WashU Epigenome Browser Documentation

Daofeng Li, Silas Hsu, Renee L. Sears and Ting Wang

Oct 09, 2023

CONTENTS:

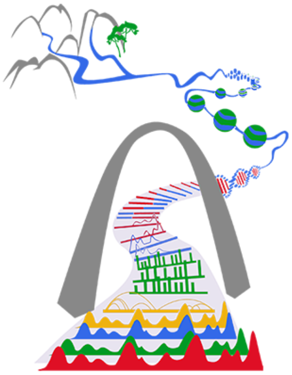


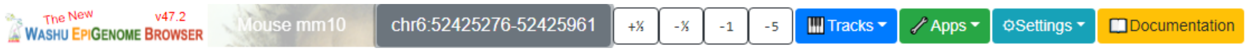
Fig. 1: Gateway to the epigenome. (Art by **Ting Wang**)

CHAPTER

ONE

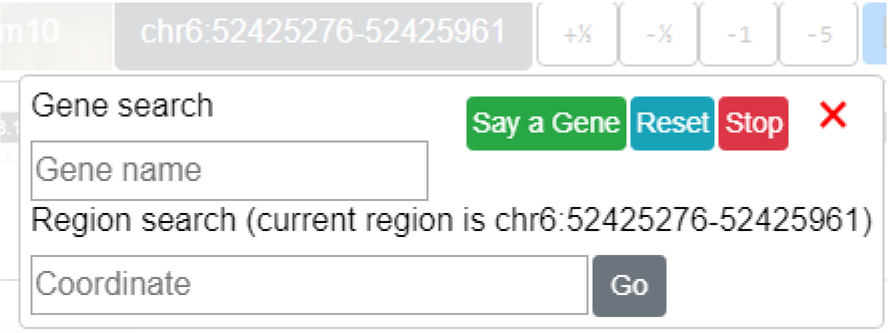
USE THE BROWSER

1.1 The Top Menu



The top navigation menu (above) controls most of the browser functionality. From left to right are the browser logo with version information, species and assembly information, genomic region locator, zoom in/out tools, Tracks menu, Apps menu, Settings menu and Documentaion link.

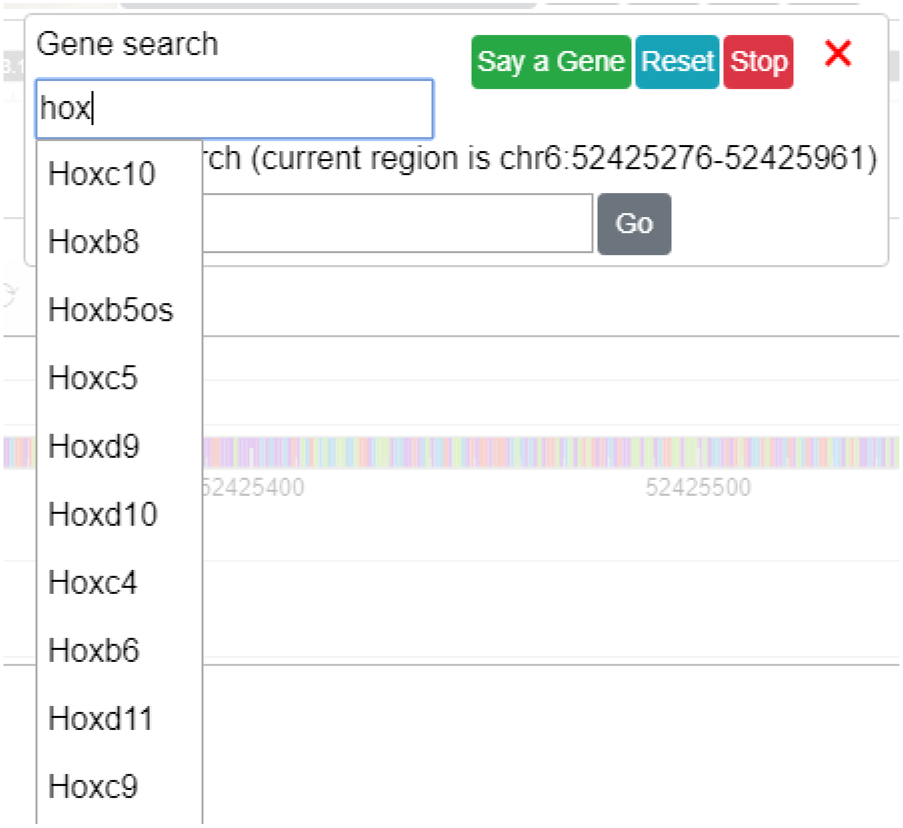
1.2 Genomic Region Locator



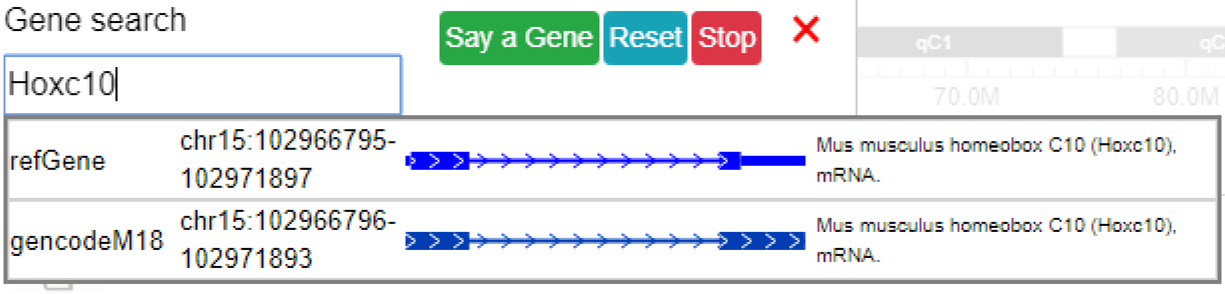
The genomic region locator allows the user to navigate to a region or gene.

1.2.1 Gene Search

You can type a gene name/symbol, like `Hox`, and when the input content reaches 3 characters the browser will try to find gene symbols starting with what you typed:



After a gene is selected a dropdown menu will pop up with isoforms for the gene. After clicking an isoform the browser will navigate you to its genomic region.



refGene	chr15:102966795-102971897	Mus musculus homeobox C10 (Hoxc10), mRNA.
gencodeM18	chr15:102966796-102971893	Mus musculus homeobox C10 (Hoxc10), mRNA.

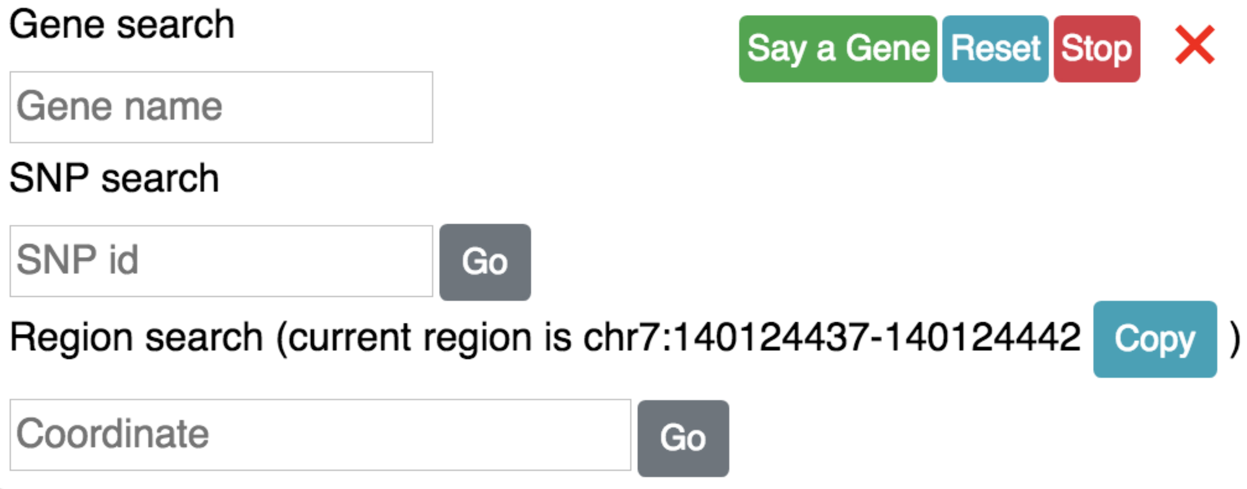
Voice input gene symbol

From this set of buttons, , click the **Say a Gene** button, your web browser will ask you for permission to access your microphone devices, choose *Allow*, and the browser will start to listen to what you are saying. You can start saying letters one by one, like H, O, X, if you click the red **stop** button, what you said "HOX" will populate the gene search box and suggested gene symbols will pop up. As before you can then choose the gene and isoform you want to navigate to.

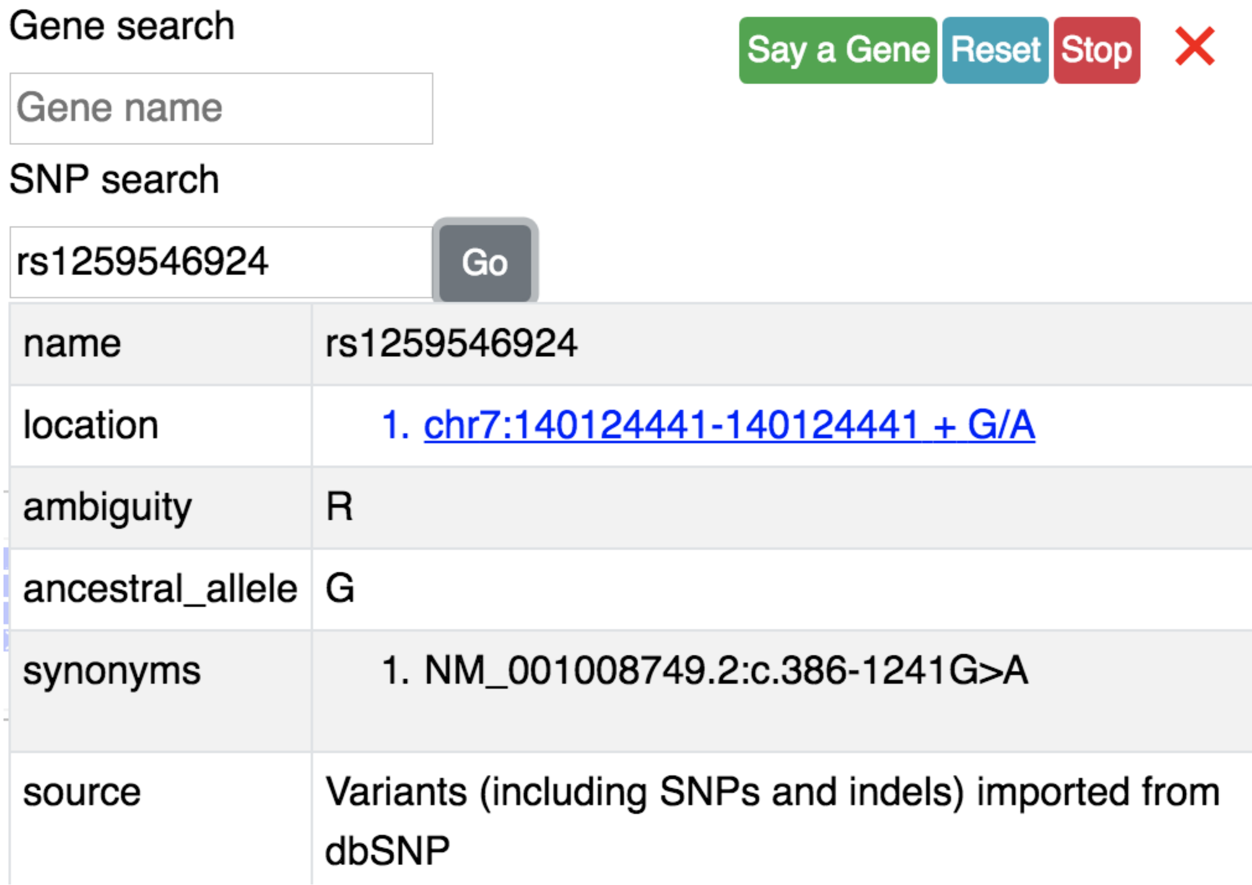
Note: This feature is dependent on web browser support. A web browser without support for speech recognition won't see this UI.

1.2.2 SNP search

SNP search is also avaiable from the genomic region locator button:



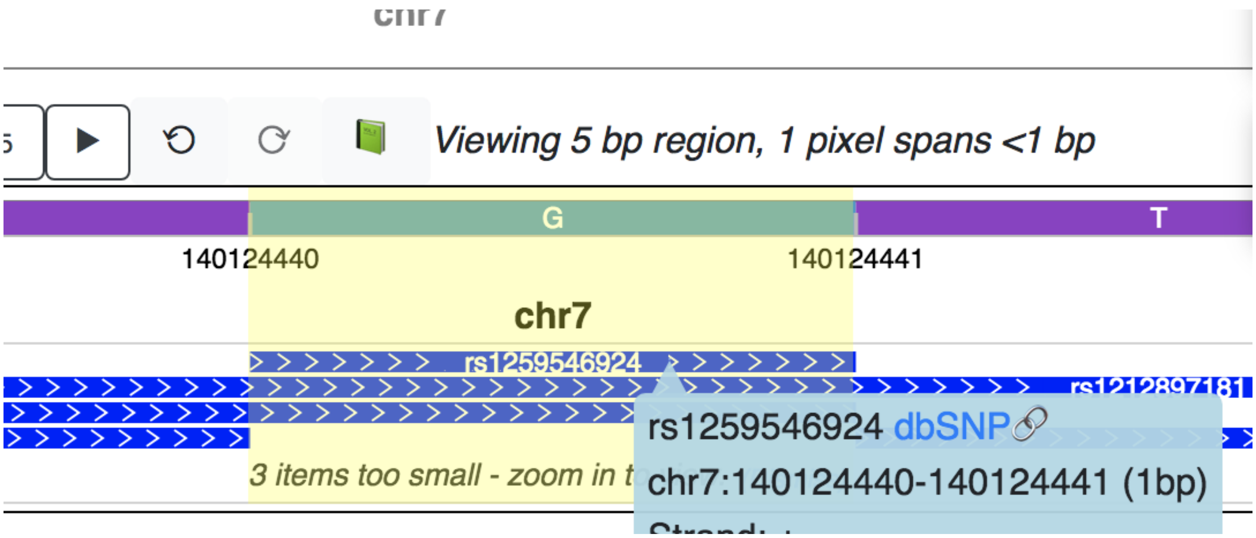
say you input a SNP id: rs1259546924, click the Go button, you will get information about this SNP:



rs1259546924

name	rs1259546924
location	1. chr7:140124441-140124441 + G/A
ambiguity	R
ancestral_allele	G
synonyms	1. NM_001008749.2:c.386-1241G>A
source	Variants (including SNPs and indels) imported from dbSNP

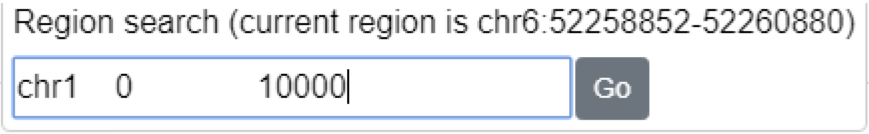
Click the blue postion link can navigate you to this SNP's position:


chr7
Viewing 5 bp region, 1 pixel spans <1 bp
G
T
140124440
140124441
chr7
rs1259546924
rs1212897181
3 items too small - zoom in t
rs1259546924 dbSNP
chr7:140124440-140124441 (1bp)
Strand: +

Note: SNP search uses the Ensembl API services: https://rest.ensembl.org

1.2.3 Region Search

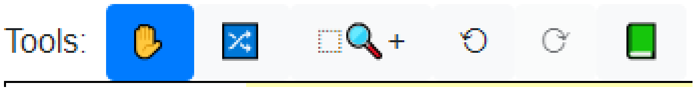
Below the gene search box you can use the region search box to navigate to specific genomic coordinates. Formats such as `chr6:52258852-52260880` or `chr6 52258852 52260880` are accepted (the browser is not sensitive to the number of spaces or tabs between the *chr*, *start*, and *stop*.



Region search (current region is chr6:52258852-52260880)

chr1   0   10000

1.3 Operations in the Tracks View Container

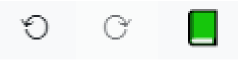


Right above the tracks on the left hand side there are a bunch of tools for operating on the tracks. From left to right these include the move, re-order, zoom, undo, redo, and history tools.

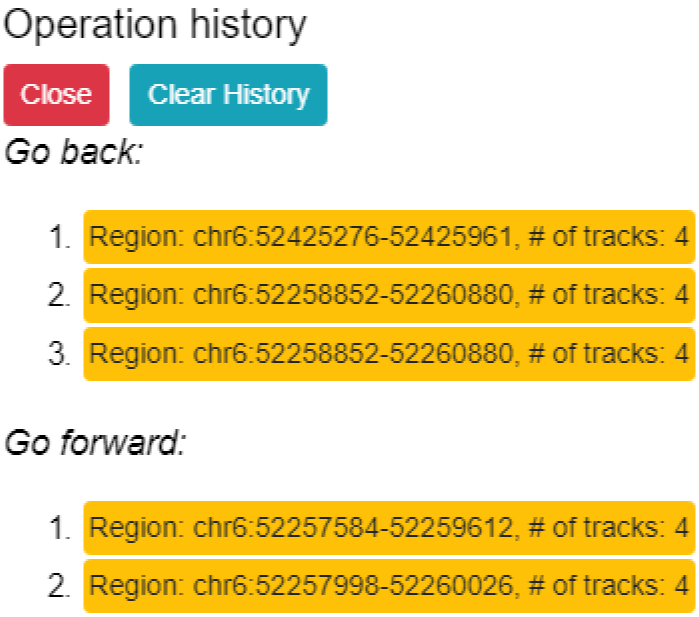
1.3.1 Move/Re-order/Zoom tools

The ✋ is the default tool selected by the browser. `Moving` mode allows the user to drag the mouse right or left and the tracks will move along mouse moving to new regions. The ⤨ allows tracks to be `re-ordered`. The user can drag one or more tracks up or down to change the order of tracks. The ⬚🔍+ allows the user to `zoom in` on a specific region within the current view using the mouse.

1.3.2 Undo/Redo/History tools



contains the `Undo`, `Redo`, and `History` tools. For instance if you accidently moved to another region and forgot what you were looking at before you can click the `Undo` button to go back. Clicking the `Redo` button allows you to go forward to the step before you clicked `Undo`. The `History` button gives you the 9 most recent operations and allows you to jump to any of these operations or clear the history.



Operation history

Go back:

1. Region: chr6:52425276-52425961, # of tracks: 4
2. Region: chr6:52258852-52260880, # of tracks: 4
3. Region: chr6:52258852-52260880, # of tracks: 4

Go forward:

1. Region: chr6:52257584-52259612, # of tracks: 4
2. Region: chr6:52257998-52260026, # of tracks: 4

1.3.3 Hotkeys

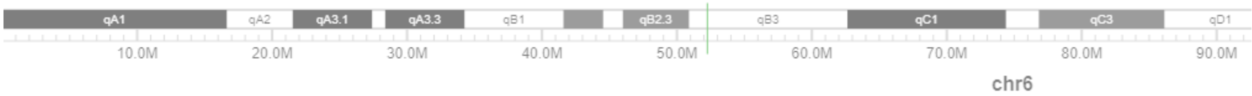
1. `Alt + H` or `Alt + D` for the Drag Tool
2. `Alt + S` or `Alt + R` for the Reorder/Swap Tool
3. `Alt + M` for the Magnify Tool
4. `Alt + Z` and `Alt + X` to pan one full panel left or right.

1.4 Settings

The `Settings` menu controls global settings for the browser.

1.4.1 Toggle display of the Genome Navigator

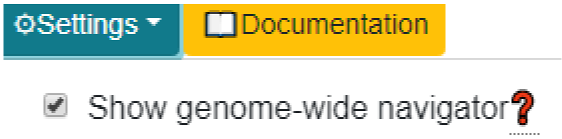
By using the genome navigator (below) users can jump to any genomic region or chromosome(s).



The operations on the genome navigator are:

- Left mouse drag: select
- Right mouse drag: pan
- Mousewheel: zoom

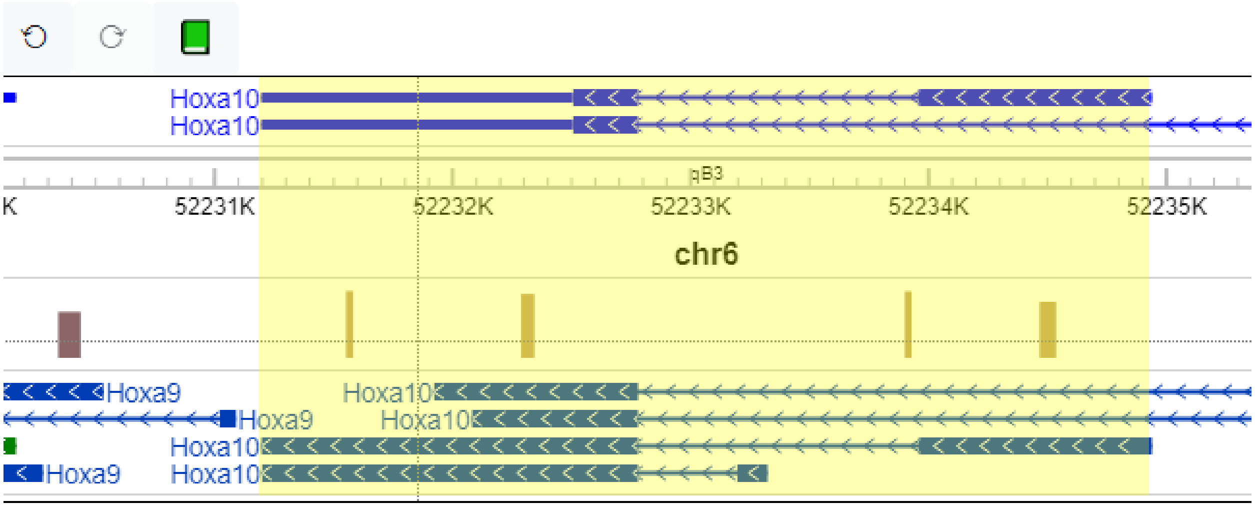
The genome navigator can also be hidden to save space when viewing tracks. Click `Settings` on the top menu and uncheck the box to switch off this feature:



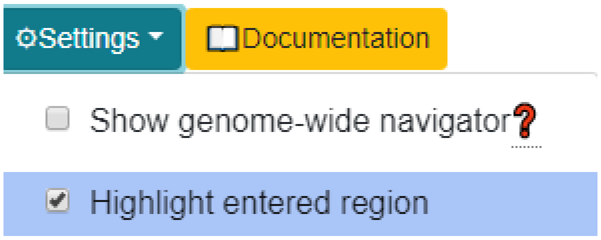
☑ Show genome-wide navigator?

1.4.2 Toggle highlighting of enter region

When a user jumps to a region or gene using the Genomic Region Locator, that region or gene is highlighted with a light yellow box.



This highlighting can be turned off/on by clicking the botton on the `Settings` menu:



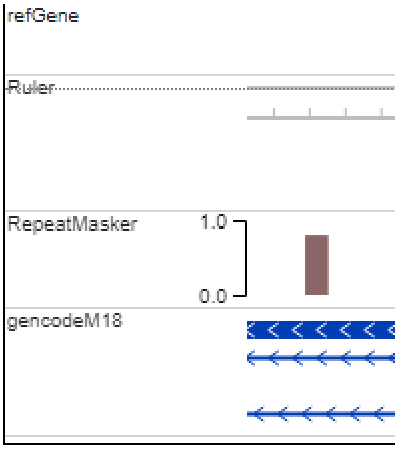
⚙Settings ▾ | Documentation

☐ Show genome-wide navigator?

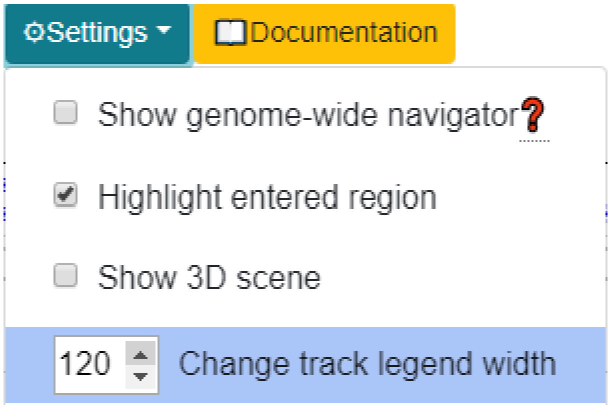
☑ Highlight entered region

1.4.3 Change track label width

The default width of track labels (below) is 120 pixels.



The width of the track label can be configured by the submenu under the `Settings` menu:



⚙Settings ▾ | Documentation

☐ Show genome-wide navigator?

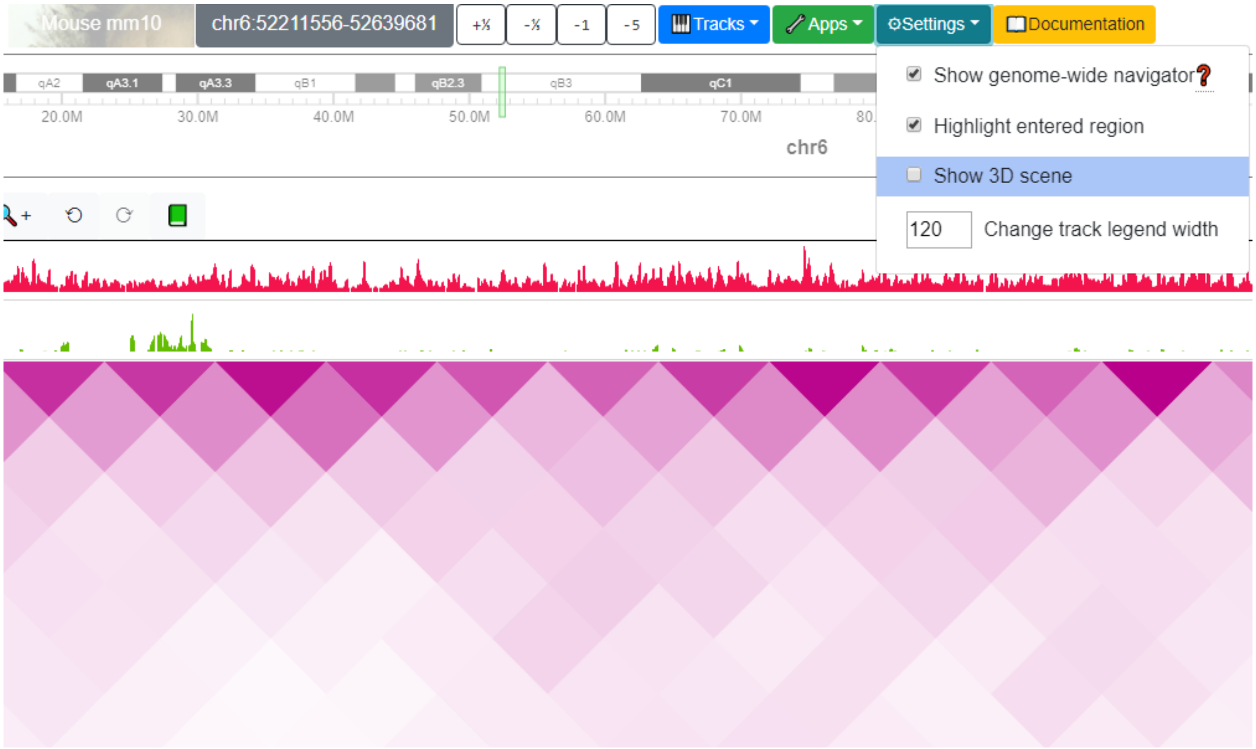
☑ Highlight entered region

☐ Show 3D scene

120 Change track legend width

1.4.4 Toggle display of VR mode

From the `Settings` menu the user can choose to toggle the VR display mode of tracks:



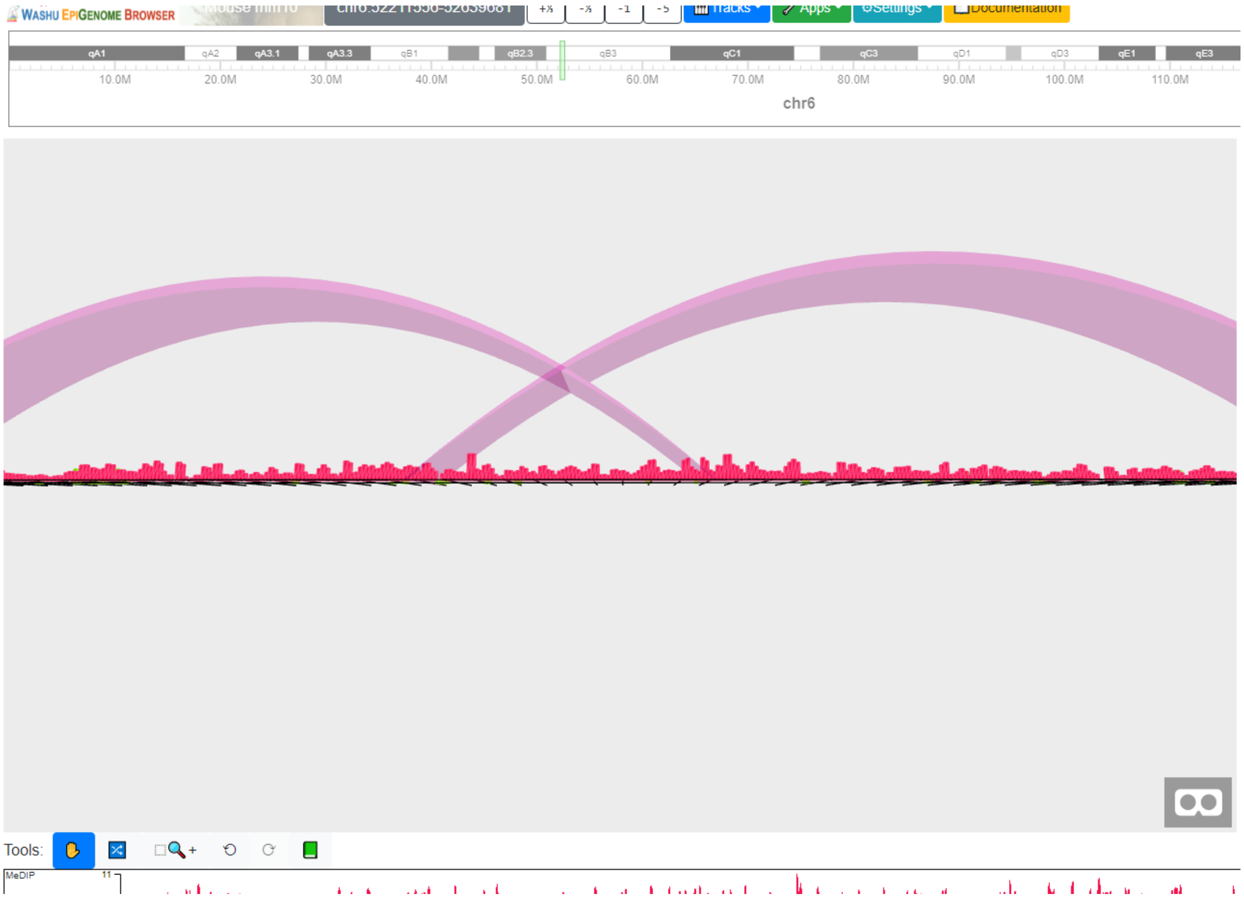
☑ Show genome-wide navigator❓

☑ Highlight entered region

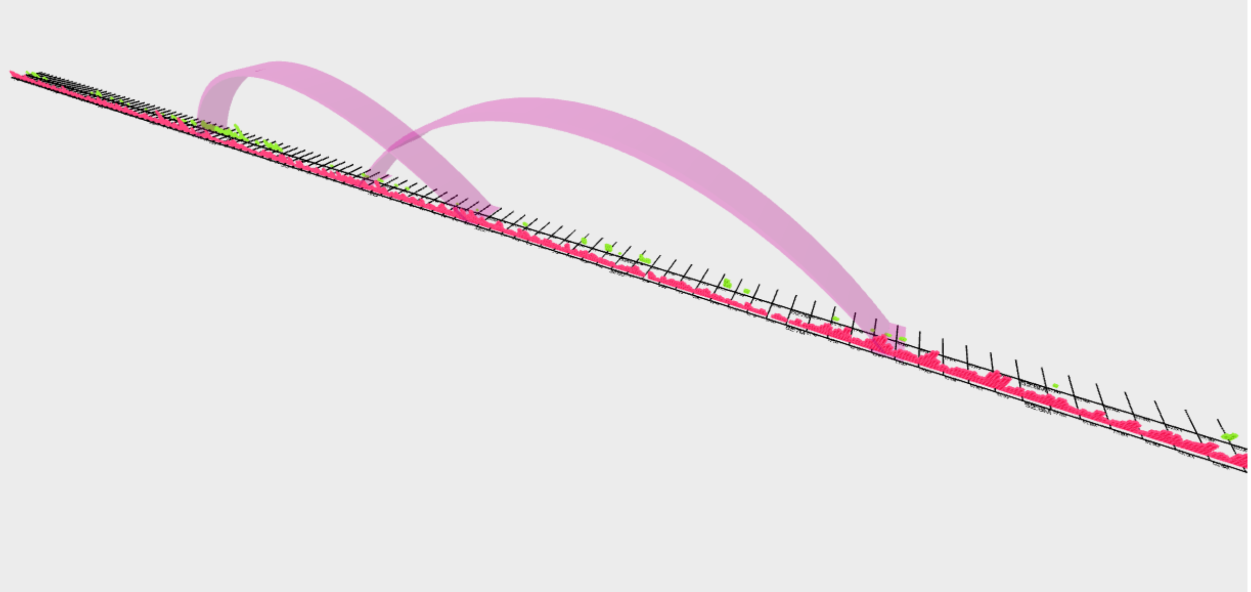
□ Show 3D scene

120 Change track legend width

After choose the **Show 3D scene** submenu, a new container with VR view of the tracks will appear:



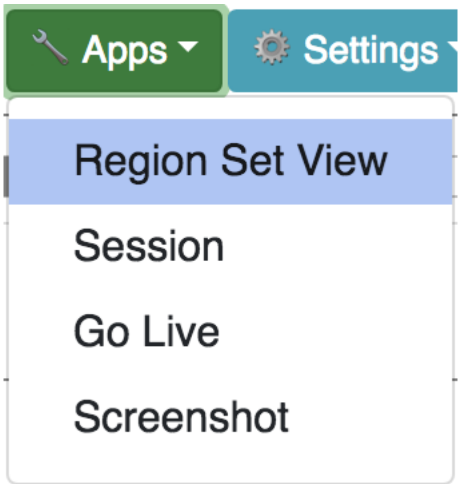
You can click the icon at the bottom right to toggle the full screen display of VR mode, then you can use your mouse and keys `W`, `A`, `S` and `D` to control the view of VR mode, like this view below can show you the interaction between two genomic loci and methylation status along this region in a 3D way.



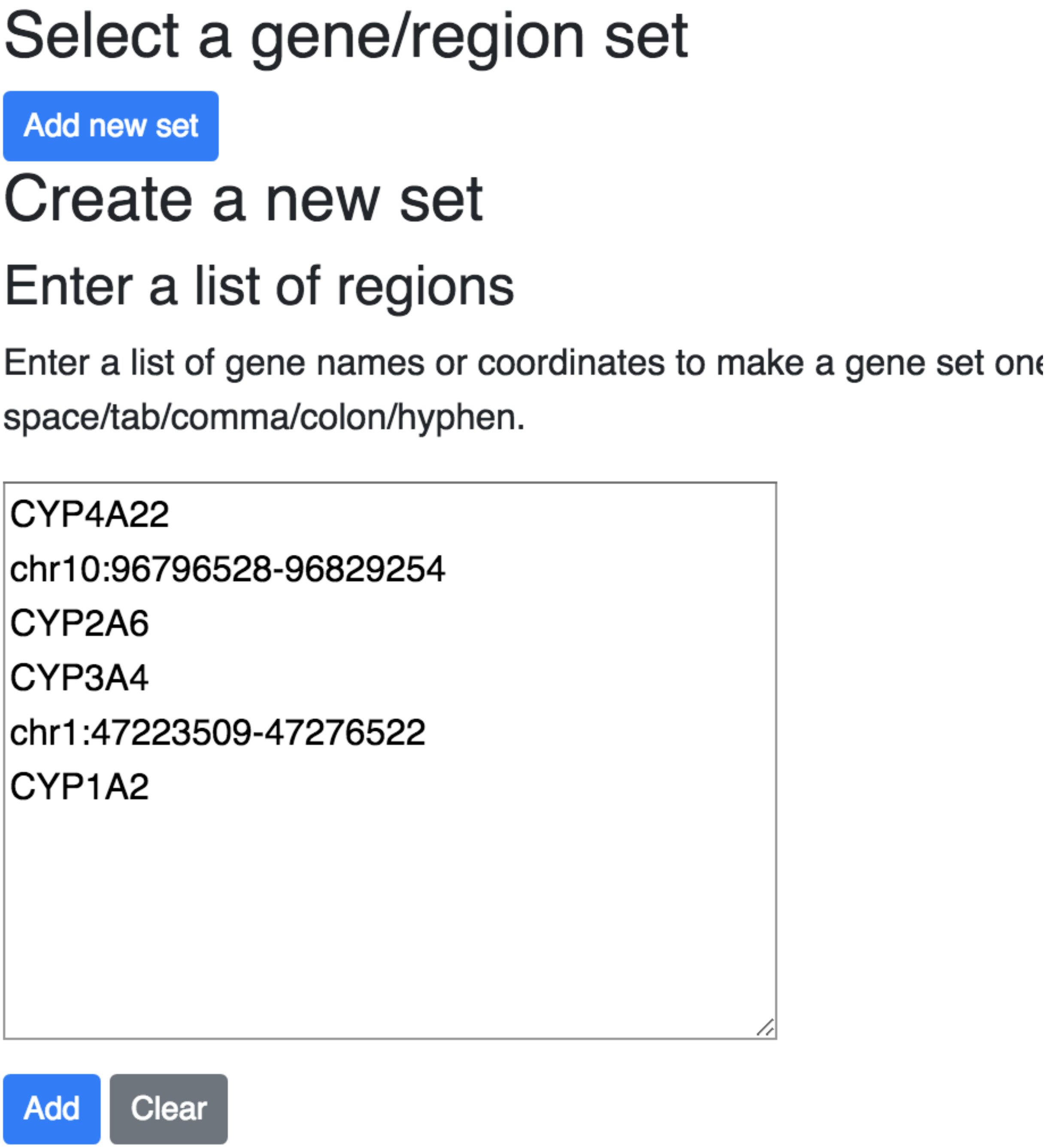
1.5 Apps

1.5.1 Region set view

Users can submit a list of regions or genes to the browser, by choose `Apps -> Region set view`:



The brings up the region set user interface, here you can enter a list of gene names or coordinates to make a gene set one item per line. Gene names and coordinates can be mixed for input. Coordinate string must be in the form of "chr1:345-678" fields can be joined by space/tab/comma/colon/hyphen.



Select a gene/region set

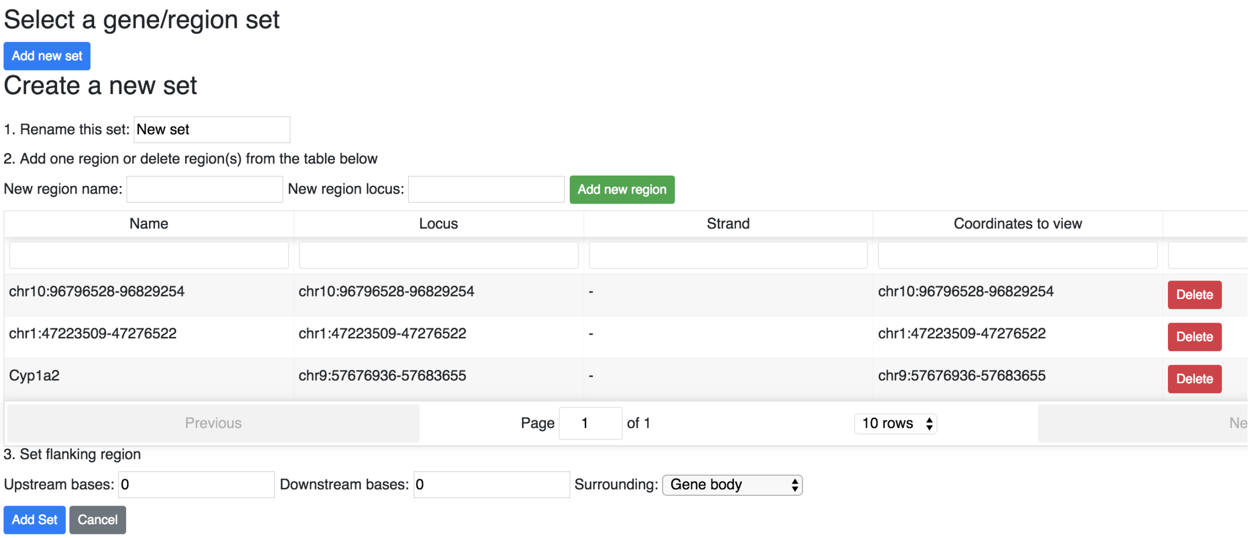
Create a new set

Enter a list of regions

Enter a list of gene names or coordinates to make a gene set one
space/tab/comma/colon/hyphen.

```
CYP4A22
chr10:96796528-96829254
CYP2A6
CYP3A4
chr1:47223509-47276522
CYP1A2
```

After Click the *Add* button, will bring you to the region set editting interface, you can either add region one by one, or delete regions from the table, and set the flanking region strategy:



Select a gene/region set

Create a new set

1. Rename this set: New set

2. Add one region or delete region(s) from the table below

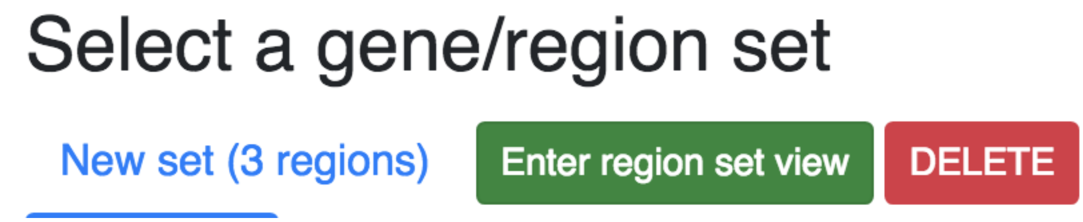
New region name: New region locus:

Name	Locus	Strand	Coordinates to view	
chr10:96796528-96829254	chr10:96796528-96829254	-	chr10:96796528-96829254	Delete
chr1:47223509-47276522	chr1:47223509-47276522	-	chr1:47223509-47276522	Delete
Cyp1a2	chr9:57676936-57683655	-	chr9:57676936-57683655	Delete

3. Set flanking region

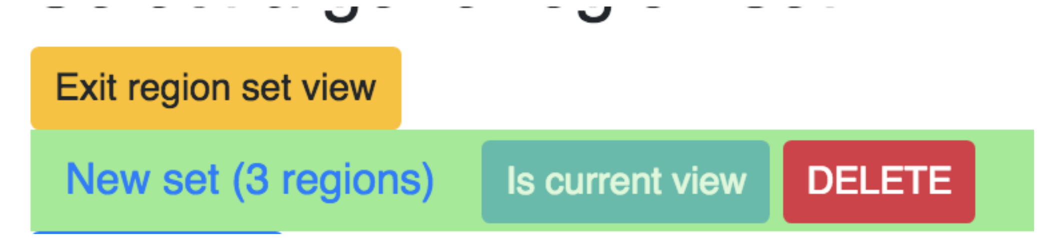
Upstream bases: 0 Downstream bases: 0 Surrounding: Gene body

Once you done with edit the set, cick the button *Add set*. Now you have the option to enter regin set view, click the button *Enter region set view*:

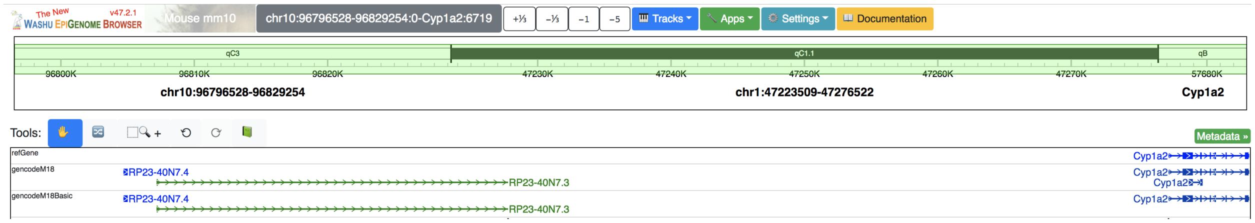


Select a gene/region set

This indicates you are in *region set view* mode and which set you are viewing:

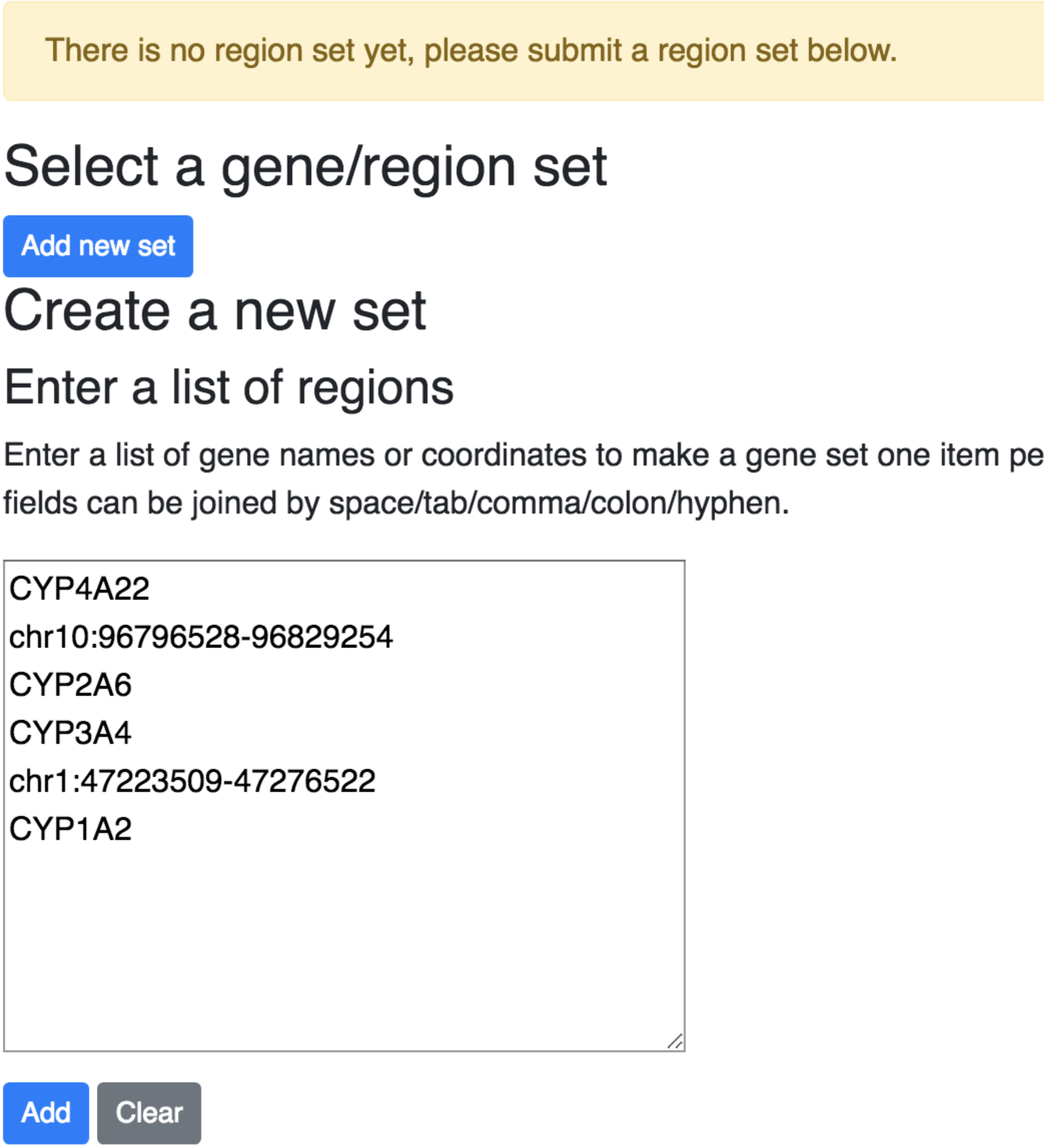


Go back to the browser, you can your browser view is ordered by your region set:



1.5.2 Geneplot

Geneplot function allows users to see overall signal of a numerical track over user selected gene/region sets. Choose `Geneplot` from the `Apps` menu, if there is no region sets added before, the browser will bring the region set adding interface:



There is no region set yet, please submit a region set below.

Select a gene/region set

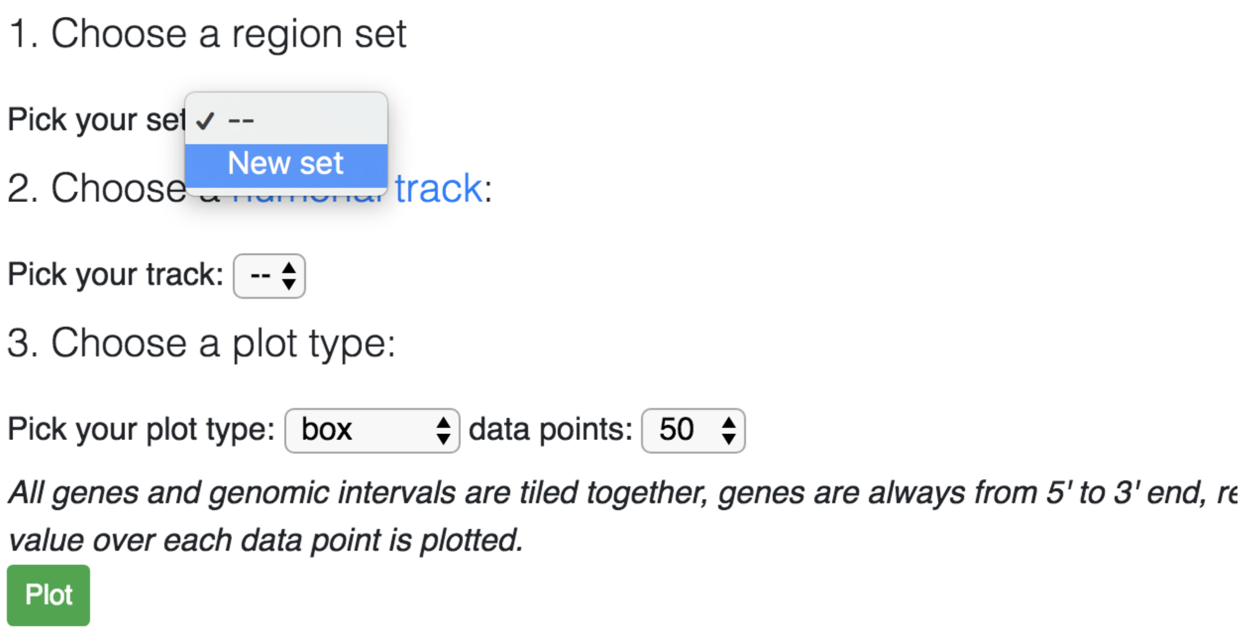
Create a new set

Enter a list of regions

Enter a list of gene names or coordinates to make a gene set one item pe fields can be joined by space/tab/comma/colon/hyphen.

```
CYP4A22
chr10:96796528-96829254
CYP2A6
CYP3A4
chr1:47223509-47276522
CYP1A2
```

After adding a region set, you can choose the available set from the dropdown in first step:



1. Choose a region set

Pick your set ✓ --

New set

2. Choose track:

Pick your track: --

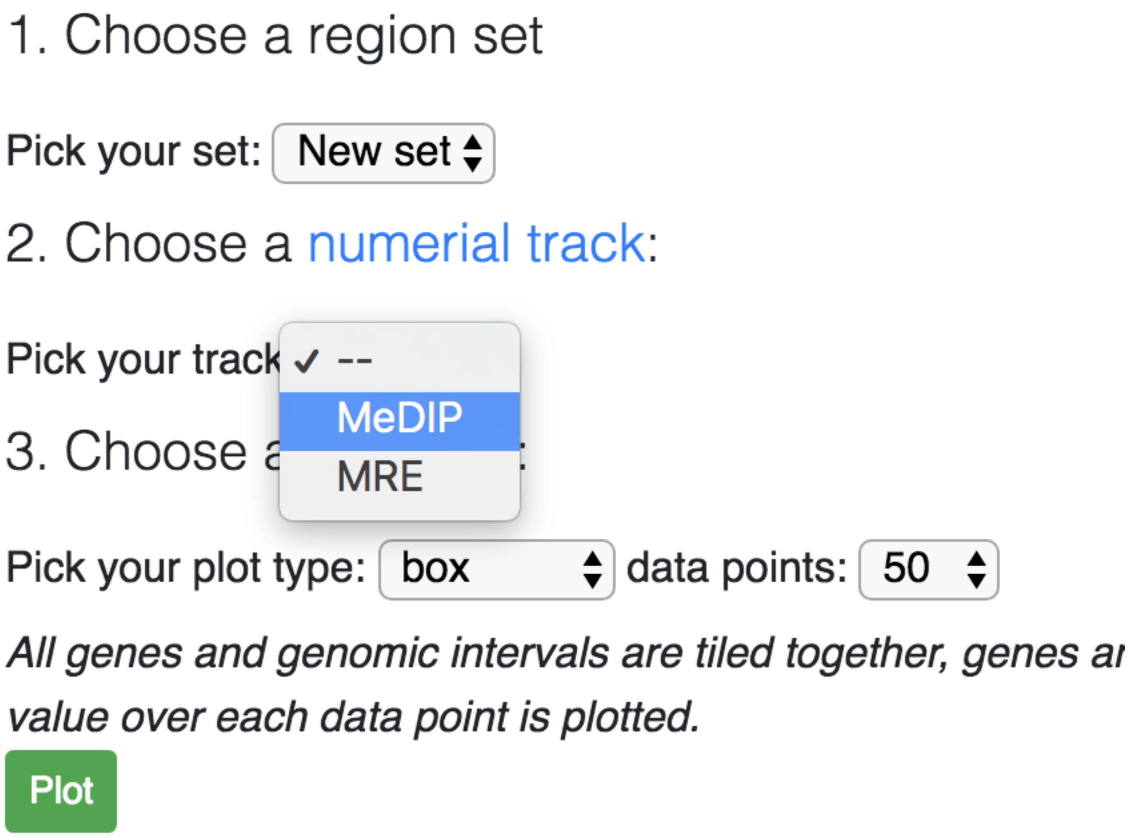
3. Choose a plot type:

Pick your plot type: box data points: 50

All genes and genomic intervals are tiled together, genes are always from 5' to 3' end, re
value over each data point is plotted.

Plot

Now you need to choose a numerical track, you can use your custom track or publicly avaiable tracks:



1. Choose a region set

Pick your set: New set

2. Choose a numerial track:

Pick your track ✓ --

MeDIP

MRE

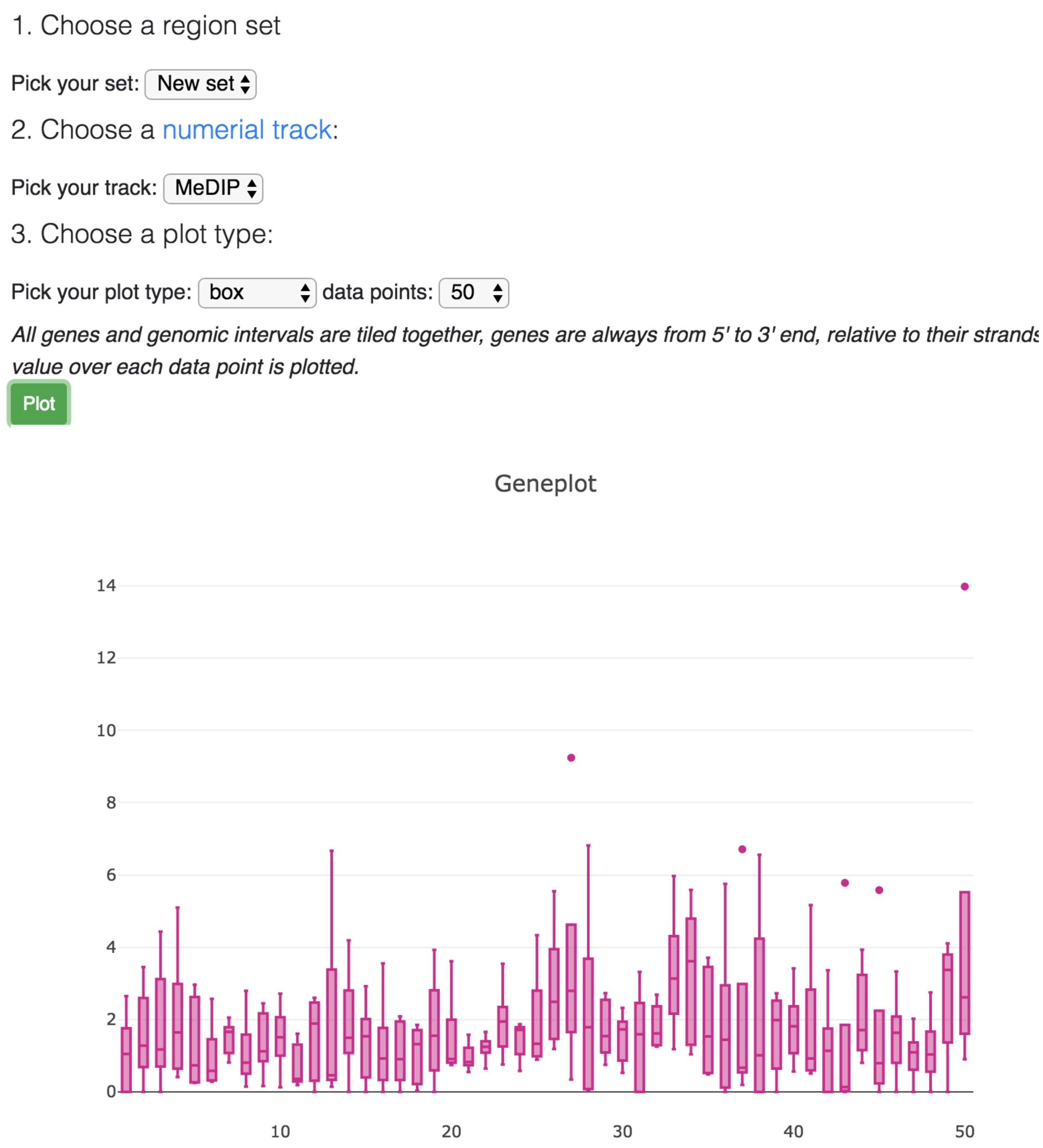
3. Choose a

Pick your plot type: box data points: 50

All genes and genomic intervals are tiled together, genes ar
value over each data point is plotted.

Plot

After choose a numerical track, click the **Plot** button, this will generate the boxplot by default:



1. Choose a region set

Pick your set: New set

2. Choose a numerial track:

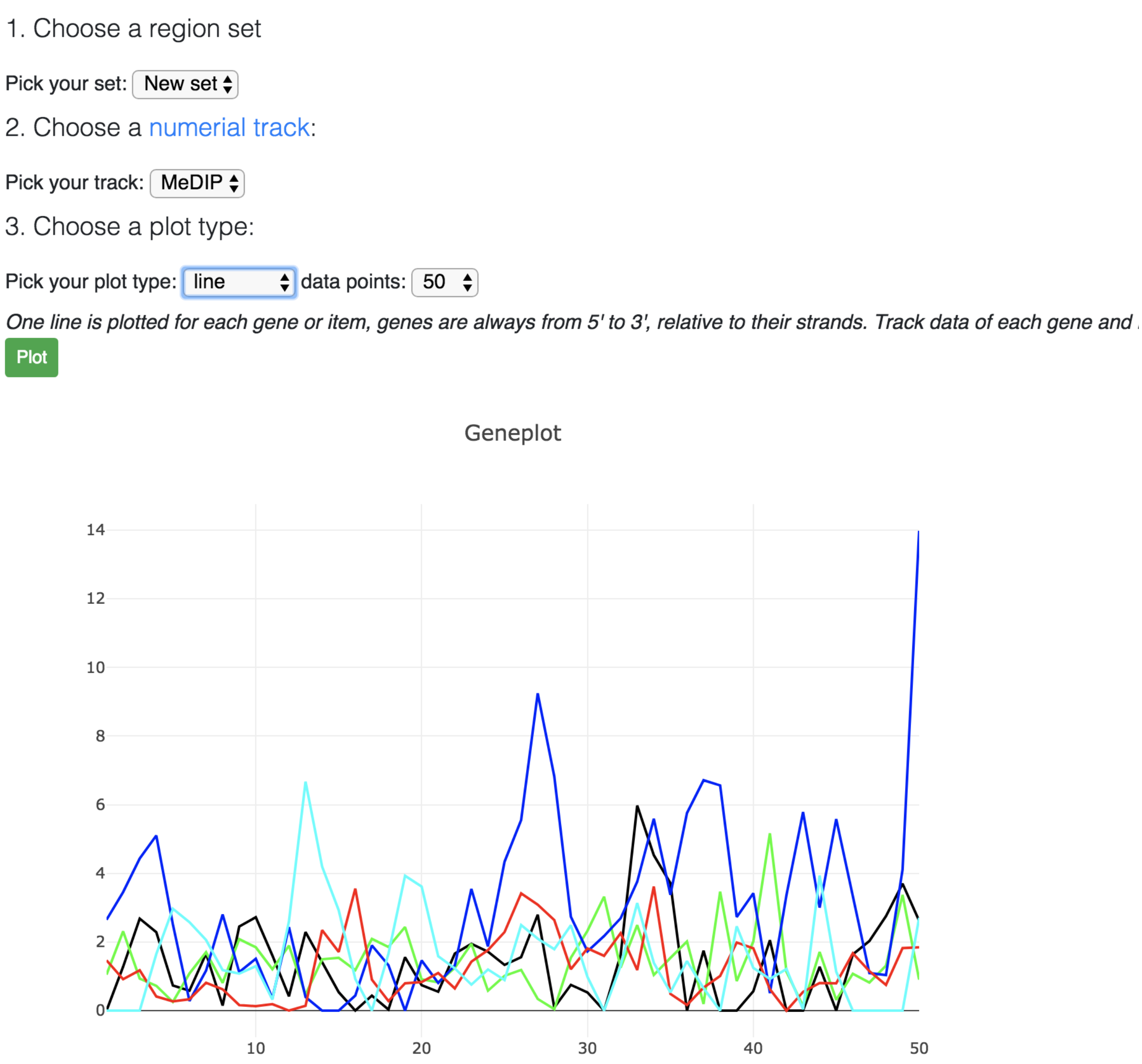
Pick your track: MeDIP

3. Choose a plot type:

Pick your plot type: box data points: 50

All genes and genomic intervals are tiled together, genes are always from 5' to 3' end, relative to their strands
value over each data point is plotted.

Choose line plot:



1. Choose a region set

Pick your set: New set

2. Choose a numerial track:

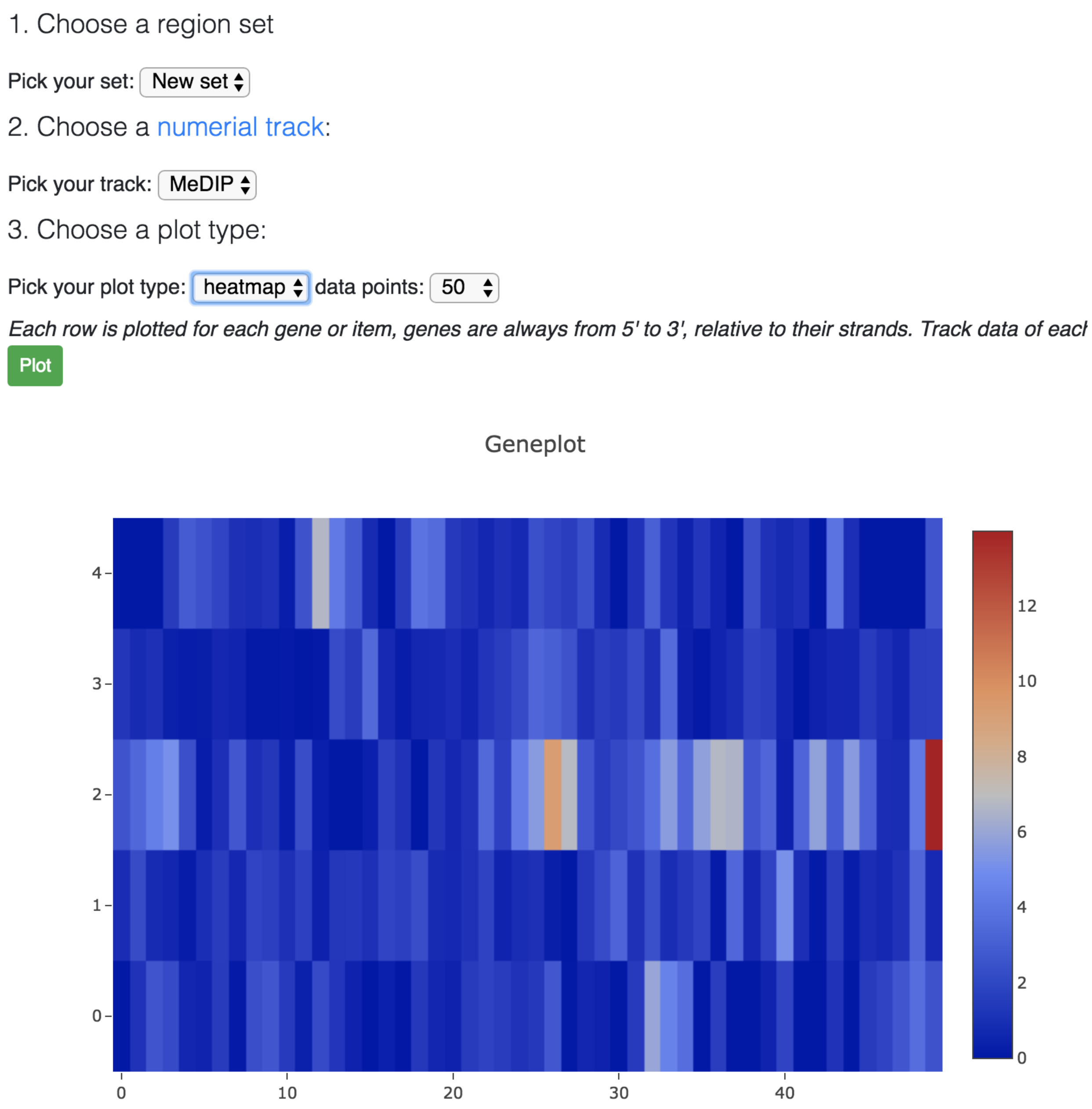
Pick your track: MeDIP

3. Choose a plot type:

Pick your plot type: line data points: 50

One line is plotted for each gene or item, genes are always from 5' to 3', relative to their strands. Track data of each gene and

Choose heatmap:



1. Choose a region set

Pick your set: New set

2. Choose a numerial track:

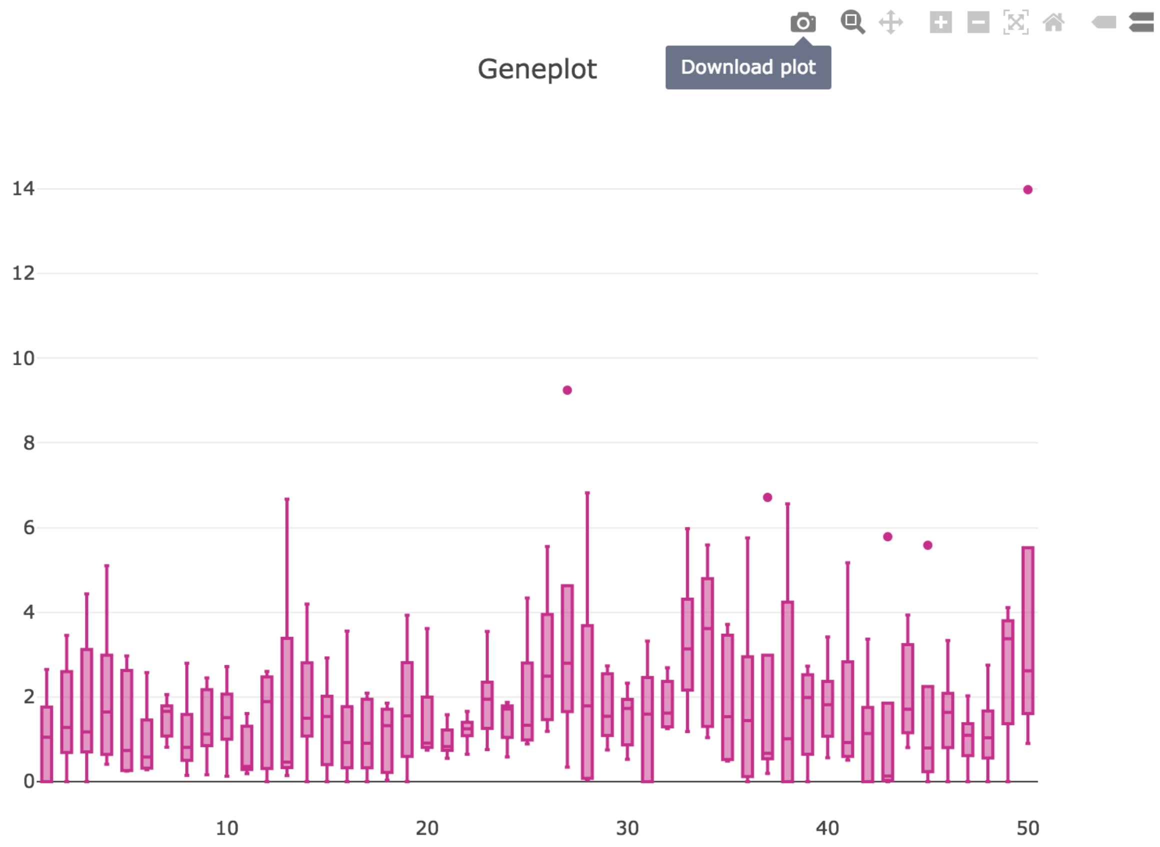
Pick your track: MeDIP

3. Choose a plot type:

Pick your plot type: heatmap data points: 50

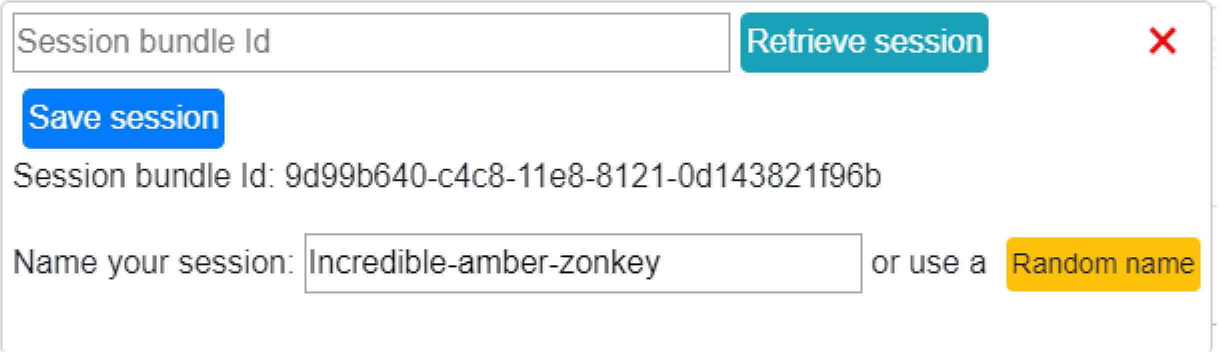
Each row is plotted for each gene or item, genes are always from 5' to 3', relative to their strands. Track data of eacl

When you mouse over the plot, there is a button for you to download the plot as SVG file:



1.5.3 Session

Choosing `Session` from the `Apps` menu will bring you to the session interface shown below:

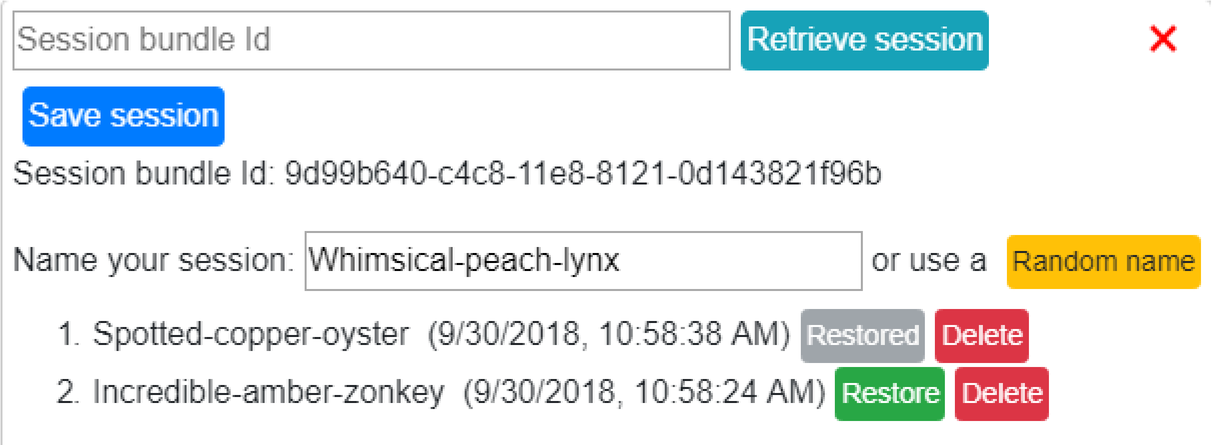


Session bundle Id: 9d99b640-c4c8-11e8-8121-0d143821f96b

Name your session: Incredible-amber-zonkey or use a Random name

Save session

Click the **Save session** button to save a session. A session bundle Id will be created which allows the user to retrieve their session at a later date.



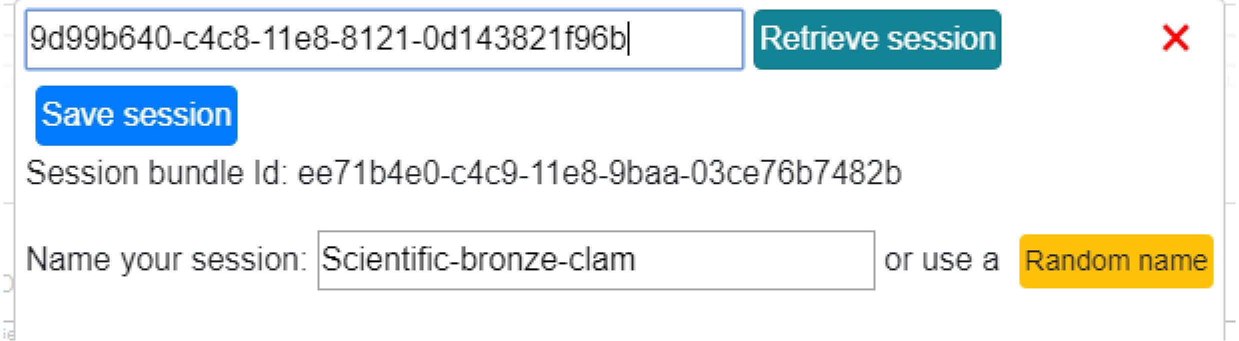
Session bundle Id: 9d99b640-c4c8-11e8-8121-0d143821f96b

Name your session: Whimsical-peach-lynx or use a

1. Spotted-copper-oyster (9/30/2018, 10:58:38 AM)
2. Incredible-amber-zonkey (9/30/2018, 10:58:24 AM)

Retrieve session

The **session bundle Id** can be used later to retrieve a session by pasting the session bundle id in the session interface and clicking the `Retrieve session` button.

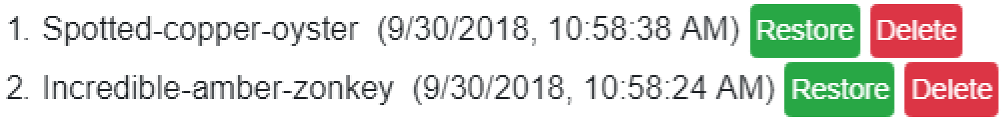


9d99b640-c4c8-11e8-8121-0d143821f96b

Session bundle Id: ee71b4e0-c4c9-11e8-9baa-03ce76b7482b

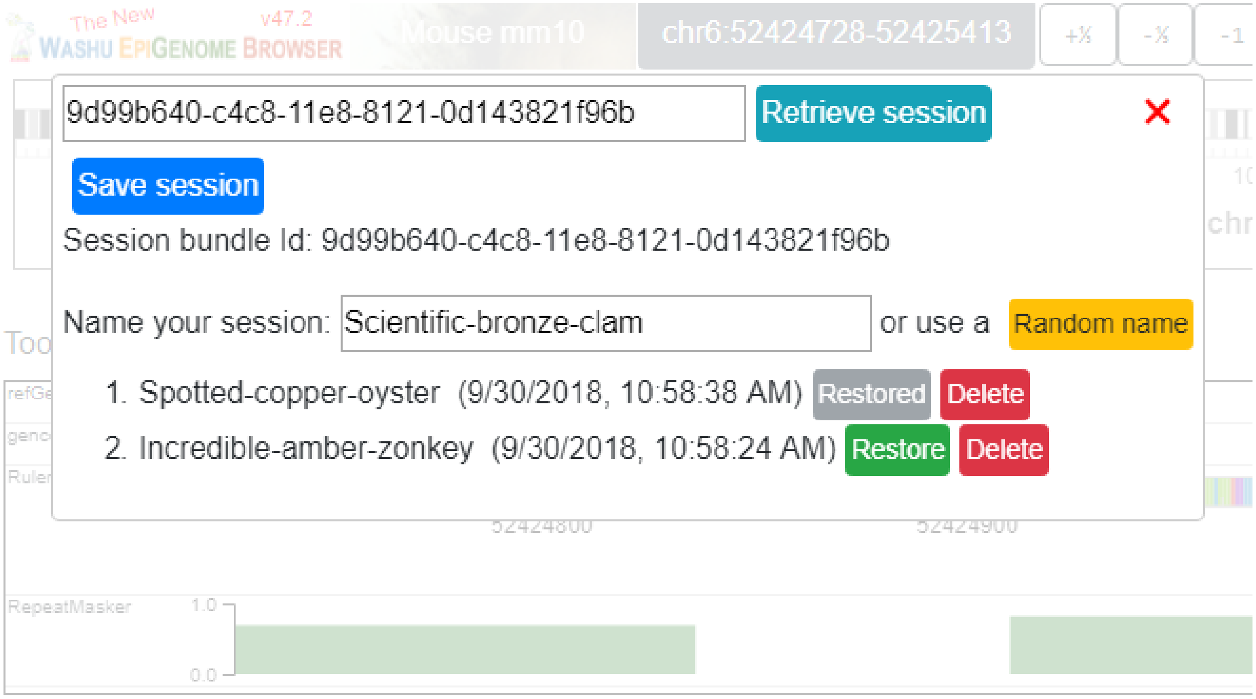
Name your session: Scientific-bronze-clam or use a

Choose which session status you want to restore:



1. Spotted-copper-oyster (9/30/2018, 10:58:38 AM)
2. Incredible-amber-zonkey (9/30/2018, 10:58:24 AM)

Click the green *Restore* button and your session will be restored:



9d99b640-c4c8-11e8-8121-0d143821f96b

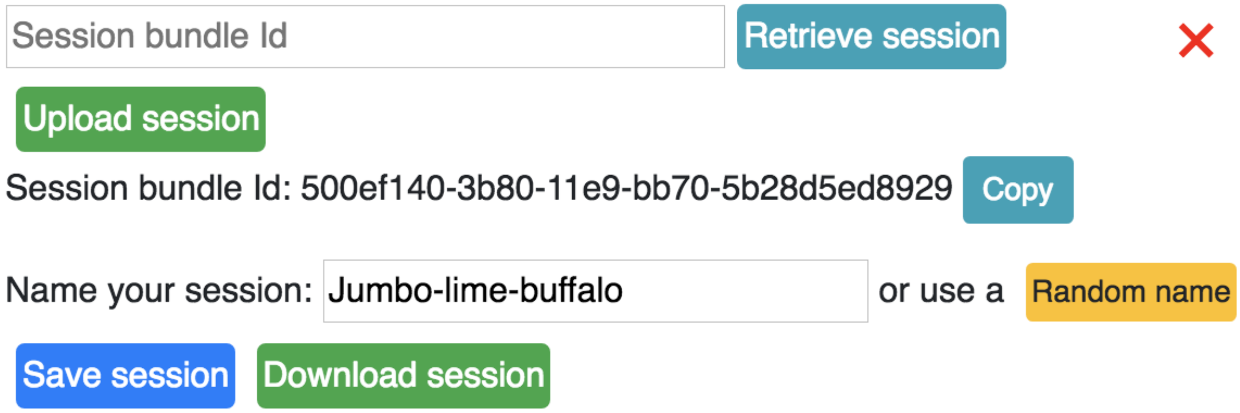
Session bundle Id: 9d99b640-c4c8-11e8-8121-0d143821f96b

Name your session: Scientific-bronze-clam or use a Random name

1. Spotted-copper-oyster (9/30/2018, 10:58:38 AM)
2. Incredible-amber-zonkey (9/30/2018, 10:58:24 AM)

Download and Upload session

Sessions can be downloaded to a json file to your local disk, or can be uploaded from your local drive as well.



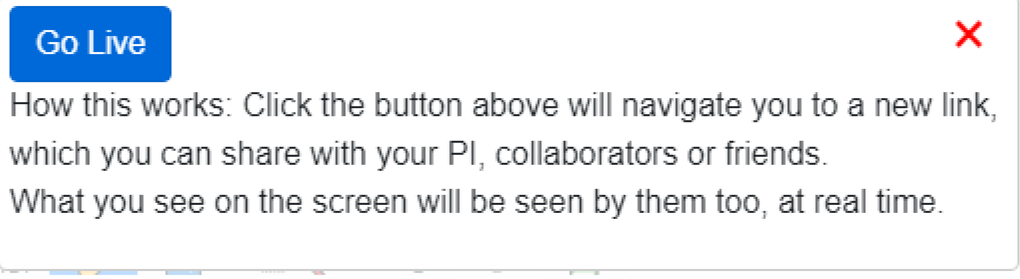
Session bundle Id: 500ef140-3b80-11e9-bb70-5b28d5ed8929

Name your session: Jumbo-lime-buffalo or use a Random name

Note: The downloaded session file can be put in a URL, then use `sessionFile` parameter for fast retrieve the session, like `http://epigenomegateway.wustl.edu/browser/?sessionFile=https://wangftp.wustl.edu/~dli/test/eg-session--1692c5f0-c392-11e9-829c-912864922e1e.json`

1.5.4 Live browsing

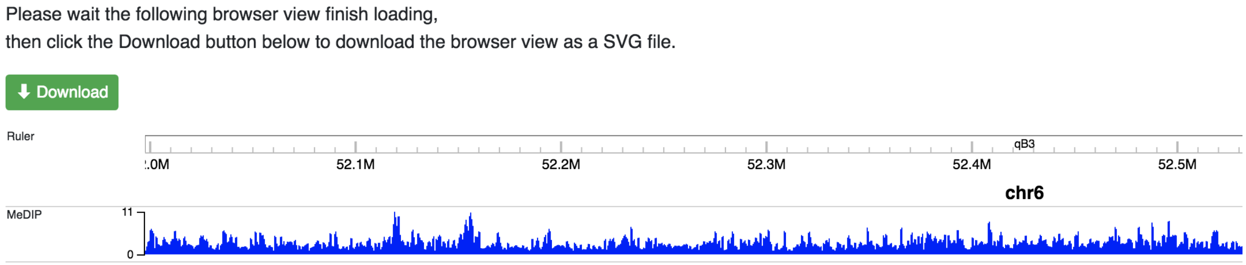
From the Apps menu choose **Go Live**, the browser will navigate you to a new link which you can share with someone else, like your collaborator, your PI, or your friends. Whatever operations are done by you are mirrored on the displays of the people who opened the same link.



How this works: Click the button above will navigate you to a new link, which you can share with your PI, collaborators or friends.
What you see on the screen will be seen by them too, at real time.

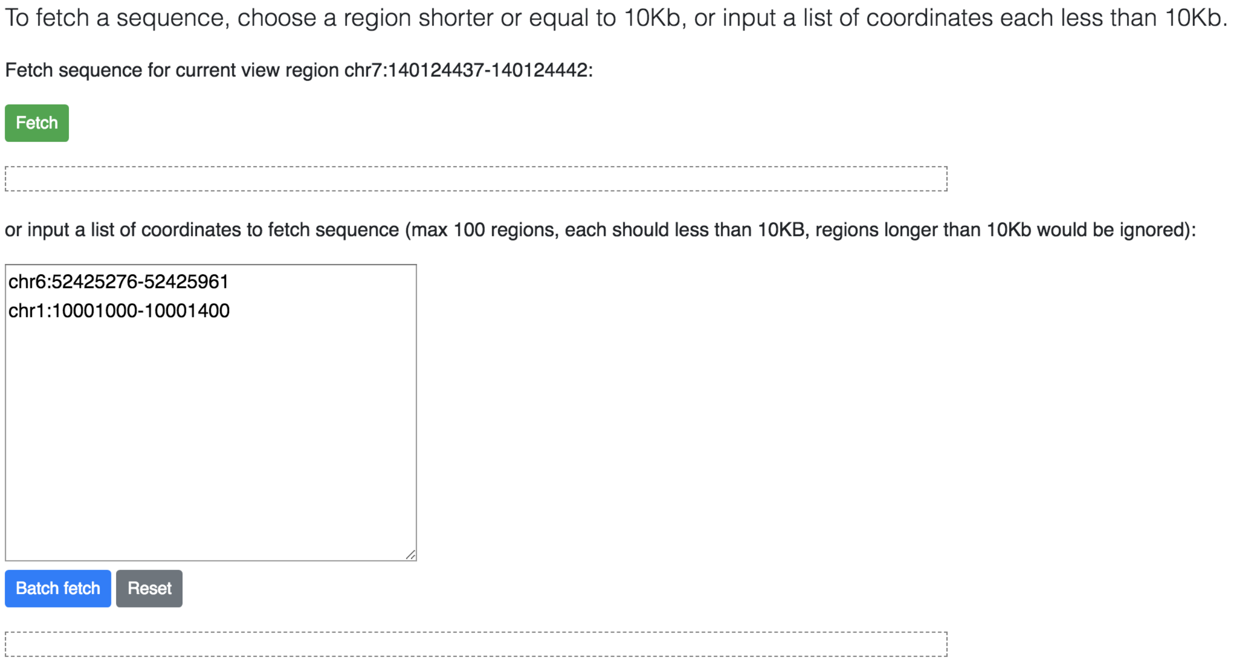
1.5.5 Screenshot

Users can create publication quality images using the *Screenshot* tool from the Apps menu. Click the *Screenshot* button and a new window will po pup that re-renders all your tracks as a new SVG file. Once rendered you can click the green download button to save the current browser view as a SVG image file.


Please wait the following browser view finish loading,
then click the Download button below to download the browser view as a SVG file.

1.5.6 Fetch Sequence

From the Apps menu choose **Fetch Sequence**,, this function allows user to retrieve genomic sequence of current view region, or users can also specified a list of regions to fetch the sequences. Each region should no longer than 10KB.



