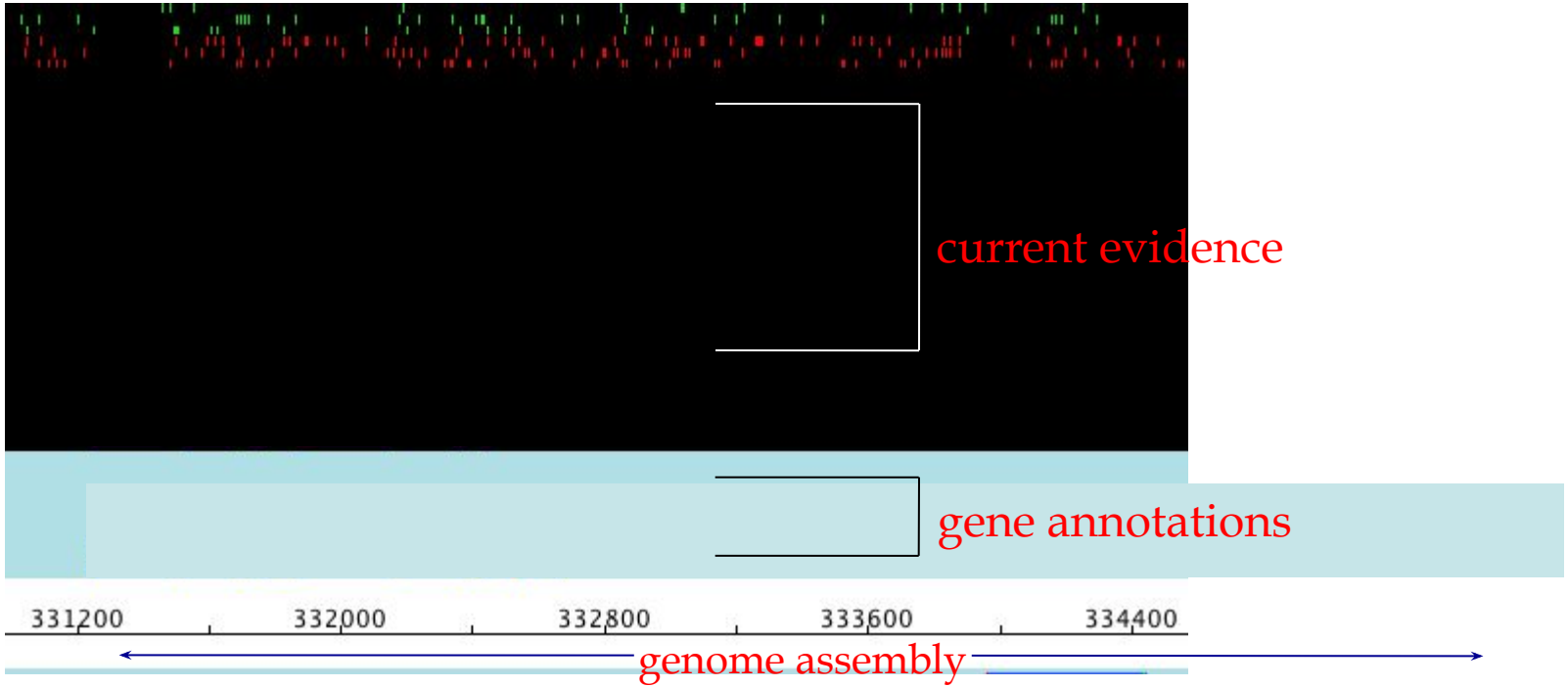
User Requirements:	Can be run by a single individual with little bioinformatics experience
System Requirements:	Can run on laptop or desktop computers (running Linux or Mac OS X)
Program Output:	Output is compatible with popular annotation tools like Apollo, GBrowse, and JBrowse

Easy-to-use by design

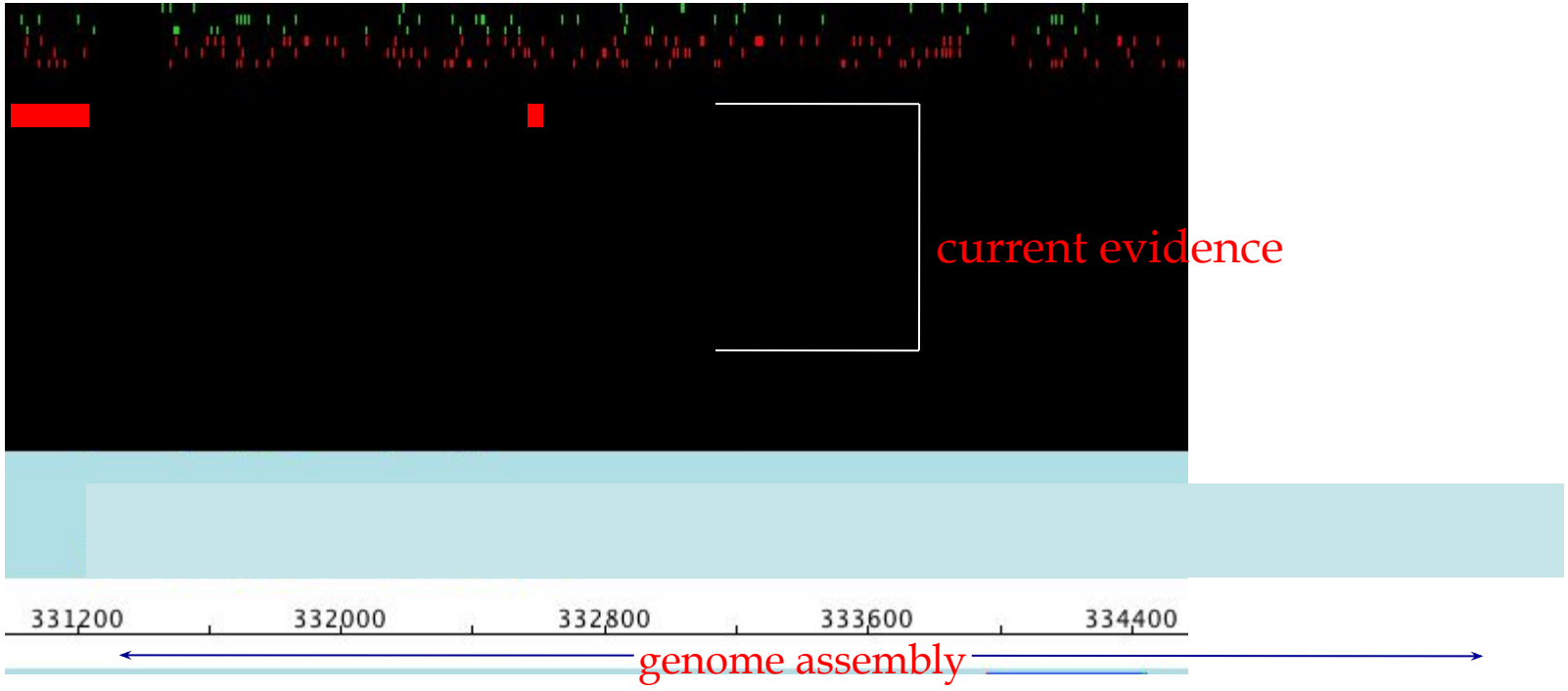
User Requirements:	Can be run by a single individual with little bioinformatics experience
System Requirements:	Can run on laptop or desktop computers (running Linux or Mac OS X)
Program Output:	Output is compatible with popular annotation tools like Apollo, GBrowse, and JBrowse
Availability:	Free open source application (for the academic community)

How does MAKER work?