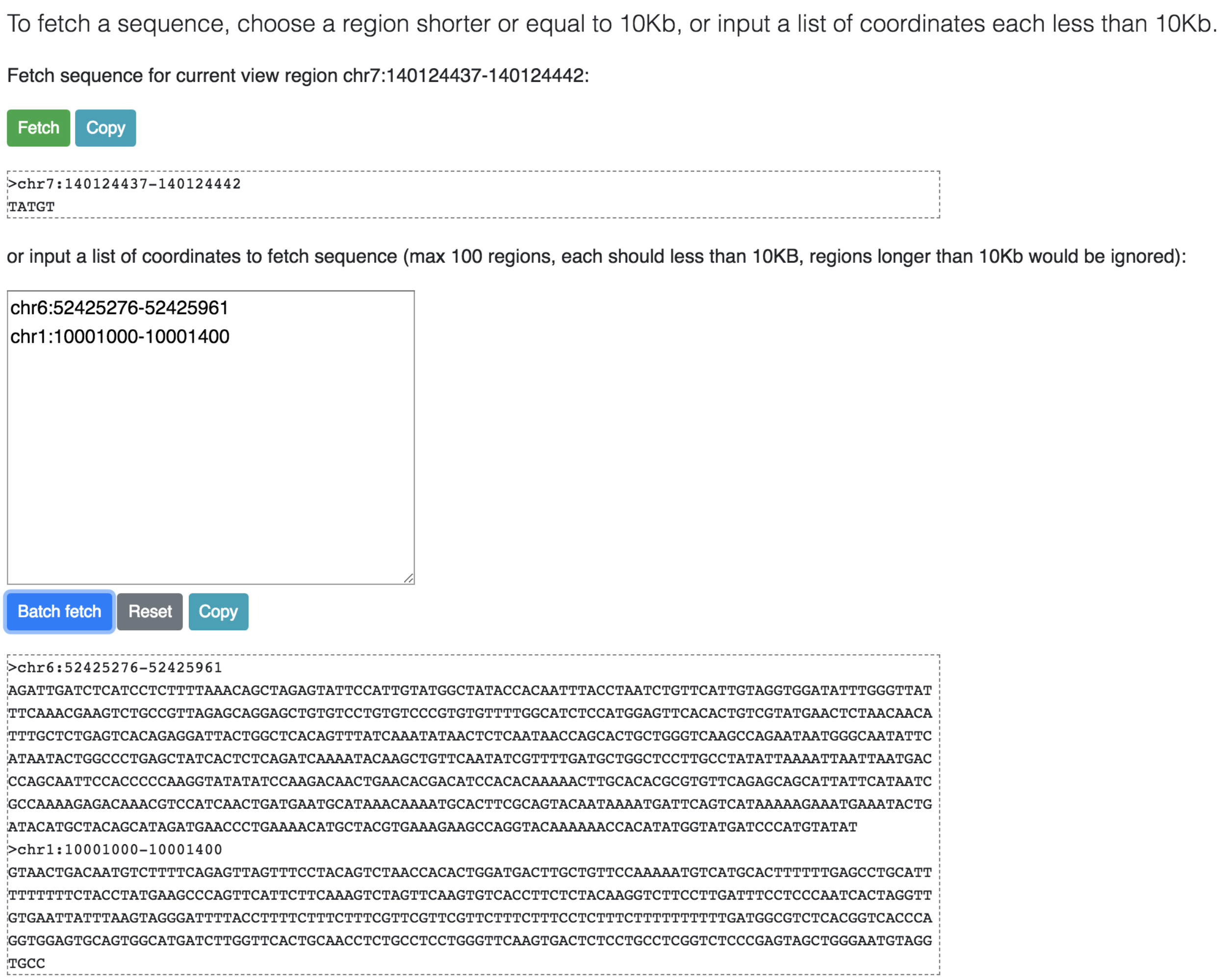
To fetch a sequence, choose a region shorter or equal to 10Kb, or input a list of coordinates each less than 10Kb.

Fetch sequence for current view region chr7:140124437-140124442:

or input a list of coordinates to fetch sequence (max 100 regions, each should less than 10KB, regions longer than 10Kb would be ignored):

```
chr6:52425276-52425961
chr1:10001000-10001400
```

Click the **Fetch** or **Batch fetch** button to fetch the sequence. Click the **Copy** button can copy the fetched sequence to your clipboard.



To fetch a sequence, choose a region shorter or equal to 10Kb, or input a list of coordinates each less than 10Kb.

Fetch sequence for current view region chr7:140124437-140124442:

```
>chr7:140124437-140124442
TATGT
```

or input a list of coordinates to fetch sequence (max 100 regions, each should less than 10KB, regions longer than 10Kb would be ignored):

```
chr6:52425276-52425961
chr1:10001000-10001400
```

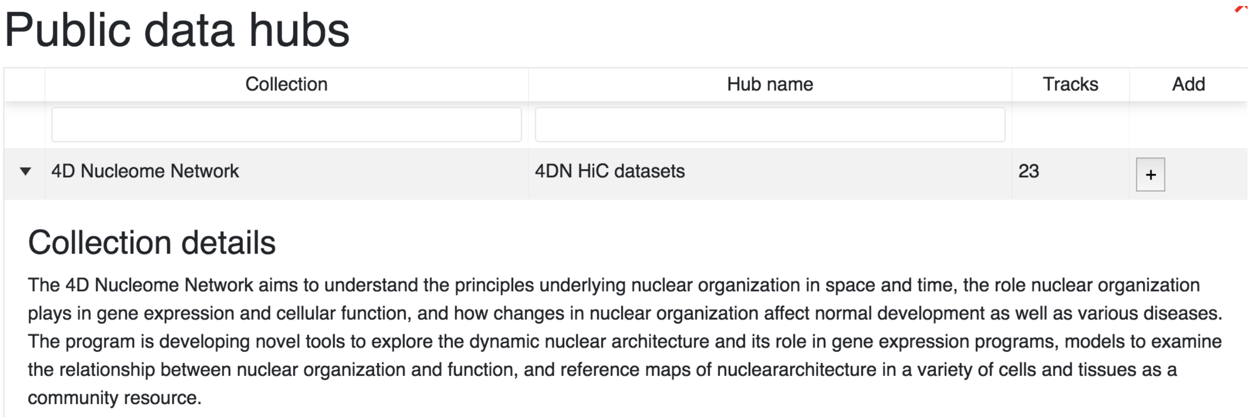
```
>chr6:52425276-52425961
AGATTGATCTCATCCTCTTTTAAACAGCTAGAGTATTCCATTGTATGGCTATACCACAATTTACCTAATCTGTTCATTGTAGGTGGATATTTGGGTTAT
TTCAAACGAAGTCTGCCGTTAGAGCAGGAGCTGTGTCCTGTGTCCCGTGTGTTTTGGCATCTCCATGGAGTTCACACTGTCGTATGAACTCTAACAACA
TTTGCTCTGAGTCACAGAGGATTACTGGCTCACAGTTTATCAAATATAACTCTCAATAACCAGCACTGCTGGGTCAAGCCAGAATAATGGGCAATATTC
ATAATACTGGCCCTGAGCTATCACTCTCAGATCAAAATACAAGCTGTTCAATATCGTTTTGATGCTGGCTCCTTGCCTATATTAAAATTAATTAATGAC
CCAGCAATTCCACCCCCAAGGTATATATCCAAGACAACTGAACACGACATCCACACAAAAACTTGCACACGCGTGTTCAGAGCAGCATTATTCATAATC
GCCAAAAGAGACAAACGTCCATCAACTGATGAATGCATAAACAAAATGCACTTCGCAGTACAATAAAATGATTCAGTCATAAAAAGAAATGAAATACTG
ATACATGCTACAGCATAGATGAACCCTGAAAACATGCTACGTGAAAGAAGCCAGGTACAAAAAACCACATATGGTATGATCCCATGTATAT
>chr1:10001000-10001400
GTAACTGACAATGTCTTTTCAGAGTTAGTTTCCTACAGTCTAACCACACTGGATGACTTGCTGTTCCAAAAATGTCATGCACTTTTTTGAGCCTGCATT
TTTTTTTCTACCTATGAAGCCCAGTTCATTCTTCAAAGTCTAGTTCAAGTGTCACCTTCTCTACAAGGTCTTCCTTGATTTCCTCCCAATCACTAGGTT
GTGAATTATTTAAGTAGGGATTTTACCTTTTCTTTCTTTCGTTCGTTCGTTCTTTCTTTCCTCTTTCTTTTTTTTTTGATGGCGTCTCACGGTCACCCA
GGTGGAGTGCAGTGGCATGATCTTGGTTCACTGCAACCTCTGCCTCCTGGGTTCAAGTGACTCTCCTGCCTCGGTCTCCCGAGTAGCTGGGAATGTAGG
TGCC
```

1.6 Track management

The browser collects data from large corsortia like Roadmap Epigenomics, ENCODE, 4DN, TaRGET, etc. The data are called public data/tracks and are organized into different collections called hubs. Along with these public hubs and tracks users can submit their own custom tracks and data hubs to allow for easy comparison.

1.6.1 Add tracks from public hubs

From the `Tracks` menu choose **Public Data Hubs**. This will display all of the public data hubbs available for the species and build you are currently working in. For example, using mouse mm10 annotation the *4D Nucleome Network* hub is available. Click the *Add* button to load this hub:



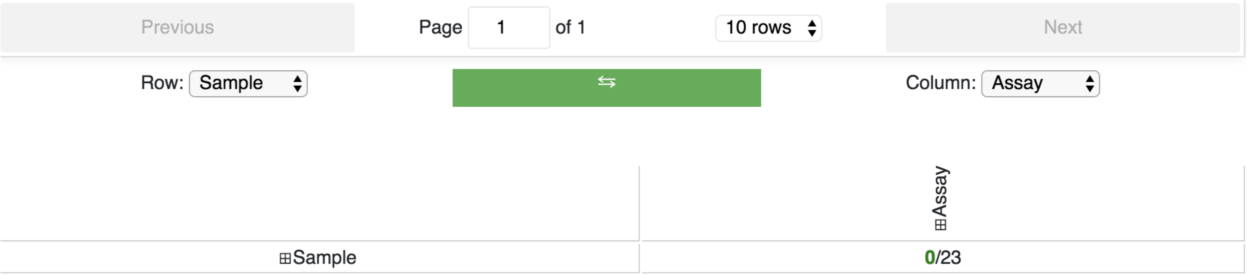
Public data hubs

	Collection	Hub name	Tracks	Add
▼	4D Nucleome Network	4DN HiC datasets	23	+

Collection details

The 4D Nucleome Network aims to understand the principles underlying nuclear organization in space and time, the role nuclear organization plays in gene expression and cellular function, and how changes in nuclear organization affect normal development as well as various diseases. The program is developing novel tools to explore the dynamic nuclear architecture and its role in gene expression programs, models to examine the relationship between nuclear organization and function, and reference maps of nucleararchitecture in a variety of cells and tissues as a community resource.

After a hub is added, a `facet table` containing all tracks will pop up. This allows you to choose any tracks you are interested in. The `facet table` can also be revisted through the menu when you choose **Track Facet Table**:

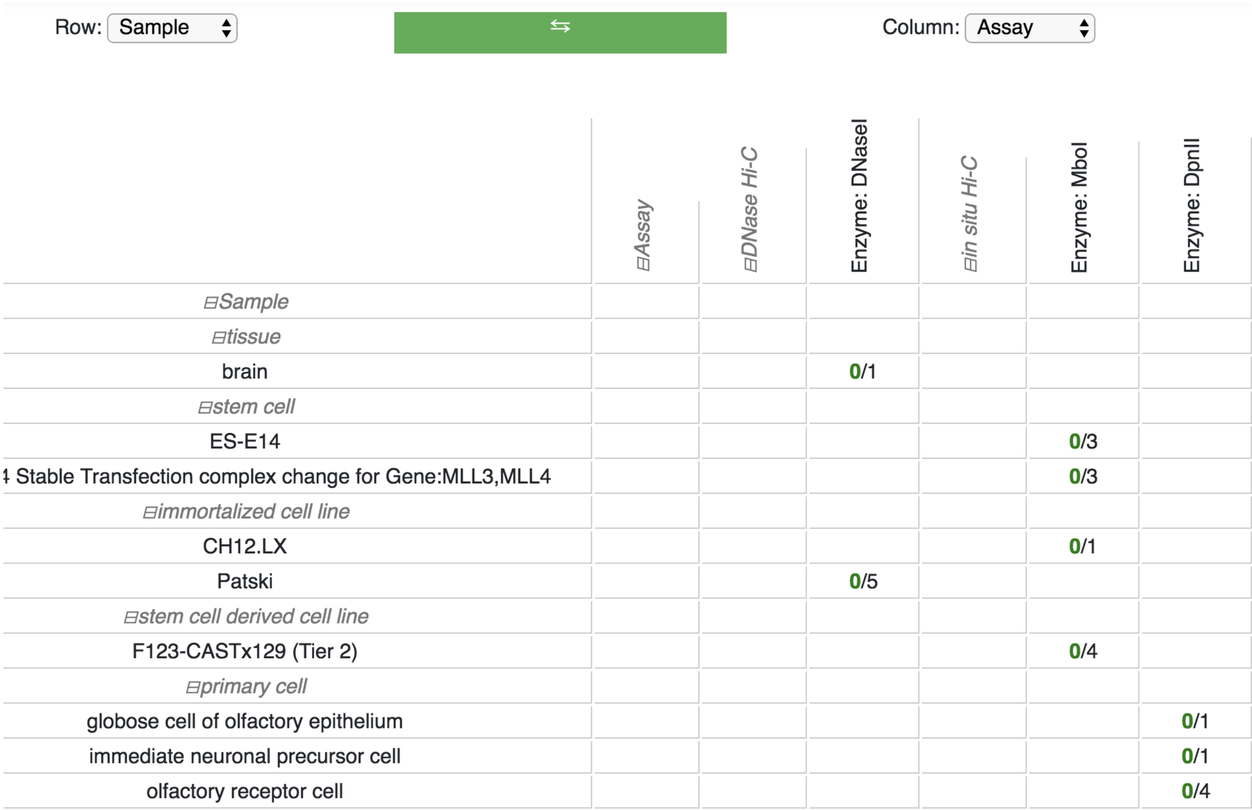


Page 1 of 1 | 10 rows

Row: Sample | Column: Assay

	⊞Assay
⊞Sample	0/23

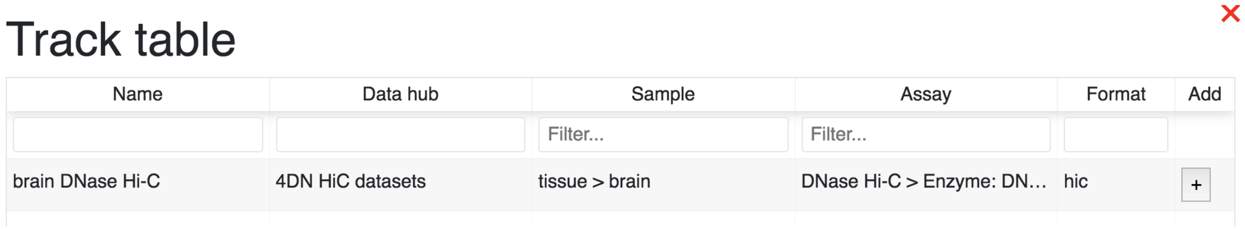
You can expand the row and/or column selection by clicking the + buttons. Row and column displays can also be easily swapped:



Row: Sample ⇆ Column: Assay

	⊟Assay	⊟DNase Hi-C	Enzyme: DNaseI	⊟in situ Hi-C	Enzyme: MboI	Enzyme: DpnII
⊟Sample						
⊟tissue						
brain			0/1			
⊟stem cell						
ES-E14					0/3	
Stable Transfection complex change for Gene:MLL3,MLL4					0/3	
⊟immortalized cell line						
CH12.LX					0/1	
Patski			0/5			
⊟stem cell derived cell line						
F123-CASTx129 (Tier 2)					0/4	
⊟primary cell						
globose cell of olfactory epithelium						0/1
immediate neuronal precursor cell						0/1
olfactory receptor cell						0/4

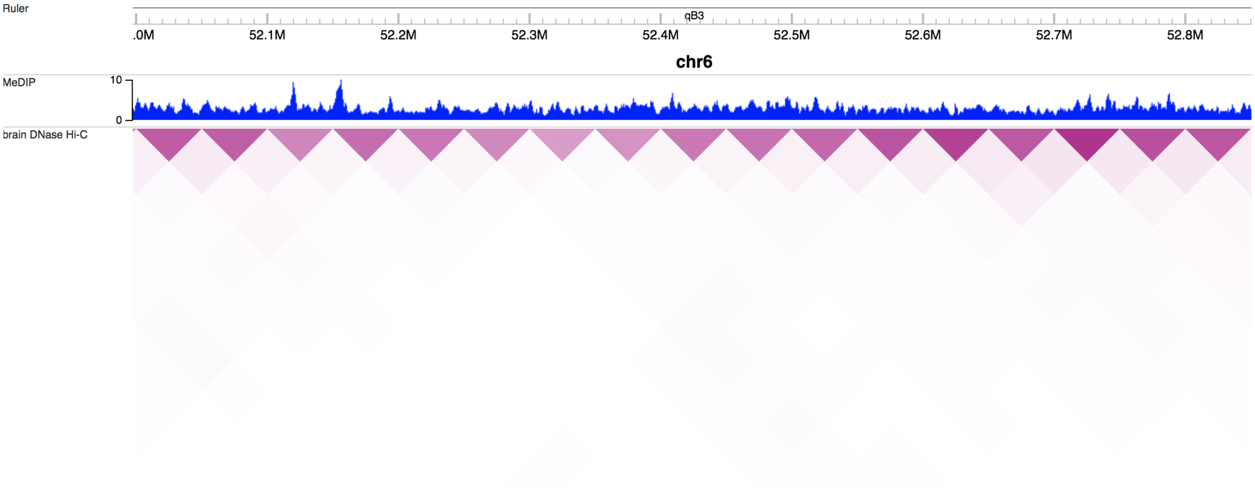
Clicking a cell within the facet table will pop up a new window containing a table with the tacks that match the row and column selections:



Track table ✕

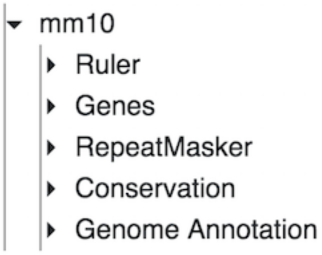
Name	Data hub	Sample	Assay	Format	Add
		Filter...	Filter...		
brain DNase Hi-C	4DN HiC datasets	tissue > brain	DNase Hi-C > Enzyme: DN…	hic	+

Click the *Add* button to add the track(s) you want. You can then view tracks in the browser view window:

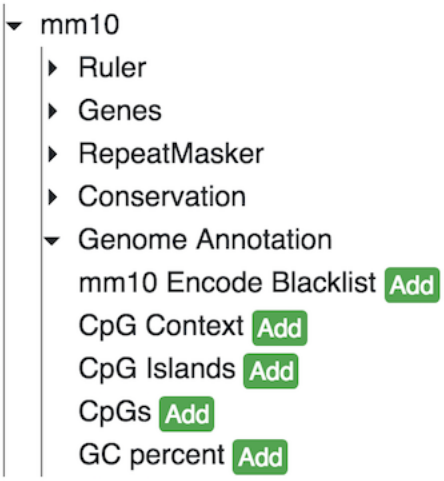


1.6.2 Adding annotation tracks

Users can add numerous annotation tracks from the `Tracks` menu by choosing **Annotation Tracks**.

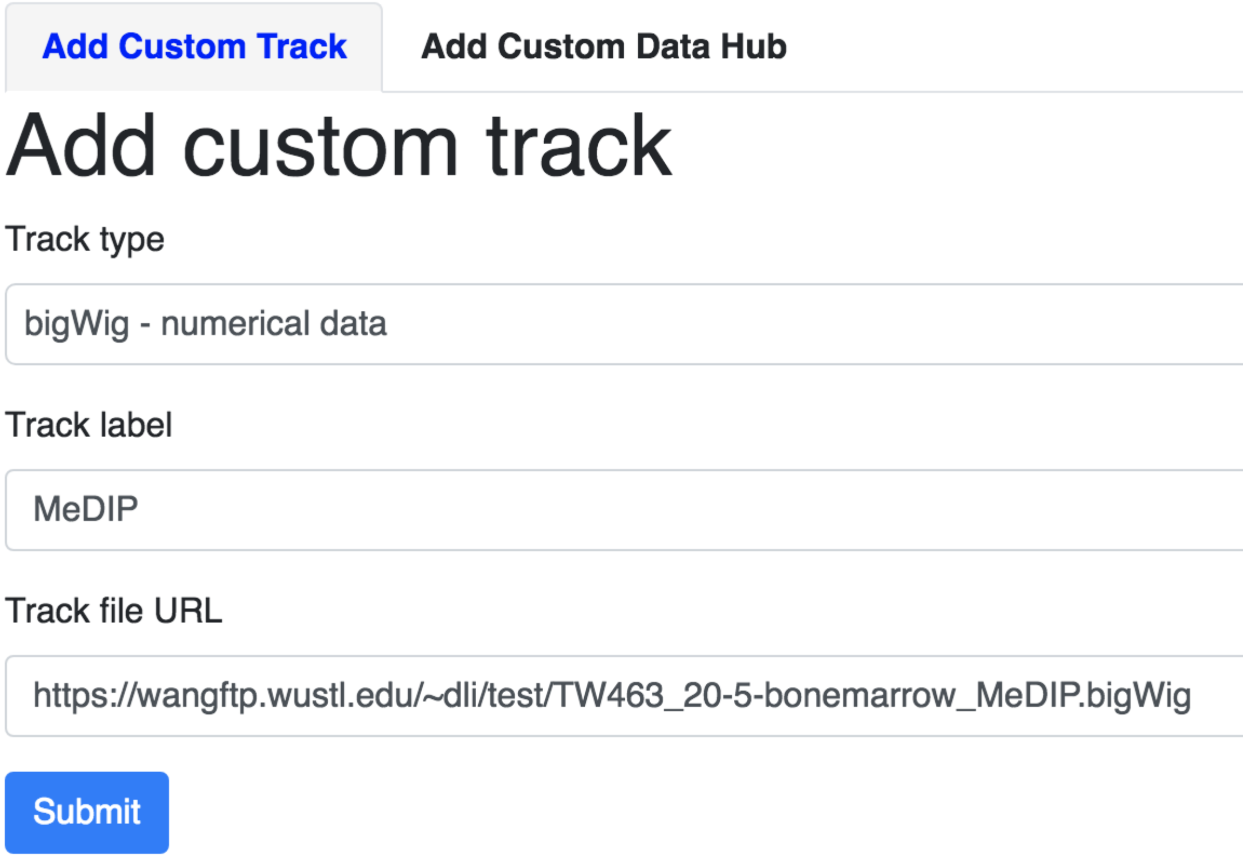


Each header can be expanded to one or more submenus that display tracks that can be added to the browser. The tracks include CpG island information, repeat information, G/C content information, and conservation information to name a few.



1.6.3 Adding a custom track or data hub

Users can also submit their own track as a custom track. For example, say we have a bigWig track located at https://vizhub.wustl.edu/public/tmp/TW463_20-5-bonemarrow_MeDIP.bigWig . From the `Tracks` menu choose **Custom tracks** and a custom track interface will pop up. Fill in the track type, label, and URL before clicking the green *Submit* button:



Add custom track

Track type

bigWig - numerical data

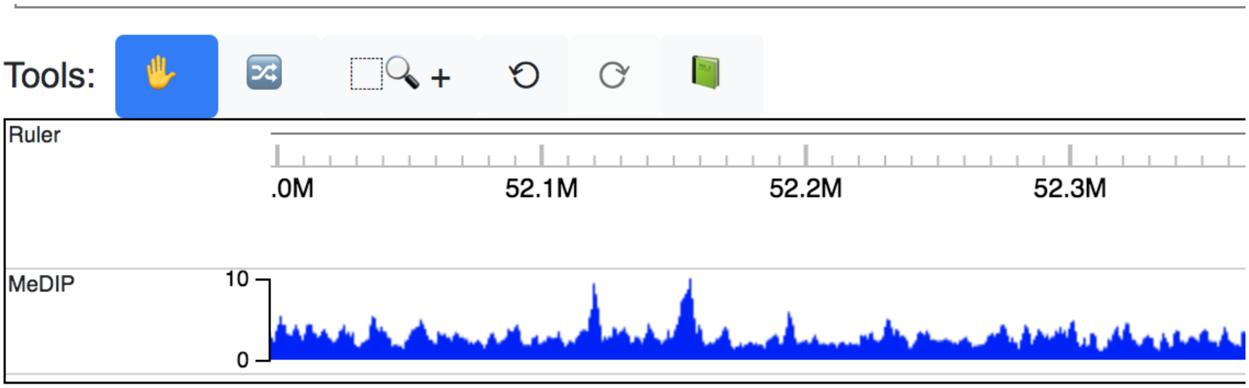
Track label

MeDIP

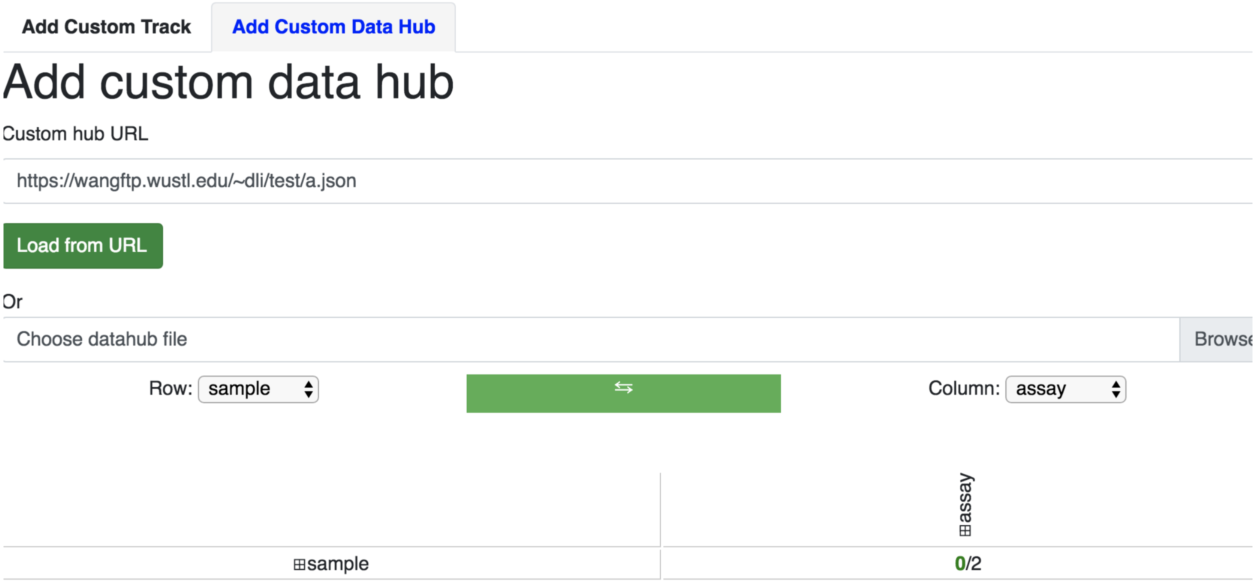
Track file URL

https://wangftp.wustl.edu/~dli/test/TW463_20-5-bonemarrow_MeDIP.bigWig

You can see the track is added:



Adding a custom data hub is similar to the steps above. For example, say you have a hub located at https://vizhub.wustl.edu/public/tmp/a.json . From the Tracks menu choose **Custom tracks**, switch to the *Add custom data hub* tab, paste the URL of your hub, and then click the green *Load From URL* button. from URL.



Add custom data hub

Custom hub URL: https://wangftp.wustl.edu/~dli/test/a.json

Choose datahub file

Row: sample

Column: assay

The tracks within the custom hub can then be added from the generated facet table.

Note: Tracks from custom hubs are hidden by default as users may submit a hub contains hundreds of tracks. Users should add tracks that they want from the facet table.

You can also load a local data hub file in JSON format from your computer using the *file upload* interface, right below the *URL submit* hub interface.

Also see the *Tracks* and *Datahub* sections for how to prepare your tracks and datahub files.

1.7 Track Customization

Tracks can be customized in a multitude of manners.

1.7.1 Selecting Tracks

An indivdual track can be selected by simply right clicking on the tracking on the track. Multiple tracks can be selected by either holding the *shift* button and left clicking on each track or by holding shift and left clicking on a shared metadata term of consecutive tracks. In this manner, multiple tracks can be customized or moved at the same time. To deselect the tracks simply right click and press the button `Deselect # tracks` .

1.7.2 Track Color

Right clicking on `annotation` and `numerical` tracks will display `Primary Color`, `Secondary Color`, and `Background Color` which can all be customized using the color picker. For `methylC` tracks and `categorical` tracks the `Color` and `Background` of each class of elements (e.g. CG, CHG, and CHH) can be personalized. Additionally, for methylC``tracks the ``Read depth line color can be customized.

1.7.3 Track Height

For each track the height can be customized by right clicking on the track and typing in a number to the panel. At 20 pixels and below for `numerical` tracks the track will display as a heatmap.

1.7.4 Track Display Mode

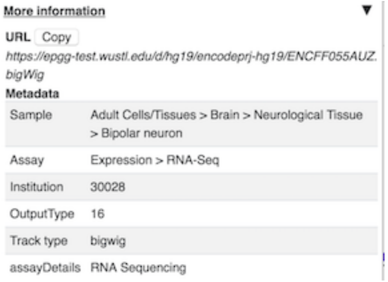
For each `numerical`, `annotation`, or `BAM` track the display can be changed to `DENSITY` or `FULL` mode by right clicking on the track.

1.7.5 Track Y-axis Scale

For each `numerical` track the y-axis can be displayed in `AUTO` or `FIXED` mode by right clicking on the track. The `AUTO` mode will scale the axis based on numerical values in the immediate area of the view range. The `FIXED` mode allows the user to select `a Y-Axis min` or `Y-axis max`. For values above the set max the `Primary color above max` can be set for easy viewing. For values below the set minimum the `Primary color below min` can bet set.

1.7.6 Track Information

If `details` were specified for a track in the data hub file these can be viewed by right clicking on the sample and clicking on the arrow to the right. An easy access `copy` but is also available to copy the `URL` for the track.



URL

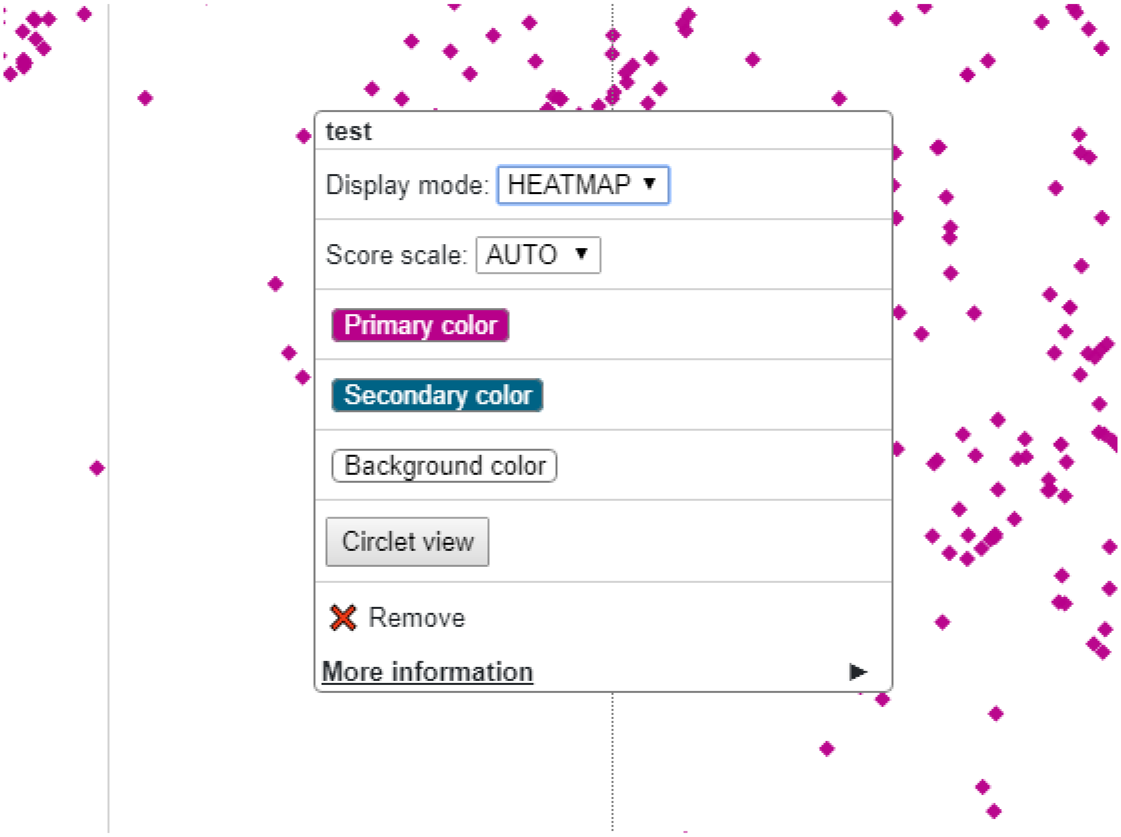
https://epgg-test.wustl.edu/d/hg19/encodeprj-hg19/ENCFF055AUZ.bigWig

Metadata

Sample	Adult Cells/Tissues > Brain > Neurological Tissue > Bipolar neuron
Assay	Expression > RNA-Seq
Institution	30028
OutputType	16
Track type	bigwig
assayDetails	RNA Sequencing

1.8 Circlet view for chromatin interaction tracks

For any chromatin interaction track type (*HiC*, *longrange*, *bigInteract*), when you right click the track, you can see the `Circlet view` button:

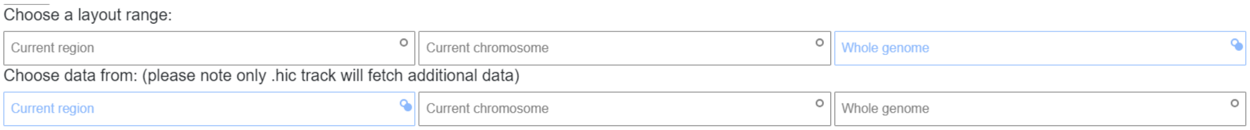


test

Display mode: HEATMAP ▼

Score scale: AUTO ▼

Click the button will bring you to the **Circlet view** interface. You can config the layout and/or the data source:

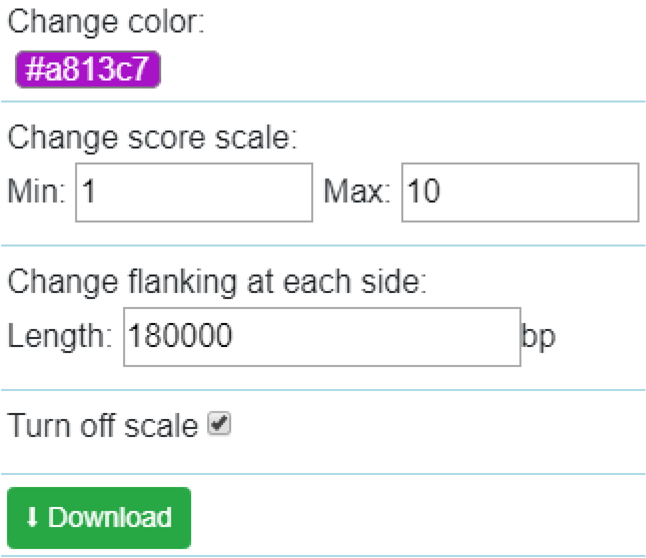


Choose a layout range:

○ Current region ○ Current chromosome ● Whole genome

Choose data from: (please note only .hic track will fetch additional data)

● Current region ○ Current chromosome ○ Whole genome



Change color:

#a813c7

Change score scale:

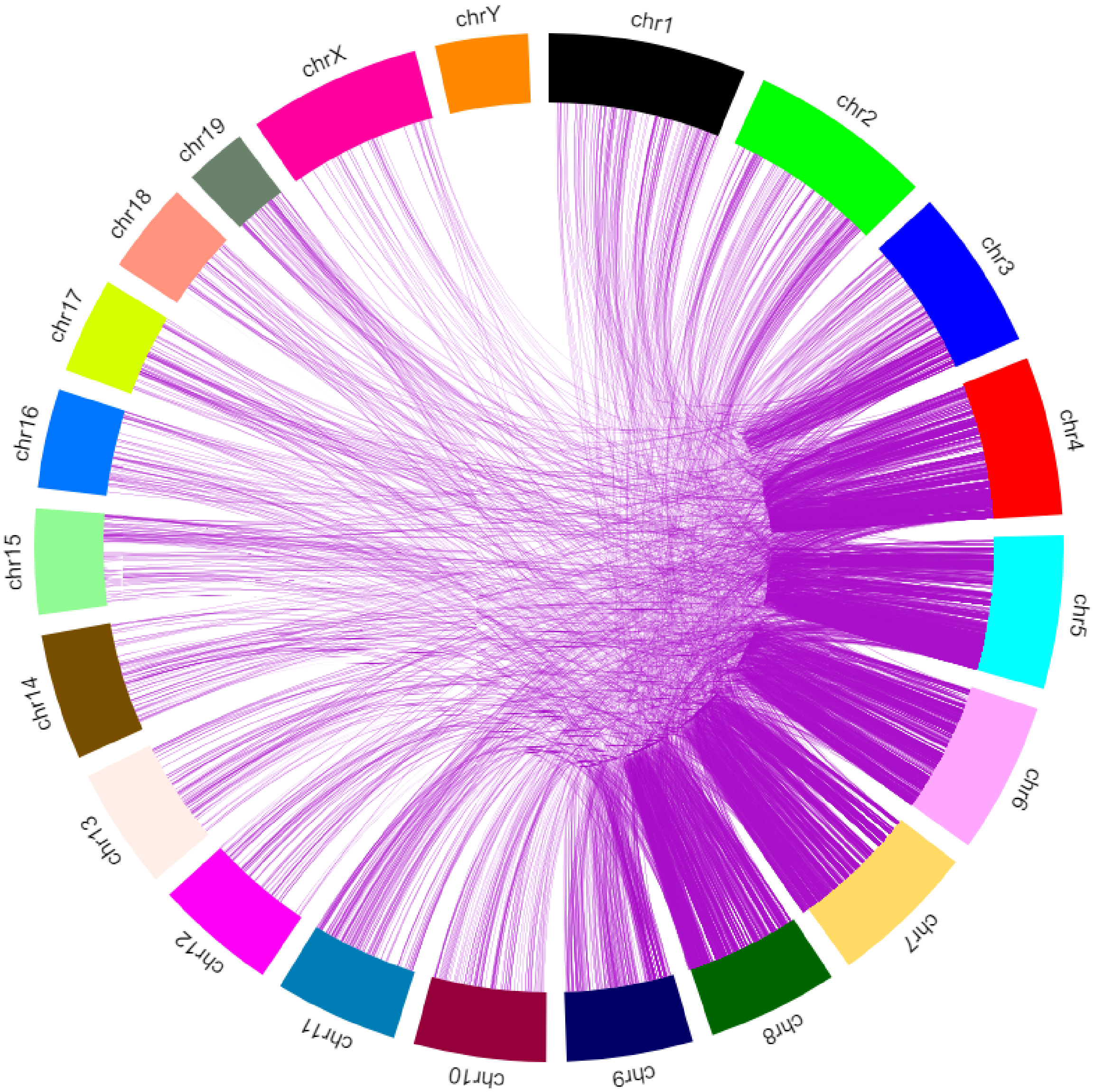
Min: 1 Max: 10

Change flanking at each side:

Length: 180000 bp

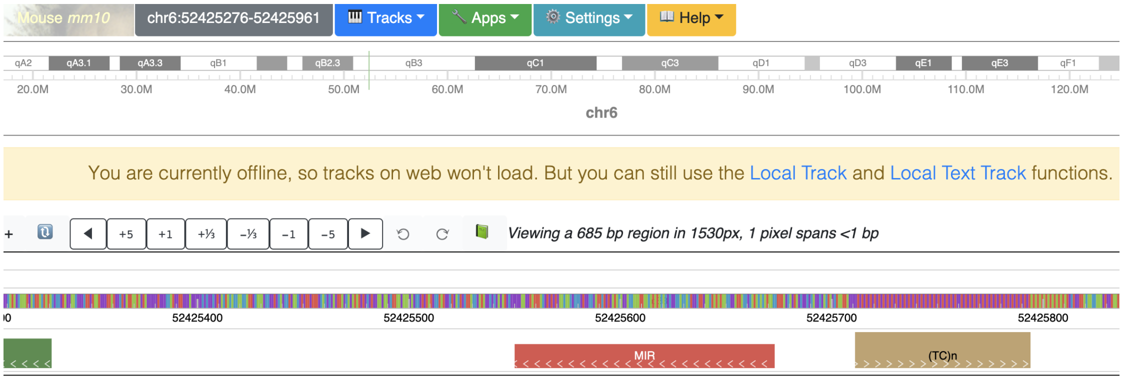
Turn off scale ☑

And config the color, scale, flanking region length at each end of one interaction. You also can download the view as a SVG file used for publication.



1.9 Offline mode

In case your device goes offline (no WIFI or network is down), the use can still use the local track and local text track function. A notice will show as below to indicate user's device is currently offline.

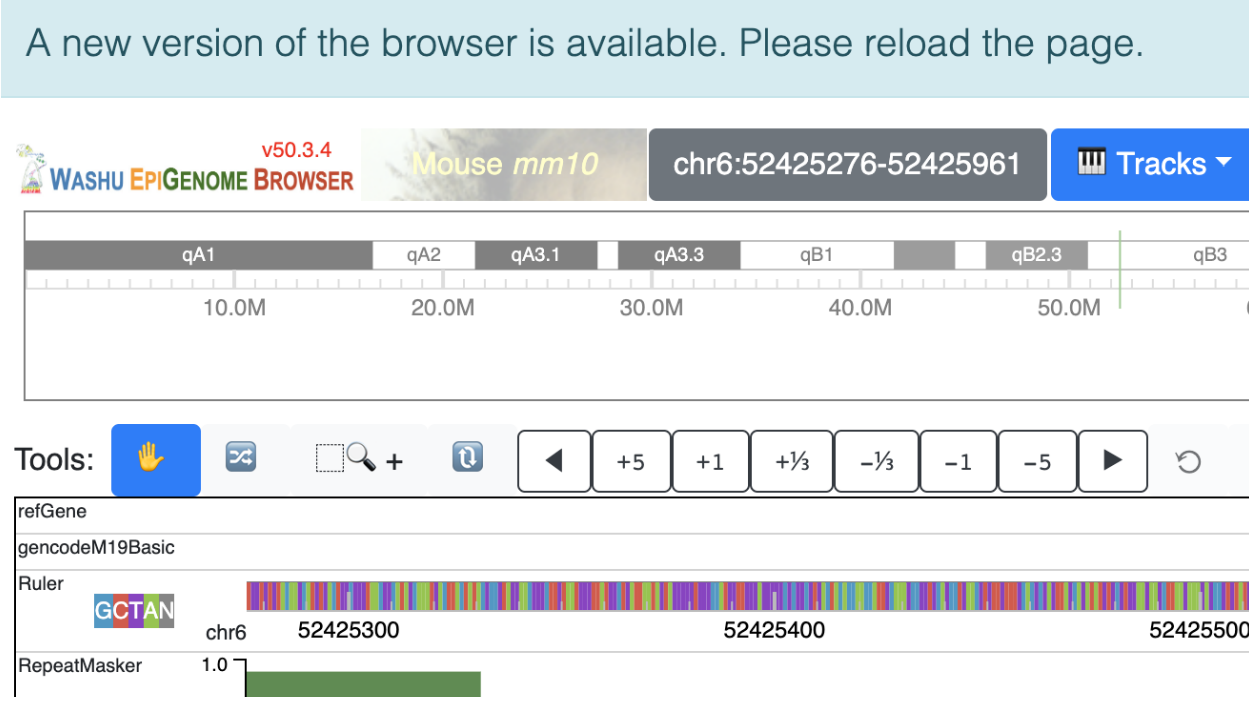


You are currently offline, so tracks on web won't load. But you can still use the Local Track and Local Text Track functions.

Viewing a 685 bp region in 1530px, 1 pixel spans <1 bp

1.10 New version notice

Whenever there is a new version, a notification will show if user still use the old version.


A new version of the browser is available. Please reload the page.

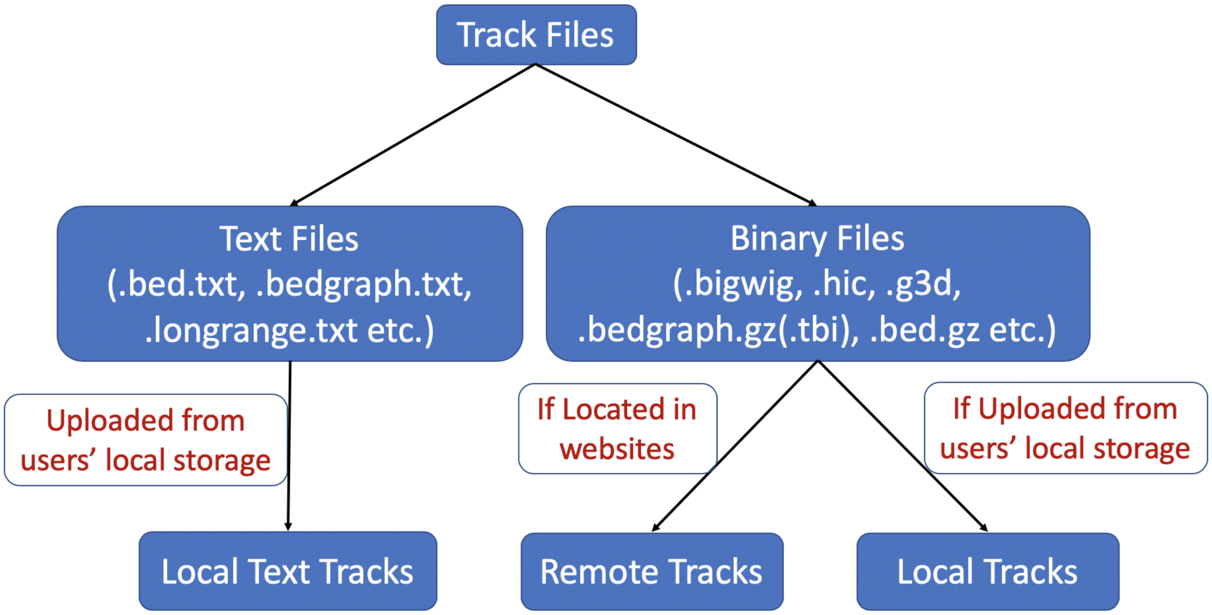
CHAPTER

TWO

TRACKS

2.1 Track groups based on file types and localtions of the track files

Track files are divided to 2 groups based on their file types, text format files and binary files like `bigWig` and `hic`. For binary track files, if the track files are located at websites, they are *Remote Tracks*, if they are located in users' computer then they are *Local Tracks*. For text track files, right now they can be uploaded from users' computer, they are called *Local Text Tracks*. Please check the following diagram as well:



Track Files
Text Files (.bed.txt, .bedgraph.txt, .longrange.txt etc.)
Binary Files (.bigwig, .hic, .g3d, .bedgraph.gz(.tbi), .bed.gz etc.)
Uploaded from users' local storage
If Located in websites
If Uploaded from users' local storage
Local Text Tracks
Remote Tracks
Local Tracks

Important: Since all remote tracks are hosted on the web with HTTP/HTTPS links provided for submission as tracks, the webservers which are hosting the track files need Cross-Origin Resource Sharing (CORS) enabled.

Quoted from MDN:

```
Cross-Origin Resource Sharing (CORS) is a mechanism that uses additional
HTTP headers to tell a browser to let a web application running at
one origin (domain) have permission to access selected resources
from a server at a different origin. A web application makes
```

(continues on next page)

(continued from previous page)

```
cross-origin HTTP request when it requests a resource that has
a different origin (domain, protocol, and port) than its own origin.
```

2.2 Configure your webserver to enable CORS

Most likely the browser domain is different from the server the tracks are hosted on. The hosting server needs CORS enabled. For any Apache web server, you might try the either following approach.

2.2.1 Enable CORS on Apache2 under Ubuntu

For an Apache web server in Ubuntu this setup (add this to the enabled .conf file) would work:

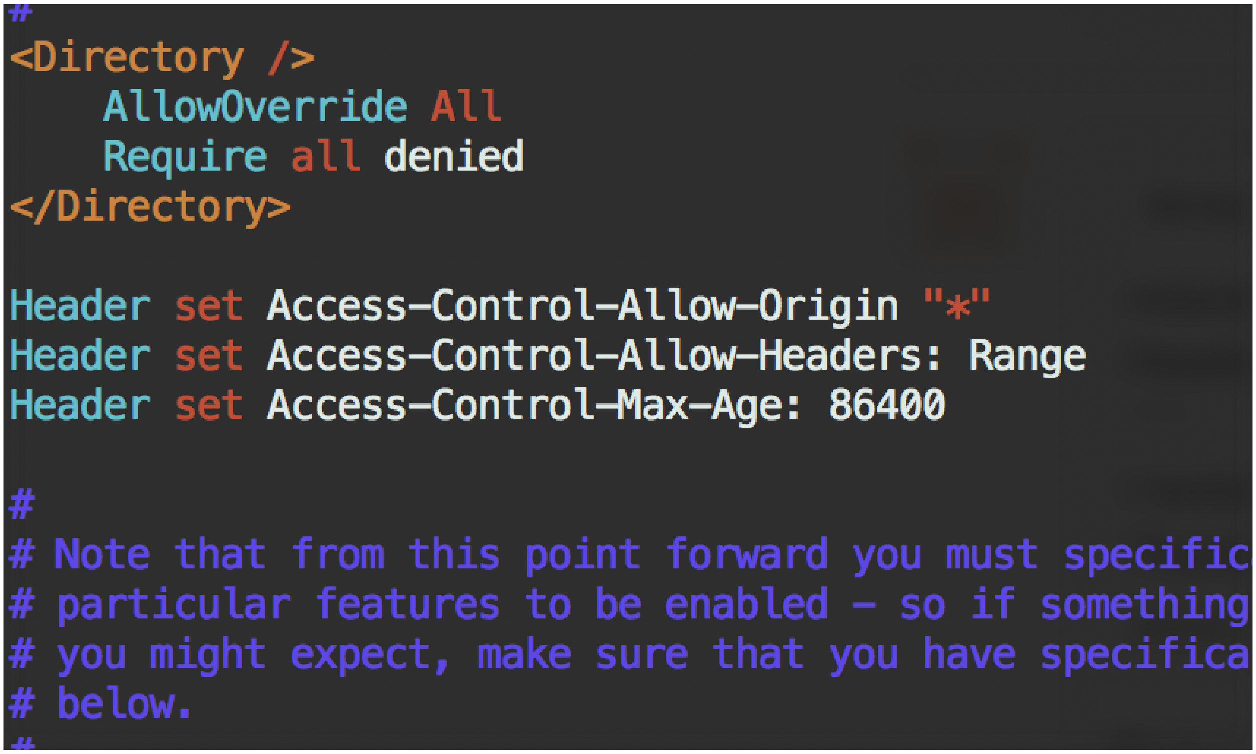
```
Header always set Access-Control-Allow-Origin "*"
Header always set Access-Control-Allow-Headers: Range
Header always set Access-Control-Max-Age: 86400
```

Then restart your Apache server.

2.2.2 Enable CORS on Apache2 under CentOS

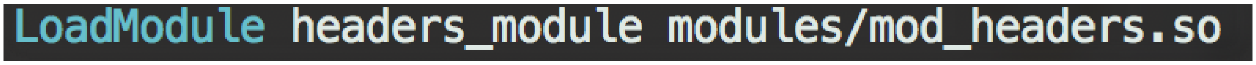
Try add this to the main configuration file `/etc/httpd/conf/httpd.conf`:

```
Header always set Access-Control-Allow-Origin "*"
Header always set Access-Control-Allow-Headers: Range
Header always set Access-Control-Max-Age: 86400
```



```
#
<Directory />
    AllowOverride All
    Require all denied
</Directory>

Header set Access-Control-Allow-Origin "*"
Header set Access-Control-Allow-Headers: Range
Header set Access-Control-Max-Age: 86400

#
# Note that from this point forward you must specific
# particular features to be enabled - so if something
# you might expect, make sure that you have specifica
# below.
#
```

in `/etc/httpd/conf.modules.d/00-base.conf`, the header module should be enabled:



```
LoadModule headers_module modules/mod_headers.so
```

Then restart your Apache server.

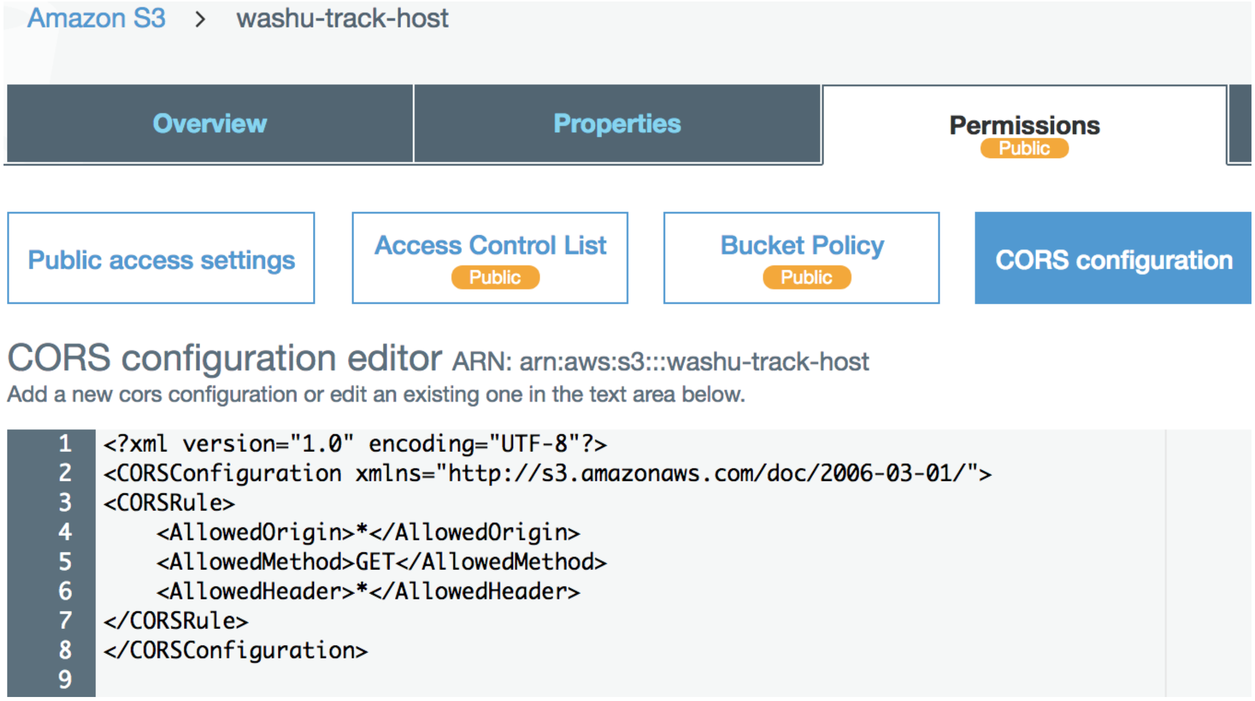
2.2.3 Enable CORS on Amazon S3 bucket

We have setup a test s3 bucket at http://washu-track-host.s3-website-us-east-1.amazonaws.com and tried *bigWig* files, the link http://washu-track-host.s3-website-us-east-1.amazonaws.com/bigwig/TW551_20-5-bonemarrow_MRE.CpG.bigWig can be displayed at the browser with following CORS setup:

```
[
    {
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
            "GET",
            "HEAD"
        ],
        "AllowedOrigins": [
            "*"
        ]
    }
]
```

If you happen to use old XML settings, you can setup it like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<CORSRule>
    <AllowedOrigin>*</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <AllowedHeader>*</AllowedHeader>
</CORSRule>
</CORSConfiguration>
```



CORS configuration editor ARN: arn:aws:s3:::washu-track-host

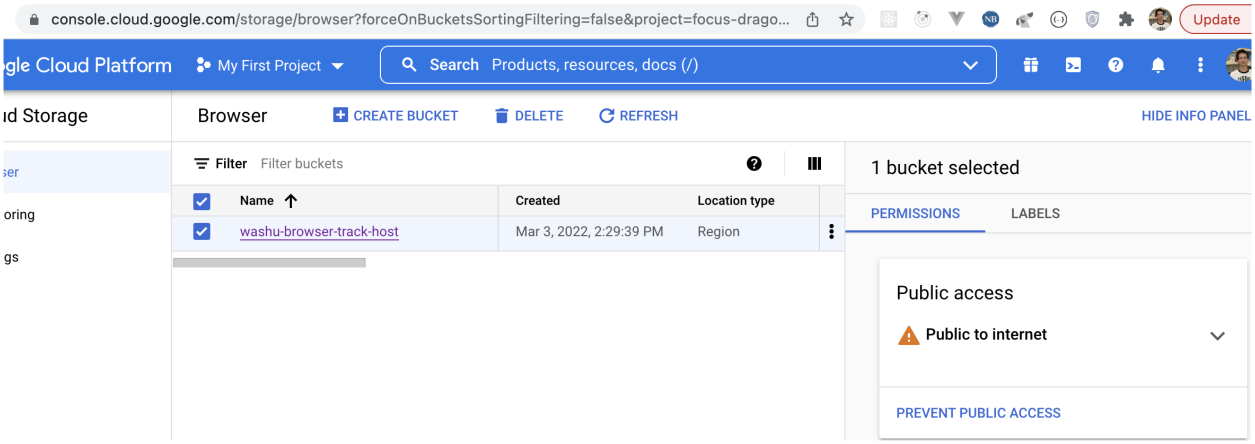
Add a new cors configuration or edit an existing one in the text area below.

```
<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<CORSRule>
    <AllowedOrigin>*</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <AllowedHeader>*</AllowedHeader>
</CORSRule>
</CORSConfiguration>
```

2.2.4 Enable CORS on Google cloud storage

If you have track files hotsed in Google cloud storage, they can be viewed in the browser as well after setting up correct CORS policy.

First you need make the bucket public, for more information you can check the docs from google:



Name	Created	Location type
washu-browser-track-host	Mar 3, 2022, 2:29:39 PM	Region

Then you can use either the gsutil tool or the CloudShell in your Google cloud's web console. Create a file called `cors.json` with contents below:

```
[
    {
    "origin": ["*"],
    "method": ["GET", "HEAD"],
    "responseHeader": ["Authorization", "Content-Range", "Accept", "Content-Type",
↪"Origin", "Range"],
```

(continues on next page)

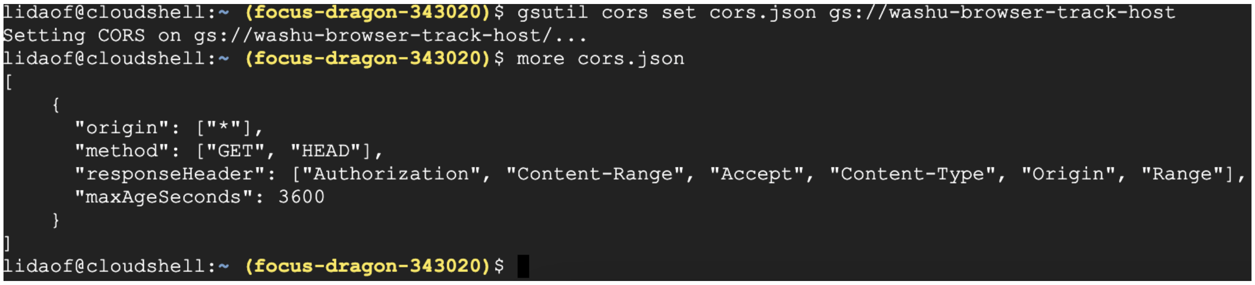
(continued from previous page)

```
    "maxAgeSeconds": 3600
    }
]
```

then set the CORS policy to your bucket with the command below:

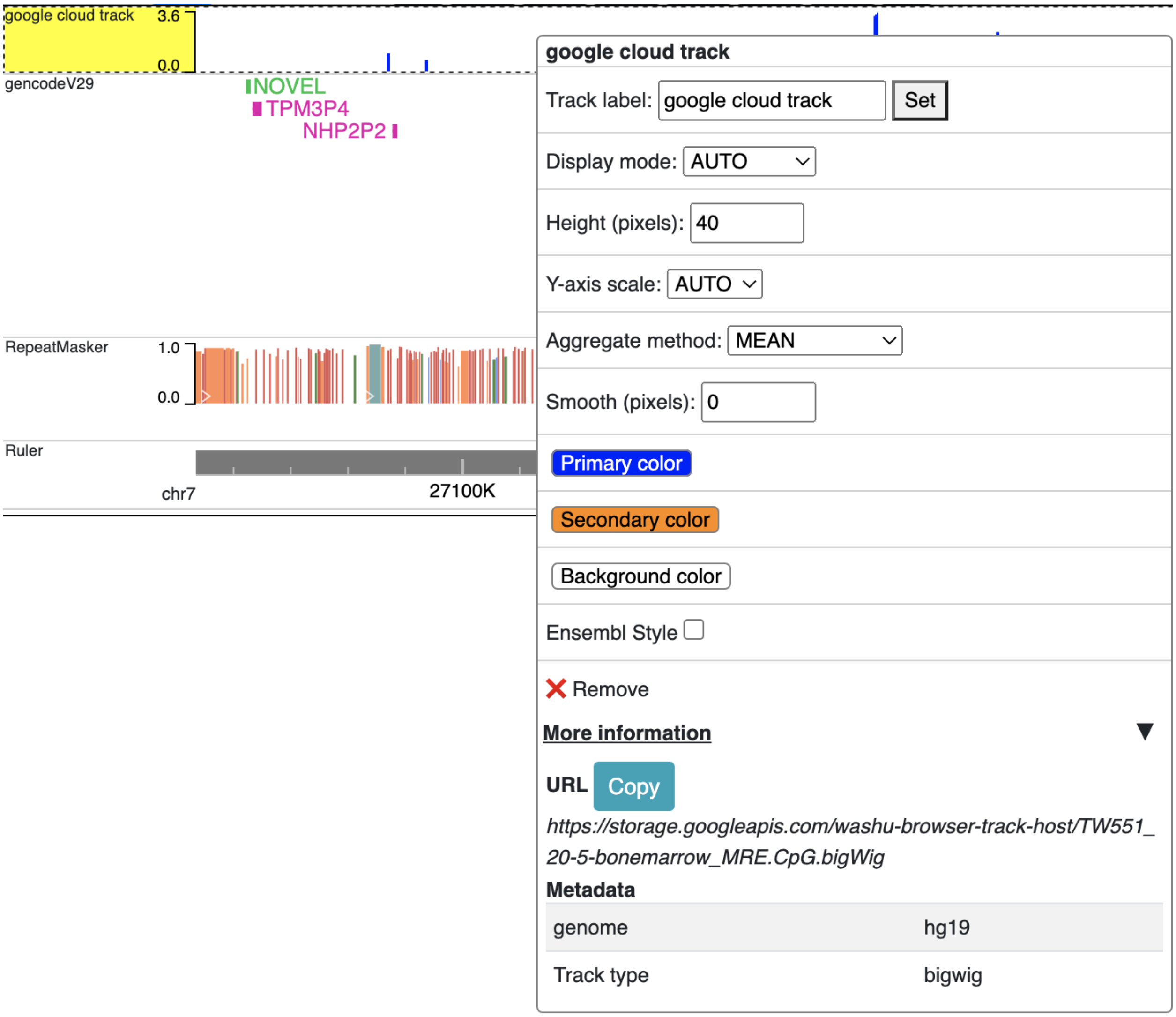
```
gsutil cors set cors.json gs://washu-browser-track-host
```

the screenshot below shows how I did in CloudShell in the console web page:



```
lidaof@cloudshell:~ (focus-dragon-343020)$ gsutil cors set cors.json gs://washu-browser-track-host
Setting CORS on gs://washu-browser-track-host/...
lidaof@cloudshell:~ (focus-dragon-343020)$ more cors.json
[
    {
      "origin": ["*"],
      "method": ["GET", "HEAD"],
      "responseHeader": ["Authorization", "Content-Range", "Accept", "Content-Type", "Origin", "Range"],
      "maxAgeSeconds": 3600
    }
]
lidaof@cloudshell:~ (focus-dragon-343020)$
```

After this, you can copy the URL to the file and submit to the browser for visualization.



google cloud track

Track label: google cloud track Set

Display mode: AUTO

Height (pixels): 40

Y-axis scale: AUTO

Aggregate method: MEAN

Smooth (pixels): 0

Primary color

Secondary color

Background color

Ensembl Style □

❌ Remove

More information ▼

URL Copy

https://storage.googleapis.com/washu-browser-track-host/TW551_20-5-bonemarrow_MRE.CpG.bigWig

Metadata

genome	hg19
Track type	bigwig

2.3 Prepare track files

The browser accesses track files from their URL. Only a portion of the data, that within the specific view region, are transferred to the browser for visualization. Thus, all the track files need be hosted in a web accssible location using HTTP or HTTPS. The following sections introduce the track types that the browser supports.

Binary track file formats like *bigWig* and *HiC* can be used directly with the browser.

bedGraph, *methylC*, *categorical*, *longrange* and *bed* track files need to be compressed by bgzip and indexed by tabix for use by the browser. The resulting index file with suffix `.tbi` needs to be located at the same URL with the `.gz` file.

Bed like format track files need be sorted before submission. For example, if we have a track file named `track.bedgraph` we can use the generic Linux `sort` command, the `bedSort` tool from UCSC, or the `sort-bed` command from BEDOPS. Here is an example command using each of the three methods:

```
# Using Linux sort
sort -k1,1 -k2,2n track.bedgraph > track.bedgraph.sorted
# Using bedSort
```

(continues on next page)

(continued from previous page)

```
bedSort track.bedgraph track.bedgraph.sorted
# Using sort-bed
sort-bed track.bedgraph > track.bedgraph.sorted
```

Then the file must be compressed using bgzip and indexed using tabix:

```
bgzip track.bedgraph.sorted
tabix -p bed track.bedgraph.sorted.gz
```

Move files "track.bedgraph.sorted.gz" and "track.bedgraph.sorted.gz.tbi" to a web server. The two files must be in the same directory. Obtain the URL to "track.bedgraph.sorted.gz" for submission.

SAM files first need to be compressed to *BAM* files. *BAM* files need to be coordinate sorted and indexed for use by the browser. The resulting index file with suffix `.bai` needs be located at the same URL with the `.bam` file.

Here is an example command:

```
# Using samtools view to convert to bam
samtools view -Sb test.sam > test.bam
# Using samtools sort to coordinate sort the file
samtools sort test.bam > test.sorted.bam
# Using samtools index
samtools index test.sorted.bam
```

2.4 Annotation Tracks

Annotation tracks represent genomic features or intervals across the genome. Popular examples include SNP files, CpG Island files, and blacklisted regions.

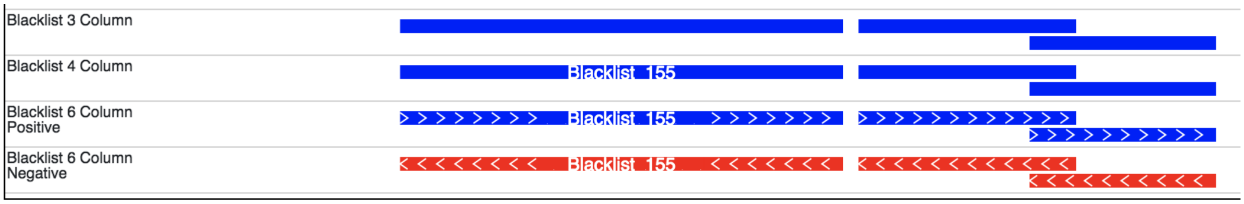
2.4.1 bed

bed format files can be used to annotate elements across the genome or to represent reads from a sequencing experiment. For more about the bed format please check the UCSC bed page.

Example lines are below:

```
chr9       3035610 3036180 Blacklist_155   .       +
chr9       3036200 3036480 Blacklist_156   .       +
chr9       3036420 3036660 Blacklist_157   .       +
```

Every line must consist of at least 3 fields separated by the `Tab` delimiter. The required fields from left to right are `chromosome`, `start position` (0-based), and `end position` (not included). A fourth (optional) column is reserved for the name of the interval and the sixth column (optional) is reserved for the strand. All other columns are ignored, but can be present in the file.



Note: The display of a bed file differs by how many columns are provided in the file (see image above). The simplest, 3 column, format just displays blocks for each interval. The four column format displays the name of each element over each interval. If the sixth column is provided in the file then >>> or <<< will be displayed over each interval to represent strand information.

This format needs to be compressed by bgzip and indexed by tabix for submission as a track. See *Prepare track files*.

2.4.2 bigbed

`bigbed` is a binary format of `bed` file. `bigbed` file can be submitted directly without bgzip/tabix processing. For more about the bed format please check the UCSC bigbed page.

2.4.3 refbed

The `refbed` format files allows you to upload a custom gene annotation track. It is similar to the refGene bed-like file downloaded from UCSC but with slight modifications. Each file of this format contains (each column is separated by *Tab*):

> chr, transcript_start, transcript_stop, translation_start, translation_stop, strand, gene_name, transcript_id, type, exon(including UTR bases) starts, exon(including UTR bases) stops, and additional gene info (*optional*)

This format needs to be compressed by bgzip and indexed by tabix for submission as a track. See *Prepare track files*.

Hint: The 9th column contains gene type, but is simplified from the Gencode/Ensembl annotations to coding, pseudo, nonCoding, problem, and other. These classes of gene type are colored differently when the track is displayed on the browser.

Hint: The 10th and 11th columns contain exon starts and ends respectively. Each start or end is seperated by a comma.

For example:

```
start1,start2,start3,start4 stop1,stop2,stop3,stop4
100,120,140,160 110,130,150,170
```

Hint: The 12th column contains extra information. This information can be manually annotated or we suggest using Ensembl Biomart to download paired Transcript stable IDs and Gene descriptions. The information in this column must be seperated by *spaces* and not tabs.

All of the below lines will work for additional information in the 12th column:

```
Gene ID:ENSMUSG00000103482.1 Gene Type:TEC Transcript Type:TEC Additional Info:predicted gene, 37999 [Source:MGI Symbol;Acc:MGI:5611227]
Gene ID:ENSMUSG00000103482.1 Gene Type:TEC Transcript Type:TEC
ENSMUSG00000103482.1 TEC
Additional Info:predicted gene, 37999 [Source:MGI Symbol;Acc:MGI:5611227]
My Favorite Gene
```

Here are a few example lines in refbed format from gencode.vM17.annotation.gtf (mouse mm10 format):

```
chr1        24910461        24911659        24910461        24911659        -       RP23-
↪109H7.1     ENSMUST00000187022.1    pseudo  24911220,24910461       24911659,24910681  ␣
↪    Gene       ID:ENSMUSG00000100808.1 Gene Type:processed_pseudogene Transcript␣
↪Type:processed_pseudogene Additional Info:predicted gene 28594         [Source:MGI␣
↪Symbol;Acc:MGI:5579300]
chr1        25203443        25205696        25203443        25205696        -      ␣
↪Adgrb3  ENSMUST00000190202.1    coding  25203443        25205696        Gene      ␣
↪                  ID:ENSMUSG00000033569.17 Gene Type:protein_coding Transcript␣
↪Type:retained_intron Additional Info:adhesion G protein-coupled receptor    B3␣
↪[Source:MGI Symbol;Acc:MGI:2441837]
chr1        25276404        25277954        25276404        25277954        -       RP23-
↪21P2.4     ENSMUST00000193138.1    problem 25276404        25277954        Gene       ␣
↪                 ID:ENSMUSG00000104257.1 Gene Type:TEC Transcript Type:TEC Additional␣
↪Info:predicted gene, 20172 [Source:MGI Symbol;Acc:MGI:5012357]
chr1        26566833        26566938        26566833        26566938        +      ␣
↪Gm24064 ENSMUST00000157486.1    nonCoding       26566833        26566938        Gene  ␣
↪                        ID:ENSMUSG00000088111.1 Gene Type:snoRNA Transcript␣
↪Type:snoRNA Additional Info:predicted gene, 24064 [Source:MGI                       ␣
↪Symbol;Acc:MGI:5453841]
```

Note: The last optional column is dislayed as a gene description when a gene is clicked on the browser. Our modified format can be easily obtained from available refGene.bed file downloads from UCSC. Gencode GTF and Ensembl GTF files can be manipulated to this format using the Converting_Gencode_or_Ensembl_GTF_to_refBed.bash script in scripts. The script by default puts `Gene ID:`, `Gene Type:`, and `Transcript Type:` in the additional information column. Run with an annotation file, with columns Transcript_ID and Description (seperated by a tab), the script will also add "Additional Info" to the 12th column. The script depends on bedtools, bgzip, and tabix. Lastly, within the script an `awk` array is used to reclassify gene type and can easily be modified for additional gene types.

The script is run as follows:

```
bash Converting_Gencode_or_Ensembl_GTF_to_refBed.bash Ensembl my.gtf my_optional_
↪annotation.txt
bash Converting_Gencode_or_Ensembl_GTF_to_refBed.bash Gencode gencode.vM17.annotation.gtf
bash Converting_Gencode_or_Ensembl_GTF_to_refBed.bash Gencode gencode.vM17.annotation.
↪gtf biomart_2col.txt
```

Warning: Spaces are used as delimiters in the GTF files so change gene names with spaces before processing.

For Example:

```
sed -i 's/ (1 of many)/_(1_of_many)/g' Danio_rerio.GRCz10.91.chr.gtf
```

2.4.4 rgbpeak

`rgbpeak` track file is based on `bigbed` format, content of a `rgbpeak` file (in bed format) looks like below:

```
chr10 46092019 46092519 chr10_46092019 537 . 46092019 46092519 117,117,117
chr10 47253553 47254053 chr10_47253553 748 . 47253553 47254053 107,107,107
```

where the columns are `chrom, start, end, peak_id, score, strand, thick_start, thick_end, RGB value`, the RBG value will be used for the color while ploting and score will be used to determin the height of the peak. if there is strand, arrow will be drew if zoom enough. thick_start and thick_end columns are ignored now.

The bed file like above can be convert to bigbed format using the commands below:

```
bedSort peaks_rgb.bed peaks_rgb.bed
bedToBigBed peaks_rgb.bed hg38.chroms.sizes peaks_rgb.bigbed
```

2.4.5 bedcolor

Simiar to *bed* track, `bedcolor` track is a 4 column bed file while the 4th column is a color string:

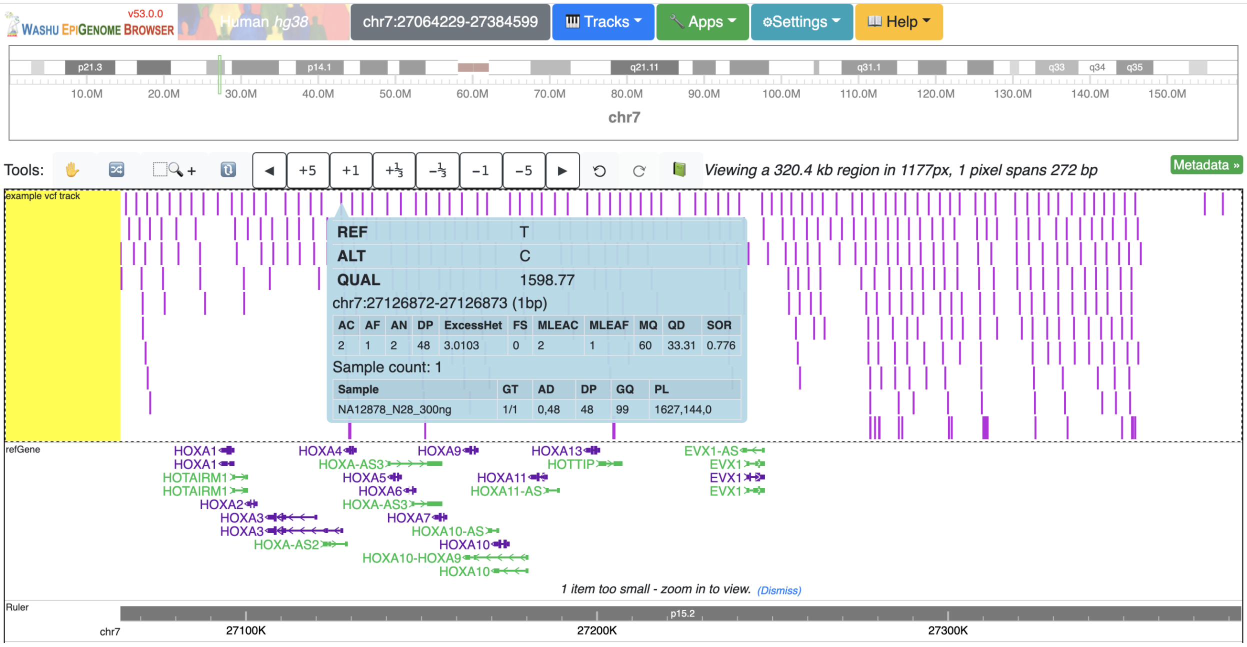
```
chr11       108280000       109080000       #ff0100
chr11       109080000       109480000       #0000ff
chr11       109720000       110160000       #018100
chr11       110200000       111400000       #0064fb
chr11       111400000       112640000       #ef8c0a
chr11       112640000       113480000       #7f007f
chr11       113520000       114520000       #520000
chr11       114520000       114880000       #39ae00
```

It can be uploaded as local text track, or indexed after bgzip/tabix and submitted as remote track.

2.5 Variant Tracks

2.5.1 VCF

`VCF` files can be visulaized in the browser for displaying variant call data. Currently VCF file need to be bgzip and tabix indexed for submission. The VCF track has 3 display modes: *auto*, *density* and *full*. By default it's on *auto* mode, this means when viewing a `VCF` track at a region greater than 100Kb, the track will be displayed as numerical track showing the density of the variant calls, and when view region is less than or equal to 100Kb, it will be displayed in Full mode. The display mode can be changed from the right clicking menu. Click each of the variant item will show the popup tooltip with more information about this variant.



Viewing a 320.4 kb region in 1177px, 1 pixel spans 272 bp

REF	T
ALT	C
QUAL	1598.77

chr7:27126872-27126873 (1bp)

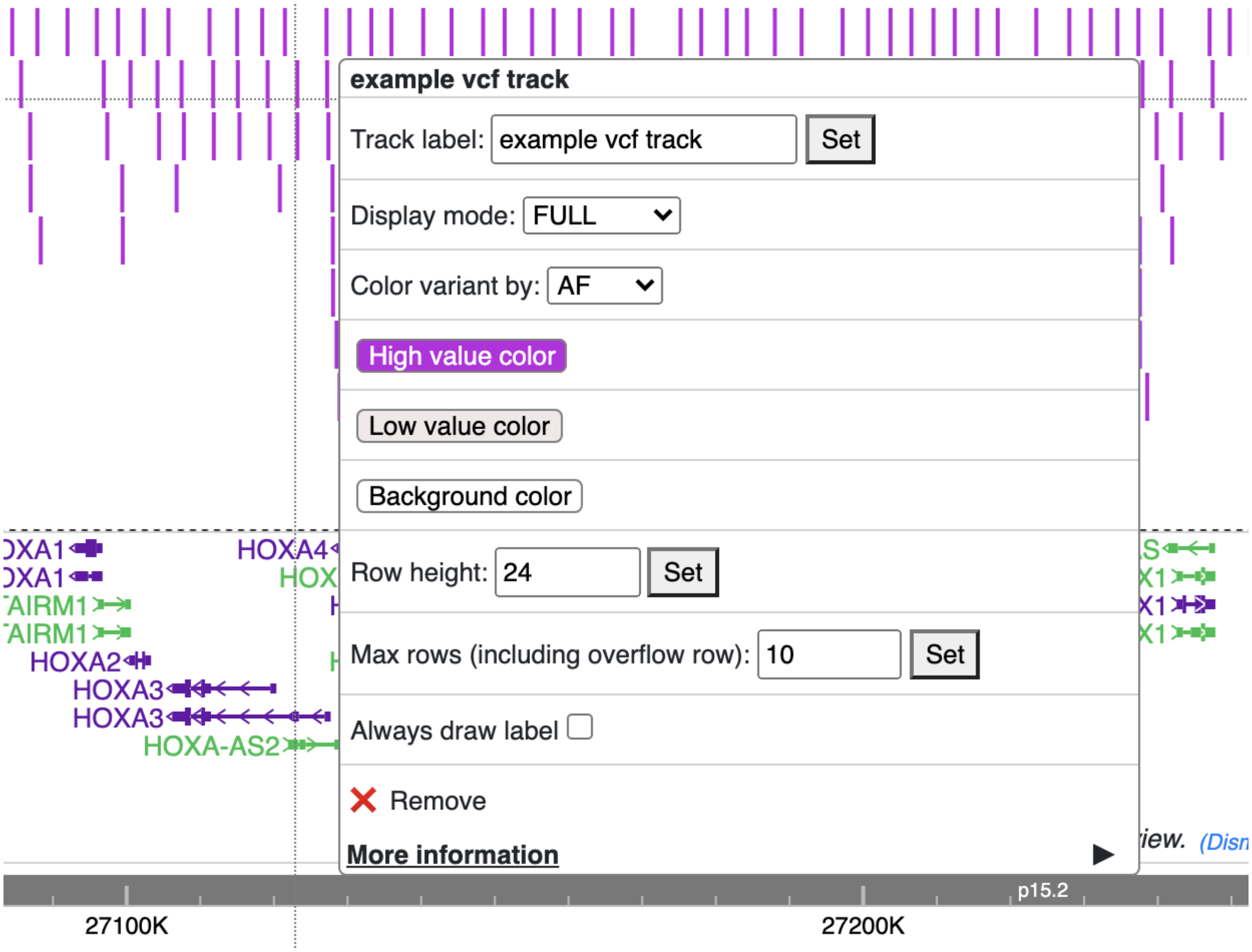
AC	AF	AN	DP	ExcessHet	FS	MLEAC	MLEAF	MQ	QD	SOR
2	1	2	48	3.0103	0	2	1	60	33.31	0.776

Sample count: 1

Sample	GT	AD	DP	GQ	PL
NA12878_N28_300ng	1/1	0,48	48	99	1627,144,0

1 item too small - zoom in to view. (Dismiss)

Color of each variant item are encoded based on the AF or quality value, using which value (AF or quality) to color the variant, or color of high and low value variant can be customized from right clicking menu as well.



example vcf track

Track label: example vcf track Set

Display mode: FULL

Color variant by: AF

High value color

Low value color

Background color

Row height: 24 Set

Max rows (including overflow row): 10 Set

Always draw label □

Remove

More information

2.6 Numerical Tracks

Currently there are two types of numerical tracks:

- *bigWig*
- *bedGraph*

2.6.1 bigWig

`bigWig` is a popular format to represent numerical values over genomic coordinates. Please check the UCSC bigWig page to learn more about this format.

2.6.2 bedGraph

`bedGraph` format also defines values in diffenent genomic locations. For more about the bedGraph format please check the UCSC bedGraph page.

Example lines are below:

```
chr12   6537598 6537599 28.80914
chr12   6537599 6537600 28.96908
chr12   6537599 6537612 -2
chr12   6537600 6537601 29.30229
```

Every line consists of 4 fields separated by the `Tab` delimiter. The fields from left to right are `chromosome`, `start position` (0-based), `end position` (not included), and `value`.

Note: You can use negative values for reverse strand. Both positive and negative values can exist over the same coordinates (they can overlap). In `bigWig` format negative values can also be specified, but they cannot overlap with positive values.

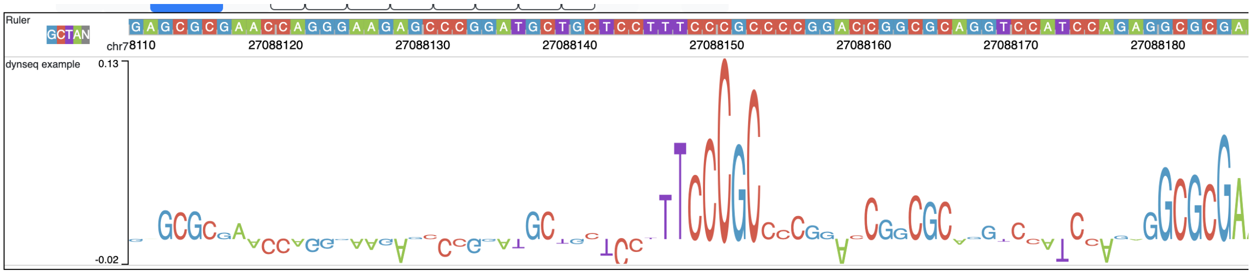
This format needs to be compressed by bgzip and indexed by tabix for submission as a track. See *Prepare track files*.

2.7 Dynamic Sequence Tracks

2.7.1 dynseq

`dynseq` is a new track type which is proposed and initially developped by Surag Nair from Anshul Kundaje's lab at Stanford University. Its track file is the same as `bigWig` format. It provides scores for each nucleotide in the genome, which can be derived from using importance scoring methods on machine learning models. We visualize them as a string of letters with different colors (for each nucleotide) and different heights scaled by the importance scores.

An example of loaded `dynseq` track highlighting an E2F motif instance is illustrated below:



2.8 Read Alignment BAM Tracks

2.8.1 bam

The `bam` format is a compressed SAM format used to store sequence alignment data. Please check the Samtools Documentation page to learn more about this format and how to manipulate these files.

2.9 Methylation Tracks

Methylation experiments like MeDIP-seq or MRE-seq can use *bigWig* or *bedGraph* format for data display. For WGBS if users want to show read depth, methylation context, and methylation level then the data is best suited for the *methylC* format, described below.

2.9.1 methylC

Methylation data are formatted in `methylC` format, which is a 7 column bed format file:

```
chr1    10542   10543   CG      0.923   -       26
chr1    10556   10557   CHH     0.040   -       25
chr1    10562   10563   CG      0.941   +       17
chr1    10563   10564   CG      0.958   -       24
chr1    10564   10565   CHG     0.056   +       18
chr1    10566   10567   CHG     0.045   -       22
chr1    10570   10571   CG      0.870   +       23
chr1    10571   10572   CG      0.913   -       23
```

Each line contains 7 fields separated by Tab. The fields are `chromosome`, `start position` (0-based), `end position` (not included), `methylation context` (CG, CHG, CHG etc.), `methylation value`, `strand`, and `read depth`.

This format needs to be compressed by bgzip and indexed by tabix for submission as a track. See *Prepare track files*.

2.10 Categorical Tracks

Categorical tracks represent genomic bins for different categories. The most popular example is the represnetation of chromHMM data which indicates which region is likely an enhancer, likely a promoter, etc. Other uses for the track include the display of different types of methylation (DMRs, DMVs, LMRs, UMRs, etc.) or even peaks colored by tissue type.

2.10.1 categorical

The `categorical` track uses the first three columns of the standard *bed* format (`chromosome`, `start position` (0-based), and `end position` (not included)) with the addition of a 4th column indicating the category type which can be a string or number:

```
chr1    start1  end1    category1
chr2    start2  end2    category2
chr3    start3  end3    category3
chr4    start4  end4    category4
```

Important: when you use numbers like 1, 2 and 3 as category names, in the datahub definition, please use it a string for the `category` attribute in options, see the example below:

```
{
    "type": "categorical",
    "name": "ChromHMM",
    "url": "https://egg.wustl.edu/d/hg19/E017_15_coreMarks_dense.gz",
    "options": {
        "category": {
            "1": {"name": "Active TSS", "color": "#ff0000"},
            "2": {"name": "Flanking Active TSS", "color": "#ff4500"},
            "3": {"name": "Transcr at gene 5' and 3'", "color": "#32cd32"}
        }
    }
}
```

This format needs to be compressed by bgzip and indexed by tabix for submission as a track. See *Prepare track files*.

2.11 Long range chromatin interaction

Long range chromatin interaction data are used to show relationships between genomic regions. *HiC* is used to show the results from a HiC experiment.

2.11.1 HiC

To learn more about the HiC format please check https://github.com/aidenlab/juicer/wiki/Data.

2.11.2 longrange

The `longrange` track is a *bed* format-like file type. Each row contains columns from left to right: `chromosome`, `start position` (0-based), and `end position` (not included), interaction target in this format `chr2:333-444,55`. As an example, interval "chr1:111-222" interacts with interval "chr2:333-444" on a score of 55, we will use following two lines to represent this interaction:

```
chr1    111 222  chr2:333-444,55
chr2    333 444  chr1:111-222,55
```

Important: Be sure to make **TWO** records for a pair of interacting loci, one record for each locus.

This format needs to be compressed by bgzip and indexed by tabix for submission as a track. See *Prepare track files*.

2.11.3 bigInteract

The bigInteract format from UCSC can also be used at the browser, for more details about this format, please check the UCSC bigInteract format page.

2.11.4 cool

Thanks to the higlass team who provides the data API, the browser is able to display cool tracks by using the data uuid from the higlass server, for example, you can use the uuid `Hyc3TZevQVm3FcTAZShLQg` to represent the track for *Aiden et al. (2009) GM06900 HINDIII 1kb*, for a full list of available cool tracks please check http://higlass.io/api/v1/tilesets/?dt=matrix

2.12 qBED Track

qBED is tab-delimited, plain text format for discrete genomic data, such as transposon insertions. This format requires a minimum of four columns and supports up to six. The four required columns are CHROM, START, END, and VALUE, where VALUE is a numeric value (i.e. an int or float). As with BED files, the START and END coordinates are 0-indexed. The fifth and sixth columns are optional and represent STRAND and ANNOTATION, respectively. The ANNOTATION column can be used to store sample- or entry- specific information, such as a replicate barcode. Here is an example of a four-column qBED file:

```
chr1    41954321        41954325        1
chr1    41954321        41954325        18
chr1    52655214        52655218        1
chr1    52655214        52655218        1
chr1    54690384        54690388        3
chr1    54713998        54714002        1
chr1    54713998        54714002        1
chr1    54713998        54714002        13
chr1    54747055        54747059        1
```

(continues on next page)

(continued from previous page)

```
chr1    54747055        54747059        4
chr1    60748489        60748493        2
```

Here is an example of a six-column qBED file:

```
chr1    51441754        51441758        1       -       CTAGAGACTGGC
chr1    51441754        51441758        21      -       CTTTCCTCCCCA
chr1    51982564        51982568        3       +       CGCGATCGCGAC
chr1    52196476        52196480        1       +       AGAATATCTTCA
chr1    52341019        52341023        1       +       TACGAAACACTA
chr1    59951043        59951047        1       +       ACAAGACCCCAA
chr1    59951043        59951047        1       +       ACAAGAGAGACT
chr1    61106283        61106287        1       -       ATGCACTACTTC
chr1    61106283        61106287        7       -       CGTTTTTCACCT
chr1    61542006        61542010        1       -       CTGAGAGACTGG
```

Your text file must be sorted by the first three columns. If your filename is example.qbed, you can sort it with the following command: `sort -k1V -k2n -k3n example.qbed > example_sorted.qbed` Alternatively, with `bedops`: `sort-bed example.qbed > example_sorted.qbed`

Note that you can have strand information without a barcode, but you cannot have barcode information without a strand column.

Place your sorted qBED file in a web-accessible directory, then compress and index as follows:

```
bgzip example_sorted.qbed
tabix -p bed example_sorted.qbed.gz
```

2.13 genome-align Track

genome-align is tab-delimited, plain text BED-like format to display pairwise whole-genome alignment. It can be directly derived from AXT file. The four required columns are CHROM, START, END, and ALIGNMENT, where ALIGNMENT indicates id number and detailed alignment information in a JSON format

```
chr1    start   end alignment
```

The Fourth column ALIGNMENT contains the following information:

```
"id":1,
"genomealign": {
    "chr": "chr4",
    "start": 154100819,
    "stop": 154100880,
    "strand": "-",
    "targetseq": "ATTGGAGGAAAGATGAGTGAGAGCATCAACTTCTCTCACAACCTAGGCCAGTAAGTAGTGCTT",
    "queryseq":  "ATTGGAGGGAGGGTGAACAAAGAGATAGACTTCTG--GCAACCTGGGCCAGTAGGTAGTGTCT"
}
```

Here is an example of the genome-align track:

```
chr1    12177   12240   id:1,genomealign:{chr:"chr4",start:154100819,stop:154100880,
↪strand:"-",targetseq:"ATTGGAGGAAAGATGAGTGAGAGCATCAACTTCTCTCACAACCTAGGCCAGTAAGTAGTGCTT",
```

(continues on next page)

(continued from previous page)

```
↪queryseq:"ATTGGAGGGAGGGTGAACAAAGAGATAGACTTCTG--GCAACCTGGGCCAGTAGGTAGTGTCT"}
chr1    12245   12273   id:2,genomealign:{chr:"chr9",start:114130992,stop:114131016,
↪strand:"+",targetseq:"CATCTCCTTGGCTGTGATACGTGGCCGG",queryseq:"TGTCCCCTTGTCTGC----
↪CGGGGCTGG"}
```

AXT file can be generated by *lastz* or *blastz*. It is also possible to make genome alignment using minimap2, and sequentially convert minimap2 SAM output to AXT file. Here is our pipeline making hg38-CHM13 genome-align AXT file:

```
minimap2 -x asm5 --cs=long hg38.fa chm13.fa > hg38-chm13.paf
sort -k6,6 -k8,8n -k9,9n hg38-chm13.paf|perl -ln unique_paf.pl > hg38-chm13.unique.paf
paftools.js view -f maf hg38-chm13.unique.paf > hg38-chm13.maf
maf-convert axt hg38-chm13.maf > hg38-chm13.axt
python3 axtSplit.py 100 hg38-chm13.axt hg38-chm13.split.axt
```

Note we used two custom script *unique_paf.pl* and *axtSplit.py* to remove redundant segments in the alignment and split long alignment records to smaller ones separated by gaps > 100bp. You can find them in the scripts directory: (https://github.com/lidaof/eg-react/blob/master/backend/scripts).

AT last, we have a script to convert AXT file to genome-align format, you can find it in the scripts directory: (https://github.com/lidaof/eg-react/blob/master/backend/scripts/axt2align.py).

Your text file must be sorted by the first three columns. If your filename is example.qbed, you can sort it with the following command: `sort -k1V -k2n -k3n example.genomealign > example.sorted.genomealign`

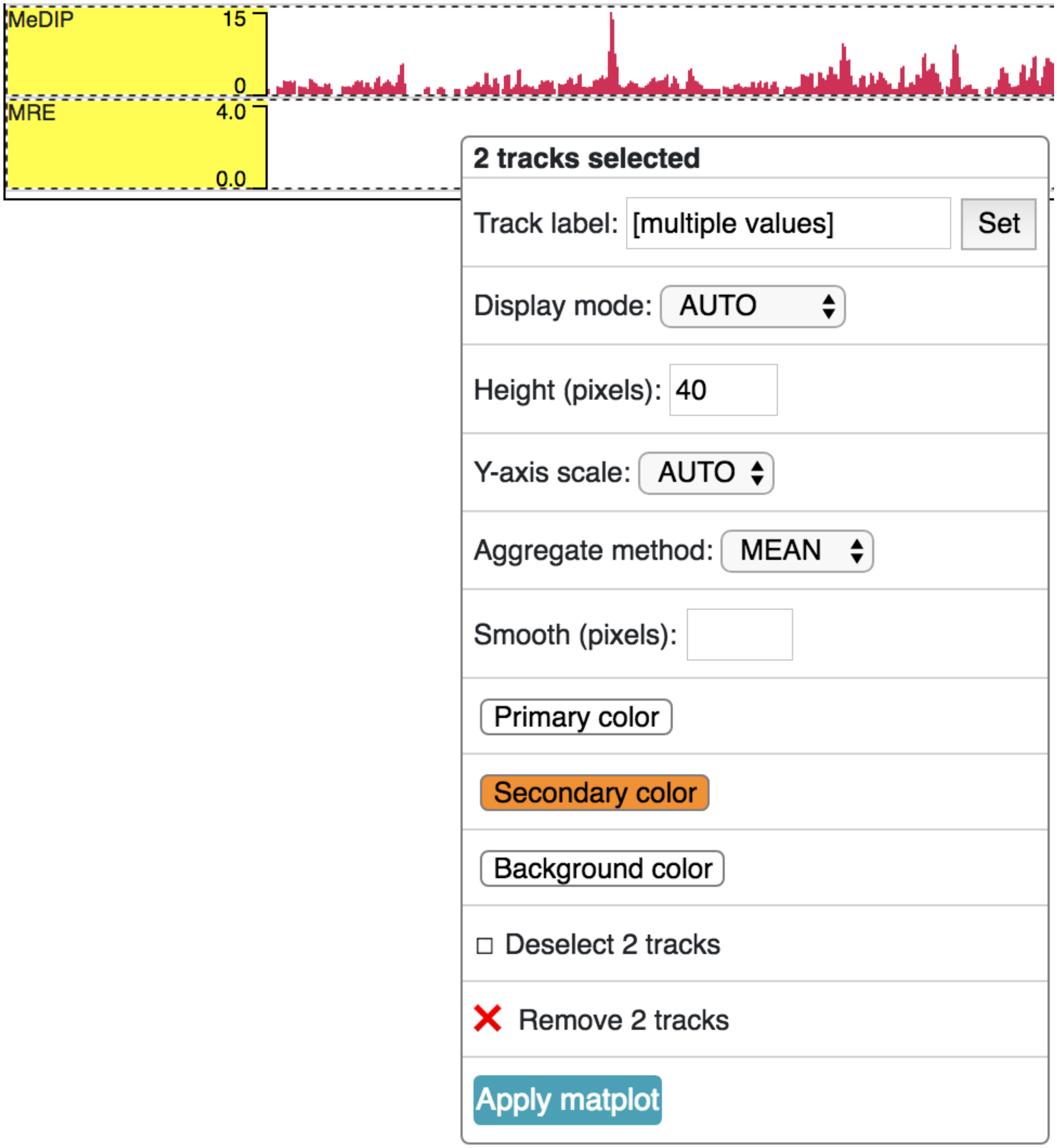
Place your sorted genome-align file in a web-accessible directory, then compress and index as follows:

```
bgzip example.sorted.genomealign
tabix -p bed example.sorted.genomealign.gz
```

2.14 Matplot Track

A matplot (also called a line plot) displays multiple numerical tracks on the same X and Y axes to easily compare datasets. Data is plotted as curves instead of bar plots.

To use matplot, choose more than 1 numerical tracks:



2 tracks selected

Track label: [multiple values]

Display mode: AUTO

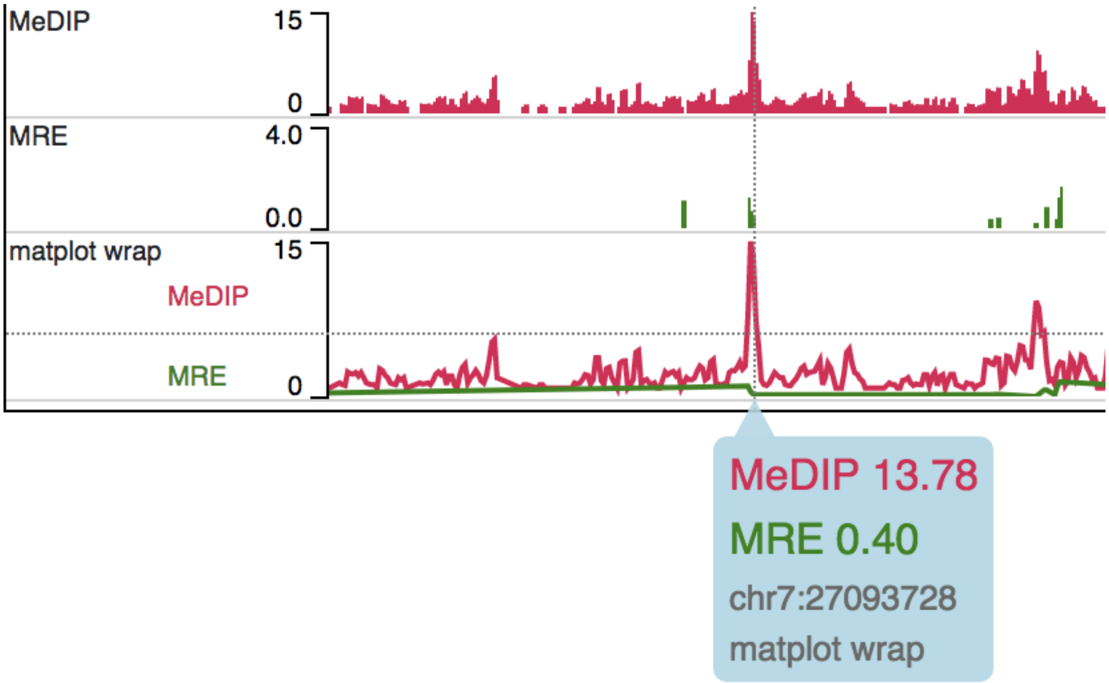
Height (pixels): 40

Y-axis scale: AUTO

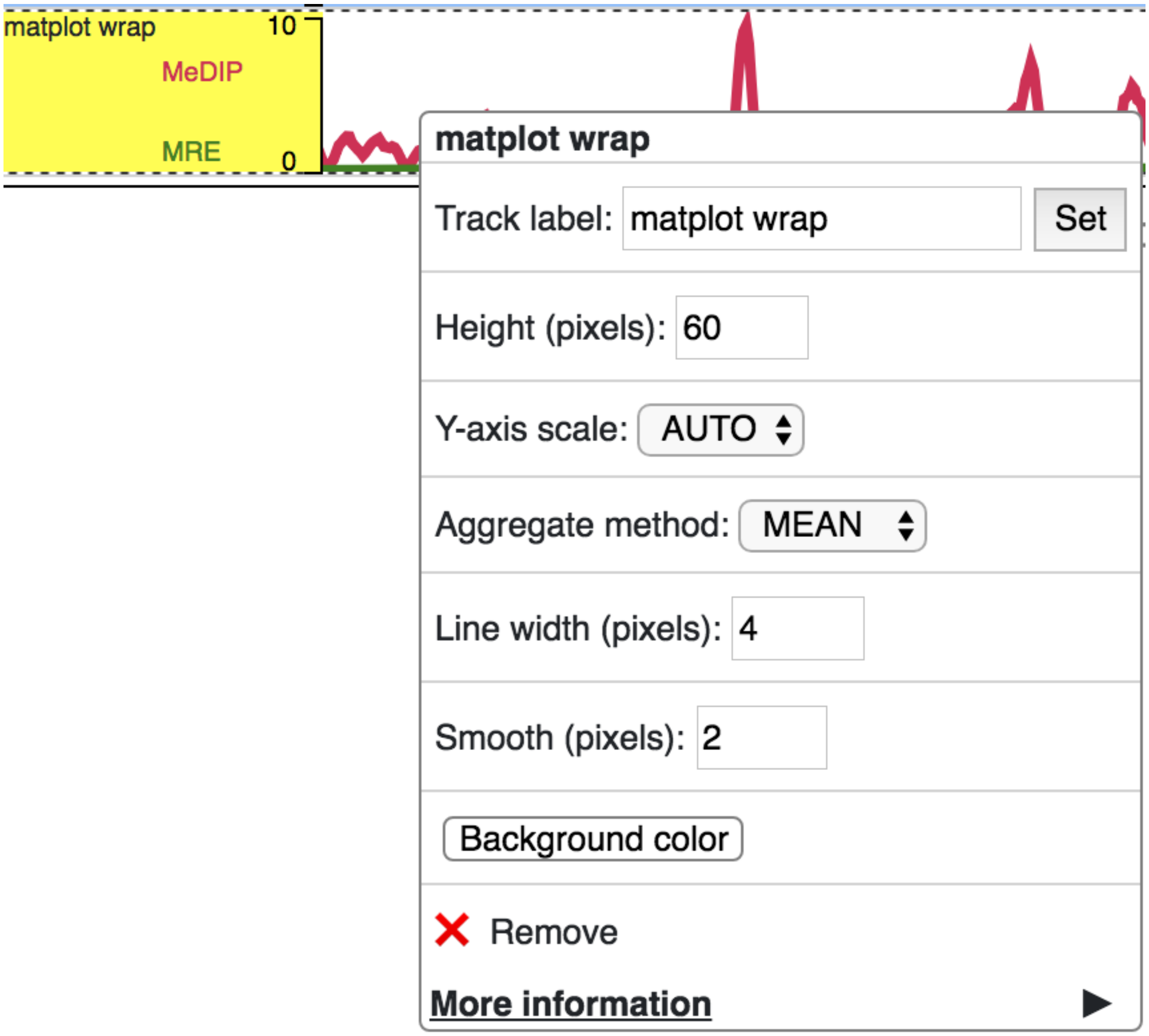
Aggregate method: MEAN

Smooth (pixels):

Right click, and choose *Apply matplot* button, The new matplot track will be shown:



and it also supports many configurations:



matplot wrap

Track label: matplot wrap

Height (pixels): 60

Y-axis scale: AUTO

Aggregate method: MEAN

Line width (pixels): 4

Smooth (pixels): 2

CHAPTER

THREE

DATAHUB

A datahub is a JSON file descibing a set of tracks in the browser. A datahub file is an array of tracks, which are also defined in JSON syntax:

```
[
    {
    "type": "track_type1",
    "name": "track_name1",
    "url": "track_url1",
    "showOnHubLoad": true,
    "options": {
        "color": "red"
        }
    },
    {
    "type": "track_type2",
    "name": "track_name2",
    "url": "track_url2",
    "showOnHubLoad": true,
    "options": {
        "color": "blue"
        }
    }
]
```

Important: For each track in datahub, `showOnHubLoad` need set to `true` for the track to be displayed in browser. Tracks without `showOnHubLoad` set to `true` won't be displayed in browser but can be added later in track facet table.

3.1 Example data hub

Pasted below is an example data hub for mouse mm10:

```
[
    {
    "type": "bigwig",
    "url": "https://vizhub.wustl.edu/public/tmp/TW463_20-5-bonemarrow_MeDIP.bigWig",
    "name": "MeDIP",
    "options": {
```

(continues on next page)

(continued from previous page)

```
        "color": "red",
        "backgroundColor":"#FFE7AB"
        },
    "metadata": {
        "sample": "bone",
        "assay": "MeDIP"
        }
    },
    {
    "type": "bigwig",
    "url": "https://vizhub.wustl.edu/public/tmp/TW551_20-5-bonemarrow_MRE.CpG.bigWig",
    "name": "MRE",
    "options": {
        "color": "blue",
        "backgroundColor":"#C0E3CC"
        },
    "metadata": {
        "sample": "bone",
        "assay": "MRE"
        }
    }
]
```

3.2 Example bigWig track

```
{
    "type": "bigwig",
    "name": "example bigwig",
    "url": "https://vizhub.wustl.edu/hubSample/hg19/GSM429321.bigWig",
    "options": {
        "color": "blue"
    }
}
```

3.3 Example dynseq track

```
{
    "type": "dynseq",
    "name": "example dynseq",
    "url": "https://target.wustl.edu/dli/tmp/deeplift.example.bw",
    "options": {
        "color": "blue",
        "height: 100
    }
}
```

3.4 Example methylC track

```
{
  "type": "methylc",
  "name": "H1",
  "url": "https://vizhub.wustl.edu/public/hg19/methylc2/h1.liftedtohg19.gz",
  "options": {
    "label": "Methylation",
    "colorsForContext": {
      "CG": { "color": "#648bd8", "background": "#d9d9d9" },
      "CHG": { "color": "#ff944d", "background": "#ffe0cc" },
      "CHH": { "color": "#ff00ff", "background": "#ffe5ff" }
    },
    "depthColor": "#01E9FE"
  },
}
```

3.5 Example categorical track

```
{
  "type": "categorical",
  "name": "ChromHMM",
  "url": "https://egg.wustl.edu/d/hg19/E017_15_coreMarks_dense.gz",
  "options": {
      "category": {
          "1": {"name": "Active TSS", "color": "#ff0000"},
          "2": {"name": "Flanking Active TSS", "color": "#ff4500"},
          "3": {"name": "Transcr at gene 5' and 3'", "color": "#32cd32"},
          "4": {"name": "Strong transcription", "color": "#008000"},
          "5": {"name": "Weak transcription", "color": "#006400"},
          "6": {"name": "Genic enhancers", "color": "#c2e105"},
          "7": {"name": "Enhancers", "color": "#ffff00"},
          "8": {"name": "ZNF genes & repeats", "color": "#66cdaa"},
          "9": {"name": "Heterochromatin", "color": "#8a91d0"},
          "10": {"name": "Bivalent/Poised TSS", "color": "#cd5c5c"},
          "11": {"name": "Flanking Bivalent TSS/Enh", "color": "#e9967a"},
          "12": {"name": "Bivalent Enhancer", "color": "#bdb76b"},
          "13": {"name": "Repressed PolyComb", "color": "#808080"},
          "14": {"name": "Weak Repressed PolyComb", "color": "#c0c0c0"},
          "15": {"name": "Quiescent/Low", "color": "#ffffff"}
      }
  }
}
```

Supported options: *backgroundColor*, *color*, *color2*, *yScale*, *yMax*, and *yMin*.

3.6 Example longrange track

```
{
    "type": "longrange",
    "name": "ES-E14 ChIA-PET",
    "url": "https://egg.wustl.edu/d/mm9/GSE28247_st3c.gz"
}
```

3.7 Example bigInteract track

```
{
    "type": "biginteract",
    "name": "test bigInteract",
    "url": "https://epgg-test.wustl.edu/dli/long-range-test/interactExample3.inter.bb"
}
```

3.8 Example repeatmasker track

```
{
    "type": "repeatmasker",
    "name": "RepeatMasker",
    "url": "https://vizhub.wustl.edu/public/mm10/rmsk16.bb"
}
```

3.9 Example geneAnnotation track

```
{
    "type": "geneAnnotation",
    "name": "refGene",
    "genome": "mm10"
}
```

Note: Please specify the `genome` attibute for gene annotation tracks.

3.10 Example bigbed track

```
{
    "type": "bigbed",
    "name": "test bigbed",
    "url": "https://vizhub.wustl.edu/hubSample/hg19/bigBed1"
}
```

3.11 Example bed track

```
{
    "type": "bed",
    "name": "mm10 bed",
    "url": "https://epgg-test.wustl.edu/d/mm10/mm10_cpgIslands.bed.gz"
}
```

3.12 Example refbed track

```
{
    "type": "refbed",
    "name": "mm10 gencode basic",
    "url": "https://vizhub.wustl.edu/public/tmp/gencodeM18_load_basic_Gene.bed.gz",
    "options": {
            "categoryColors": {
                "coding": "rgb(101,1,168)",
                "nonCoding": "rgb(1,193,75)",
                "pseudo": "rgb(230,0,172)",
                "problem": "rgb(224,2,2)",
                "other":"rgb(128,128,128)"
            }
        }
}
```

Note: `categoryColors` designates colors for the gene type as indicated in the 9th column. The default scheme is shown above for the five classes created by the `Converting_Gencode_or_Ensembl_GTF_to_refBed.bash` script, but any number of categories can be defined.

3.13 Example HiC track

```
{
    "type": "hic",
    "name": "test hic",
    "url": "https://epgg-test.wustl.edu/dli/long-range-test/test.hic",
    "options": {
        "displayMode": "arc"
    }
}
```

3.14 Example cool track

```
{
    "type": "cool",
    "name": "Aiden et al. (2009) GM06900 HINDIII 1kb",
    "url": "Hyc3TZevQVm3FcTAZShLQg",
    "options": {
        "displayMode": "arc"
    }
}
```

Note: please note we are using the uuid `Hyc3TZevQVm3FcTAZShLQg` here from higlass API server instead of a file URL to represent a cool track.

3.15 Example genomealign track

```
{
    "name": "hg19 to mm10 alignment",
    "type": "genomealign",
    "metadata": {
        "genome": "mm10"
    }
}
```

3.16 Example qBED track

```
{
    "type":"qbed",
    "url":"https://htcf.wustl.edu/files/RdNgrGeQ/HCT116-PBase.qbed.gz",
    "name":"piggyBac insertions",
    "showOnHubLoad":"true",
    "options":{
        "color":"#D12134",
```

(continues on next page)

(continued from previous page)

```
        "height":100,
        "logScale":"log10",
        "show":"sample",
        "sampleSize":1000,
        "markerSize":5,
        "opacity":[50],
      },
}
```

Note: Default qBED track options are `"logScale":"none"`, `"show":"all"`, `"markersize":3`, and `"opacity":[100]`. Log-scaling can be set by specifying `"logScale":"log10"`. To change opacity, pass a single value in brackets, as in the above example. qBED tracks will, by default, plot all entries in view. For information-dense regions, this can lead to excessive memory consumption. To plot a random subsample instead, specify `"show":"sample"` and pass the number of points to visualize to `"sampleSize"`, e.g. `"sampleSize":1000`

3.17 Example matplot track

```
{
    "type": "matplot",
    "name": "matplot wrap",
    "tracks": [
        {
        "type": "bigwig",
        "url": "https://vizhub.wustl.edu/public/tmp/TW463_20-5-bonemarrow_MeDIP.bigWig",
        "name": "MeDIP",
        "options": {
            "color": "red",
            "backgroundColor":"#FFE7AB"
            },
        "metadata": {
            "sample": "bone",
            "assay": "MeDIP"
            }
        },
        {
        "type": "bigwig",
        "url": "https://vizhub.wustl.edu/public/tmp/TW551_20-5-bonemarrow_MRE.CpG.bigWig
↪",
        "name": "MRE",
        "options": {
            "color": "blue",
            "backgroundColor":"#C0E3CC"
            },
        "metadata": {
            "sample": "bone",
            "assay": "MRE"
            }
        }
```

(continues on next page)

(continued from previous page)

```
    ]
}
```

3.18 Example g3d track

```
{
    "type": "g3d",
    "url": "https://wangftp.wustl.edu/~dli/tmp/test.g3d",
    "name": "example 3d track",
    "showOnHubLoad": true
}
```

3.19 Example Ruler track

```
{
    "type": "ruler",
    "name": "Ruler"
}
```

3.20 Track properties

3.20.1 type

Requried. `type` specifies the track type, currently supported track types:

- bigWig
- bedGraph
- dynseq
- methylC
- categorical
- hic
- bed
- bigbed
- refbed
- repeatmasker
- geneAnnotation
- genomealign
- longrange
- bigInteract

- qBED
- matplot
- snp
- ruler

Note: `type` is case insensitive.

3.20.2 name

Requried. `name` specifies the track name used internally by the browser. It is also displayed as the track legend if no *label* speficied. Value can be any string.

3.20.3 label

Optional. `label` specifies the track legend displayed in the browser. It overrides the *name* arrtibute. Value can be any string.

3.20.4 url

Requried. `url` contains the URL to the track file and needs to be HTTP or HTTPS location string.

Important: A `url` is requried for all the tracks in binary format. Gene annotaion tracks, like `refGene`, do not need a `url` as they are stored in the Mongo database. Additional annotation tracks, such as the `ruler` track, also do not need a `url`.

Caution: Each user-provided `url` must link to a publically available website, without password protection, so that the browser can read in the file.

Note: `url` can use a relative child path to the datahub url, say you have a file `a.bigWig` with your datahub `http://your.host/your.hub.json`, when you add the track entry for `a.bigWig`, the `url` can be either `http://your.host/a.bigWig` or just `a.bigWig`.

3.20.5 showOnHubLoad

Optional. If specified to `true`, the track will be displayed when hub is loaded. Default value: `false`.

3.20.6 metadata

Optional. An object specifying the metadata of the track.

In this basic example the value of each metadata term is a **string**.

```
"metadata": {
    "sample": "bone",
    "assay": "MRE"
}
```

You can also use this syntax for customized color:

```
"sample": {"name"" "bone", "color": "#FF0000"}
```

The value of color can also be “red”, “blue”, and any CSS color.

This example public Roadmap data hub has more complex metadata definitions and makes use of a **list of strings** to build a *hierarchical structure*.

```
{
    "url": "https://egg.wustl.edu/d/hg19/GSM997242_1.bigWig",
    "metadata": {
        "Sample": [
            "Adult Cells/Tissues",
            "Blood",
            "Other blood cells",
            "CD4+_CD25-_Th_Primary_Cells"
        ],
        "Donor": [
            "Donor Identifier",
            "Donor_332"
        ],
        "Assay": [
            "Epigenetic Mark",
            "Histone Mark",
            "H3",
            "H3K9",
            "H3K9me3"
        ],
        "Institution": [
            "Broad Institute"
        ]
    },
    "type": "bigwig",
    "options": {
        "color": "rgb(159,0,72)"
    },
    "name": "H3K9me3 of CD4+_CD25-_Th_Primary_Cells"
}
```

The list of metadata is ordered from more generic to more specific and helps build the facet table hierarchy making the **search** and **filter** functions in track table easier.

3.20.7 details

Optional. If you want to add more information for each track then the `details` attribute is helpful. After right clicking on the track you can click **More Information** and see the `details`, `url`, and `metadata` for each track in the dropdown menu.

```
"details": {
    "data source": "Roadmap Project",
    "date collected": "May 7 2016"
}
```

3.20.8 queryEndPoint

Optional. Most time this parameter will be used with geneAnnotation track. When users deal with custom genome, or genome not listed in NCBI or Ensembl database, gene search link would not work, so in such case, user can specify a custom database to query detailed information. For example, for some trypanosome genome, gene search should be queried through TriTryDB, we can define the track like this then:

```
{
    type: "geneAnnotation",
    name: "gene",
    label: "TriTrypDB genes",
    genome: "TbruceiLister427",
    queryEndpoint: { name: "TriTrypDB", endpoint: "https://tritrypdb.org/tritrypdb/app/
↪search?q=" },
}
```

3.20.9 options

Optional. All track render options are placed in an object called `options`. This object can have the following properties:

color

`color` is used to define the color for each track. A color name, RGB values, or hex color code can be used. For more about color name or RGB please see https://www.w3schools.com/css/css_colors.asp.

color2

`color2` is used to define the color for negative values from the track data. The default is the same as *color*.

backgroundColor

`backgroundColor` defines the background color of the track.

height

`height` controls the height of the track which is specified as a number and displayed in *pixels*.

ensemblStyle

currently for *bigwig* track, specify `ensemblStyle` option to *true* can enable data with chromosome names as 1, 2, 3…work in the browser

yScale

`yScale` allows you to configure the track's y-scale. Options include *auto* or *fixed*. *auto* sets the y-scale from 0 to the max value of values in the view region for a given track. *fixed* means you can specify the *minimal* and *maximal* value.

yMax

`yMax` is used to define the *maximum* value of a track's y-axis. Value is number.

yMin

`yMin` is used to define the *minimum* value of a track's y-axis. Value is number.

Important: If you need the track to be in *fixed* scale, you need to specify `yScale` to *fixed* besides of set `yMax` and `yMin`.

group

Numerical tracks can be grouped to same group, tracks from same group will share y-axis scale settings. For example, when 2 tracks are in one group, the y-axis will both set to max value of current views of both tracks. Users can find one example data hub with `group` settings from here: https://wangftp.wustl.edu/~dli/test/a-group.json

scoreScale/scoreMax/scoreMin

These options work similar as yScale/yMax/yMin, but these are for interaction tracks.

colorAboveMax

`colorAboveMax` defines the color displayed when a *fixed* *yScale* is used and a value exceeds the *yMax* defined.

color2BelowMin

`color2BelowMin` defines the color displayed when a *fixed* *yScale* is used and a value is below the *yMin* defined.

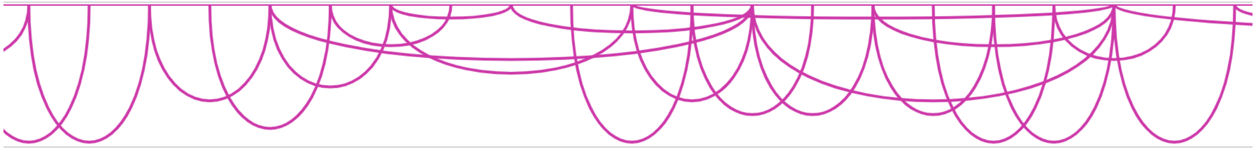
displayMode

`displayMode` specifies display mode for each tracks. Different tracks have different display modes as listed below.

type	display mode
bigWig	*auto*, *bar*, *heatmap*
bedGraph	*auto*, *bar*, *heatmap*
geneAnnotation	*full*, *density*
HiC	*arc*, *heatmap*, *flatarc*, *square*
genomealign	*rough*, *fine*

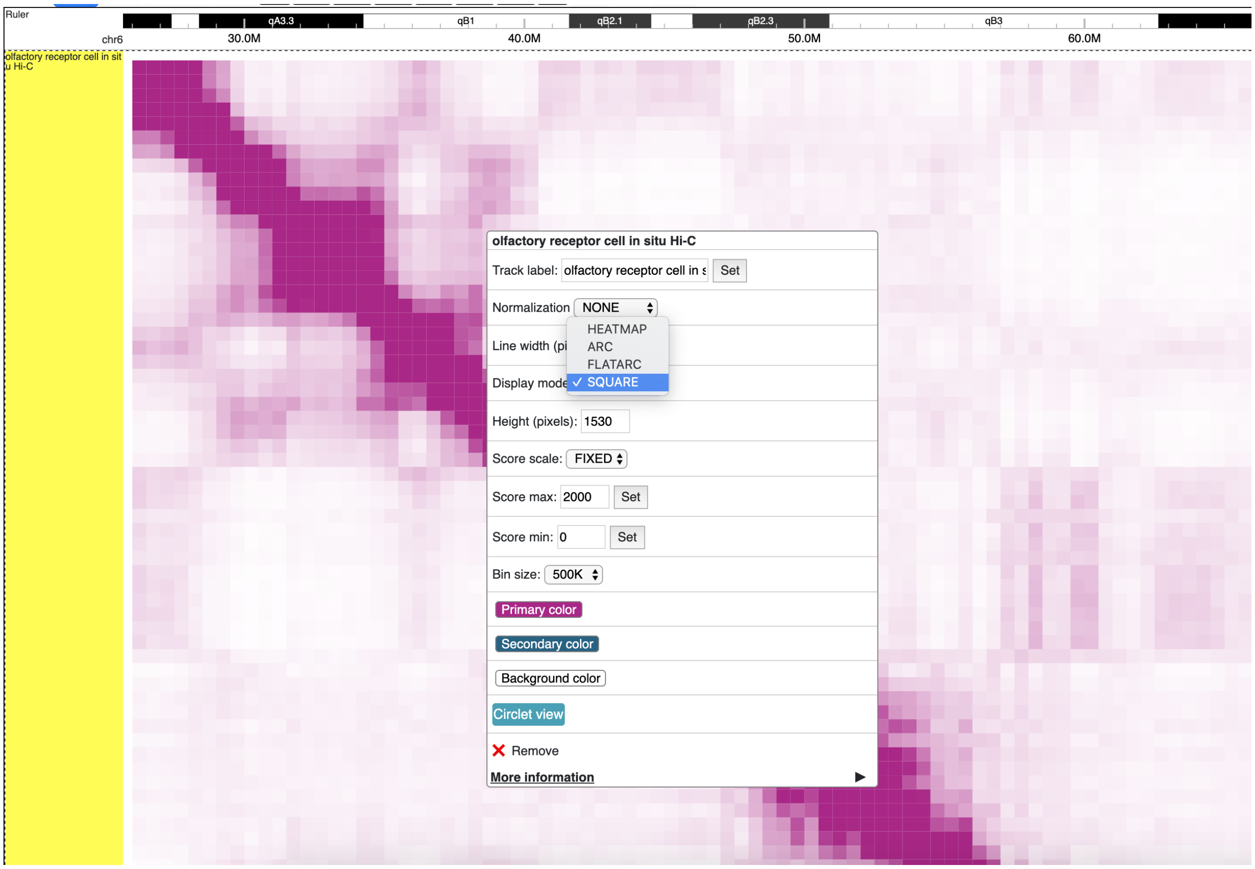
flatarc mode

For interaction track. `flatarc` mode is like `arc` mode, sometimes the curve would be displayed flatter, in fact it's a cubic curve.



square mode

For interaction track. `square` mode gives JuiceBox style like view for HiC maps.



Track label: olfactory receptor cell in s

Normalization NONE

Display mode ✓ SQUARE

Height (pixels): 1530

Score scale: FIXED

Score max: 2000

Score min: 0

Bin size: 500K

aggregateMethod

At high zoom-out level when 1 on-screen pixel spans >1bp, the underlying track data needs to be summarized into a single value for browser display. `aggregateMethod` is used to control how the data is summarized. Supported values include: `MEAN`, `SUM`, `COUNT`, `MAX`, `MIN`. Default value is `MEAN`.

smooth

`smooth` option allows you to smooth the graph of a quantitative track using window mean values. The browser will use the mean values from region [current_position - smooth, current_position + smooth]. Default value is 0 (no smooth applied).

maxRows

`maxRows` options controls the number of rows for the annotation track, like a geneAnnotation track.

hiddenPixels

For annotation tracks, when an element spans less than *hiddenPixels* in the screen, this item will not be displayed. Default value is 0.5 pixel. Set to 0 will display all elements.

isCombineStrands

For methylC tracks, `isCombineStrands` will specificy if the strands should be combined `true` or not combined `false`. We recommend combining stands for viewing CpG methylation, but leaving strand information for non-CpG methylation.

depthFilter

For methylC tracks a `depthFilter` can be set to filter out any bases with less than the depth(coverage) specified.

depthColor

For methylC tracks specify a `depthColor` for the depth line that overlays the bars.

maxMethyl

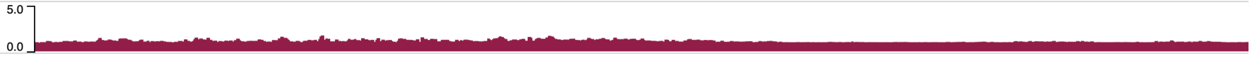
For methylC tracks specify the y-axis max (for both strands) using `maxMethyl`. Options range from (0-1].

zoomLevel

For `bigWig` track only. `bigWig` files usually contain multiple resolutions, when viewing a large region, the Browser usually fetches a lower resolution for faster response, user can change this behaviour by changing this option.

The example below first show viewing a bigWig track in a big region with `AUTO` zoom level, you can see the data is pretty flat, when we change zoom level to 0, 1, etc, we can see more details from the data, but takes more time to load.

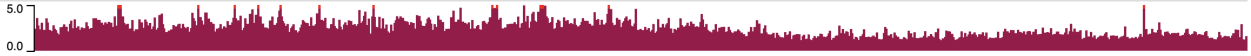
Automatical zoom level:



Right click, change zoom level to 0: (can also setup in data hub under `options`)

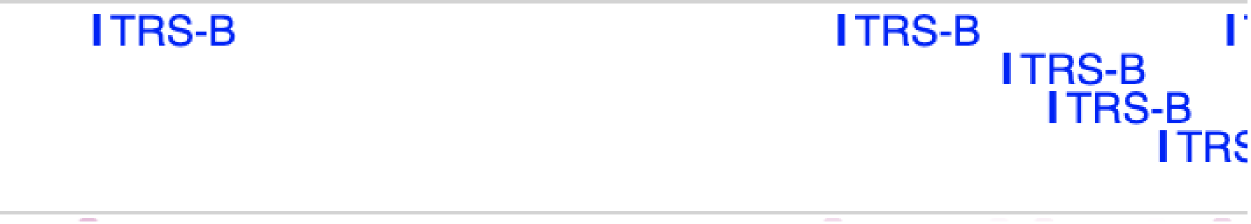
Zoom level: 0

View changed after change zoom level to 0:



alwaysDrawLabel

For `bed` and `categorical` tracks only. Usually for each `bed` and ``categorical` item in those tracks, the label are only drawn only when there are enough space inside the item block, by specificy this option to *true*, the label will always be drawn in the screen.



CHAPTER

FOUR

URL PARAMETERS

4.1 genome

Specify the genome in URL. It’s required for all other URL parameters.

Example: `?genome=hg19`

4.2 hub

Specify a data hub URL in JSON format.

Example: `?genome=hg19&hub=https://vizhub.wustl.edu/public/tmp/a.json`

4.3 bundle

Specify a session bundle ID in URL.

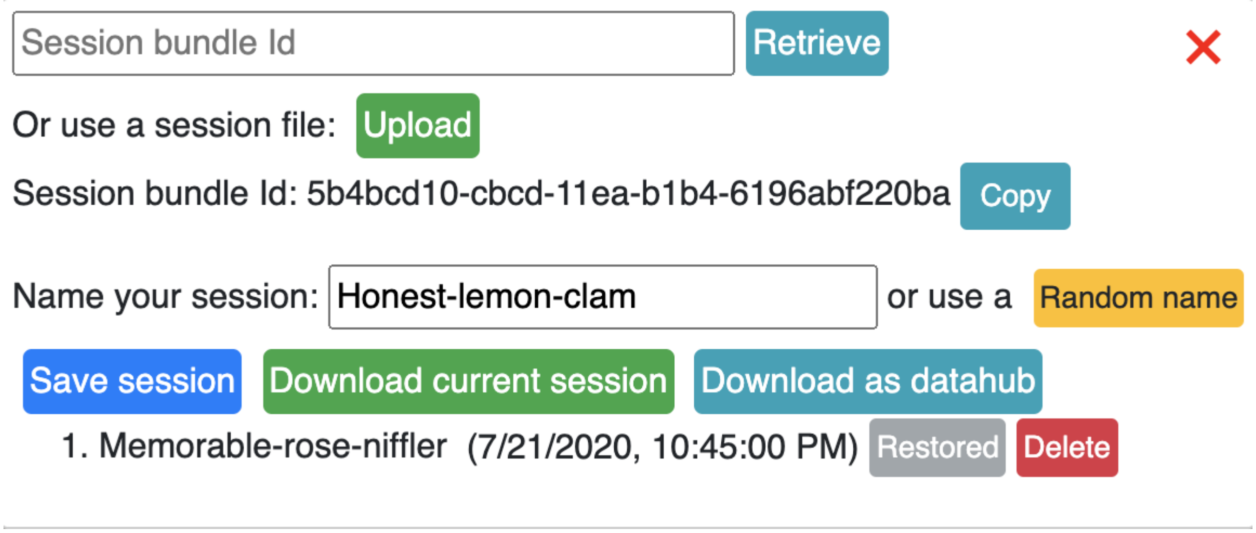
Example: `?genome=hg19&bundle=session-bundle-id`

4.4 sessionFile

Specify a session file in URL.

Example: `?sessionFile=https://wangftp.wustl.edu/~dli/test/eg-session--1692c5f0-c392-11e9-829c-912864922e1`
`json`

Note: `sessionFile` can be downloaded using the `Download current session` button in Session user interface.



Session bundle Id

Or use a session file:

Session bundle Id: 5b4bcd10-cbcd-11ea-b1b4-6196abf220ba

Name your session: Honest-lemon-clam or use a

1. Memorable-rose-niffler (7/21/2020, 10:45:00 PM)

4.5 hicUrl

Specify an HiC track in URL.

Example: `?genome=hg19&hicUrl=https://your.url.to.hic.file`

4.6 position

Specify the default genomic position once the browser is loaded.

Example: `?genome=hg19&position=chr1:1000-2000`

4.7 noDefaultTracks

Remove the default tracks when load a data hub.

Example: `?genome=hg19&noDefaultTracks=1`

4.8 datahub

Redirects to the legacy browser.

Example: `?genome=hg19&datahub=https://your.url.to.datahub`

4.9 session

Redirects to the legacy browser.

Example: `?genome=hg19&session=legacy-browser-session-id`

CHAPTER

FIVE

LOCAL TRACK FILES

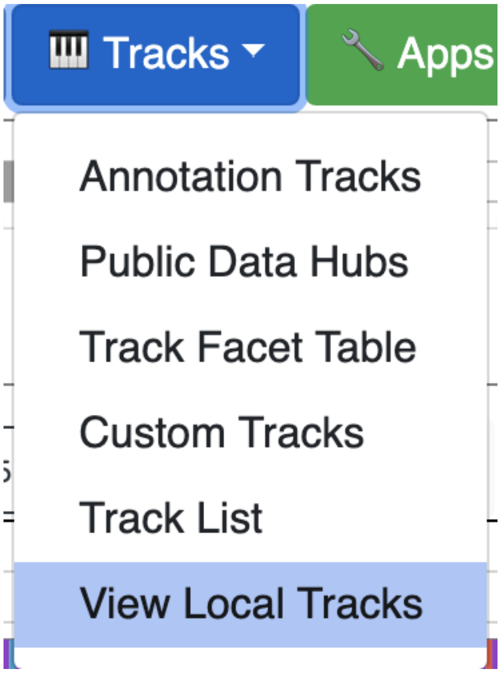
We realize that not every research group has the ability to set up a web server with CORS enabled. As of version 48.0.0, we added the ability to read local track files or an entire folder as a datahub. This means the user can **view** track files from their hard drive to the browser. The format for these track files is same as those that are hosted on a webserver.

Note: The tracks *viewed* through the local track feature *cannot* be saved to the browser's local storage. Thus, when you refresh the browser your local tracks will be gone. Tracks need to be re-selected or re-grant the browser permission to read local files. This is a security setup of Javascript. To avoid uploading multiple files repeatedly, the user can create a `hub.config.json` file to specify files as a local datahub. In this manner, after a refresh event the user can just choose their local datahub again instead of choosing individual tracks separately.

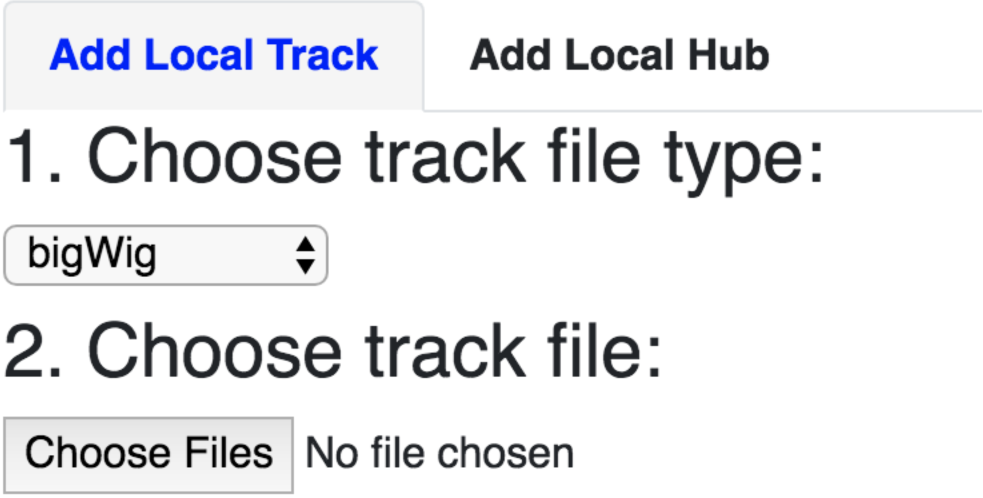
Important: Since the user needs to give permission for the web browser to access files on a local hard drive, tracks from a local hard drive **cannot** be saved into `Session`. Please consider using HTTP(S) hosted tracks to work with the session function.

5.1 View local files as tracks

To use your local files, make sure to format your files correctly. First, open the `View Local Tracks` menu from `Tracks`:



By default, you will be on the `Add Local Track` tab. Second, choose your track file format:



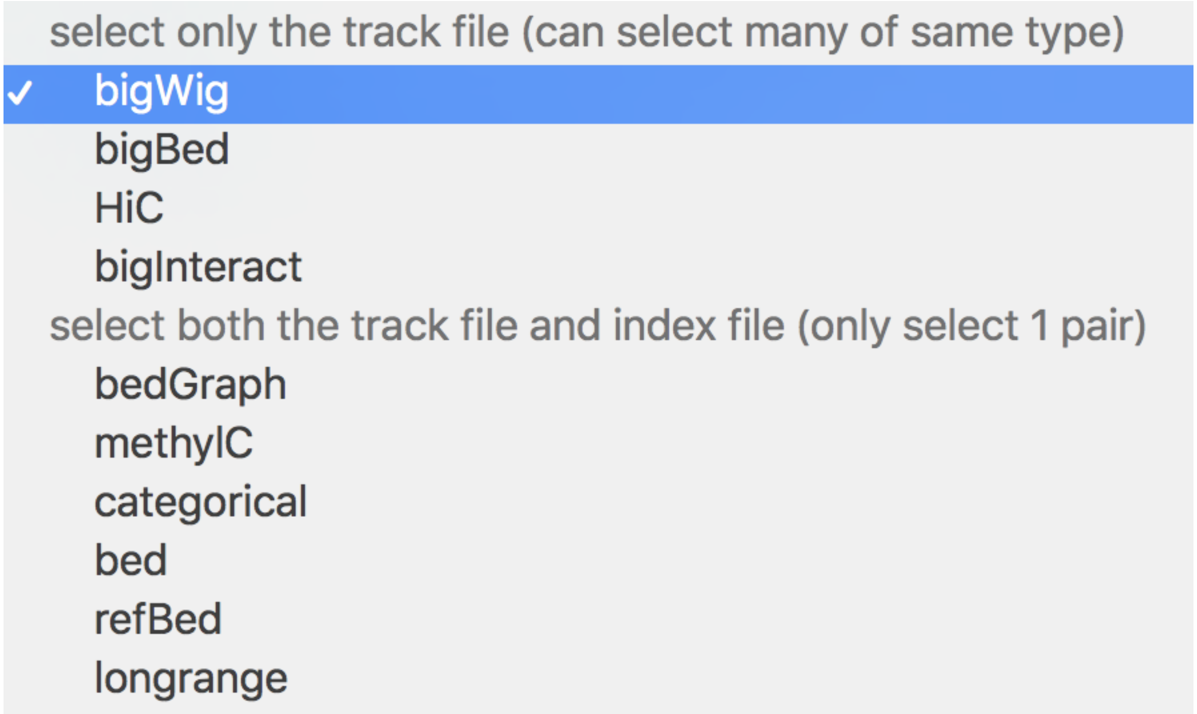
1. Choose track file type:

bigWig

2. Choose track file:

No file chosen

Third, choose your files. You can choose many files of same type if the track type only requires one file (`bigWig`, `bigBed`, `HiC`, and `bigInteract`) or if the track type requires a data file and **index** file (`bedGraph`, `methylC`, etc.) then you need to choose *each pair* individually.



select only the track file (can select many of same type)

✓ bigWig

bigBed

HiC

bigInteract

select both the track file and index file (only select 1 pair)

bedGraph

methylC

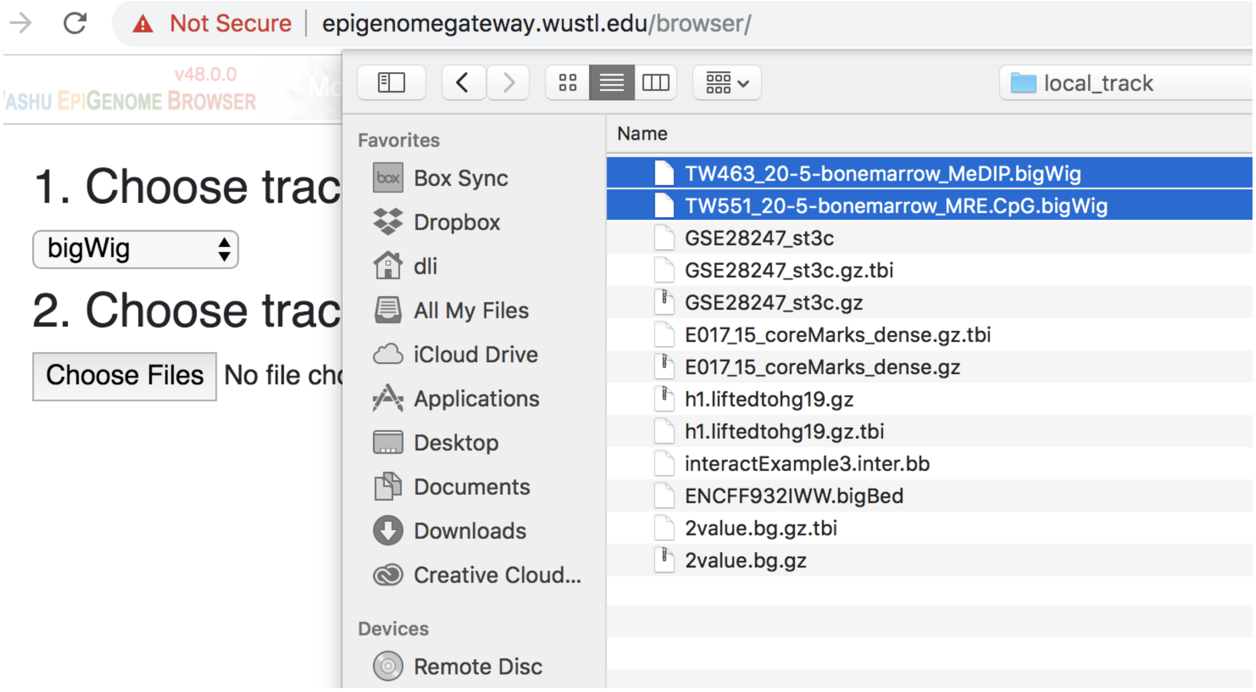
categorical

bed

refBed

longrange

Example upload of 2 local bigWig files:



TW463_20-5-bonemarrow_MeDIP.bigWig

TW551_20-5-bonemarrow_MRE.CpG.bigWig

GSE28247_st3c

GSE28247_st3c.gz.tbi

GSE28247_st3c.gz

E017_15_coreMarks_dense.gz.tbi

E017_15_coreMarks_dense.gz

h1.liftedtohg19.gz

h1.liftedtohg19.gz.tbi

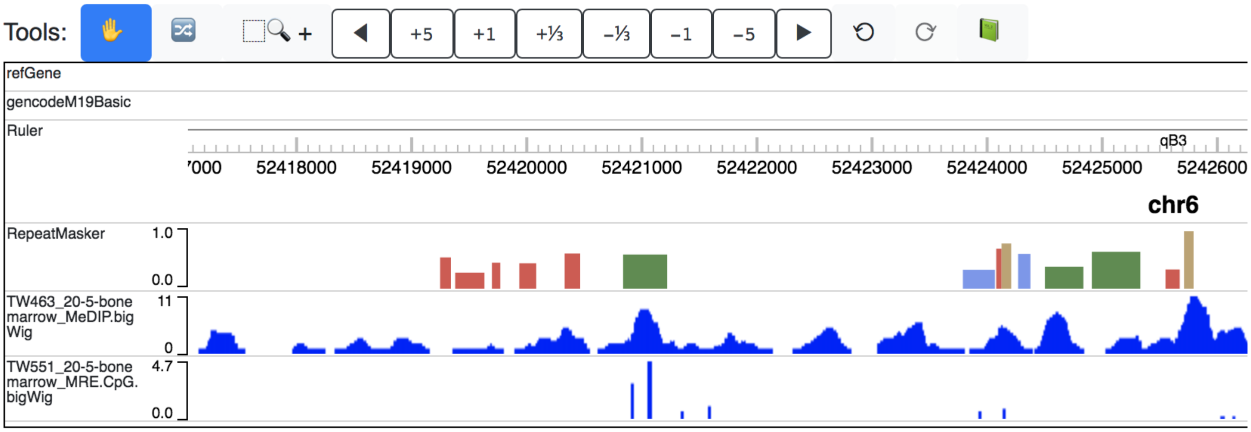
interactExample3.inter.bb

ENCFF932IWW.bigBed

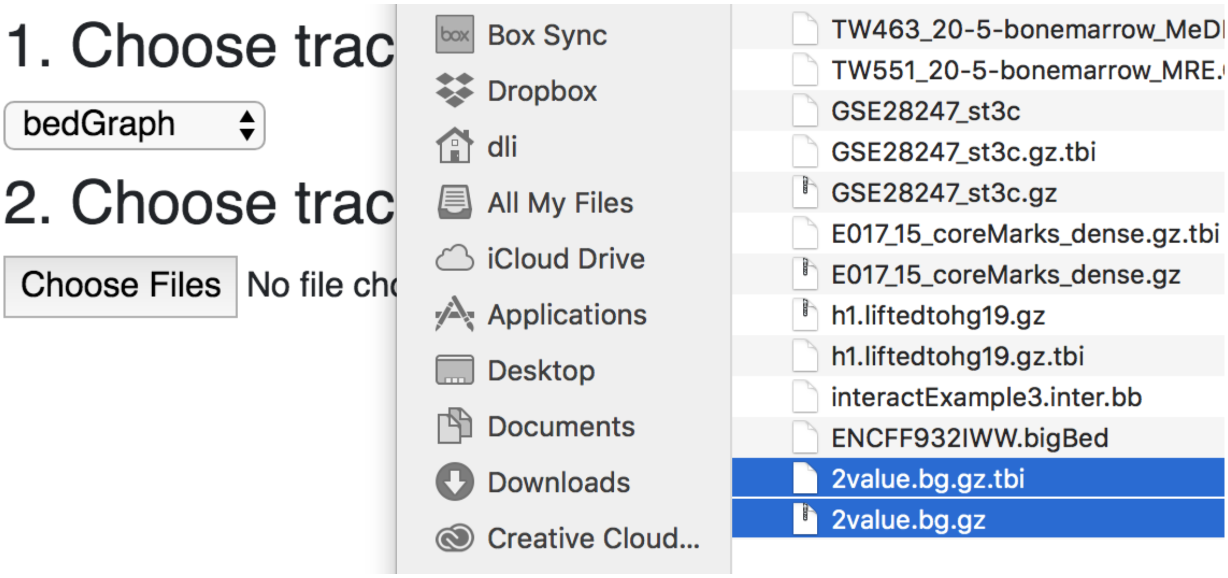
2value.bg.gz.tbi

2value.bg.gz

The 2 bigWig tracks are added to the browser view:



Example upload of 1 local Bedgraph track and its associated index file:

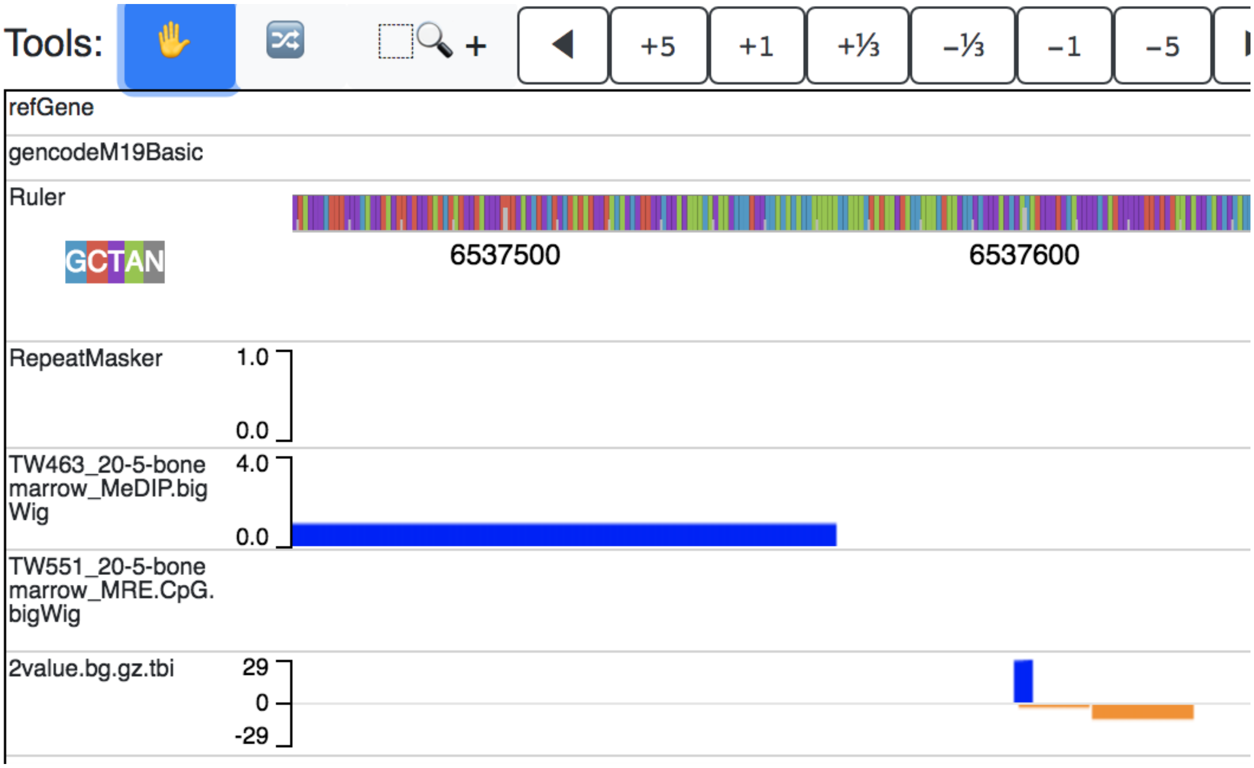

1. Choose trac

bedGraph

2. Choose trac

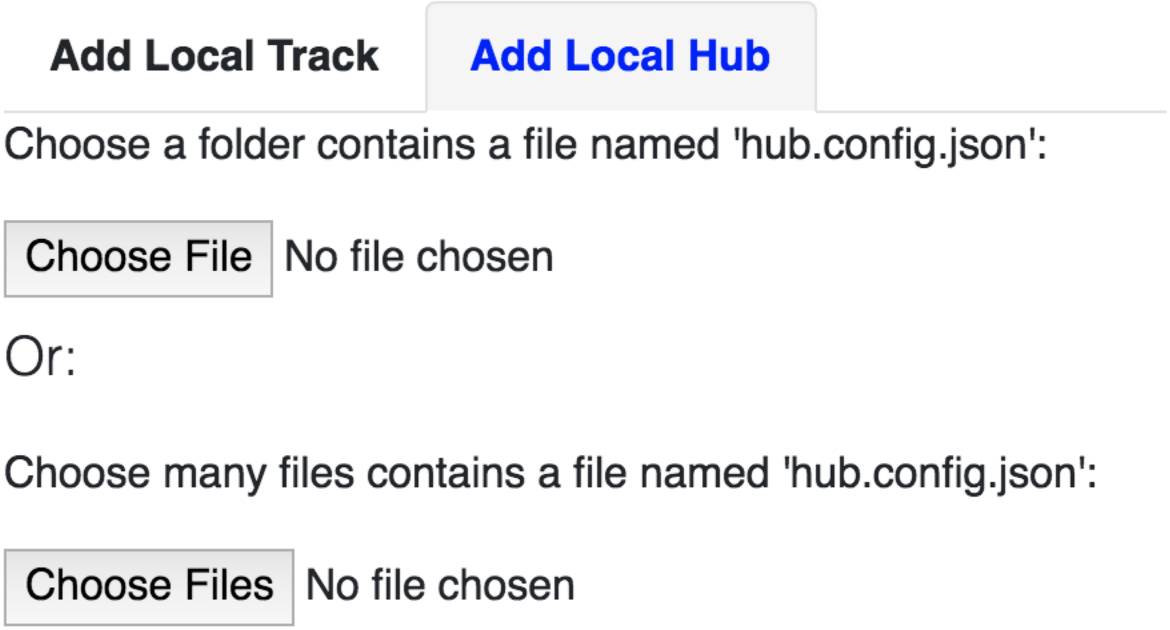
Choose Files No file ch

The bedGraph track is added to the browser view:



5.2 View files or a folder as a datahub

If you want to upload many different types of track files to the browser, you can do that too! Choose the `Add Local Hub` tab from the track upload menu as before:



Choose a folder contains a file named 'hub.config.json':

No file chosen

Or:

Choose many files contains a file named 'hub.config.json':

No file chosen

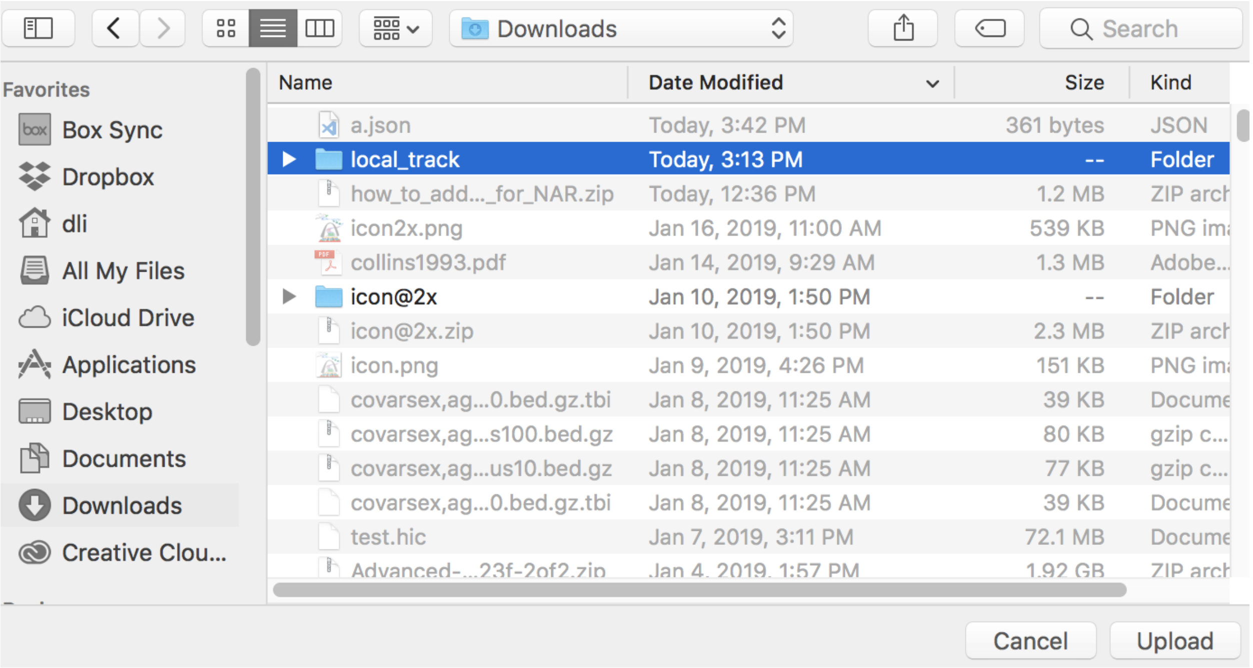
Create a file called `hub.config.json` for the browser to configure your local data hub (example below):

```
[
    {
        "filename": "2value.bg.gz",
        "type": "bedgraph",
        "name": "test bedgraph",
        "options": {"height": 100}
    },
    {
        "filename": "TW463_20-5-bonemarrow_MeDIP.bigWig",
        "type": "bigwig",
        "name": "MeDIP",
        "options": {"color": "pink"}
    },
    {
        "filename": "TW551_20-5-bonemarrow_MRE.CpG.bigWig",
        "type": "bigwig",
        "name": "MRE",
        "options": {"color": "red"}
    },
    {
        "filename": "h1.liftedtohg19.gz",
        "type": "methylc"
    }
]
```

Note: Please note the `name` and `options` attribute specified in this file. The syntax is the same as a remote datahub file.

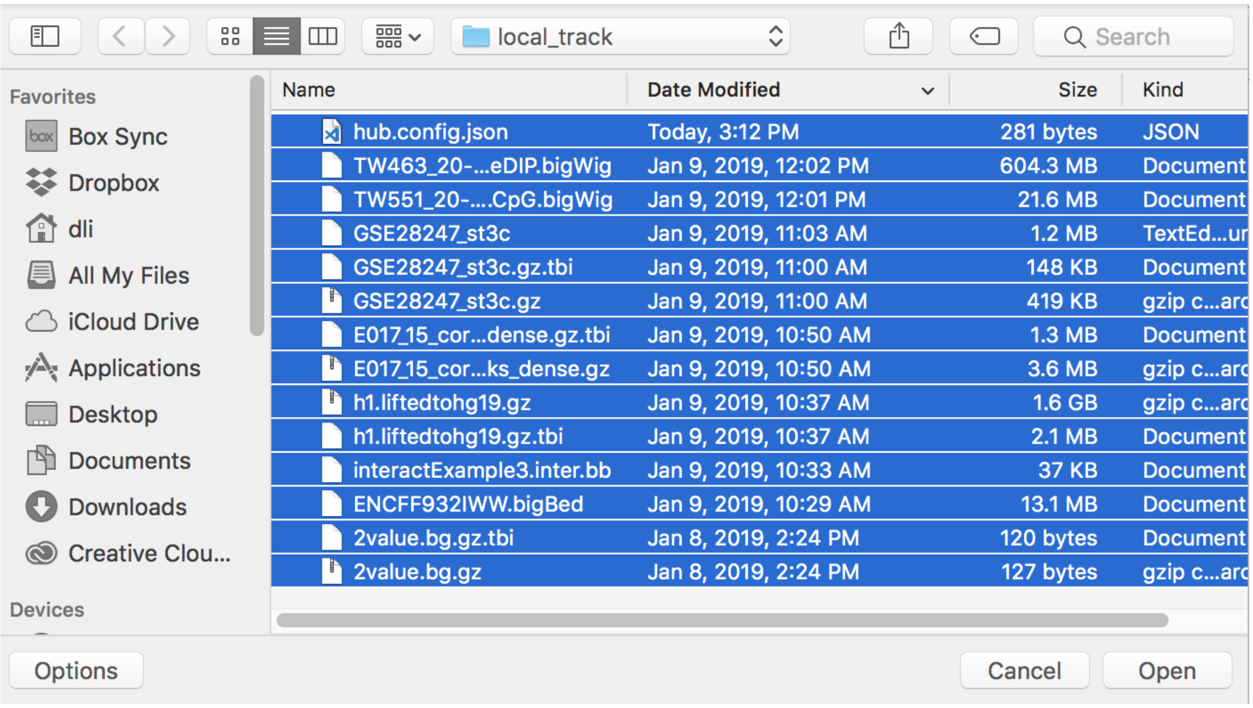
Note: Track files not specified in `hub.config.json` will be skipped. Orders of tracks will follow the order defined in `hub.config.json` file.

You can either choose an entire folder by clicking the first button:



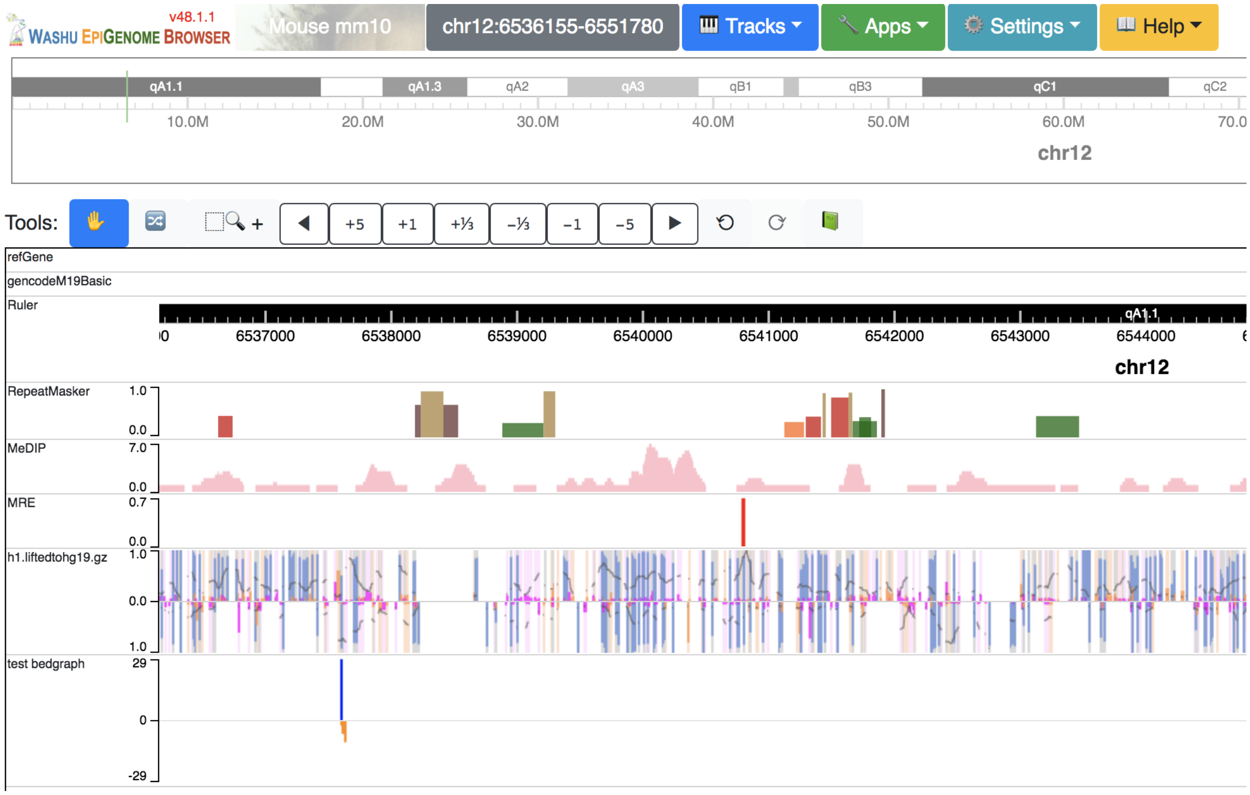
Name	Date Modified	Size	Kind
a.json	Today, 3:42 PM	361 bytes	JSON
local_track	Today, 3:13 PM	--	Folder
how_to_add..._for_NAR.zip	Today, 12:36 PM	1.2 MB	ZIP arch
icon2x.png	Jan 16, 2019, 11:00 AM	539 KB	PNG ima
collins1993.pdf	Jan 14, 2019, 9:29 AM	1.3 MB	Adobe...
icon@2x	Jan 10, 2019, 1:50 PM	--	Folder
icon@2x.zip	Jan 10, 2019, 1:50 PM	2.3 MB	ZIP arch
icon.png	Jan 9, 2019, 4:26 PM	151 KB	PNG ima
covarsex,ag...0.bed.gz.tbi	Jan 8, 2019, 11:25 AM	39 KB	Docume
covarsex,ag...s100.bed.gz	Jan 8, 2019, 11:25 AM	80 KB	gzip c...
covarsex,ag...us10.bed.gz	Jan 8, 2019, 11:25 AM	77 KB	gzip c...
covarsex,ag...0.bed.gz.tbi	Jan 8, 2019, 11:25 AM	39 KB	Docume
test.hic	Jan 7, 2019, 3:11 PM	72.1 MB	Docume
Advanced-...23f-2of2.zip	Jan 4, 2019, 1:57 PM	1.92 GB	ZIP arch

or choose many files by clicking the second button:



Name	Date Modified	Size	Kind
hub.config.json	Today, 3:12 PM	281 bytes	JSON
TW463_20-...eDIP.bigWig	Jan 9, 2019, 12:02 PM	604.3 MB	Document
TW551_20-....CpG.bigWig	Jan 9, 2019, 12:01 PM	21.6 MB	Document
GSE28247_st3c	Jan 9, 2019, 11:03 AM	1.2 MB	TextEd...ur
GSE28247_st3c.gz.tbi	Jan 9, 2019, 11:00 AM	148 KB	Document
GSE28247_st3c.gz	Jan 9, 2019, 11:00 AM	419 KB	gzip c...arc
E017_15_cor...dense.gz.tbi	Jan 9, 2019, 10:50 AM	1.3 MB	Document
E017_15_cor...ks_dense.gz	Jan 9, 2019, 10:50 AM	3.6 MB	gzip c...arc
h1.liftedtohg19.gz	Jan 9, 2019, 10:37 AM	1.6 GB	gzip c...arc
h1.liftedtohg19.gz.tbi	Jan 9, 2019, 10:37 AM	2.1 MB	Document
interactExample3.inter.bb	Jan 9, 2019, 10:33 AM	37 KB	Document
ENCFF932IWW.bigBed	Jan 9, 2019, 10:29 AM	13.1 MB	Document
2value.bg.gz.tbi	Jan 8, 2019, 2:24 PM	120 bytes	Document
2value.bg.gz	Jan 8, 2019, 2:24 PM	127 bytes	gzip c...arc

After uploading either a folder or many files, your local datahub will be displayed: (Please note the `name` and `options` specified in your `hub.config.json` file will also be applied)


v48.1.1
WASHU EPIGENOME BROWSER
Mouse mm10
chr12:6536155-6551780
Tracks
Apps
Settings
Help
qA1.1
qA1.3
qA2
qA3
qB1
qB3
qC1
qC2
10.0M
20.0M
30.0M
40.0M
50.0M
60.0M
70.0
chr12
Tools:
+
+5
+1
+⅓
−⅓
−1
−5
refGene
gencodeM19Basic
Ruler
6537000
6538000
6539000
6540000
6541000
6542000
6543000
6544000
chr12
RepeatMasker
1.0
0.0
MeDIP
7.0
0.0
MRE
0.7
0.0
h1.liftedtohg19.gz
1.0
0.0
1.0
test bedgraph
29
0
-29

CHAPTER

SIX

TEXT TRACKS

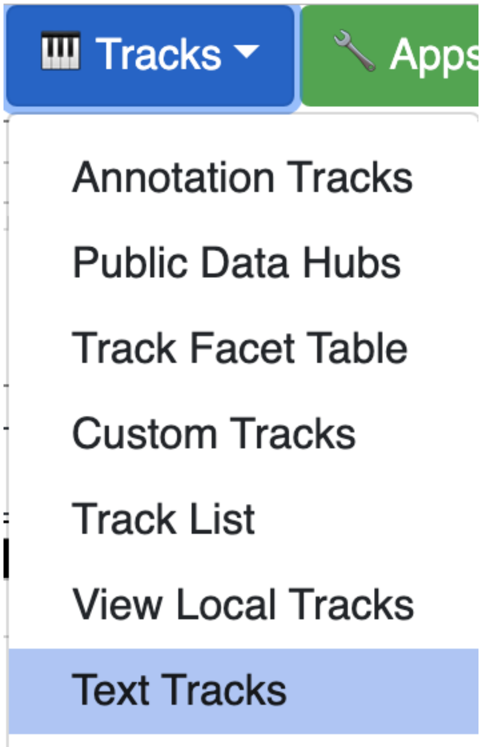
Tracks are usually prepared in binary format for efficient region access, like *bigWig*, *bigBed*, *HiC* and etc. Text file format is very flexible, thus caused some trouble for us to standardize the data input. While there are some circumstances that text track files could also be useful, it can be very convenient to just upload a text file and visualize the data on the browser without formatting the data to a binary format. Once added, text tracks can be very fast since there is no more network requests or transfers, also we have received requests that adding text files as tracks from our users.

6.1 bed

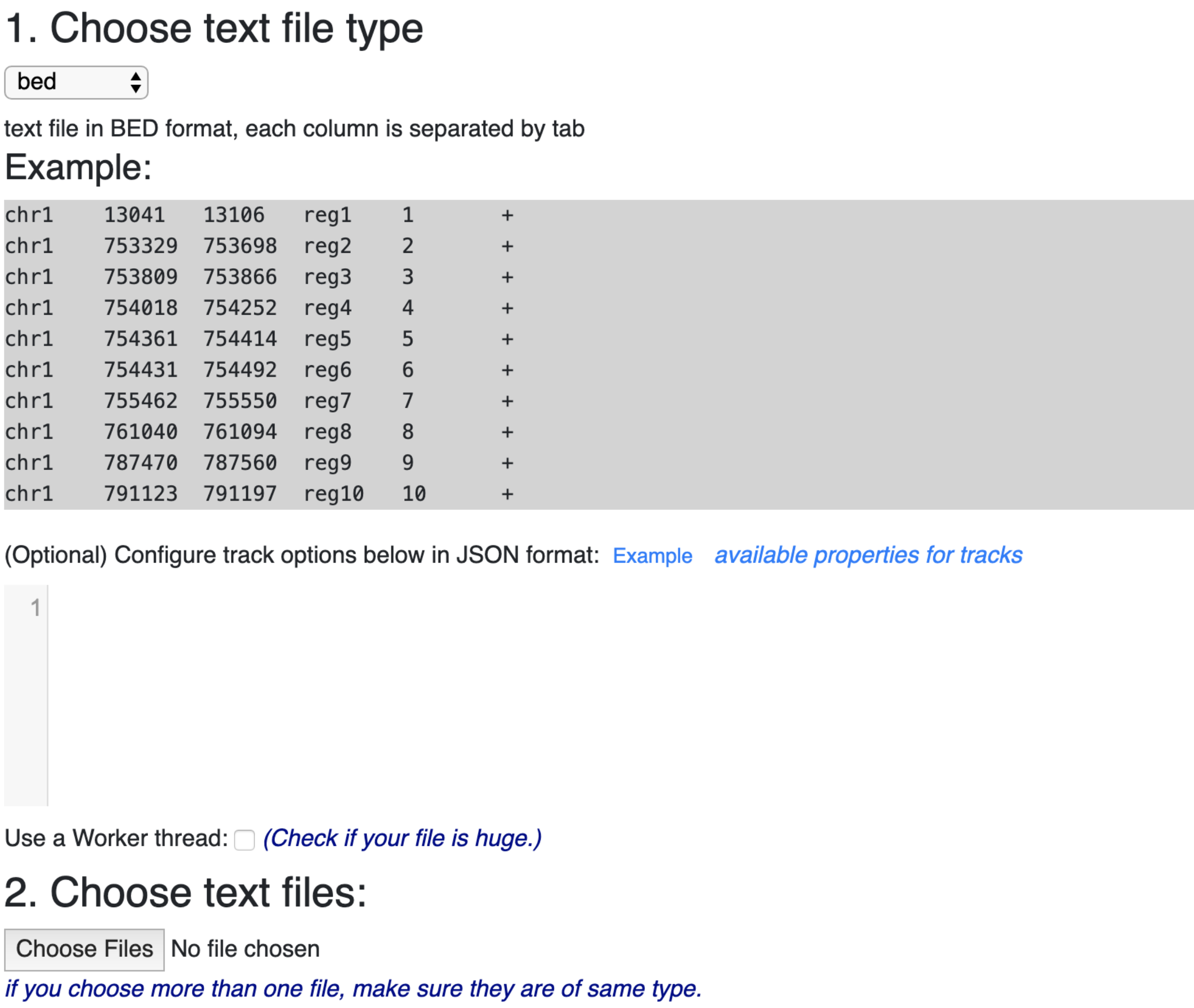
The most common text file would in *bed* format, like this one below:

```
chr1        13041   13106   reg1    1       +
chr1        753329  753698  reg2    2       +
chr1        753809  753866  reg3    3       +
chr1        754018  754252  reg4    4       +
chr1        754361  754414  reg5    5       +
chr1        754431  754492  reg6    6       +
chr1        755462  755550  reg7    7       +
chr1        761040  761094  reg8    8       +
chr1        787470  787560  reg9    9       +
chr1        791123  791197  reg10   10      +
```

Say if we have a text file named `bed-text.txt` with the content above. Open the `Text Tracks` menu:



This will bring the text track UI:



1. Choose text file type

bed

text file in BED format, each column is separated by tab

Example:

```
chr1    13041   13106   reg1    1       +
chr1    753329  753698  reg2    2       +
chr1    753809  753866  reg3    3       +
chr1    754018  754252  reg4    4       +
chr1    754361  754414  reg5    5       +
chr1    754431  754492  reg6    6       +
chr1    755462  755550  reg7    7       +
chr1    761040  761094  reg8    8       +
chr1    787470  787560  reg9    9       +
chr1    791123  791197  reg10   10      +
```

(Optional) Configure track options below in JSON format: Example *available properties for tracks*

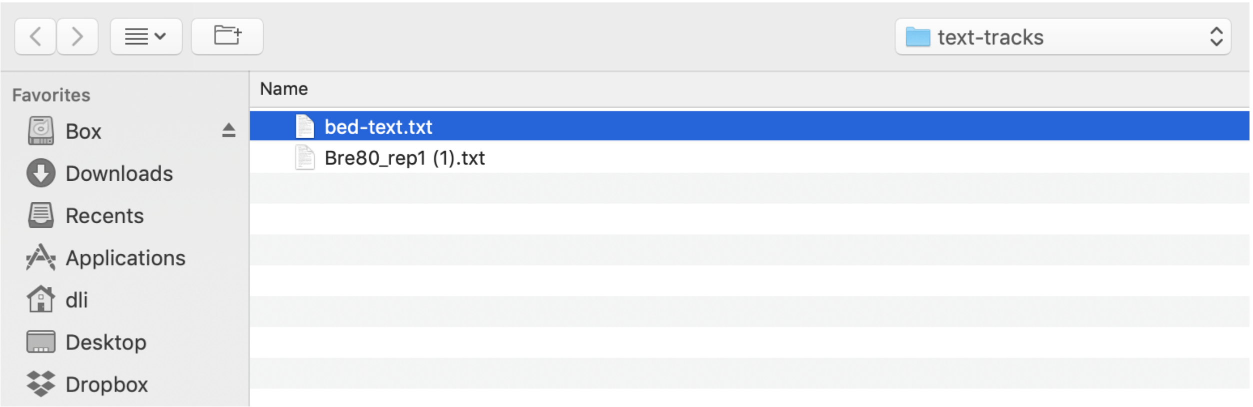
Use a Worker thread: □ *(Check if your file is huge.)*

2. Choose text files:

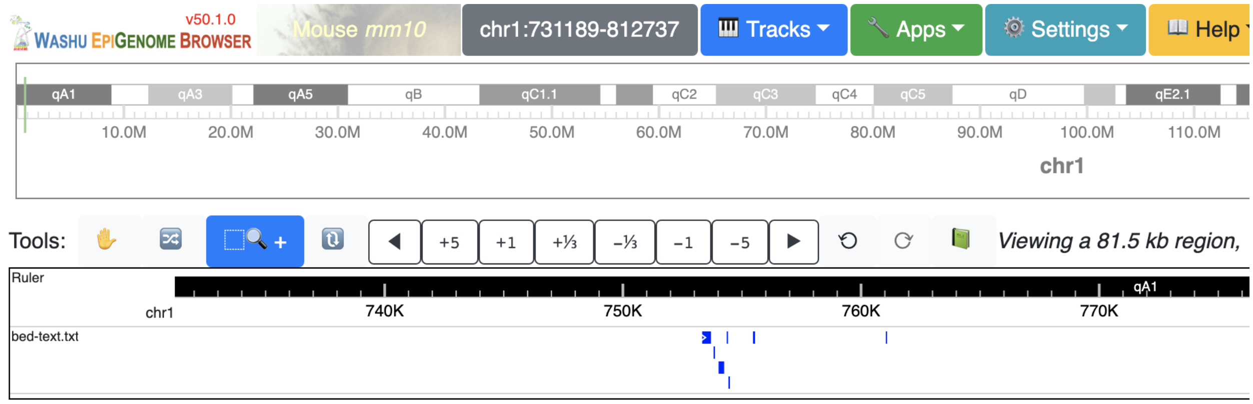
Choose Files No file chosen

if you choose more than one file, make sure they are of same type.

Choose *bed* as the text file type, the choose our text file:



After we submit this file, the track is added with the content of our text file:

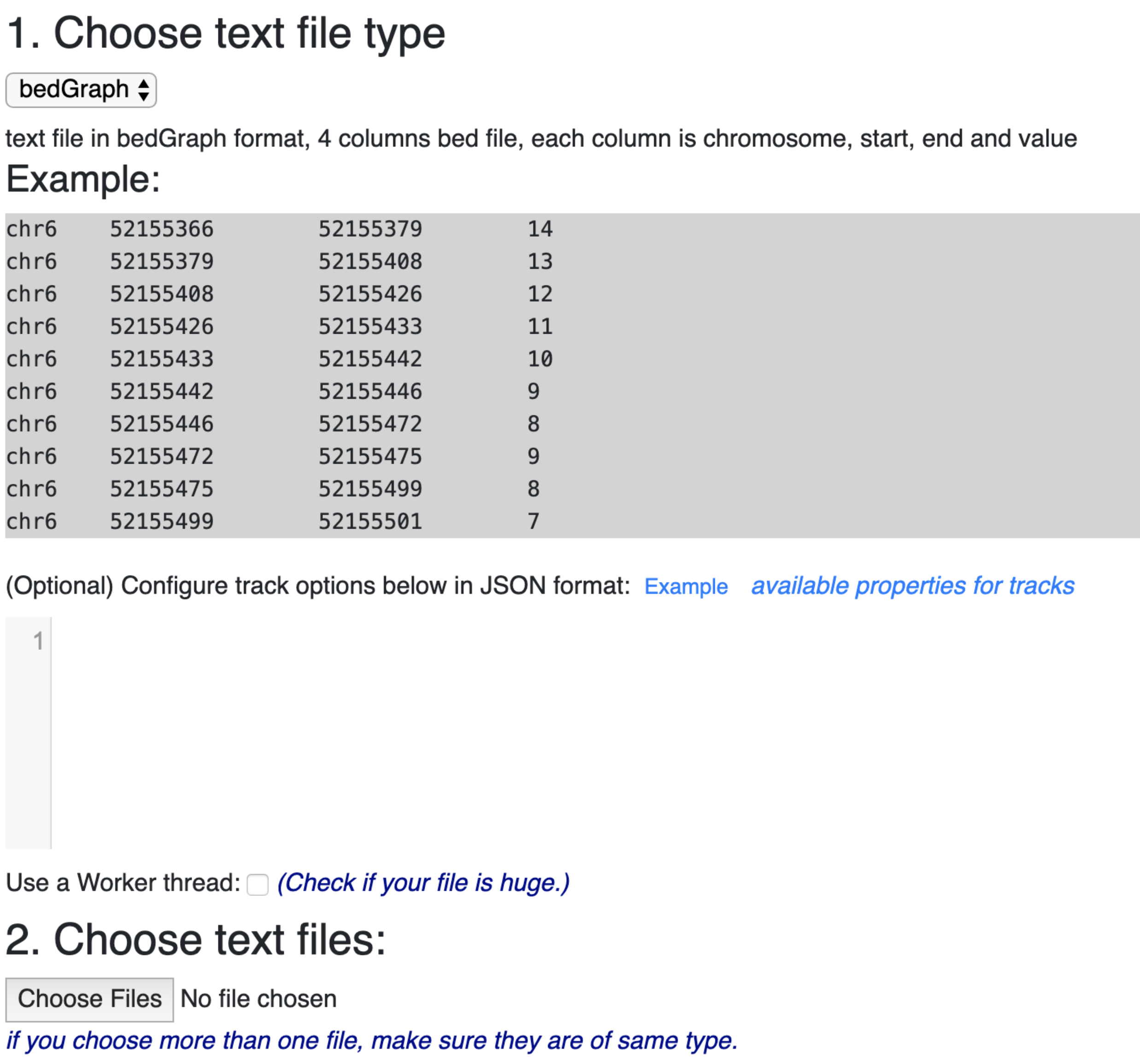


6.2 bedGraph

bedGraph is also very common, it's typically 4 columns bed file like below:

```
chr6        52155366        52155379        14
chr6        52155379        52155408        13
chr6        52155408        52155426        12
chr6        52155426        52155433        11
chr6        52155433        52155442        10
chr6        52155442        52155446        9
chr6        52155446        52155472        8
chr6        52155472        52155475        9
chr6        52155475        52155499        8
chr6        52155499        52155501        7
```

Choose *bedGraph* from the track track UI:



1. Choose text file type

bedGraph ▴▾

text file in bedGraph format, 4 columns bed file, each column is chromosome, start, end and value

Example:

```
chr6	52155366	52155379	14
chr6	52155379	52155408	13
chr6	52155408	52155426	12
chr6	52155426	52155433	11
chr6	52155433	52155442	10
chr6	52155442	52155446	9
chr6	52155446	52155472	8
chr6	52155472	52155475	9
chr6	52155475	52155499	8
chr6	52155499	52155501	7
```

(Optional) Configure track options below in JSON format: Example *available properties for tracks*