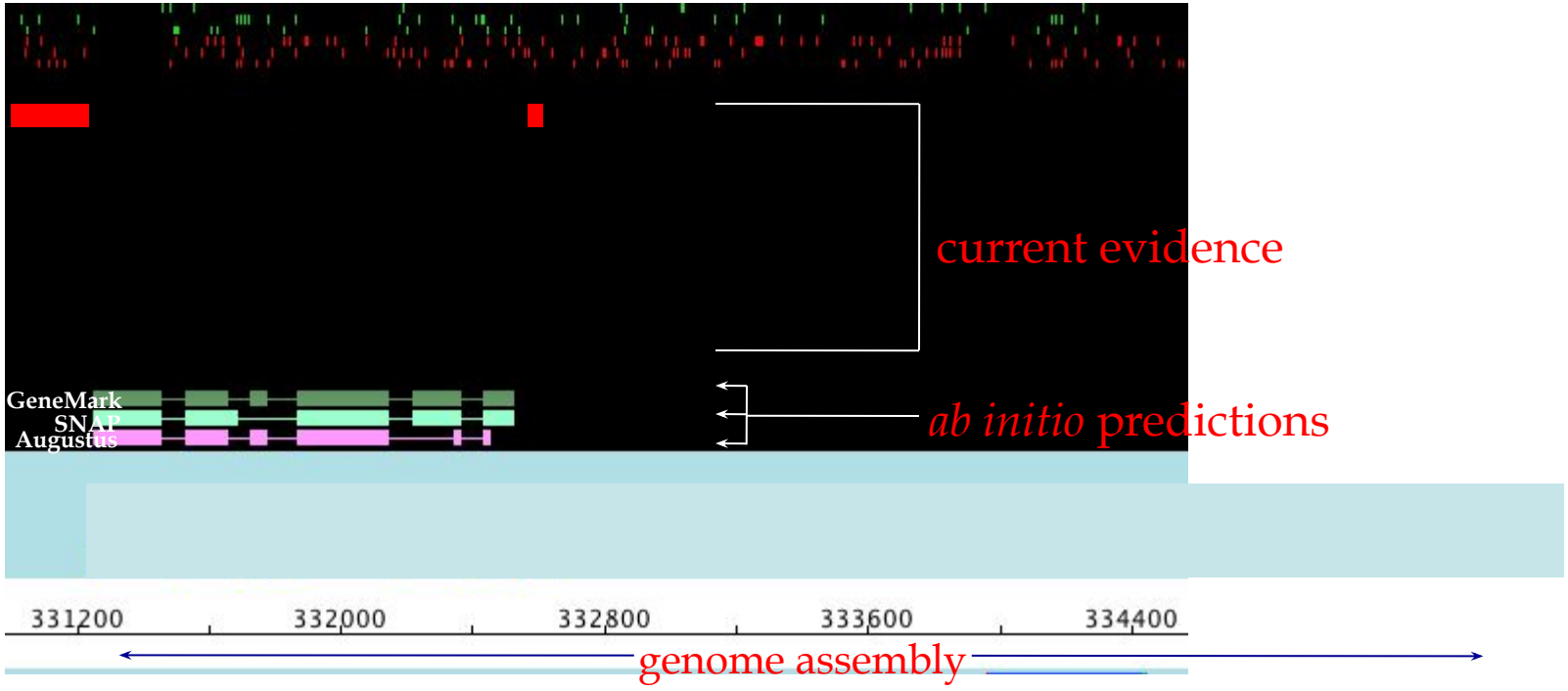
Annotating the Genome – Apollo View



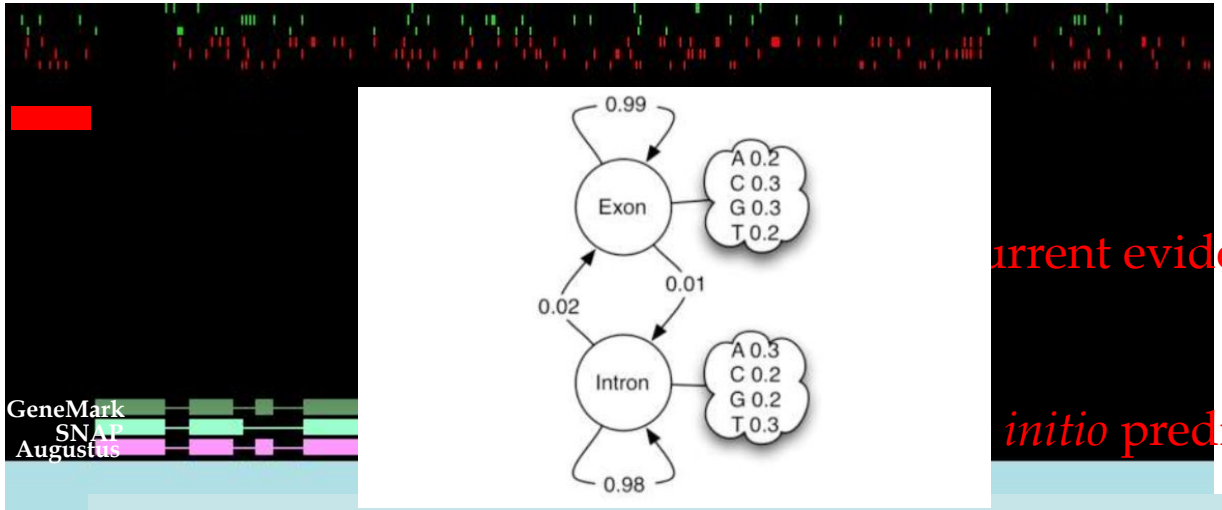
Identify and mask repetitive elements



Generate *ab initio* gene predictions



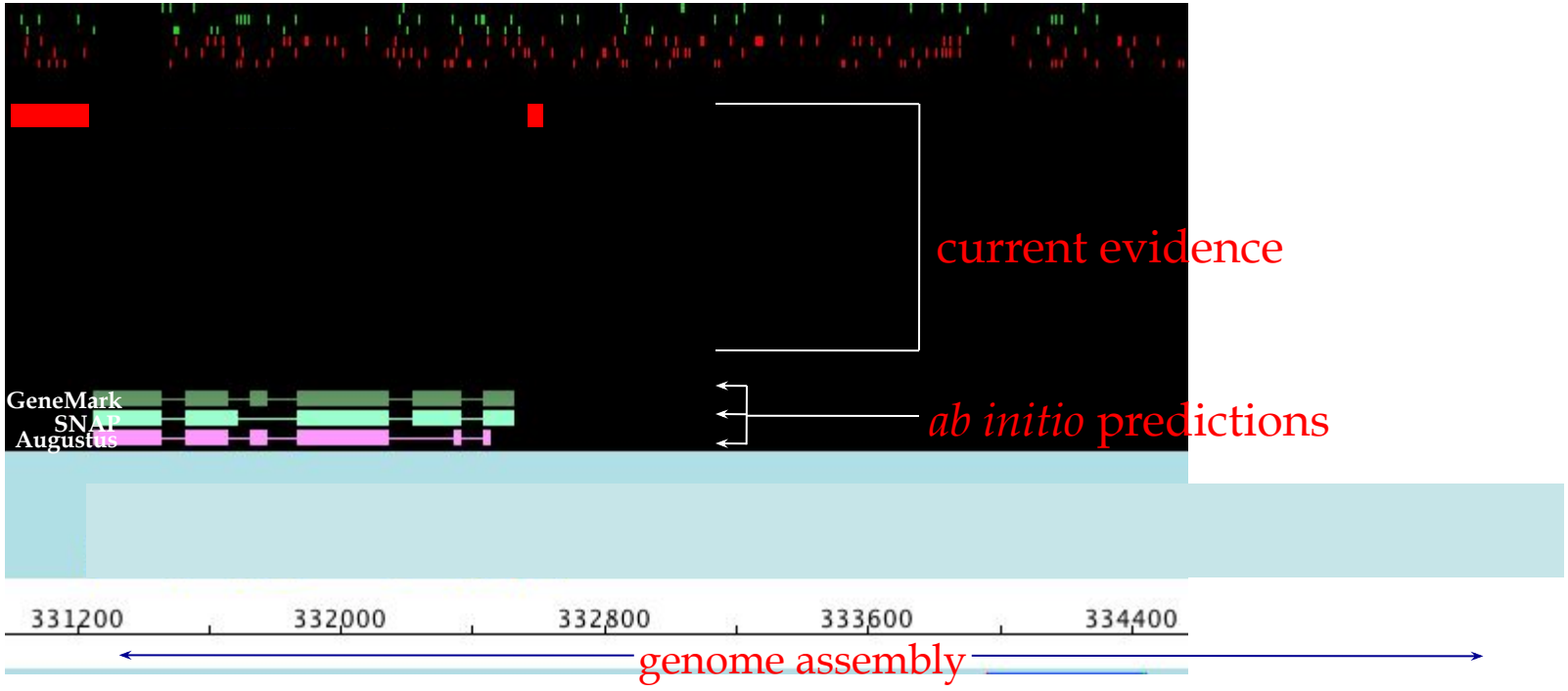
Generate *ab initio* gene predictions



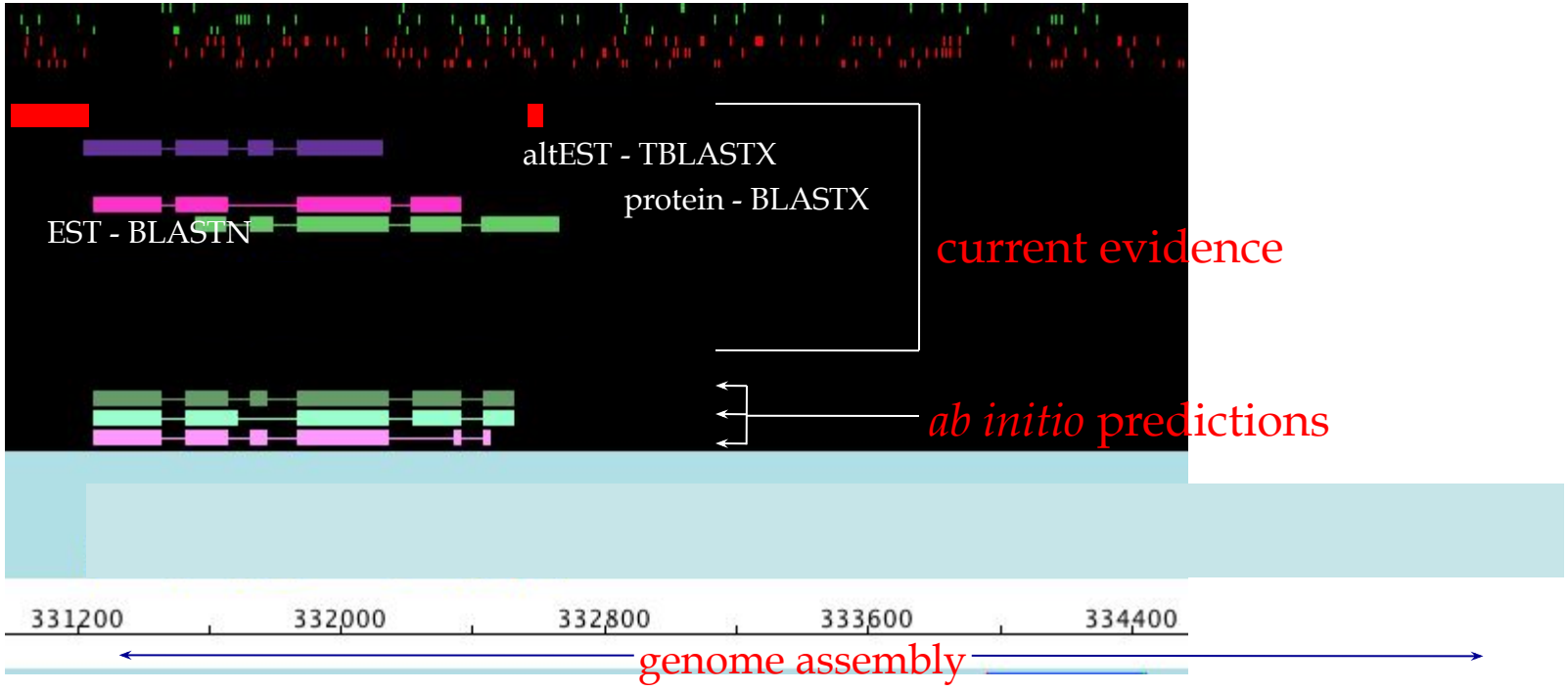
urrent evidence

initio predictions

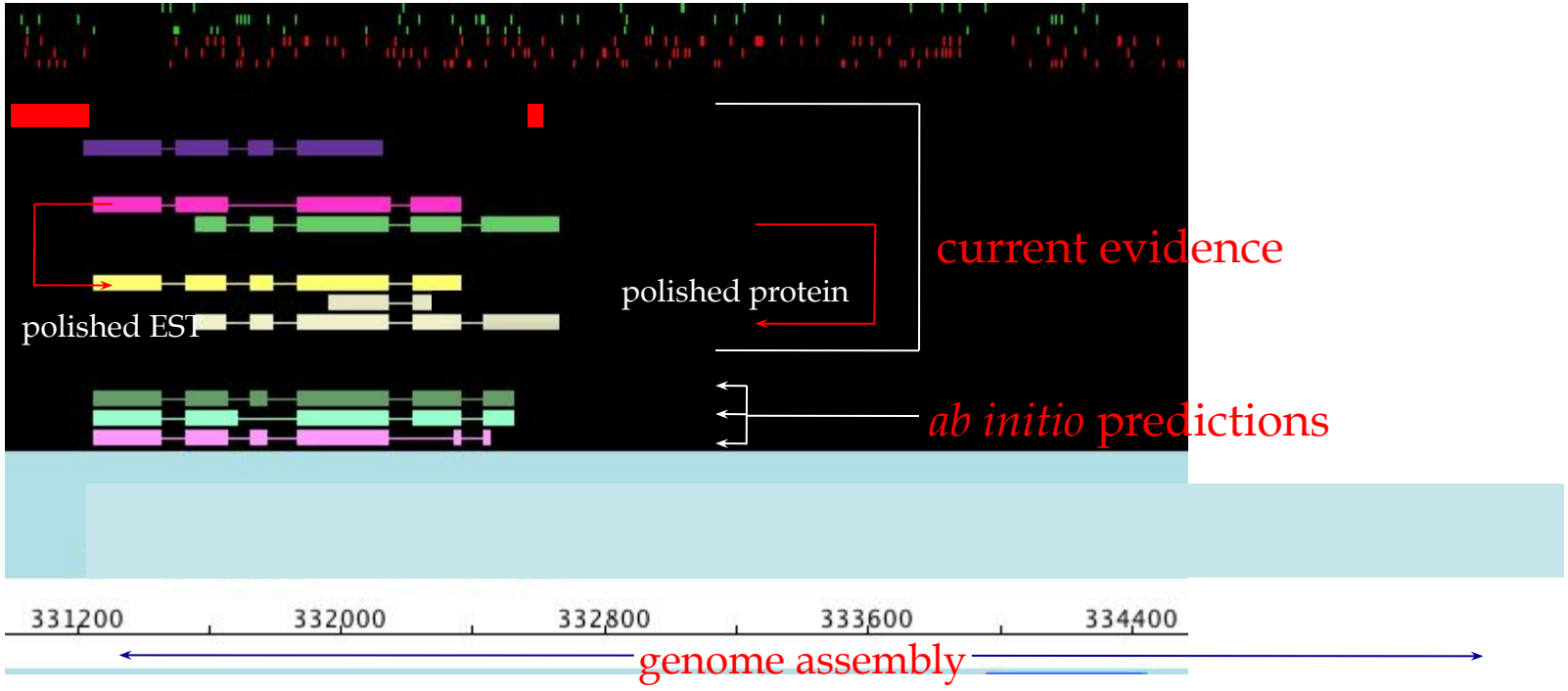
Generate *ab initio* gene predictions



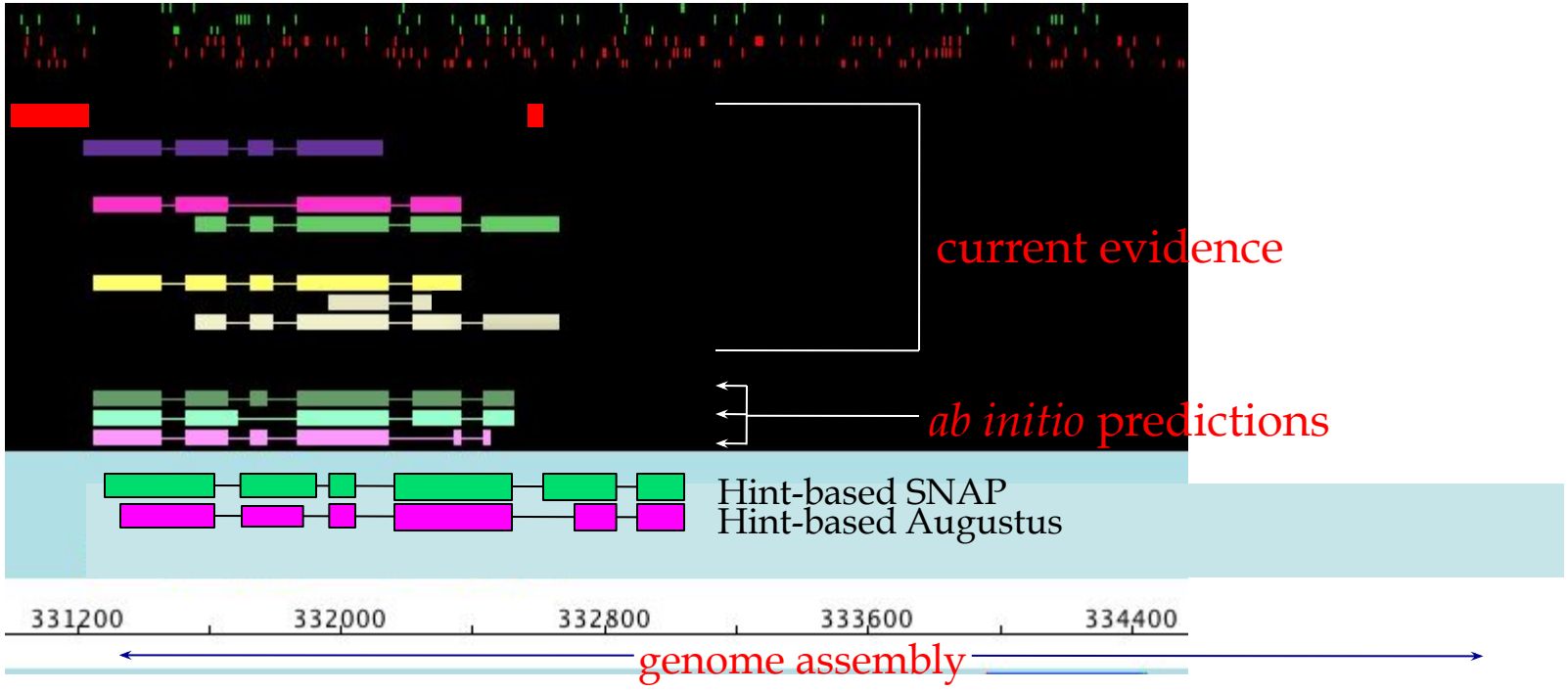
Align EST and protein evidence



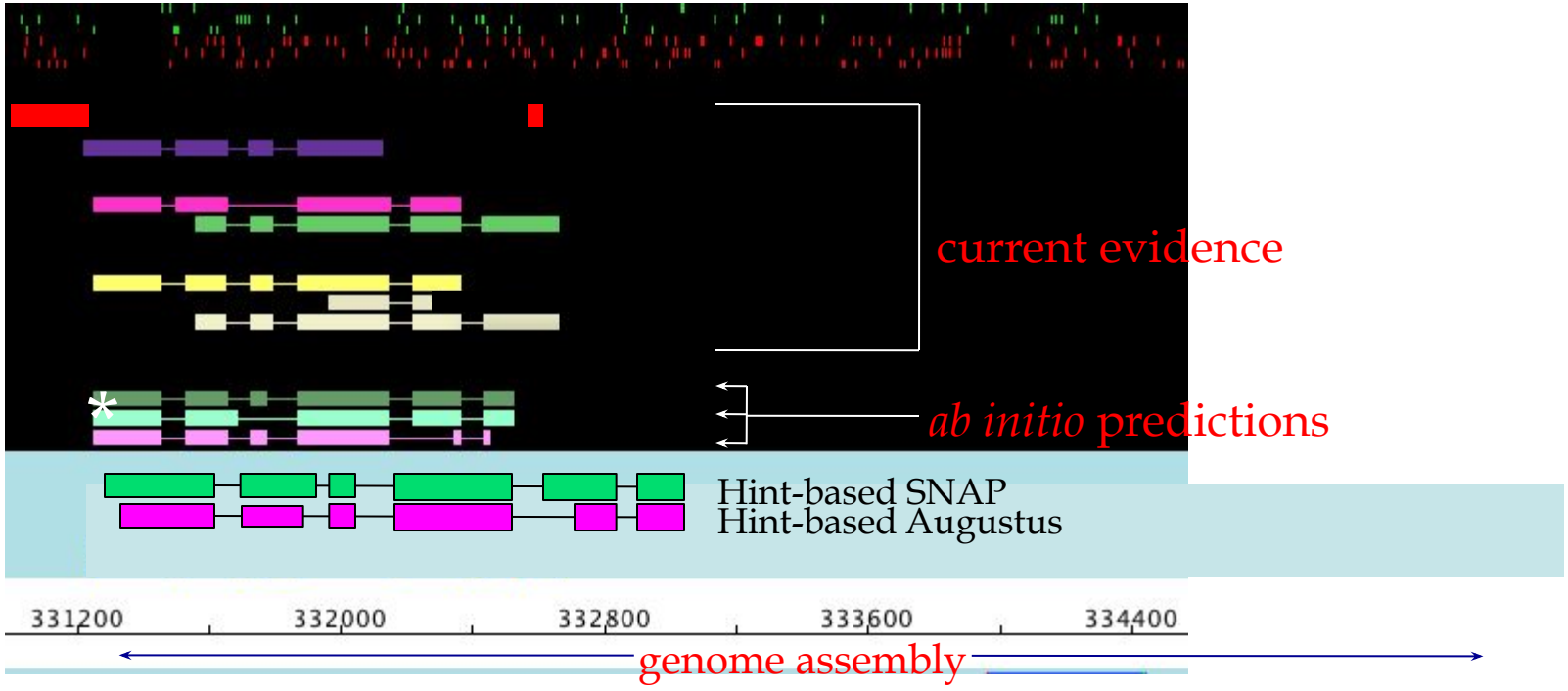
Polish BLAST alignments with Exonerate



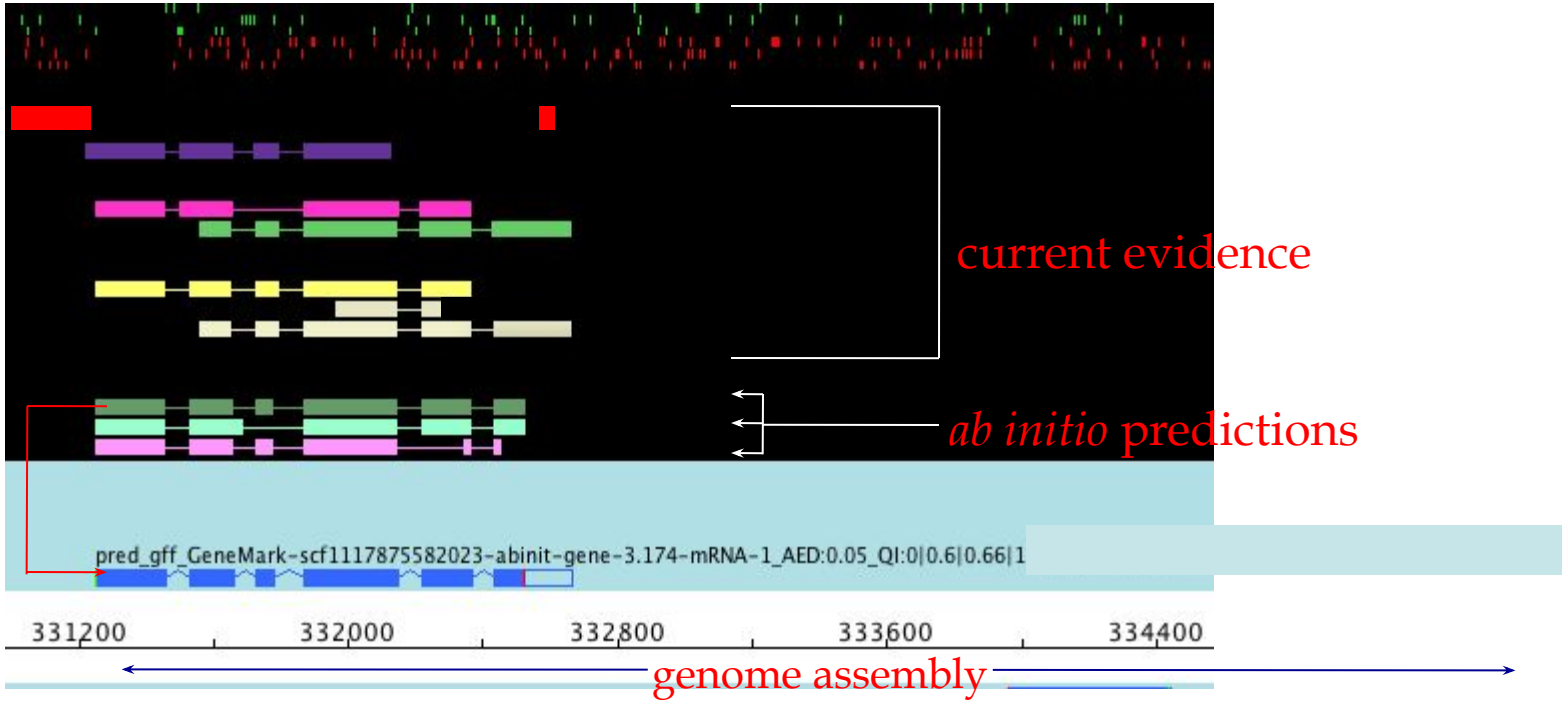
Pass gene-finders evidence-based ‘hints’



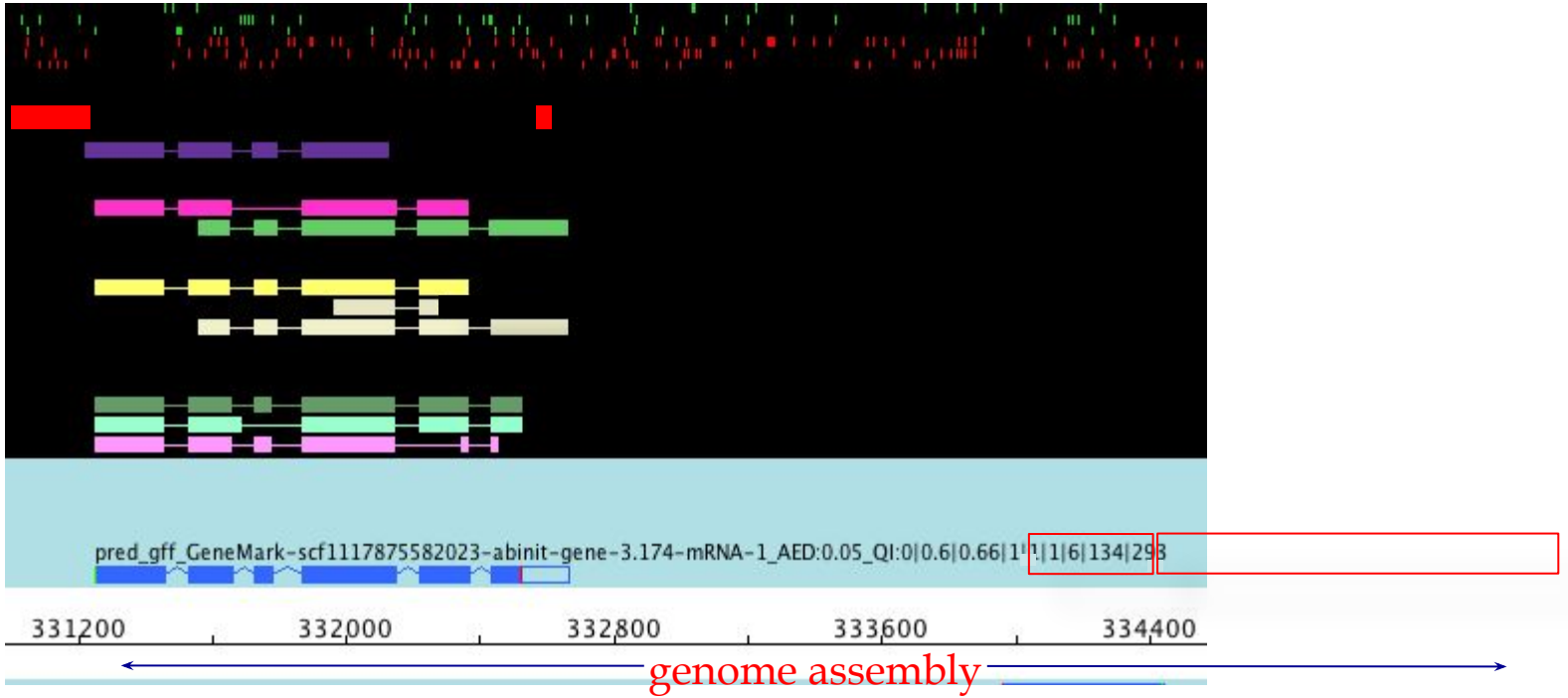
Identify gene model most consistent with evidence



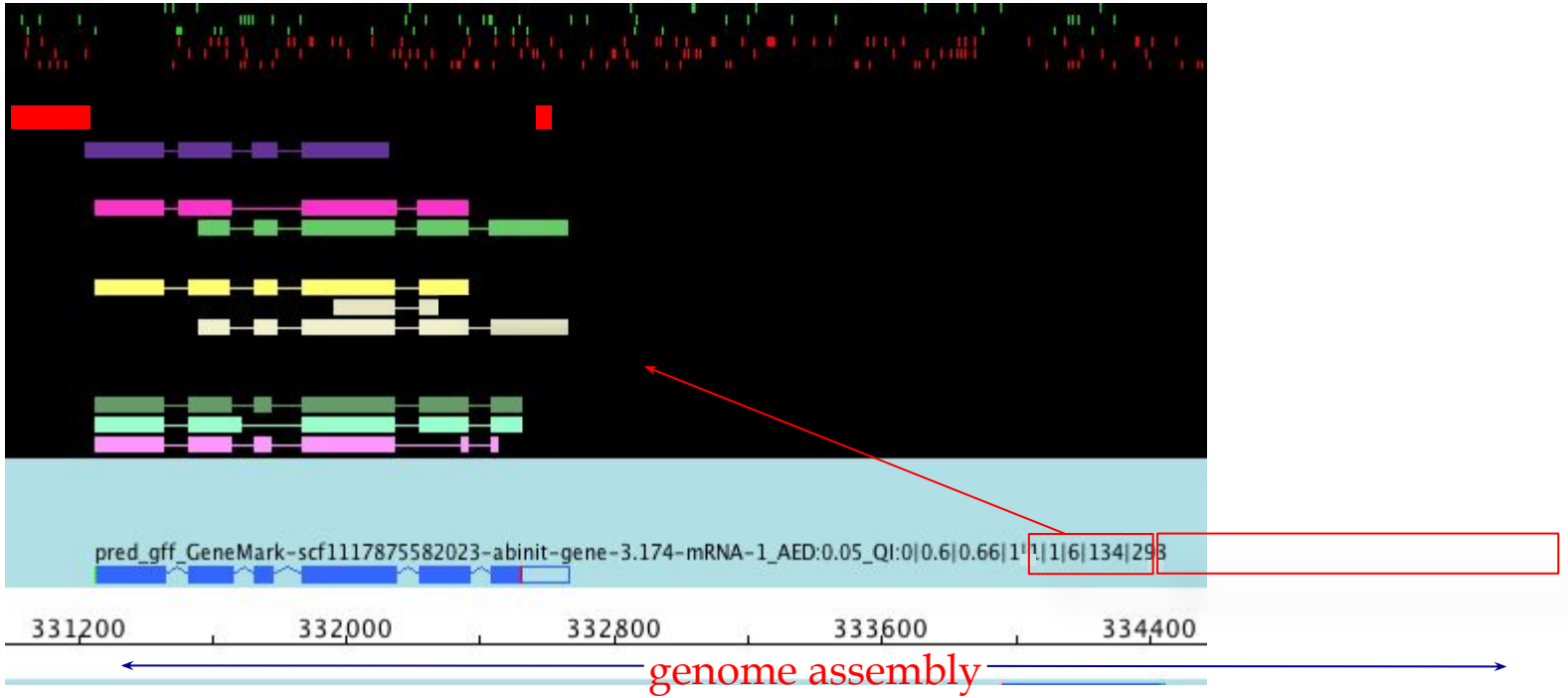
Revise it further if necessary; create new annotation



Compute support for each portion of gene model



Compute support for each portion of gene model



Compute support for each portion of gene model

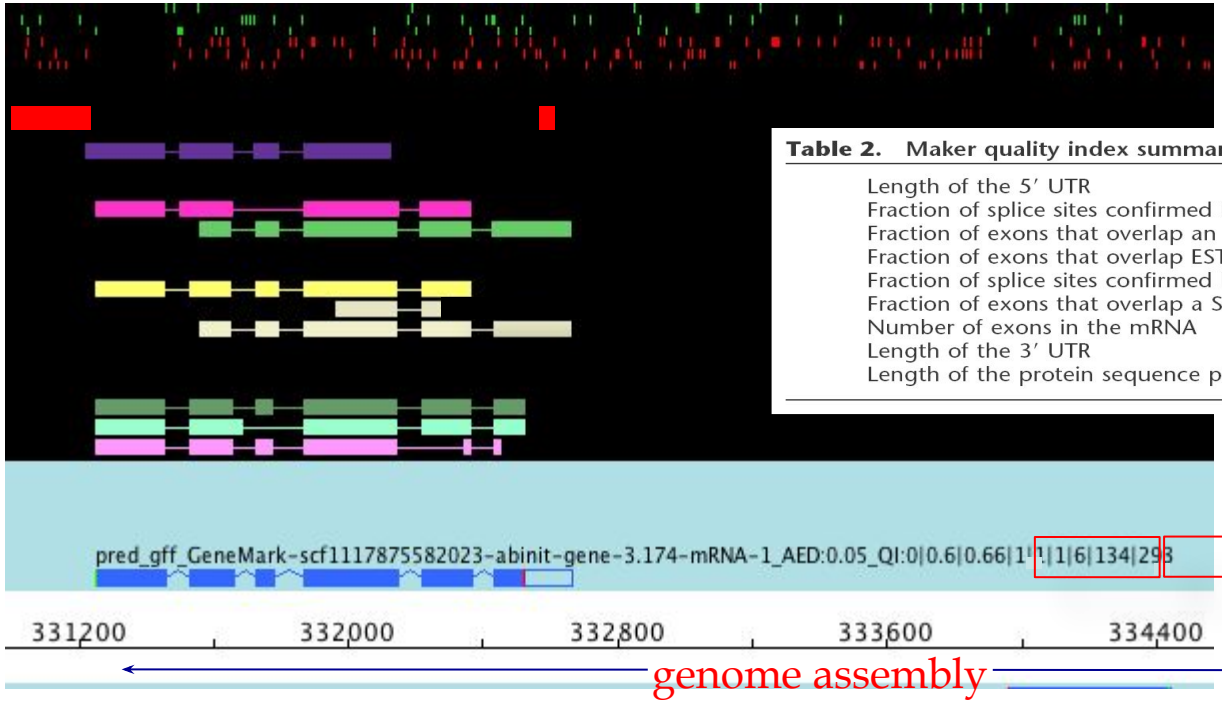


Table 2. Maker quality index summary

Length of the 5′ UTR
Fraction of splice sites confirmed by an EST alignment
Fraction of exons that overlap an EST alignment
Fraction of exons that overlap EST or Protein alignments
Fraction of splice sites confirmed by a SNAP prediction
Fraction of exons that overlap a SNAP prediction
Number of exons in the mRNA
Length of the 3′ UTR
Length of the protein sequence produced by the mRNA


bin — bmoore@derringer:/data1/genomes/Homo/sapiens/36.2 — ssh
scaffold00080 repeatmasker match_part 723561 723736 914 + . ID=scaffold00080:hsp:4987;Par
scaffold00080 repeatmasker match 724950 725171 495 + . ID=scaffold00080:hit:3405;Name=specie
scaffold00080 repeatmasker match_part 724950 725171 495 + . ID=scaffold00080:hsp:4988;Par
scaffold00080 repeatmasker match 726925 727293 724 + . ID=scaffold00080:hit:3406;Name=specie
scaffold00080 repeatmasker match_part 726925 727293 724 + . ID=scaffold00080:hsp:4989;Par
scaffold00080 maker gene 56197 58302 . + . ID=ACEP_00015614;Name=ACEP_00015614;Alias=mak
scaffold00080 maker mRNA 56197 58302 . + . ID=ACEP_00015614-RA;Parent=ACEP_00015614;Name
scaffold00080 maker exon 56197 56274 . + . ID=ACEP_00015614-RA:exon:126;Parent=ACEP_0001
scaffold00080 maker exon 56569 56584 . + . ID=ACEP_00015614-RA:exon:127;Parent=ACEP_0001
scaffold00080 maker exon 56797 56906 . + . ID=ACEP_00015614-RA:exon:128;Parent=ACEP_0001
scaffold00080 maker exon 57851 57941 . + . ID=ACEP_00015614-RA:exon:129;Parent=ACEP_0001
scaffold00080 maker exon 58067 58302 18.726 + . ID=ACEP_00015614-RA:exon:130;Parent=ACEP_0001
scaffold00080 maker CDS 56197 56274 . + 0 ID=ACEP_00015614-RA:cds:124;Parent=ACEP_00015
scaffold00080 maker CDS 56569 56584 . + 0 ID=ACEP_00015614-RA:cds:125;Parent=ACEP_00015
scaffold00080 maker CDS 56797 56906 . + 2 ID=ACEP_00015614-RA:cds:126;Parent=ACEP_00015
scaffold00080 maker CDS 57851 57941 . + 1 ID=ACEP_00015614-RA:cds:127;Parent=ACEP_00015
scaffold00080 maker CDS 58067 58302 . + 2 ID=ACEP_00015614-RA:cds:128;Parent=ACEP_00015
scaffold00080 maker gene 2797 3250 . - . ID=ACEP_00015615;Name=ACEP_00015615;Alias=mak
GFF3
bin — bmoore@derringer:/data1/genomes/Homo/sapiens/36.2 — ssh
>ACEP_00015614-RA protein AED:0.401129943502825 QI:0|0|0|0.2|1|1|5|0|177
MEKSDDYQDHYVIQFLEVYMYYFVHEKQYYLKVLWKRIITLSNKYYPIGVIWAFFVPIVF
IRHPDDLKTILSNPKHIKKSFFYDNIKPWLGTSILTSEDCTLSHFIPMLSNRHFTIVHCI
VHVLGTKWQLQRKILISTFHFDILNQFVEIFEKESKNMIKSLKNAEGTVVKDLSSFI
>ACEP_00015615-RA protein AED:0.223684210526316 QI:60|0|0|0.5|0|0|2|0|93
MDTQQKHKREIPTEPGEINSVISFQICISCCDLSYCNIESPTNATNAIYSRQRRAKSKSK
RKRPGRNGVDAATRLYGSSLLWLPASLFYAYHC
>ACEP_00015613-RA protein AED:0.191135958515638 QI:9|0.4|0.33|0.83|0.2|0.5|6|0|238
MEKSDNSLTYQDHYSIFGSVYVLFGSREILWKIITLSNKFLRTFYPIGIIRAFFVPIVSI
RHPDDLKSFFYDNIKPWLGTSILINEGTKWQLQRKILIPTFHFDILNQFVEIFEKESKNM
IKSLKNAEGTVVKDLSSFISEYTLNAICESYLIPAGTVLHININGVHTDPNFWRYFQIQR
YLILIDFCPRGSEIVTLTHIYHSVLDHVIVSNLRFSKFDHERSGIVRE
>ACEP_00015612-RA protein AED:0.469714351691491 (
MYFVNFNYFYANASVTNHRNGRTSLHTELLSSKVKREQTLRNILTARR
RILSSRRPRPGDIISKDPQDVLETQETRNGIYIANARTEISIKEEIPQ
INLNATGEELESRTTKTSKNESLSPTTEKIENISSIVSTPATNQKLGDAEMRKKNTVSMP
ESLYNHFRPVESNIPVEDMSQFLYFGQKLQPDALNVTSSSNNSVETTSTNPTSSRRRYST
KRFTATIATPMEEIMNEEMNIALETLERRTVEKNQVSRIKNTFRSSGNGLYYRKPRPTAD
VVSNIIPGNESSPGIIGGSETKSEMNTVDDKFHRSDSSAHADGSRDRVPLERKNAAHVSE
VSMTTQIPRQSDVEARVIVESEIGAKLSNKTITASNGIVESLQRINDFGKDNEKKKITEA
EAGNVEKSEHKLMSVERTISTSQRESERFRESDSTETMSLVPSRAESSTLRIVVPDNSFG
FASTA