```
1
```

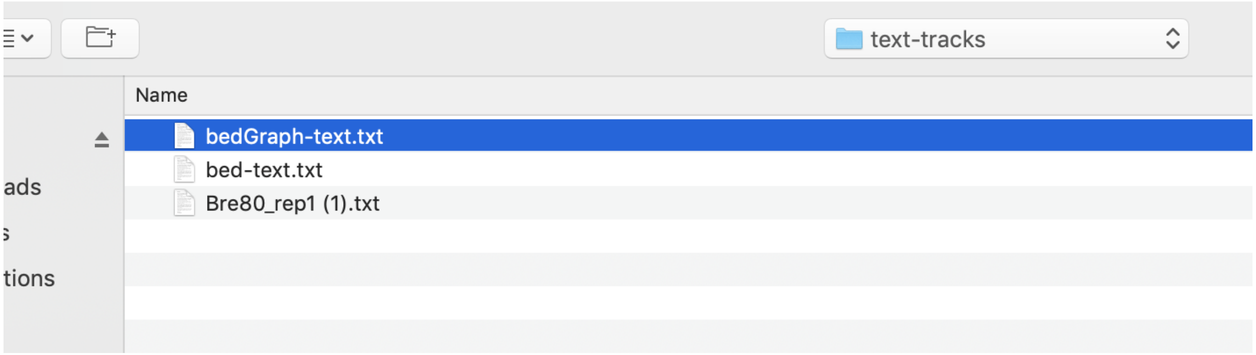
Use a Worker thread: □ *(Check if your file is huge.)*

2. Choose text files:

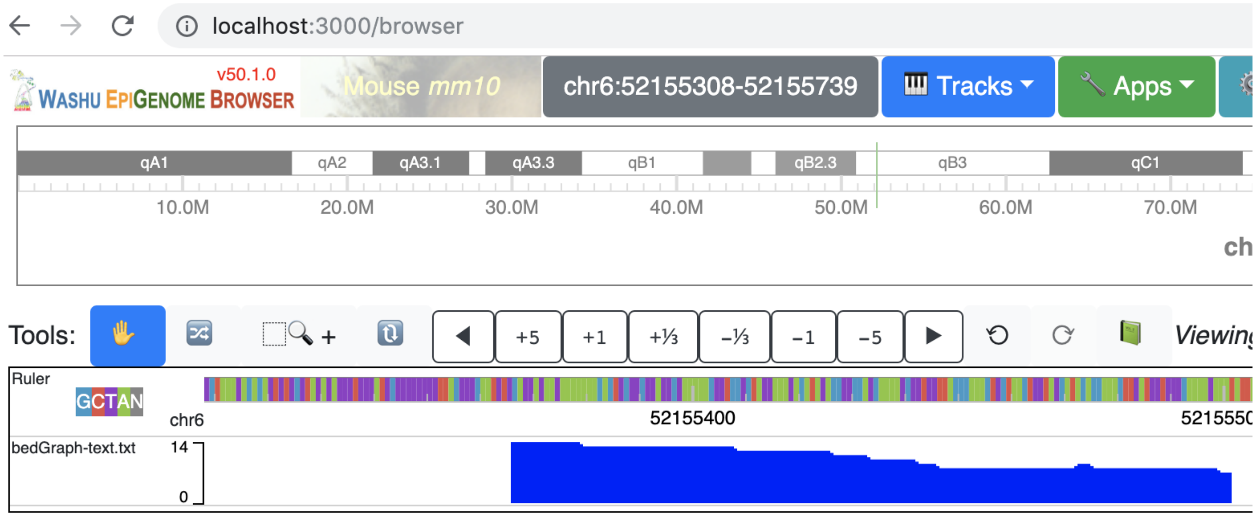
Choose Files No file chosen

if you choose more than one file, make sure they are of same type.

The choose the bedGraph text file:

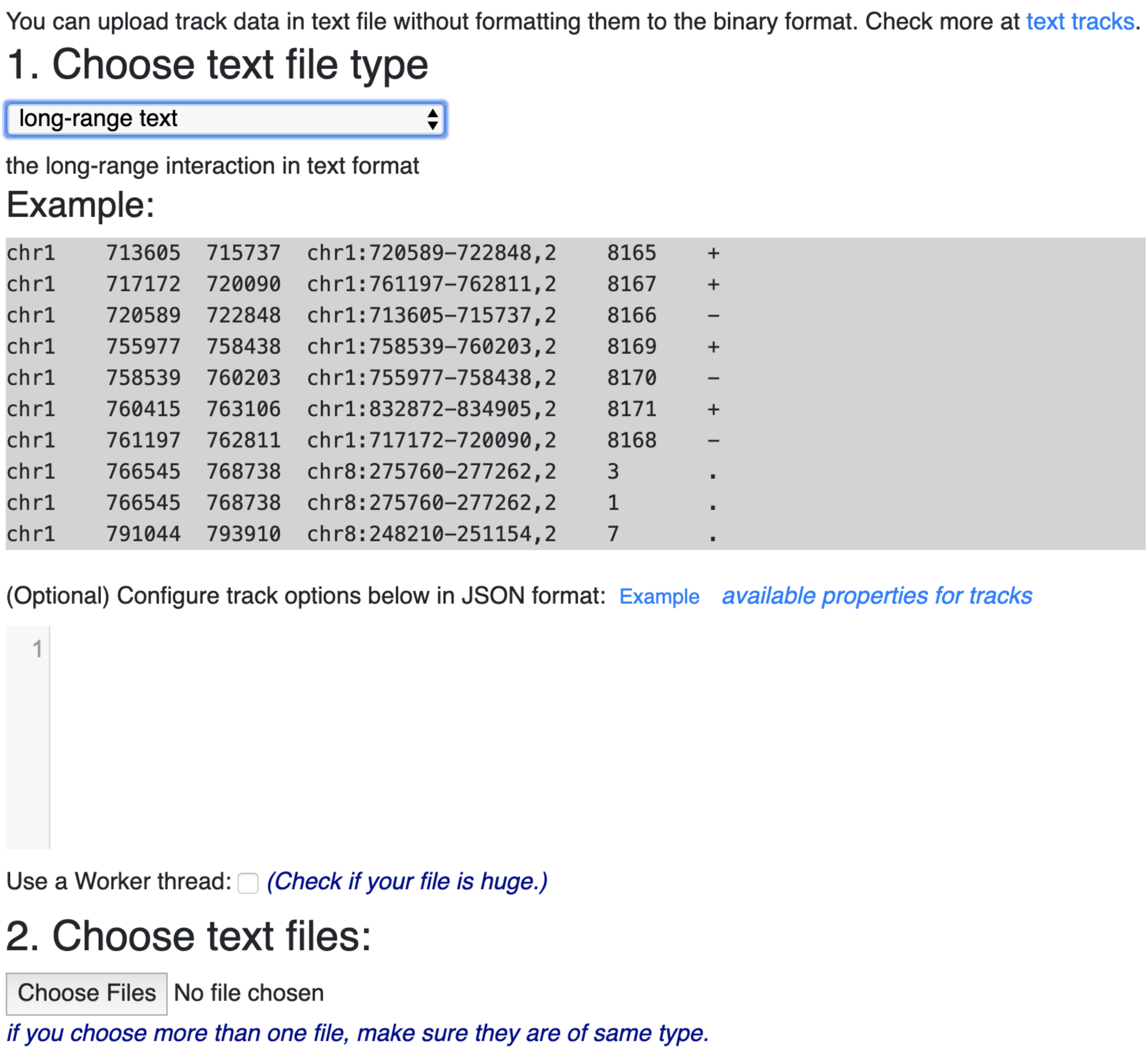


The text track file is added:



6.3 longrange

The *longrange* format can also be uploaded directly as text file, choose *longrange* in the text type dropdown menu:



You can upload track data in text file without formatting them to the binary format. Check more at text tracks.

1. Choose text file type

the long-range interaction in text format

Example:

```
chr1    713605  715737  chr1:720589-722848,2    8165    +
chr1    717172  720090  chr1:761197-762811,2    8167    +
chr1    720589  722848  chr1:713605-715737,2    8166    -
chr1    755977  758438  chr1:758539-760203,2    8169    +
chr1    758539  760203  chr1:755977-758438,2    8170    -
chr1    760415  763106  chr1:832872-834905,2    8171    +
chr1    761197  762811  chr1:717172-720090,2    8168    -
chr1    766545  768738  chr8:275760-277262,2    3       .
chr1    766545  768738  chr8:275760-277262,2    1       .
chr1    791044  793910  chr8:248210-251154,2    7       .
```

(Optional) Configure track options below in JSON format: Example *available properties for tracks*

Use a Worker thread: □ *(Check if your file is huge.)*

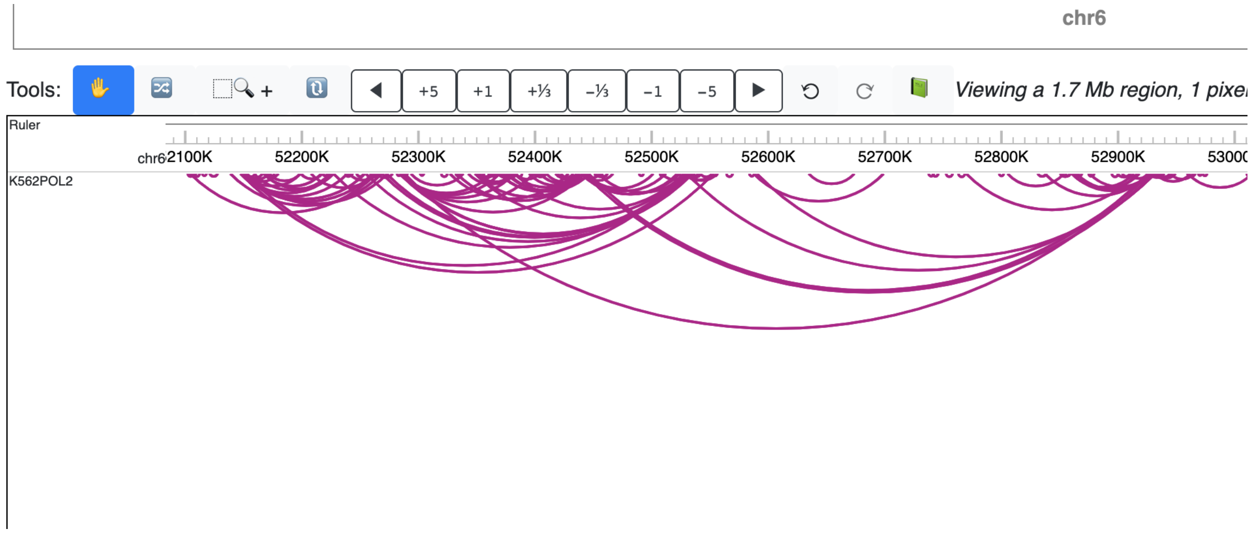
2. Choose text files:

No file chosen

if you choose more than one file, make sure they are of same type.

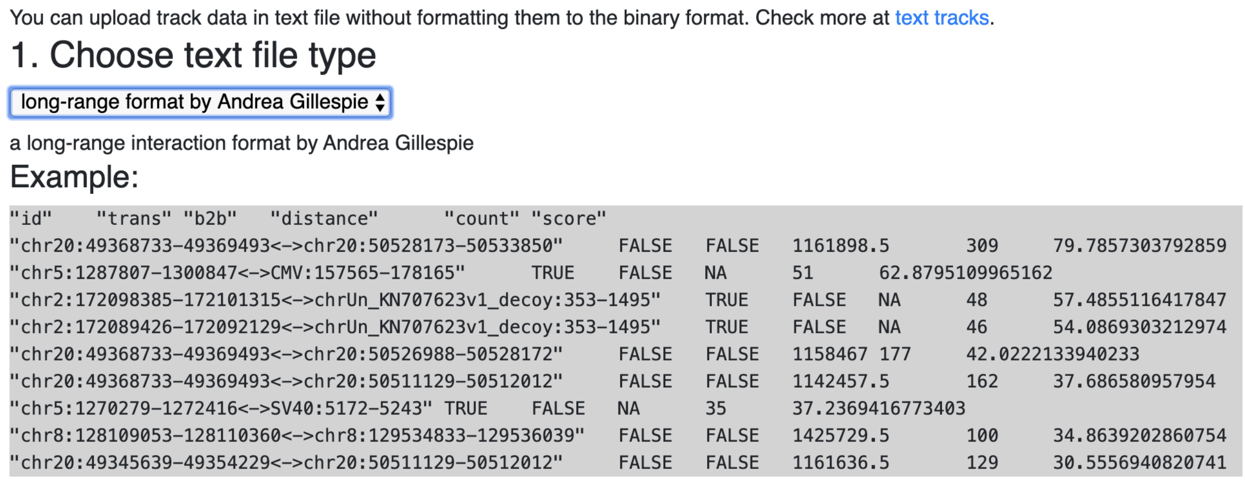
Choose the text file in *longrange* format:

The track will be added as below (adjust region and display style as arc):



6.4 customized long-range format

One of our user proposed a long-range format as below:



You can upload track data in text file without formatting them to the binary format. Check more at text tracks.

1. Choose text file type

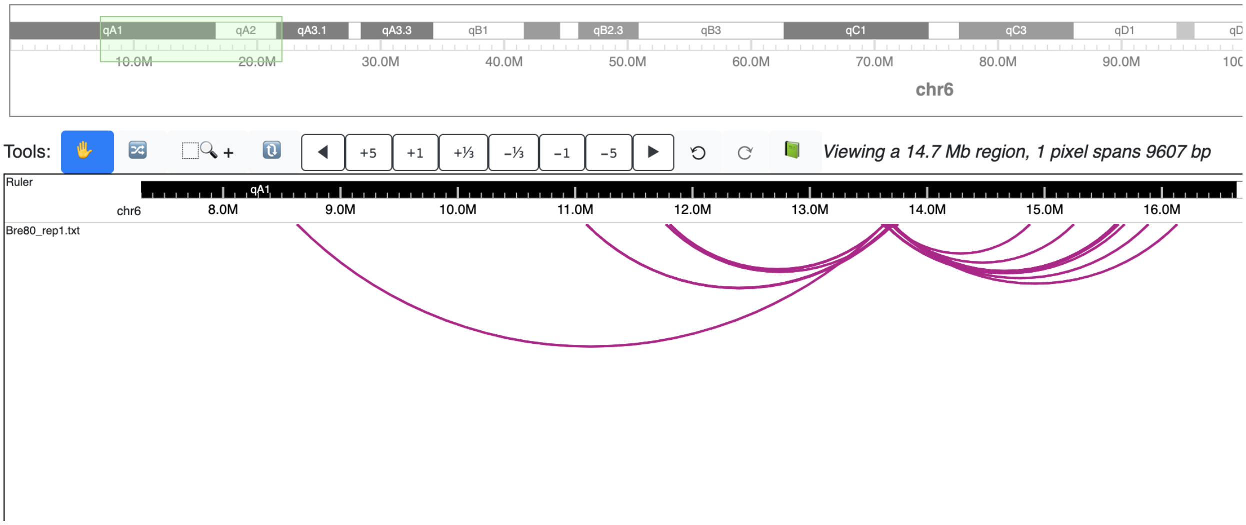
long-range format by Andrea Gillespie ▲▼

a long-range interaction format by Andrea Gillespie

Example:

```
"id"    "trans" "b2b"   "distance"      "count" "score"
"chr20:49368733-49369493<->chr20:50528173-50533850"     FALSE   FALSE   1161898.5       309     79.7857303792859
"chr5:1287807-1300847<->CMV:157565-178165"      TRUE    FALSE   NA      51      62.8795109965162
"chr2:172098385-172101315<->chrUn_KN707623v1_decoy:353-1495"    TRUE    FALSE   NA      48      57.4855116417847
"chr2:172089426-172092129<->chrUn_KN707623v1_decoy:353-1495"    TRUE    FALSE   NA      46      54.0869303212974
"chr20:49368733-49369493<->chr20:50526988-50528172"     FALSE   FALSE   1158467 177     42.0222133940233
"chr20:49368733-49369493<->chr20:50511129-50512012"     FALSE   FALSE   1142457.5       162     37.686580957954
"chr5:1270279-1272416<->SV40:5172-5243" TRUE    FALSE   NA      35      37.2369416773403
"chr8:128109053-128110360<->chr8:129534833-129536039"   FALSE   FALSE   1425729.5       100     34.8639202860754
"chr20:49345639-49354229<->chr20:50511129-50512012"     FALSE   FALSE   1161636.5       129     30.5556940820741
```

We also added the support and file in this format can be loaded as track:

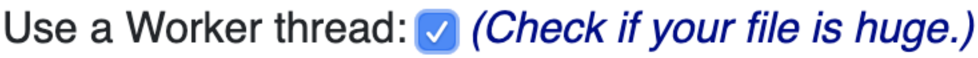


Viewing a 14.7 Mb region, 1 pixel spans 9607 bp

Feel free to contact us if you need more formats supported.

6.5 What if the text file is huge?

If your text track is huge in size, convert to binary format is recommended. However, you can still use the text file if you want. Make sure you check the *Use a Worker thread* checkbox, the browser will use a background thread for text file loading.



Use a Worker thread: ☑ *(Check if your file is huge.)*

CHAPTER

SEVEN

DYNAMIC TRACKS

Dynamic tracks is a new track type in the Browser to show dynamics of data as animations. Currently annotation features (bed), numerical data (bigWig, bedgraph) and chromatin interaction data (like HiC) (hic, longrange) can be visualized with this new track type.

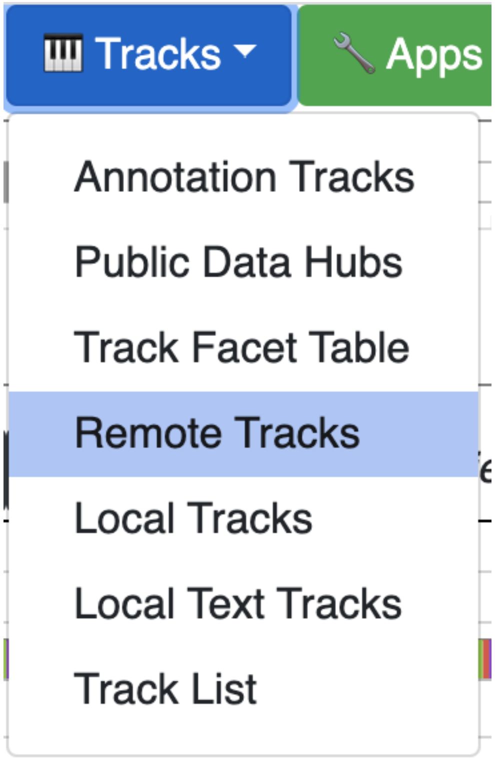
7.1 Use dynamic bedgraph format

For the dynamic track visualization, we developed a new and easy to use format called *dynamic bedgraph*, it's almost same as regular bedgraph format, except last column is an array of values:

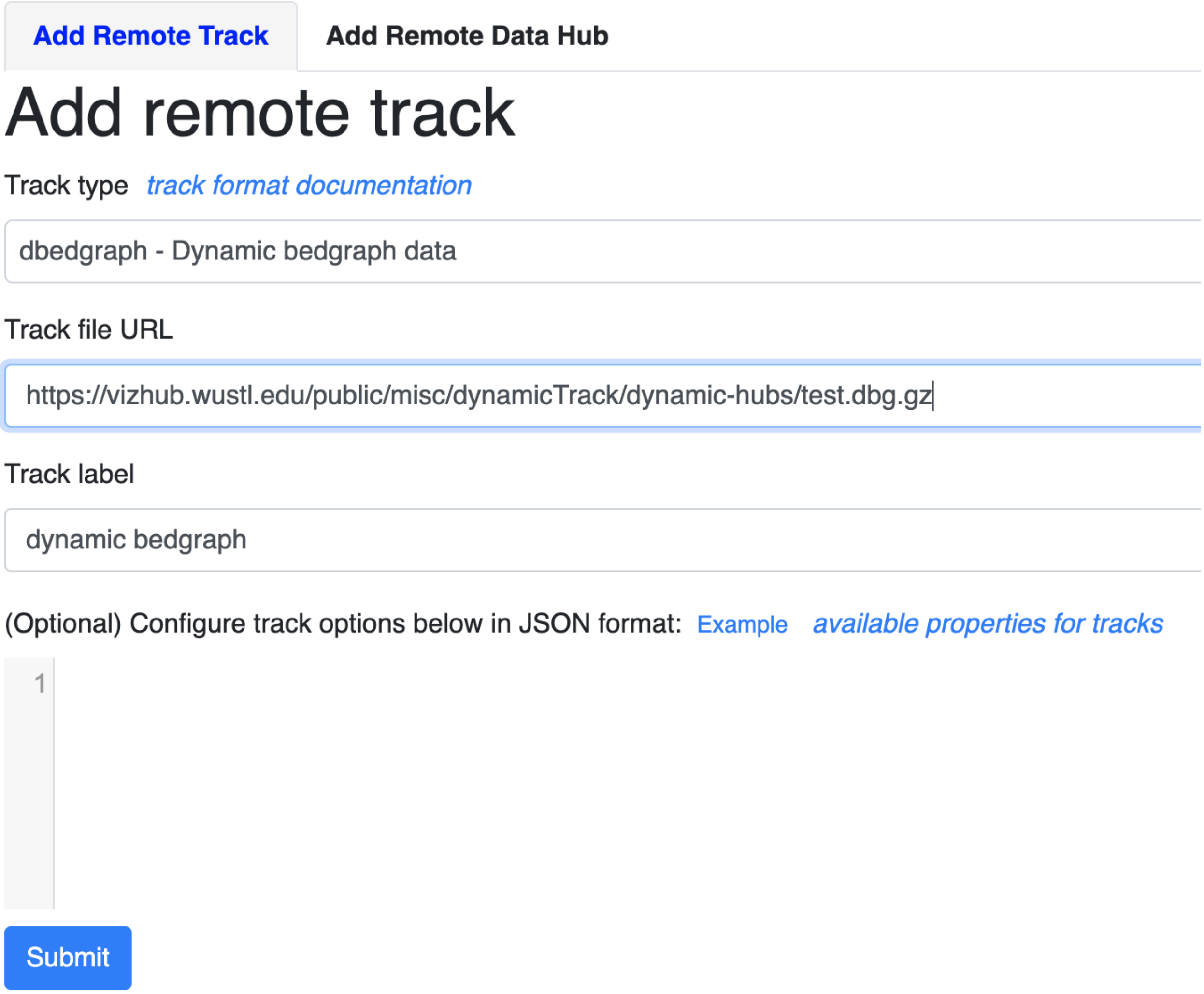
```
chr6    52424961        52425161        [10,9,8,7,6,5,4,3,2,1]
chr6    52425286        52425296        [1,2,3,4,5,6,7,8,9,10]
```

Important: The data array in 4th column are not required to be always with same length, for shorter arrays, values will be used repeatly from beginning of the array if the same view window has longer arrays. To avoid this repeating process, users need to fill the missing data with 0.

Format the file with bgzip and tabix, this example file can be accessed from https://vizhub.wustl.edu/public/misc/dynamicTrack/dynamic-hubs/test.dbg.gz, you can submit the new track file as a remote track:



Use the URL to the track file and choose track type as `dbedgraph`:



Add remote track

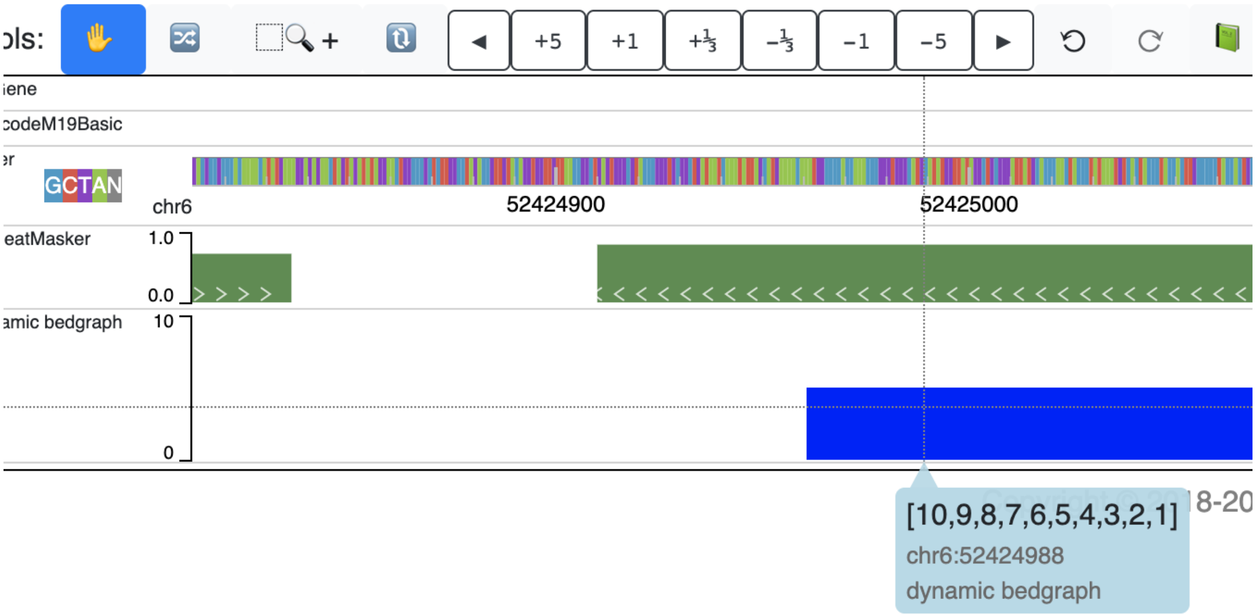
Track type: dbedgraph - Dynamic bedgraph data

Track file URL: https://vizhub.wustl.edu/public/misc/dynamicTrack/dynamic-hubs/test.dbg.gz

Track label: dynamic bedgraph

(Optional) Configure track options below in JSON format:

After click *Submit* button, the new track will be added:



An animated version is here:

7.1.1 Dynamic labels with dynamic track

The track file above can also be used to prepare a data hub file as below, specify the `dynamicLabels` at same time:

```
[
    {
        "type": "dbedgraph",
        "url": "https://vizhub.wustl.edu/public/misc/dynamicTrack/dynamic-hubs/test.dbg.
↪gz",
        "options": {
            "dynamicLabels": ['stage1','stage2','stage3','stage4','stage5','stage6',
↪'stage7','stage8','stage9','stage10']
        },
        "showOnHubLoad": true
    }
]
```

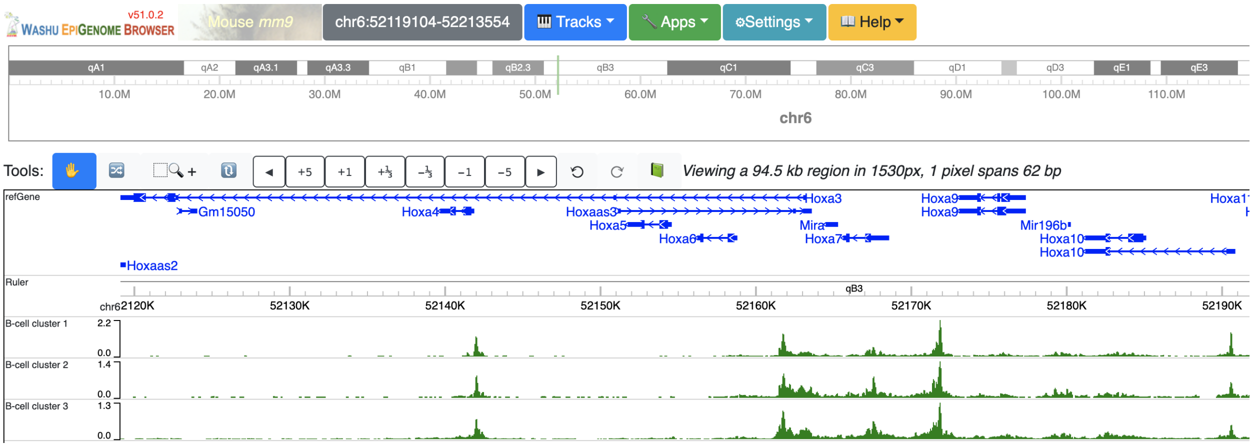
When you submit this file as a data hub, you could see the labels are ploted along with the corresponding data:

7.1.2 Dynamic bedgraph track with negative values

In the data array of dynamic bedgraph track file, negative values are also allowed, animation will toward downside for nagetive values. An example is displayed below:

7.2 Make dynamic plot track from user interface

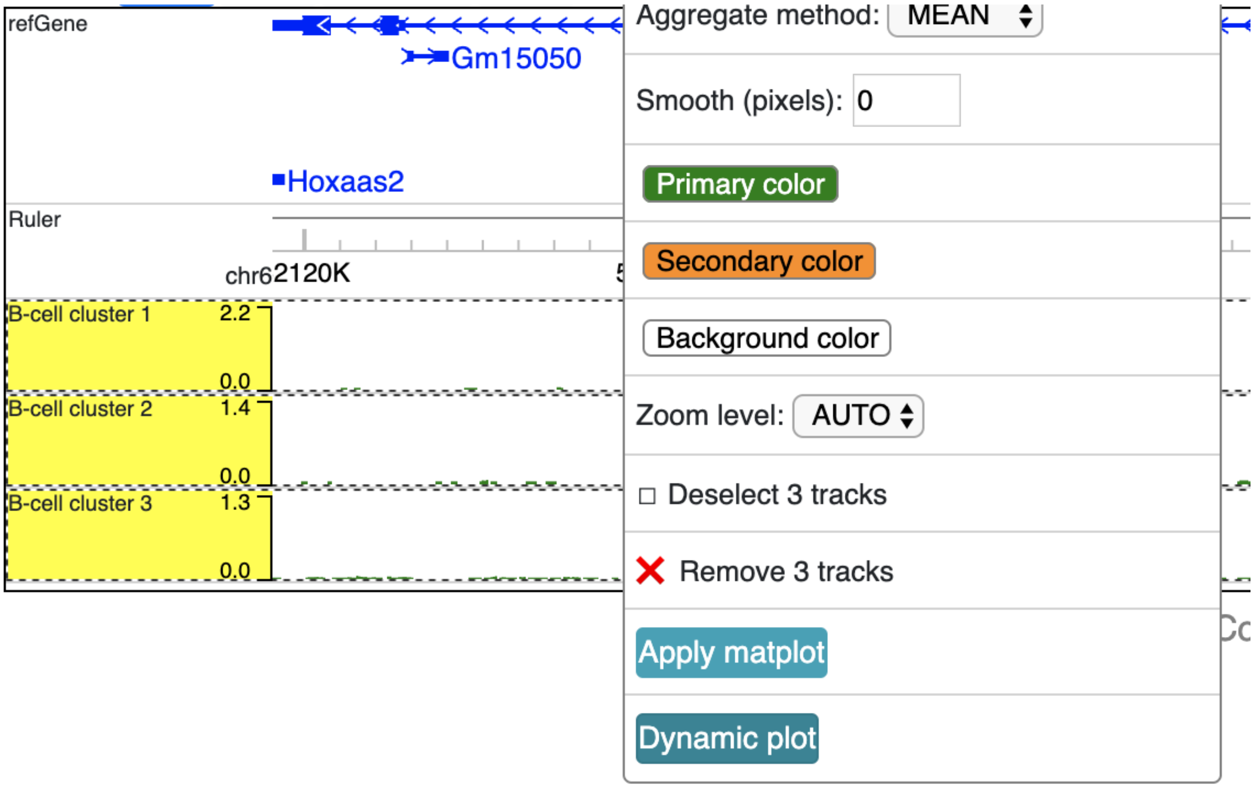
Dynamic tracks can also be made from multiple existing numerical tracks, without any further reformat. In the screenshot below we have 3 bigWig tracks loaded to mm9 genome:



chr6:52119104-52213554

Viewing a 94.5 kb region in 1530px, 1 pixel spans 62 bp

Select these 3 tracks while holding Shift key, then choose 'Dynamic plot' menu:

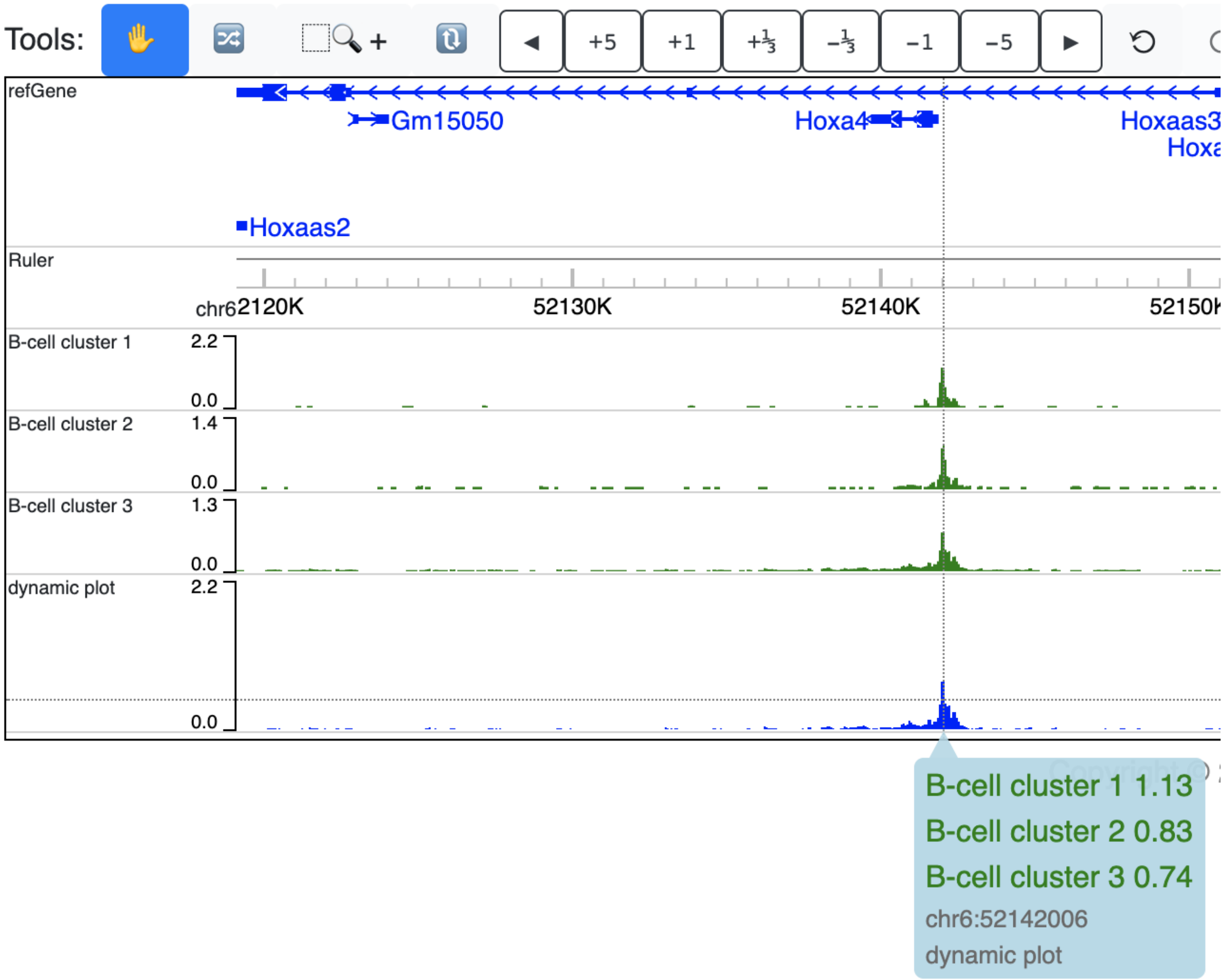


Aggregate method: MEAN

Smooth (pixels): 0

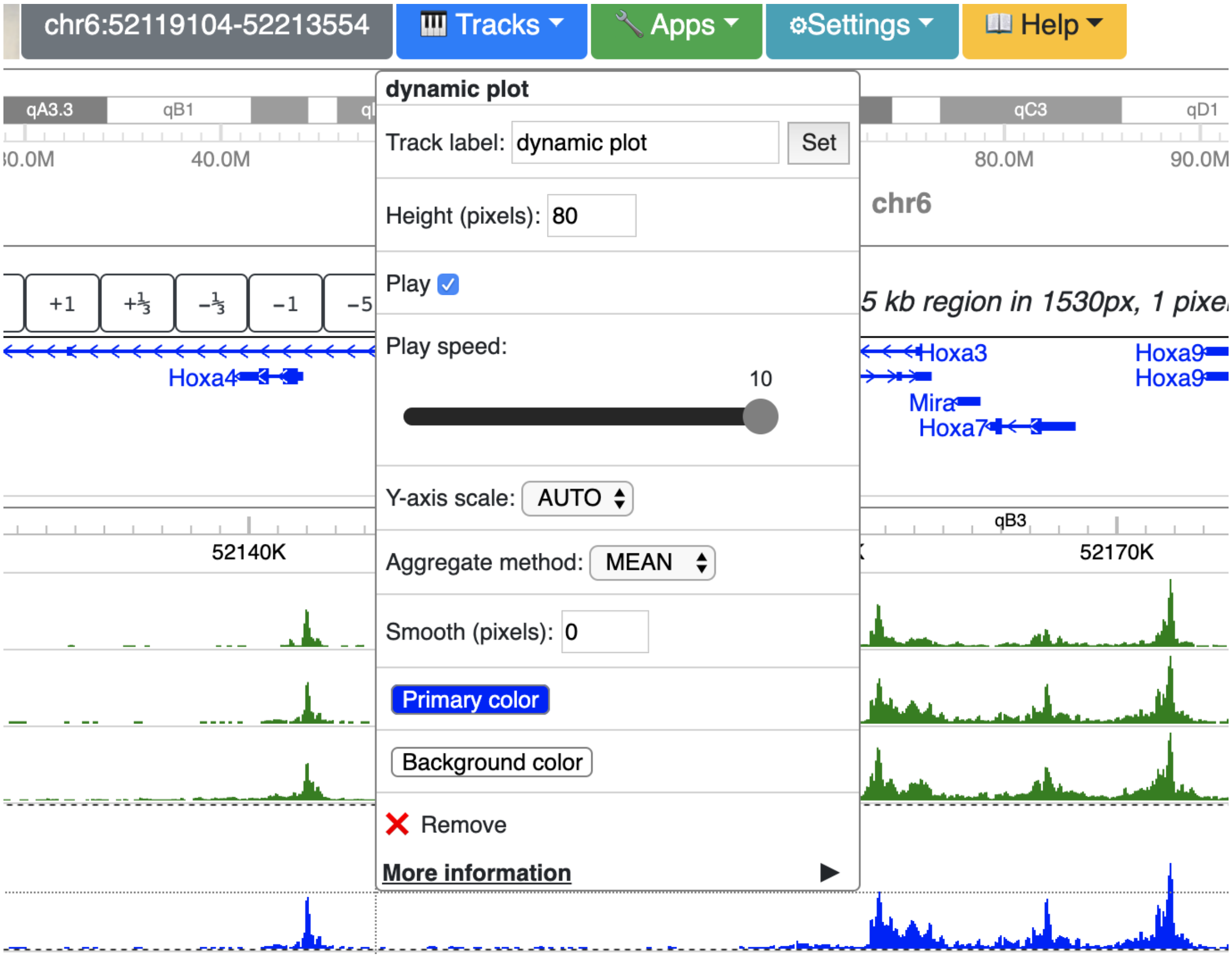
Zoom level: AUTO

A new dynamic track will be displayed:



B-cell cluster 1 1.13
B-cell cluster 2 0.83
B-cell cluster 3 0.74
chr6:52142006
dynamic plot

Right click the track gives your configuration menu:



dynamic plot

Track label: dynamic plot Set

Height (pixels): 80

Play ☑

Play speed:

10

Y-axis scale: AUTO

Aggregate method: MEAN

Smooth (pixels): 0

Primary color

Background color

✕ Remove

More information ▶

An animated version can be seen below:

7.3 Make dynamic plot track using data hub

Dynamic tracks can also be submitted to the browser by using the remote data hub function, just same as existing data hub syntax, dynamic tracks are coded in the JSON format as below:

```
[
    {
        "type": "dynamic",
        "name": "dynamic plot example",
        "showOnHubLoad": true,
        "tracks": [
        {
            "type": "bigwig",
            "url": "https://vizhub.wustl.edu/public/misc/dynamicTrack/markers/
↪ENCFF051LQD_H3K4me1.bigWig",
            "name": "CH12 H3K4me1"
        },
```

(continues on next page)

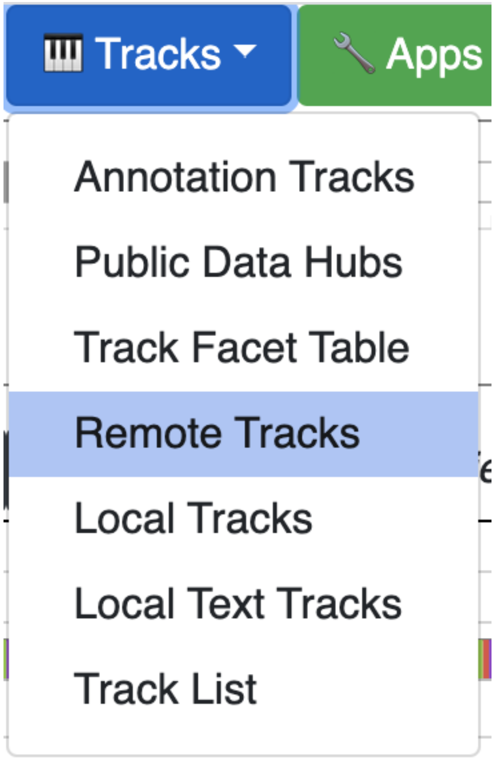
(continued from previous page)

```
        {
            "type": "bigwig",
            "url": "https://vizhub.wustl.edu/public/misc/dynamicTrack/markers/
↪ENCFF096TSJ_H3K27ac.bigWig",
            "name": "CH12 H3K27ac"
        },
        {
            "type": "bigwig",
            "url": "https://vizhub.wustl.edu/public/misc/dynamicTrack/markers/
↪ENCFF011TAF_H3K4me3.bigWig",
            "name": "CH12 H3K4me3"
        },
        {
            "type": "bigwig",
            "url": "https://vizhub.wustl.edu/public/misc/dynamicTrack/markers/
↪ENCFF700XWH_H3K36me3.bigWig",
            "name": "CH12 H3K36me3"
        }
        ]
    }
]
```

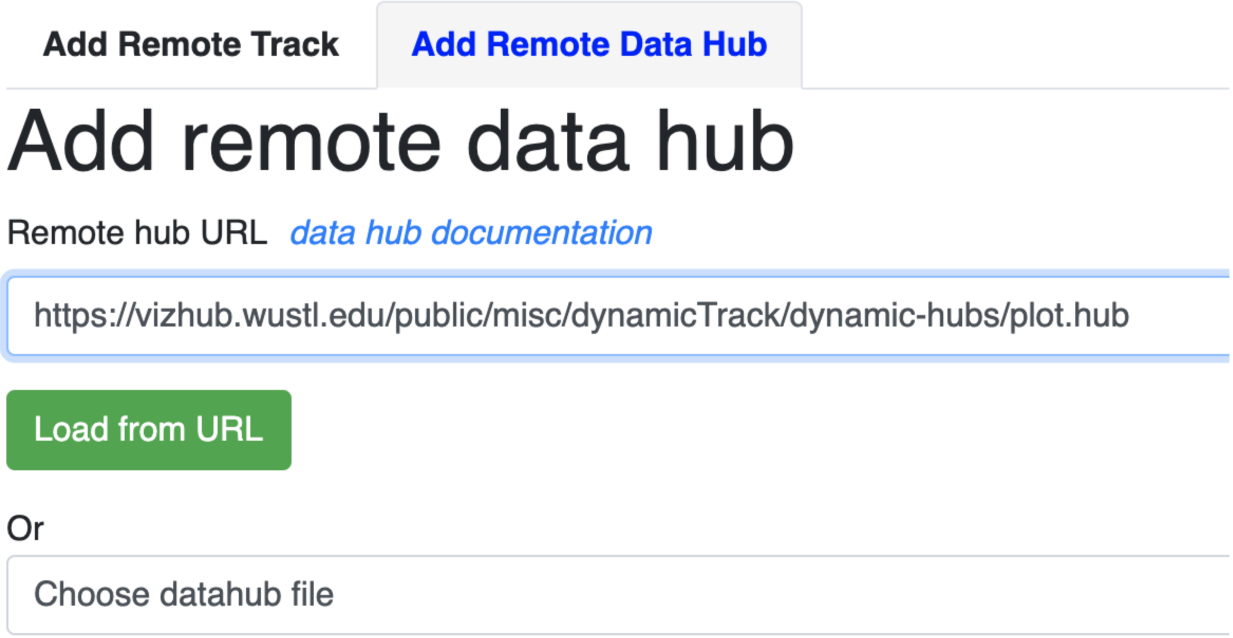
Please notice the track type is `dynamic`, the *tracks* attribute indicates the member tracks of this dynamic track.

This hub is also available at https://vizhub.wustl.edu/public/misc/dynamicTrack/dynamic-hubs/plot.hub

Open the Remote tracks menu:



Then choose remote hub and load the hub from your hub's URL:

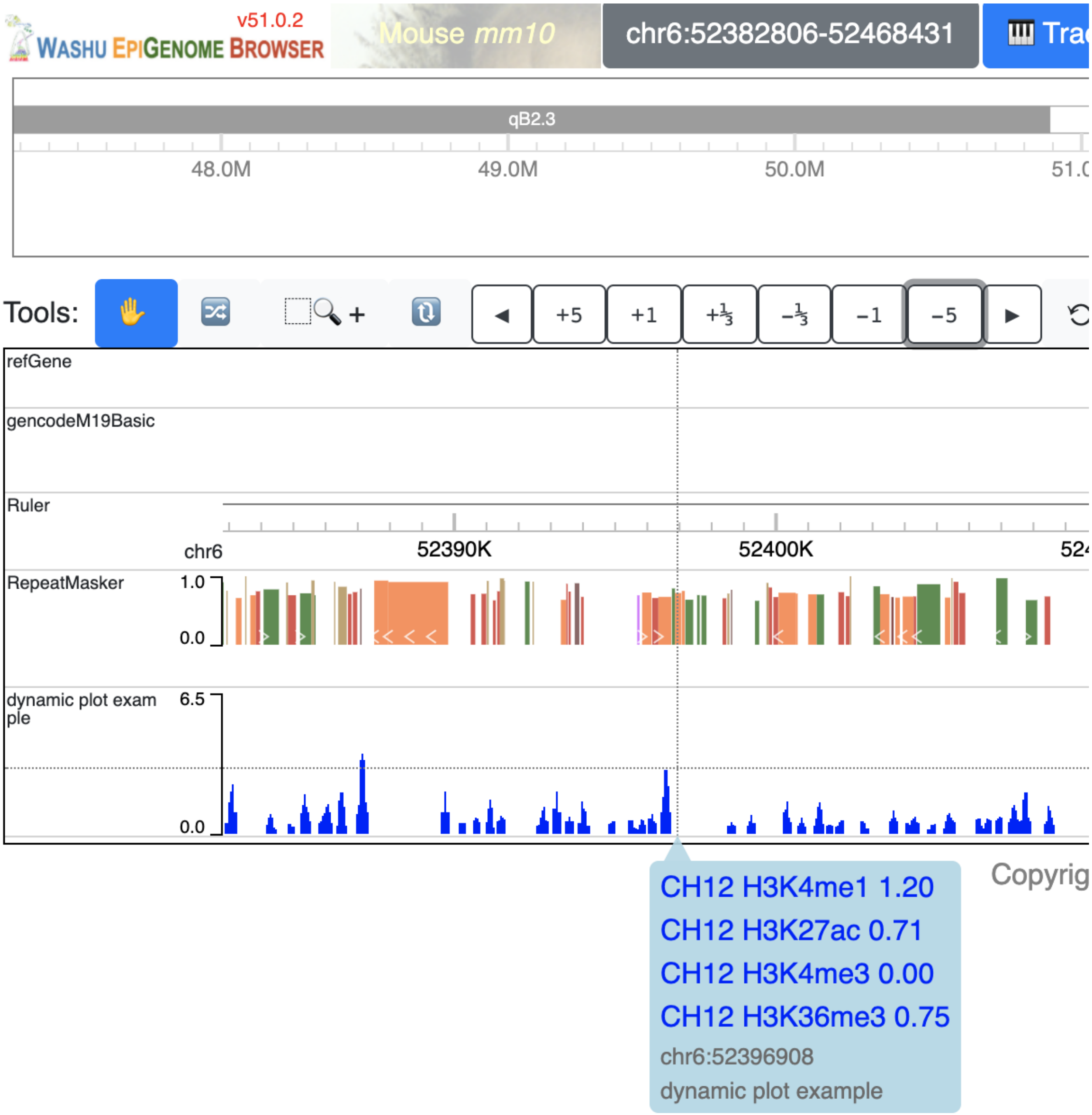


Add remote data hub

Remote hub URL *data hub documentation*

https://vizhub.wustl.edu/public/misc/dynamicTrack/dynamic-hubs/plot.hub

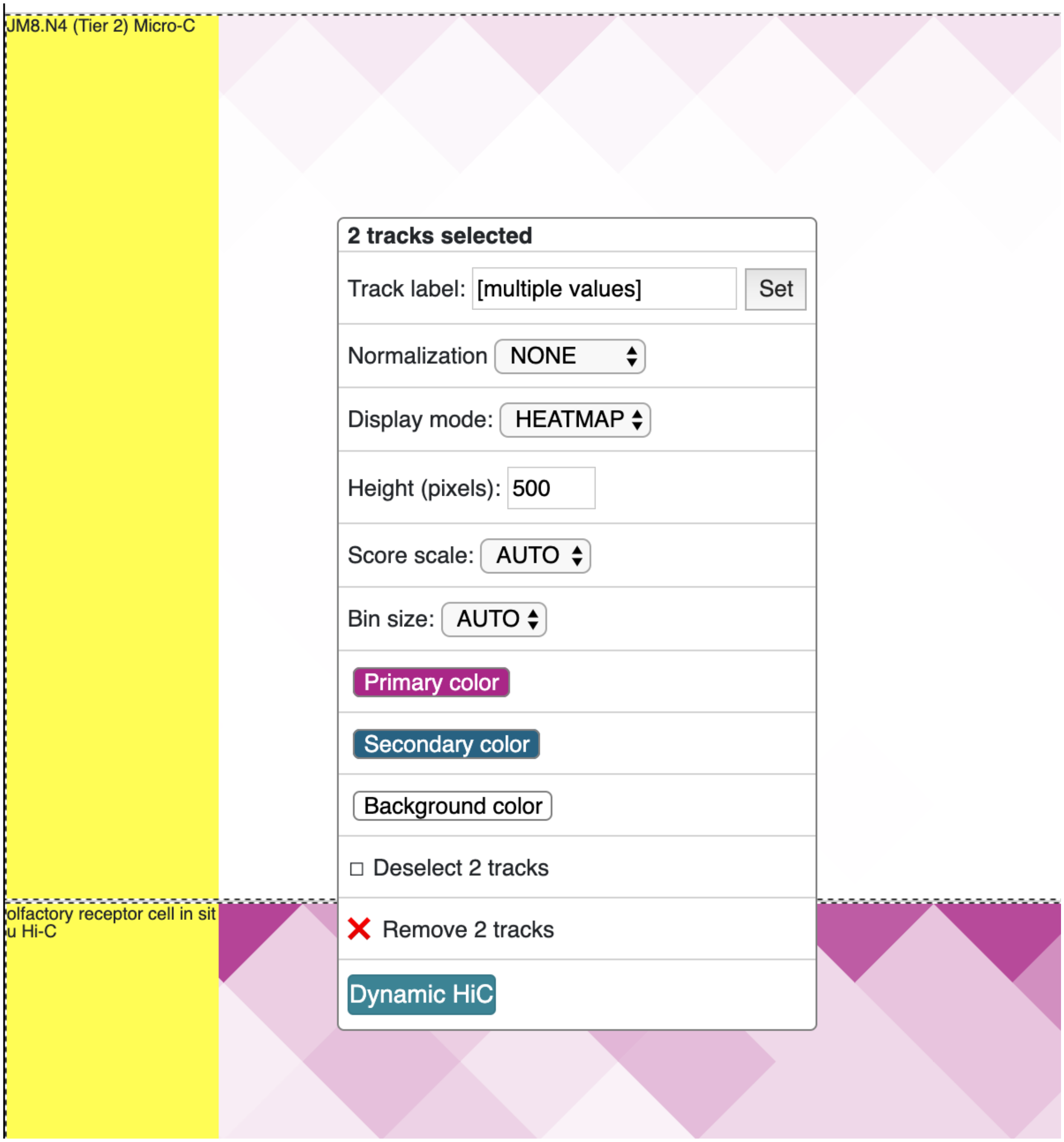
The track will be loaded as below:



CH12 H3K4me1 1.20
CH12 H3K27ac 0.71
CH12 H3K4me3 0.00
CH12 H3K36me3 0.75
chr6:52396908
dynamic plot example

7.4 Make dynamic HiC maps from the user interface

Load more than 2 HiC tracks, selct all of them by holding *Shift* key, and click the *Dynamic HiC* button:



2 tracks selected

Track label: [multiple values]

Normalization: NONE

Display mode: HEATMAP

Height (pixels): 500

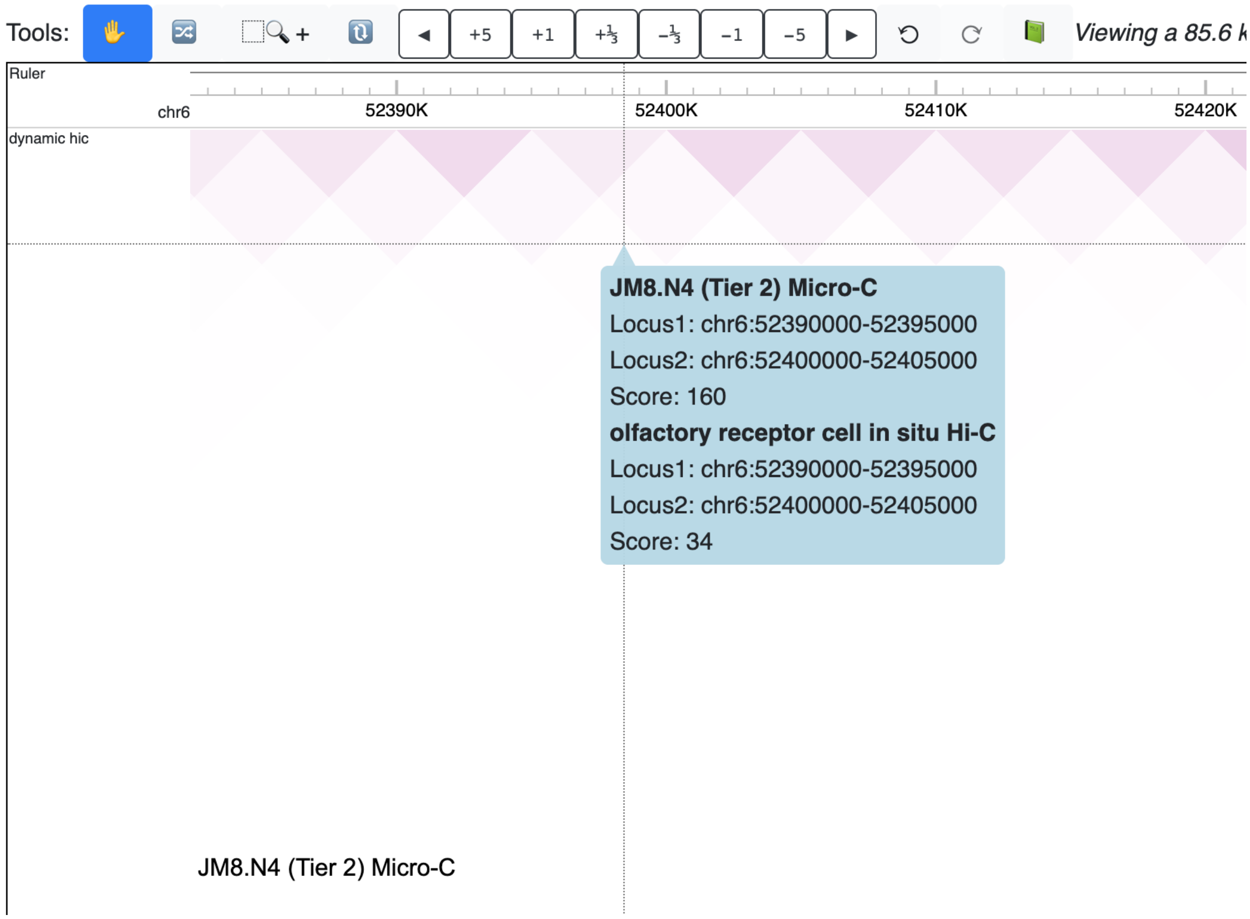
Score scale: AUTO

Bin size: AUTO

□ Deselect 2 tracks

✕ Remove 2 tracks

The new track is added as below:



JM8.N4 (Tier 2) Micro-C
Locus1: chr6:52390000-52395000
Locus2: chr6:52400000-52405000
Score: 160
olfactory receptor cell in situ Hi-C
Locus1: chr6:52390000-52395000
Locus2: chr6:52400000-52405000
Score: 34

Check the animated verison below:

7.5 Make dynamic HiC maps using data hub

Dynamic HiC tracks can also be submitted using remote data hub function. Prepare a data hub file like below:

```
[
{
    "name": "dynamic hic",
    "type": "dynamichic",
    "tracks": [
    {
        "name": "olfactory receptor cell in situ Hi-C [4DNFIT4I5C6Z]",
        "type": "hic",
        "url": "https://data.4dnucleome.org/files-processed/4DNFIT4I5C6Z/@@download/
↪4DNFIT4I5C6Z.hic"
    },
    {
        "name": "olfactory receptor cell in situ Hi-C [4DNFIXKC48TK]",
        "type": "hic",
        "url": "https://data.4dnucleome.org/files-processed/4DNFIXKC48TK/@@download/
```

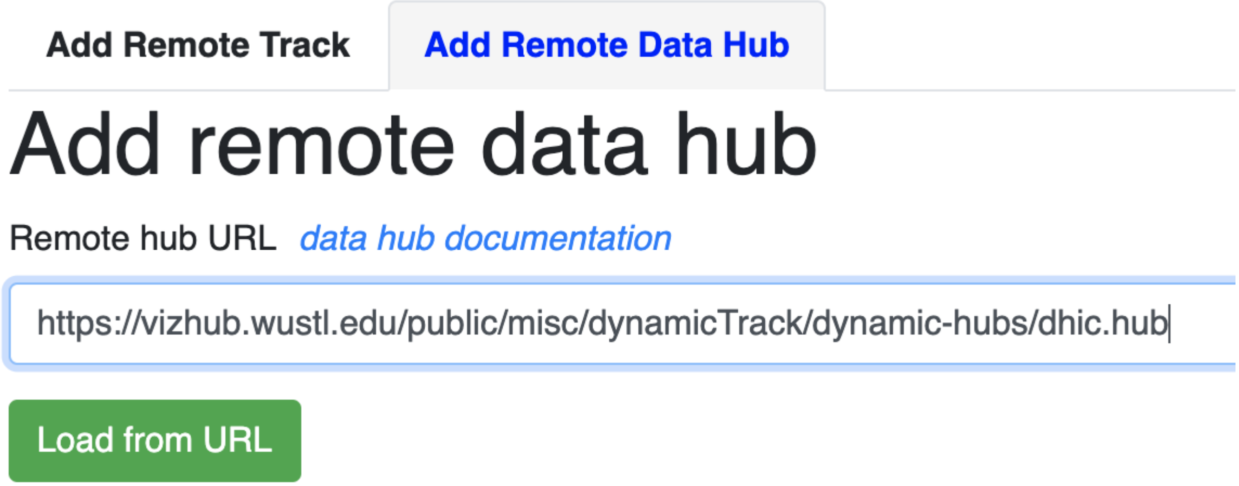
(continues on next page)

(continued from previous page)

```
↪4DNFIXKC48TK.hic"
    }
    ],
    "showOnHubLoad": true
}
]
```

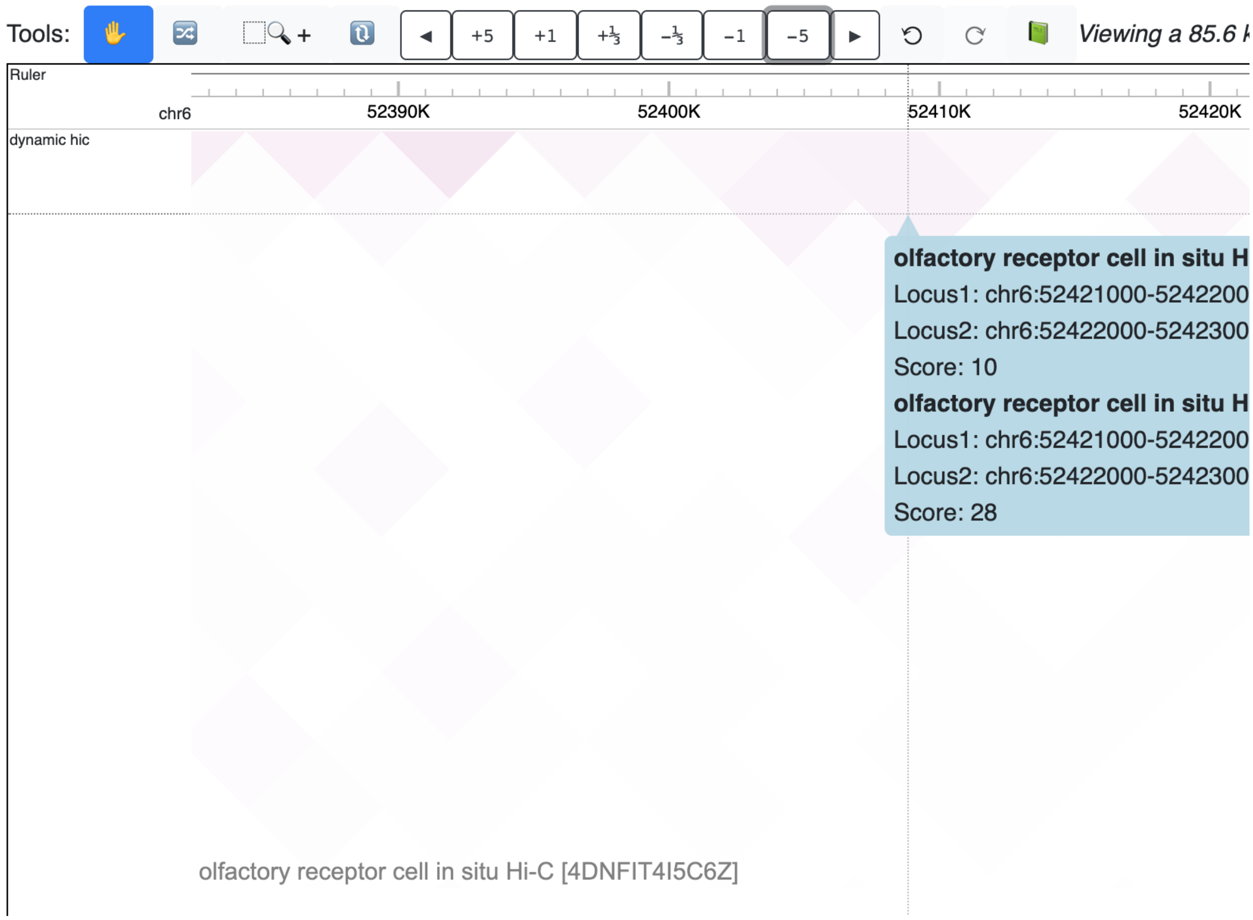
This hub is located at: https://vizhub.wustl.edu/public/misc/dynamicTrack/dynamic-hubs/dhic.hub

Submit this link as a remote data hub:



https://vizhub.wustl.edu/public/misc/dynamicTrack/dynamic-hubs/dhic.hub

The new dynamic HiC track is added:

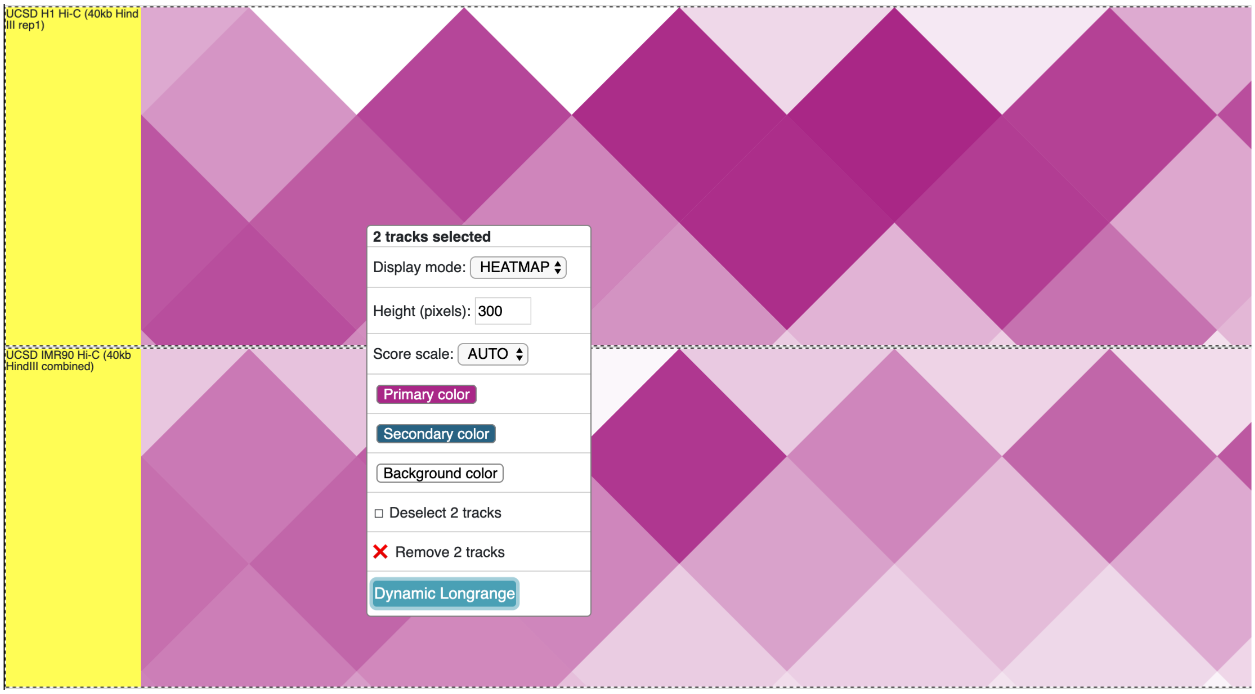


olfactory receptor cell in situ H
Locus1: chr6:52421000-5242200
Locus2: chr6:52422000-5242300
Score: 10
olfactory receptor cell in situ H
Locus1: chr6:52421000-5242200
Locus2: chr6:52422000-5242300
Score: 28

Check the animated version below:

7.6 Make dynamic longrange chromatin interaction track

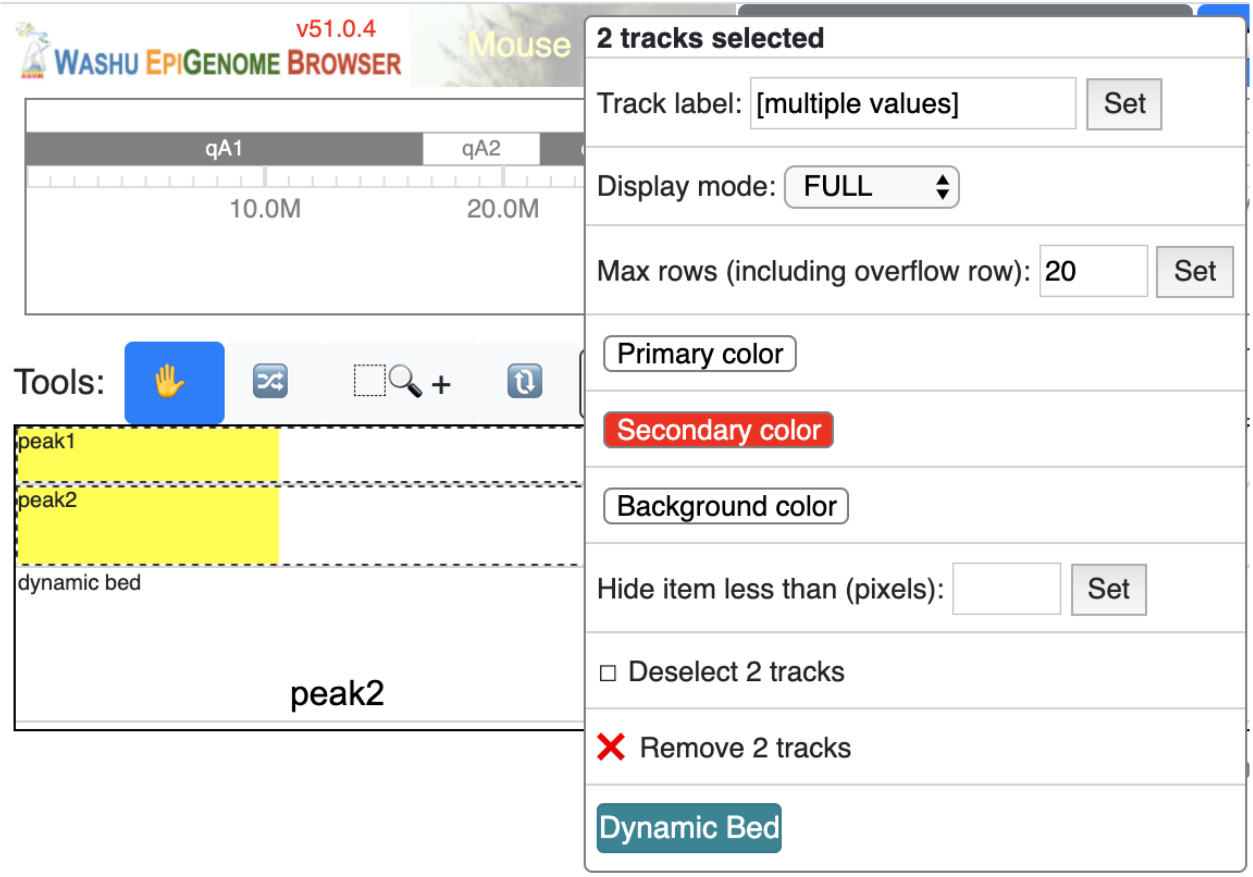
`longrange` chromatin interaction tracks can also be used to make dynamic tracks. First, load more than 1 `longrange` track, select all of them while holding *Shift* key, right click on the selction, and choose *Dynamic Longrange*:



The new dynamic interaction track will be added, an animated version is displayed below:

7.7 Make Dynamic bed track for annotation data

`bed` tracks can also be made to be dynamic. Load more than 1 `bed` track in the browser, select all of them while holding *Shift* key, right click, and choose *Dynamic bed* button:



2 tracks selected

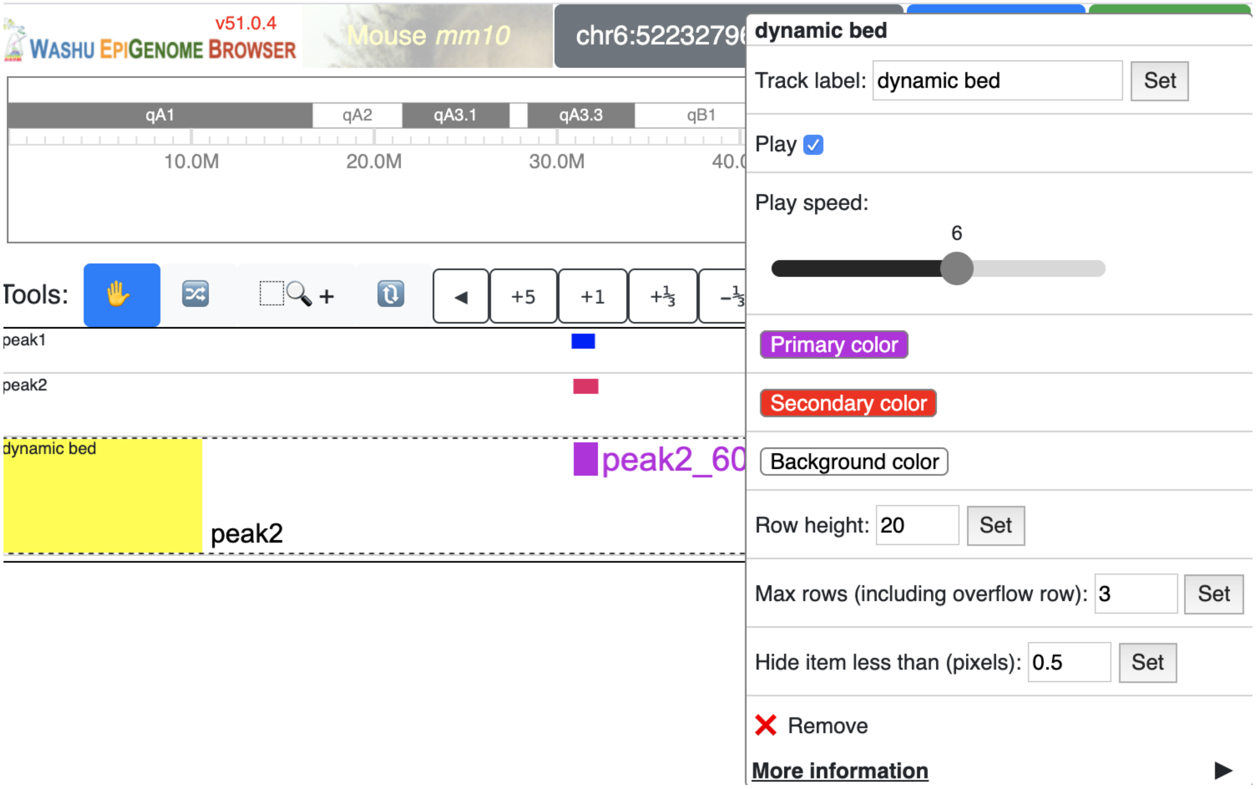
Track label: [multiple values]

Display mode: FULL

Max rows (including overflow row): 20

Hide item less than (pixels):

a new `dynamicbed` track will be added, right click on it will give you the configuration options:



dynamic bed

Track label: dynamic bed Set

Play ☑

Play speed: 6

Row height: 20 Set

Max rows (including overflow row): 3 Set

Hide item less than (pixels): 0.5 Set

An animated version is displayed below:

7.8 Make dynamic bed track using data hub

The dynamic bed track shown above can also be submitted using data hub function, prepare a datahub file like below, and submit it as a remote data hub:

```
[
{
    "type": "dynamicbed",
    "name": "dynamic bed",
    "showOnHubLoad": true,
    "tracks": [
    {
        "type": "bed",
        "url": "https://vizhub.wustl.edu/public/misc/dynamicTrack/bed/peak1.bed.gz",
        "name": "peak1"
    },
    {
        "type": "bed",
        "url": "https://vizhub.wustl.edu/public/misc/dynamicTrack/bed/peak2.bed.gz",
        "name": "peak2"
    }
    ]
}
```

(continues on next page)

(continued from previous page)

```
]
```

7.9 Dynamic track options

Besides regular propeties like `color`, `backgroundColor` and `height` etc, dynamic track has a set of propeties just for this track type.

7.9.1 playing

`playing` indicates if the track animation is playing or paused, value can be *true* or *false*

7.9.2 speed

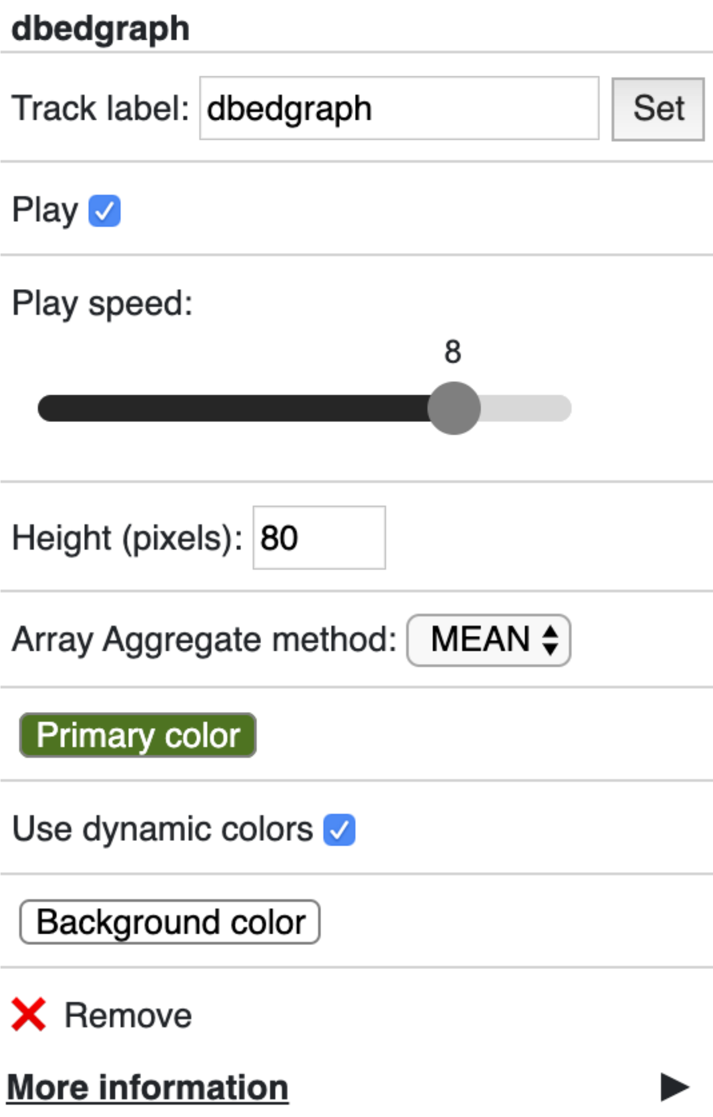
`speed` indicates the playing speed of the animation, range from 1 to 10 where 1 is the slowest and 10 is the fastest. Value need be set in an array format, like `[1]` or `[5]`

7.9.3 dynamicLabels

for `dbedgraph` and `dynamicplot` track types. Specify the labels with each data points. Values should be an array of strings. `dynamicplot`, `dynamichic` `dynamicbed` by default the dynamic track will use the label of each member track. `dynamicplot`'s default dynamic lables can be overwritten by using `dynamicLabels`.

7.9.4 useDynamicColors

`useDynamicColors` toggles if use a dynamic color set defined in data hub, see the option `dynamicColors` for more details. Right click on a dynamic track will also bring the menu to change this option.



dbedgraph

Track label: dbedgraph

Play ☑

Play speed: 8

Height (pixels): 80

Array Aggregate method: MEAN

Use dynamic colors ☑

7.9.5 dynamicColors

For each step of the animation, user can also set different colors for each step. `dynamicColors` is used for this purpose. Check this example data hub below and an animated track display:

```
[
{
    "type": "dbedgraph",
    "url": "https://wangftp.wustl.edu/~dli/test/a.dbg.gz",
    "options": {
        "dynamicLabels": ["stage1","stage2","stage3","stage4","stage5","stage6","stage7","stage8","stage9","stage10"],
        "dynamicColors": ["red", "blue", "#00FF00", 0x000000],
        "useDynamicColors": true
    },
    "showOnHubLoad": true
    }
```

(continues on next page)

(continued from previous page)