MAKER Resources

http://yandell-lab.org/software/maker.html Docs and demo

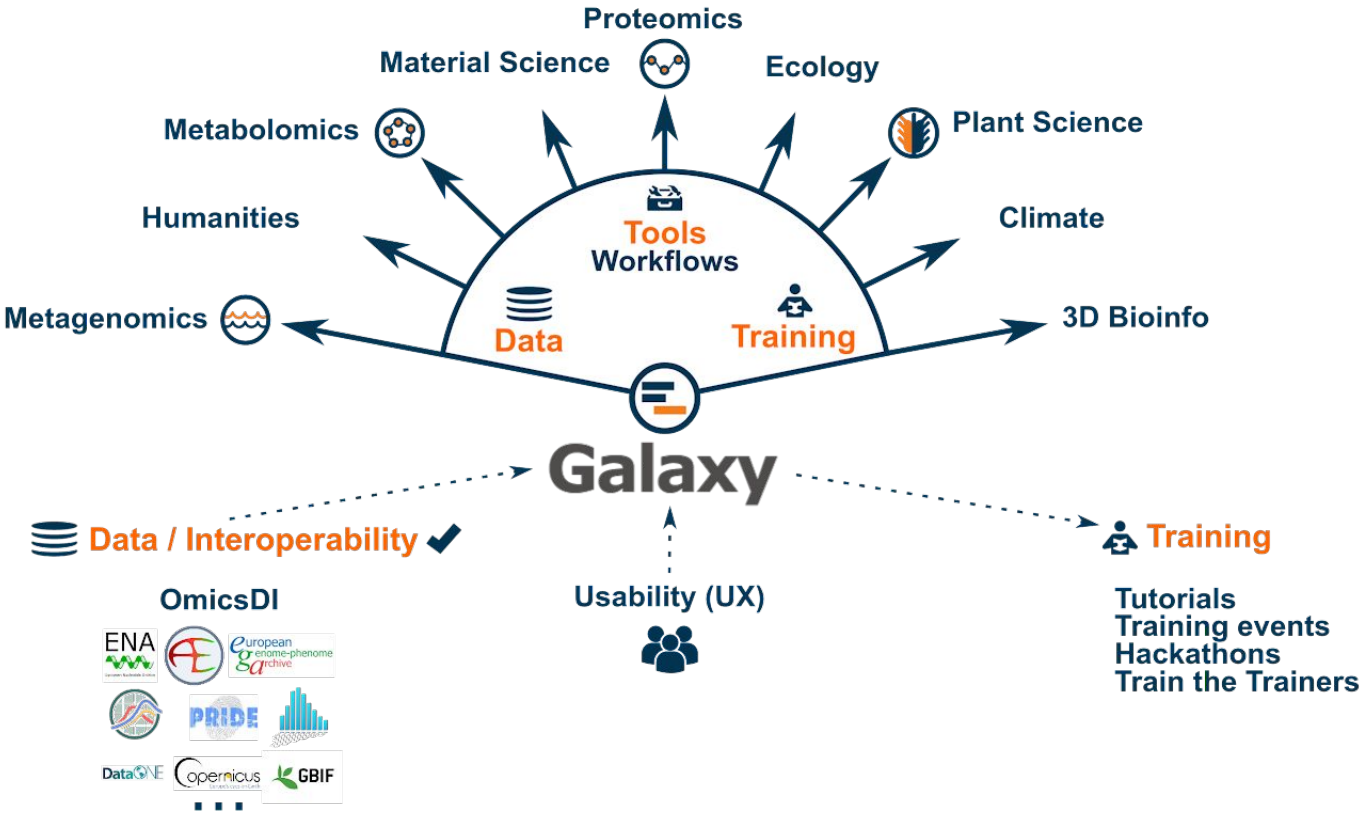
Galaxy PROJECT

Galaxy is an **open**, web-based platform for **accessible**, **reproducible**, and **transparent** computational research

https://galaxyproject.org

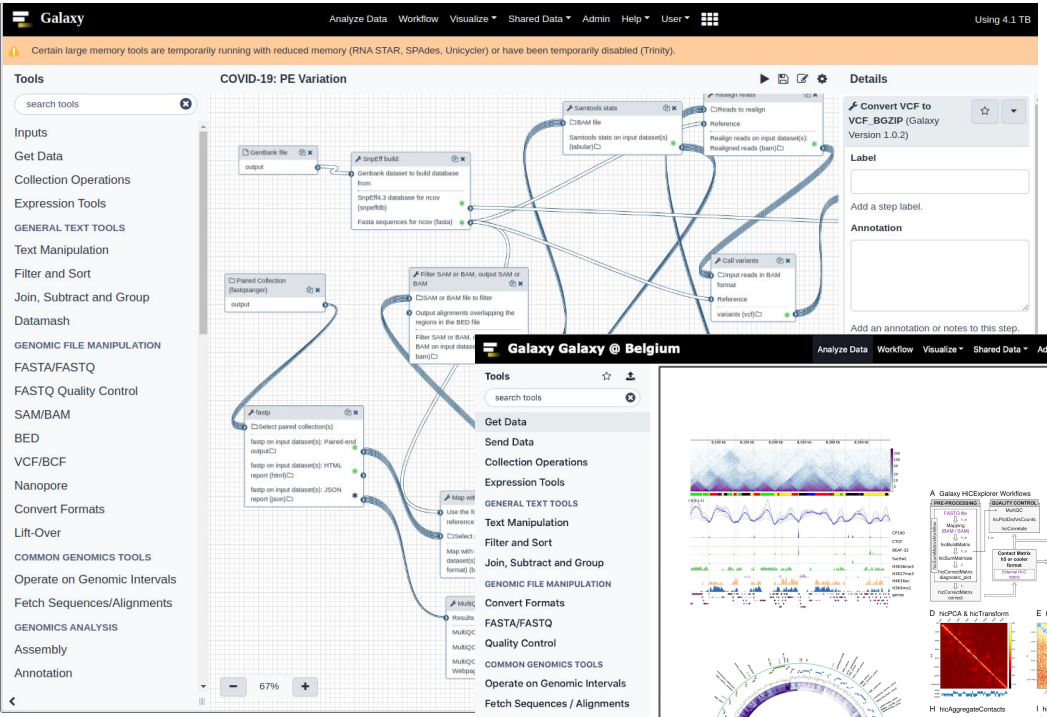
GMOD

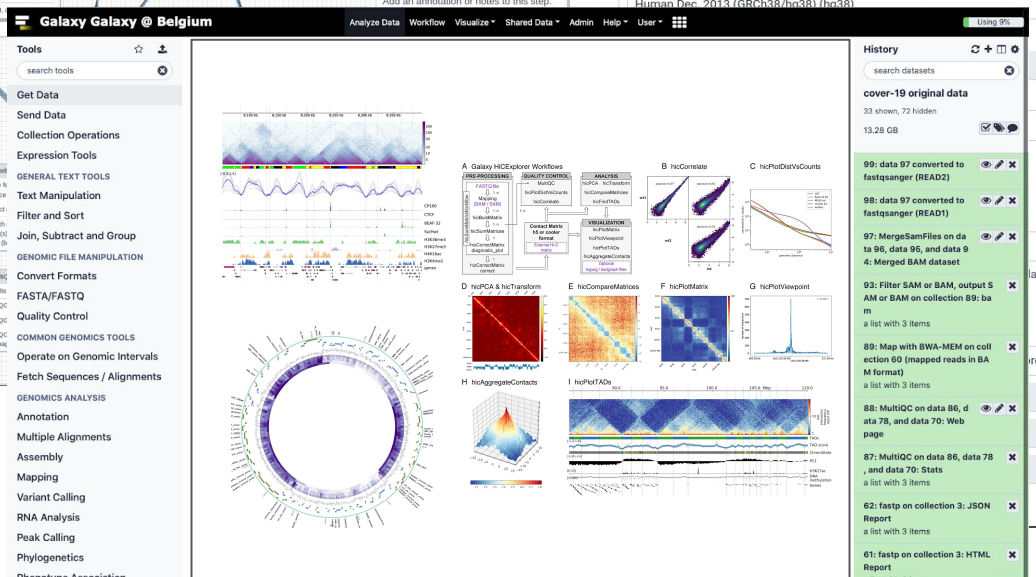
A Data Analysis Gateway for Everyone!

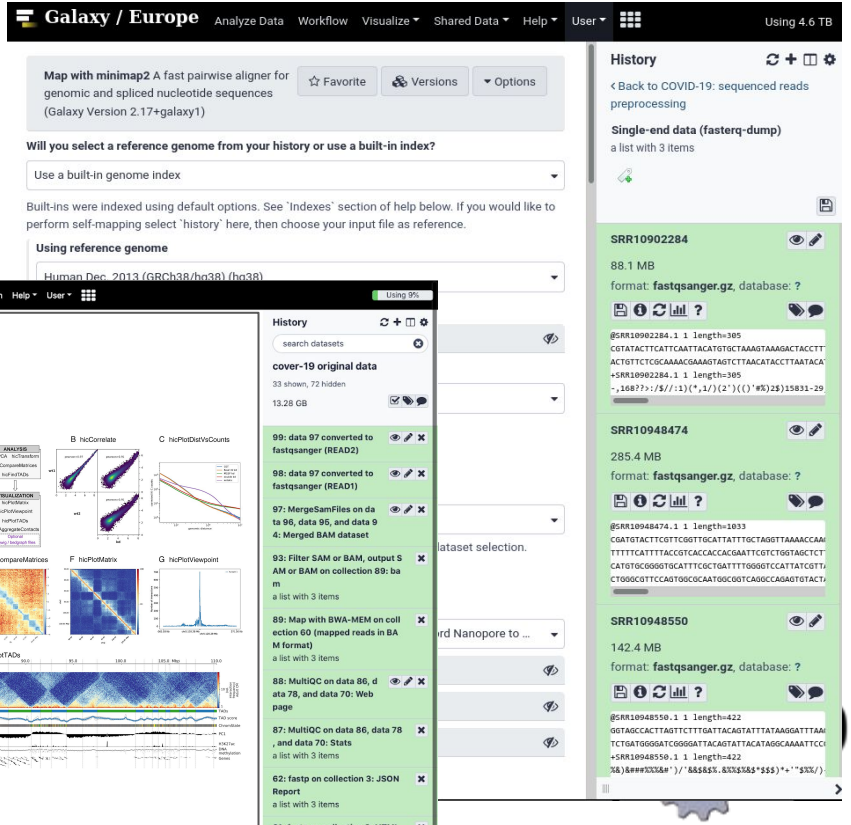


GMOD

Web-based User Interface (+ API access)







Tools

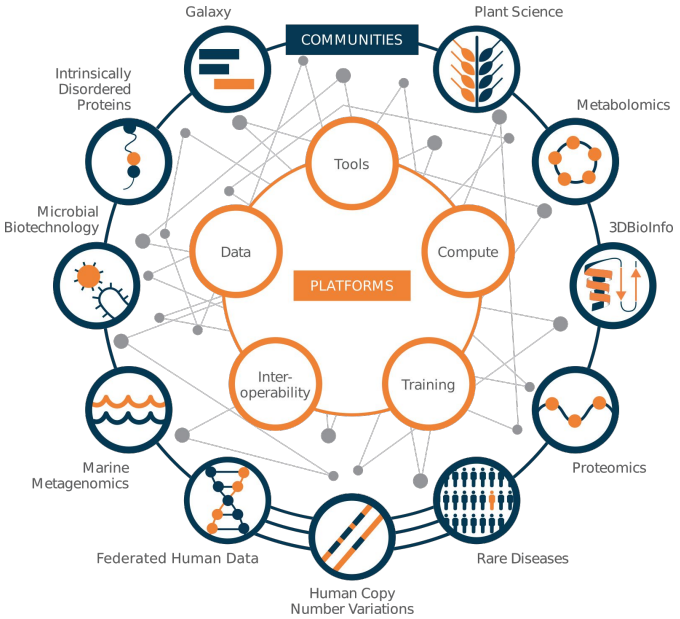
- ~8000 tools available
- App Store-like distribution system
- Can run in Conda, Containers, modules, etc.
- DOTADIW philosophy

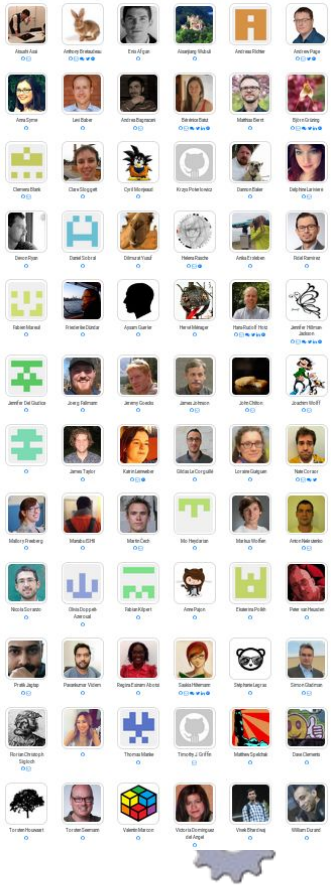
BIOCONDA

CONDA-FORGE

Communities sharing one coherent framework

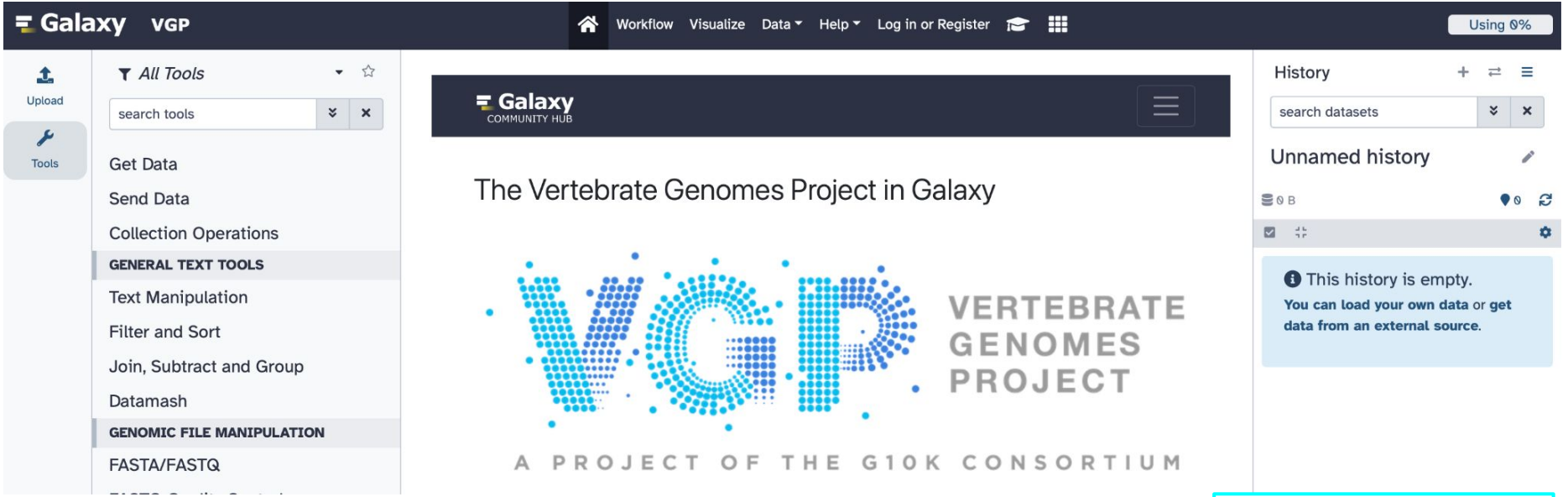
- rna.usegalaxy.eu
- clipseq.usegalaxy.eu
- metagenomics.usegalaxy.eu
- hicexplorer.usegalaxy.eu
- cheminformatics.usegalaxy.eu
- proteomics.usegalaxy.eu
- imaging.usegalaxy.eu
- metabolomics.usegalaxy.eu
- ecology.usegalaxy.eu
- nanopore.usegalaxy.eu
- singlecellomics.usegalaxy.eu
- humancellatlas.usegalaxy.eu
- virology.usegalaxy.eu
- climate.usegalaxy.eu
- streetscience.usegalaxy.eu
- mi.usegalaxy.eu
- annotation.usegalaxy.eu
- assembly.usegalaxy.eu





HUMAN CELL ATLAS

ERGA
EUROPEAN REFERENCE GENOME ATLAS



Who pays for computation?

The computational resources required for assembly are supported by **public** computational infrastructure. In turn, this computational infrastructure is brought you by:

- **EU** - de.NBI, Uni-Freiburg, EOSC and ELIXIR
- **US** - ACCESS-CI, TACC, Jetstream2 (additional funding is provided by NSF and NIH)
- **Australia** - Australian BioCommons, QCIF, Melbourne Bioinformatics, AARNet

Not just for vertebrates!

https://vgp.usegalaxy.org

GMOD

Galaxy resources

http://galaxyproject.org/

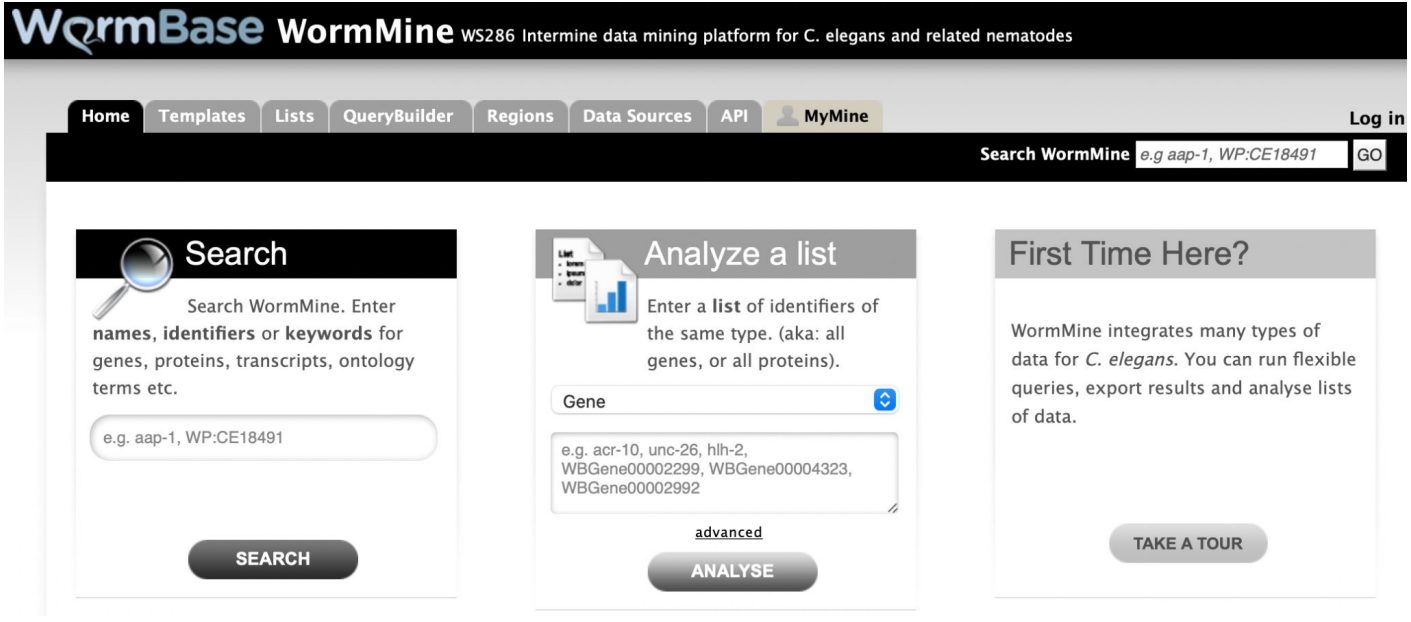
http://usegalaxy.org/

And especially

http://usegalaxy.eu/

InterMine info

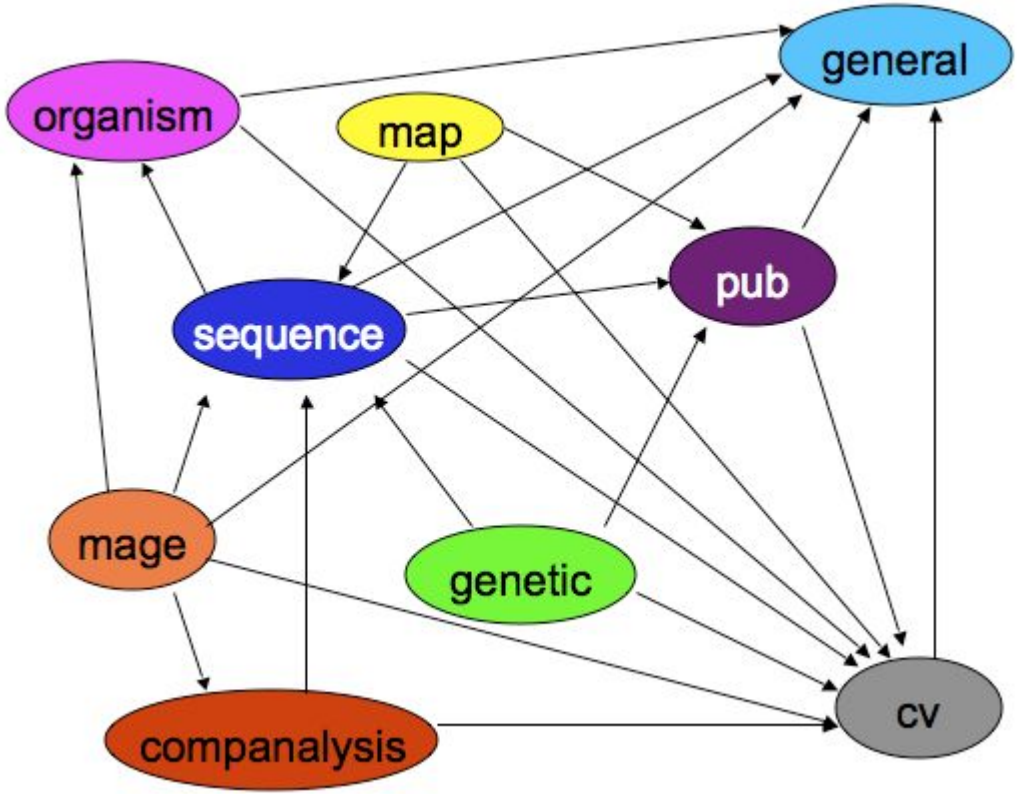
- Takes a variety of biological data in, denormalizes it, and makes it queriable in a very fast, powerful way.
- Used by many MODs (WormMine, YeastMine, FlyMine, etc)
- http://intermine.org



Intro to Chado

- Chado is a highly normalized relational database schema (most often used in PostgreSQL)
- Designed to support multiple organisms' genomes (organism agnosticly)
- Makes heavy use of ontologies and controlled vocabularies to allow a wide variety of data and metadata types.
- Originally developed (and still in use by) FlyBase.

Chado diagram



Chado is most often used in conjunction with …

What is Tripal?

An open-source Biological Database toolkit that

Aids in the creation of community-centric, data-driven websites

Facilitates FAIR data sharing

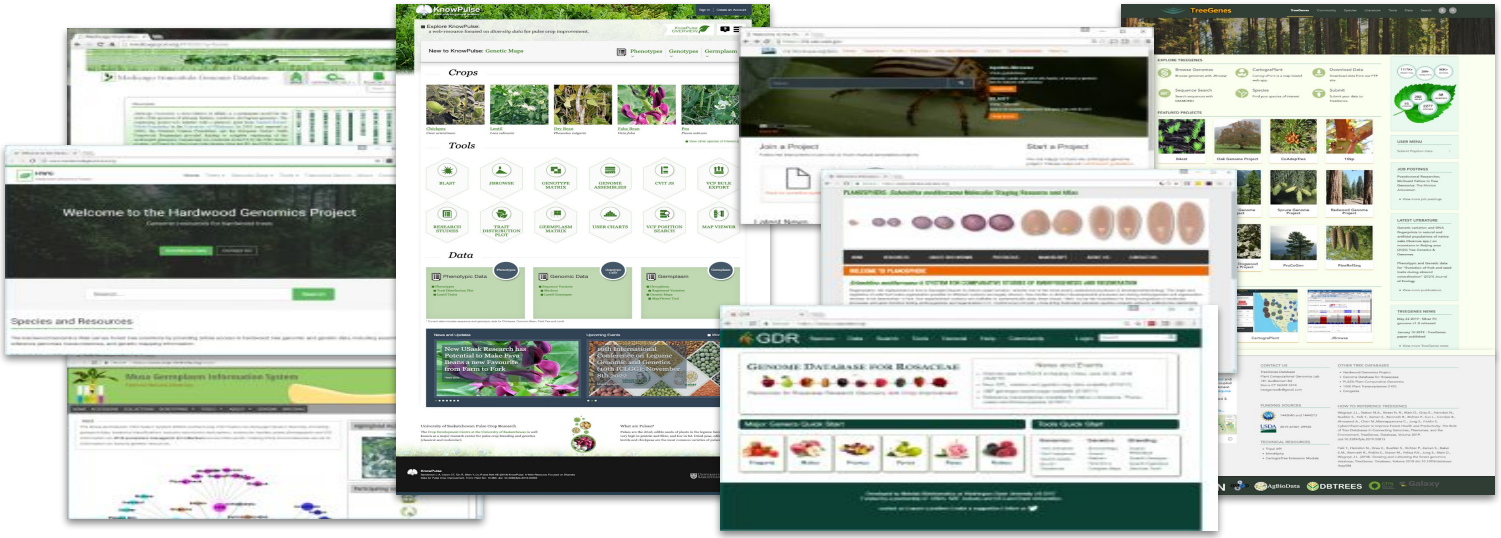
- Ontology-Focused
- Consistent web services

More features through shared development

- Focus on what makes your community unique

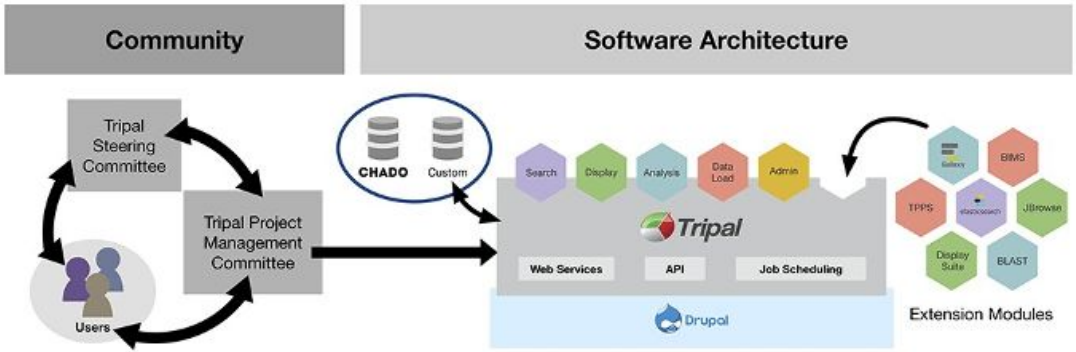
Tripal Community

- Over 125 sites report using Tripal
- Species include plants, animals, insects, bacteria



Tripal Structure

- Easily extendable with rich APIs, documentation and core services
- Open-source to the core: integrates with and extends many other open source projects including Drupal, Chado, Galaxy, JBrowse, Elastic Search, etc.
- Community developed with users and developers forming the Project Management and Steering committees: organized and community powered



Briefings in Bioinformatics, Volume 22, Issue 6, November 2021, bbab238, https://doi.org/10.1093/bib/bbab238

Tripal Features

Rich Pages for biological data.

- Extensive, flexible metadata
- Links to related data, data summaries, context
- Bulk Import or individual creation + curation

Flexible Searching makes it easy to use rich metadata to focus your results.

Robust security managed by the world-wide Drupal community.

- Keep data private until published,
- Share with collaborators using user roles + permissions

Both science and community focused extensions through Tripal and Drupal to customize your site.

Tripal Resources

http://tripal.info/

http://tripal.info/join/slack

Some Tripal examples:

https://knowpulse.usask.ca

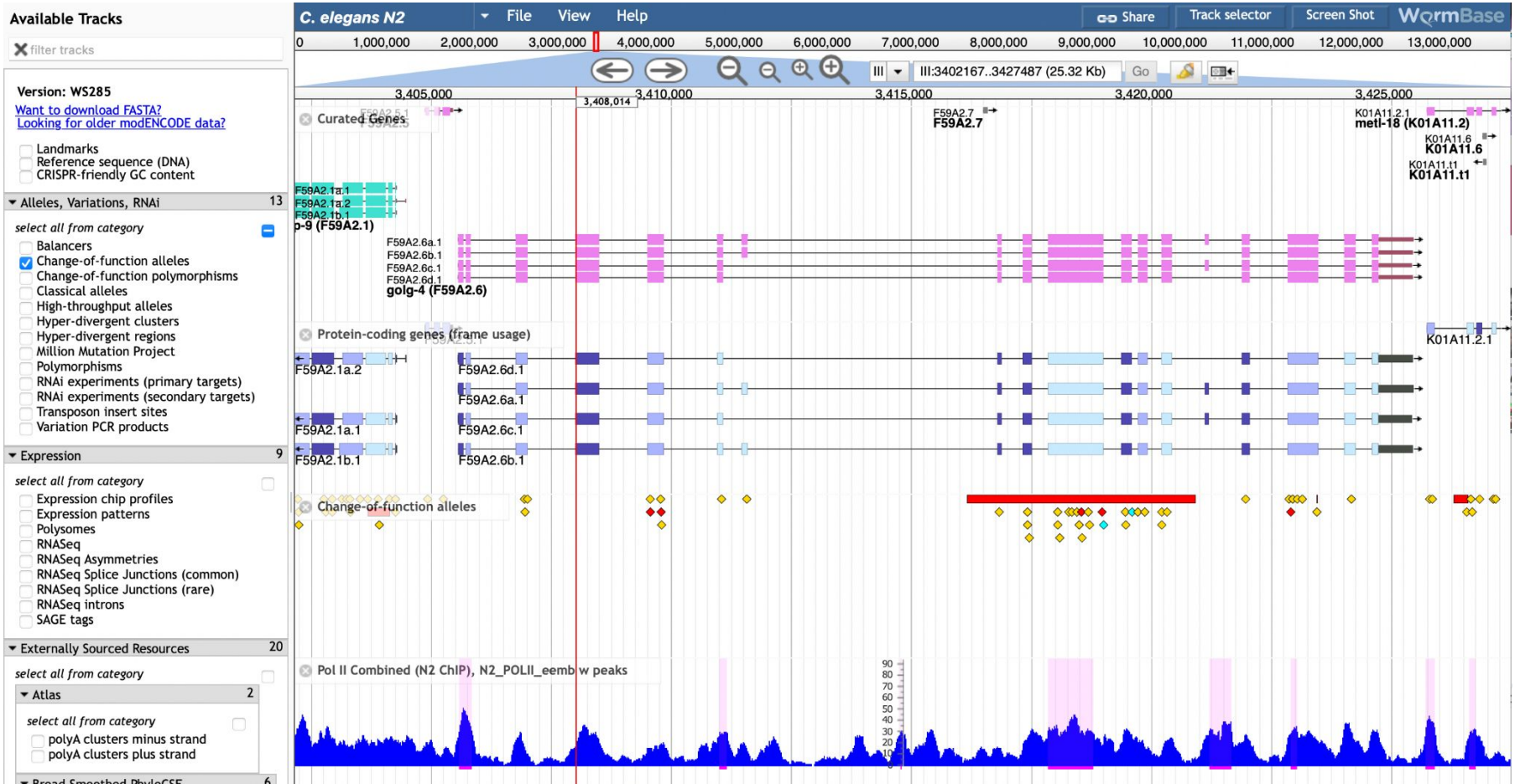
https://www.rosaceae.org

https://planosphere.stowers.org/smedgd

Intro to JBrowse

- A JavaScript-powered genome browser (think IGV or UCSC genome browser, but faster, more responsive and much prettier).
- Originally developed ~2009 to replace the Perl CGI-based Generic Genome Browser (GBrowse) using what is now a pretty outdated JavaScript framework.
- Recently redeveloped in ReactJS and given MUCH more functionality
- Reads many common genomic formats: GFF, VCF, BigWig, BigBed, HiC, CRAM, BAM, PAF, FASTA
- Available as a web app or a desktop application

Old JBrowse

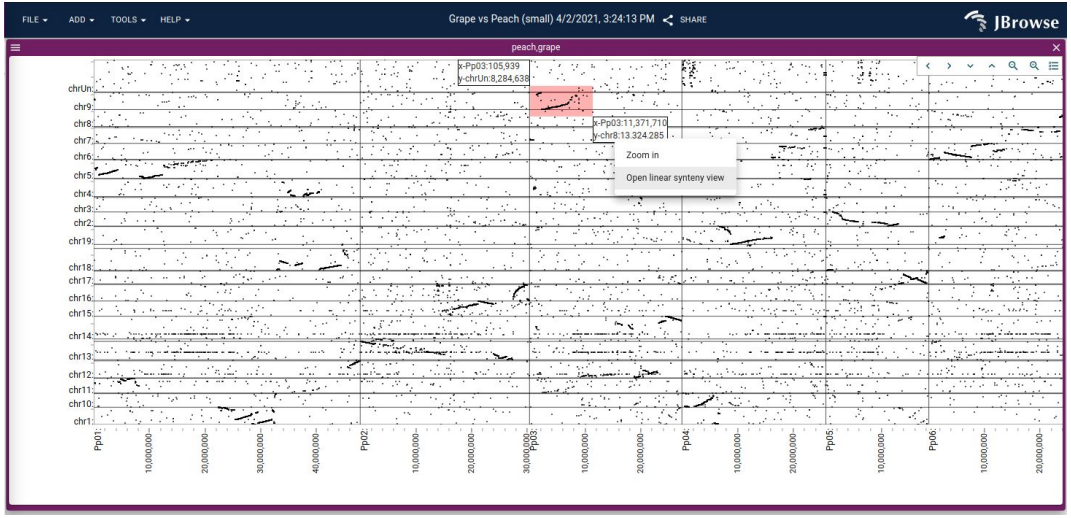

Available Tracks
filter tracks
Version: WS285
Want to download FASTA?
Looking for older modENCODE data?
Landmarks
Reference sequence (DNA)
CRISPR-friendly GC content
Alleles, Variations, RNAi 13
select all from category
Balancers
Change-of-function alleles
Change-of-function polymorphisms
Classical alleles
High-throughput alleles
Hyper-divergent clusters
Hyper-divergent regions
Million Mutation Project
Polymorphisms
RNAi experiments (primary targets)
RNAi experiments (secondary targets)
Transposon insert sites
Variation PCR products
Expression 9
select all from category
Expression chip profiles
Expression patterns
Polysomes
RNASeq
RNASeq Asymmetries
RNASeq Splice Junctions (common)
RNASeq Splice Junctions (rare)
RNASeq introns
SAGE tags
Externally Sourced Resources 20
select all from category
Atlas 2
select all from category
polyA clusters minus strand
polyA clusters plus strand
C. elegans N2
File
View
Help
Share
Track selector
Screen Shot
WormBase
III:3402167..3427487 (25.32 Kb)
Go
Curated Genes
F59A2.7
K01A11.2.1
metl-18 (K01A11.2)
K01A11.6
K01A11.t1
F59A2.1a.1
F59A2.1a.2
F59A2.1b.1
p-9 (F59A2.1)
F59A2.6a.1
F59A2.6b.1
F59A2.6c.1
F59A2.6d.1
golg-4 (F59A2.6)
Protein-coding genes (frame usage)
Change-of-function alleles
Pol II Combined (N2 ChIP), N2_POLII_eemb w peaks