```
]
```

Warning: in order for `dynamicColors` to be effect, `useDynamicColors` need set to be `true`. *color* in the array can be color name, or any CSS color, or color hex number. If `useDynamicColors` is `false`, the `color` attribute in options will be used to paint the animation.

Hint: `dynamicColors` with `useDynamicColors` will overwrite the `color` or `color2` settings, once `useDynamicColors` was set to *false*, the color set in `color` will be used.

CHAPTER

EIGHT

VIEW 3D STRUCTURE

8.1 3D visualization video tutorial on YouTube

Note: If you cannot access YouTube, we also put the video tutorial on Bilibili, please .

8.2 g3d format

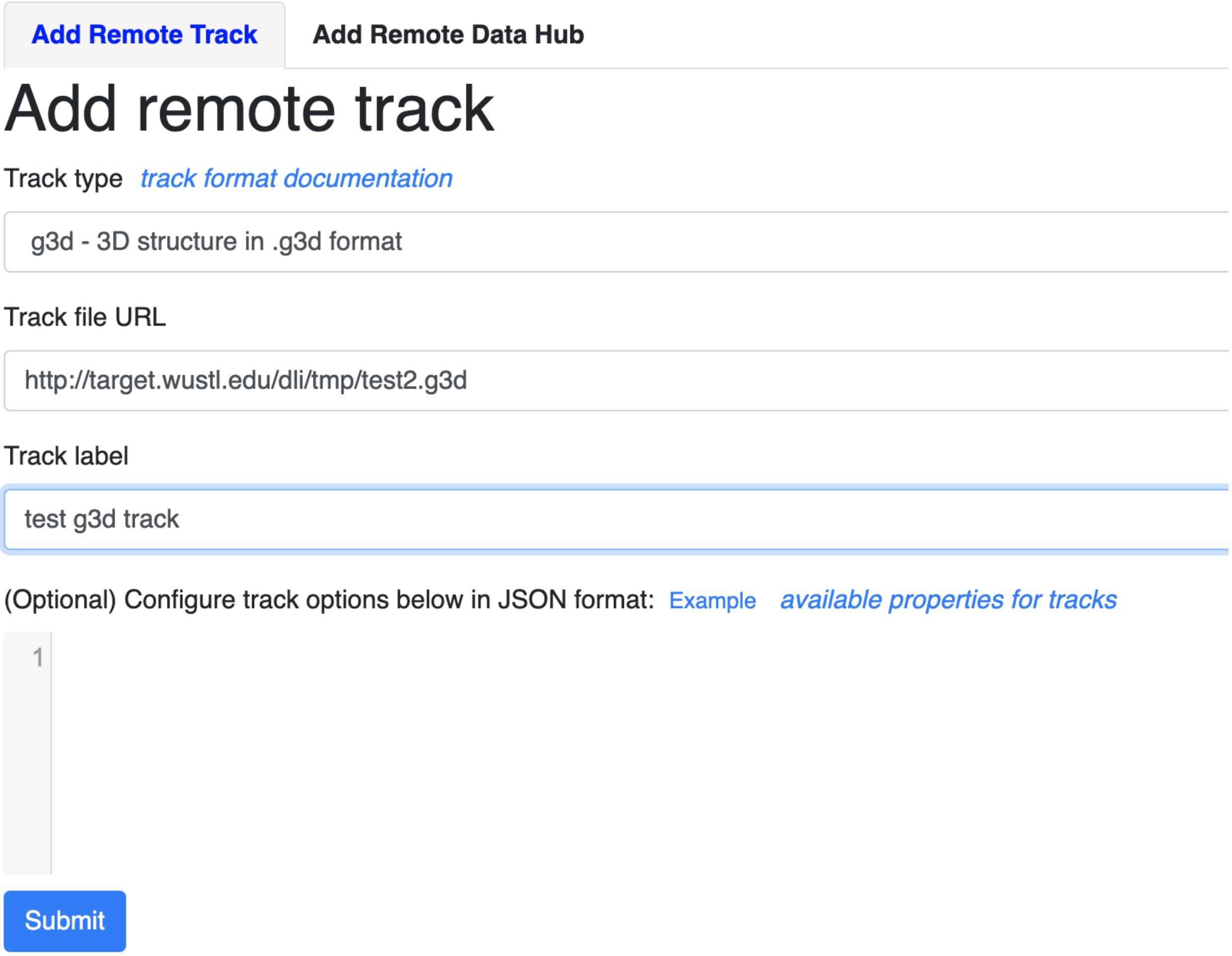
3D genomic structure data can also be displayed at the browser. `g3d` is a new format we developped for visualizing 3D structure data in the Browser. `g3d` format is a binary format based in bed-like format of data contains x, y, z coordinates for each genomic bin. Documentations for how to prepare .g3d file is available at g3dtools documentation.

G3d files can be submitted as custom tracks from `Tracks -> Custom Tracks`, or using a datahub. Submitting a `g3d` track will trigger a new panel opened in the browser, which also contains menu allows you to customize the visulization, like change resolution and decorating the 3D structure using bigwig (like GC percentage) or compartment annotations.

8.3 g3d track

From *Tracks* menu, choose *Remote Tracks*, choose *g3d* track type and paste the g3d file url, you can also input a track label which is optional.

Note: you can use our example file for testing purpose, this file is converted from data published in Three-dimensional genome structures of single diploid human cells, Science Vol. 361, Issue 6405, pp. 924-928 and the original data was obtained from NCBI GEO database.



Add remote track

Track type *track format documentation*

g3d - 3D structure in .g3d format

Track file URL

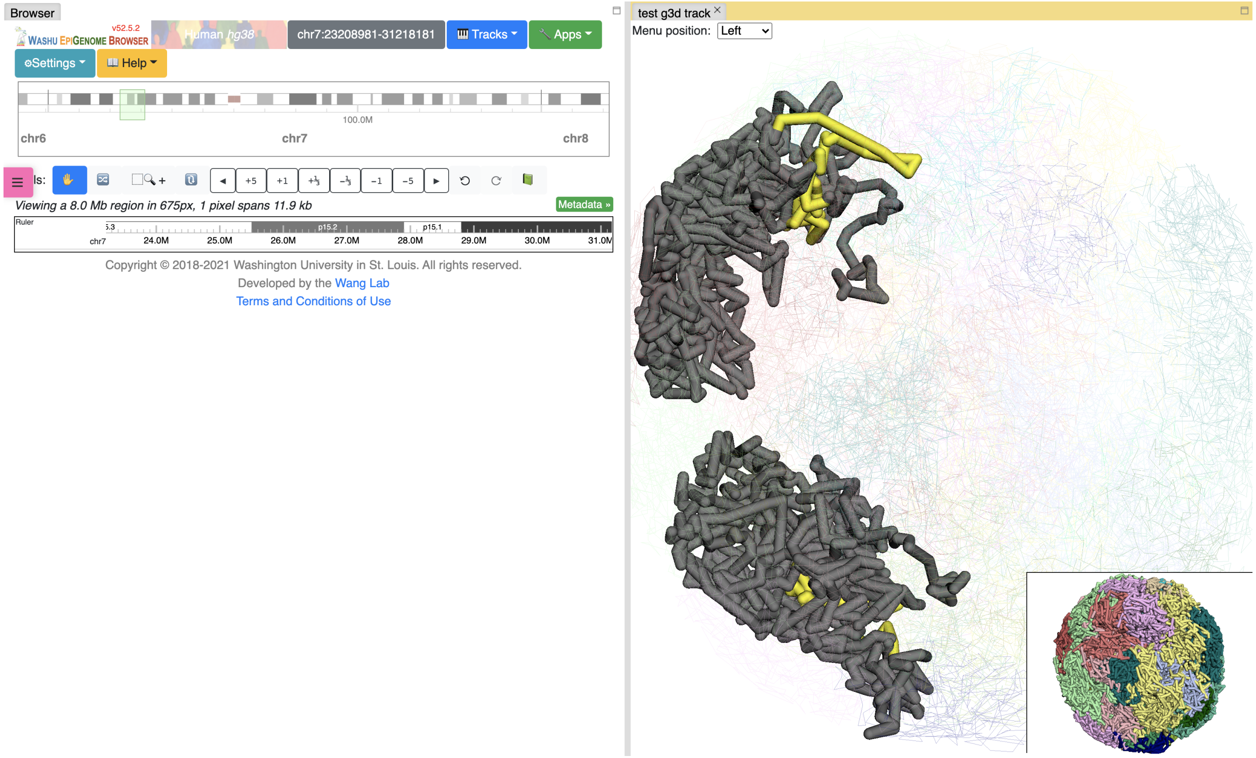
http://target.wustl.edu/dli/tmp/test2.g3d

Track label

test g3d track

(Optional) Configure track options below in JSON format: Example *available properties for tracks*

click the *Submit* button, close the Remote track panel, this is how it looks like with default view:



Menu position: Left

Viewing a 8.0 Mb region in 675px, 1 pixel spans 11.9 kb

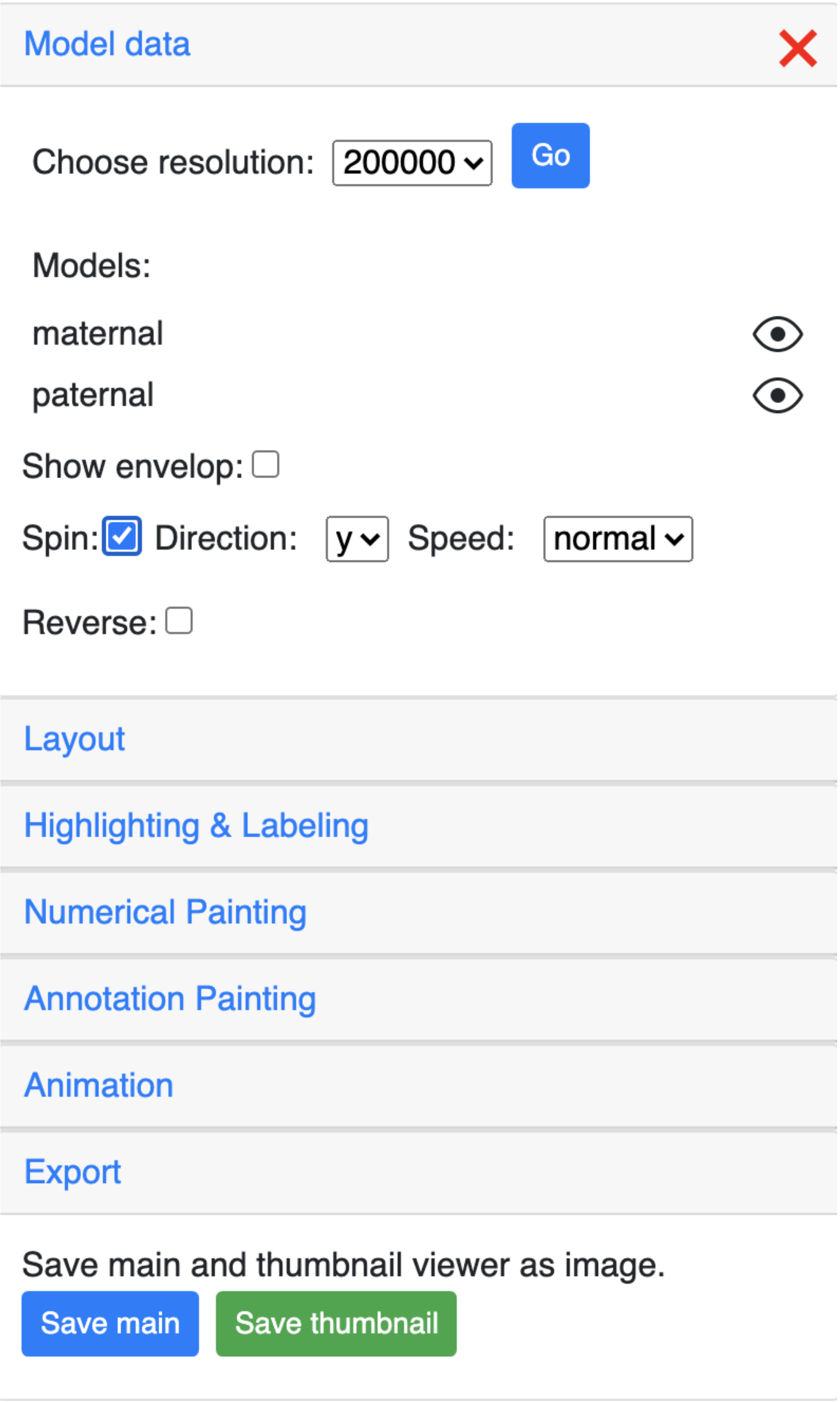
Copyright © 2018-2021 Washington University in St. Louis. All rights reserved.
Developed by the Wang Lab
Terms and Conditions of Use

By default the 3D viewer contains main and thumbnail viewer, if you zoom/rotate in one of them, both of them will be synchronized. Yellow highlighted region indicates browser region, change region in browser will also change the highlighted region in yellow.

Note: by default the model will spin after intially loaded, you can disable the spin in configuration menu. Please see the instructions in the sections below.

8.4 3D viewer menu

Clicking the Open menu button will open the configuration menu for the 3D visulization, which by default floats to the left of the screen, the menu is grouped to control the model data, layout, highlighting & labeling, painting, animation and export. Each group can be clicked to toggle expansion.



Choose resolution: 200000

Models:

maternal

paternal

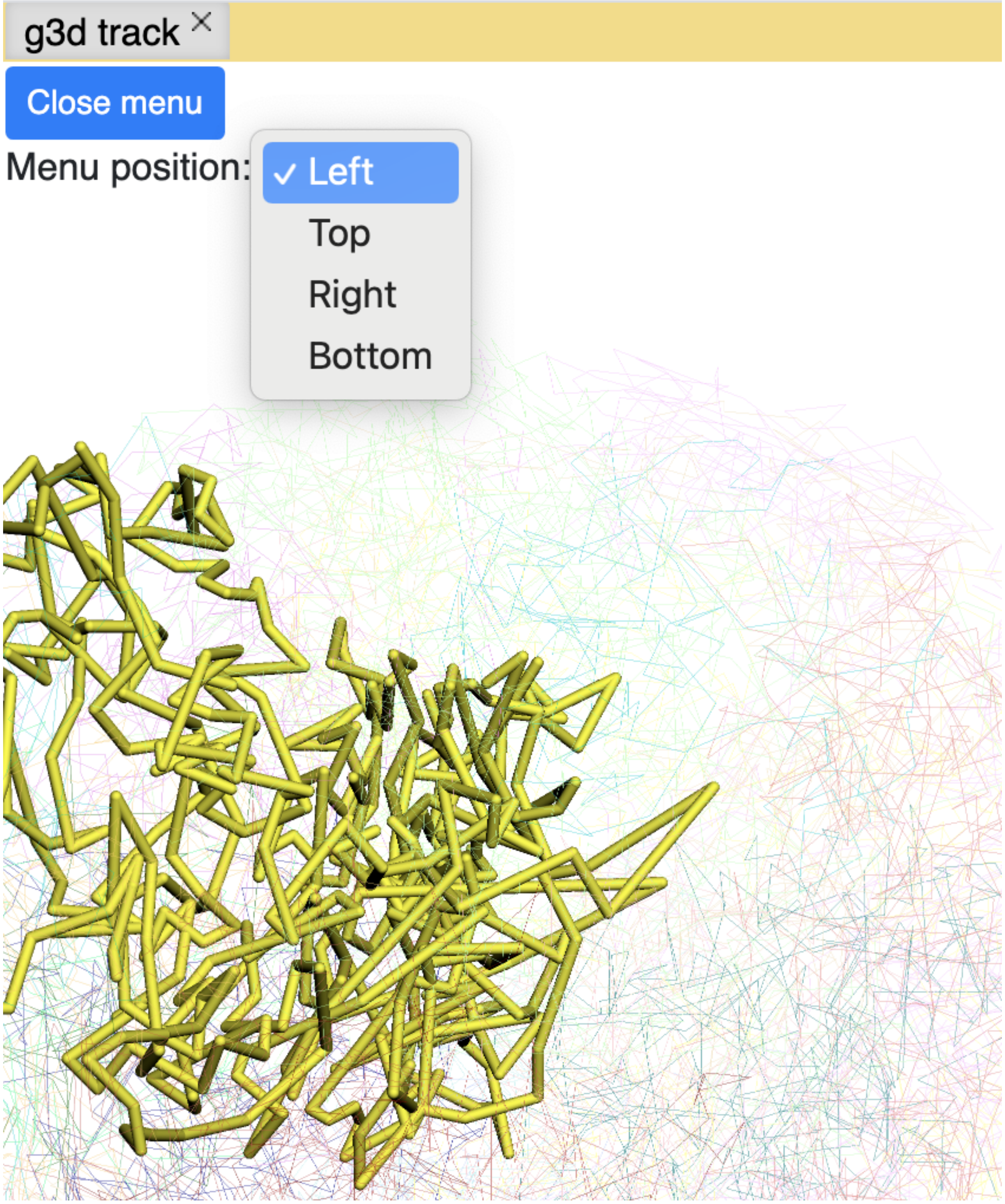
Show envelop: □

Spin: ☑ Direction: y Speed: normal

Reverse: □

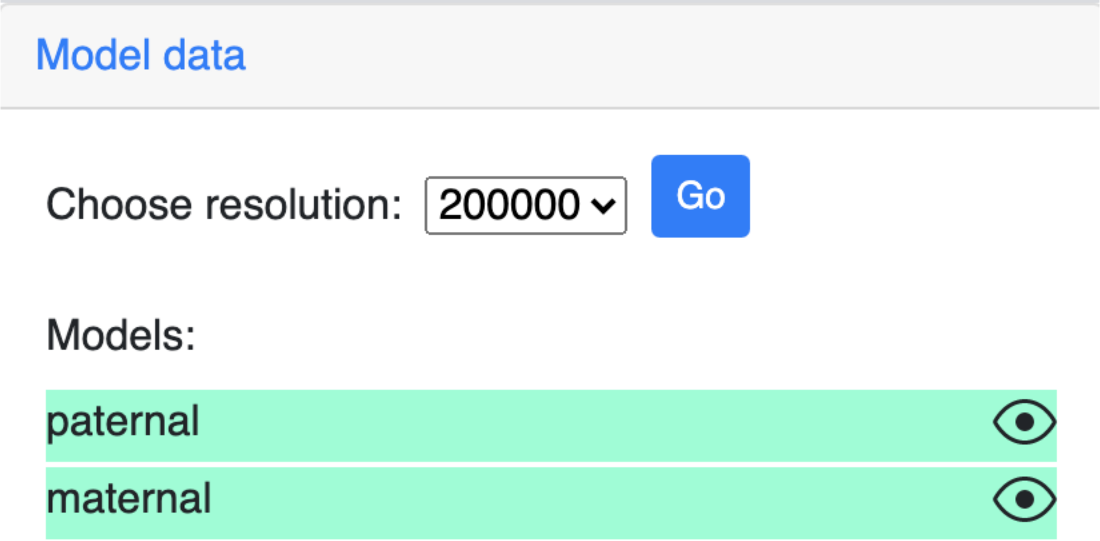
Save main and thumbnail viewer as image.

The menu icon position can also be adjusted using the dropdown menu on the 3D viewer:



8.5 Config 3D model data

The *Model data* section can control the resolution of the g3d data.



Model data

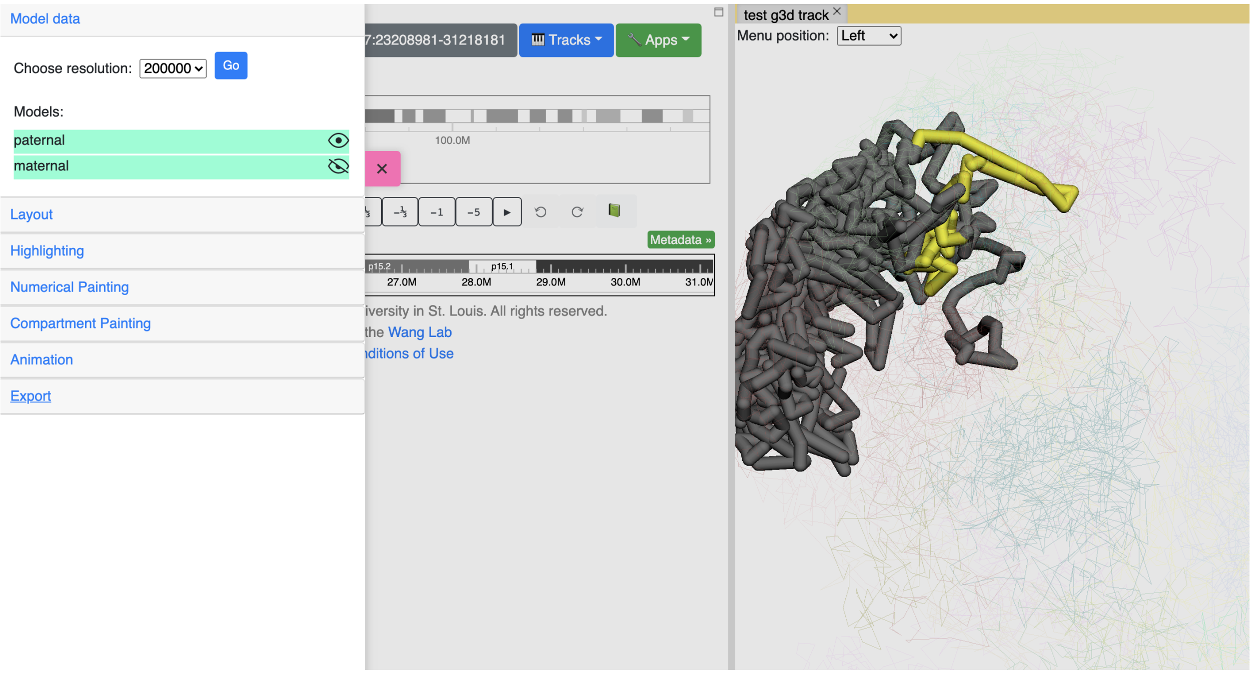
Choose resolution: 200000

Models:

paternal

maternal

All the models in the g3d file will be listed here and can be displayed or hidden using the eye icon button. The example screenshot below indicates the *maternal* model is hidden by click the its eye icon:



Model data

Choose resolution: 200000

Models:

paternal

maternal

test g3d track

Menu position: Left

7:23208981-31218181

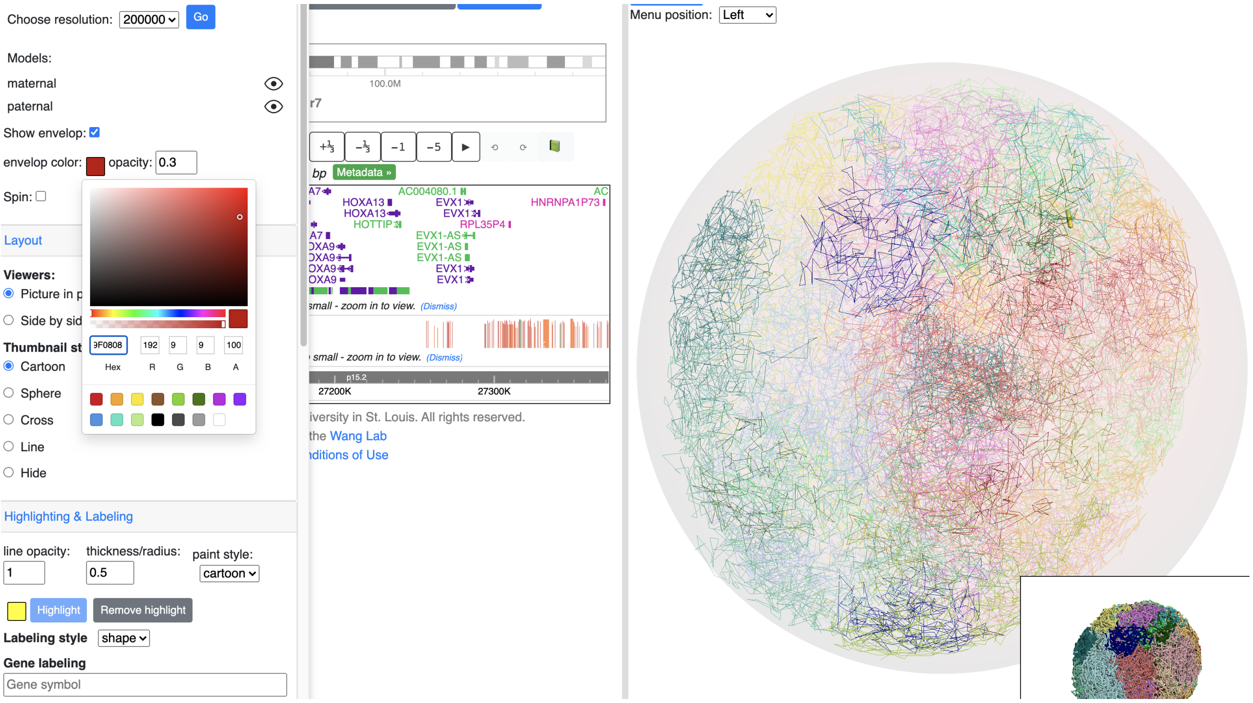
iversity in St. Louis. All rights reserved.

the Wang Lab

ditions of Use

8.5.1 Show nuclear envelop

The 3d viewer uses an outer sphere to mimic the display of the nuclear envelop. Envelop can be configed to show or hide, color and opacity can also be customized.



Choose resolution: 200000 Go

Models:

maternal

paternal

Show envelop: ☑

envelop color: 🟥 opacity: 0.3

Spin: □

Layout

Viewers:

● Picture in p

○ Side by sid

Thumbnail st

● Cartoon

○ Sphere

○ Cross

○ Line

○ Hide

Highlighting & Labeling

line opacity: 1 thickness/radius: 0.5 paint style: cartoon

Labeling style shape

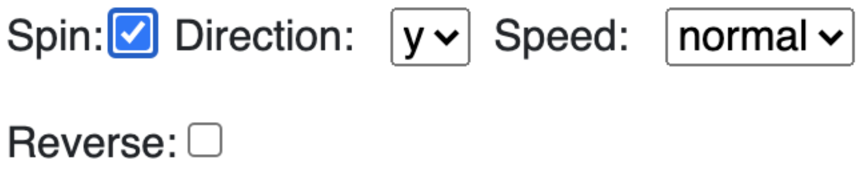
Gene labeling

Gene symbol

Menu position: Left

8.5.2 Config the spin of 3d model

The model will spin after intially load, the spin section can used to toggle the spin status, and control the direction, speed of the spin.

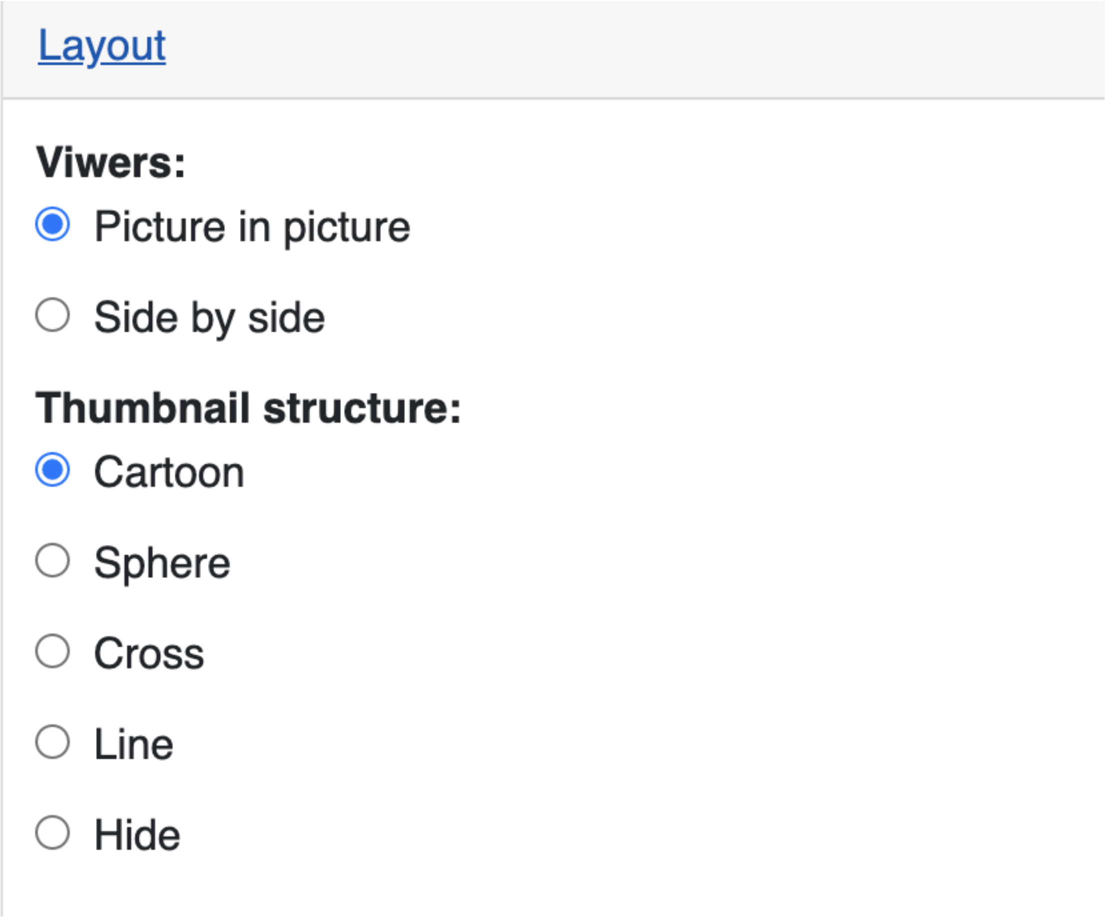


Spin: ☑ Direction: y Speed: normal

Reverse: □

8.6 Config 3D viewer layout

The *Layout* section is used to control the layout of main and thumbnail viewer.



Layout

Viwers:

● Picture in picture

○ Side by side

Thumbnail structure:

● Cartoon

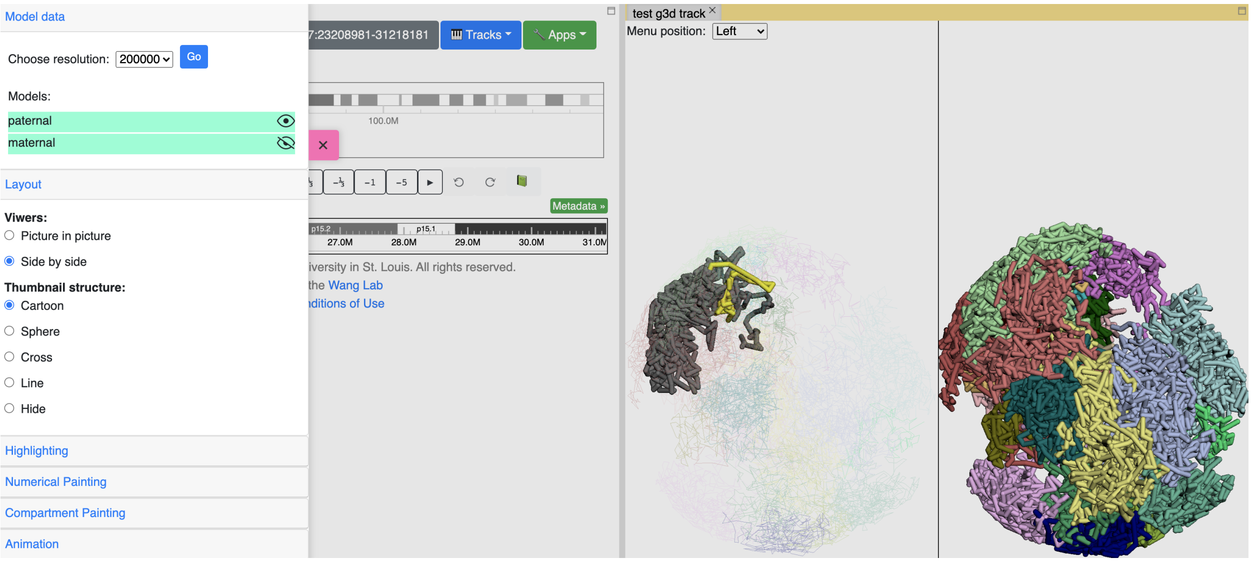
○ Sphere

○ Cross

○ Line

○ Hide

Change the view layout to *side by side*:



Model data

Choose resolution: 200000 Go

Models:

paternal

maternal

Layout

Viwers:

○ Picture in picture

● Side by side

Thumbnail structure:

● Cartoon

○ Sphere

○ Cross

○ Line

○ Hide

Highlighting

Numerical Painting

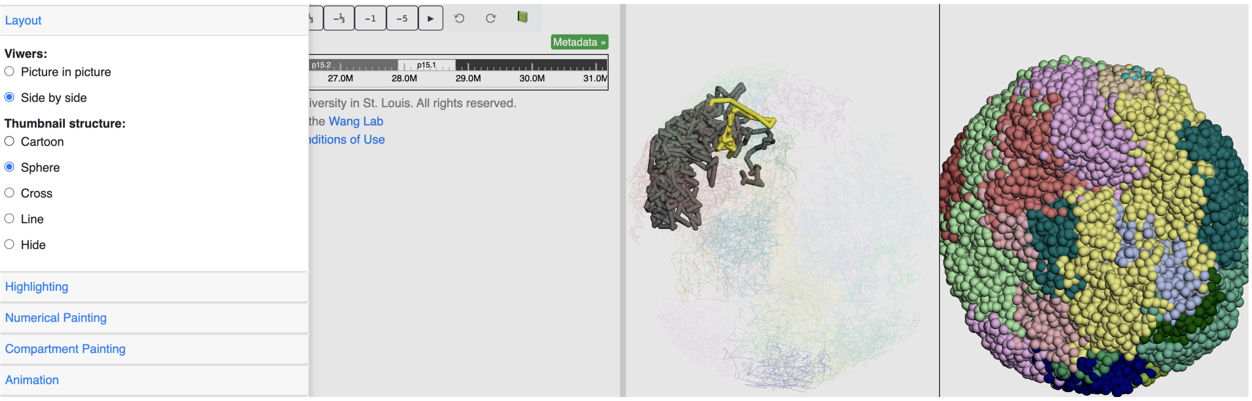
Compartment Painting

Animation

test g3d track

Menu position: Left

You can also change how the thumbnail structure looks like, for example, *sphere* style as below:



Layout

Viwers:

- ○ Picture in picture
- ● Side by side

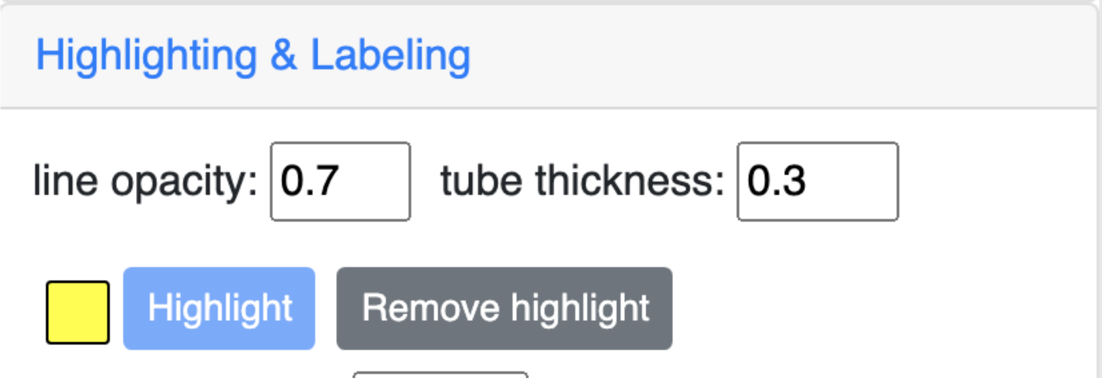
Thumbnail structure:

- ○ Cartoon
- ● Sphere
- ○ Cross
- ○ Line
- ○ Hide

8.7 Highlighting & labeling

8.7.1 Toggle browser region highlighting

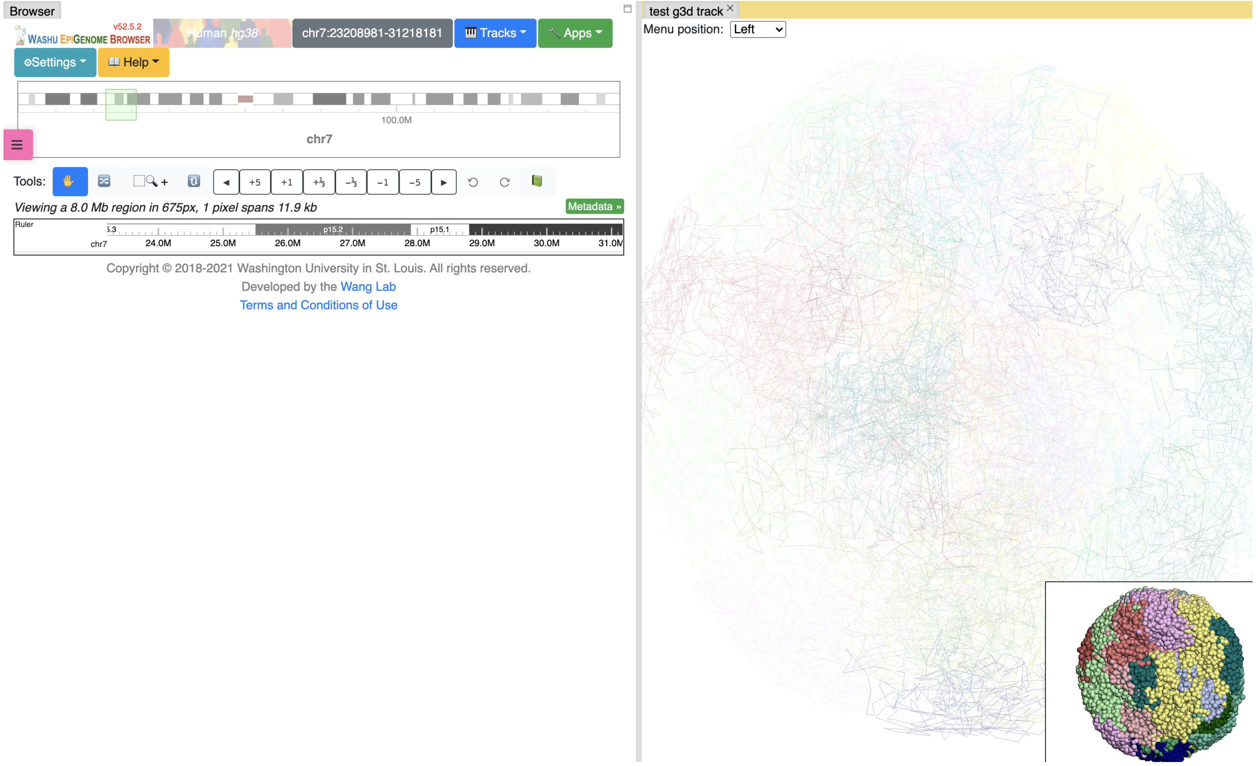
By default the main viewer would highlight the structure part belongs to current browser region in yellow, the *Highlighting* section is used to control this behaviour, click the *Remove highlight* will turn off the highlighting.



Highlighting & Labeling

line opacity: 0.7 tube thickness: 0.3

This is how it looks like when the highlighting is turned off:



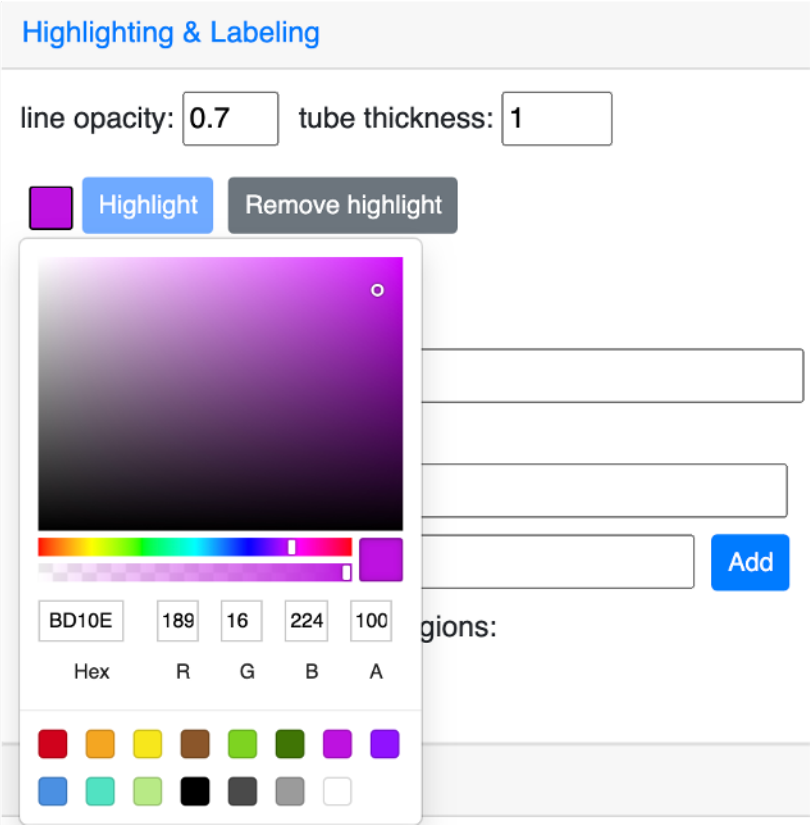
Menu position: Left

Viewing a 8.0 Mb region in 675px, 1 pixel spans 11.9 kb

Copyright © 2018-2021 Washington University in St. Louis. All rights reserved.
Developed by the Wang Lab
Terms and Conditions of Use

8.7.2 Customize highlighting

Highlighting color and tube thickness can be customized to get a different viewer. As shown below, we changed the color to purple and thickness to 1:

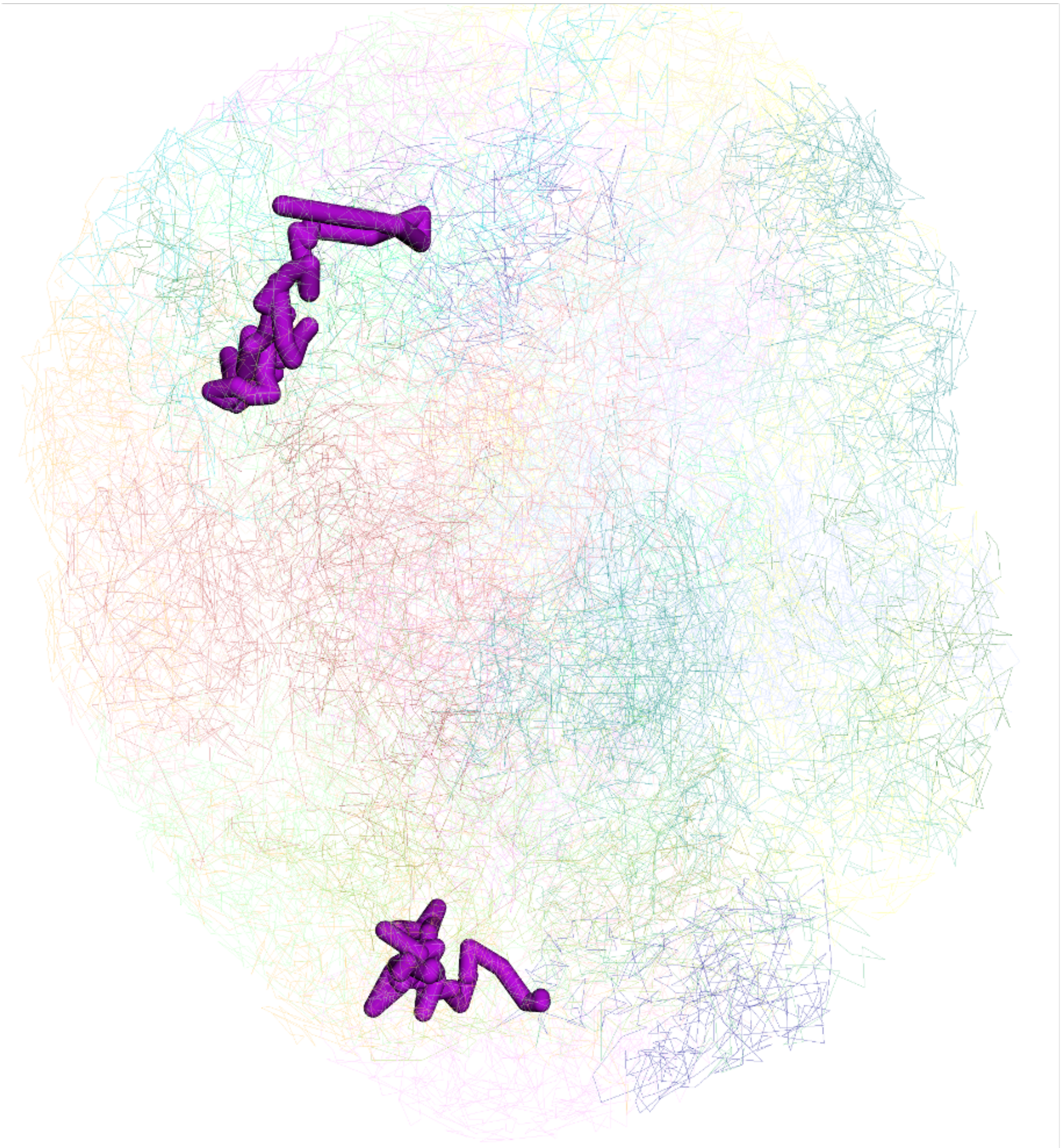

Highlighting & Labeling

line opacity: 0.7 tube thickness: 1

BD10E 189 16 224 100

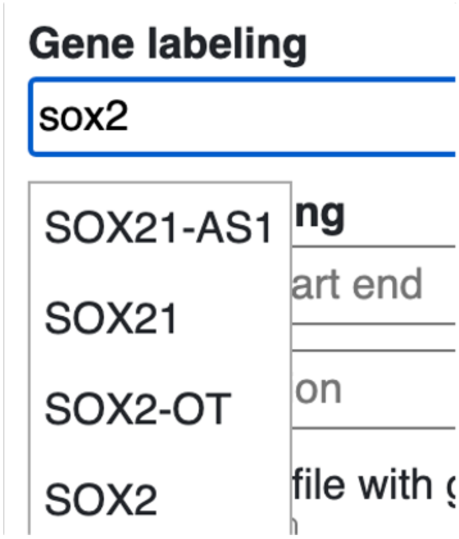
Hex R G B A

and this is the updated and view:



8.7.3 Labeling by gene

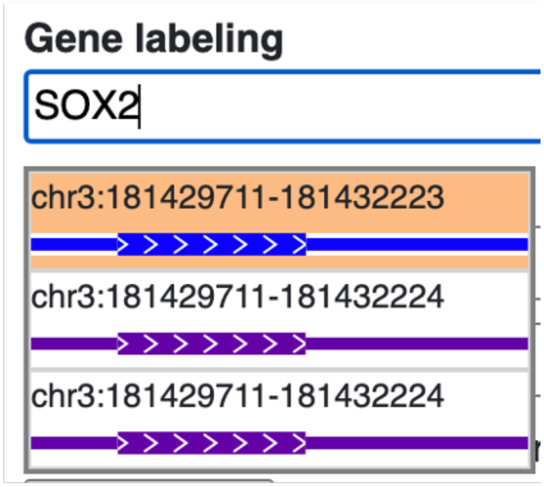
Gene symbol can be searched for labeling, start with search any gene symbol, the menu will auto complete the search based on users' input.



Gene labeling

sox2

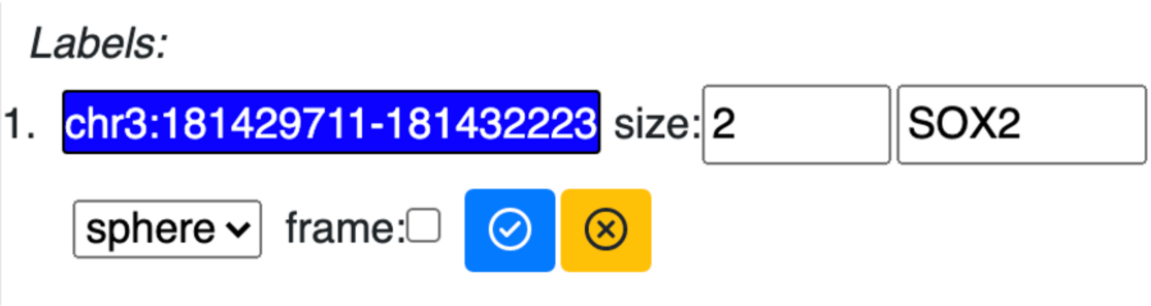
choose the isoform wanted:



Gene labeling

SOX2

the gene will be added as a new label in the label list:

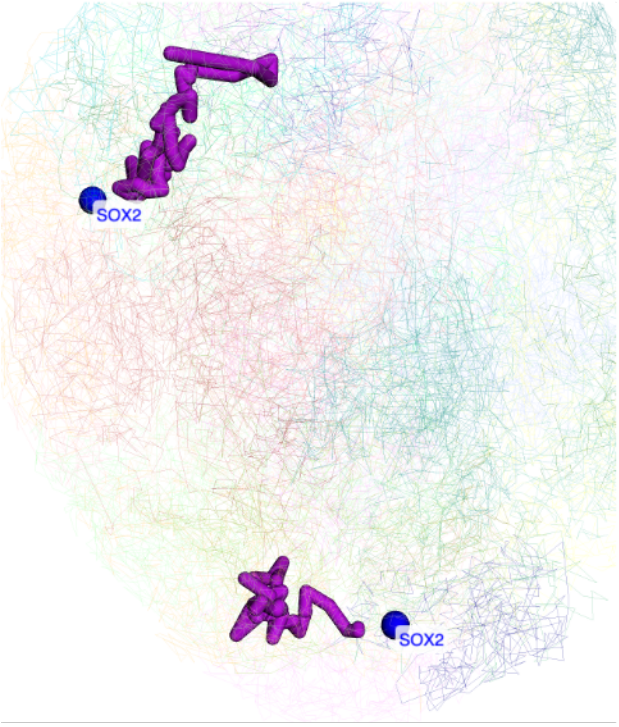


Labels:

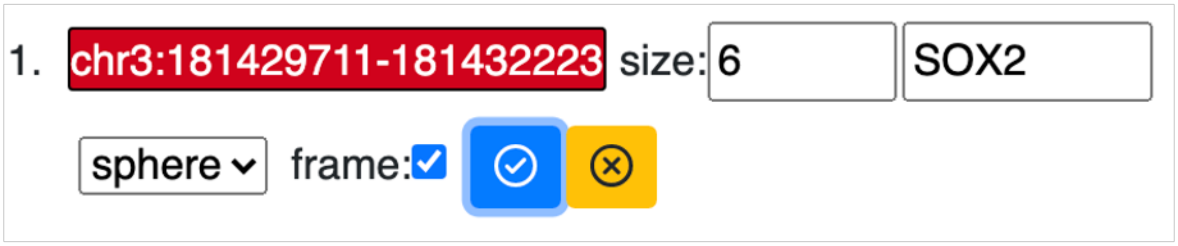
1. chr3:181429711-181432223 size: 2 SOX2

sphere frame: □

and shown in 3D view:



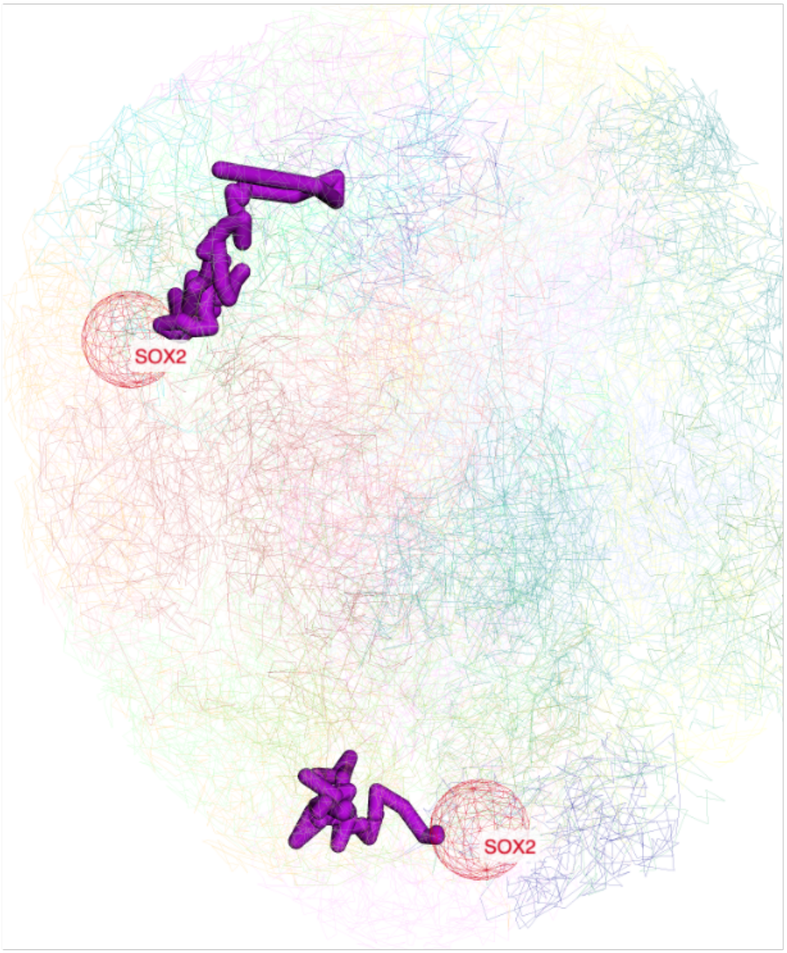
update the display style of the label:



1. chr3:181429711-181432223 size: 6 SOX2

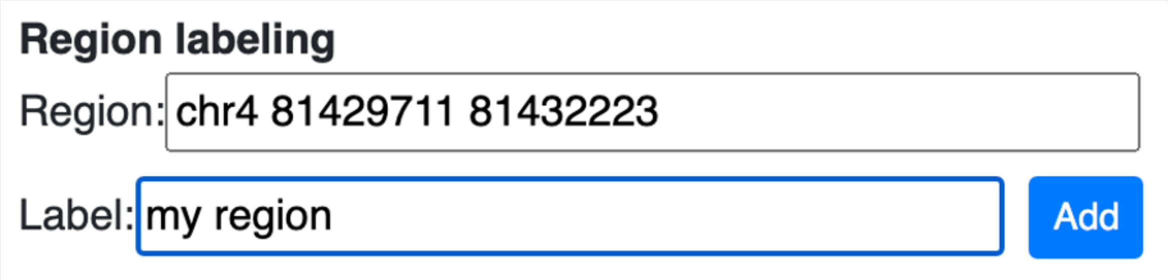
sphere frame: ☑

updated view of the label:



8.7.4 Labeling by region

User can also manully type a region for highlighting:

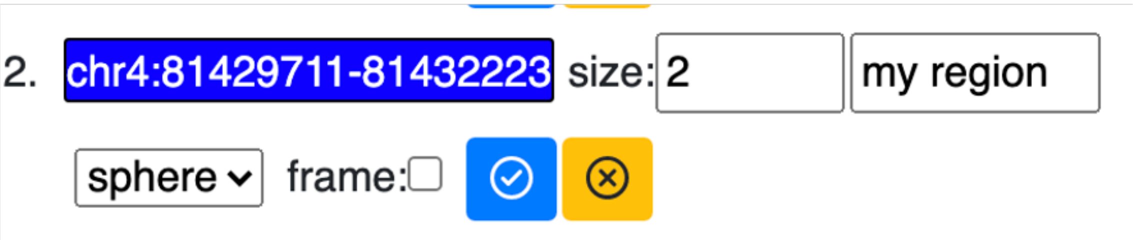


Region labeling

Region: chr4 81429711 81432223

Label: my region

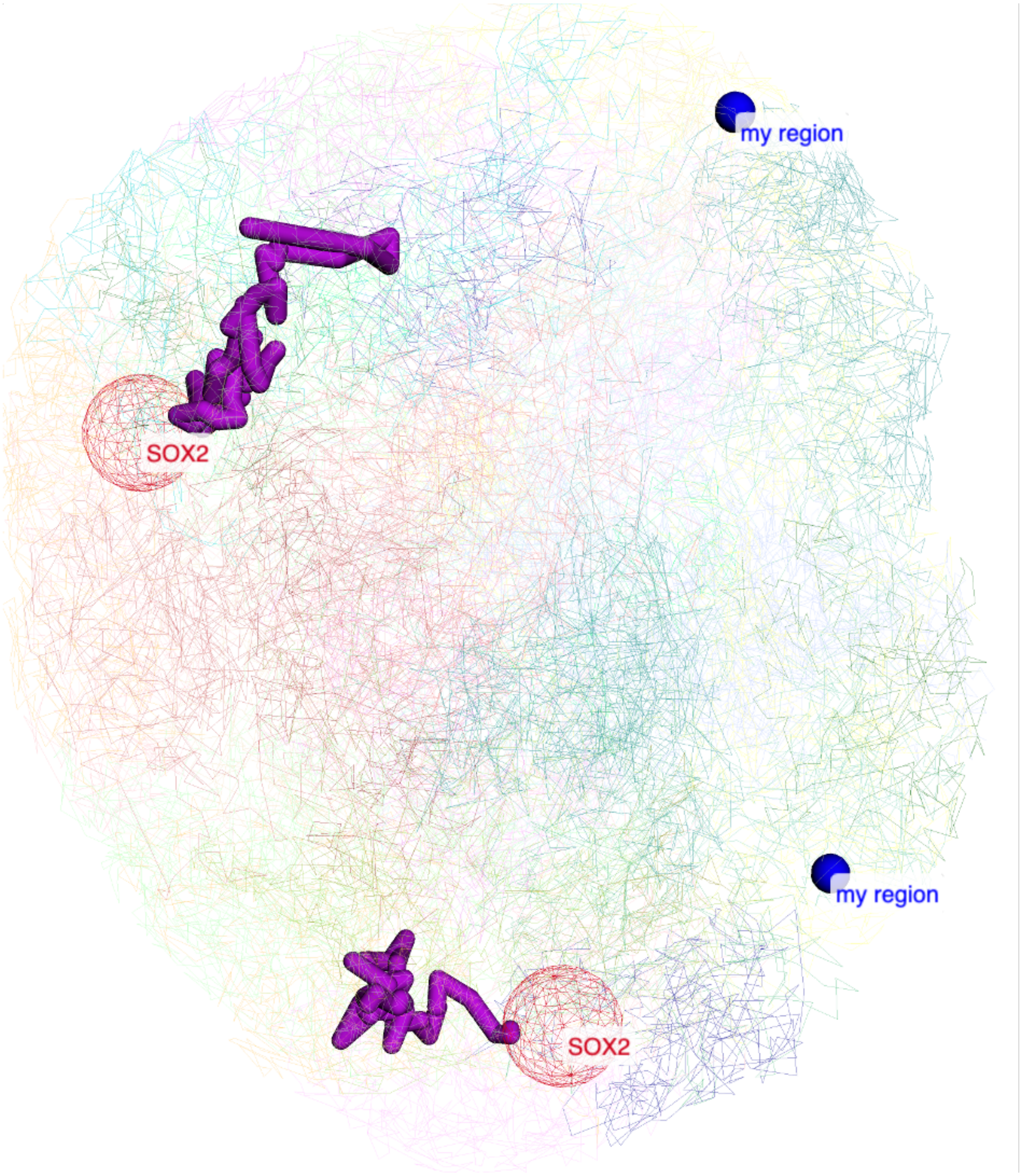
the added label by region search by also be updated in the menu control:



2. chr4:81429711-81432223 size: 2 my region

sphere frame: □

the view after added region label:

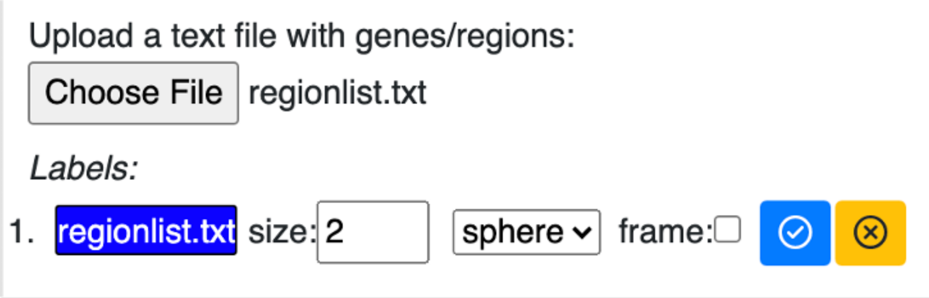


8.7.5 Upload a file for labeling

A text file contains list of regions/gene symbols can also be uploaded for batch labeling, as shown below, the text file contains content:

```
CYP4A22
chr10:96796528-96829254
CYP2A6
CYP3A4
chr1:47223509-47276522
CYP1A2
```

upload this file:



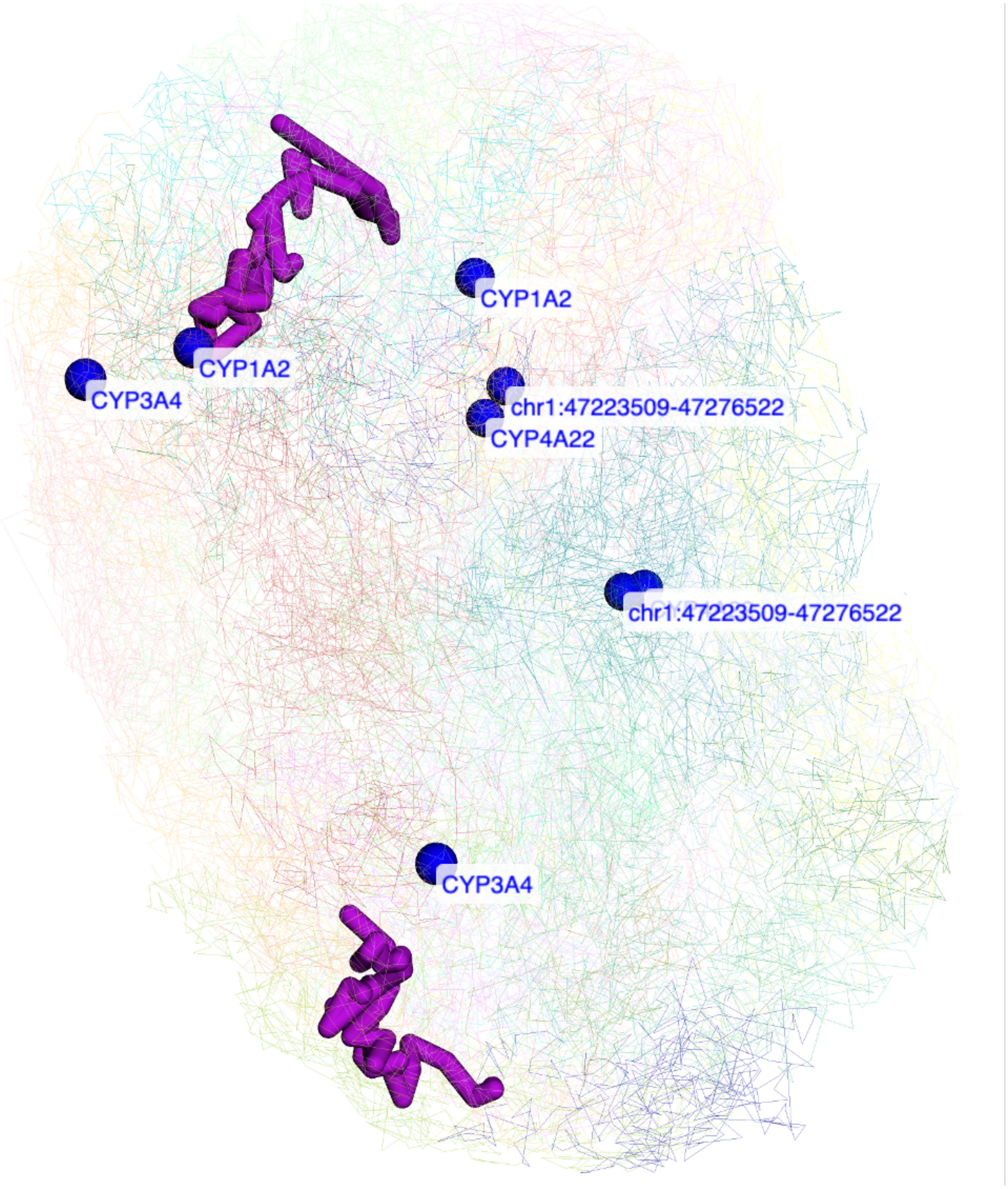
Upload a text file with genes/regions:

Choose File regionlist.txt

Labels:

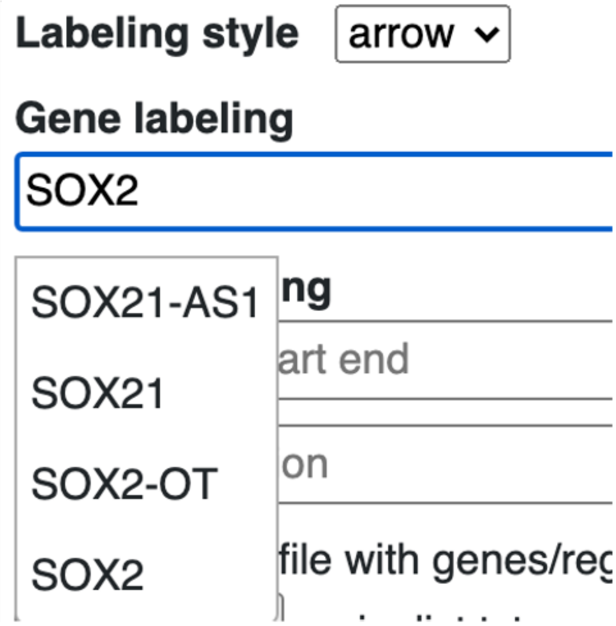
1. regionlist.txt size: 2 sphere frame: □

regions in the file are all labeled:


CYP1A2
CYP1A2
CYP3A4
chr1:47223509-47276522
CYP4A22
chr1:47223509-47276522
CYP3A4

8.7.6 Pointing using arrows

instead of using shapes for labels, arrows can also be used to pointing the region desired. Choose label style as arrow:

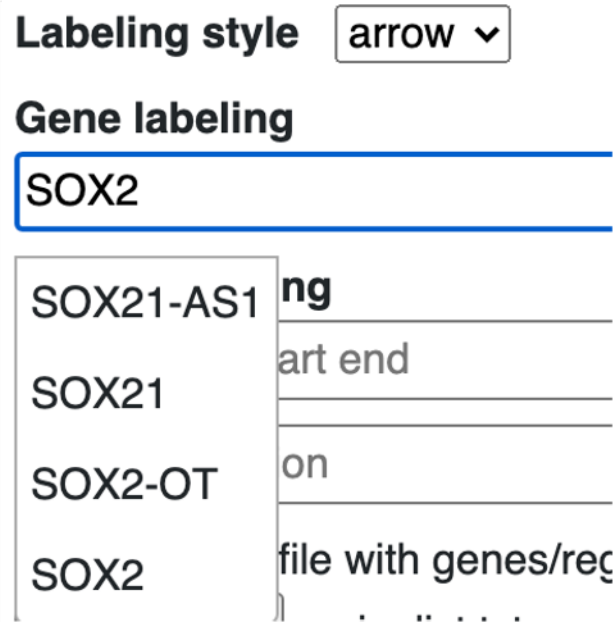


Labeling style arrow

Gene labeling

SOX2

use either gene search or region labeling:

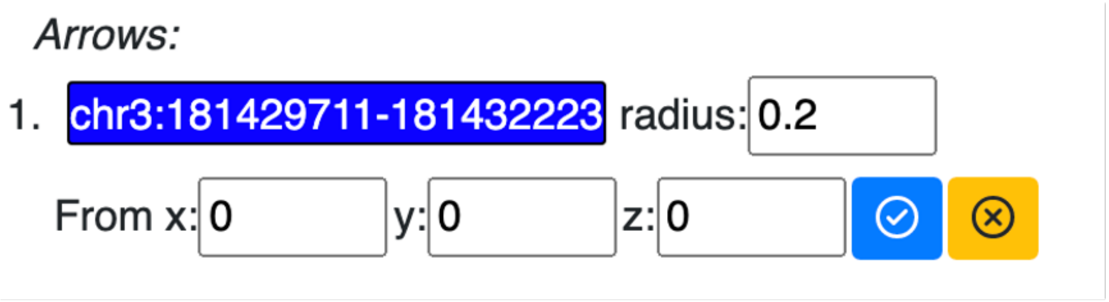


Labeling style arrow

Gene labeling

SOX2

the new added label will be displayed under arrow list:

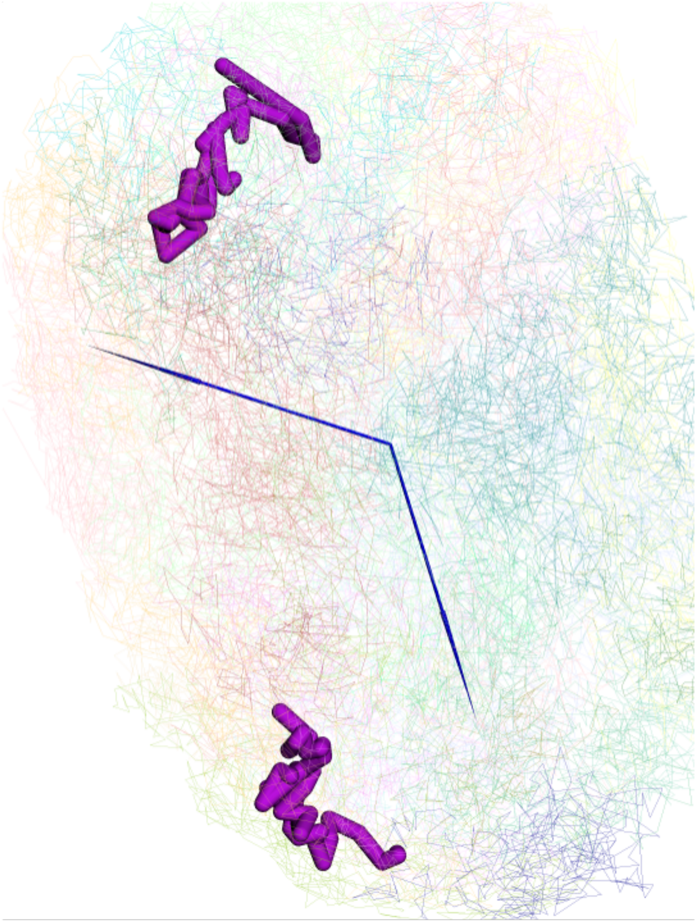


Arrows:

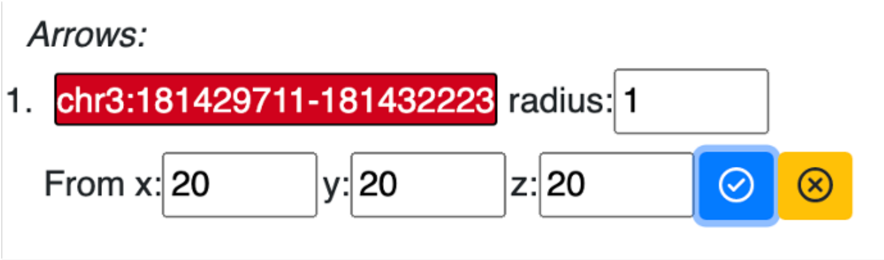
1. chr3:181429711-181432223 radius: 0.2

From x: 0 y: 0 z: 0

and displayed in 3d viewer:



config the style of arrow:

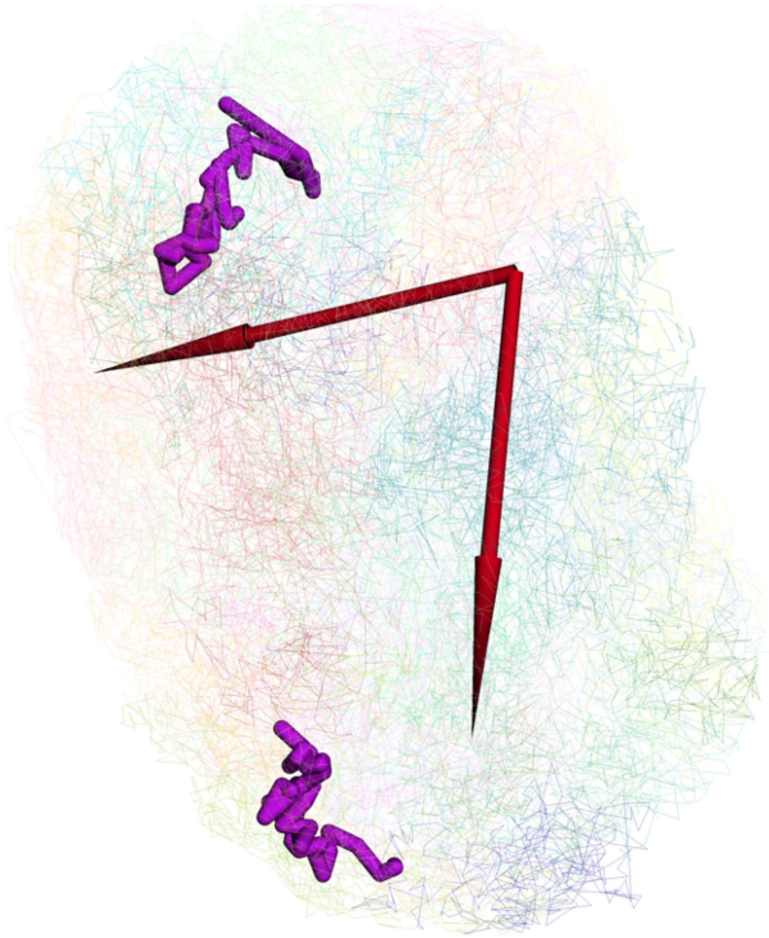


Arrows:

1. chr3:181429711-181432223 radius: 1

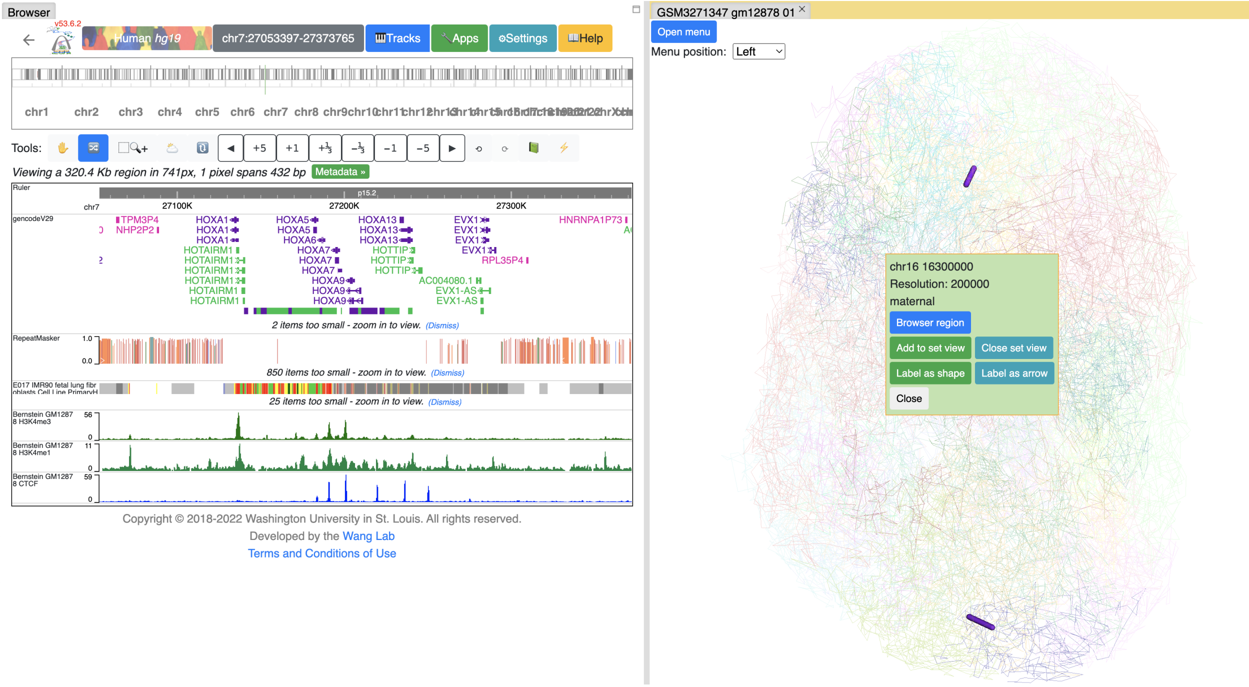
From x: 20 y: 20 z: 20

updated arrow style in the viewer:

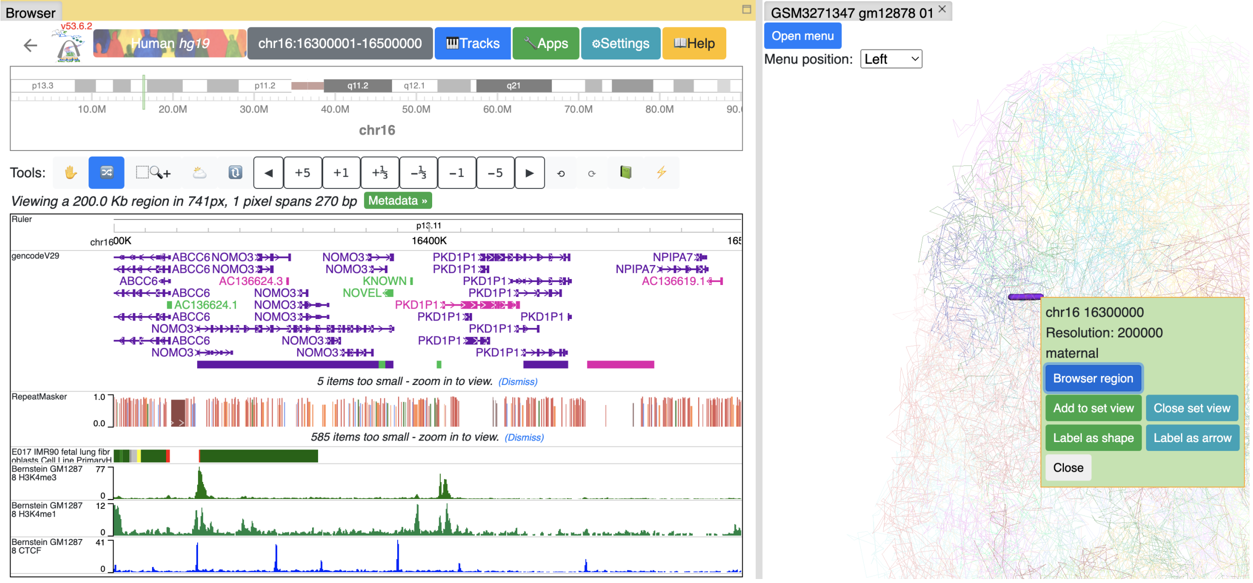


8.8 Interactivity on 3D model

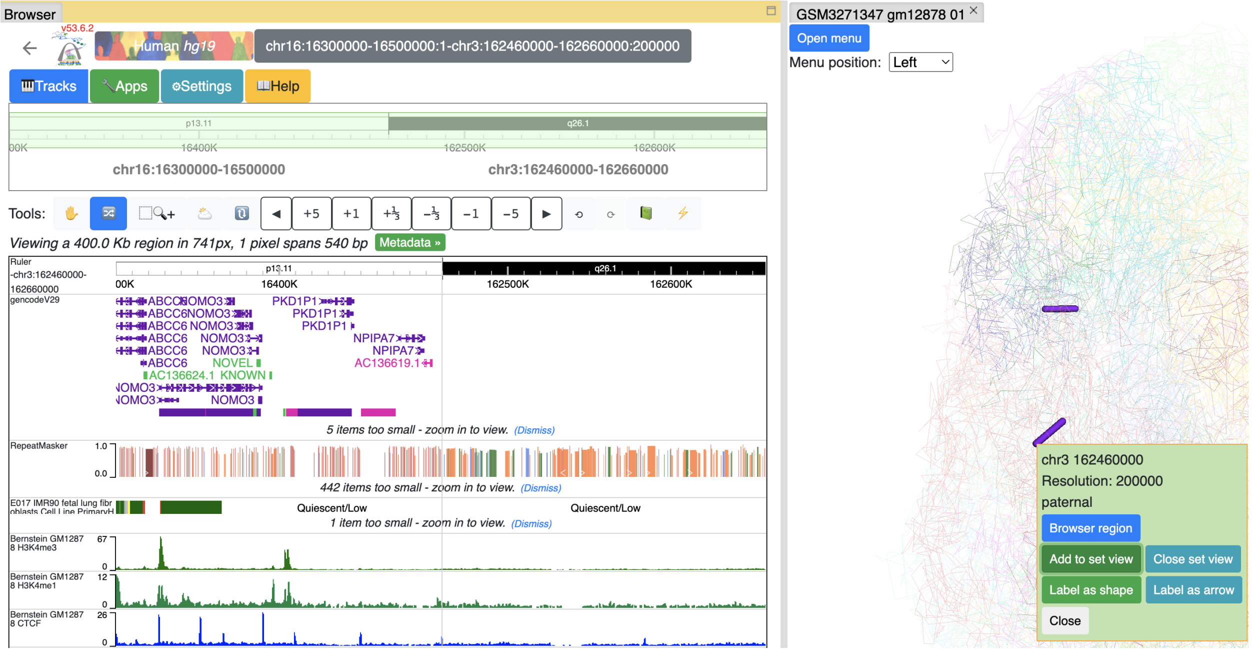
Click on the segments of 3D model will bring up the menu to interact with the Browser:


Browser
v53.6.2
Human hg19
chr7:27053397-27373765
Tracks
Apps
Settings
Help
Tools:
+5
+1
−1
−5
Viewing a 320.4 Kb region in 741px, 1 pixel spans 432 bp
Metadata »
gencodeV29
RepeatMasker
E017 IMR90 fetal lung fibroblasts Cell Line PrimaryH
Bernstein GM12878 H3K4me3
Bernstein GM12878 H3K4me1
Bernstein GM12878 CTCF
2 items too small - zoom in to view. (Dismiss)
850 items too small - zoom in to view. (Dismiss)
25 items too small - zoom in to view. (Dismiss)
Copyright © 2018-2022 Washington University in St. Louis. All rights reserved.
Developed by the Wang Lab
Terms and Conditions of Use
GSM3271347 gm12878 01
Open menu
Menu position: Left
chr16 16300000
Resolution: 200000
maternal
Browser region
Add to set view
Close set view
Label as shape
Label as arrow
Close

Click `Browser region` button will let the browser jump to the region of this segment:



or you can add this segment to current browser reigon by click `Add to set view` button, the browser will enter Region set view mode:


chr16:16300000-16500000:1-chr3:162460000-162660000:200000

Menu position: Left

Viewing a 400.0 Kb region in 741px, 1 pixel spans 540 bp

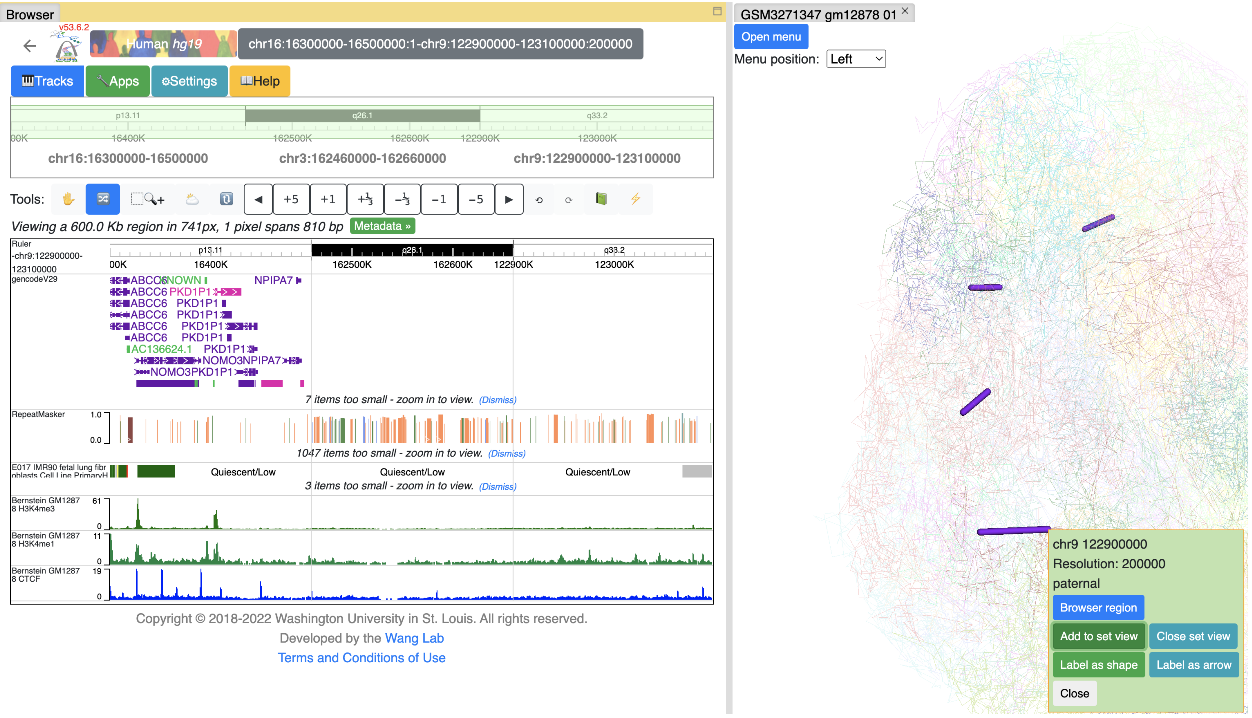
5 items too small - zoom in to view. (Dismiss)

442 items too small - zoom in to view. (Dismiss)

1 item too small - zoom in to view. (Dismiss)

chr3 162460000
Resolution: 200000
paternal

you can keep adding as many segments as you want. Click `Close set view` button to restore to the default view.


chr16:16300000-16500000:1-chr9:122900000-123100000:200000

Menu position: Left

Viewing a 600.0 Kb region in 741px, 1 pixel spans 810 bp

7 items too small - zoom in to view. (Dismiss)

1047 items too small - zoom in to view. (Dismiss)

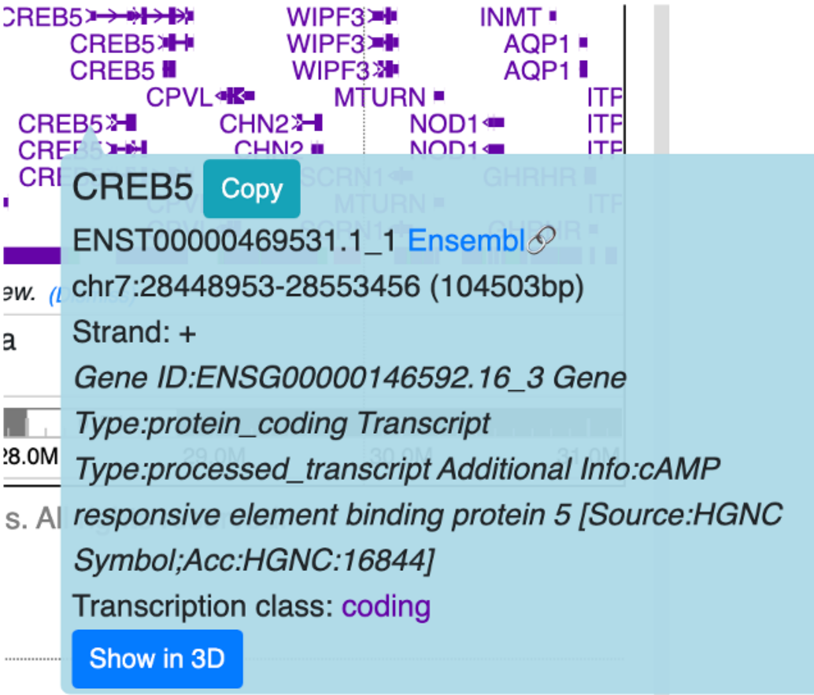
3 items too small - zoom in to view. (Dismiss)

Copyright © 2018-2022 Washington University in St. Louis. All rights reserved.
Developed by the Wang Lab
Terms and Conditions of Use

chr9 122900000
Resolution: 200000
paternal

8.9 Interactivity on tracks

From certain track types like gene and HiC track, users can choose to display gene or HiC anchors on 3D structure directly. As shown below, the tooltip of the gene has `Show in 3D` button, click it will add this gene to the label list and highlight it in 3D view:



CREB5

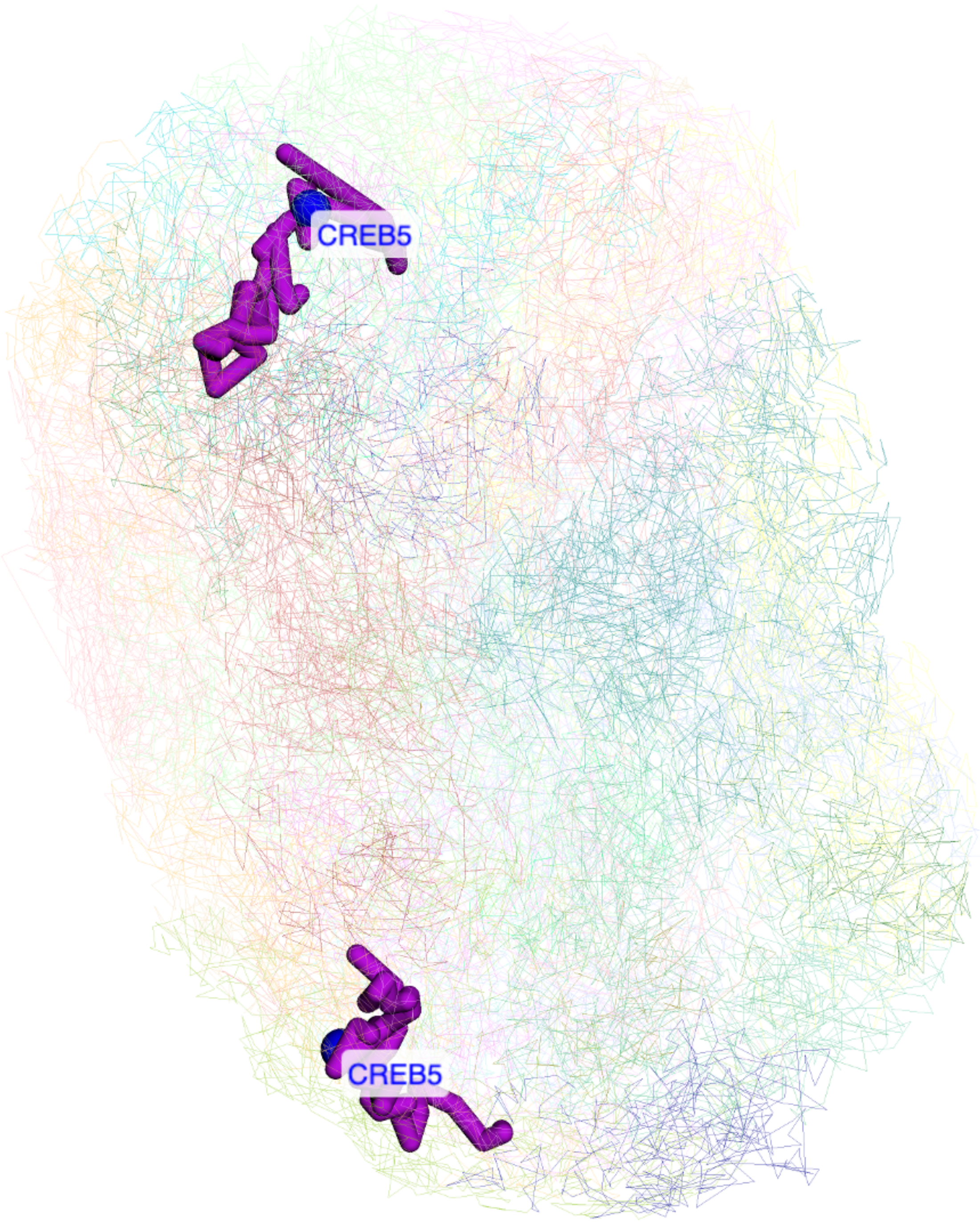
ENST00000469531.1_1 Ensembl

chr7:28448953-28553456 (104503bp)

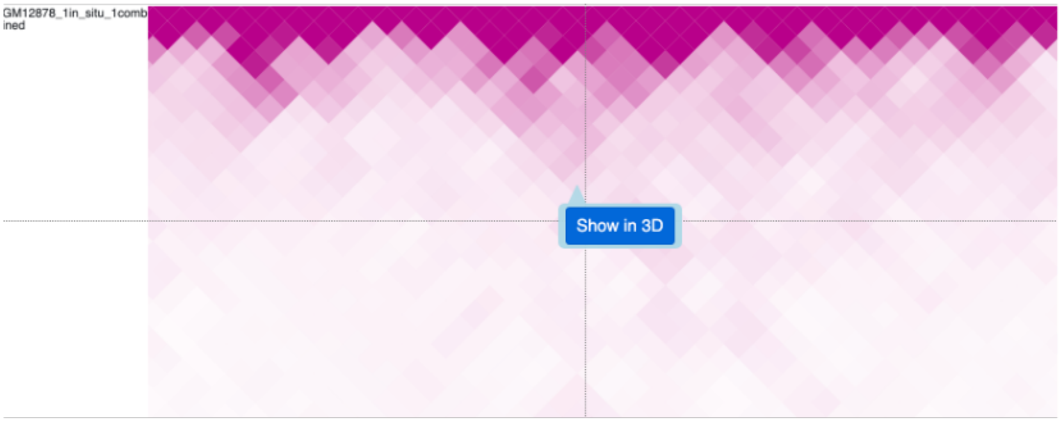
Strand: +

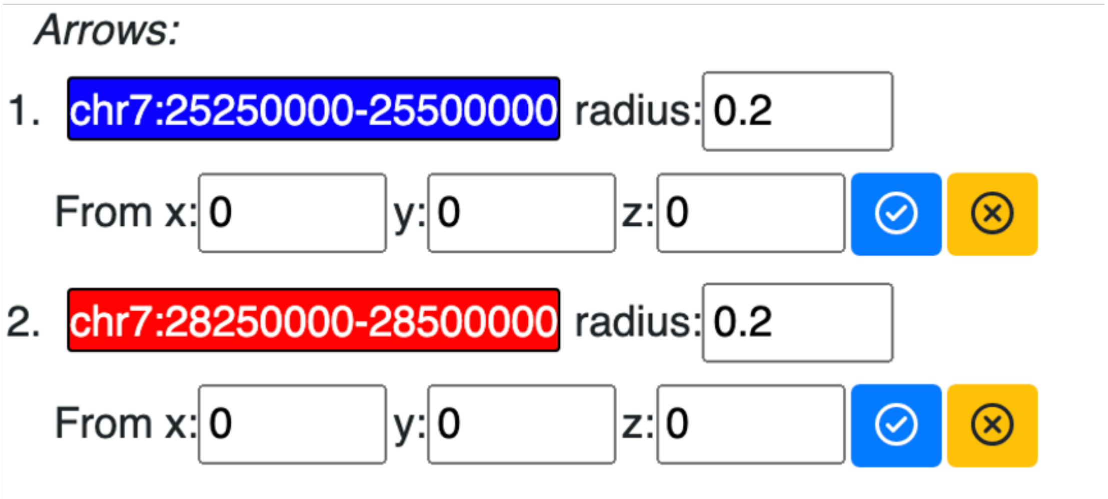
Gene ID:ENSG00000146592.16_3 Gene Type:protein_coding Transcript Type:processed_transcript Additional Info:cAMP responsive element binding protein 5 [Source:HGNC Symbol;Acc:HGNC:16844]

Transcription class: coding



Clicking any dimond on a HiC track will also bring the `Show in 3D` button, click it will add both anchors of this contact to the arrow list by default:





Arrows:

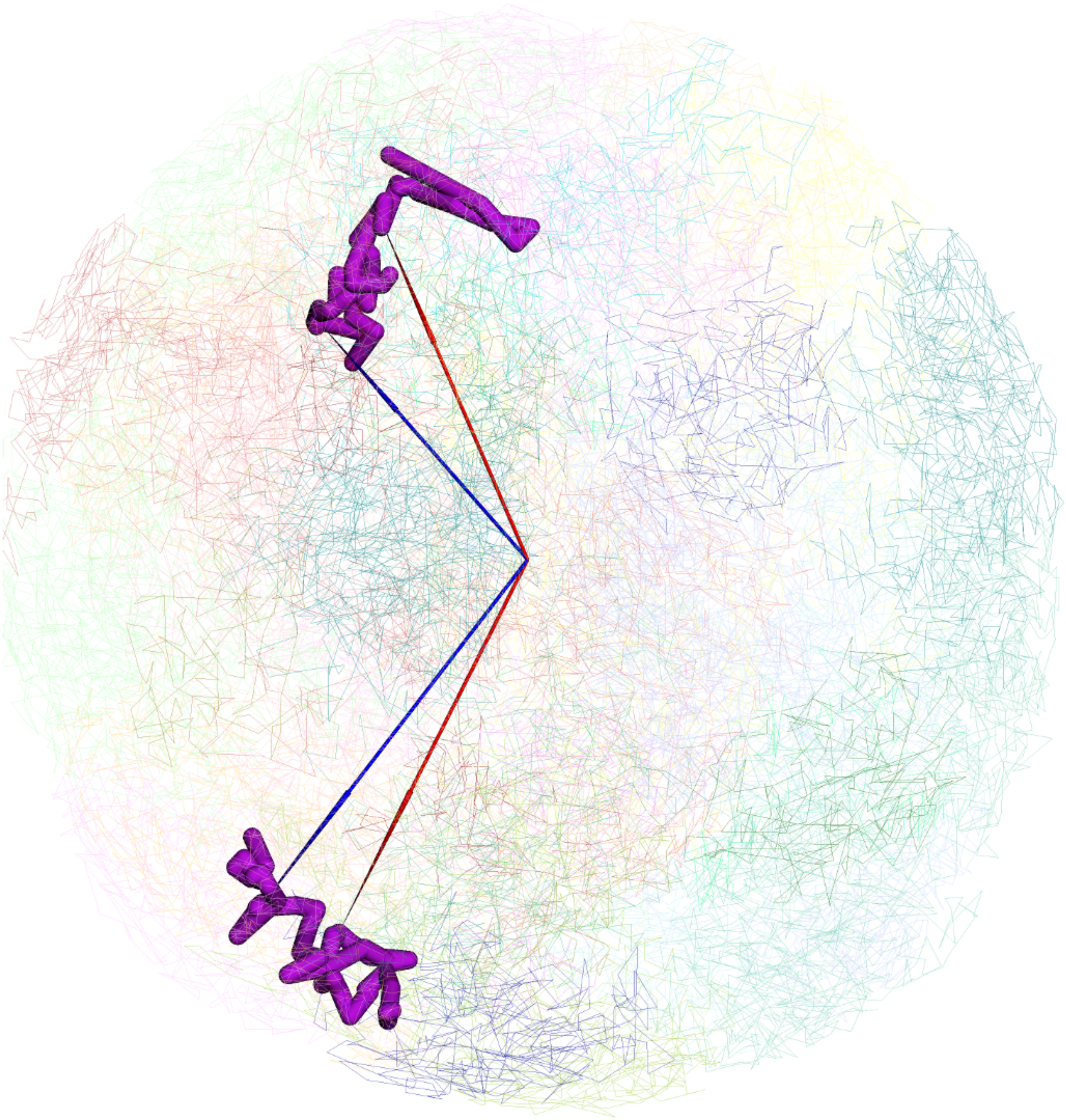
1. chr7:25250000-25500000 radius: 0.2

   From x: 0 y: 0 z: 0 ✓ ⊗

2. chr7:28250000-28500000 radius: 0.2

   From x: 0 y: 0 z: 0 ✓ ⊗

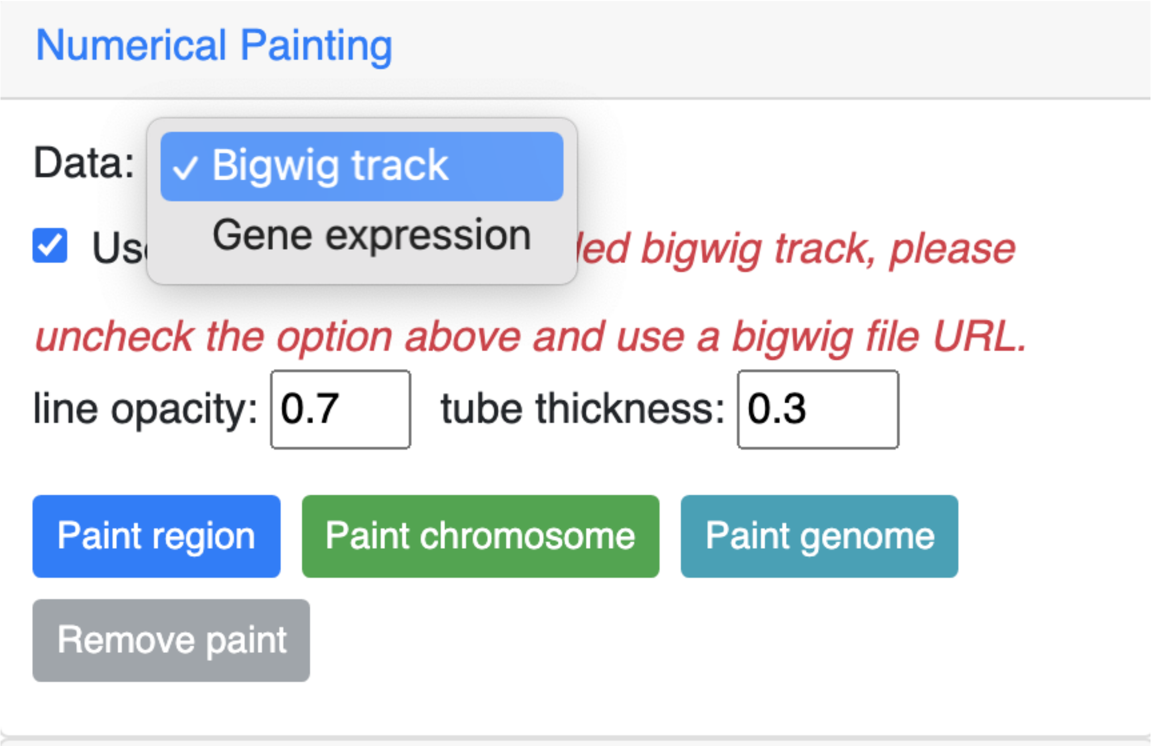
arrows pointing both anchors will be displayed in 3D view (there are 2 models in this structure, patenal and maternal, so 4 arrows displayed here):



8.10 Numerical painting

8.10.1 Numerical painting with bigwig data

The numerical track in `bigWig` format can be used to paint the 3D structure. The *Use loaded tracks* check menu allows user to load either loaded bigWig tracks in browser or submit another bigWig track with file URL.


Numerical Painting

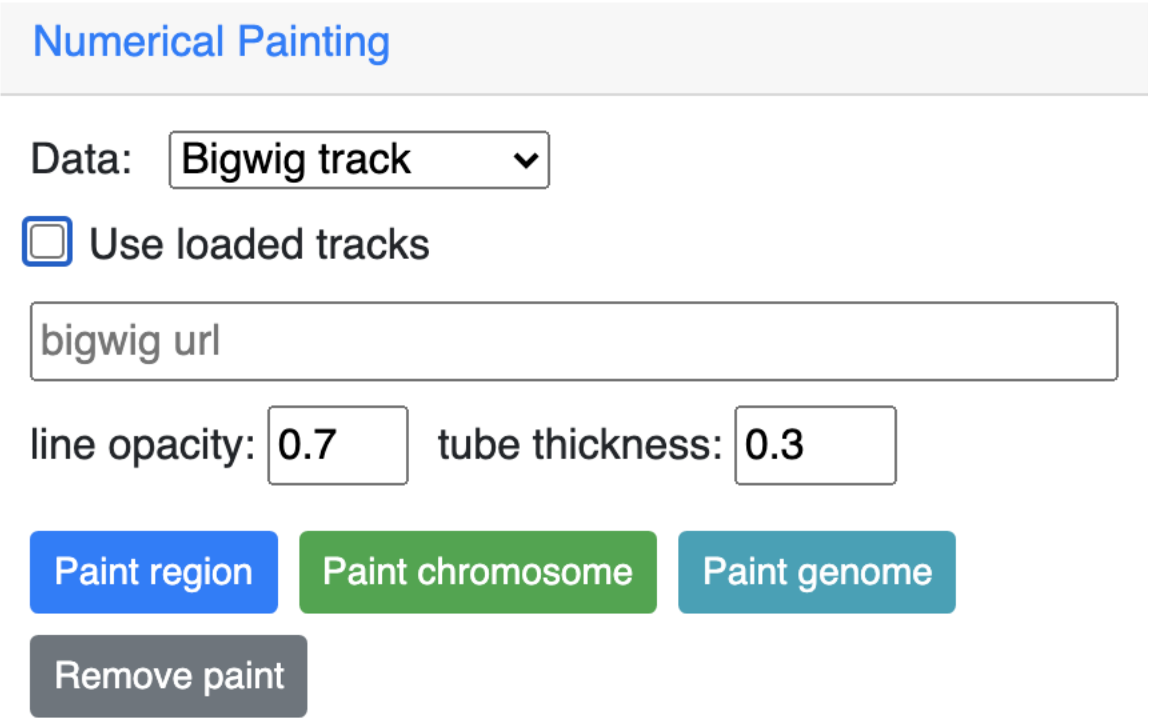
Data: ✓ Bigwig track

☑ Us

...led bigwig track, please uncheck the option above and use a bigwig file URL.

line opacity: 0.7 tube thickness: 0.3

If uncheck *Use loaded tracks*, a URL input will be provided for bigWig URL input:


Numerical Painting

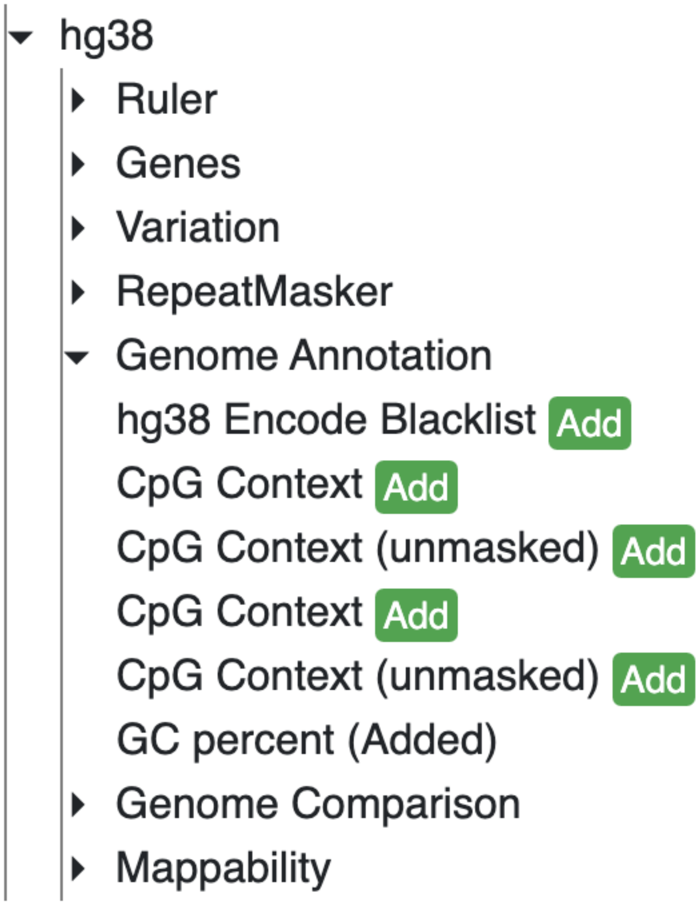
Data: Bigwig track

☐ Use loaded tracks

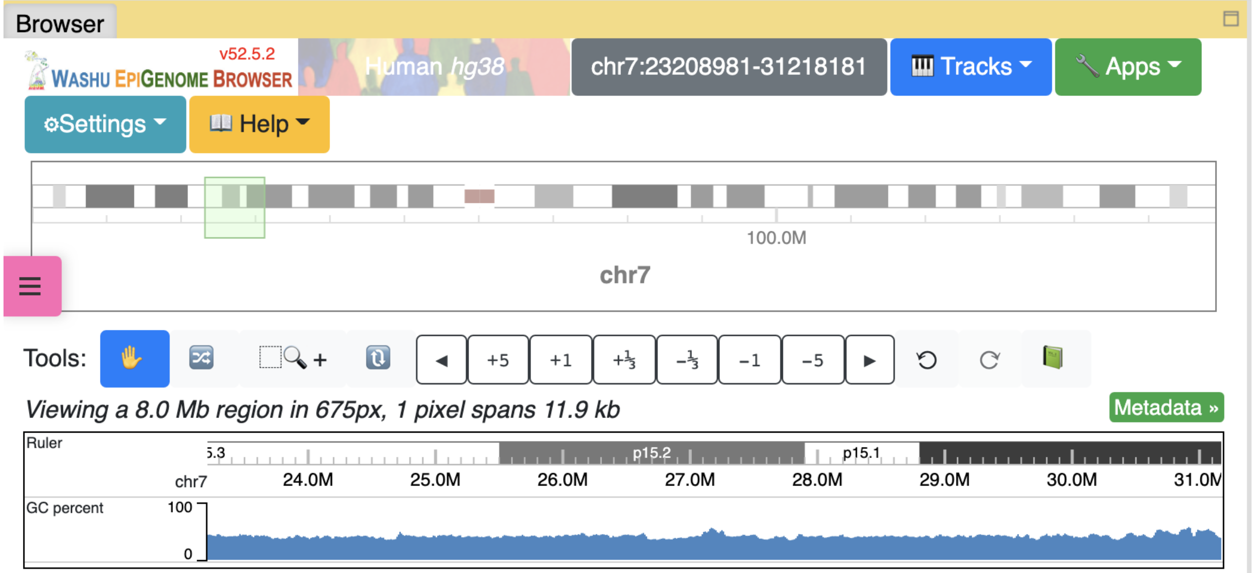
bigwig url

line opacity: 0.7 tube thickness: 0.3

Here we are using the GC percentage data of *hg38* genome as example, add the *GC Percent* track from *Annotation Tracks*:

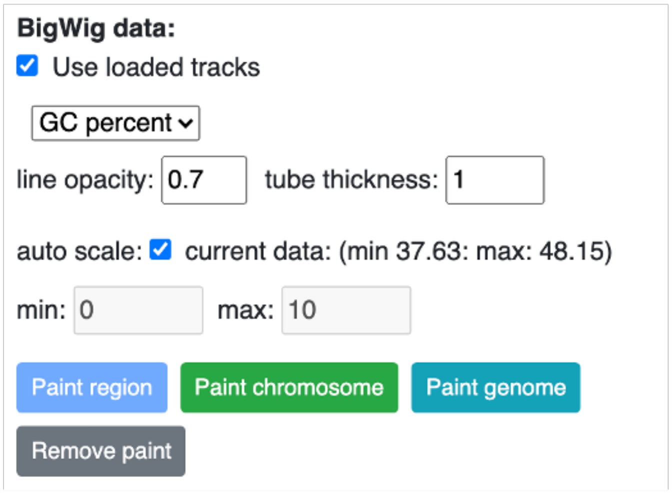


The GC Percent track is added:



Viewing a 8.0 Mb region in 675px, 1 pixel spans 11.9 kb

Choose the track from the dropdown menu:



BigWig data:

☑ Use loaded tracks

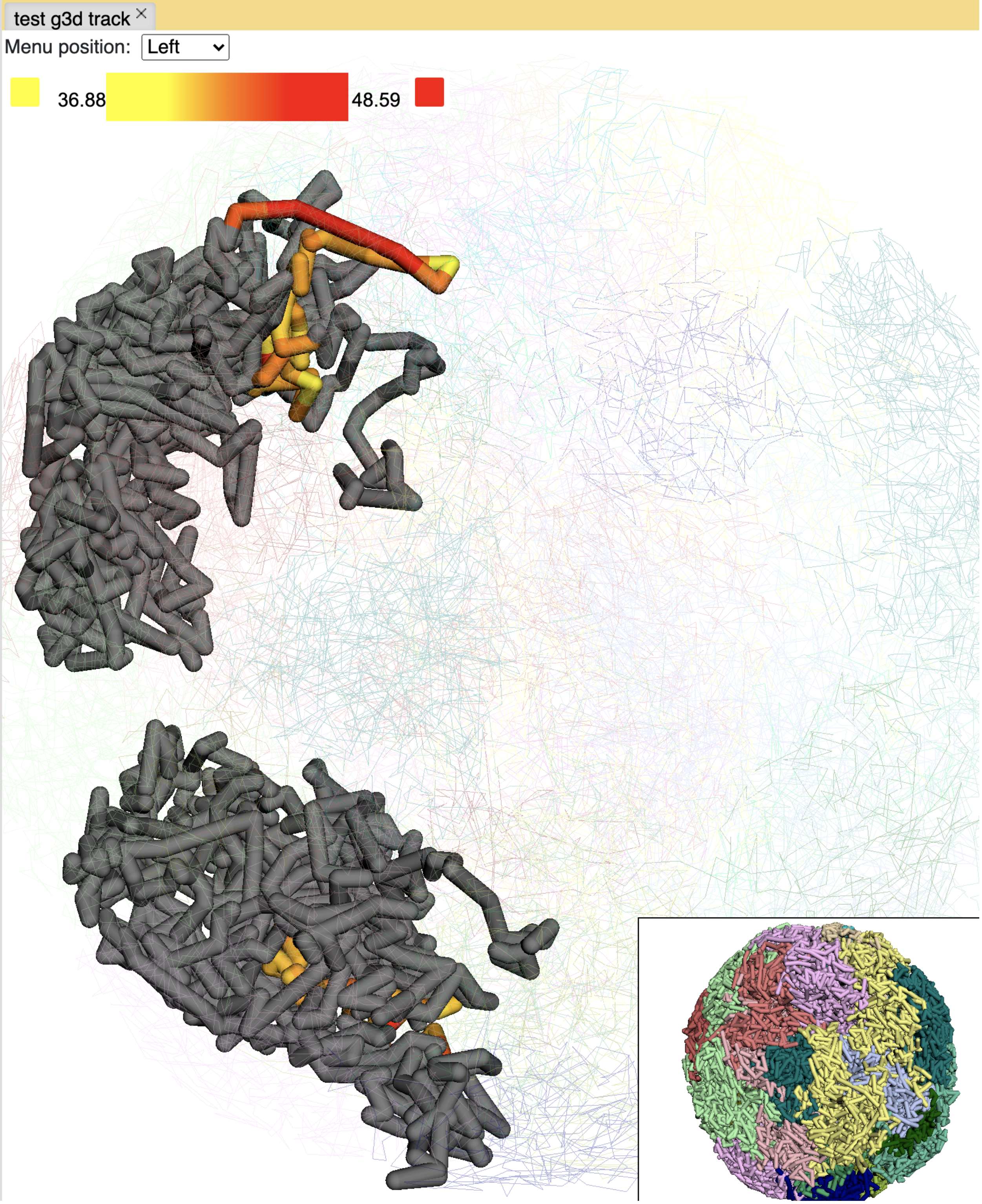
GC percent

line opacity: 0.7 tube thickness: 1

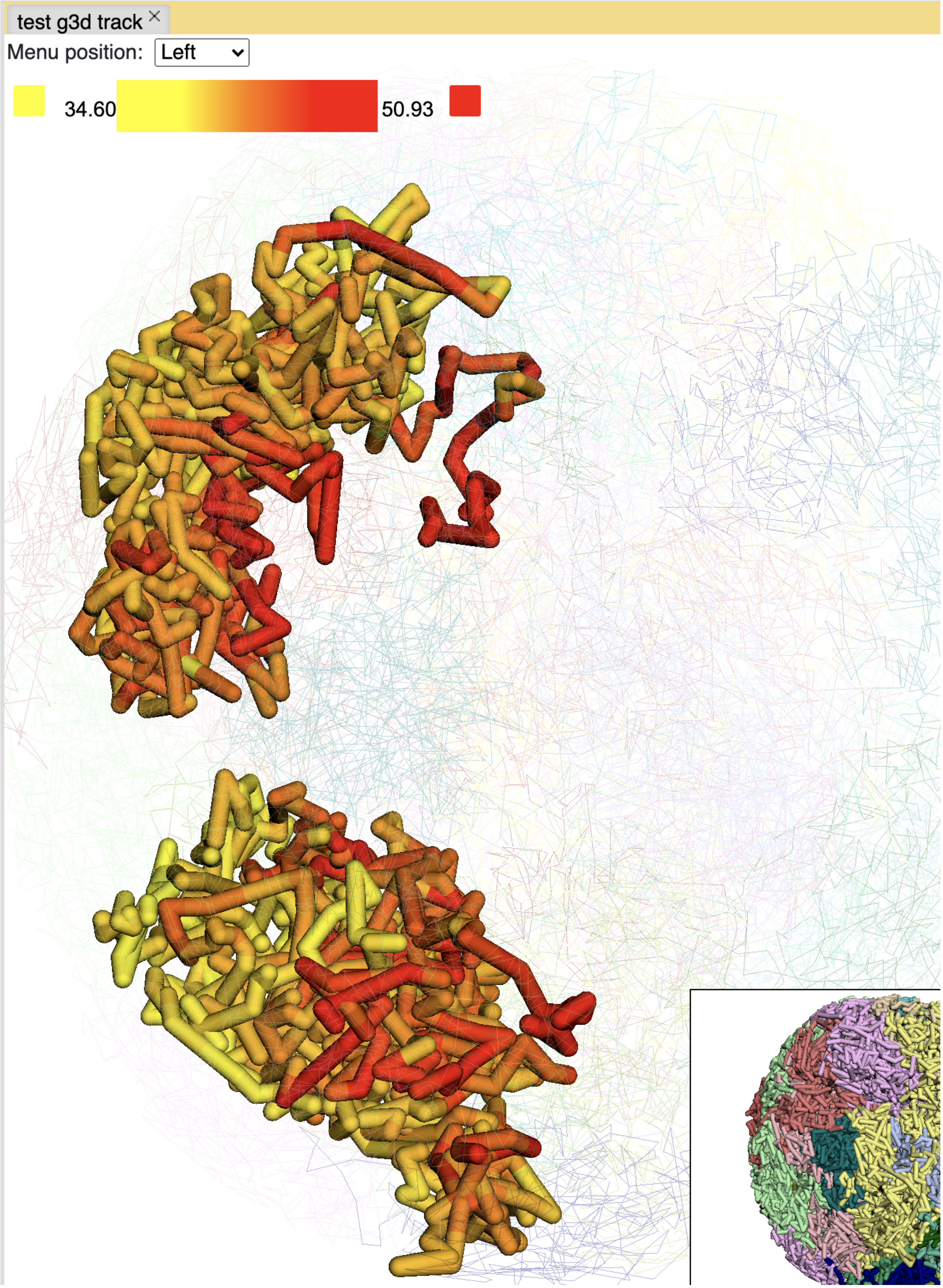
auto scale: ☑ current data: (min 37.63: max: 48.15)

min: 0 max: 10

Click *Paint region* button:

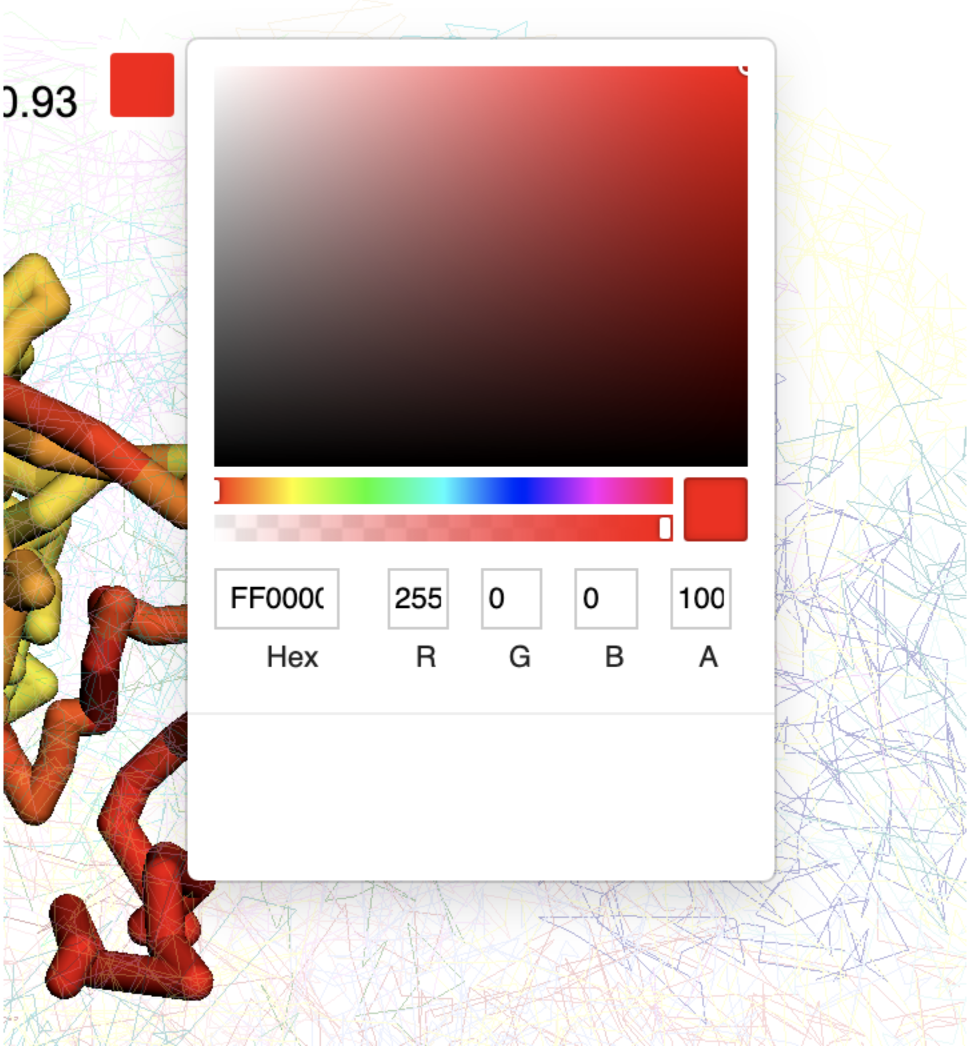


you can also paint the whole chromosome by click the *Paint chromosome* button:



Menu position: Left

Click the color box on the color legend will bring a color palette for choosing colors:



FF000(

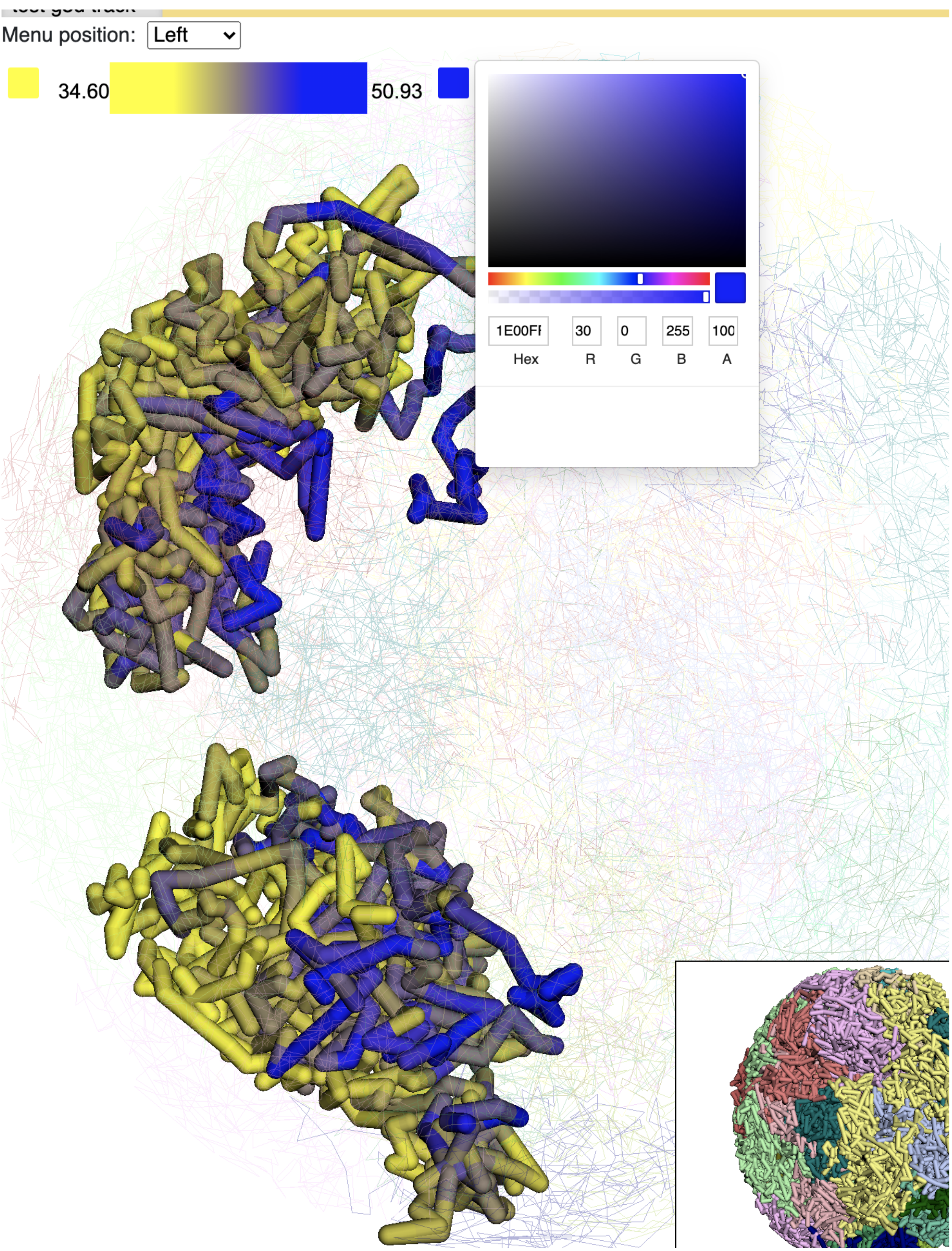
255

0

0

100

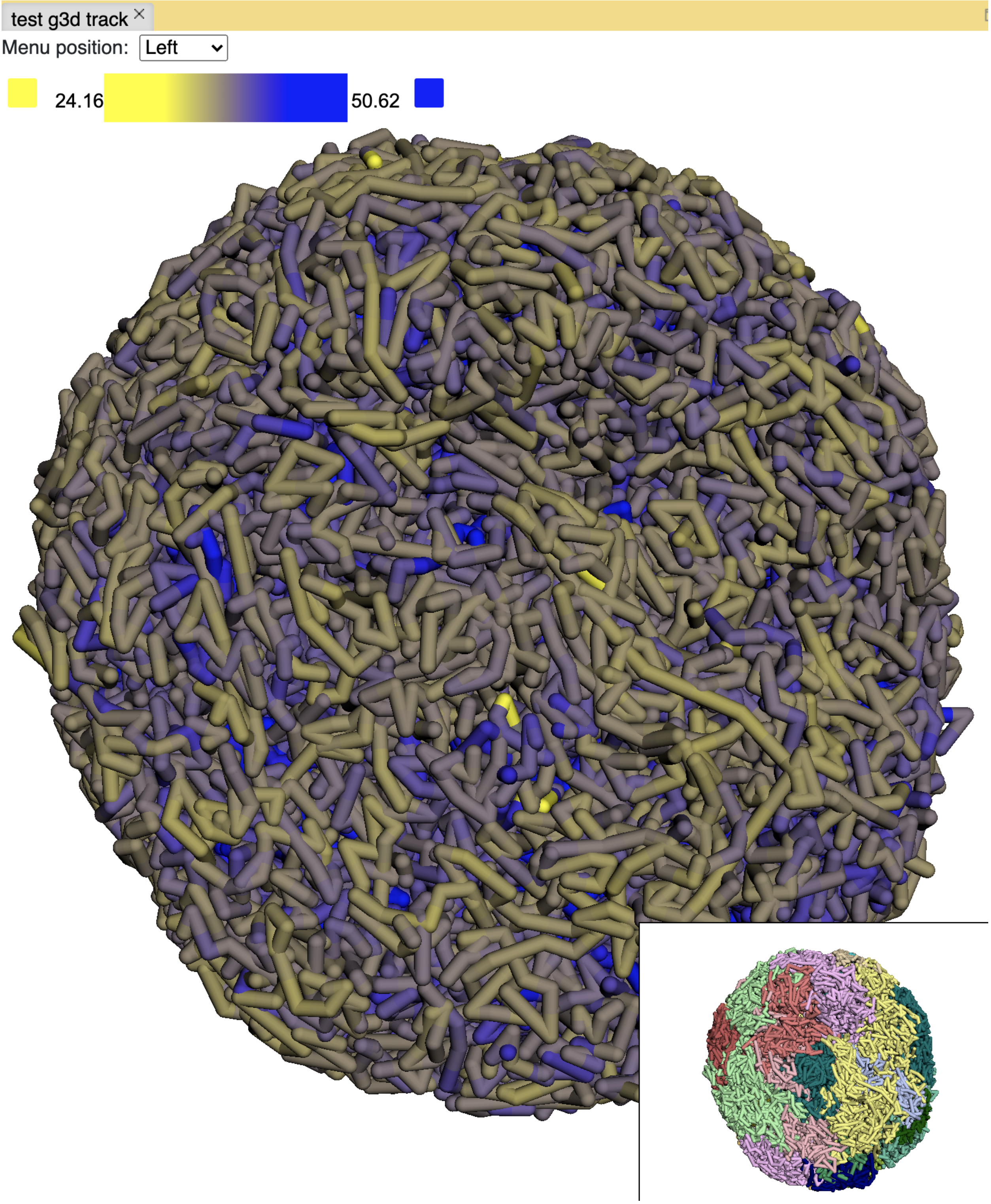
Choose a different color will rerender the structure with color chosen:



Menu position: Left

Hex 1E00FF
R 30
G 0
B 255
A 100

Paint the whole genome is also doable, click the *Paint genome* button:



Menu position: Left

Note: by default the color gradient uses the min and max values from the bigwig file, users can also set the min and max value manually by unchecking the `auto scale` option.

Click the *Remove paint* button will remove the painting.

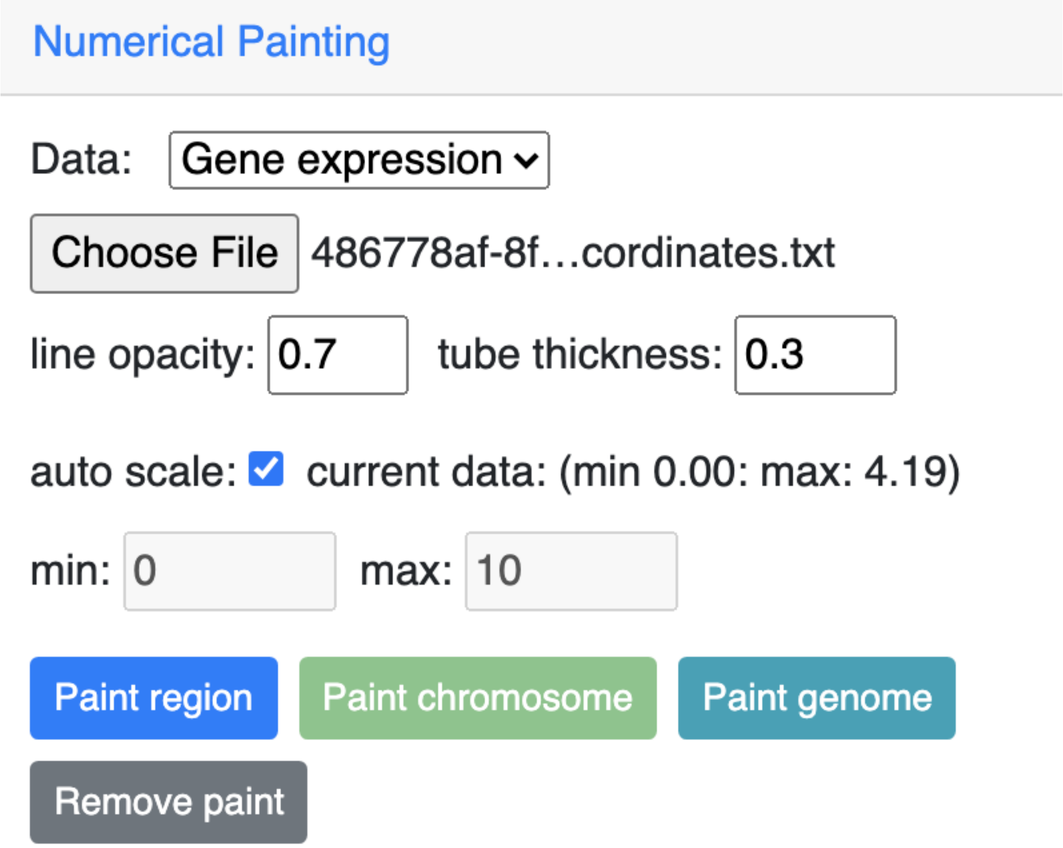
8.10.2 Numerical painting with gene expression data

For painting with gene expression data, the data need be organized in the following format:

```
chr3        168903366       168921996       ENSG00000242268.2       2.40146671319
chr18       46756487        46764408        ENSG00000270112.3       0.0287250976522
chr3        11900011        11901245        ENSG00000225275.4       0.0
chr15       41921417        41928883        ENSG00000259883.1       0.305029986379
chr13       98949719        98950447        ENSG00000231981.3       0.0806509326125
chrX        152682810       152683842       ENSG00000269475.2       0.0
chr12       44880868        44880969        ENSG00000201788.1       0.0
chr17       57092145        57096425        ENSG00000263089.1       0.295277363304
```

This is a 5 column bed format file, each column is chromosome, start, end, gene id or symbol, gene expression value (can be FPKM, RPKM or whatever types of value you want to plot).

Choose *Gene expression* from the dropdown menu, then upload your file, click one the paint button.



Numerical Painting

Data: Gene expression

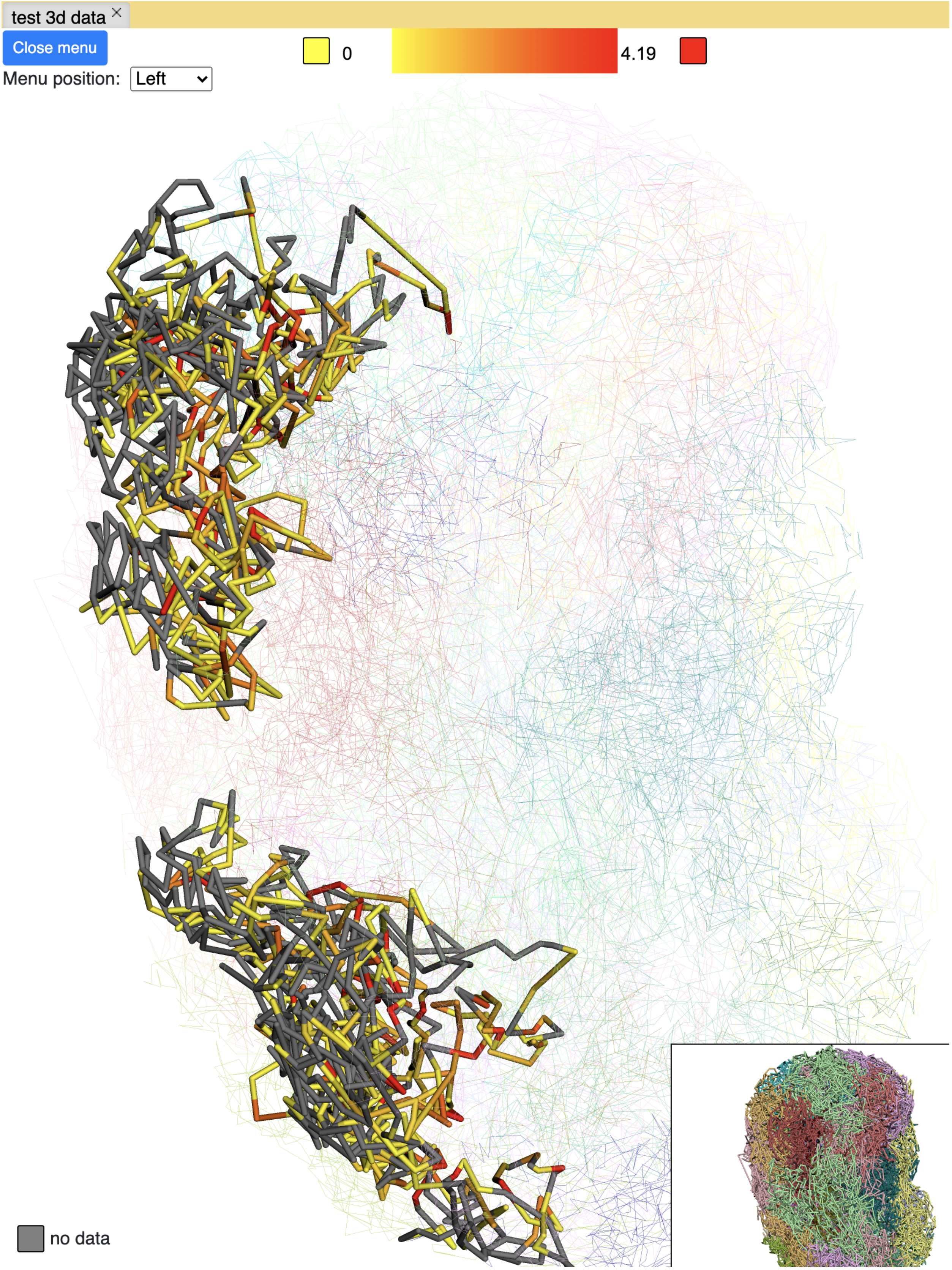
486778af-8f…cordinates.txt

line opacity: 0.7 tube thickness: 0.3

auto scale: ☑ current data: (min 0.00: max: 4.19)

min: 0 max: 10

And this is the view after painting with the expresion data, color and scale can be customized as described before:



test 3d data ×
Close menu
Menu position: Left
0
4.19
no data

8.11 Annotation painting

8.11.1 Supported file formats for 3D annotation painting

cytoband

For *cytoband* there is no need to upload a file, the cytoband data will be read from current loaded genome data.

refGene

The standard *refGene* format from UCSC can be used for painting gene positions on 3D:

```
2085        NR_046630       chr3    +       196666747       196669405       196669405   ␣
↪   196669405       3       196666747,196667841,196669263,  196666995,196668013,
↪196669405,  0       NCBP2-AS1       unk     unk     -1,-1,-1,
2051        NR_046598       chr3    +       192232810       192234362       192234362   ␣
↪   192234362       2       192232810,192234269,    192233297,192234362,    0       ␣
↪FGF12-AS2       unk     unk     -1,-1,
1312        NR_046514       chr13   +       95364969        95368199        95368199    ␣
↪   95368199        2       95364969,95365891,      95365647,95368199,      0       ␣
↪SOX21-AS1       unk     unk     -1,-1,
585 NR_106918       chr1    -       17368   17436   17436   17436   1       17368, ␣
↪17436,  0       MIR6859-1       unk     unk     -1,
585 NR_107062       chr1    -       17368   17436   17436   17436   1       17368, ␣
↪17436,  0       MIR6859-2       unk     unk     -1,
```

bed 9 columns

bed file with 9th column as RGB values can be used as well, for example, the chromHMM from Roadmap project looks like:

```
chr10       0       94800   15_Quies        0       .       0       94800   255,255,255
chr10       94800   95600   9_Het   0       .       94800   95600   138,145,208
chr10       95600   102200  15_Quies        0       .       95600   102200  255,255,255
chr10       102200  104400  9_Het   0       .       102200  104400  138,145,208
chr10       104400  110000  15_Quies        0       .       104400  110000  255,255,255
chr10       110000  111200  9_Het   0       .       110000  111200  138,145,208
```

bed 4 columns

To make things simple, a 4 column bed format is supported as well, with the 4th column has color value:

```
chr11       108280000       109080000       #ff0100
chr11       109080000       109480000       #0000ff
chr11       109720000       110160000       #018100
chr11       110200000       111400000       #0064fb
chr11       111400000       112640000       #ef8c0a
chr11       112640000       113480000       #7f007f
chr11       113520000       114520000       #520000
chr11       114520000       114880000       #39ae00
```

4DN compartment data

Compartment calls table file can also be used to paint the 3D structure. We supported the compartment calls data 4DNFIL65C8ZI from 4DN data portal. The file is pretty small about 1MB in size. The file can either in raw text file (example text) or in compressed gzip format example gzipped text for upload.

The 4DN compartment data looks like:

```
chrom       start   end     gene_count      gene_coverage   E1      E2      E3
chr1        0       100000  595     0.8812700000000001
chr1        100000  200000  952     1.0
chr1        200000  300000  159     0.09797
chr1        300000  400000  132     0.05368
chr1        400000  500000  471     0.24454
chr1        500000  600000  390     0.15467999999999998
chr1        600000  700000  229     0.05782999999999999
```

Rao et.al compartment data

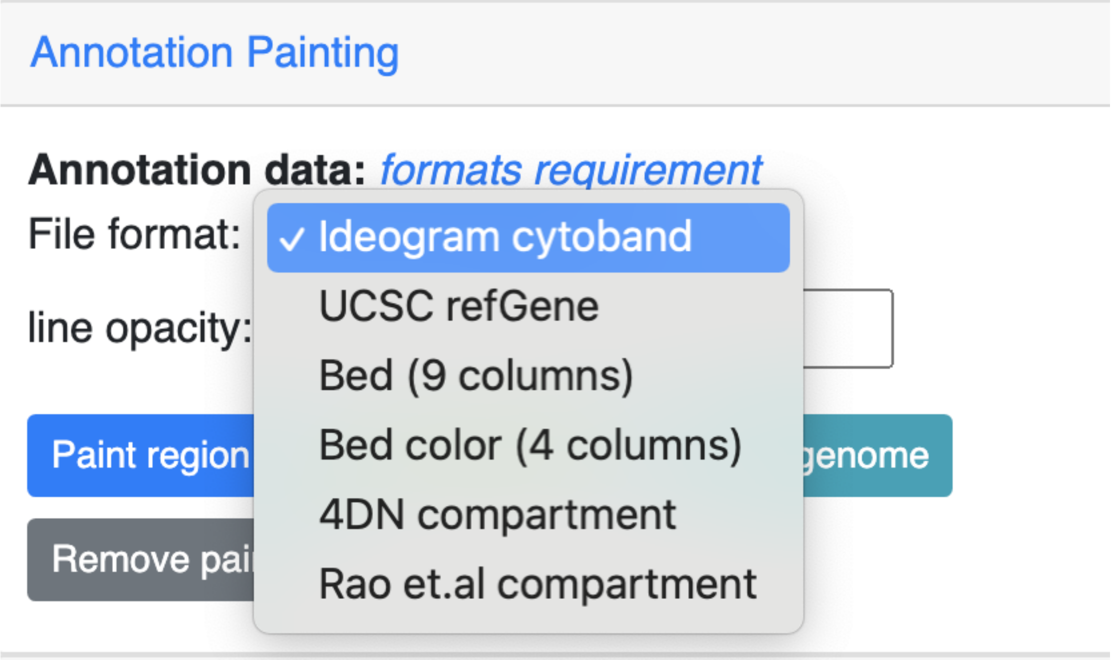
The paper from Rao et.al published in Cell in 2014 also containes a compartment format, the format looks like below:

```
chr19       0       200000  NA      0       .       0       200000  255,255,255
chr19       200000  500000  B1      -1      .       200000  500000  220,20,60
chr19       500000  3800000 A1      2       .       500000  3800000 34,139,34
chr19       3800000 3900000 B1      -1      .       3800000 3900000 220,20,60
chr19       3900000 5000000 A1      2       .       3900000 5000000 34,139,34
chr19       5000000 5600000 B1      -1      .       5000000 5600000 220,20,60
```

Important: The uploaded file for annotation painting can be raw text file or compressed with gzip, but *NOT* with bgzip.

8.11.2 Example annotation painting

Choose the format of your data be used to painting from the dropdown menu:



Annotation Painting

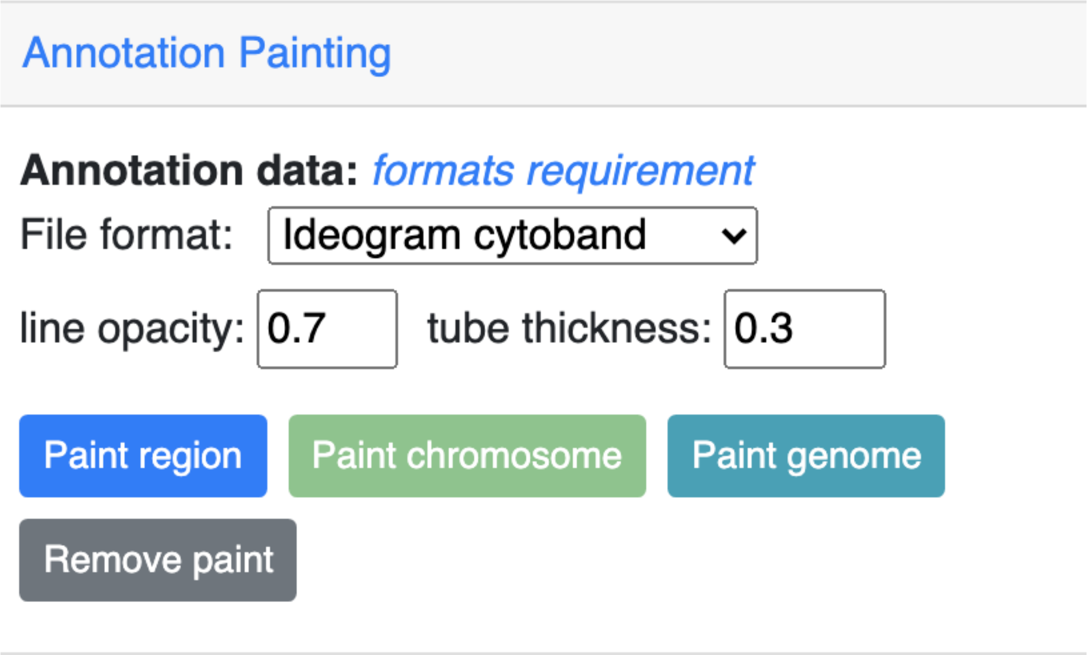
Annotation data: *formats requirement*

File format: ✓ Ideogram cytoband

line opacity:

Then click one the paint button, the upload file button will appear if the format is not cytoband.

cytoband painting

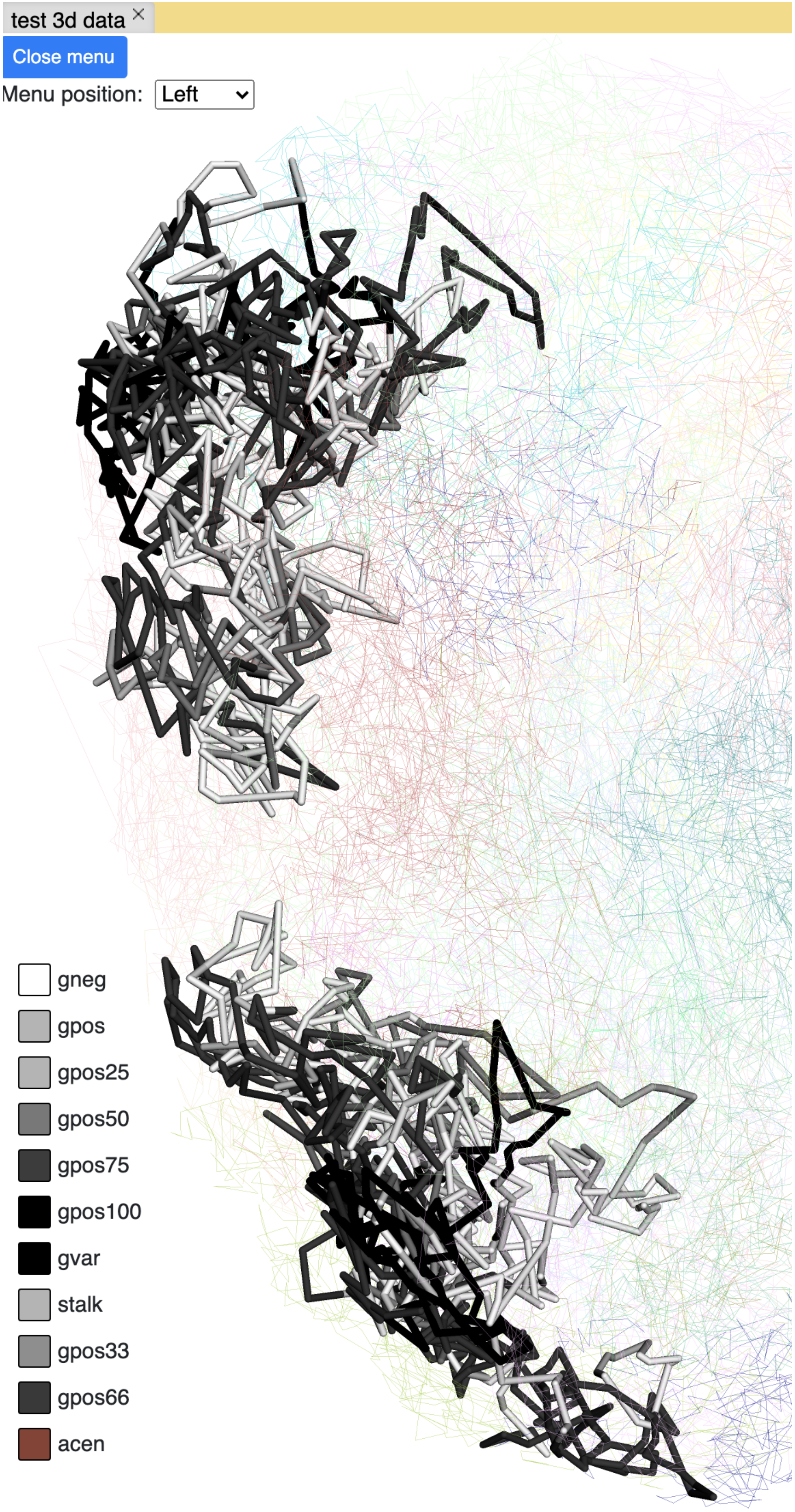


Annotation Painting

Annotation data: *formats requirement*

File format: Ideogram cytoband

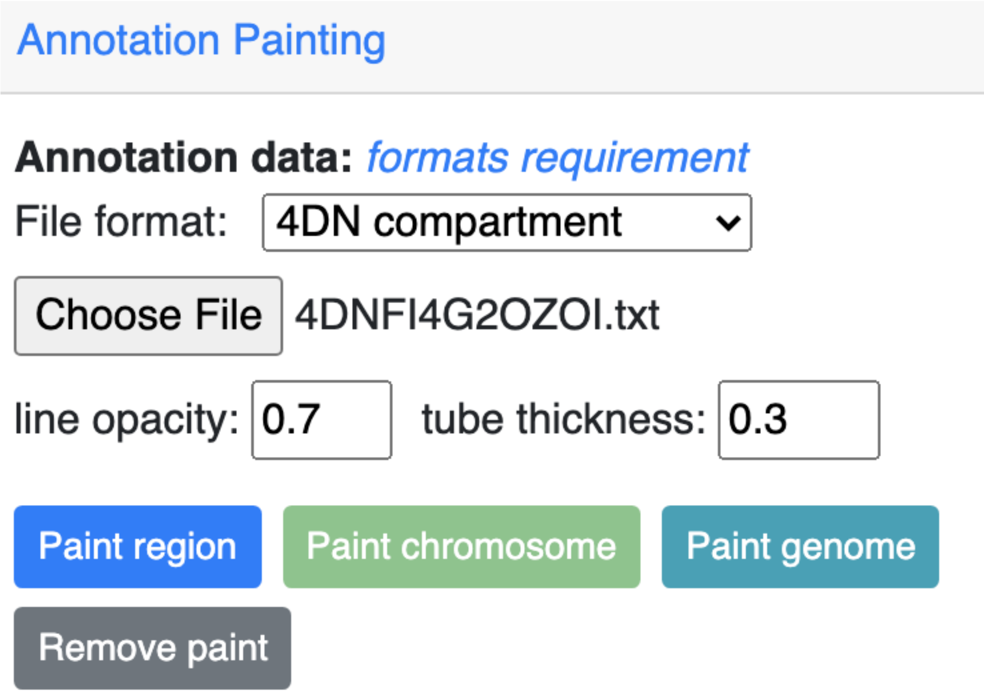
line opacity: 0.7 tube thickness: 0.3



Menu position: Left

4DN compartment painting

The screenshot below is an example using the compartment calls table mentioned above to paint the whole chromosome, green part indicates compartment A and red part indicates compartment B, color can also be customized. The operations are similar to numerical painting, and the painting can also be removed with provided button.



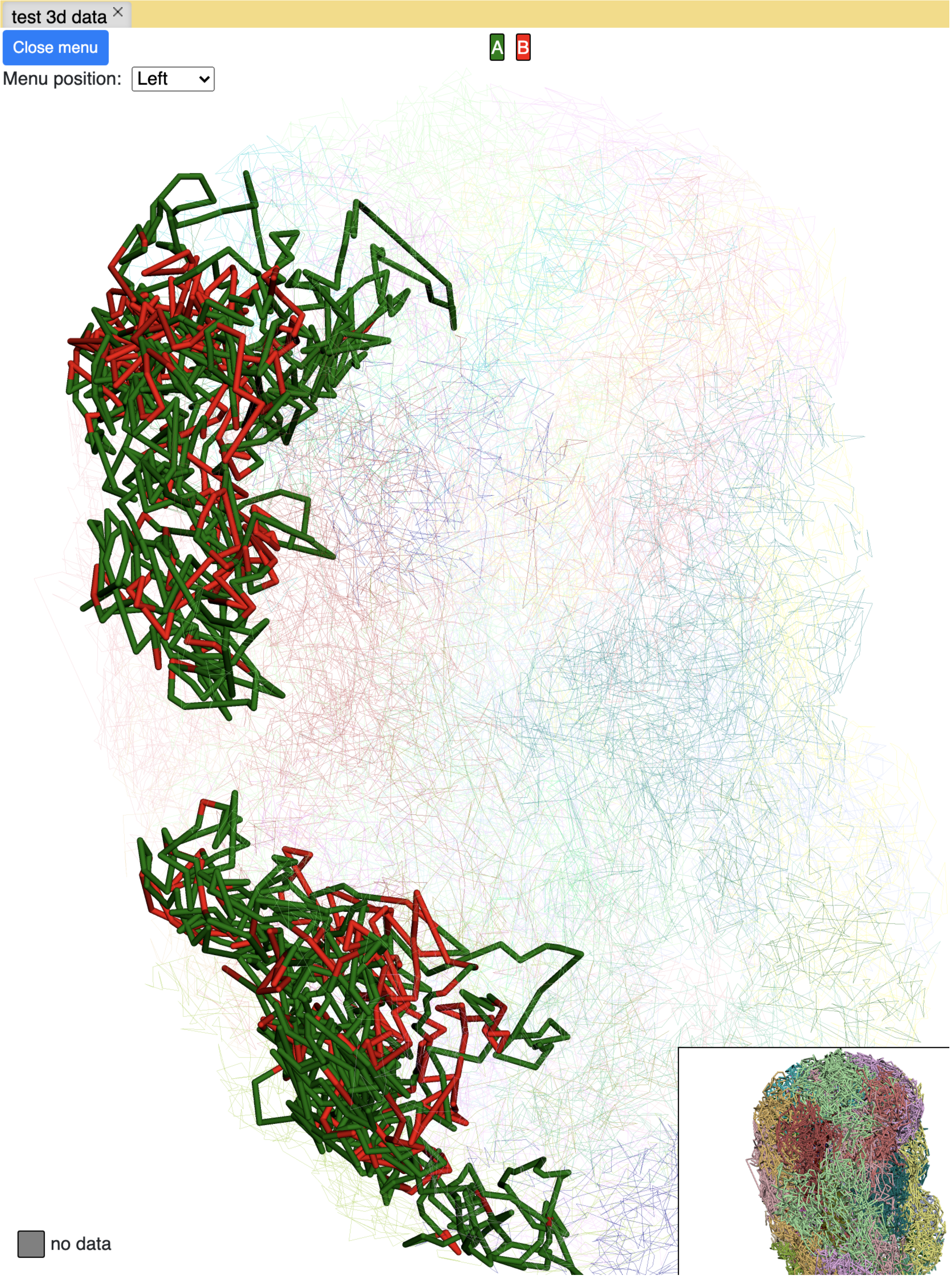
Annotation Painting

Annotation data: *formats requirement*

File format: 4DN compartment

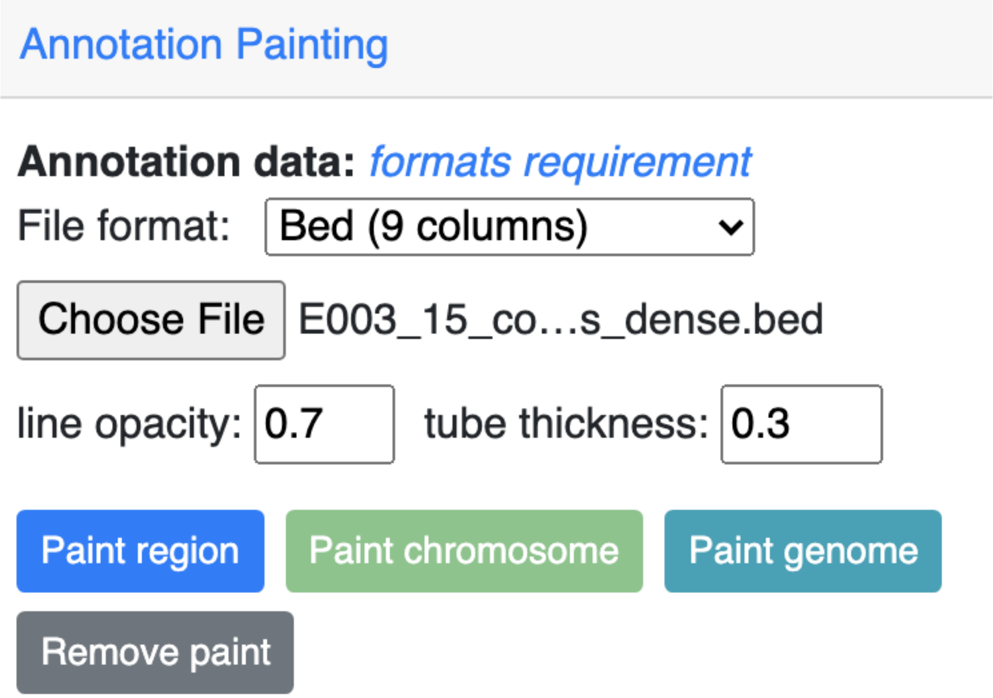
Choose File 4DNFI4G2OZOI.txt

line opacity: 0.7 tube thickness: 0.3


test 3d data ×
Close menu
Menu position: Left
A
B
no data

chromHMM painting

The screenshot below is an example using the chromHMM data from Roadmap to paint the whole chromosome.



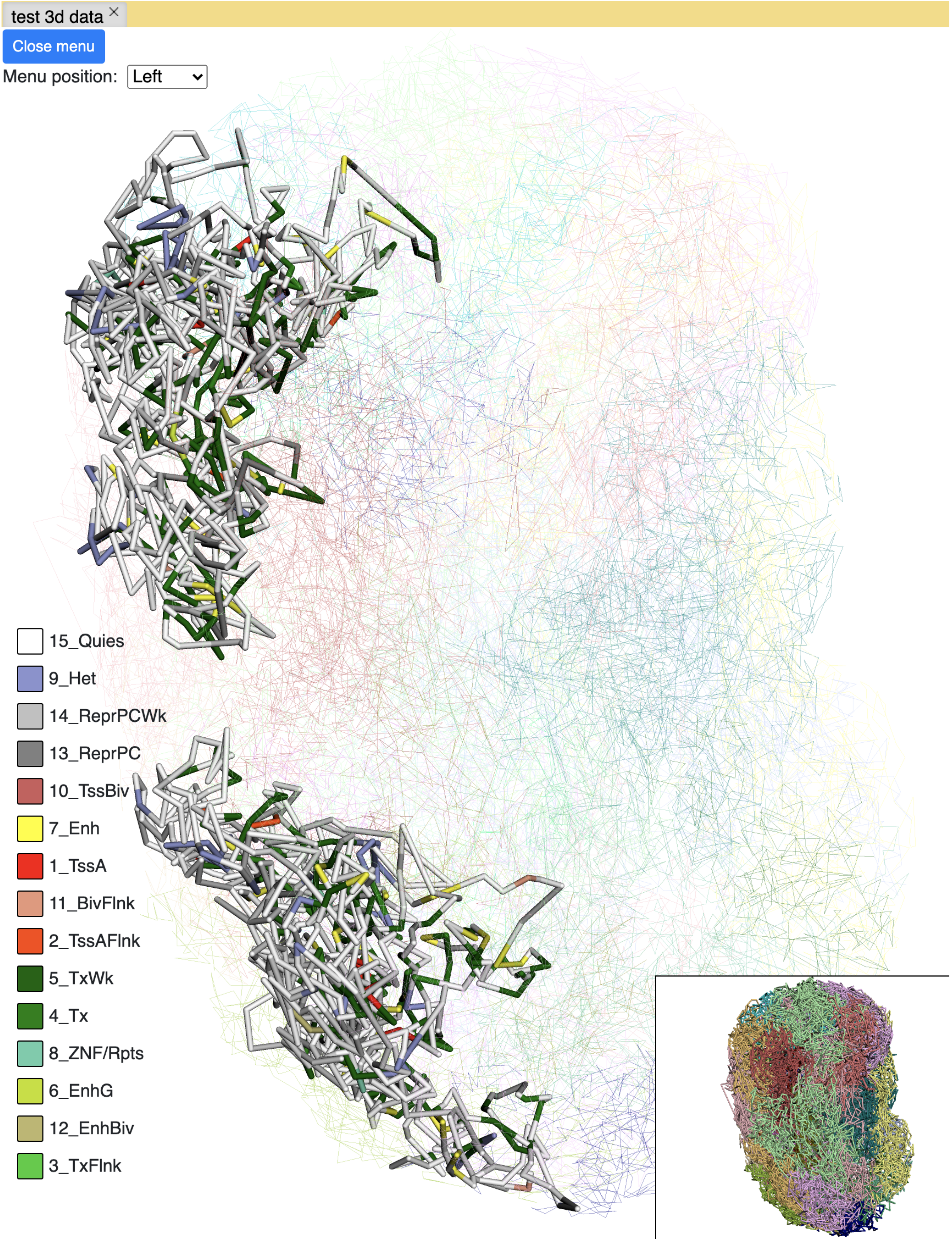
Annotation Painting

Annotation data: *formats requirement*

File format: Bed (9 columns)

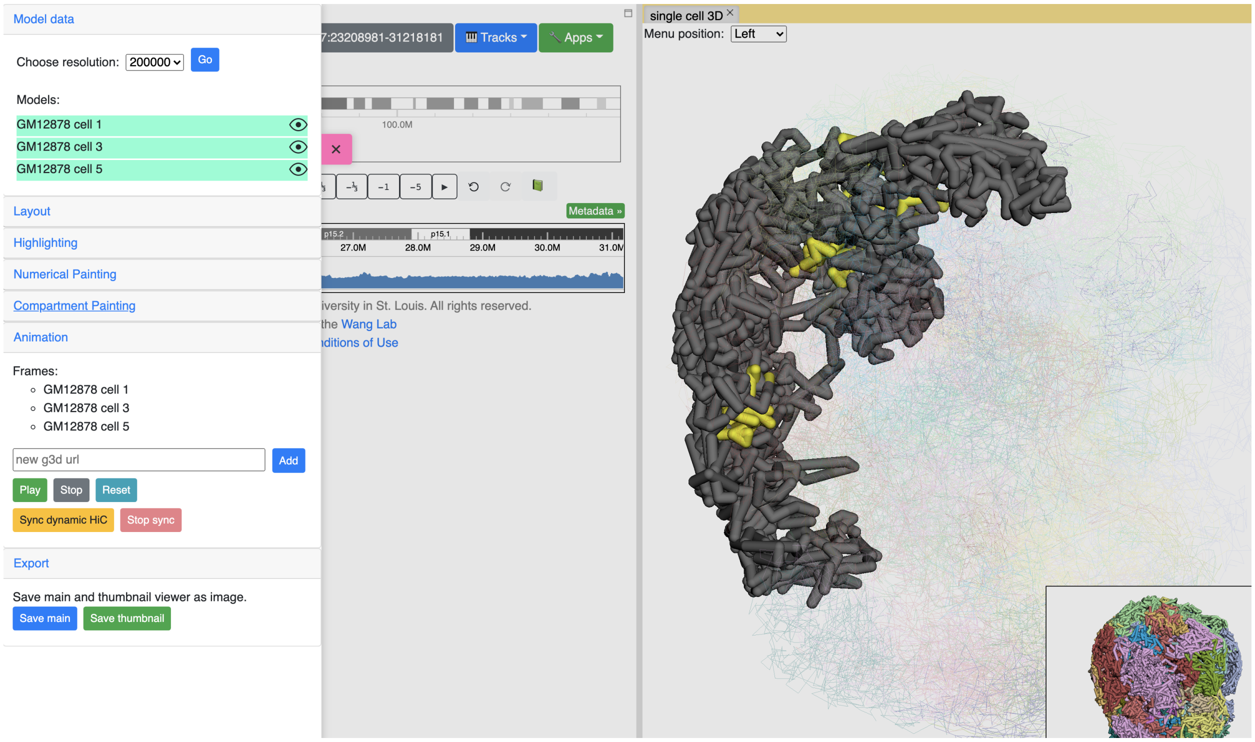
E003_15_co…s_dense.bed

line opacity: 0.7 tube thickness: 0.3


test 3d data ×
Close menu
Menu position: Left
15_Quies
9_Het
14_ReprPCWk
13_ReprPC
10_TssBiv
7_Enh
1_TssA
11_BivFlnk
2_TssAFlnk
5_TxWk
4_Tx
8_ZNF/Rpts
6_EnhG
12_EnhBiv
3_TxFlnk

8.12 Animations on 3D

g3d format is designed to be a container file format, it might contain multiple models from haplotypes or different cells/samples, each model may also contain data at different resolution. This example file contains 3D structure data from 3 different cell at different resolutions. When there are multiple models available, the 3D viewer can play animation while each model will be displayed as a frame and loop over every model. Add this example as g3d track, this is how it looks like:



Choose resolution: 200000

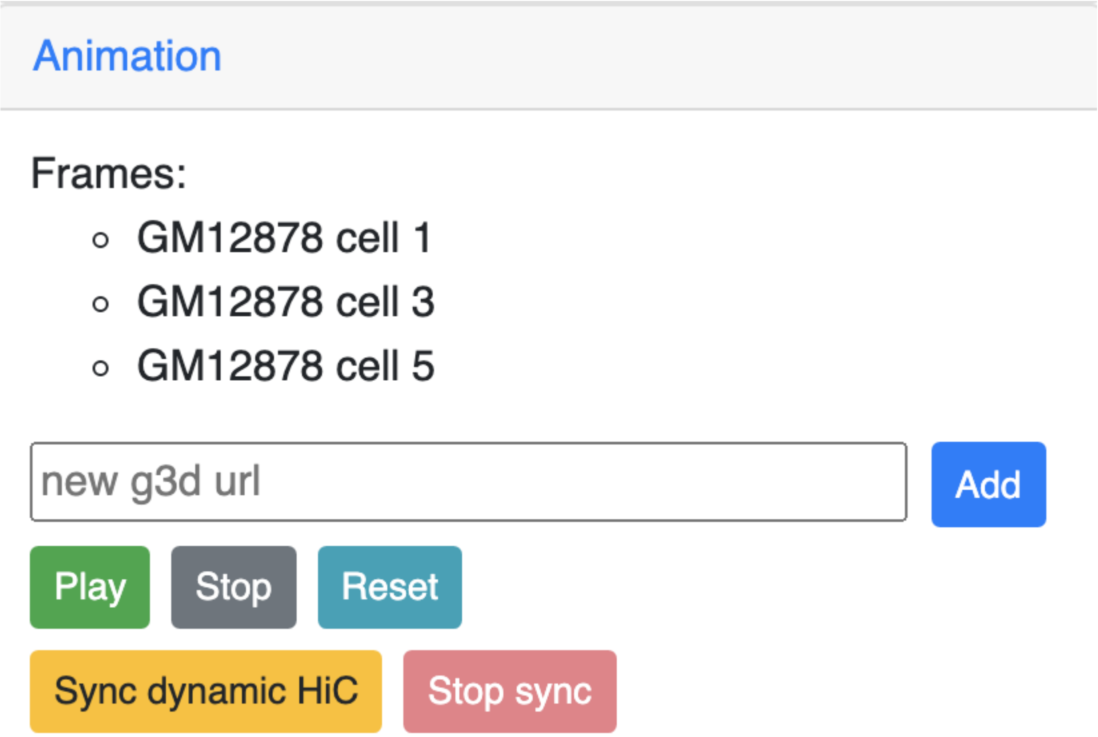
Models:
- GM12878 cell 1
- GM12878 cell 3
- GM12878 cell 5

Frames:
- GM12878 cell 1
- GM12878 cell 3
- GM12878 cell 5

Menu position: Left

Save main and thumbnail viewer as image.

in the *Animation* section, click the *Play* button the animation will start, *Stop* will stop the animation, and *Reset* will reset the viewer to default view style.



Animation

Frames:

- GM12878 cell 1
- GM12878 cell 3
- GM12878 cell 5

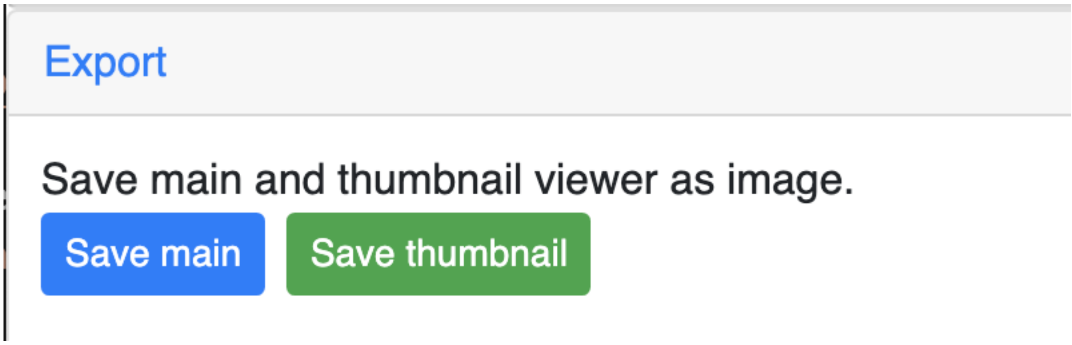
Please check the animation below (speed was adjusted to reduce animation file size for documentation):

8.12.1 Sync 3D structure with dynamic hic

Since the browser have both dynamic hic track type and animation over 3D structures, there is a way to sync the animation between dynamic hic track and 3D structure. The *Sync dynamic HiC* button enables animation synchronization between dynamic hic and models in 3D structure. Please see the animation below for example:

8.13 Export 3D images

The 3D viewer can export current view as image in png format for download. Simplely click the buttons under *Export* section, users can download the image in main and thumbnail viewer.



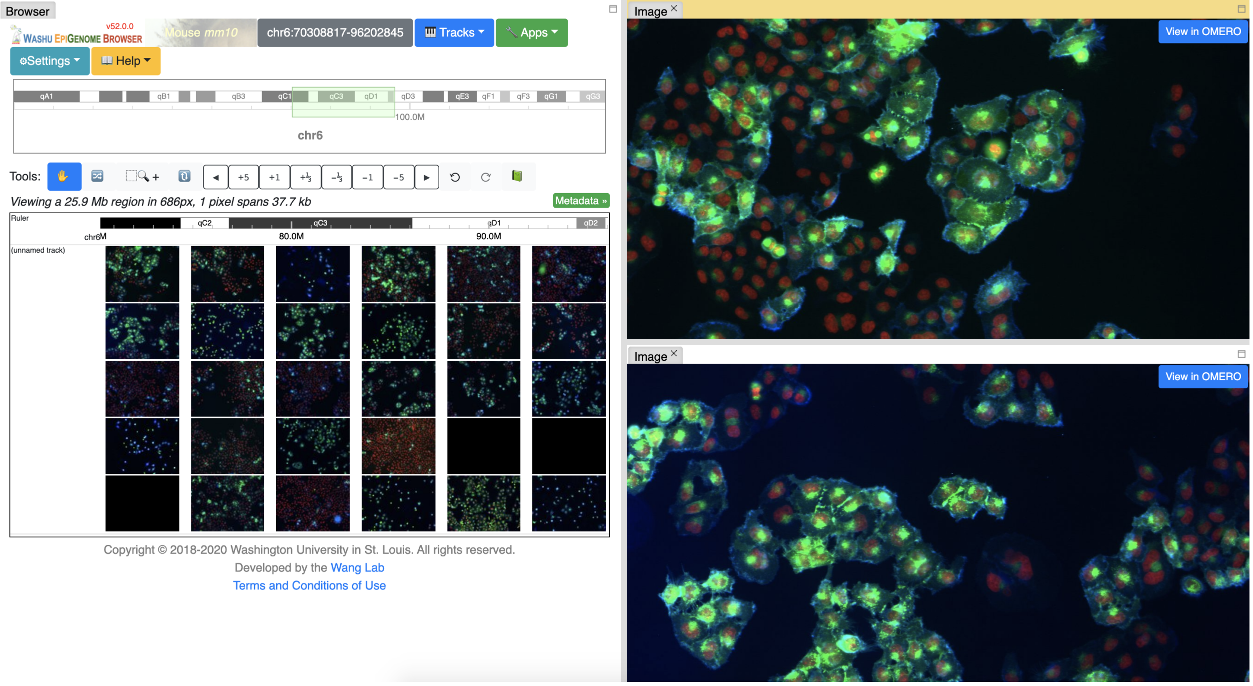
Export

Save main and thumbnail viewer as image.

CHAPTER NINE

IMAGE TRACKS

The browser currently supports loading image data from Omero server hosted by 4DN data portal and Image Data Resource (IDR). An example interface after loading one image track is illustrated below:

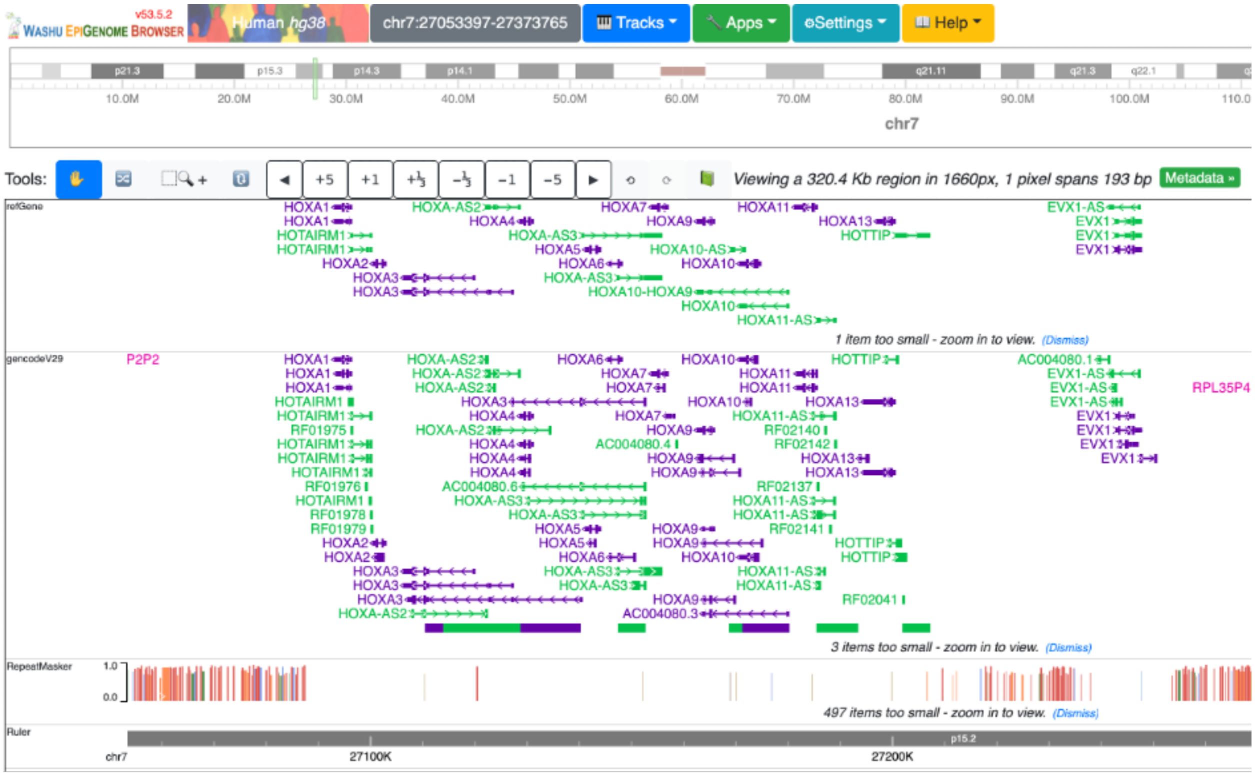


Viewing a 25.9 Mb region in 686px, 1 pixel spans 37.7 kb

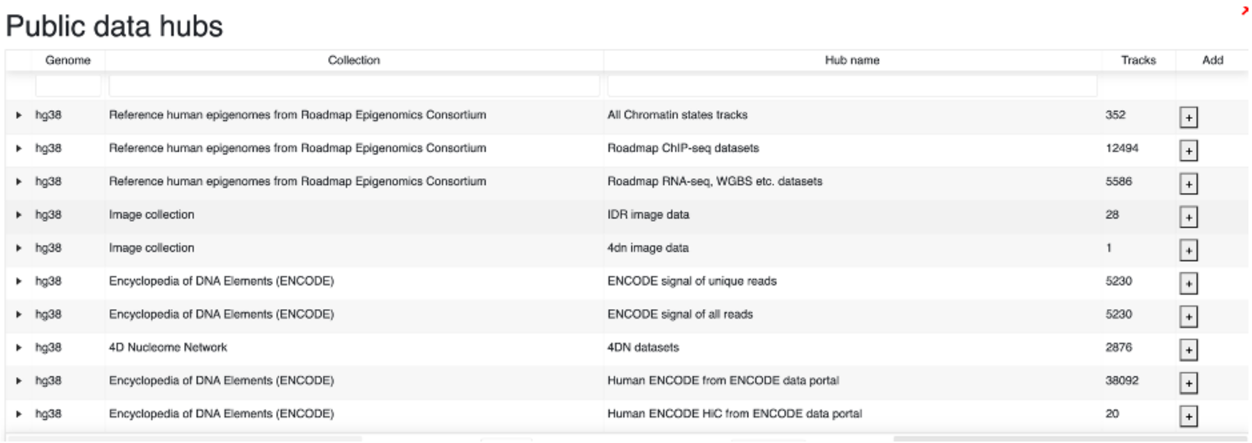
Copyright © 2018-2020 Washington University in St. Louis. All rights reserved.
Developed by the Wang Lab
Terms and Conditions of Use

9.1 view human image data from IDR

Go the browser, choose hg38:


v53.5.2
WashU EpiGenome Browser
Human hg38
chr7:27053397-27373765
Tracks
Apps
Settings
Help
chr7
Tools:
+5
+1
+1/3
−1/3
−1
−5
Viewing a 320.4 Kb region in 1660px, 1 pixel spans 193 bp
Metadata
refGene
gencodeV29
RepeatMasker
Ruler
1 item too small - zoom in to view. (Dismiss)
3 items too small - zoom in to view. (Dismiss)
497 items too small - zoom in to view. (Dismiss)
27100K
27200K

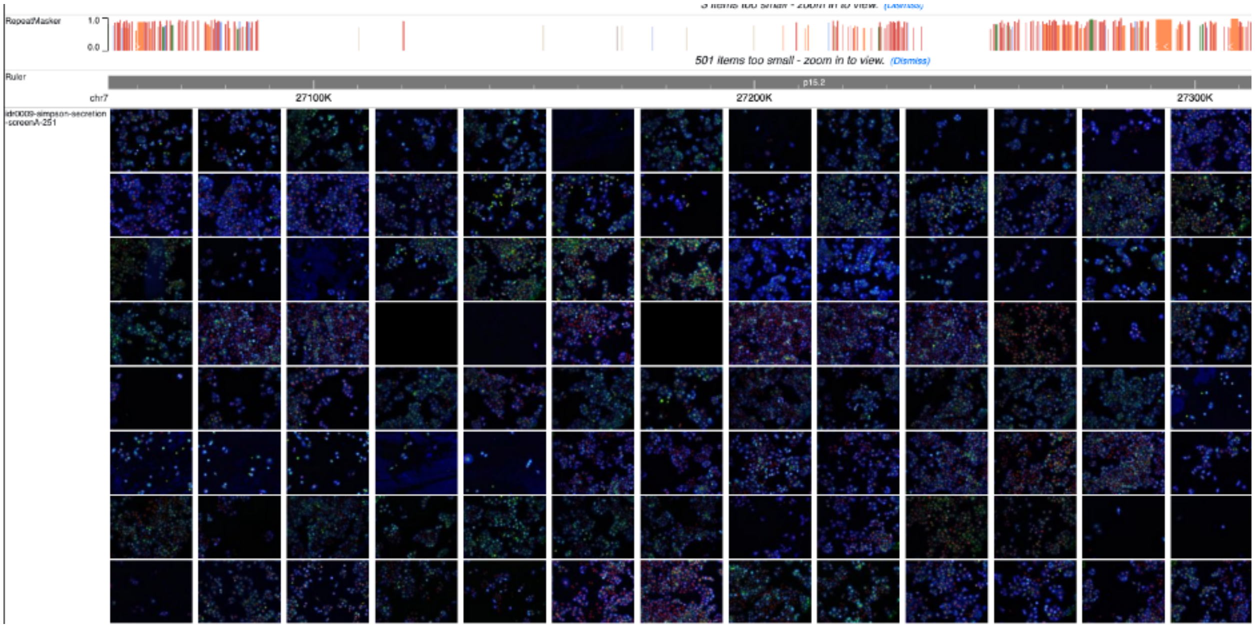
Go to Tracks, public data hubs, load the IDR image data hub:



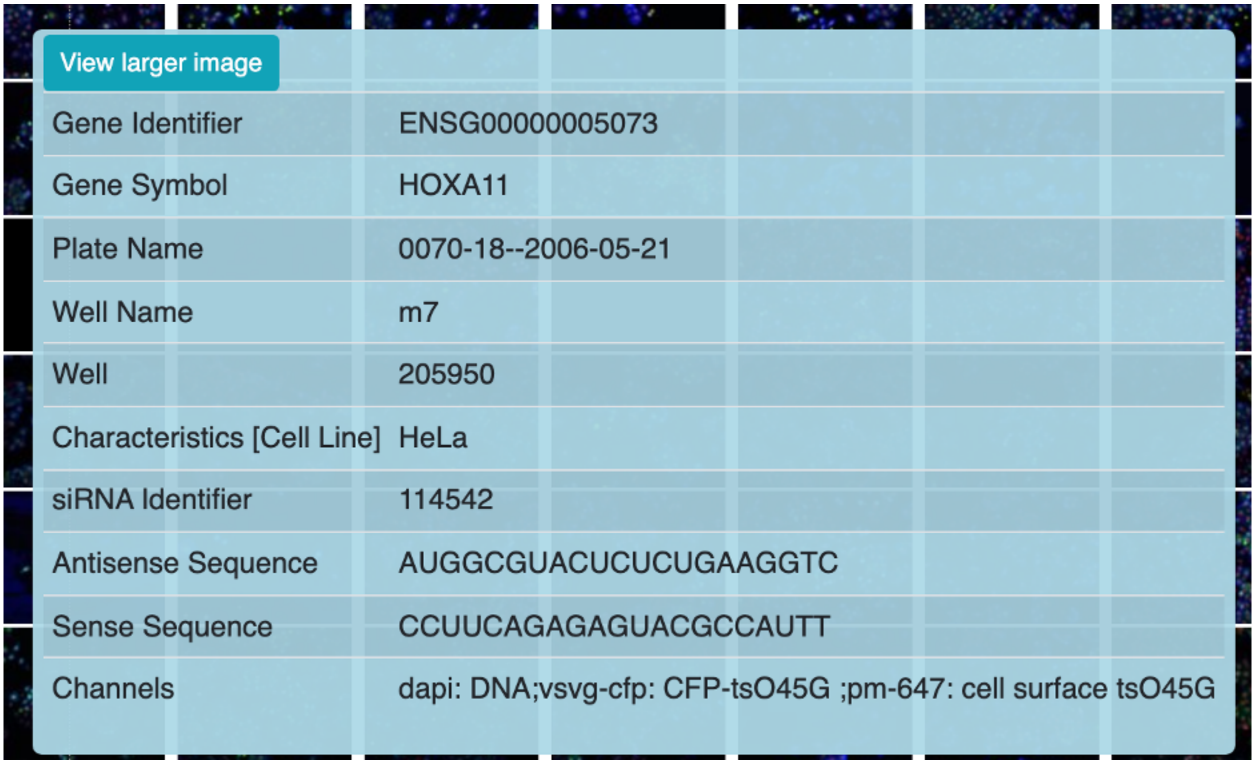
Public data hubs

	Genome	Collection	Hub name	Tracks	Add
▸	hg38	Reference human epigenomes from Roadmap Epigenomics Consortium	All Chromatin states tracks	352	+
▸	hg38	Reference human epigenomes from Roadmap Epigenomics Consortium	Roadmap ChIP-seq datasets	12494	+
▸	hg38	Reference human epigenomes from Roadmap Epigenomics Consortium	Roadmap RNA-seq, WGBS etc. datasets	5586	+
▸	hg38	Image collection	IDR image data	28	+
▸	hg38	Image collection	4dn image data	1	+
▸	hg38	Encyclopedia of DNA Elements (ENCODE)	ENCODE signal of unique reads	5230	+
▸	hg38	Encyclopedia of DNA Elements (ENCODE)	ENCODE signal of all reads	5230	+
▸	hg38	4D Nucleome Network	4DN datasets	2876	+
▸	hg38	Encyclopedia of DNA Elements (ENCODE)	Human ENCODE from ENCODE data portal	38092	+
▸	hg38	Encyclopedia of DNA Elements (ENCODE)	Human ENCODE HiC from ENCODE data portal	20	+

You can see the browser window is load with the image track:

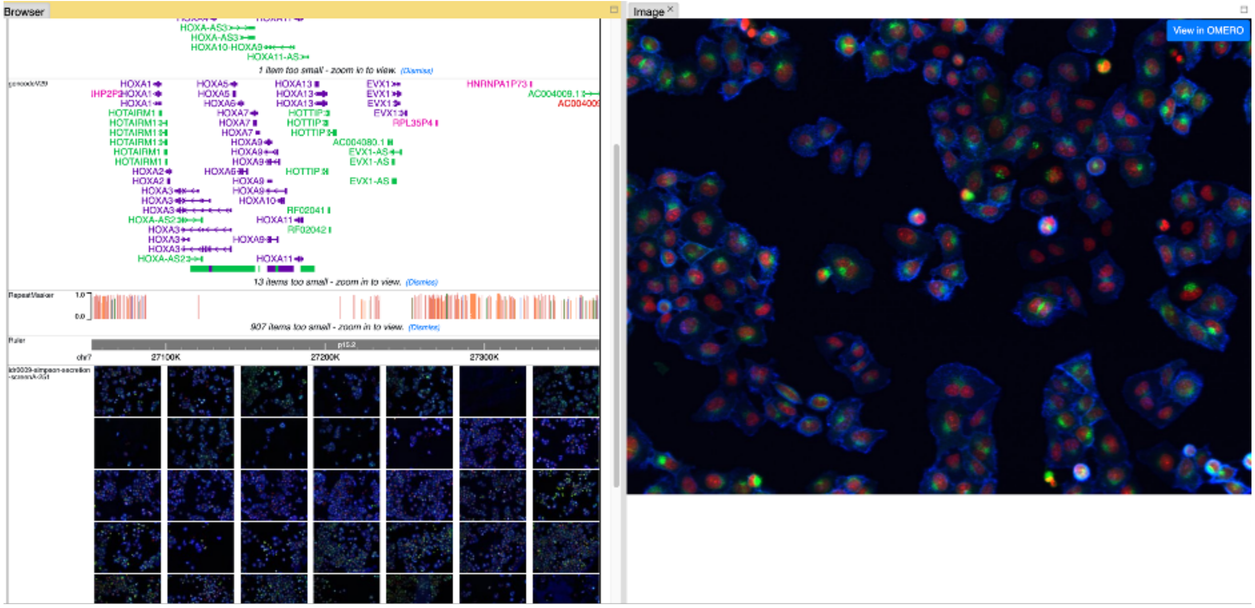

RepeatMasker
1.0
0.0
501 items too small - zoom in to view. (Dismiss)
Ruler
p15.2
chr7
27100K
27200K
27300K
idr0009-simpson-secretion-screenA-251

Click any of the image to show metadata popup:

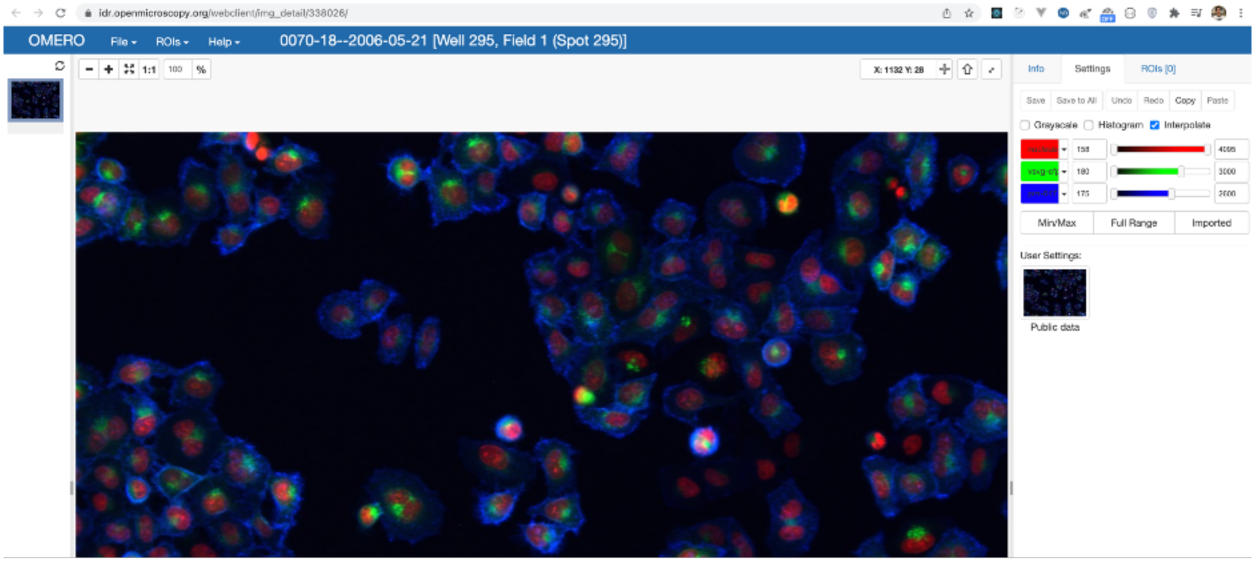


Gene Identifier	ENSG00000005073
Gene Symbol	HOXA11
Plate Name	0070-18--2006-05-21
Well Name	m7
Well	205950
Characteristics [Cell Line]	HeLa
siRNA Identifier	114542
Antisense Sequence	AUGGCGUACUCUCUGAAGGTC
Sense Sequence	CCUUCAGAGAGUACGCCAUTT
Channels	dapi: DNA;vsvg-cfp: CFP-tsO45G ;pm-647: cell surface tsO45G

Click View larger image button to see the image in a new panel:

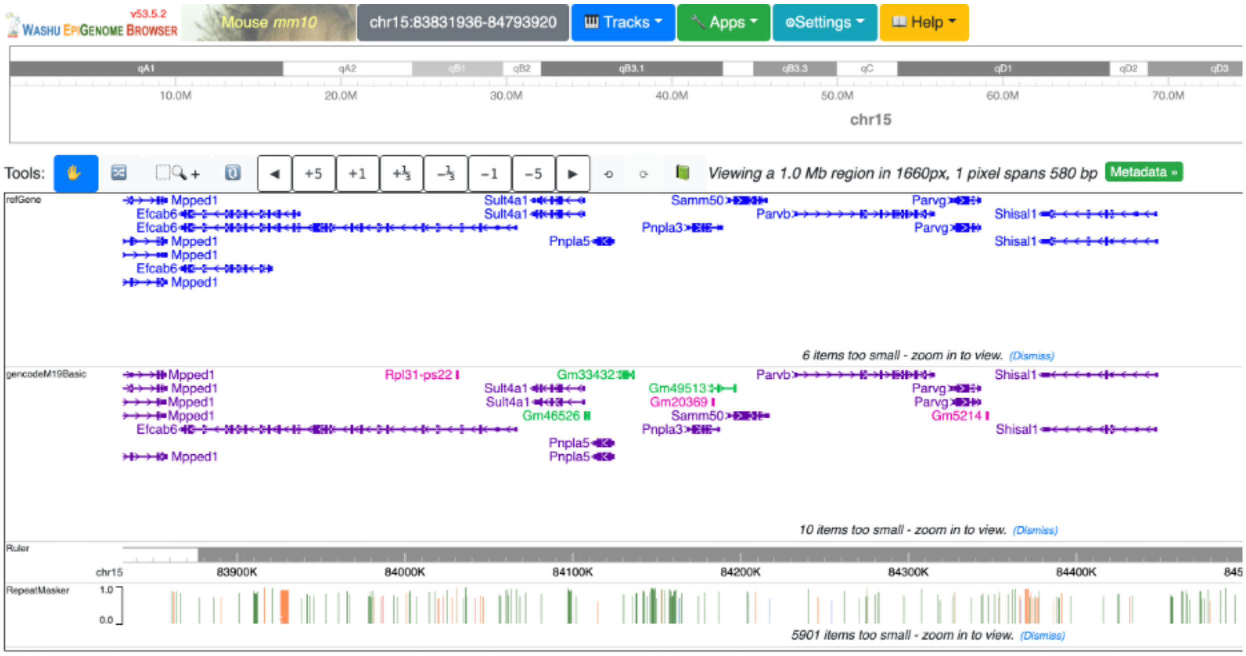

Browser
Image
View in OMERO
1 item too small - zoom in to view.
13 items too small - zoom in to view.
907 items too small - zoom in to view.
RepeatMasker
Ruler
chr7
27100K
27200K
27300K

Click `View in Omero` button in the new panel to see the image details in IDR website:

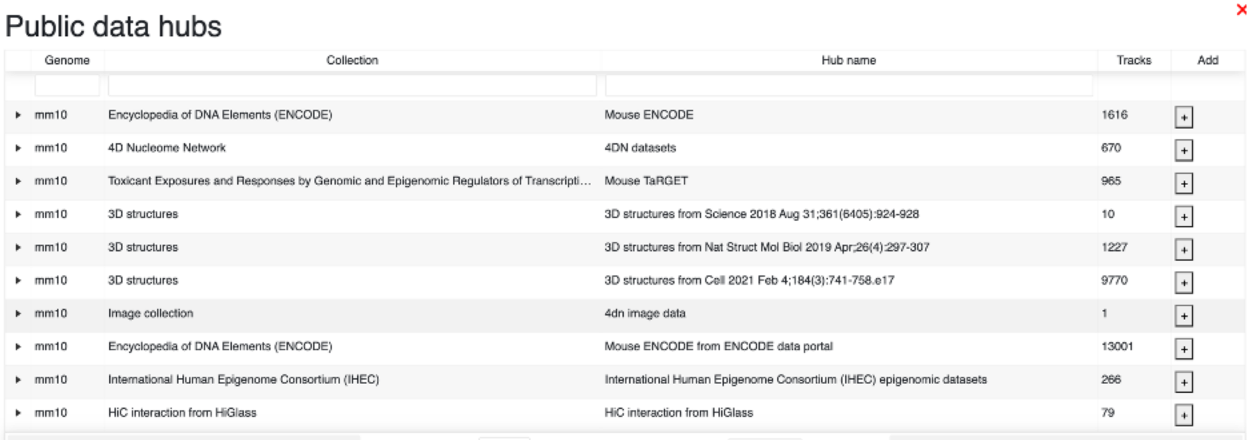


9.2 view mouse image data from 4DN

Go to the browser, load mm10 genome, since the image data is sparse in mm10, we would navigate to `chr15:83831936-84793920` first:



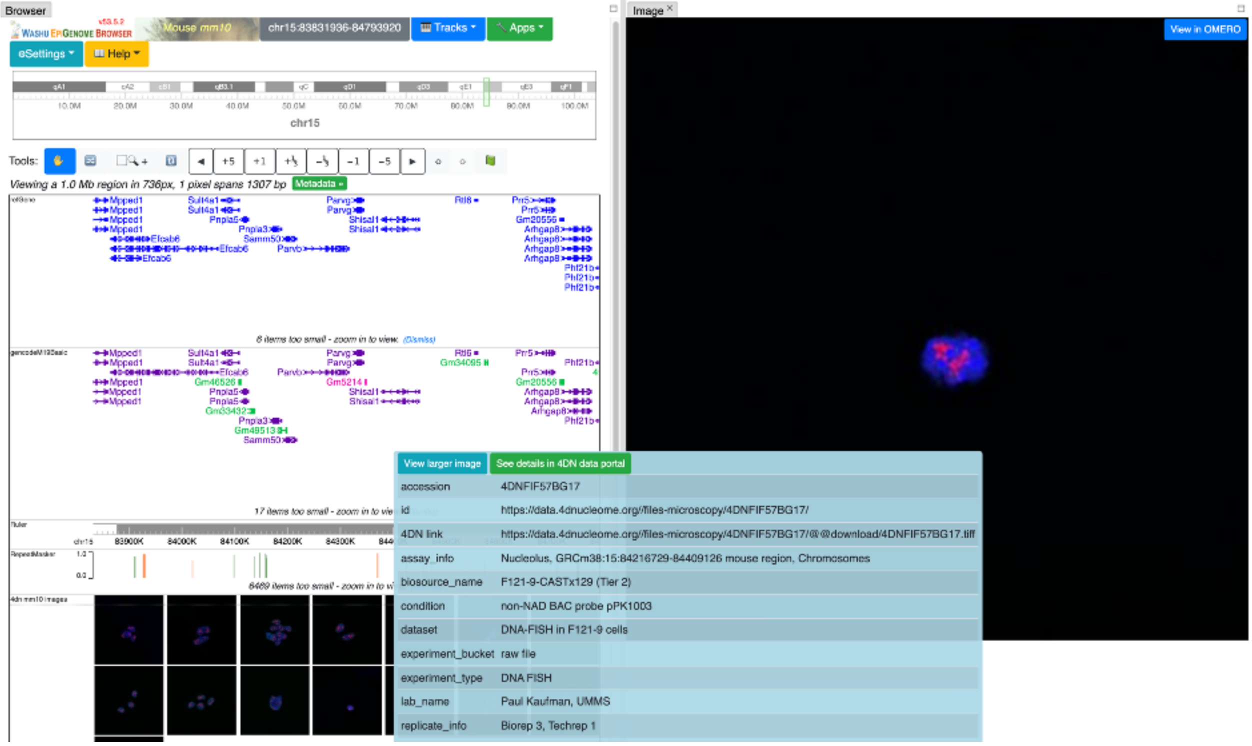
Go to Tracks, public data hubs, load the 4DN image data hub:



Public data hubs

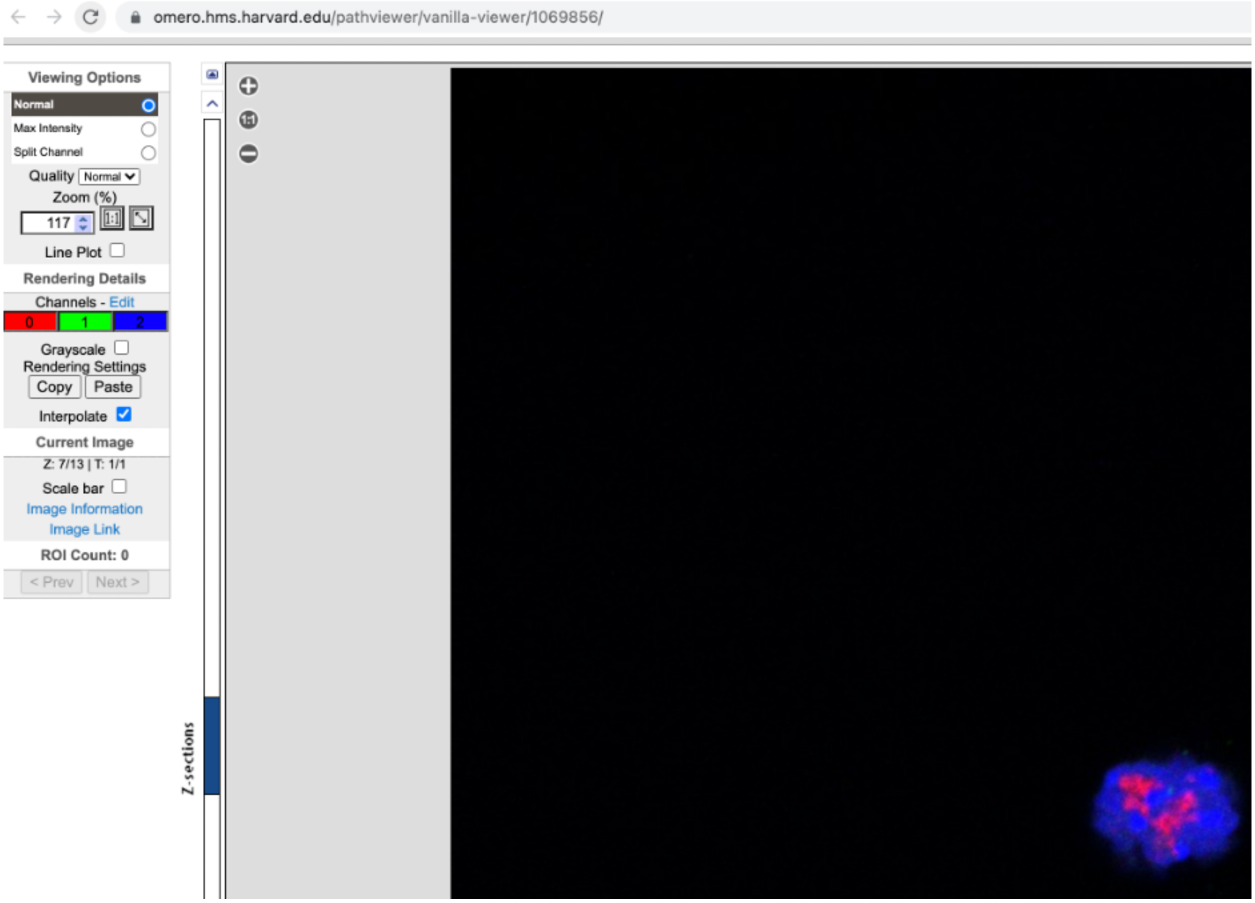
	Genome	Collection	Hub name	Tracks	Add
▸	mm10	Encyclopedia of DNA Elements (ENCODE)	Mouse ENCODE	1616	+
▸	mm10	4D Nucleome Network	4DN datasets	670	+
▸	mm10	Toxicant Exposures and Responses by Genomic and Epigenomic Regulators of Transcripti…	Mouse TaRGET	965	+
▸	mm10	3D structures	3D structures from Science 2018 Aug 31;361(6405):924-928	10	+
▸	mm10	3D structures	3D structures from Nat Struct Mol Biol 2019 Apr;26(4):297-307	1227	+
▸	mm10	3D structures	3D structures from Cell 2021 Feb 4;184(3):741-758.e17	9770	+
▸	mm10	Image collection	4dn image data	1	+
▸	mm10	Encyclopedia of DNA Elements (ENCODE)	Mouse ENCODE from ENCODE data portal	13001	+
▸	mm10	International Human Epigenome Consortium (IHEC)	International Human Epigenome Consortium (IHEC) epigenomic datasets	266	+
▸	mm10	HiC interaction from HiGlass	HiC interaction from HiGlass	79	+

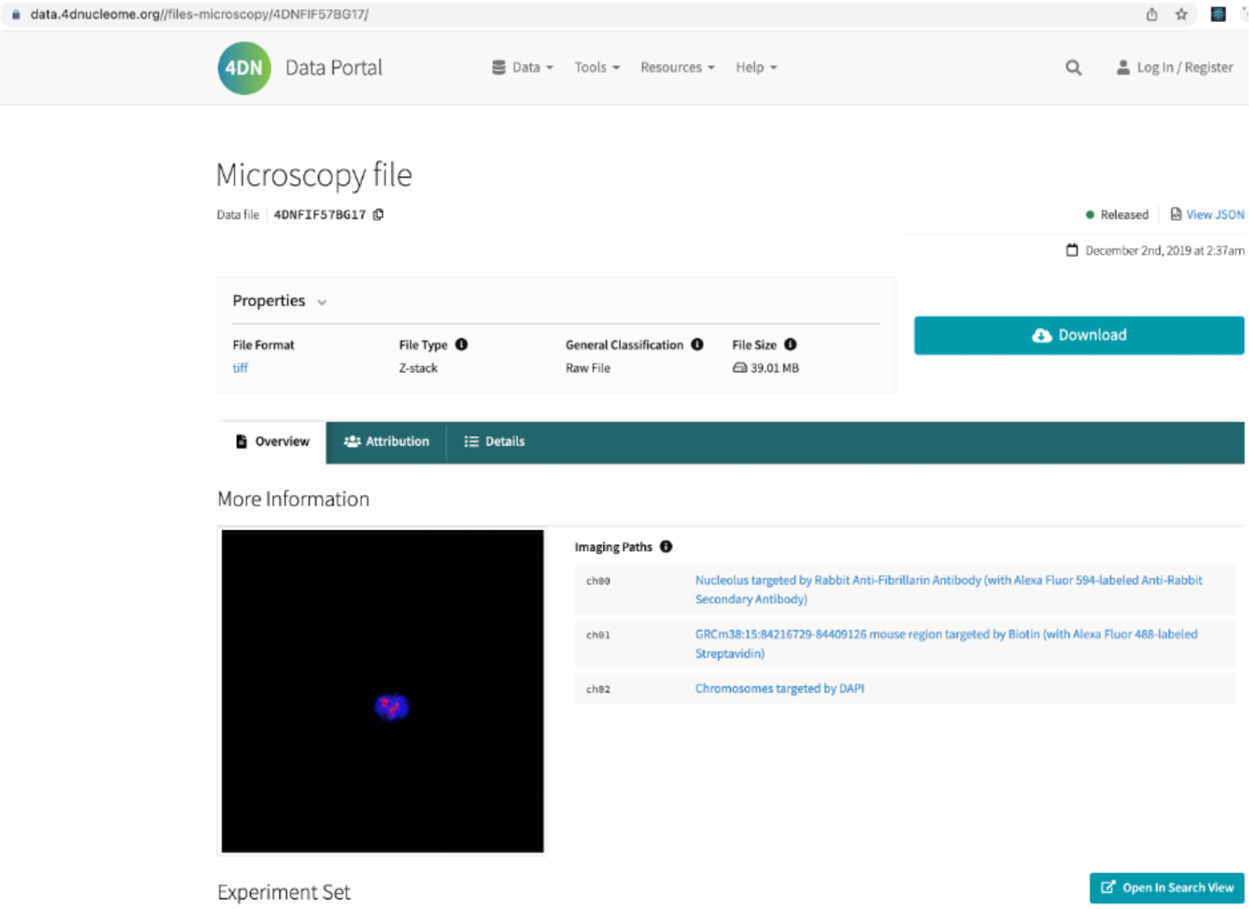
You can see the image track is loaded, and you can check metadata, open image in new panel:



accession	4DNFIF57BG17
id	https://data.4dnucleome.org//files-microscopy/4DNFIF57BG17/
4DN link	https://data.4dnucleome.org//files-microscopy/4DNFIF57BG17/@@download/4DNFIF57BG17.tiff
assay_info	Nucleolus, GRCm38:15:84216729-84409126 mouse region, Chromosomes
biosource_name	F121-9-CASTx129 (Tier 2)
condition	non-NAD BAC probe pPK1003
dataset	DNA-FISH in F121-9 cells
experiment_bucket	raw file
experiment_type	DNA FISH
lab_name	Paul Kaufman, UMMS
replicate_info	Biorep 3, Techrep 1

Besides the link to Omero server is provided, there is also a button which links you to the 4DN details page:


omero.hms.harvard.edu/pathviewer/vanilla-viewer/1069856/



Microscopy file

Data file 4DNFIF57BG17

● Released

December 2nd, 2019 at 2:37am

Properties

File Format	File Type	General Classification	File Size
tiff	Z-stack	Raw File	39.01 MB

More Information

Imaging Paths

ch00	Nucleolus targeted by Rabbit Anti-Fibrillarin Antibody (with Alexa Fluor 594-labeled Anti-Rabbit Secondary Antibody)
ch01	GRCm38:15:84216729-84409126 mouse region targeted by Biotin (with Alexa Fluor 488-labeled Streptavidin)
ch02	Chromosomes targeted by DAPI

Experiment Set

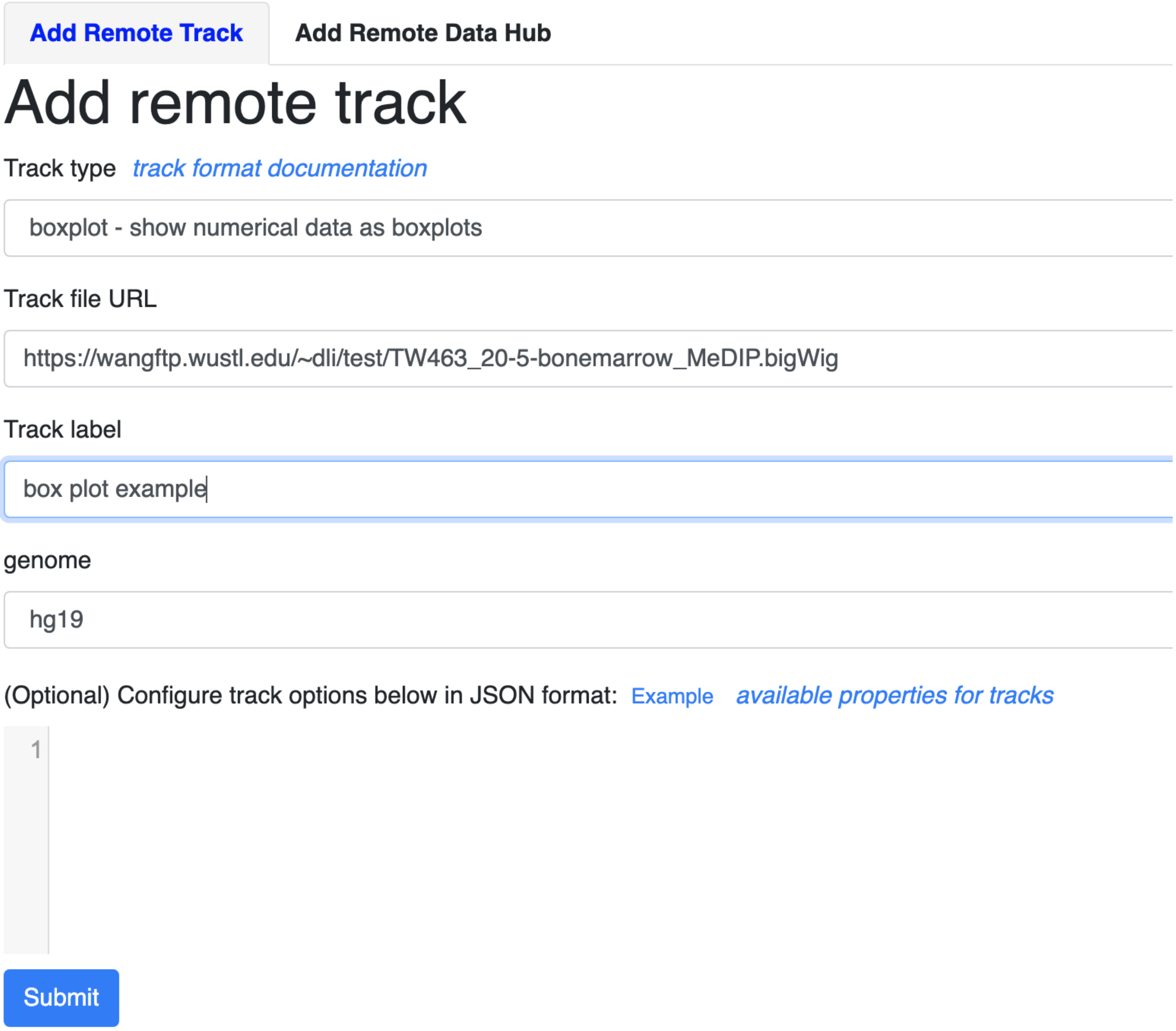
CHAPTER

TEN

STATISTICAL TRACKS

10.1 boxplot

`boxplot` is a track type that show data as boxplots. It accepts numberical data (*bigWig* or *bedGraph*) as track files. To submit a *boxplot* track, it' same as submit a numerical track, instead choose *boxplot* from the track type dropdown menu:



Add remote track

Track type *track format documentation*

boxplot - show numerical data as boxplots

Track file URL

https://wangftp.wustl.edu/~dli/test/TW463_20-5-bonemarrow_MeDIP.bigWig

Track label

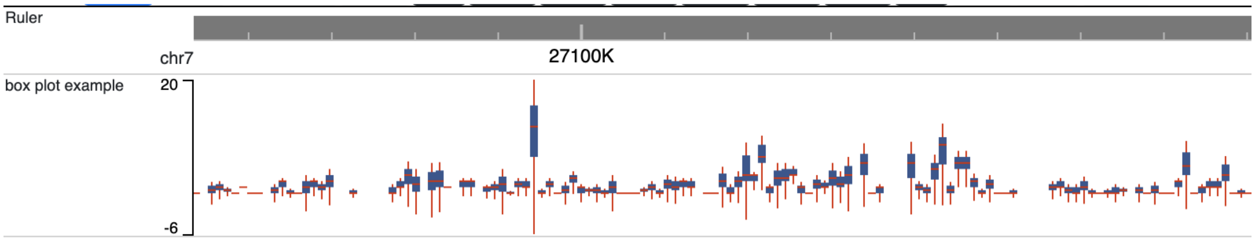
box plot example

genome

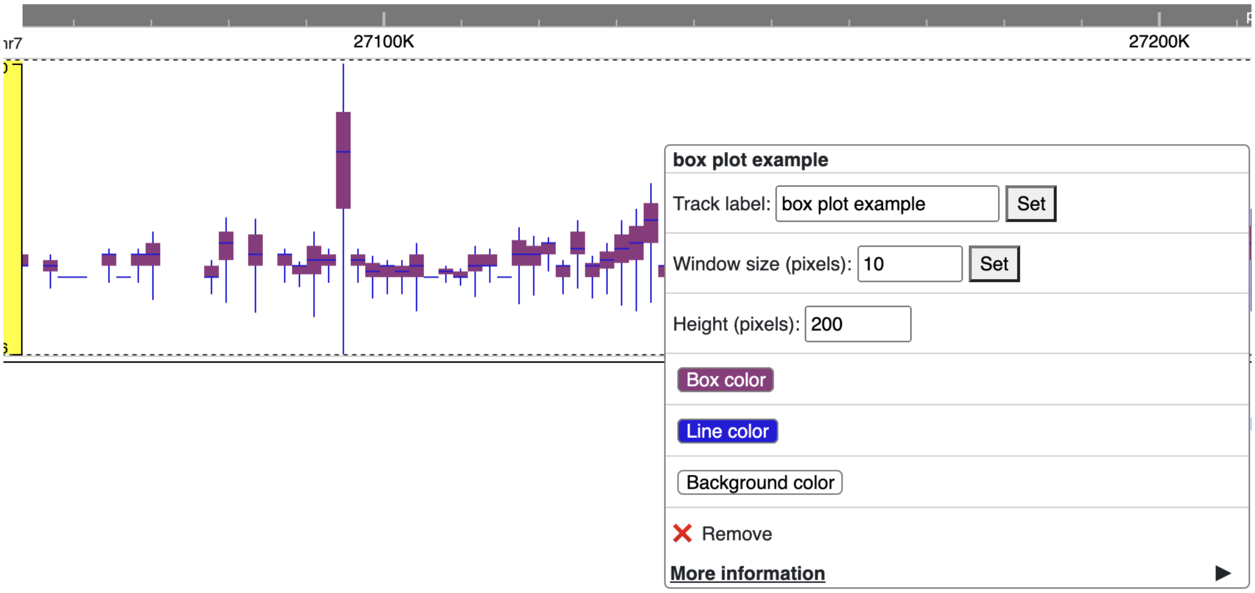
hg19

(Optional) Configure track options below in JSON format: Example *available properties for tracks*

The default view after submit a boxplot track:



Right clicking the track brings the configuration menu, from which window size, height, box and line colors etc can be customized:



box plot example

Track label: box plot example Set

Window size (pixels): 10 Set

Height (pixels): 200

CHAPTER

ELEVEN

INSTALLATION

11.1 Setup on MacOSX/Linux

- Install NodeJS from https://nodejs.org/en/

Note: Feel free to use any package manager tool on your system for installation (`brew`, etc.).

- Get the source code from our github repo: https://github.com/lidaof/eg-react

Important: For MacOSX with Apple M1 Chip, please use NodeJS version 16 and above, then try with `npm install --force`.

11.2 Start the browser

1. Enter the `frontend` directory
2. **Type `npm install` (just for the first time)**
   This step will install dependent packages.
3. Type `npm start`

Warning: if `npm install` gives you error, you might try `npm install --force`.

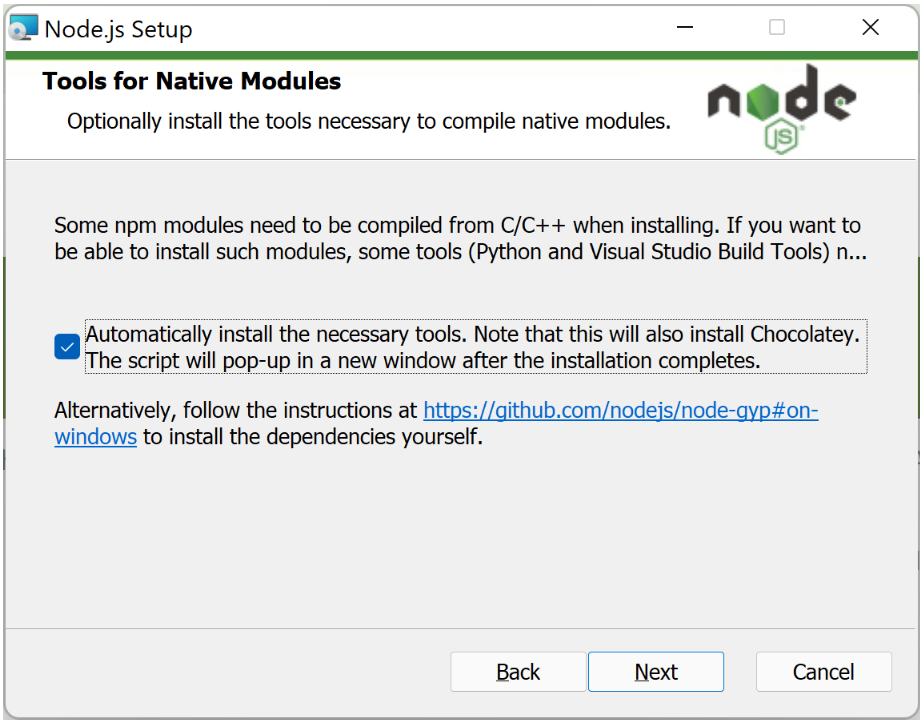
That's it! You are done with your mirror site. The browser is now accessible from http://localhost:3000/browser.

11.3 Setup on Windows

To run the browser App on Windows, you can either install a subsystem for Linux, after that, the steps are pretty much same as in MacOSX and Linux, for more about linux system in Windows, please check Install Windows Subsystem for Linux (WSL).

Another option is to run the App directly in Windows, steps are also similar, assume you are using PowerShell.

1. install nodejs: (you may need check *install necessary tools*)



Node.js Setup

Tools for Native Modules

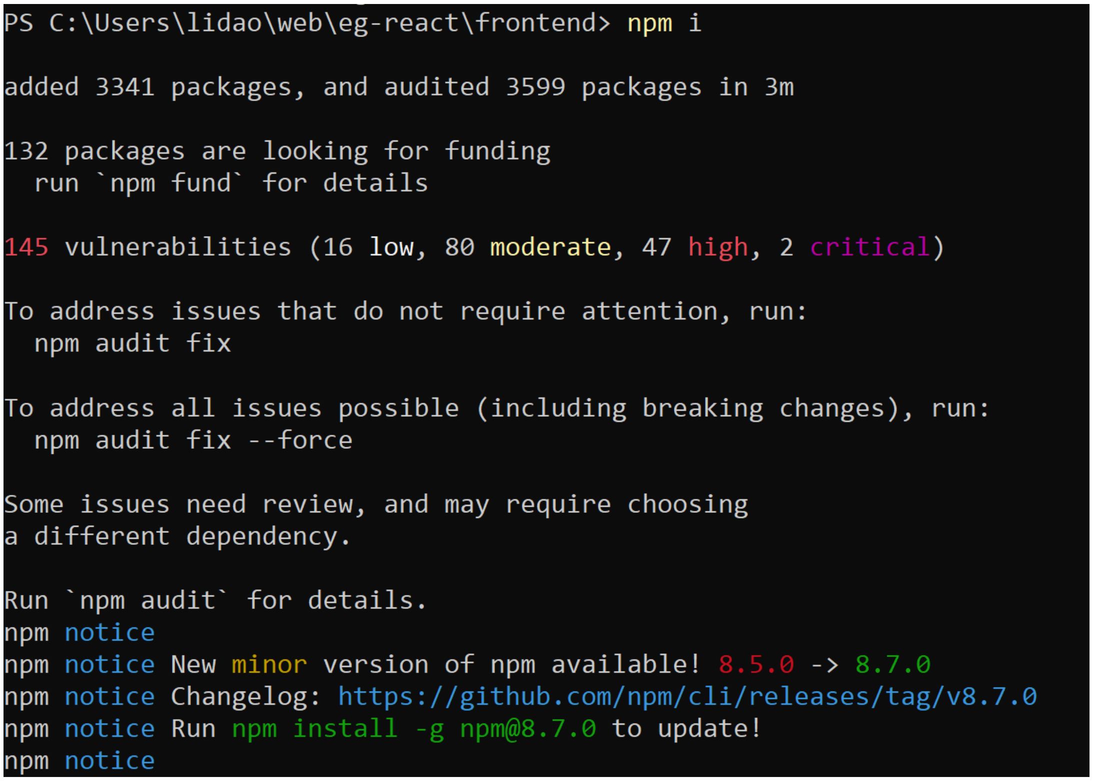
Optionally install the tools necessary to compile native modules.

Some npm modules need to be compiled from C/C++ when installing. If you want to be able to install such modules, some tools (Python and Visual Studio Build Tools) n...

☑ Automatically install the necessary tools. Note that this will also install Chocolatey. The script will pop-up in a new window after the installation completes.

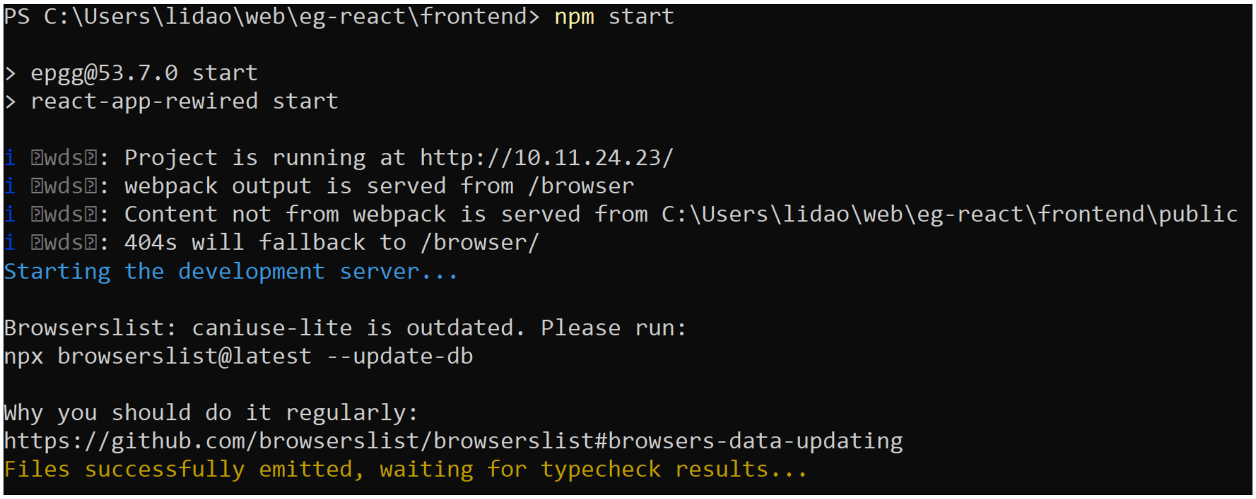
Alternatively, follow the instructions at https://github.com/nodejs/node-gyp#on-windows to install the dependencies yourself.

2. install git: https://git-scm.com/download/win
3. get the code from github and run `npm install` under `frontend` folder:



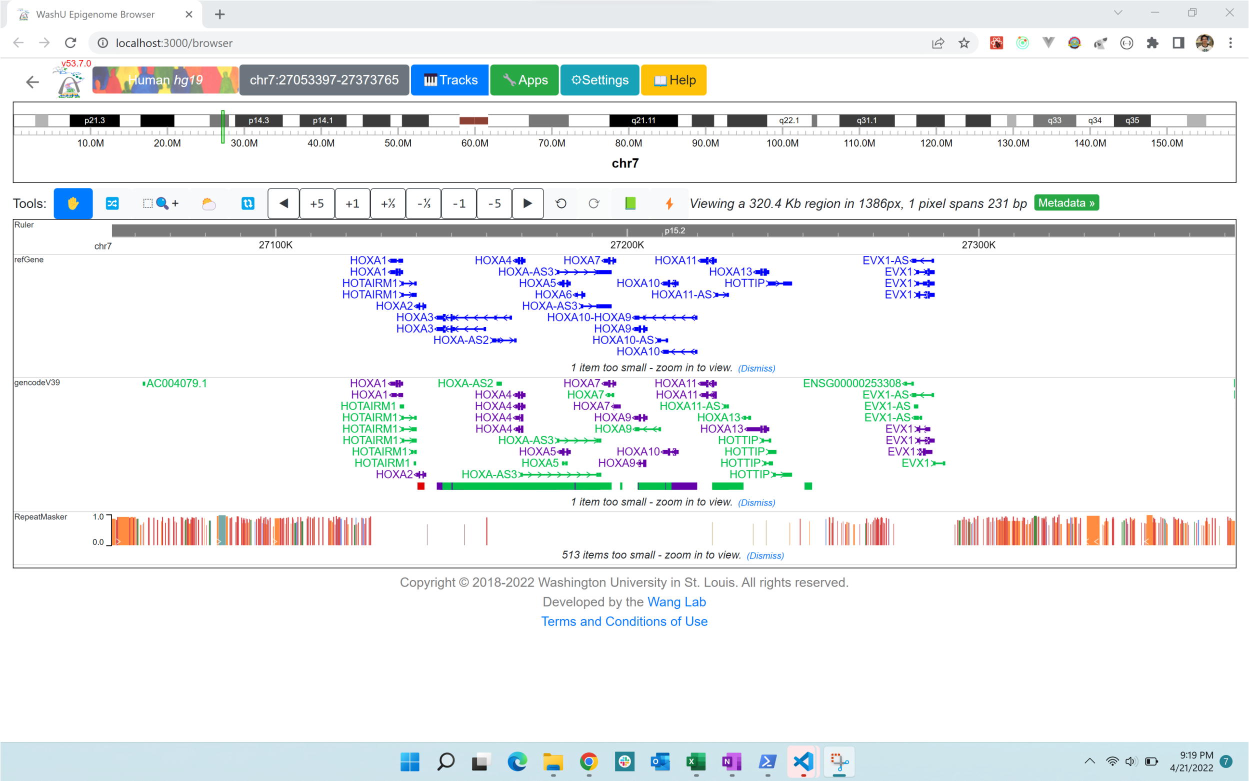
```
PS C:\Users\lidao\web\eg-react\frontend> npm i