New functionality/views in JBrowse 2

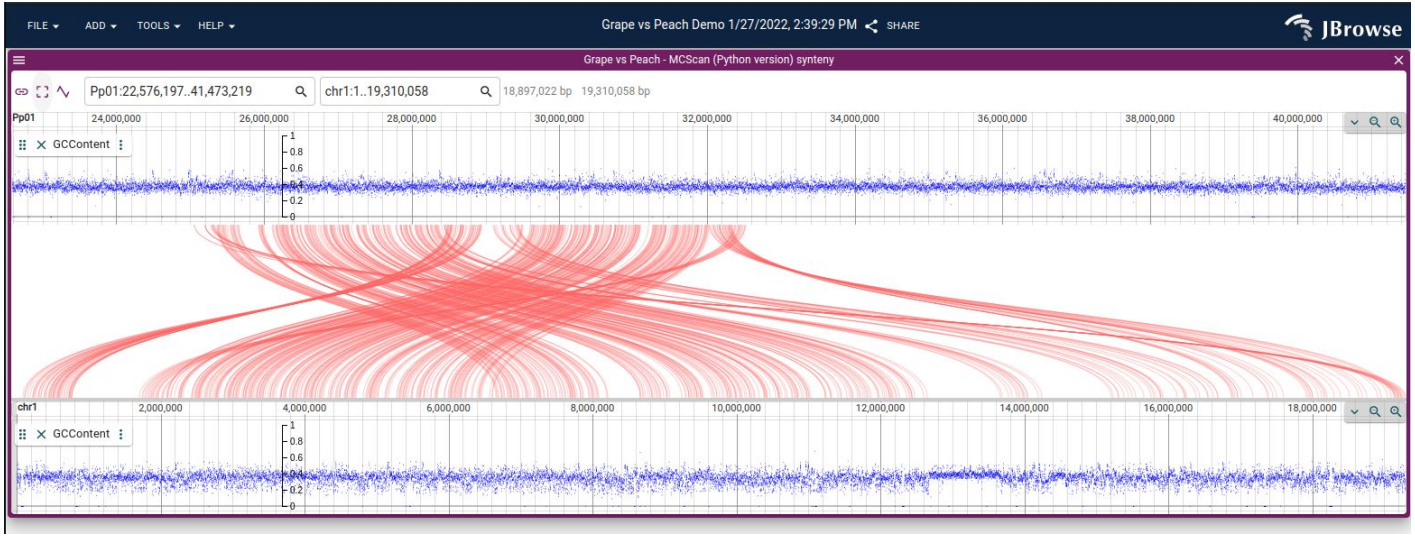
- Visualize comparative genome data with dot plot and synteny views
- Visualize structural variation with circular views and “split” linear genome views
- Use a spreadsheet view to sort and filter features and then click to get context in linear genome views.
- JEXL (JavaScript EXpression Language) allows both admins and users to create callbacks that modify views.
- Plugins written in ReactJS allow a lot of customization: new view types, new track types, new JEXL functions

Comparative Genomics

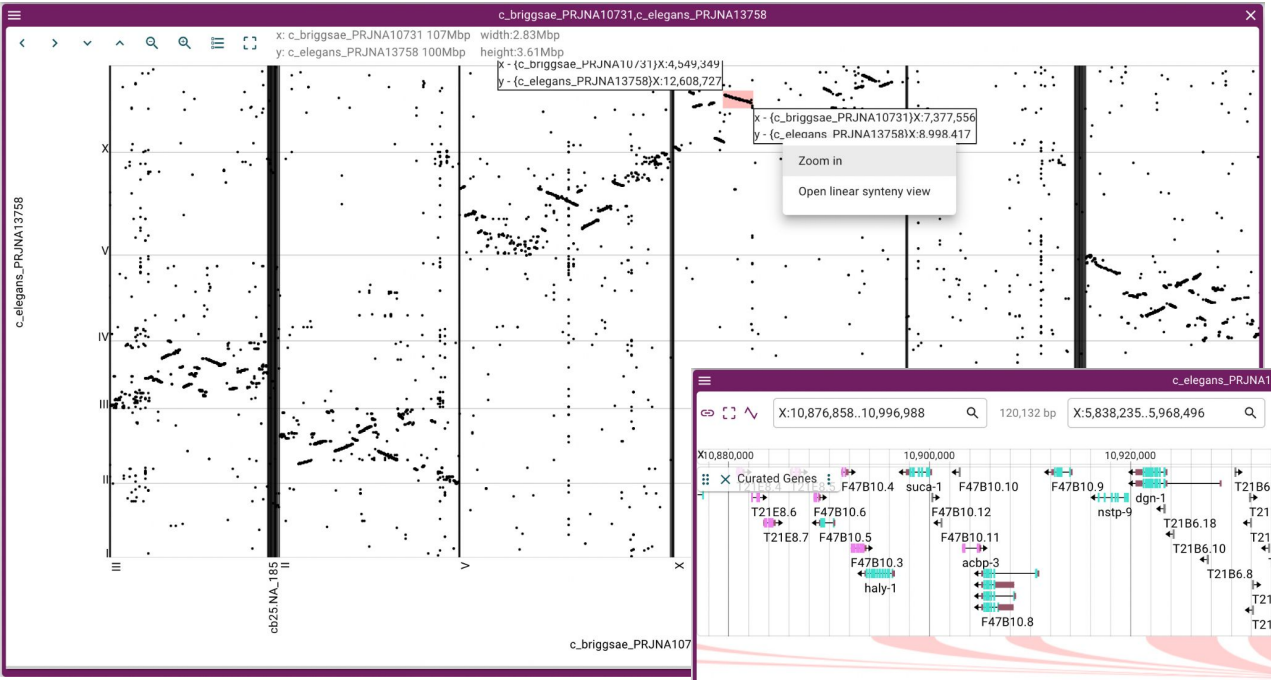
Synteny view



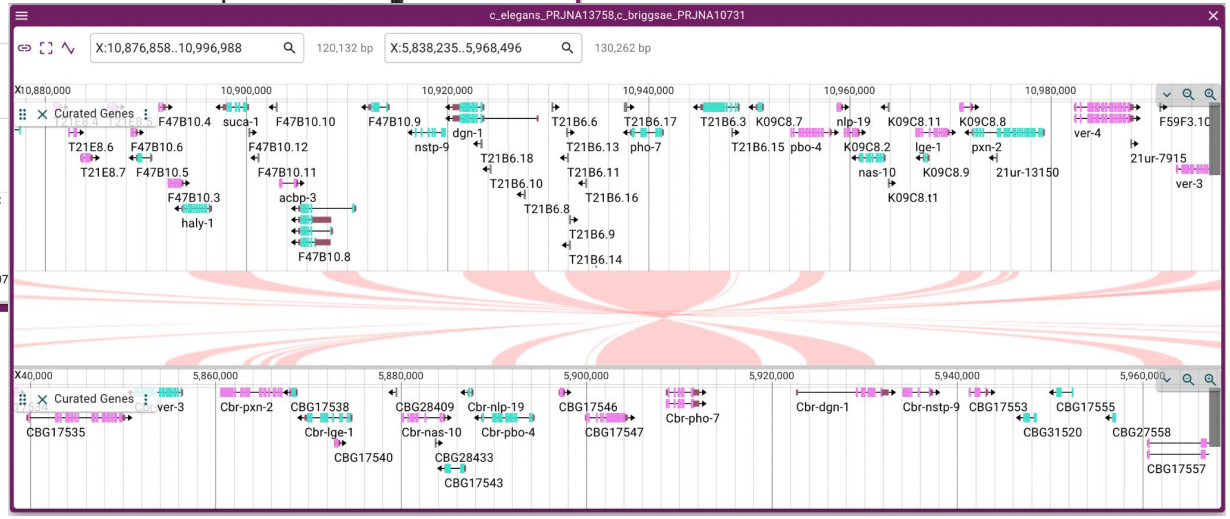
Dotplot view



Dotplot and synteny views



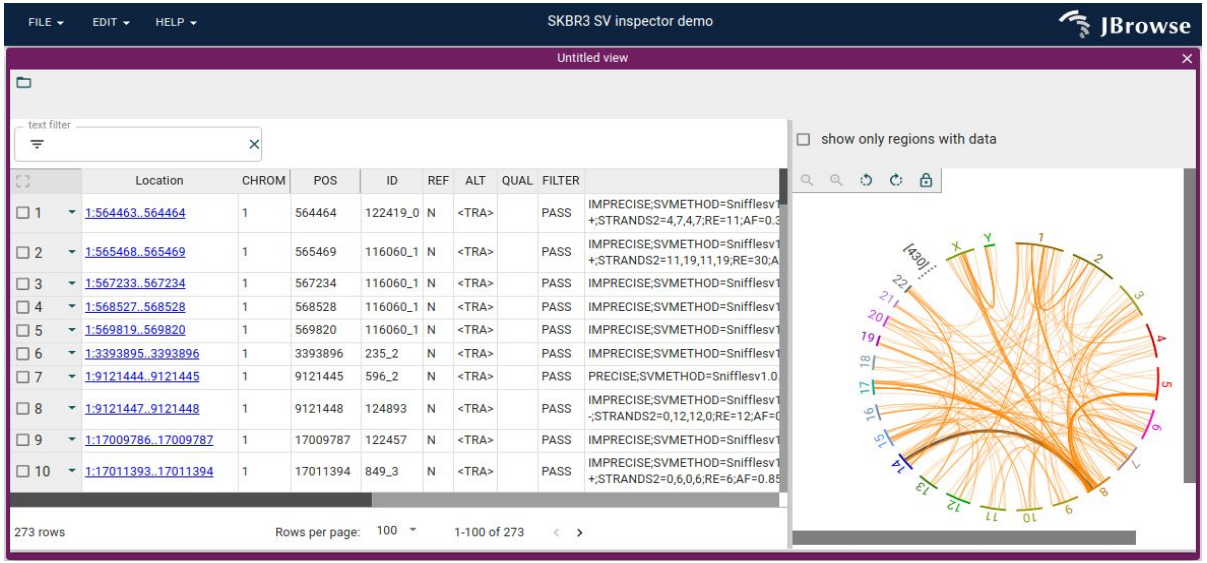
Dotplot comparing *C. elegans* and *C. briggsae* genomes



Synteny view showing about 100kb of elgans and briggsae genomes, showing an inversion.

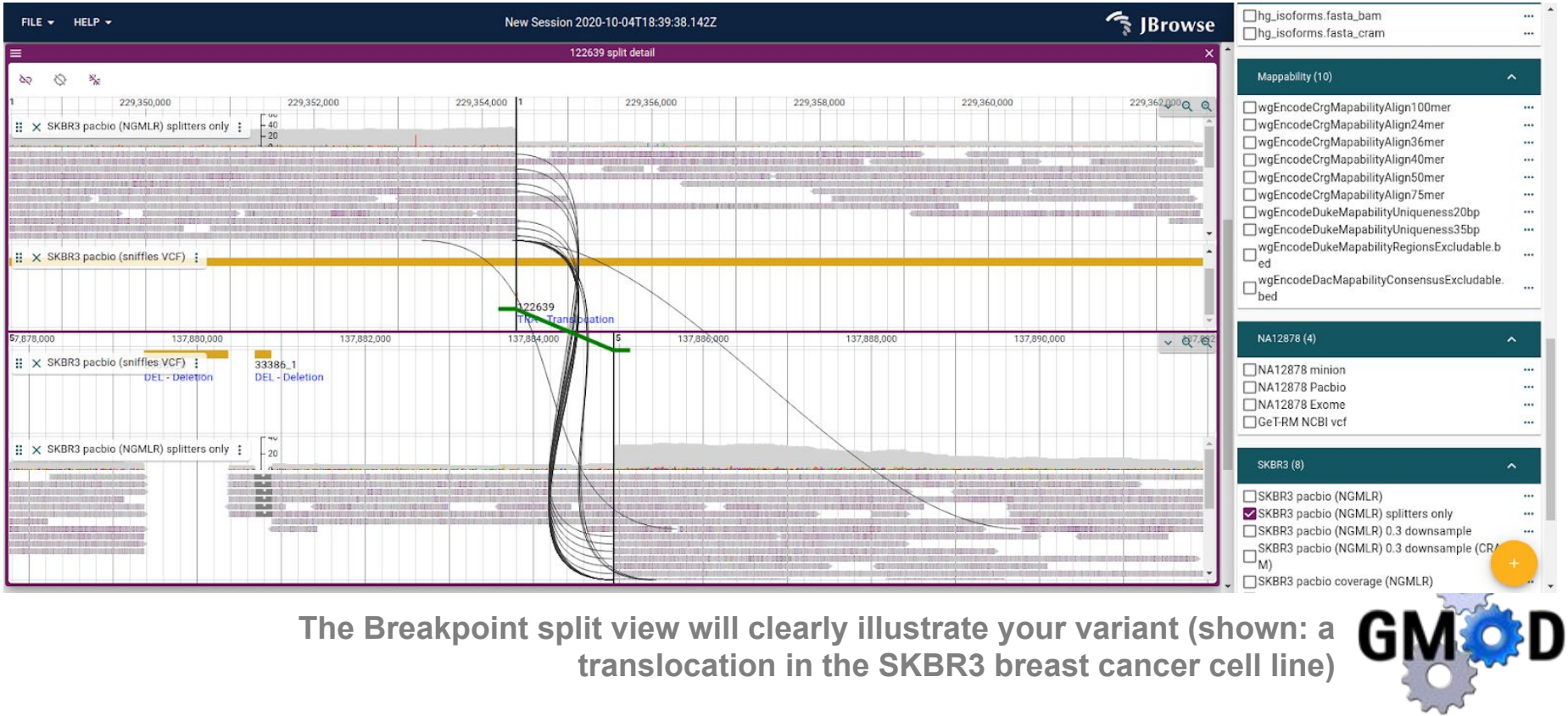
The Structural Variant Inspector

- Provide JBrowse with a tabular file of structural variants and it will produce an interactive series of views for analysis
- Clicking on a chord on the circular view opens a breakpoint split view



A core component of the SV inspector is the joint spreadsheet and circular views that assist the user in navigating their list of SV's

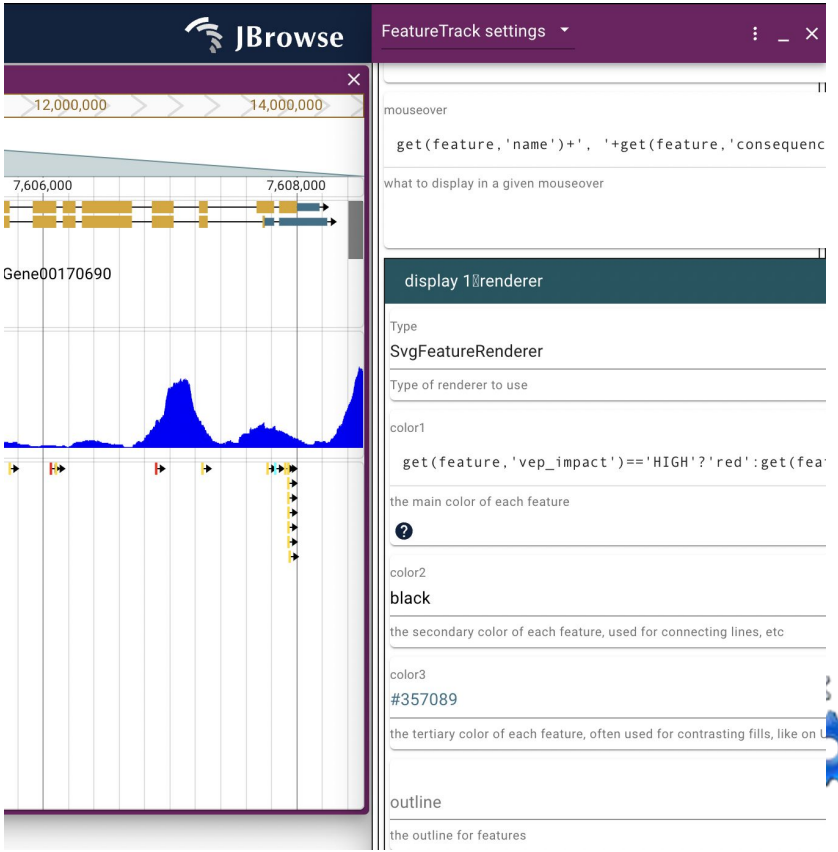
The Structural Variant Inspector



The Breakpoint split view will clearly illustrate your variant (shown: a translocation in the SKBR3 breast cancer cell line)

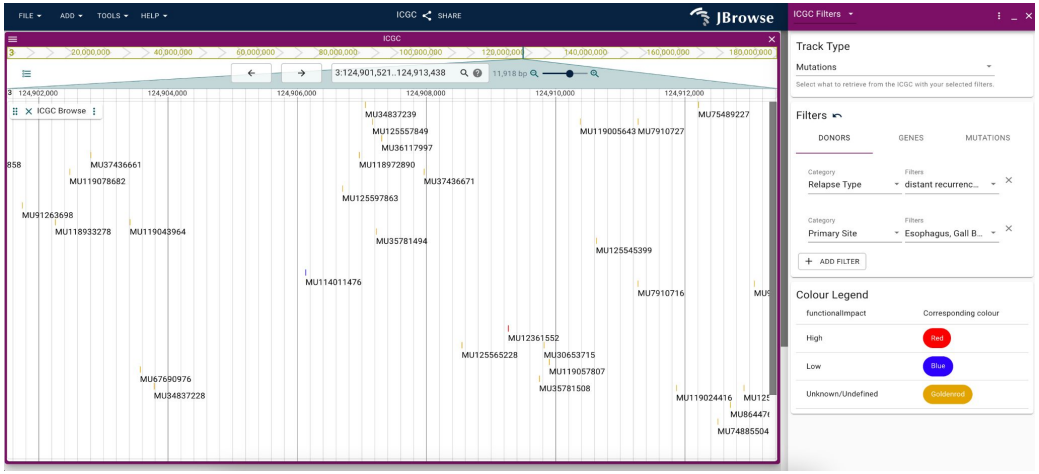
Graphical configuration editor

- Both desktop and web apps include a dynamic graphical configuration editor (ie, changes immediately appear in tracks)
    - “Regular” users are restricted to editing personal copies of tracks
- Also, there is a admin editor (called *jbrowse-admin*) that runs a small web server and allows direct editing of the web app config.
- Config uses JEXL instead of JS for security.