added 3341 packages, and audited 3599 packages in 3m

132 packages are looking for funding
  run `npm fund` for details

145 vulnerabilities (16 low, 80 moderate, 47 high, 2 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues possible (including breaking changes), run:
  npm audit fix --force

Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.
npm notice
npm notice New minor version of npm available! 8.5.0 -> 8.7.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.7.0
npm notice Run npm install -g npm@8.7.0 to update!
npm notice
```

4. run `npm start` to start the browser in local development:



```
PS C:\Users\lidao\web\eg-react\frontend> npm start

> epgg@53.7.0 start
> react-app-rewired start

i □wds□: Project is running at http://10.11.24.23/
i □wds□: webpack output is served from /browser
i □wds□: Content not from webpack is served from C:\Users\lidao\web\eg-react\frontend\public
i □wds□: 404s will fallback to /browser/
Starting the development server...

Browserslist: caniuse-lite is outdated. Please run:
npx browserslist@latest --update-db

Why you should do it regularly:
https://github.com/browserslist/browserslist#browsers-data-updating
Files successfully emitted, waiting for typecheck results...
```

5. browser is running under Windows:



WashU Epigenome Browser
localhost:3000/browser
v53.7.0
Human hg19
chr7:27053397-27373765
Tracks
Apps
Settings
Help
chr7
Tools:
Viewing a 320.4 Kb region in 1386px, 1 pixel spans 231 bp
Metadata »
Ruler
refGene
gencodeV39
RepeatMasker
1 item too small - zoom in to view. (Dismiss)
513 items too small - zoom in to view. (Dismiss)
Copyright © 2018-2022 Washington University in St. Louis. All rights reserved.
Developed by the Wang Lab
Terms and Conditions of Use
9:19 PM
4/21/2022

11.4 Example commands for installation on a RHEL system

```
$ cat /etc/redhat-release
 Red Hat Enterprise Linux Server release 7.9 (Maipo)
 # remove system nodejs (optional)
 sudo yum remove nodejs
 #install n for node version control
 curl -L https://git.io/n-install | bash
```

(continues on next page)

(continued from previous page)

```
source .bashrc
$ node -v
v14.17.0

# get the browser code go to frontend folder
git clone https://github.com/lidaof/eg-react.git
cd eg-react/frontend/
npm install --force
npm install react-app-rewired
npm start

# if get error like: System limit for number of file watchers reached, run this command below

echo fs.inotify.max_user_watches=524288 | sudo tee -a /etc/sysctl.conf && sudo sysctl -p

# details see: https://stackoverflow.com/questions/55763428/react-native-error-enospc-system-limit-for-number-of-file-watchers-reached
```

11.5 Setup your own backend API (optional)

By default, your local browser mirror site uses our API service at `https://lambda.epigenomegateway.org/v2`, while if you find the species or assembly you are interested is not listed by our API, you can either contact us to add it or build your own API. To build your own API, please follow the steps below:

1. Install MongoDB from https://www.mongodb.com/
2. start mongdb server
3. Enter the `backend` directory
4. Type `npm install`

Then prepare your gene annotation files like the ones for `hg19`, `mm10` etc:

1. Make sure MongoDB is running
2. Enter the `backend` directory
3. **Run `npm run setup`**

   This step will load the gene annotation data to the MongoDB database
4. Type `npm start`

Now your own backend API is running, change `AWS_API` variable to empty string in `GeneSource.js` file. After this you are using your own API for gene annotation tracks and gene search.

Our current API in serice in `GeneSource.js`:

> export const AWS_API = "https://lambda.epigenomegateway.org/v2";

This API is for testing only:

> https://api.epigenomegateway.org/documentation

11.6 Firebase setup

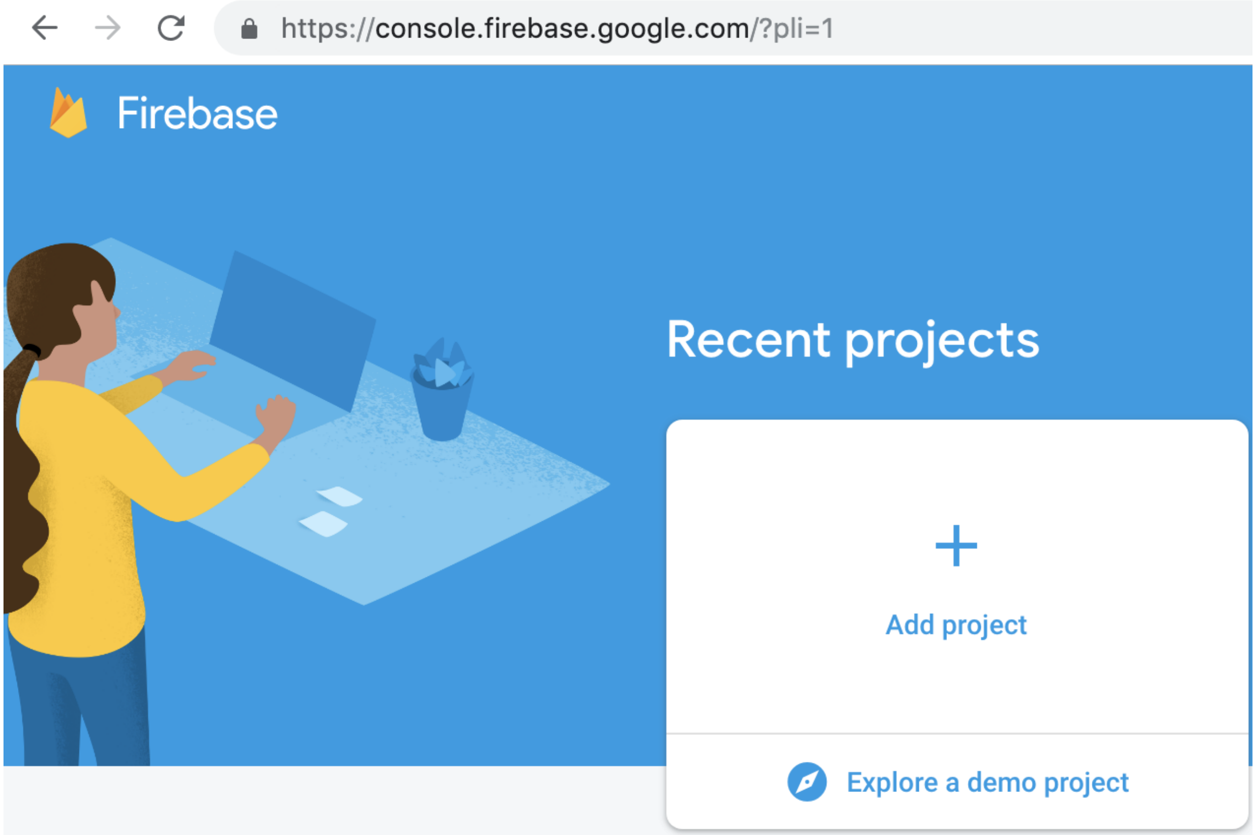
If you installed a local browser mirror, you also need setup a Firebase instance to enable `Session` and `Go Live` function, signup a Firebase account at https://firebase.google.com/, which is free.

Create a `.env` file under `frontend/` folder with following content:

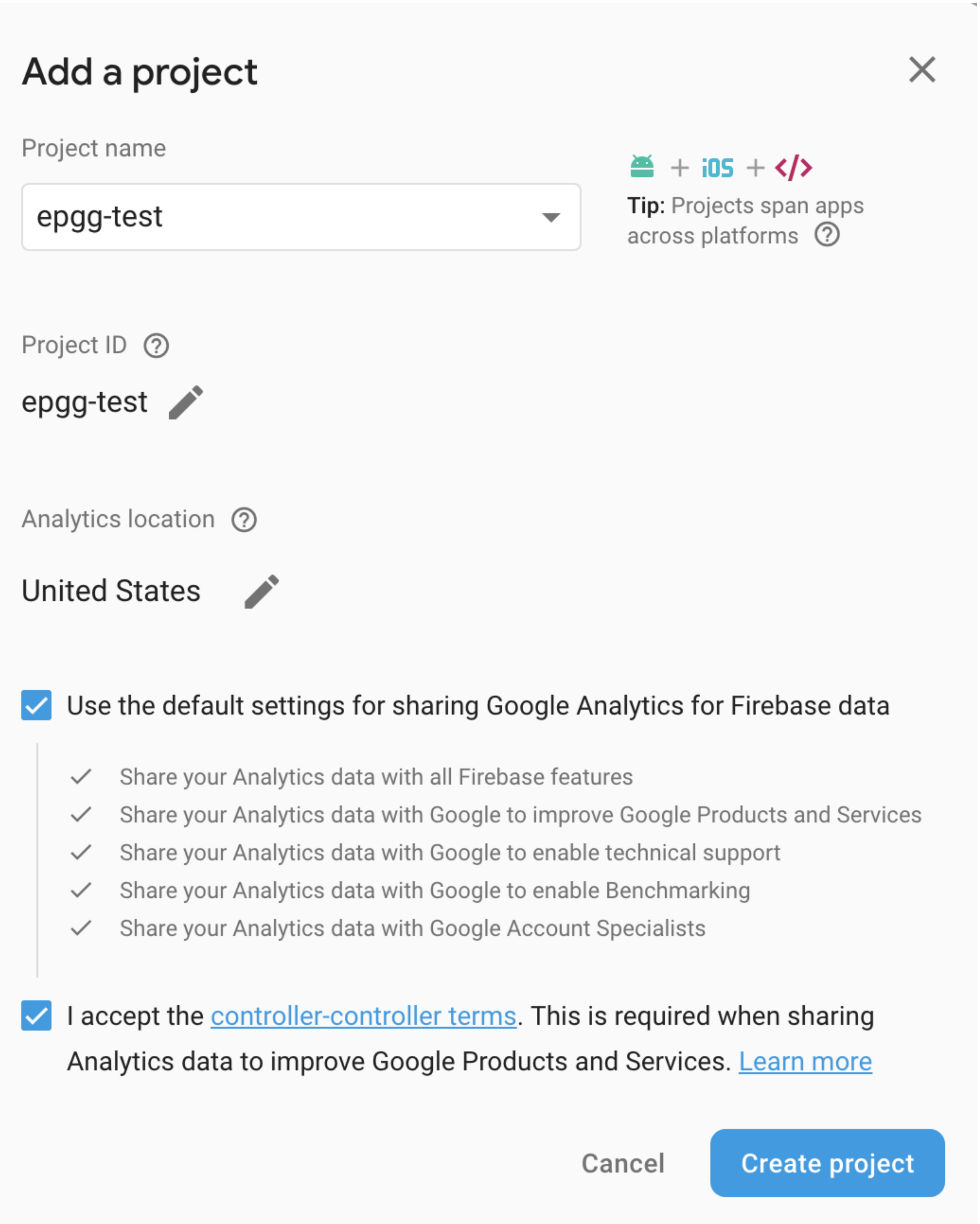
```
REACT_APP_FIREBASE_KEY="Your own info"
REACT_APP_FIREBASE_DOMAIN="Your own info"
REACT_APP_FIREBASE_DATABASE="Your own info"
REACT_APP_FIREBASE_STORAGE_BUCKET="Your own info"
```

The detailed steps of how to get the information above are illustrated in the following screenshots:

Signup a firebase account at Google if you don't have one, then login into your account, create a new prioject:



Type in the project name and click the Create project button:



Add a project

Project name

epgg-test

Tip: Projects span apps across platforms

Project ID

epgg-test

Analytics location

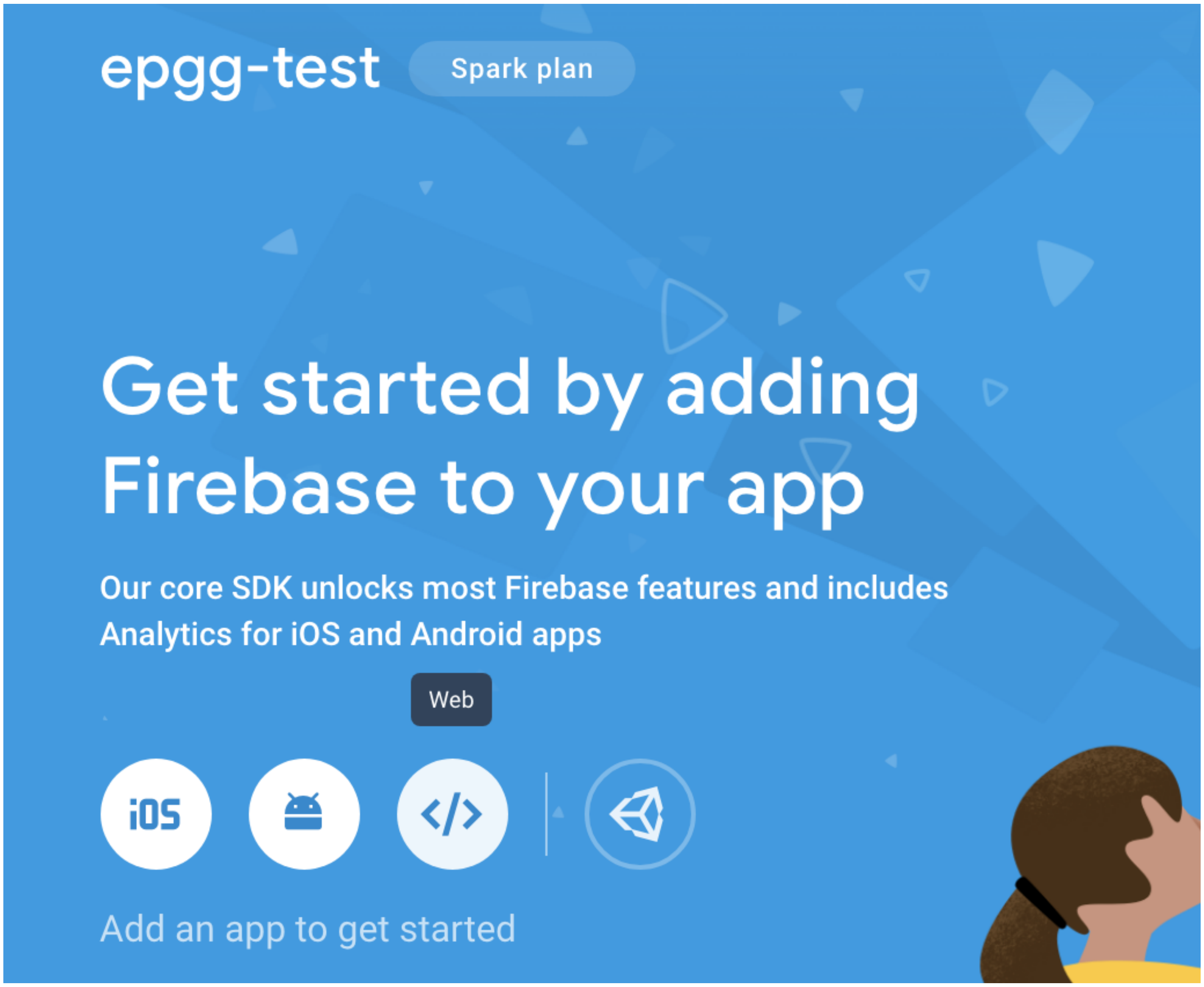
United States

☑ Use the default settings for sharing Google Analytics for Firebase data

- ✓ Share your Analytics data with all Firebase features
- ✓ Share your Analytics data with Google to improve Google Products and Services
- ✓ Share your Analytics data with Google to enable technical support
- ✓ Share your Analytics data with Google to enable Benchmarking
- ✓ Share your Analytics data with Google Account Specialists

☑ I accept the controller-controller terms. This is required when sharing Analytics data to improve Google Products and Services. Learn more

Click the Web button to add a Web app:



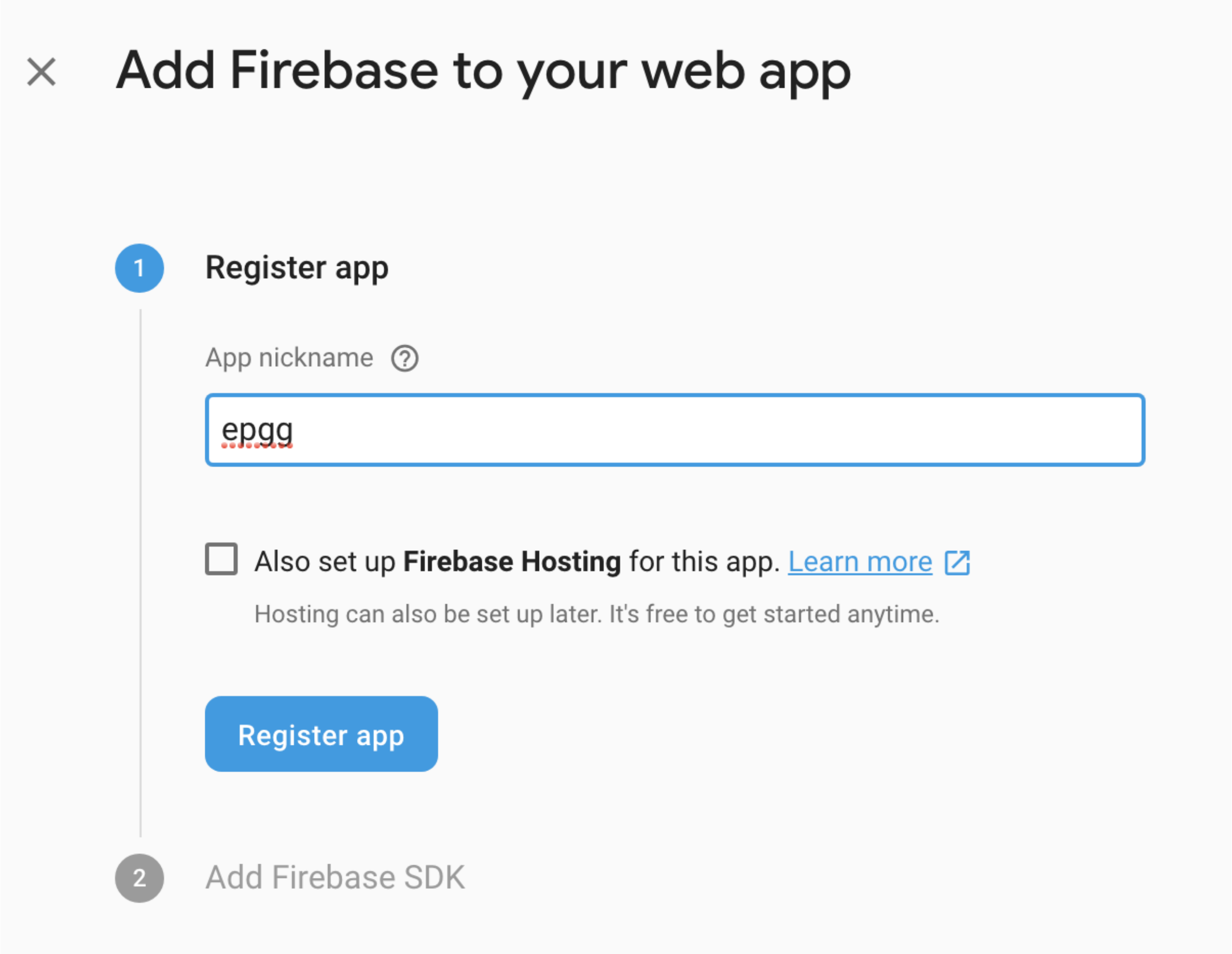
epgg-test

Get started by adding Firebase to your app

Our core SDK unlocks most Firebase features and includes Analytics for iOS and Android apps

Add an app to get started

Type in a web app name and click the Register app button:



Add Firebase to your web app

1 Register app

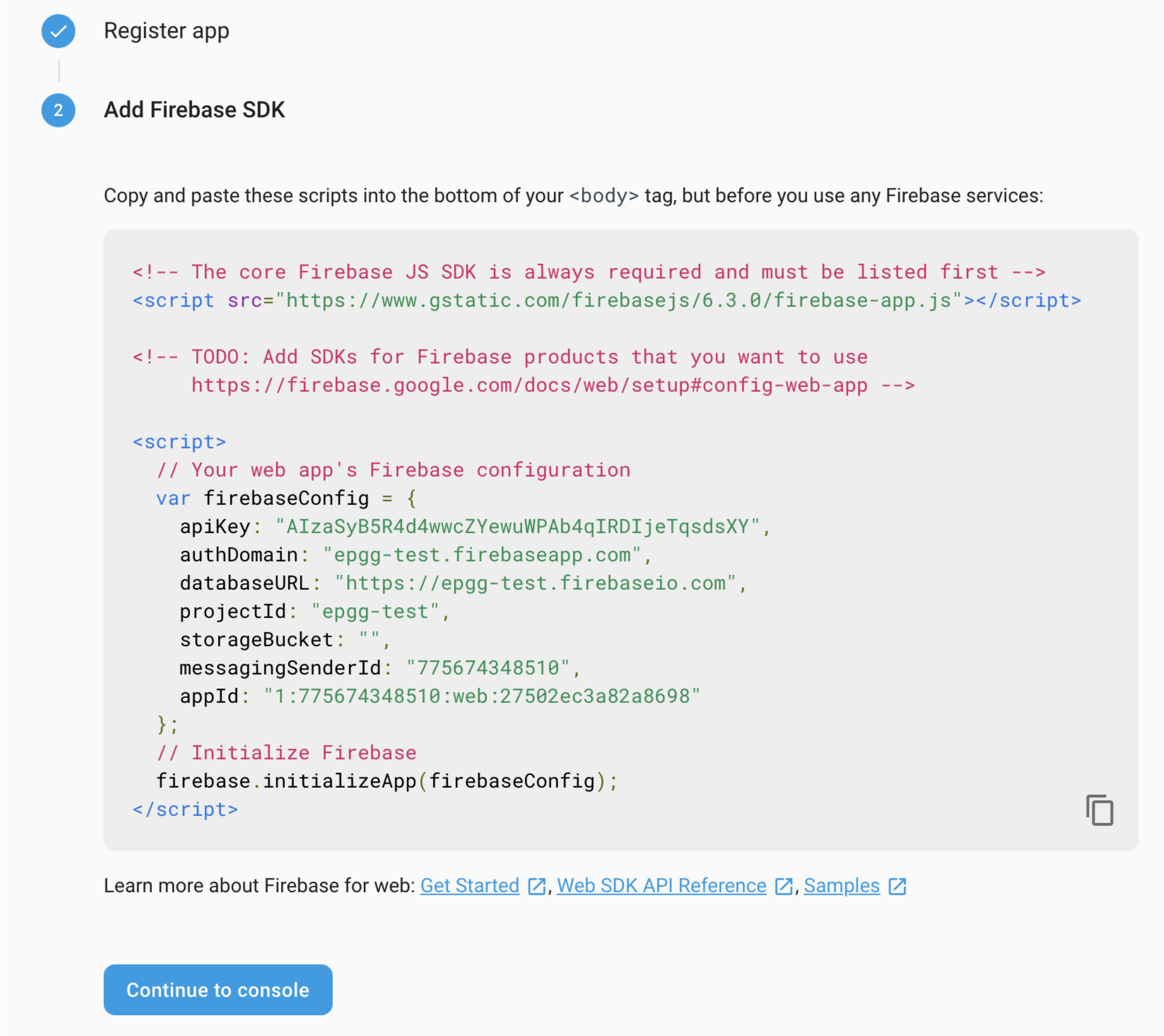
App nickname

epgg

□ Also set up **Firebase Hosting** for this app. Learn more

Hosting can also be set up later. It's free to get started anytime.

The firebase configuration info will be displayed:



Register app

2 Add Firebase SDK

Copy and paste these scripts into the bottom of your <body> tag, but before you use any Firebase services:

```
<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/6.3.0/firebase-app.js"></script>

<!-- TODO: Add SDKs for Firebase products that you want to use
     https://firebase.google.com/docs/web/setup#config-web-app -->

<script>
  // Your web app's Firebase configuration
  var firebaseConfig = {
    apiKey: "AIzaSyB5R4d4wwcZYewuWPAb4qIRDIjeTqsdsXY",
    authDomain: "epgg-test.firebaseapp.com",
    databaseURL: "https://epgg-test.firebaseio.com",
    projectId: "epgg-test",
    storageBucket: "",
    messagingSenderId: "775674348510",
    appId: "1:775674348510:web:27502ec3a82a8698"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
</script>
```

Learn more about Firebase for web: Get Started, Web SDK API Reference, Samples

11.7 Use without firebase

Firebase setup is necessary for using with Session and Live function, if browser mirror users think they won't be necessary, the firebase setup setup can be avoided then.

In the `frontend` folder, create a `.env` file, add the line below:

> REACT_APP_NO_FIREBASE=1

rerun `npm start`, the browser will start without session/live function.

CHAPTER TWELVE

USE DOCKER

The browser is also available as Docker images, to run the browser instance, get Docker from https://www.docker.com/, our official docker image page is at https://cloud.docker.com/repository/docker/epgg/eg-react, the image is based on Ubuntu 18.04, to run the image, run following commands:

```
docker run -it -p 3000:3000 epgg/eg-react
```

Note: The first 3000 port is the port will be used on your local computer, you can change it to any other port.

After the docker image is running, to start the browser:

```
cd eg-react/frontend
npm start
```

Open your web browser and locate to http://localhost:3000 to see the browser.

CHAPTER

THIRTEEN

EMBEDDING

To embed the browser in any HTML file, create a HTML page with following contents: (the example shows how to embed a mouse browser with 2 bigWig tracks from ENCODE data portal)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no
↪" />
    <meta name="theme-color" content="#000000" />
    <title>WashU Epigenome Browser</title>
    <link
        rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
        integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/
↪dAiS6JXm"
        crossorigin="anonymous"
    />
    <script src="https://aframe.io/releases/0.8.0/aframe.min.js"></script>
    <script
        src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
        integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/
↪GpGFF93hXpG5KkN"
        crossorigin="anonymous"
    ></script>
    <script
        src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
        integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/
↪ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
        crossorigin="anonymous"
    ></script>
    <script
        src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
        integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/
↪JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
        crossorigin="anonymous"
    ></script>
    <script crossorigin src="https://unpkg.com/react@16/umd/react.development.js"></
↪script>
    <script crossorigin src="https://unpkg.com/react-dom@16/umd/react-dom.development.js
```

(continues on next page)

(continued from previous page)

```
↪"></script>
    <script crossorigin src="https://cdn.plot.ly/plotly-cartesian-latest.min.js"></
↪script>
    <script crossorigin src="https://unpkg.com/react-plotly.js@2.3.0/dist/create-plotly-
↪component.min.js"></script>
    <!-- the browser script and styles -->
    <script src="https://unpkg.com/epgg@53.6.0/umd/epgg.js"></script>
    <link rel="stylesheet" href="https://unpkg.com/epgg@53.6.0/umd/epgg.css" />
</head>

<body>
    <noscript> You need to enable JavaScript to run this app. </noscript>
    <h1>Embedding test</h1>
    <div id="embed" style="width: 1000px"></div>
    <h2>some other headings</h2>

    <script>
        const container = document.getElementById("embed");
        const contents = {
            genomeName: "mm10",
            displayRegion: "chr5:51997494-52853744",
            trackLegendWidth: 120,
            isShowingNavigator: true,
            tracks: [
                {
                    type: "geneannotation",
                    name: "refGene",
                    genome: "mm10",
                },
                {
                    type: "geneannotation",
                    name: "gencodeM19Basic",
                    genome: "mm10",
                },
                {
                    type: "ruler",
                    name: "Ruler",
                },
                {
                    type: "bigWig",
                    name: "ChipSeq of Heart",
                    url: "https://www.encodeproject.org/files/ENCFF641FBI/@@download/
↪ENCFF641FBI.bigWig",
                    options: { color: "red" },
                    metadata: { Sample: "Heart" },
                },
                {
                    type: "bigWig",
                    name: "ChipSeq of Liver",
                    url: "https://www.encodeproject.org/files/ENCFF555LBI/@@download/
↪ENCFF555LBI.bigWig",
                    options: { color: "blue" },
```

(continues on next page)

(continued from previous page)

```
                    metadata: { Sample: "Liver" },
                },
                {
                    type: "bedGraph",
                    name: "test",
                    url: "https://wangftp.wustl.edu/~rsears/Stuart_Little/RNA_083018/
↪WangT_7176-5_ALDH_STRANDED_Signal.Unique.combo.out.bg.gz",
                },
            ],
            metadataTerms: ["Sample"],
            regionSets: [],
            regionSetViewIndex: -1,
        };
        renderBrowserInElement(contents, container);
        </script>
    </body>
</html>
```

The key API is the function `renderBrowserInElement`, it accepts the contents array as first argument, and container as second argument which is a DOM element.

CHAPTER

FOURTEEN

FRONTEND CODE ARCHITETURE

Note: This section explains how frontend code is organized, intend to be used for development purpose. Regular browser users don't need to care about this section.

14.1 Quick tour

The client code is in the `frontend` folder. Here is a quick tour of `frontend/src`:

- `components`: All React components.
  - `genomeNavigator`: The navigation bar at the top that allows users to navigate
  - `track`: Track-related components
  - `trackManagers`: UI that manages adding tracks
- `dataSources`: API calls, AJAX calls, database connections, etc. that get data to display.
- `model`: Data models.
- `stories`: Stories for Storybook on which unit tests depend.
- `vendor`: 3rd-party libraries that are not in NPM.

14.2 Suggested order of reading

If you plan to understand the app as a whole here is a suggested order to read the code in:

1. `Feature`: A feature or annotation in the genome.
2. `NavigationContext`: A list of `Feature`s that represent everywhere a user can navigate. If the `Feature`s are actually entire chromosomes then the user can effectively navigate the whole genome.
3. `DisplayedRegionModel`: An interval in a `NavigationContext`.
4. `App`: The root component of the app.
5. From `App`, descend into interested components.

14.3 Making a new track type

14.3.1 Make a new TrackConfig

Make a new class that extends TrackConfigor one of its subclasses. This class packages many essential track characteristics:

- getComponent() - Gets the component that renders the main visualizer and legend of the track.
- getMenuComponents() - Specifies context menu items in an array of components. You can choose existing ones in the contextMenu directory or make new ones.
- getOptions() - The visualizer probably renders with default options like a color. This method returns a plain object containing those options.

You do not have to implement these methods immediately as the base TrackConfig class provides minimal defaults. Just work on making the browser render *some* temporary placeholder at first.

14.3.2 Specify when to use the TrackConfig

1. Import your new TrackConfig into trackConfig/getTrackConfig.js.
2. Add an appropriate entry to TYPE_NAME_TO_SUBTYPE, which maps track type name to track renderer.

14.3.3 Write a new track visualizer component (implement getComponent())

1. Make a new component expecting to receive a bunch of props from TrackContainer. Track.js documents the props to expect.
2. If you need data assume it will come through the data prop. We will add data fetch in the next step.
3. Your new component may render anything though it is **highly** recommended you render a <Track> component, if not one of the more specialized components like <AnnotationTrack> or <NumericalTrack>. Pass *all* track container props to these sub-components.
4. In addition to track container props you need to provide certain props to these sub-components, all of which the respective files document.
   - For example, <Track> requires a legend and visualizer element. Use the track container props, which includes view region and width, to render a visualizer and pass it to <Track>.

14.3.4 Add data fetch

Available data sources are in the dataSources folder. If none of them fulfill your needs, write a new class that fulfills the interface of DataSource.js. More can be found in that file.

How do we give your visualizer data? Higher-order components! track/commonComponents contains track-specific HOCs; their names start with config- or with-.

configStaticDataSource requests a callback that returns a DataSource and then returns a *function* that wraps React components. After you use this function, a component will automatically receive three props data, isLoading, and error. These update with the browser's current view region. In particular, the HOC guarantees synchronization of the data prop with the current view region if isLoading is false.

14.3.5 2. Specify context menu components (implement `getMenuComponents()`)

Specify context menu items with an array of components. You can choose existing ones in the `contextMenu` directory or make new ones.

- Make sure the method returns Component *classes*, not component instances.

14.3.6 3. Specify default options

Default option objects look like the `options` prop of `TrackModel` objects. Context menu items will read these options if the track model does not specify them. Make sure these options are consistent with the way you are rendering your track component! The `configOptionMerging` HOC should help with that.

Once you have a default options object, call `setDefaultOptions()` in the constructor of `TrackConfig` to use them.

14.4 Performance tips

Querying the width or height of any element, for example through `clientWidth` or `getBoundingClientRect()`is slow. Such queries take on the order of 2 to 20 ms. While it is fine to do it once or twice, avoid doing it in a loop. Suppose you aim to plot 500 data points on a SVG and for each point you query the SVG's width. That is already a second or more of computation – very noticable to the user!

14.5 React (and other) gotchas

- On Macs, control + click is the same as a right click which fires a `contextmenu` event. Note that `click` events do not fire on `contextmenu` events. The `mousedown` and `mouseup` events will still fire though.
- When using native DOM events they take priority over React events. This is because React waits for events to bubble to the root component before handling them. This can cause undesirable effects: for example, calling `stopPropagation()` on a React event will not actually stop native events. This StackOverflow post may also help if you have propagation problems: https://stackoverflow.com/questions/24415631/reactjs-syntheticevent-stoppropagation-only-works-with-react-events
- React *always* unmounts components if their parents change type. The `Reparentable` component works around this by using app-unique IDs, but it can cause side effects with React's native events. Use with care.
- Webpack does not support circular dependencies, and while compilation may be successful, an import may resolve as `undefined` at runtime.

14.6 Lessons trying to refactor into WebWorkers

1. Data fetch and track display options are intimately related. For example, what if someone wants HiC data and selects the 5KB resolution option?
2. Thus, for each track type, we have one object that gets the track component, default rendering options, and data fetch/processing.
3. Webpack hangs forever if it encounters a cyclic dependency involving a webworker.
4. The code as in (2) causes a cyclic depdendency. This cycle is [config object] –> [data source] –> [worker] –> [track config deserializer] –> [config object]
5. We cannot have our cake and eat it too.

Unfortunately, this means we cannot pipeline all expensive computation in worker context, while also ensuring track component and data source live in the same place.

CHAPTER

FIFTEEN

ADD A NEW GENOME

Here we will use mouse mm10 for example to illustate how to add a new genome build to the Browser.

15.1 Prepare genome sequence file

The browser expect genome sequence file in the 2bit file developed by UCSC, for mouse mm10, you can get it from http://hgdownload.soe.ucsc.edu/goldenPath/mm10/bigZips/mm10.2bit, after download the 2bit file, you will need put the 2bit file on a web place with CORS access.

15.2 Create the folder for genome configuration files

Create a folder called mm10 under frontend/src/model/genomes, now all the files listed below should be placed under this new mm10 folder.

15.2.1 Get cytoband file

Download the cytoband information from http://hgdownload.soe.ucsc.edu/goldenPath/mm10/database/cytoBandIdeo.txt.gz, unzip it, run the following command to get a file called cytoBand.json:

```
node frontend/src/model/genomes/cytobandTextToJson.js cytoBandIdeo.txt
```

15.2.2 Prepare annotation annotation tracks

In the mm10 folder, create a file called annotationTracks.json with following content:

```
{
    "Ruler": [
        {
            "type": "ruler",
            "label": "Ruler",
            "name": "Ruler"
        }
    ],
    "Genes": [
        {
            "name": "refGene",
```

(continues on next page)

(continued from previous page)

```
            "label": "RefSeq genes",
            "filetype": "geneAnnotation"
        },
        {
            "name": "gencodeM19",
            "label": "GENCODE M19 genes",
            "options": {
                "categoryColors": {
                    "coding": "rgb(0,60,179)",
                    "nonCoding": "rgb(0,128,0)",
                    "pseudogene": "rgb(230,0,172)",
                    "problem": "rgb(255,0,0)",
                    "polyA": "rgb(0,0,51)"
                }
            },
            "filetype": "geneAnnotation"
        },
        {
            "name": "gencodeM19Basic",
            "label": "GENCODE M19 genes (basic set)",
            "options": {
                "categoryColors": {
                    "coding": "rgb(0,60,179)",
                    "nonCoding": "rgb(0,128,0)",
                    "pseudogene": "rgb(230,0,172)",
                    "problem": "rgb(255,0,0)",
                    "polyA": "rgb(0,0,51)"
                }
            },
            "filetype": "geneAnnotation"
        }
    ],
    "RepeatMasker": {
        "All Repeats": [
            {
                "name": "rmsk_all",
                "label": "RepeatMasker",
                "filetype": "repeatmasker",
                "url": "https://vizhub.wustl.edu/public/mm10/rmsk16.bb",
                "height": 30
            }
        ]
    },
    "Genome Comparison": [
        {
            "name": "hg38tomm10",
            "label": "Human hg38 to mm10 blastz",
            "querygenome": "hg38",
            "filetype": "genomealign",
            "url": "https://vizhub.wustl.edu/public/mm10/weaver/mm10_hg38_axt.gz"
        }
    ]
```

(continues on next page)

(continued from previous page)

```
}
```

15.2.3 Get chromosome sizes

Download the file from http://hgdownload.soe.ucsc.edu/goldenPath/mm10/bigZips/mm10.chrom.sizes, you can following awk command to create the chromSize.json file:

```
sort -V mm10.chrom.sizes | awk 'BEGIN{ORS="";print "["}{sep=NR==1?"\n":",\n"; print sep"\t{\"chr\": \""$1"\", \"size\": "$2"}"}END{print "\n]"}' > chromSize.json
```

Move the chromSize.json file into the mm10 folder.

15.2.4 Create genome configuration file

Create a file called mm10.js, filling the following contents:

```
import Chromosome from '../Chromosome';
import Genome from '../Genome';
import TrackModel from '../../TrackModel';
import cytobands from './cytoBand.json';
import annotationTracks from "./annotationTracks.json";
import chromSize from "./chromSize.json";

const allSize = chromSize.map(genom => new Chromosome(genom.chr, genom.size));
const genome = new Genome("mm10", allSize);

const navContext = genome.makeNavContext();
const defaultRegion = navContext.parse("chr6:52425276-52425961");
const defaultTracks = [
    new TrackModel({
        type: "geneAnnotation",
        name: "refGene",
        genome: "mm10",
    }),
    new TrackModel({
        type: "geneAnnotation",
        name: "gencodeM19Basic",
        genome: "mm10",
    }),
    new TrackModel({
        type: "ruler",
        name: "Ruler",
    }),
    new TrackModel({
        type: 'repeatmasker',
        name: 'RepeatMasker',
        url: 'https://vizhub.wustl.edu/public/mm10/rmsk16.bb',
    })
];
```

(continues on next page)

(continued from previous page)

```
const publicHubData = {
    "4D Nucleome Network": "The 4D Nucleome Network aims to understand the principles␣
↪underlying nuclear " +
    "organization in space and time, the role nuclear organization plays in gene␣
↪expression and cellular function, " +
    "and how changes in nuclear organization affect normal development as well as␣
↪various diseases.  The program is " +
    "developing novel tools to explore the dynamic nuclear architecture and its role in␣
↪gene expression programs, " +
    "models to examine the relationship between nuclear organization and function, and␣
↪reference maps of nuclear" +
    "architecture in a variety of cells and tissues as a community resource.",
    "Encyclopedia of DNA Elements (ENCODE)": "The Encyclopedia of DNA Elements (ENCODE)␣
↪Consortium is an " +
        "international collaboration of research groups funded by the National Human␣
↪Genome Research Institute " +
        "(NHGRI). The goal of ENCODE is to build a comprehensive parts list of␣
↪functional elements in the human " +
        "genome, including elements that act at the protein and RNA levels, and␣
↪regulatory elements that control " +
        "cells and circumstances in which a gene is active.",
};

const publicHubList = [
    {
        collection: "4D Nucleome Network",
        name: "4DN HiC datasets",
        numTracks: 23,
        oldHubFormat: false,
        url: "https://vizhub.wustl.edu/public/mm10/4dn_mm10.json",
        description: {
            'hub built by': 'Daofeng Li (dli23@wustl.edu)',
            'hub built date': 'Sep 1 2018',
            'hub built notes': 'metadata information are obtained directly from 4DN data␣
↪portal'
        },
    }
]

const MM10 = {
    genome: genome,
    navContext: navContext,
    cytobands: cytobands,
    defaultRegion: defaultRegion,
    defaultTracks: defaultTracks,
    twoBitURL: "https://vizhub.wustl.edu/public/mm10/mm10.2bit",
    publicHubData,
    publicHubList,
    annotationTracks,
};

export default MM10;
```

defaultRegion

This variable controls the default region when you open the browser for mm10.

defaultTracks

This variable controls default tracks when you open the browser for mm10.

publicHubList

The field contains a list of public hubs.

15.3 Add the new genome to the system

Modify `frontend/src/model/genomes/allGenomes.ts`:

```
import MM10 from './mm10/mm10';
```

Include `MM10` to `allGenomes` variable:

```
const allGenomes = [
    HG19,
    HG38,
    MM10,
    PANTRO5,
    DAN_RER10,
    RN6,
];
```

In variable `treeOfLife` add the entry for mm10:

```
mouse: {
    logoUrl: 'https://epigenomegateway.wustl.edu/browser/images/Mouse.png',
    assemblies: [ MM10.genome.getName() ],
    color: 'white',
},
```

Note: one species can have many assemblies, you can also include *mm9* in the `assemblies` array.

Save all the edits, restart the browser (or recompile) you can see the new added genome assembly.

CHAPTER

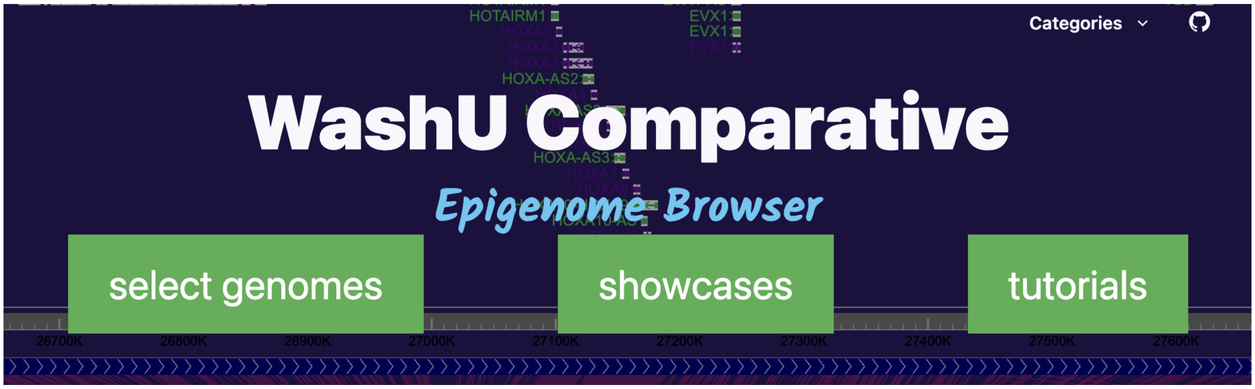
SIXTEEN

THE COMPARATIVE EPIGENOME BROWSER

16.1 Landing page

The WashU Comparative Epigenome Browser is a valuable resource for scientists studying comparative genomics and epigenomics. The browser is available at http://comparativegateway.wustl.edu/. It allows users to easily select and compare multiple assemblies from different species.

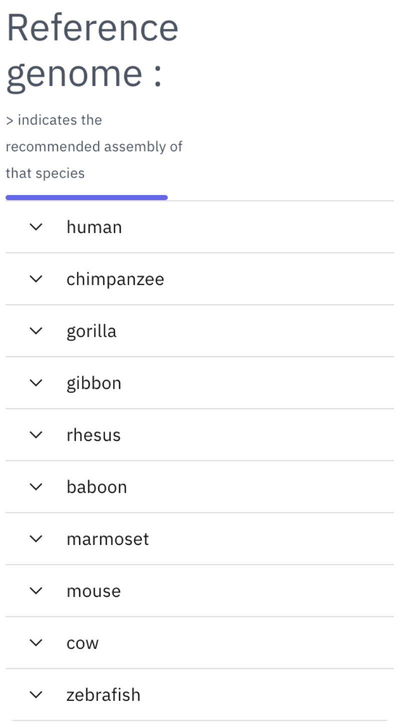
- Click "select genomes" on the page to begin. A few examples are available as "showcases", and video tutorials are available on the "tutorials" page:



16.2 Select a reference genome and one or more secondary genomes

After clicking "select genomes", the species selection tool will become available for users to choose a reference genome. Next, users can select one or multiple species to compare to the reference. For species with multiple assemblies available, we marked one assembly with a ">" as the recommended assembly based on genome completeness and genome-alignment availability.

- The WashU Comparative Epigenome Browser uses a selected reference genome to compare other genomes. Available assemblies can be found in the dropdown menu:

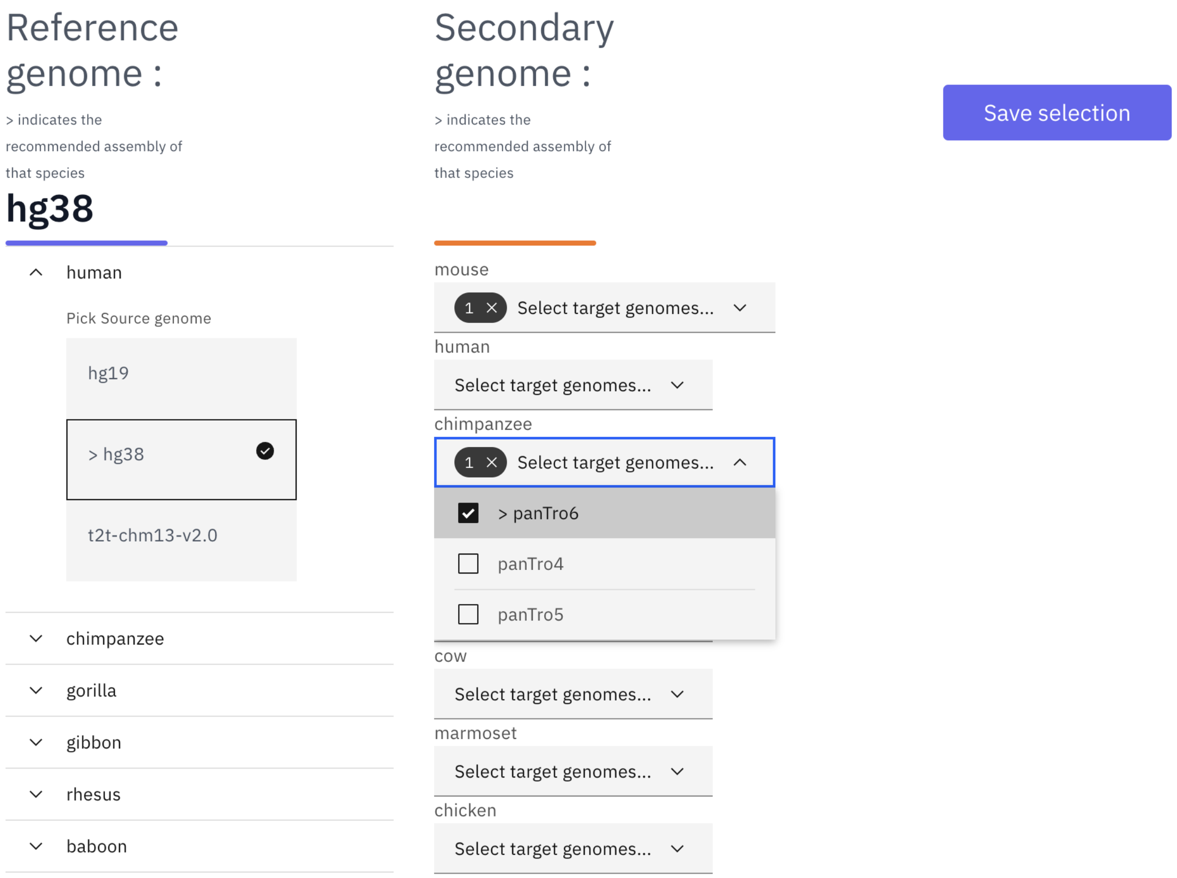


Reference genome :

> indicates the recommended assembly of that species

- human
- chimpanzee
- gorilla
- gibbon
- rhesus
- baboon
- marmoset
- mouse
- cow
- zebrafish

- After selecting the reference genome, users can select available secondary genome(s). In the following example, after selecting hg38 as the reference genome, both mm10 and panTro6 are selected as secondary genomes:



Reference genome :

> indicates the recommended assembly of that species

hg38

human

Pick Source genome

hg19

> hg38 ✓

t2t-chm13-v2.0

chimpanzee

gorilla

gibbon

rhesus

baboon

Secondary genome :

> indicates the recommended assembly of that species

mouse

1 × Select target genomes...

human

Select target genomes...

chimpanzee

1 × Select target genomes...

☑ > panTro6

□ panTro4

□ panTro5

cow

Select target genomes...