Plugin infrastructure

- Allows supplementary feature sets to be added to an instance of JBrowse
- Nearly all core components of JBrowse can be “plugged in” or extended upon through plugins
- Simple development steps to create your own third party plugin for JBrowse 2



Screenshot of the ICGC plugin: a plugin that retrieves and visualizes consortium data within JBrowse 2

JBrowse 2 as an embedded component

- Python, R and JavaScript interfaces exist so that JBrowse can be used with them
- For example, in Jupyter Notebooks
- Or in web pages that use them

File Edit View Run Kernel Tabs Settings Help

Filter files by name

/

Name	Last Modified
Carsonella...	4 days ago
Finding rar...	4 days ago
Haemophil...	4 days ago
LICENSE	4 days ago
README.md	4 days ago

Haemophilus influenzae_JE

Code

Python 3 (ipykernel)

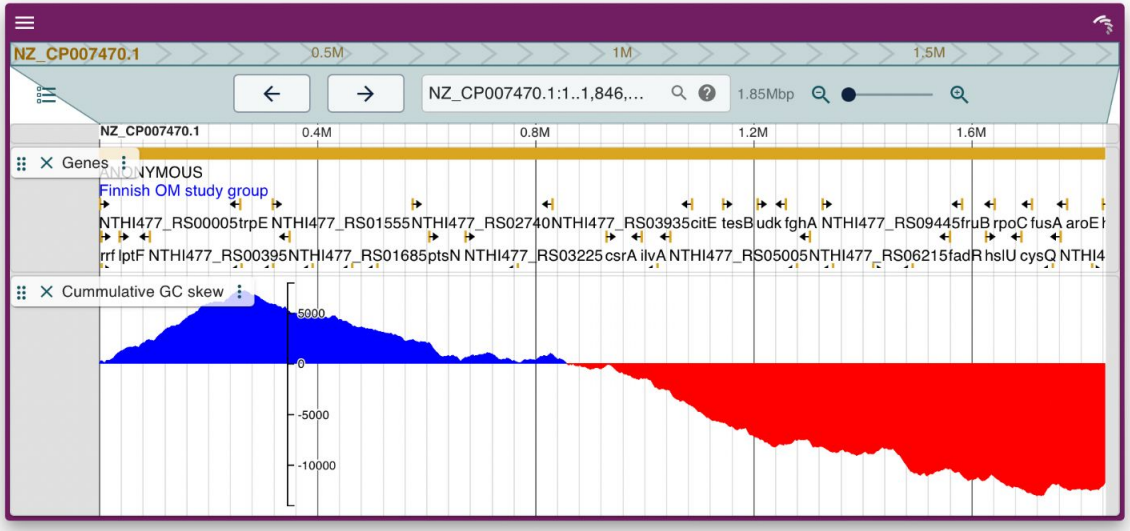
```
config.add_df_track(skew_df,'Cummulative GC skew',track_id='df_gc_skew',overwrite=True)
```

Finally, we set a few things about how we want the initial view of JBrowse to look (location, tracks that are turned on) and then launch the genome browser. Note that as we zoom, it might get a little "jerky," as the config file is fairly large, since all of the data we created above is in the config. If we wanted to make this into something we showed other people, we'd want to create data files for each of these tracks.

[20]:
```
location = refseq_name + ":1000..3500"
print(location)
```

```
NZ_CP007470.1:1000..3500
```

[21]:
```
config.set_location(location)
#config.set_default_session(['Genes','df_gc_skew','df_percent_gc','df_gc_asym'], False)
config.set_default_session(['Genes','df_gc_skew'], False)
full_conf = config.get_config()
launch(full_conf, port=3003, height=600) #height sets the height of the iframe containing JBrowse, but
                                         # JBrowse sets the height of itself. To use all of the height allocated,
                                         # you have to manually drag the bottom edge of jbrowse down.
```



GMOD

Gene » *etr-1*

Overview

Served from Datomic

etr-1 **(ELAV-Type RNA binding-protein family)**

Predicted to enable RNA binding activity. Involved in determination of adult lifespan. Located in nucleus. Expressed in body wall musculature; cloacal sphincter muscle; copulatory spicule; non-striated muscle; and somatic gonad. Human ortholog(s) of this gene implicated in developmental and epileptic encephalopathy. Is an ortholog of human CELF1 (CUGBP Elav-like family member 1).

evidence

- Legacy manual gene description
- Also refers to
- Curatorial remarks

Species: *Caenorhabditis elegans*
Sequence: T01D1.2
Other name: CELE_T01D1.2
Type: protein coding
Gene class: etr
Status: Live
Clone: T01D1
Parent seq: CHROMOSOME_II
Gene name evidence: Jonathan Hodgkin
Comparative Info: Integrated model organism details available at the ALLIANCE of GENOME RESOURCES
WormBase ID: WBGene00001340

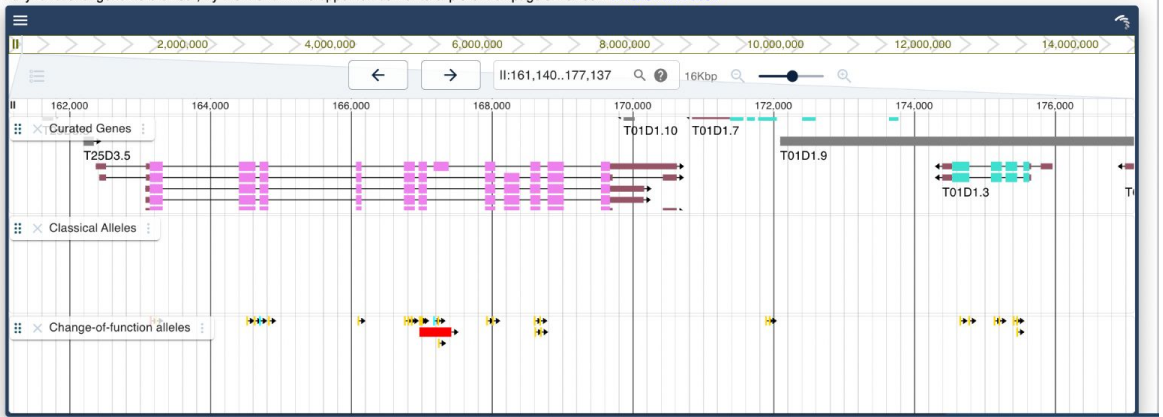
Location

Served from Datomic

Genetic position: II:-15.90 +/- 0.014 cM

Genomic position: II:162374..170631 (*Legacy Genome Browser:* II:162374..170631)

JBrowse 2 Beta: Fully functional genome browser; try the menu in the upper left corner to explore. Full page JBrowse2: II:162374..170631



Embedded JBrowse is essentially fully featured

- Embedded JBrowse works as a complete genome browser
- You can add/remove tracks
- Pan/zoom
- Go to other locations

JBrowse 2 resources

- http://jbrowse.org/
- https://github.com/GMOD/jbrowse-components - primary repo
- https://github.com/GMOD - many other JS repos
- https://github.com/GMOD/jbrowse-jupyter
- https://github.com/scottcain/jbrowse-jupyter-examples
- https://gitter.im/GMOD/jbrowse

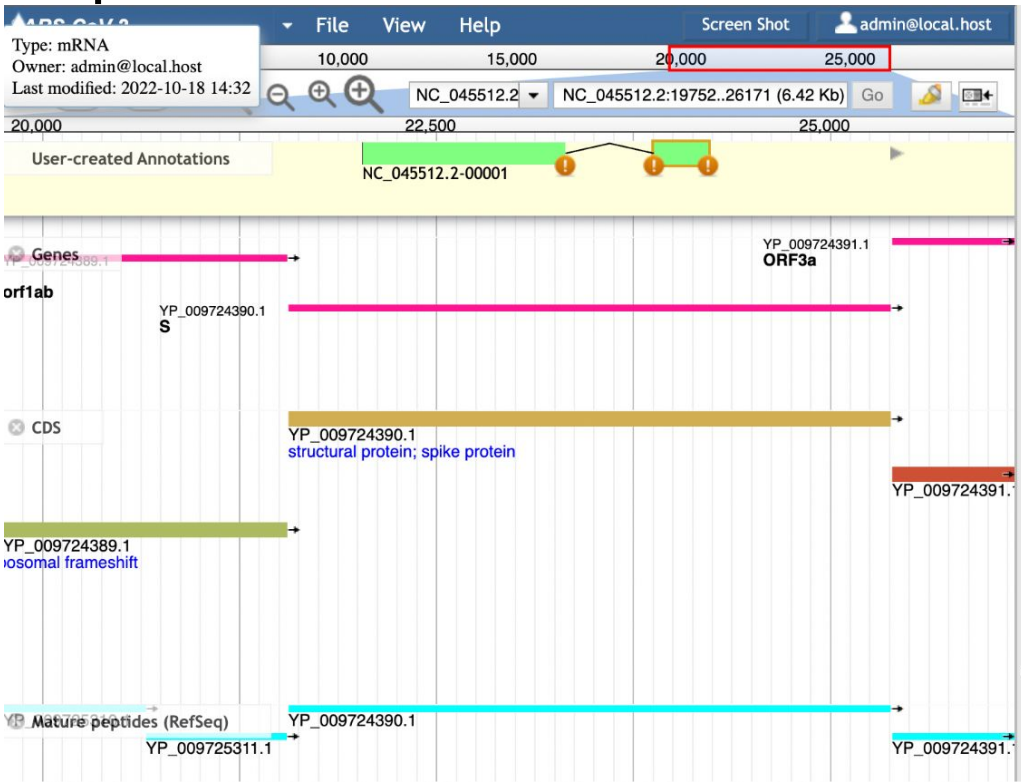
Intro to Apollo

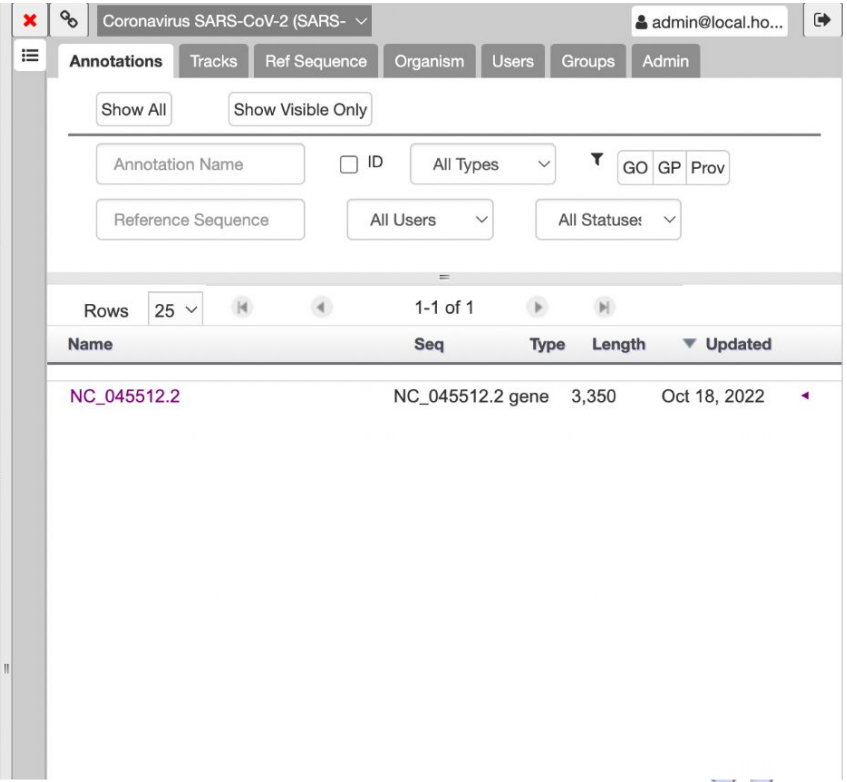
- Apollo is a graphical genome feature structure editor (ie, so you can create and modify features like genes, transcripts, etc)
- First incarnation was a Java desktop application written by FlyBase in the early 2000s (“Apollo 1”).
- Rewritten as a JBrowse 1 plugin several years ago (“WebApollo” and “Apollo 2”)
- Currently being rewritten as a JBrowse 2 plugin (“Apollo 3”)
- Apollo 2 and 3 are both live and interactive like Google Docs (so “gene editing wars” are possible)

Who uses Apollo?

- Professional curators at model organism databases
- “Casual” users who are interested in “their” gene or gene family
- “Annotation Jamboree” users (locked in a room with a group for a few days to annotate a genome)
- Students learning about genomics

Apollo user interface





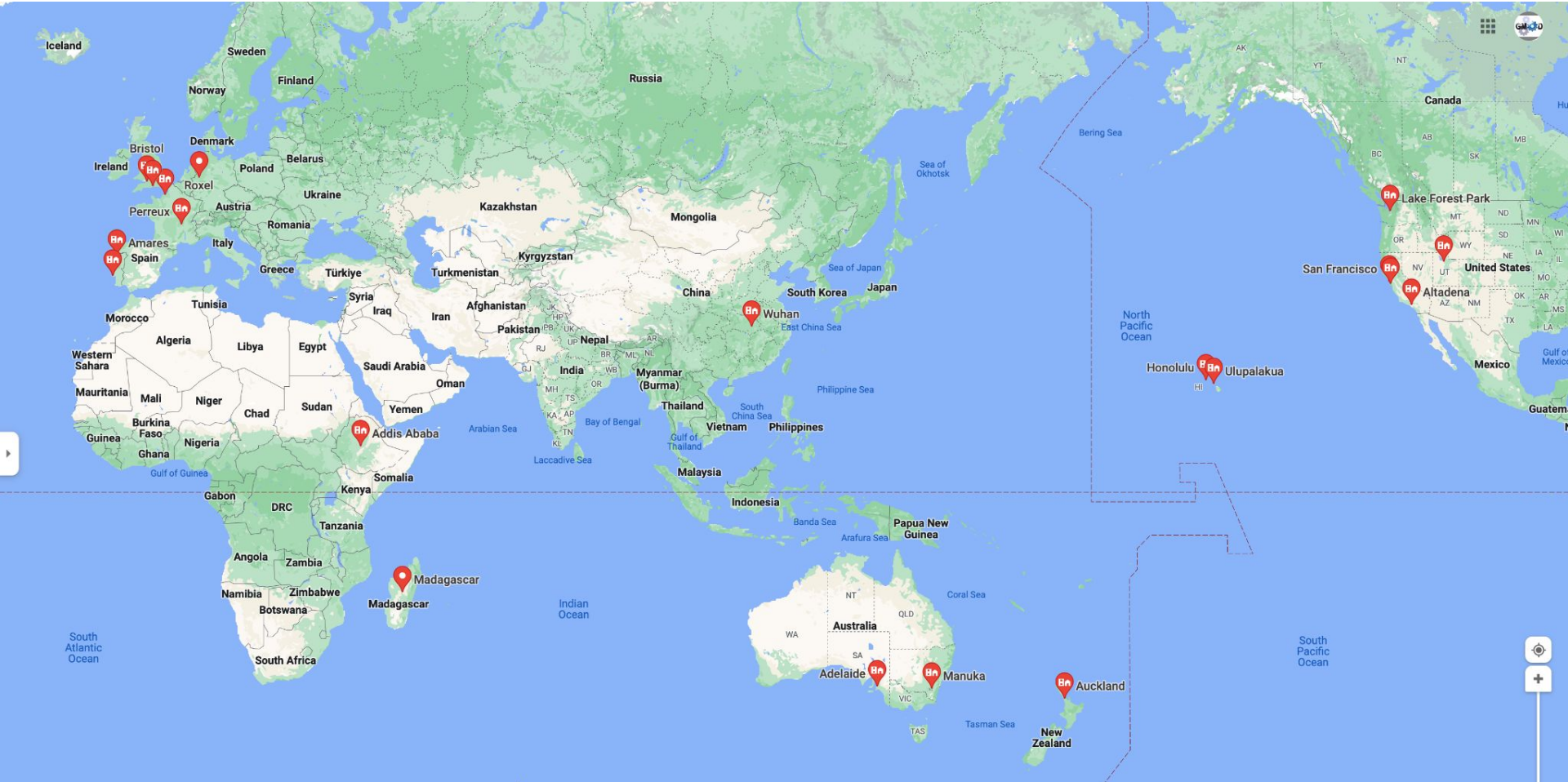
Apollo resources

- Docs: https://genomearchitect.readthedocs.io/en/latest/
- Source: https://github.com/GMOD/Apollo/

A little about a “side” project I’m working on

Building a web portal for …

World Wide Worms (WWW)!



Err, what?

- Cristian Riccio, researcher from Cambridge University, contacted WormBase with a proposal:
  - He was wrapping up his PhD dissertation that involved natural diversity of *C. elegans*
  - He had 20 fully assembled genomes from worms around the world
  - And felt that other worm researchers would be interested in this data set
- The feeling I get is that most of the people at WormBase had the response along the lines of “Maybe…?”
- Then it landed in my lap, and a plan began to form.

Build a stand alone portal

- There isn’t really a good way to incorporate data like this into “WormBase proper”
- That’s part of why my colleagues at WB weren’t exactly sure what to do with this data
- As an alternative, I suggested a portal that was either completely or mostly a JBrowse instance with comparative data between the various strains

Never underestimate the bandwidth…

of a station wagon full of tapes hurtling down the highway. ~~Andrew S. Tannenbaum

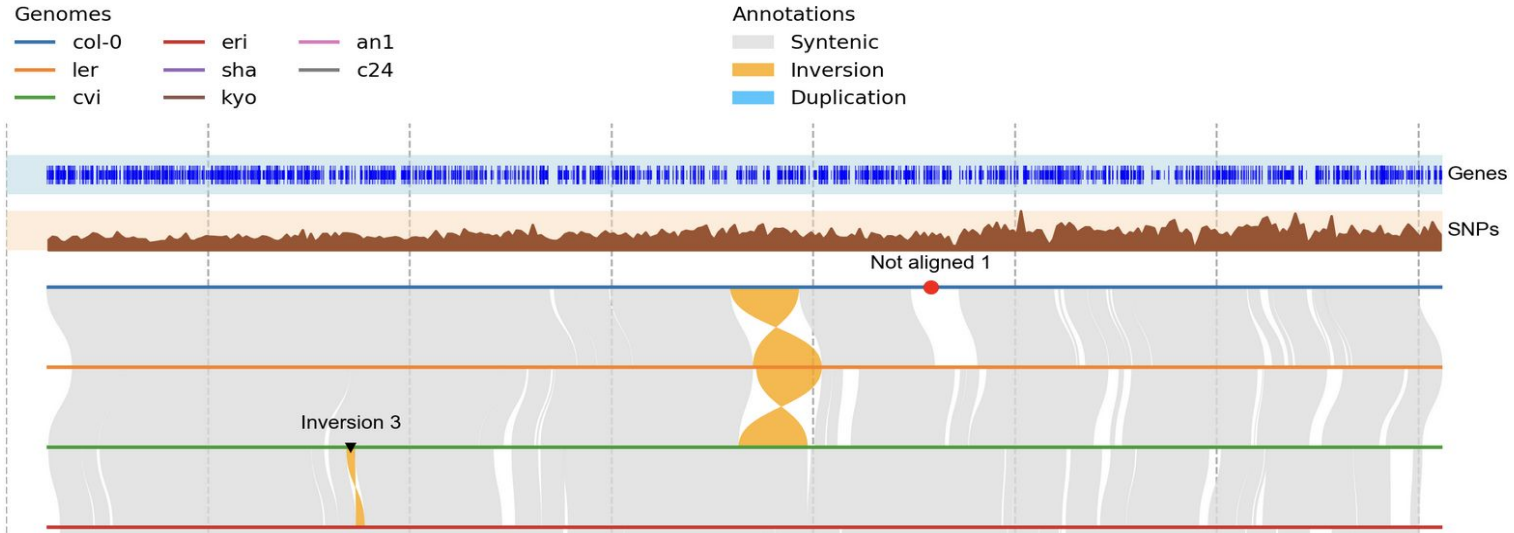
So I got a hard drive with (just) a few hundred GB of annotations, assemblies and mapping data sent via post from Switzerland.

How to best guide users to things that might be interesting?

- Of course, users could search for a gene and compare that between different strains, but most of the time, those are going to be exactly or nearly exactly the same between them
- Find the globally visible differences between the assemblies.
- Already familiar with minimap2 (Heng Li) (JBrowse natively displays results from it) but it didn’t quite do the trick visually for largely identical assemblies.

Found SyRI and PlotSR

- Compares alignments between two chromosome-level assemblies and identifies synteny and structural rearrangements.
- https://github.com/schneebergerlab/syri
- Goel, M., Sun, H., Jiao, W. et al. SyRI: finding genomic rearrangements and local sequence differences from whole-genome assemblies. Genome Biol 20, 277 (2019) doi:10.1186/s13059-019-1911-0



Imagine multiple slides here

Of me complaining about the difficulties of setting up python applications to run inside of Docker containers. Suffice it to say, I was quite whiny but got it working in the the end.

But Python

- Really not a fan.
- Wanted in insulate me from the hassles of Python environments
- So build a Docker container

But…

- It turns out, installing Anaconda in a Docker container is a hassle of its own special variety.
- (And yes, I could build it from source in the Docker container, but I didn't have much success doing that on the command line; I really didn't want to puzzle through it in Docker.)

In a base container…

```
# install miniconda
ENV CONDA_DIR /opt/conda
RUN wget --quiet https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh -O ~/miniconda.sh && \
        /bin/bash ~/miniconda.sh -b -p /opt/conda
ENV PATH=$CONDA_DIR/bin:$PATH

#RUN conda update conda
# install syri and plotsr
COPY environment.yml .
RUN conda env create -f environment.yml

RUN echo "conda activate syri_env" > ~/.bashrc
```

```
name: syri_env
channels:
  - conda-forge
  - bioconda
dependencies:
  - python=3.9
  - syri
  - plotsr
```

Turns out, though, putting in .bashrc isn’t enough

Due to the way Docker containers are virtualized, the .bashrc isn’t automatically used when executing commands later in the Docker file.

To get around this, write a shell script that explicitly sources it (in ENTRYPOINT):

```
#…run minimap2 on the assemblies…

conda init bash

source /root/.bashrc

conda activate syri_env

#... now go on to run syri and plotsr…
```

Why create a base container

- Want a container that has minimap2, syri and plotsr installed and then do analysis
- This takes about 15 minutes to build, making debugging turn around slooooow
- Once the details around installing Anaconda were sorted, installing the apps is done, so creating a base container with these installed makes the debug turnaround time seconds instead.
- Main container just copies the entrypoint script and executes it.

So the real work is in the ENTRYPOINT

```
FROM minimap_syri_base as app

COPY entrypoint.sh .

WORKDIR /data

ENTRYPOINT ["../entrypoint.sh"]
```

That’s the whole thing.

What the entrypoint script does

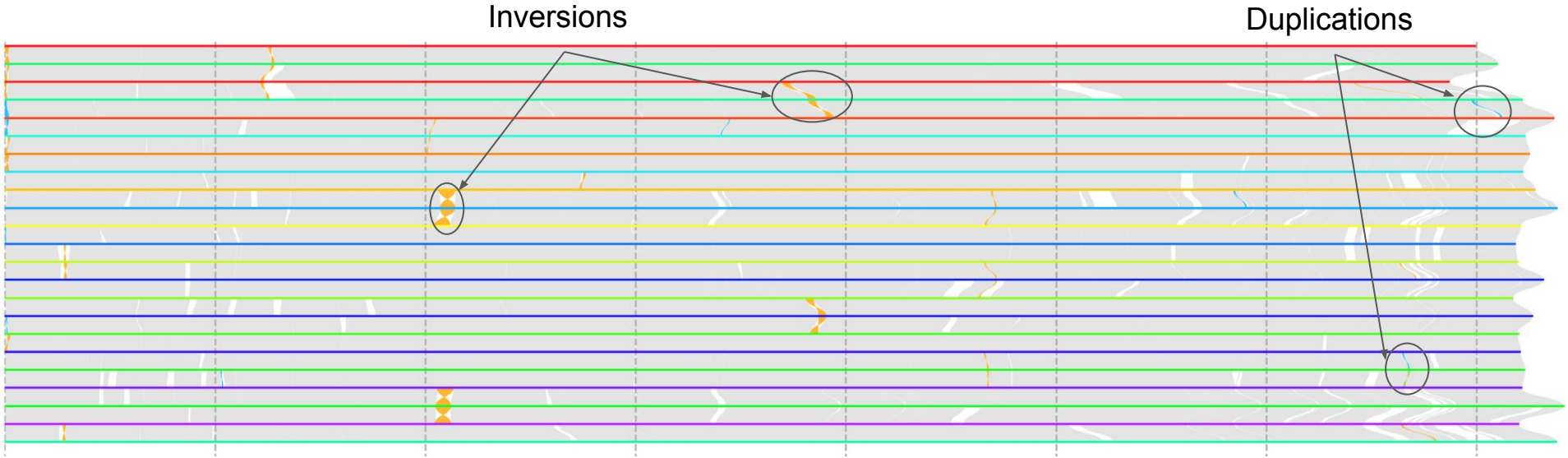
- Run “serial pairwise” minimap2 (A vs B, B vs C, C vs D, etc) (n-1 executions)
- Activate conda
- Run syri on the minimap2 output for each pair (n-1 executions)
- Run plotsr once on all of the syri outputs.

Tall output

The result is all 21 assemblies stacked by chromosome (I through V, plus X)

Output can be SVG, PDF or PNG

plotsr output (chromosome II)

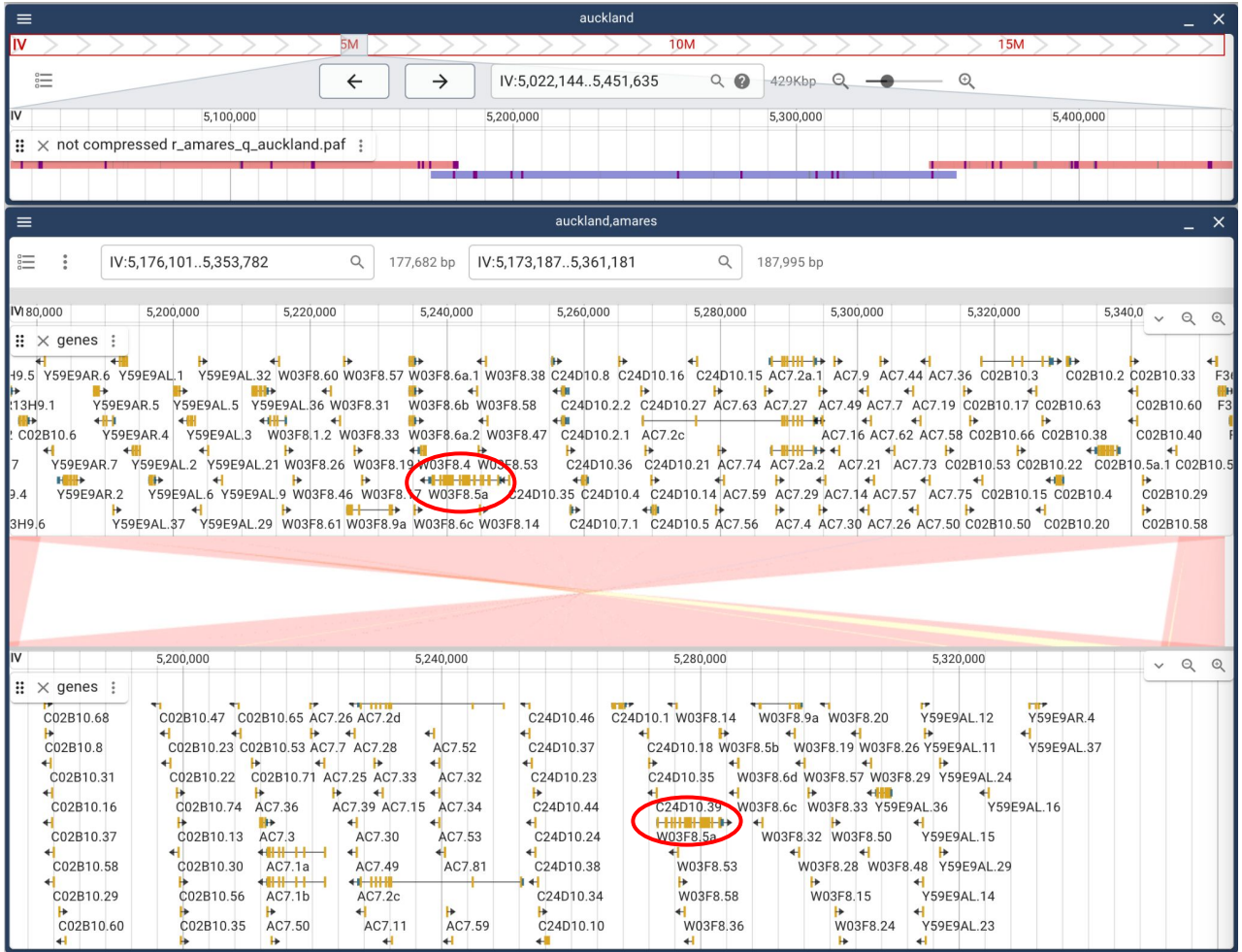


Probably also translocations and deletions in here (hard to tell what's going on in the gaps)

Idea: Break into per-chromosome images and make them into image maps

(That is, a set of defined coordinates on an image that, when clicked on, go to a specific (JBrowse) link)

To open a view like this



So what does the “portal” look like (in my head)?

- An index.html page that has a little information and six image maps
- A JBrowse instance that has 21 assemblies, each with:
  - a set of gene annotations
  - four minimap2 outputs (to the ones before and after in the series, and to the main references)
  - (Possibly–still trying to decide if useful) BAM alignments from the approx 250x coverage
- Also in JBrowse: the main reference sequences (N2 and VC2010) with VCFs for each of the strain assemblies and (why not) all of the data that they have at WormBase.

Wrap up/Demos

I can demo most things–I'll start with JBrowse 2, but please speak up if there something else you'd like to see.

Lots o'worms: https://riccio.d2jjb0xowet5mr.amplifyapp.com/multiple_alignment.html

Structural variation: https://jbrowse.org/code/jb2/v2.16.0/?config=test_data%2Fconfig_demo.json&session=share-pjaAq1hNxB&password=Z9teR

Alliance variants: https://www.alliancegenome.org/jbrowse2/?session=share-QHEAVYBr_c&password=wTry6

Interactive (play along/if you want/have the energy): http://jbrowse.programmingforbiology.org/

Acknowledgements

People who contributed slides to make my life easier:

- Lacey Ann Sanderson (Tripal)
- Carson Holt (MAKER)
- Beatriz Serrano-Solano (Galaxy)

The literally hundreds (thousands?) of people in the GMOD community who have contributed in so many roles (developers, curators, researchers, documenters, etc).

Finally, Lincoln Stein (co-originator of this course) for supporting me and the GMOD project even when direct funding ended.

Resources

http://gmod.org/

http://jbrowse.org/

http://tripal.info/

http://usegalaxy.org/