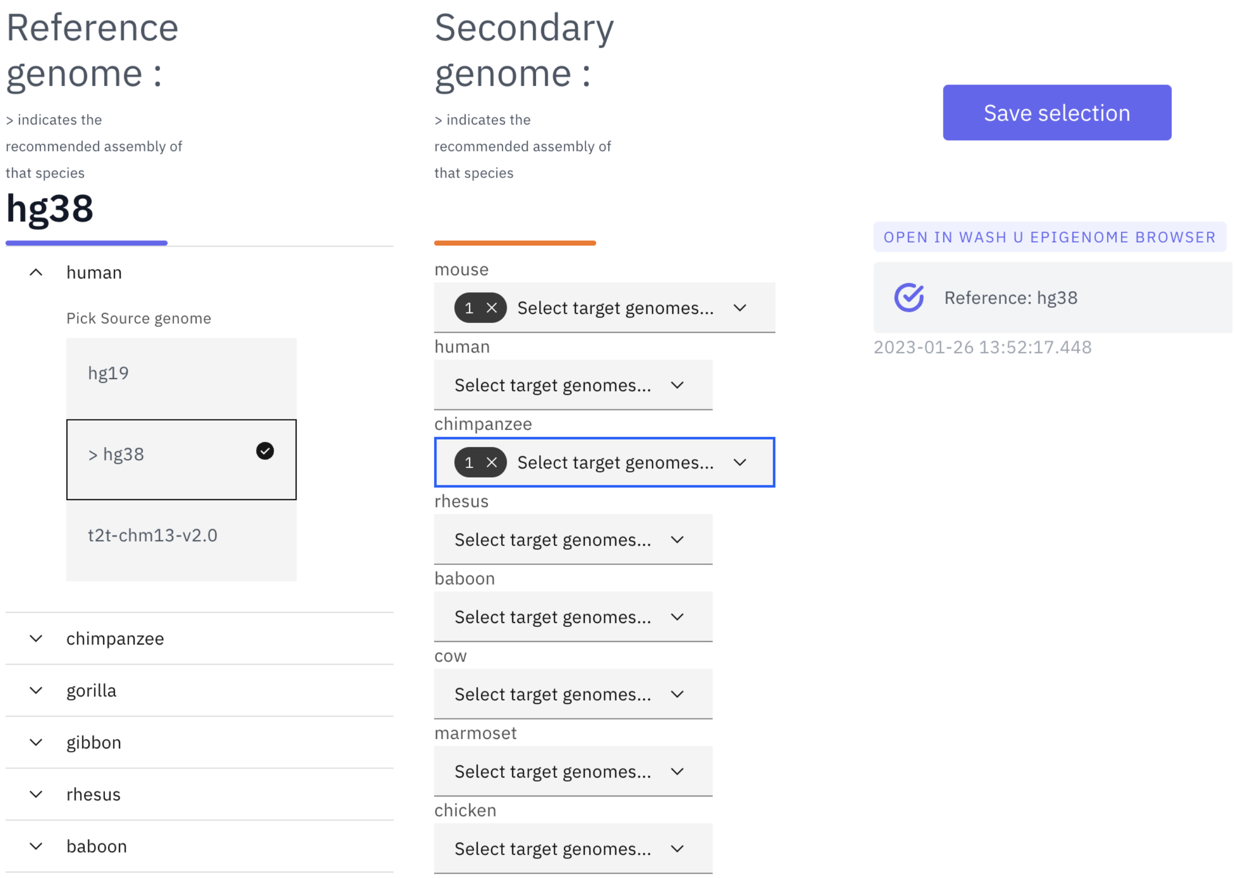
marmoset

Select target genomes...

chicken

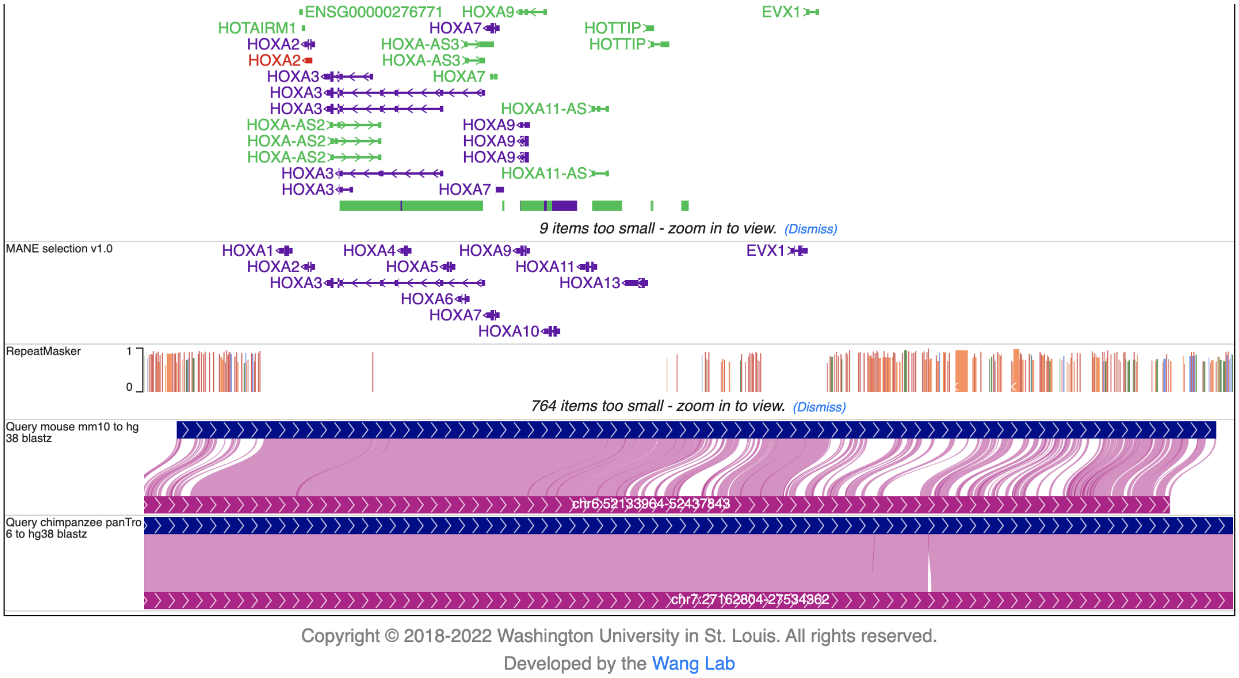
Select target genomes...

- With all the desired genomes selected, click “save selection” and a temporary datahub link will be generated. Once it is ready, click the datahub link under “OPEN IN WASHU EPIGENOME BROWSER” and a new browser view will be opened in a new tab:



16.3 Organizing tracks on the WashU Epigenome Browser

The new browser tab contains basic annotation tracks of the reference genome and the selected genome-align tracks that connects the syntenic regions from the reference genome to the secondary genomes. In the example, hg38-mm10 and hg38-panTro6 genome-align tracks are attached to the hg38 reference genome tracks:



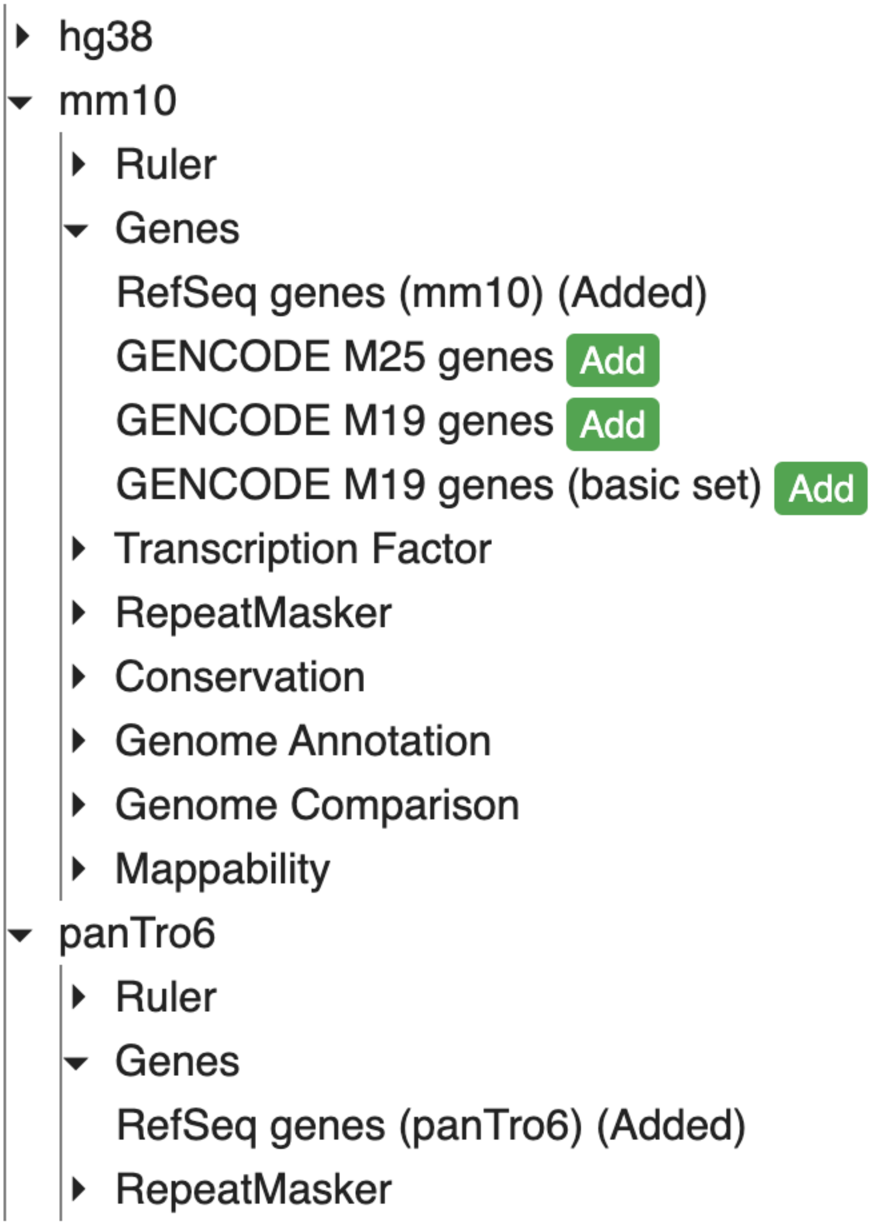
9 items too small - zoom in to view. (Dismiss)

764 items too small - zoom in to view. (Dismiss)

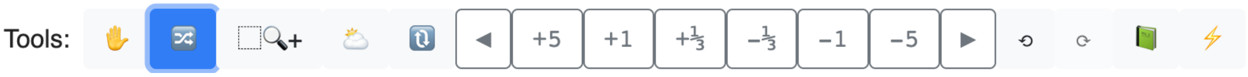
Copyright © 2018-2022 Washington University in St. Louis. All rights reserved.

Developed by the Wang Lab

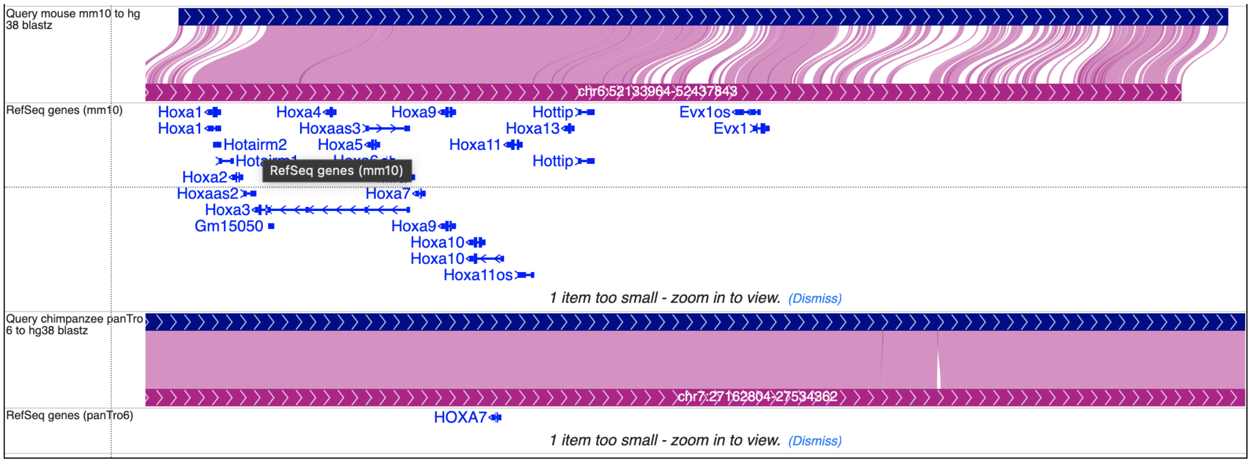
To add annotations to the secondary genomes, click “Tracks” -> “Annotation tracks” and the available annotation tracks will be listed in a dropdown menu. Click the checkbox to add a particular track to the browser view:



Here, we added Refseq gene annotations for both mm10 and panTro6, and both gene annotation tracks will be added to the bottom of the browser view. To change the order of the tracks, click the "Reorder tool" icon on the tools menu:



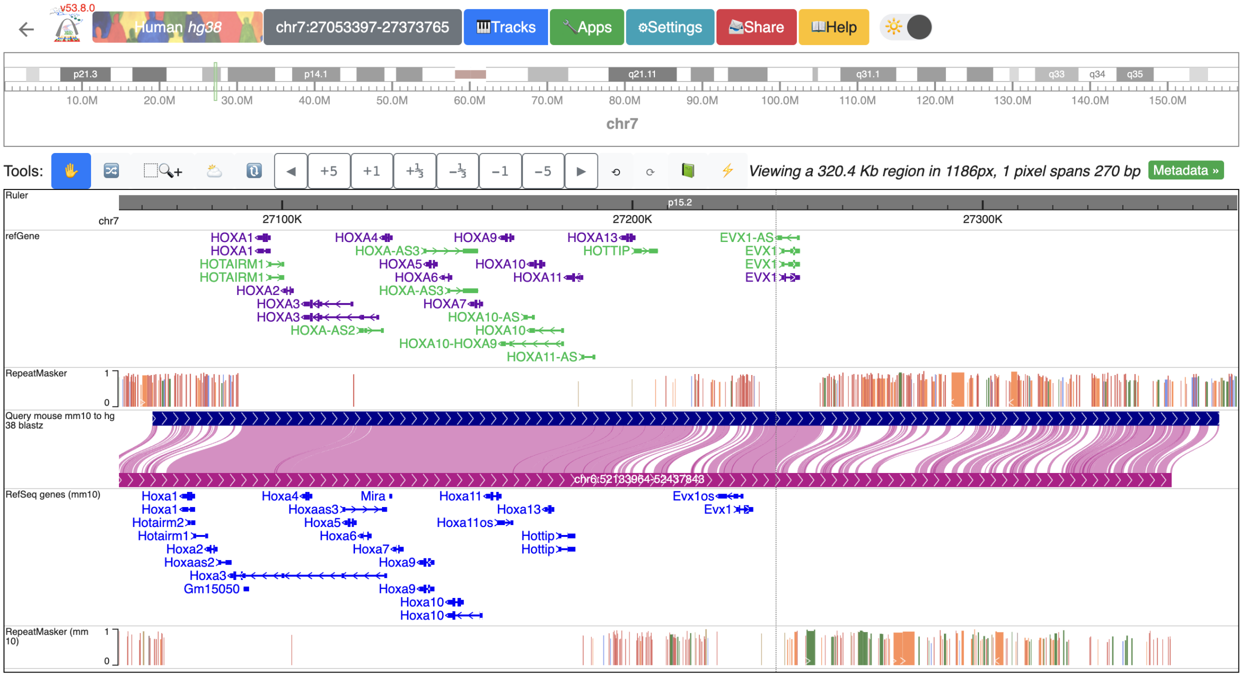
Now drag the tracks up and down to the desired position, as shown here:



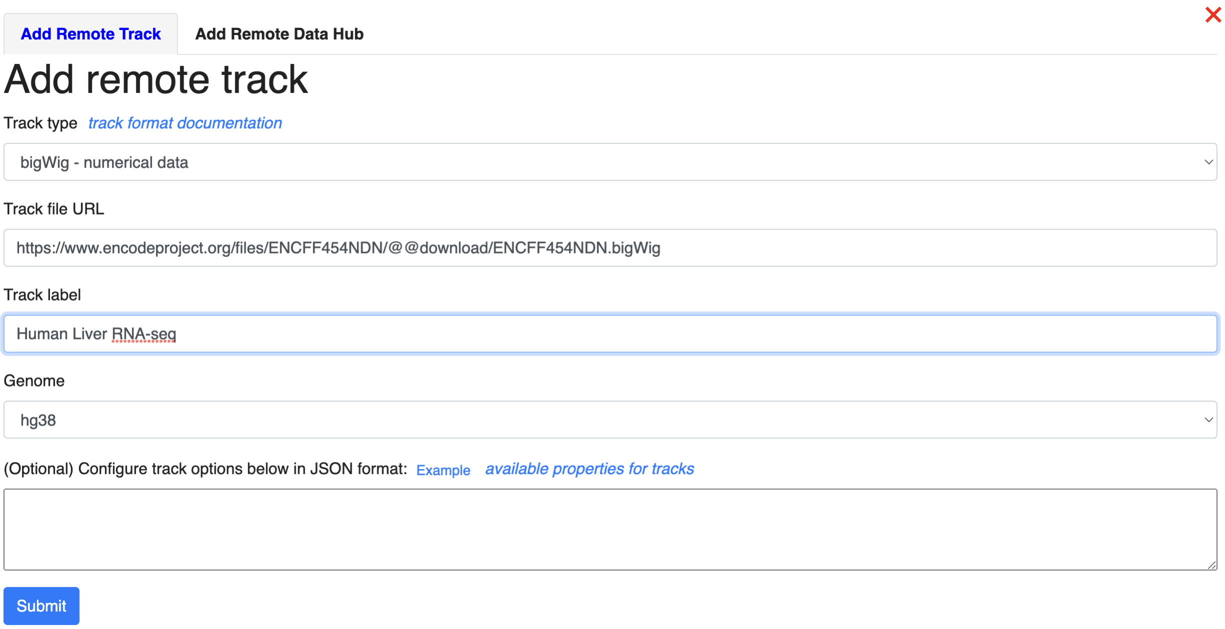
16.4 Add data tracks to the Comparative Epigenome Browser

16.4.1 Add new tracks to the Comparative Epigenome Browser

With genome-align track loaded, we can start loading additional data tracks onto it and perform analysis. Here, we have human genome hg38 as the reference genome, mouse mm10 as the secondary genome. We also have gene and repeat annotations mapped to both genomes:



Now, we can start loading our data. We can load our data from the local file system or from a URL. Let's start by loading our data from the URL. Click Tracks -> remote Tracks: We are using a human liver RNA-seq data from ENCODE (https://www.encodeproject.org/files/ENCFF861FSP/@@download/ENCFF861FSP.bigWig) in the demo. We will specify the "Track type" as "bigWig", enter the URL in the "Track file URL" entry, name the track using "Track label" entry, and select "hg38" as the assembly to map to.



Add remote track

Track type *track format documentation*

bigWig - numerical data

Track file URL

https://www.encodeproject.org/files/ENCFF454NDN/@@download/ENCFF454NDN.bigWig

Track label

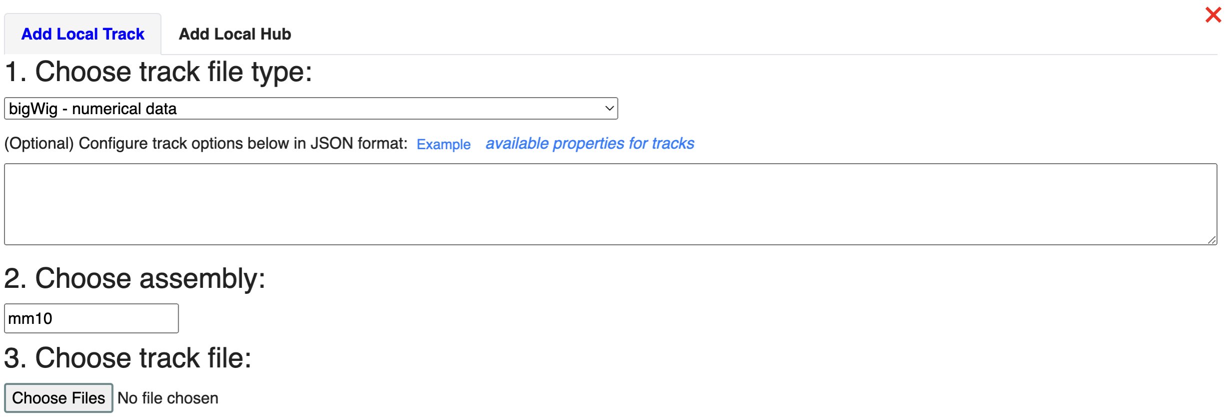
Human Liver RNA-seq

Genome

hg38

(Optional) Configure track options below in JSON format: Example *available properties for tracks*

Next, let's load our data from the local file system. Click Tracks -> Local Tracks: The track file is downloaded from ENCODE (https://www.encodeproject.org/files/ENCFF798FMB/@@download/ENCFF798FMB.bigWig) and renamed MouseLiverRNA-seq.bigWig. We will choose "bigWig" as the track file type, and choose "mm10" as the assembly it will map to. Click "Choose Files" to select the file.



1. Choose track file type:

bigWig - numerical data

(Optional) Configure track options below in JSON format: Example *available properties for tracks*

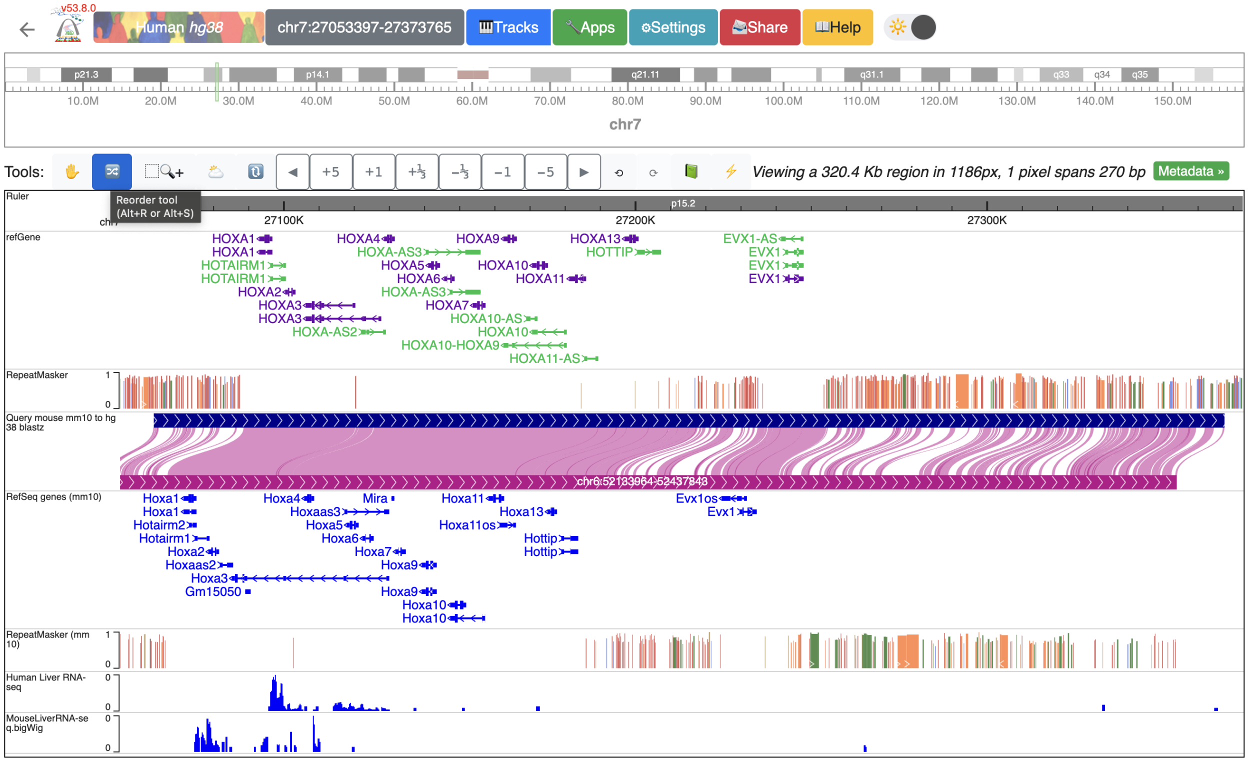
2. Choose assembly:

mm10

3. Choose track file:

No file chosen

Now, we have liver RNA-seq data from both human and mouse, mapped to hg38 and mm10 respectively, loaded at the bottom of the window ready to compare:

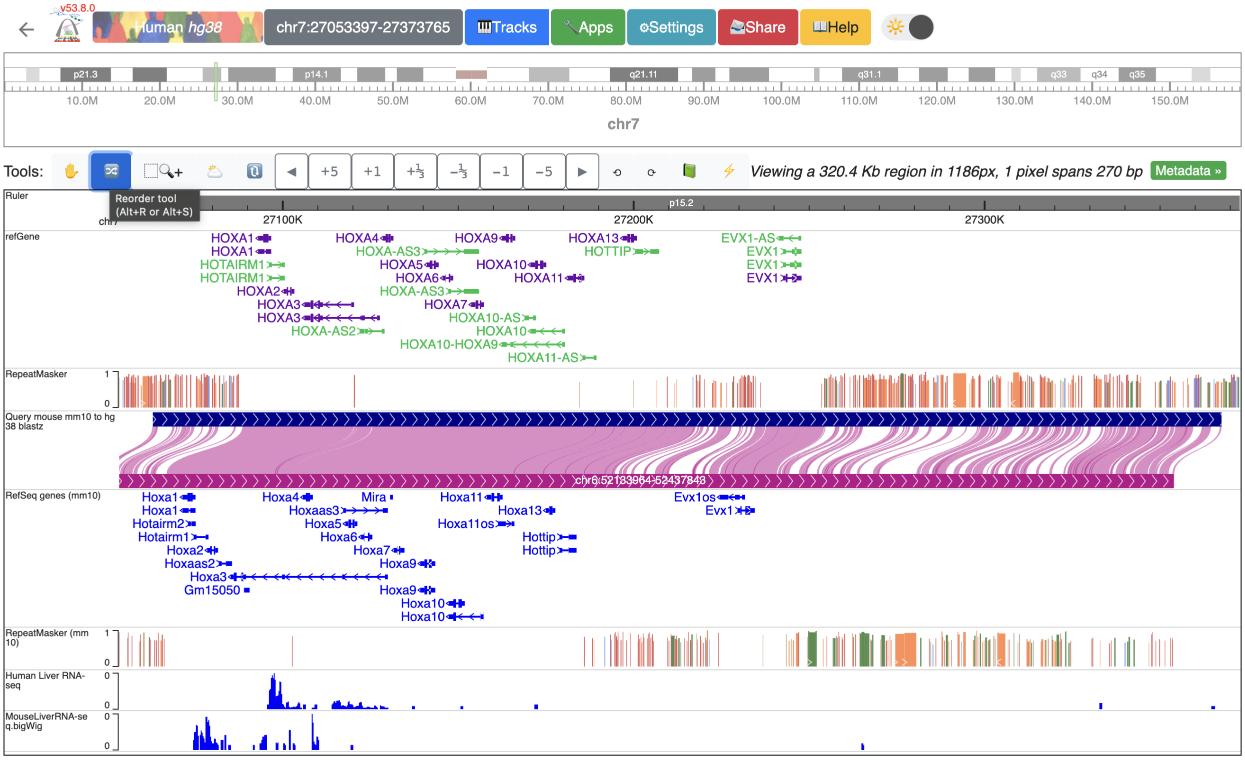


Viewing a 320.4 Kb region in 1186px, 1 pixel spans 270 bp

Reorder tool (Alt+R or Alt+S)

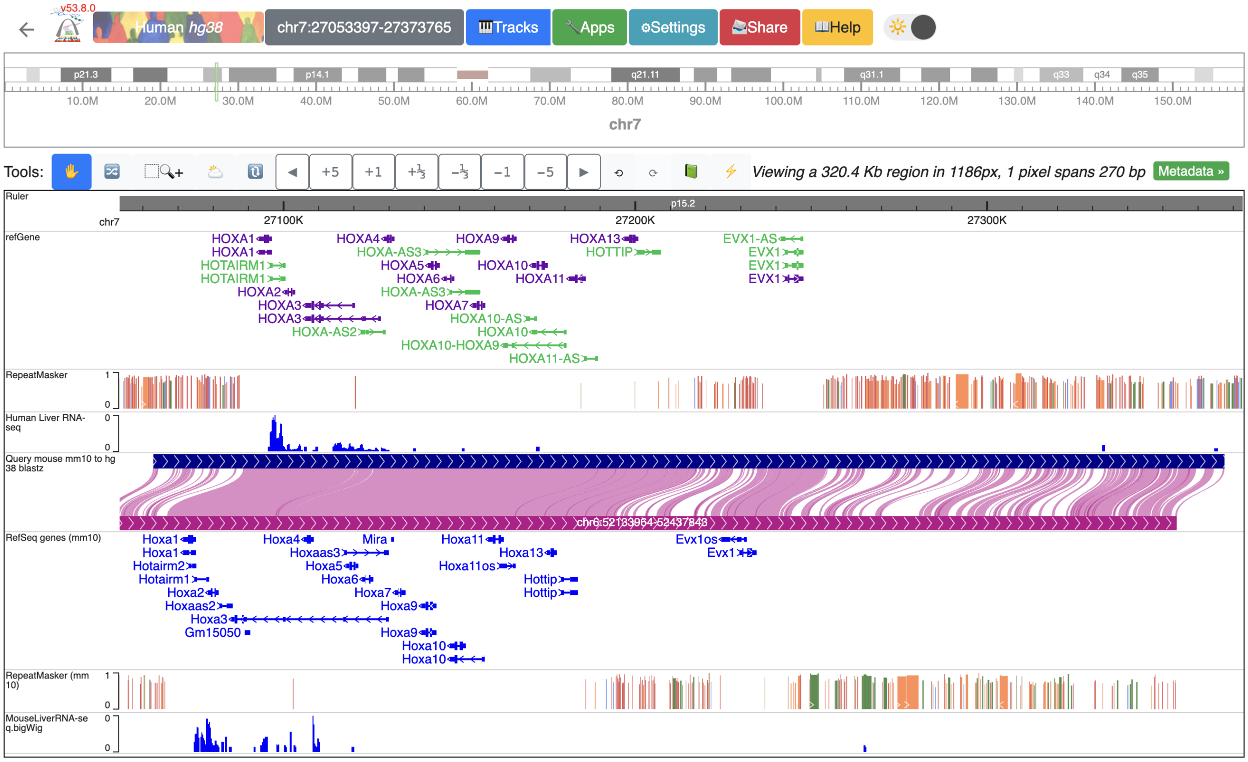
16.4.2 Organizing all the tracks in the browser

To better compare the data, we can reorder the tracks. Here, we will group the tracks by species and have them separated by the genome-align track. Click the "Reorder tool", and drag the "Human liver RNA-seq" track above the genome-align track:



Viewing a 320.4 Kb region in 1186px, 1 pixel spans 270 bp

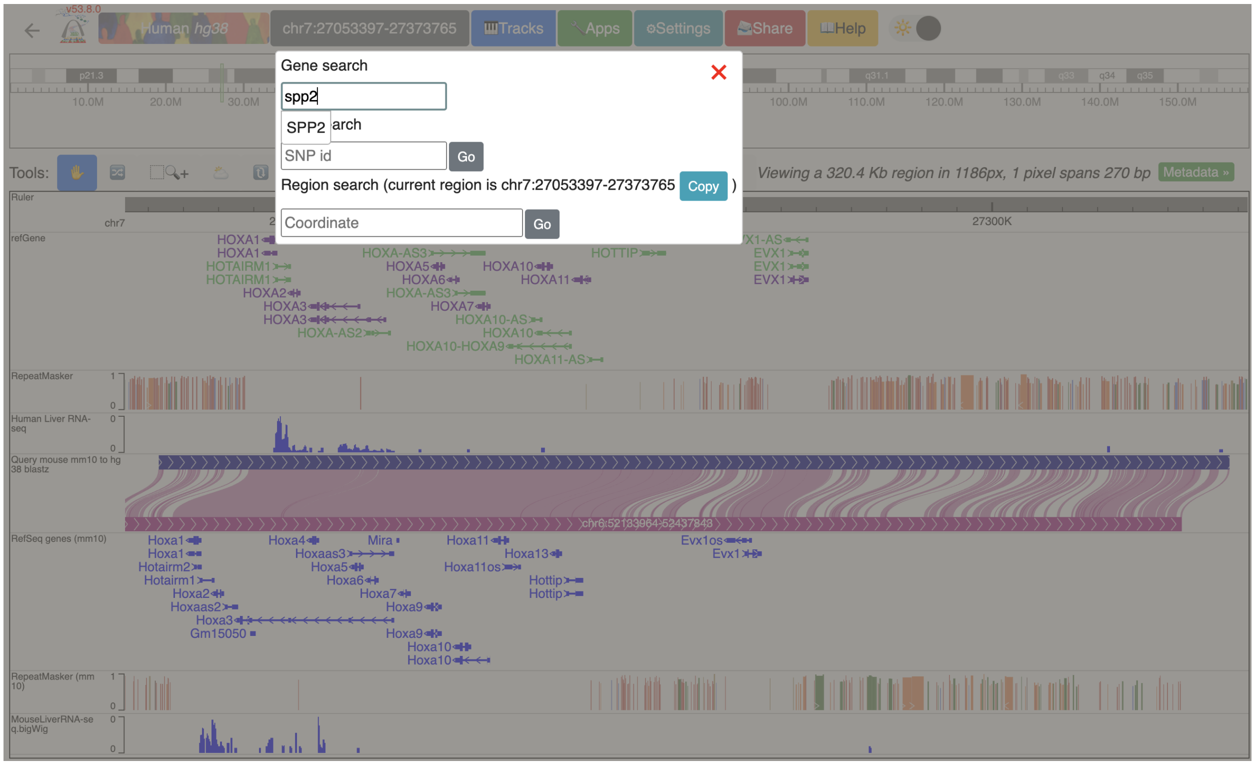
After reordering, the human track is between the human repeatMasker track and the hg38-mm10 genome-align track:



Viewing a 320.4 Kb region in 1186px, 1 pixel spans 270 bp

16.4.3 Navigation in the browser

The browser allows navigation in the reference genome using either gene name, SNP, or coordinates directly. Click the coordinates box at the top to enter the navigation window. Let's navigate to the gene "SPP2":



Gene search

spp2

SPP2 arch

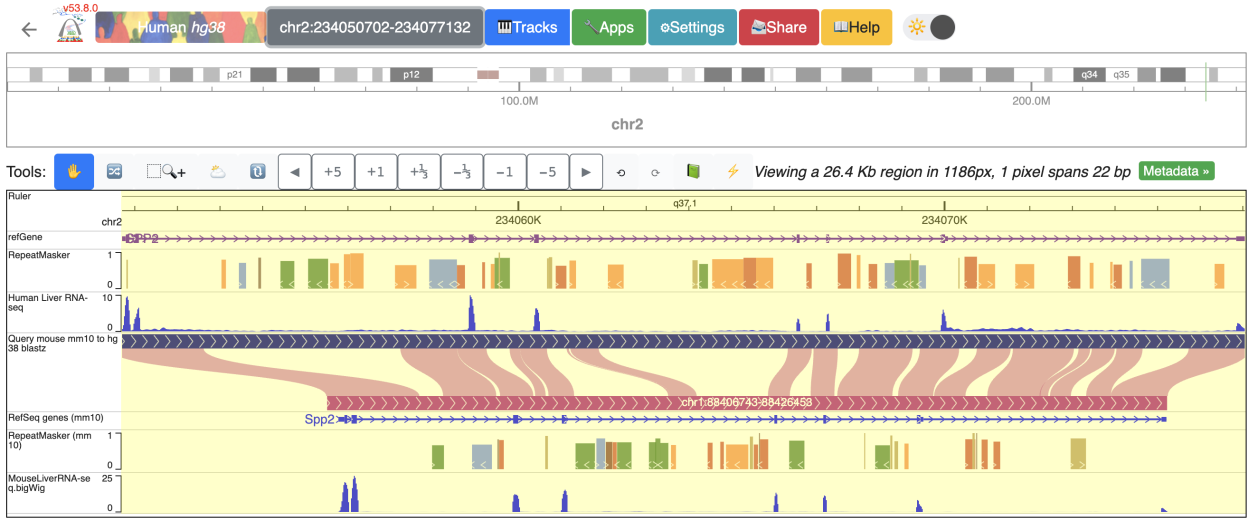
SNP id Go

Region search (current region is chr7:27053397-27373765 Copy)

Coordinate Go

Viewing a 320.4 Kb region in 1186px, 1 pixel spans 270 bp

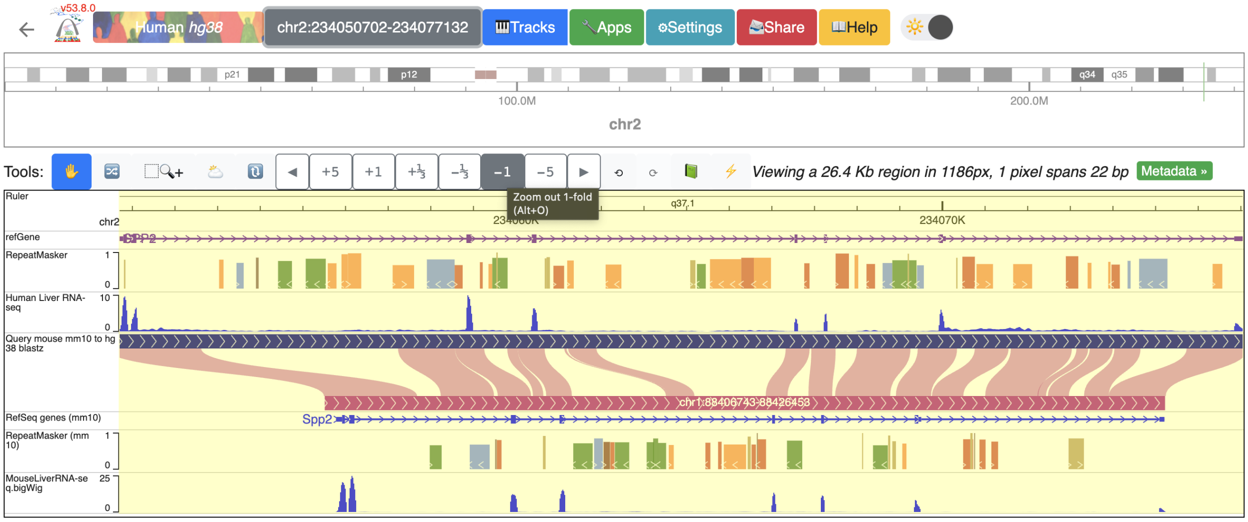
The browser window should display the entirety of SPP2 gene spanning the whole width of the browser window now:



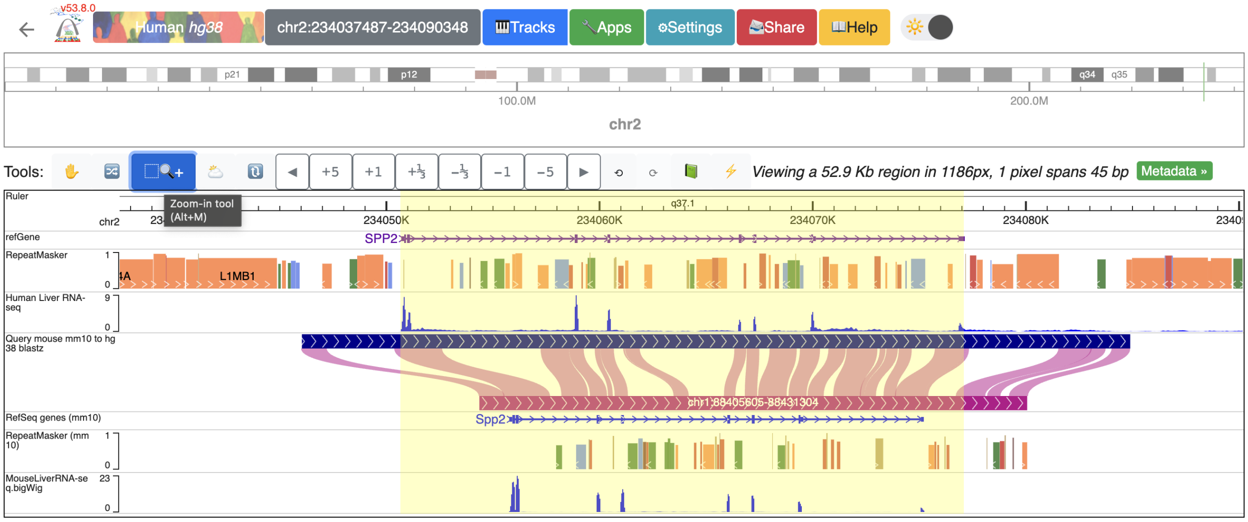
Viewing a 26.4 Kb region in 1186px, 1 pixel spans 22 bp

16.4.4 Using tools to zoom in and out

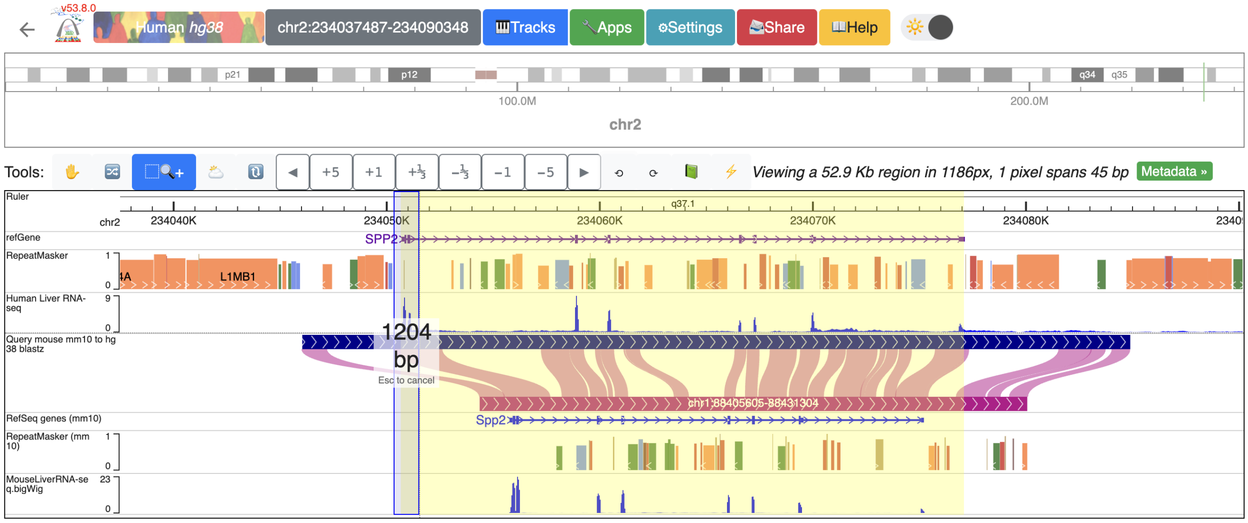
We built a "tools" bar at the top of the browser window to allow users to perform some basic operations within the browser. There are different buttons to zoom in or out with different resolutions or pan left/right. For example, to zoom out one time, click the "-1" button:



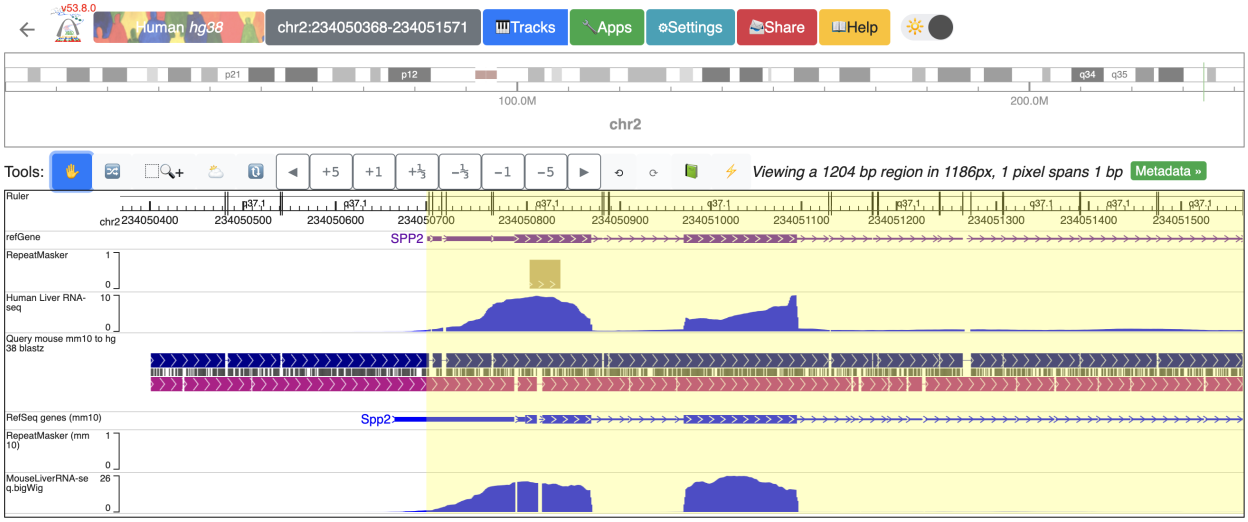
It is possible to zoom into a selected region using the "Zoom-in tool". Click the "Zoom-in tool", then click and drag over the region you want to zoom to:



To zoom into the SPP2 gene's promoter region, click and drag over the regions that covers the promoter and the first extron of SPP2:



Now, the browser displays the comparison between human SPP2 gene's promoter region with the orthologous Spp2 gene promoter in mouse, with gene annotation, repeat annotation and liver RNA-seq data tracks from both species mapped to the hg38 and mm10, respectively:

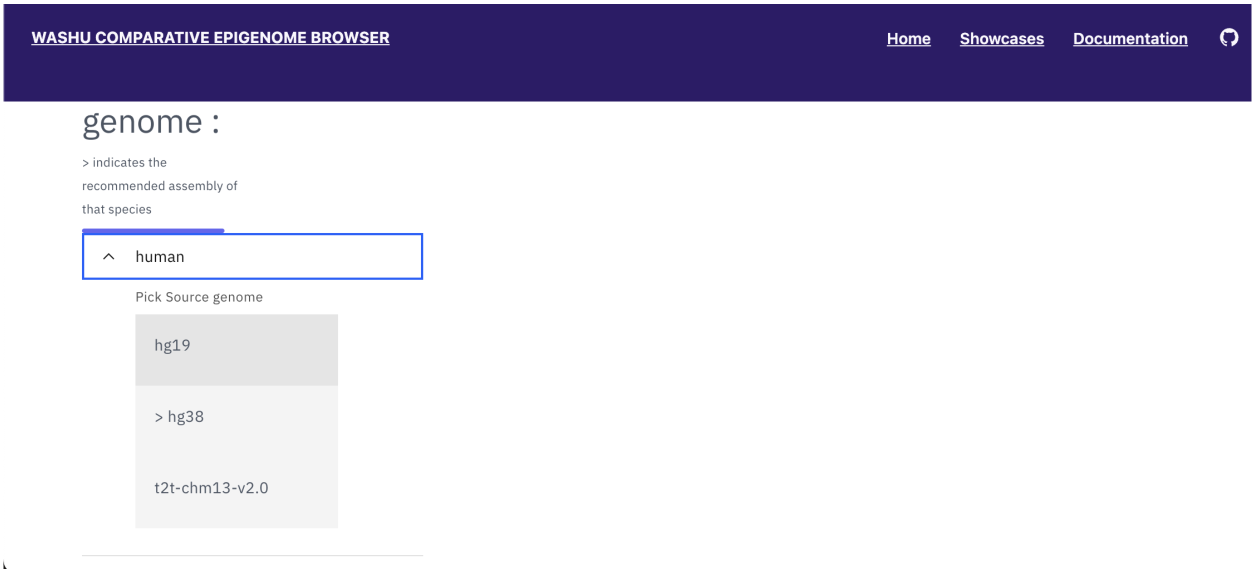


16.5 Example: create a human-mouse multiple tracks comparison view using the Comparative Epigenome Browser

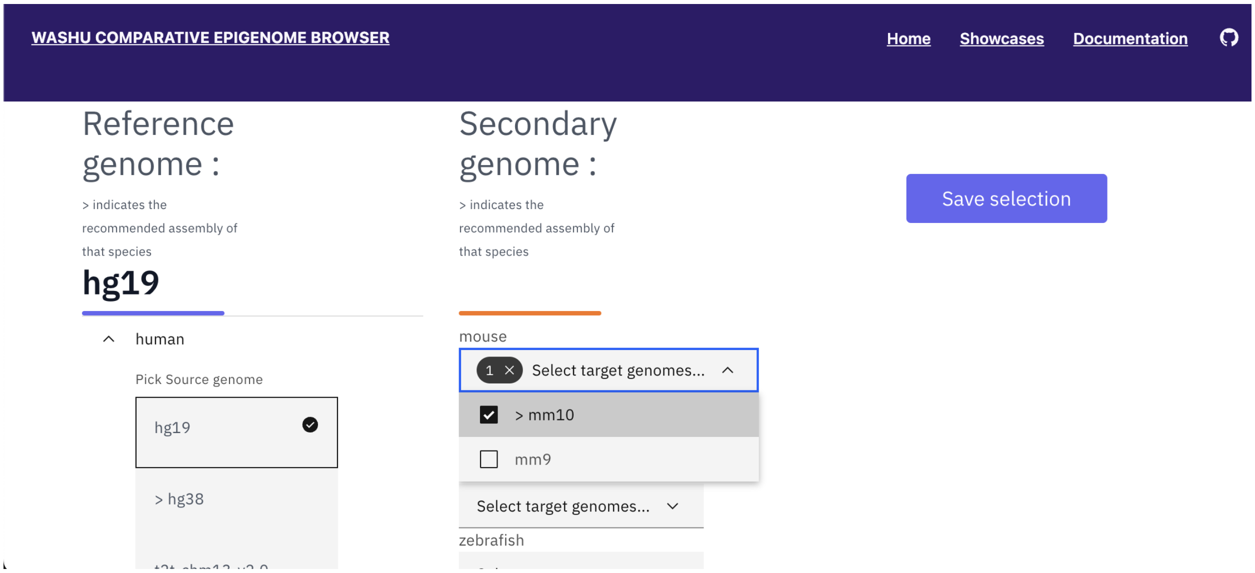
Here we will create a human-mouse multiple tracks comparison view using the Comparative Epigenome Browser. We will use remote tracks to add the following data tracks to the browser and recreate the browser view for Figure 3b from the paper (https://genome.cshlp.org/content/33/5/824):

16.5.1 Select assemblies and annotations

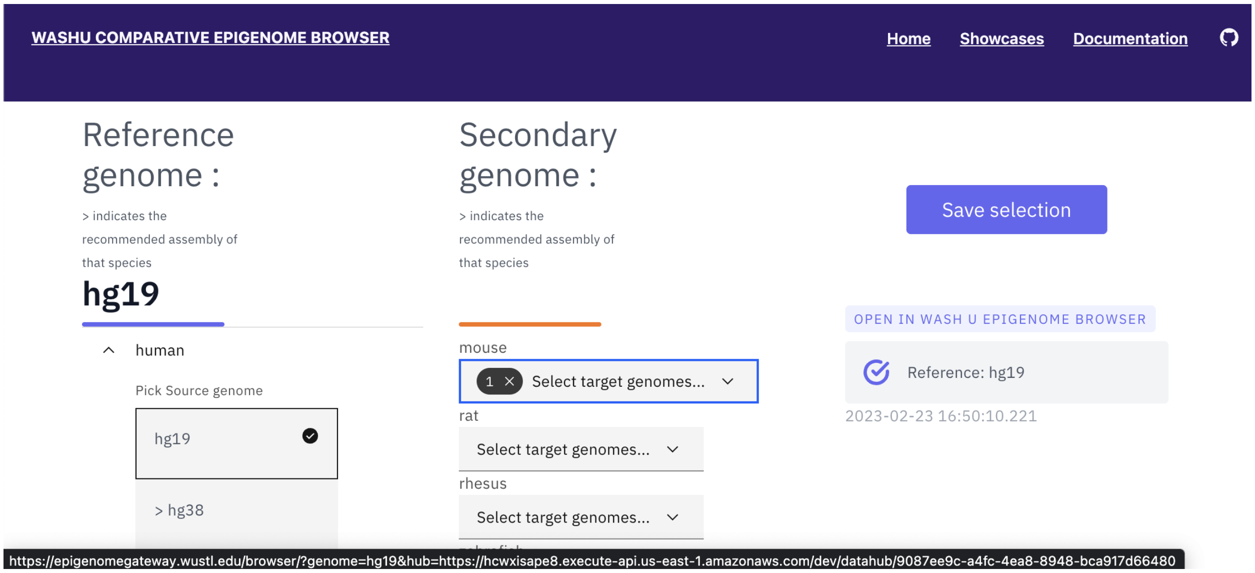
Click "select genomes", the species selection tool will become available for users to choose a reference genome. Select human, hg19.


WASHU COMPARATIVE EPIGENOME BROWSER
Home
Showcases
Documentation
genome :
> indicates the recommended assembly of that species
human
Pick Source genome
hg19
> hg38
t2t-chm13-v2.0

With hg19 selected as the reference genome, available secondary genomes will be available. Select mouse, mm10.

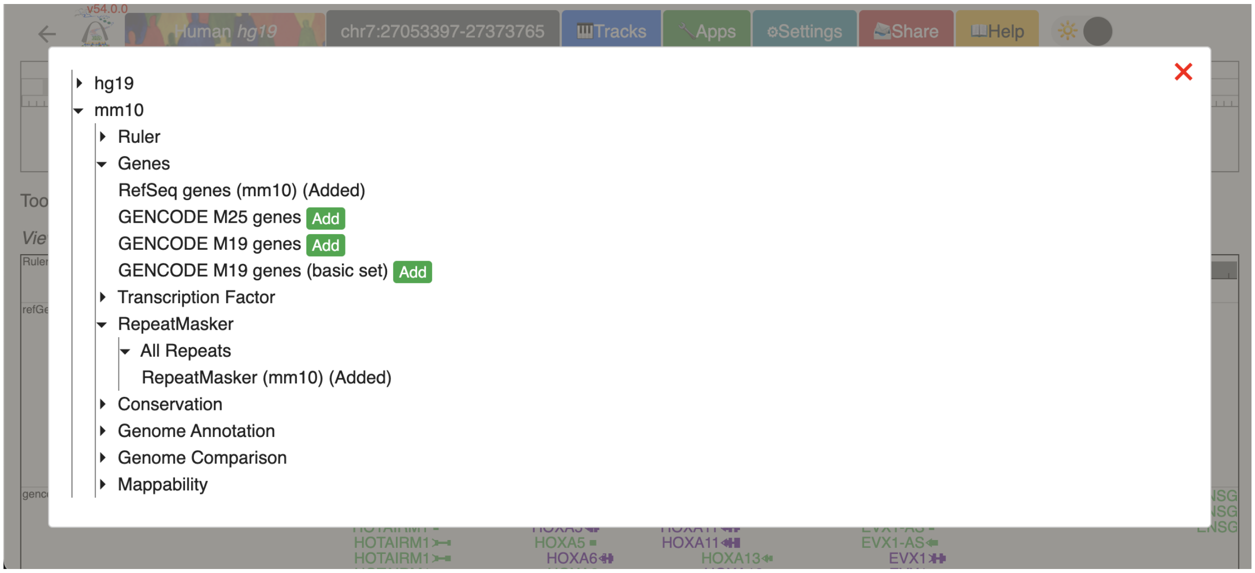

WASHU COMPARATIVE EPIGENOME BROWSER
Home
Showcases
Documentation
Reference genome :
> indicates the recommended assembly of that species
hg19
human
Pick Source genome
hg19
> hg38
Secondary genome :
> indicates the recommended assembly of that species
mouse
1 × Select target genomes...
☑ > mm10
☐ mm9
Select target genomes...
zebrafish
Save selection

Click "Save selection", and click the datahub link under "OPEN IN WASHU EPIGENOME BROWSER" to open a the browser window in a new browser tab.



https://epigenomegateway.wustl.edu/browser/?genome=hg19&hub=https://hcwxisape8.execute-api.us-east-1.amazonaws.com/dev/datahub/9087ee9c-a4fc-4ea8-8948-bca917d66480

Click "Tracks" -> "Annotation Tracks", and add "mm10":"Genes":"RefSeq genes" and "mm10":"RepeatMasker":"All Repeats":"RepeatMasker" tracks to the browser window.



16.5.2 Add Epigenomic data tracks to the comparative epigenome browser window.

We collected the following epigenomic dataset from ENCODE and roadmap projects:

human liver H3K4me3 ChIP-seq bigwig file mapped on hg19: https://egg.wustl.edu/d/hg19/GSM621675.bigWig

human liver H3K27ac ChIP-seq bigwig file mapped on hg19: https://egg.wustl.edu/d/hg19/GSM1112809_1.bigWig

human liver WGBS methylC track file mapped on hg19: https://remc.wustl.edu/dli/WGBS/E066.methylc2.gz

human liver RNA-seq bigwig file mapped on hg19: https://www.encodeproject.org/files/ENCFF975NSG/@@download/ENCFF975NSG.bigWig

human brain H3K4me3 ChIP-seq bigwig file mapped on hg19: https://egg.wustl.edu/d/hg19/GSM773012.bigWig

human brain H3K27ac ChIP-seq bigwig file mapped on hg19: https://egg.wustl.edu/d/hg19/GSM773015.bigWig human brain WGBS methylC track file mapped on hg19: https://remc.wustl.edu/dli/WGBS/E071.methylc2.gz

human brain RNA-seq bigwig file mapped on hg19: https://www.encodeproject.org/files/ENCFF386BQW/@@download/ENCFF386BQW.bigWig

mouse liver H3K4me3 ChIP-seq bigwig file mapped on mm10: https://epgg-test.wustl.edu/d/mm10/ENCFF072QFI.bigWig

mouse liver H3K27ac ChIP-seq bigwig file mapped on mm10: https://epgg-test.wustl.edu/d/mm10/ENCFF041ONG.bigWig

mouse liver WGBS methylC track file mapped on mm10: https://vizhub.wustl.edu/public/comparativeBrowser/tracks/mouseAdultLiver.sort.methylC.gz

mouse liver RNA-seq bigwig file mapped on mm10: https://epgg-test.wustl.edu/d/mm10/ENCFF697PQZ.bigWig

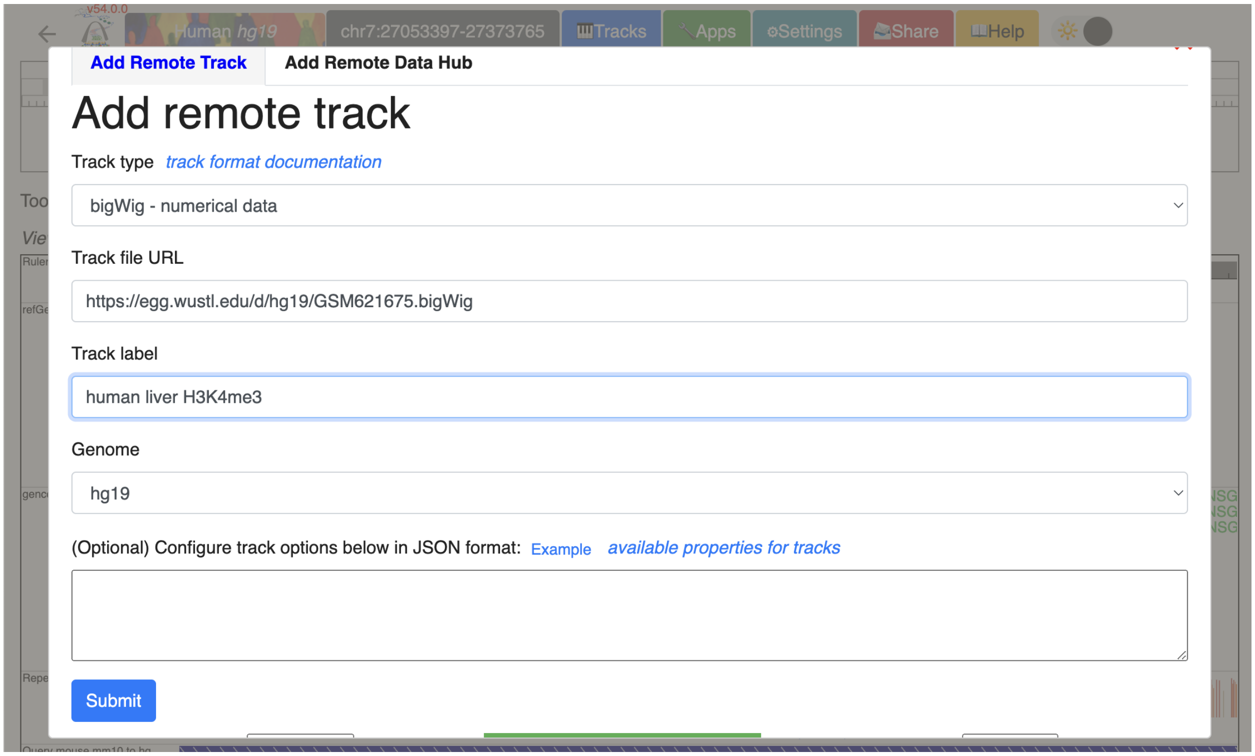
mouse brain H3K4me3 ChIP-seq bigwig file mapped on mm10: https://epgg-test.wustl.edu/d/mm10/ENCFF389PES.bigWig

mouse brain H3K27ac ChIP-seq bigwig file mapped on mm10: https://epgg-test.wustl.edu/d/mm10/ENCFF269ZNW.bigWig

mouse brain WGBS methylC track file mapped on mm10: https://vizhub.wustl.edu/public/comparativeBrowser/tracks/mouseForebrain.sort.methylC.gz

mouse brain RNA-seq bigwig file mapped on mm10: https://epgg-test.wustl.edu/d/mm10/ENCFF368ACN.bigWig

Use the "Tracks" -> "Remote tracks" function to add them one by one to the Browser window. Using human liver H3K4me3 ChIP-seq bigwig file as an example:



Add remote track

Track type *track format documentation*

bigWig - numerical data

Track file URL

https://egg.wustl.edu/d/hg19/GSM621675.bigWig

Track label

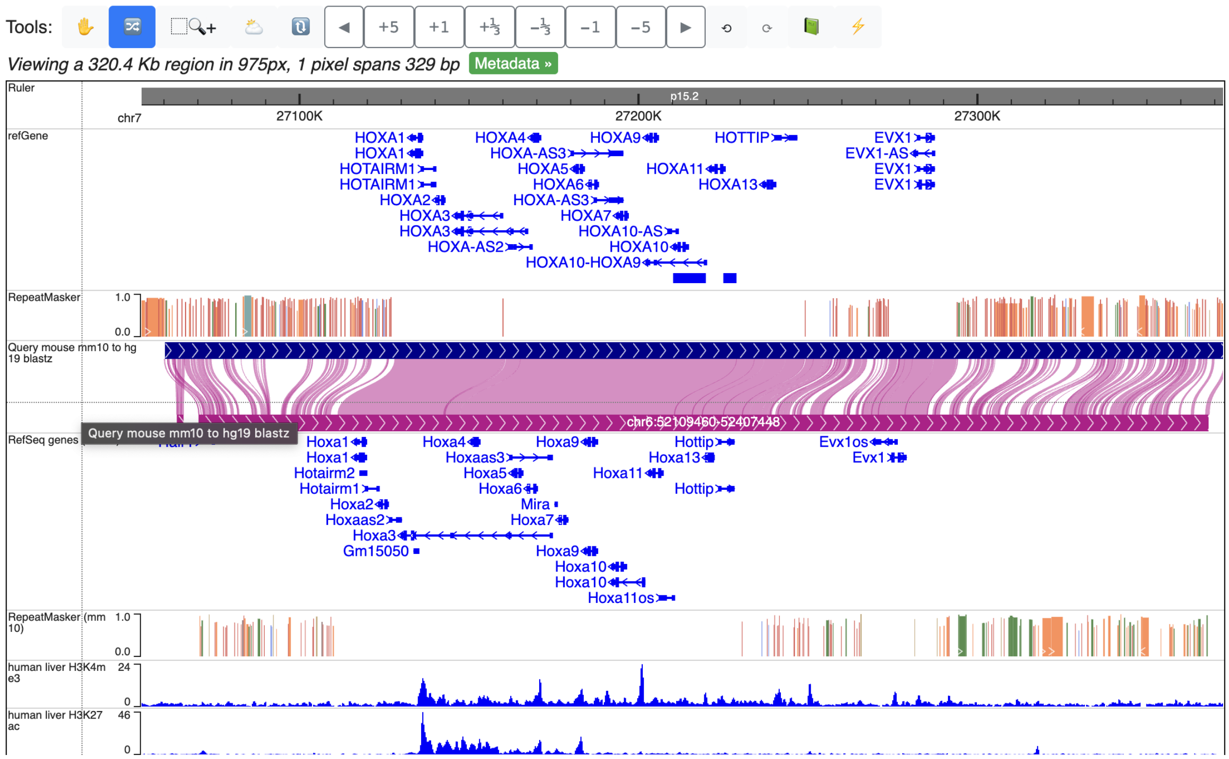
human liver H3K4me3

Genome

hg19

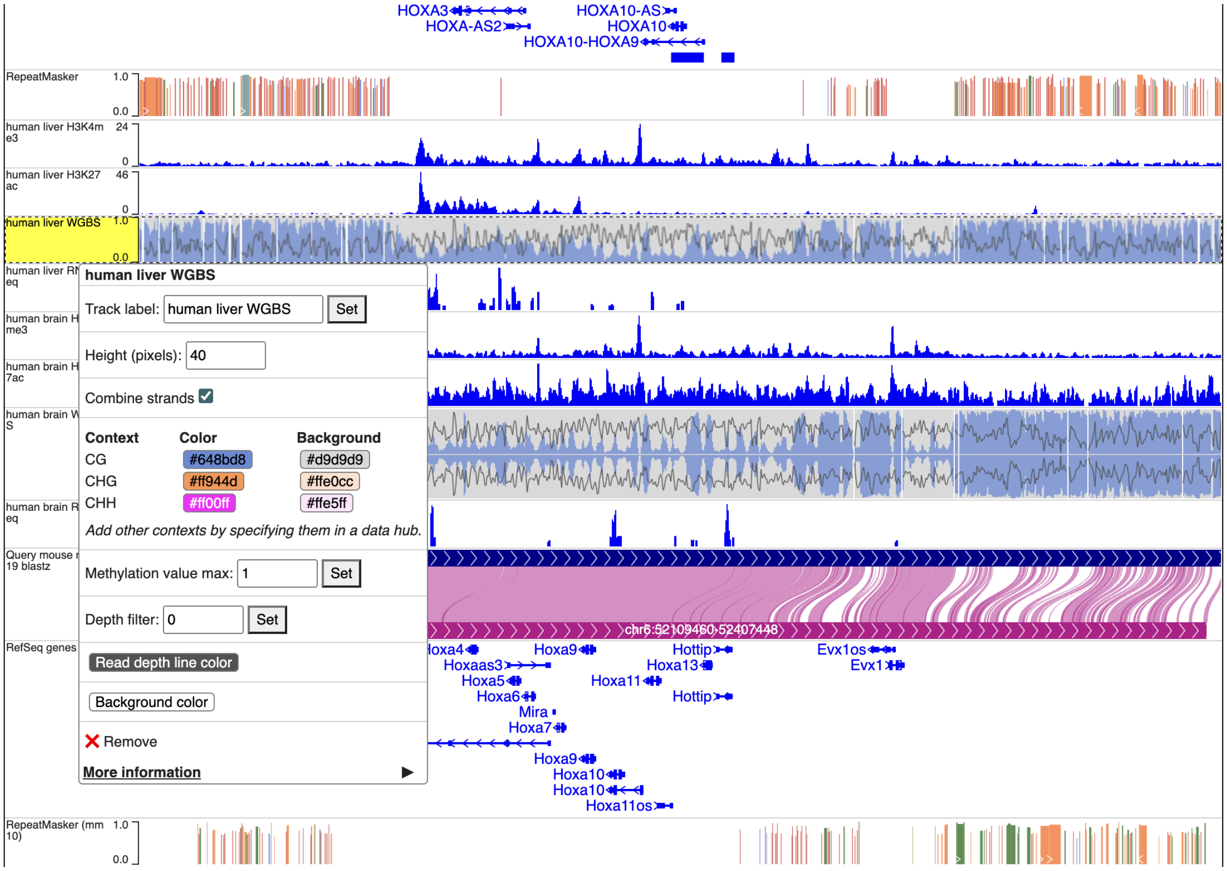
(Optional) Configure track options below in JSON format: Example *available properties for tracks*

Repeat the process to load all the tracks from the list above. With All tracks added, Click "Reorder tool" in the tools bar, and drag tracks up and down to order all the tracks by genomes and tissue.



Viewing a 320.4 Kb region in 975px, 1 pixel spans 329 bp

If only CpG methylation were characterized, we can also check "Combine strands" to merge both strands in all the methylC tracks.



human liver WGBS

Track label: human liver WGBS Set

Height (pixels): 40

Combine strands ☑

Context	Color	Background
CG	#648bd8	#d9d9d9
CHG	#ff944d	#ffe0cc
CHH	#ff00ff	#ffe5ff

Add other contexts by specifying them in a data hub.

Methylation value max: 1 Set

Depth filter: 0 Set

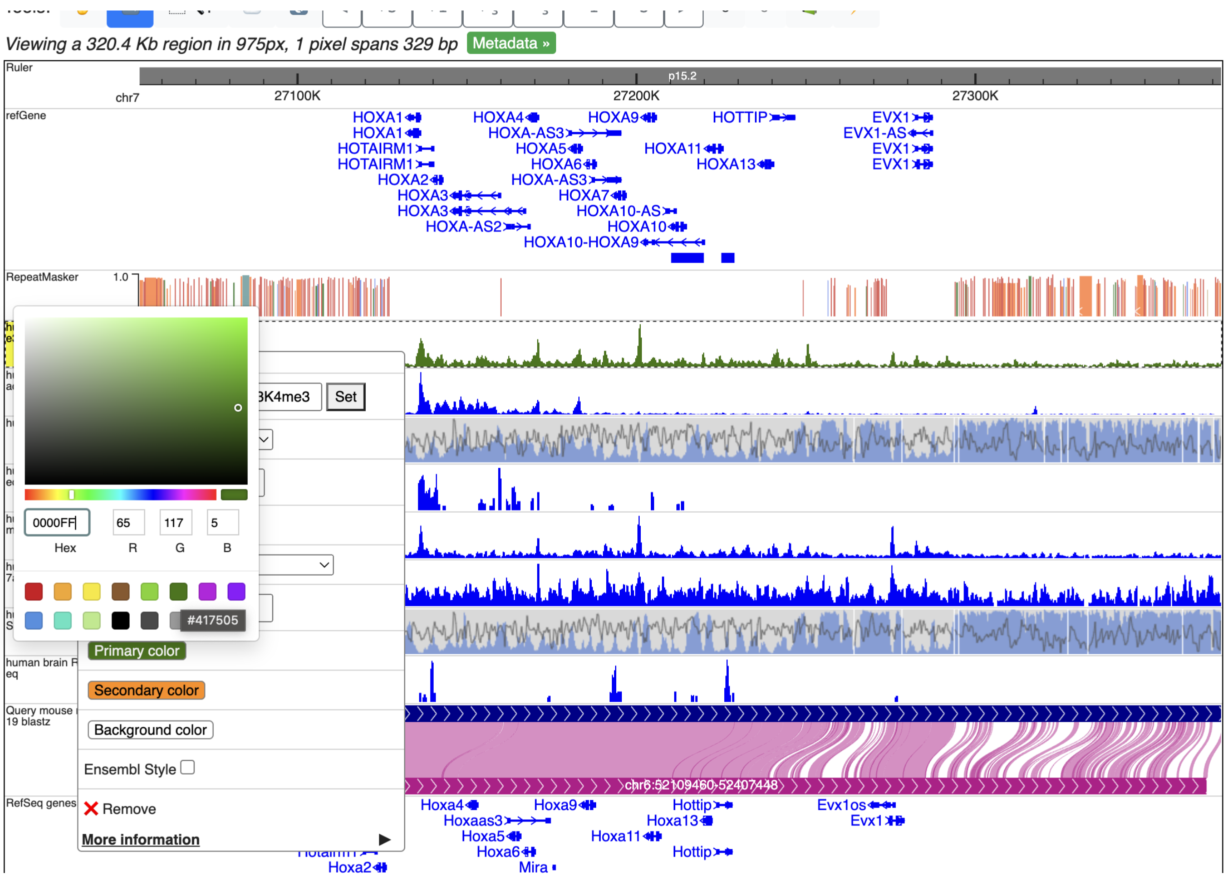
Read depth line color

Background color

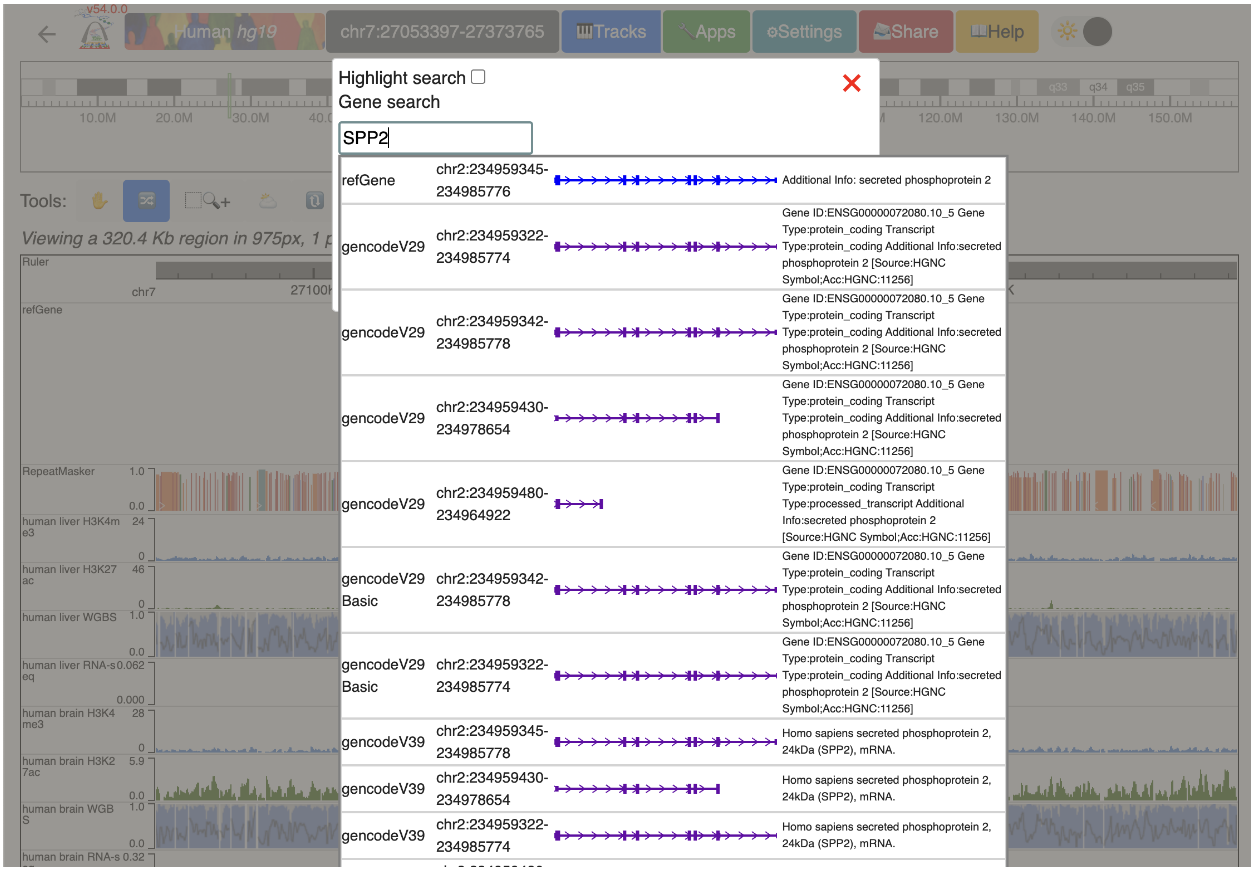
✕ Remove

More information ▶

We can then change the peak display color by right click each track, and change the primary color.



Click genome coordinates on the top and navigate to any gene or coordinates. Here we are navigating to gene SPP2.



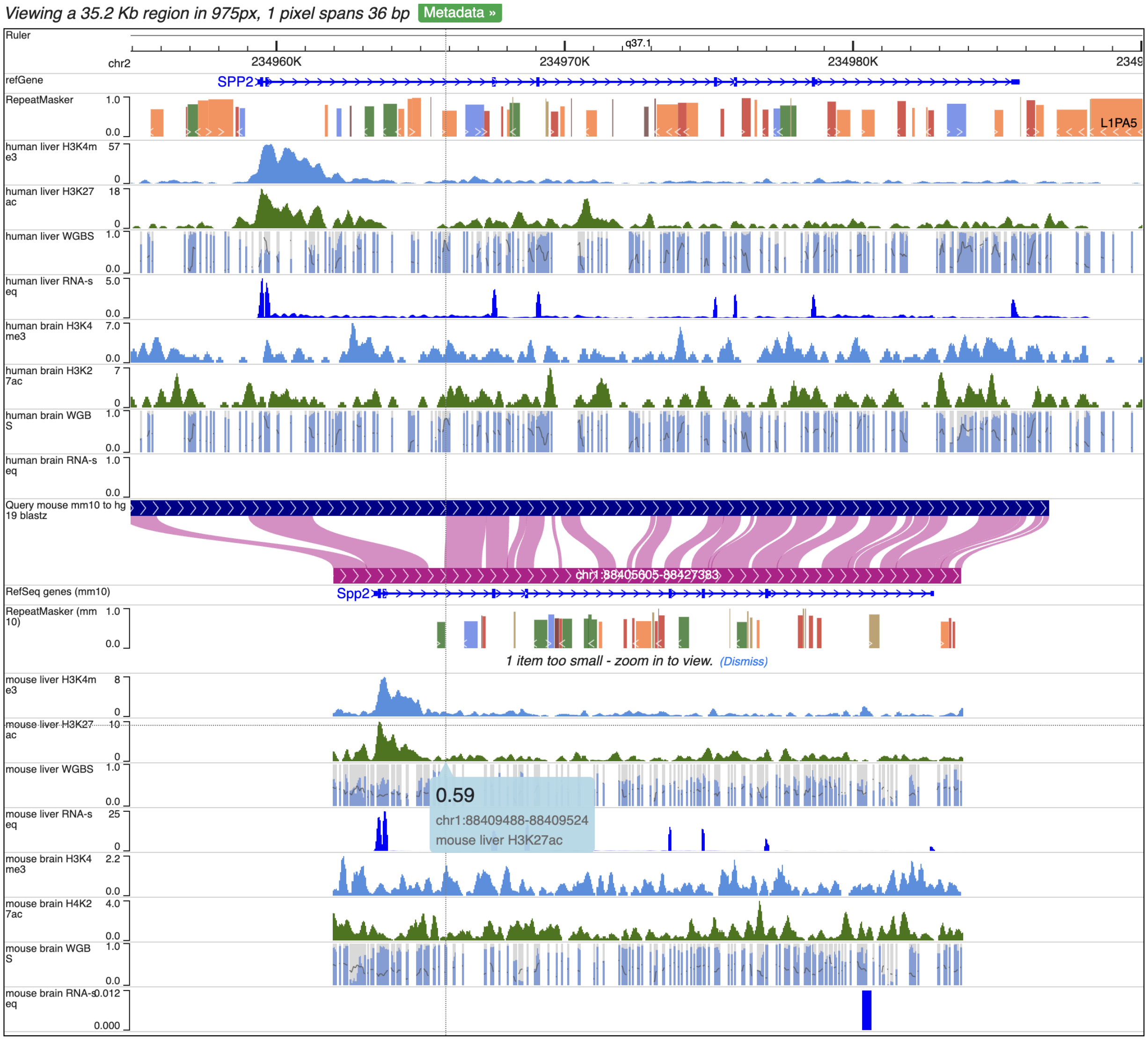
Highlight search ☐

Gene search

SPP2

refGene	chr2:234959345-234985776		Additional Info: secreted phosphoprotein 2
gencodeV29	chr2:234959322-234985774		Gene ID:ENSG00000072080.10_5 Gene Type:protein_coding Transcript Type:protein_coding Additional Info:secreted phosphoprotein 2 [Source:HGNC Symbol;Acc:HGNC:11256]
gencodeV29	chr2:234959342-234985778		Gene ID:ENSG00000072080.10_5 Gene Type:protein_coding Transcript Type:protein_coding Additional Info:secreted phosphoprotein 2 [Source:HGNC Symbol;Acc:HGNC:11256]
gencodeV29	chr2:234959430-234978654		Gene ID:ENSG00000072080.10_5 Gene Type:protein_coding Transcript Type:protein_coding Additional Info:secreted phosphoprotein 2 [Source:HGNC Symbol;Acc:HGNC:11256]
gencodeV29	chr2:234959480-234964922		Gene ID:ENSG00000072080.10_5 Gene Type:protein_coding Transcript Type:processed_transcript Additional Info:secreted phosphoprotein 2 [Source:HGNC Symbol;Acc:HGNC:11256]
gencodeV29 Basic	chr2:234959342-234985778		Gene ID:ENSG00000072080.10_5 Gene Type:protein_coding Transcript Type:protein_coding Additional Info:secreted phosphoprotein 2 [Source:HGNC Symbol;Acc:HGNC:11256]
gencodeV29 Basic	chr2:234959322-234985774		Gene ID:ENSG00000072080.10_5 Gene Type:protein_coding Transcript Type:protein_coding Additional Info:secreted phosphoprotein 2 [Source:HGNC Symbol;Acc:HGNC:11256]
gencodeV39	chr2:234959345-234985778		Homo sapiens secreted phosphoprotein 2, 24kDa (SPP2), mRNA.
gencodeV39	chr2:234959430-234978654		Homo sapiens secreted phosphoprotein 2, 24kDa (SPP2), mRNA.
gencodeV39	chr2:234959322-234985774		Homo sapiens secreted phosphoprotein 2, 24kDa (SPP2), mRNA.

Zoom out 1/3 times, and we can see the whole SPP2 gene with all the data tracks marked by different colors.



Viewing a 35.2 Kb region in 975px, 1 pixel spans 36 bp

1 item too small - zoom in to view. (Dismiss)

0.59
chr1:88409488-88409524
mouse liver H3K27ac

CHAPTER

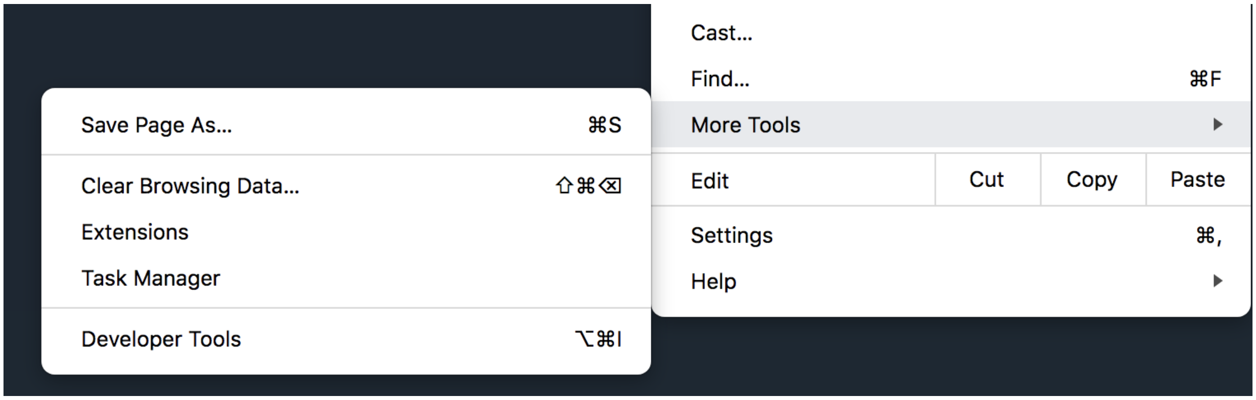
SEVENTEEN

FAQ

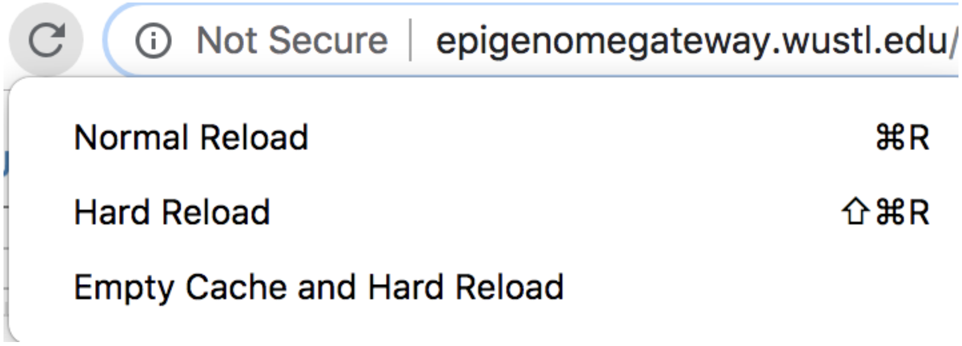
17.1 Hard Reload

Sometimes your web browser might cached old Javascript code of the browser, if you didn't see updated feature after refresh, you can do a `Hard Reload`. This is how you do this on Google Chrome:

Open *Developer Tools*:



Click and Hold the Refresh button for a while, then you can see the *Hard Reload* option:



17.2 Data fetch failed

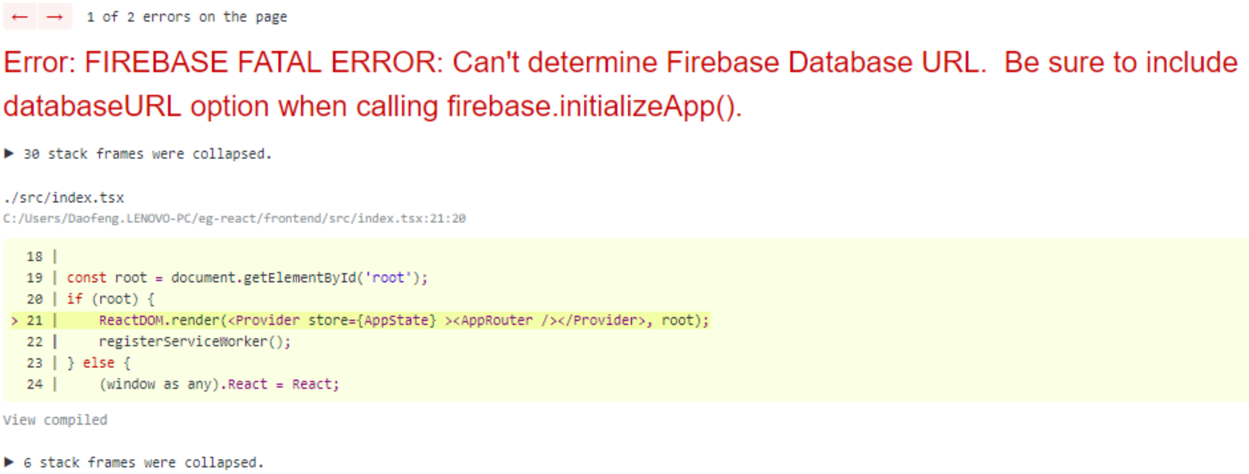
Please check the URL to your file is correct. If yes, most case, your webserver doesn't enable CORS. Please see *Tracks* page for how to enable CORS settings.

17.3 Use HTTP or HTTPS

Both our main site and AWS mirror support both HTTP and HTTPS protocol, since webpage hosted through HTTPS cannot access resource hosted by HTTP, you should use our HTTP site. For example, when you visit https://epigenomegateway.wustl.edu/browser, and you want to display a custome track hosted at http://your.track.url.bigwig, the browser will display `Data fetch failed` for that track because due to security settings, the browser in HTTPS page cannot access HTTP resource. In such case you can use http://epigenomegateway.wustl.edu/browser instead (without the `s`).

17.4 Firebase fetal error

After you installed a new mirror, when you start your mirror instance by running `npm start`, if you see a Firebase fetal error like following:



```
← →  1 of 2 errors on the page

Error: FIREBASE FATAL ERROR: Can't determine Firebase Database URL.  Be sure to include
databaseURL option when calling firebase.initializeApp().

▶ 30 stack frames were collapsed.

./src/index.tsx
C:/Users/Daofeng.LENOVO-PC/eg-react/frontend/src/index.tsx:21:20

  
  const root = document.getElementById('root');
  if (root) {
>     ReactDOM.render(<Provider store={AppState} ><AppRouter /></Provider>, root);
      registerServiceWorker();
  } else {
      (window as any).React = React;

View compiled

▶ 6 stack frames were collapsed.
```

This means you need to setup a Firebase database for the Session/Live function to work properly, check *Firebase setup* please.

17.5 Can I use without setup Firebase?

Yes. But this means you would not have the Session/Live function, check *Use without firebase* please.

17.6 Local track security

Local track function is perfect for view protected or private data, since there is no data transfer on the web. More discussions about this please check here.

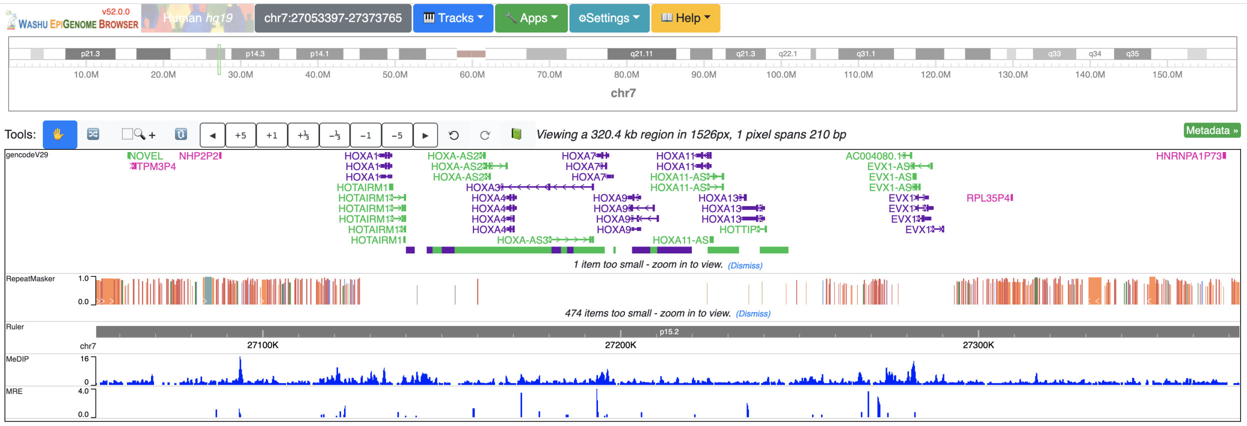
17.7 What's the different between `hub` and `sessionFile` URL parameter

Both browser links with `hub` or `sessionFile` can be shared with others or be used for publication purpose. `sessionFile` not only contains all the *tracks* as `hub`, it also contains genomic coordinates, metadata, show/hide of genome navigator etc.

So `sessionFile` link contains more information than `hub` link, as the `sessionFile` contains more contents only used by the browser (it's much more complex than `hub` file syntax).

Examples links:

hub link

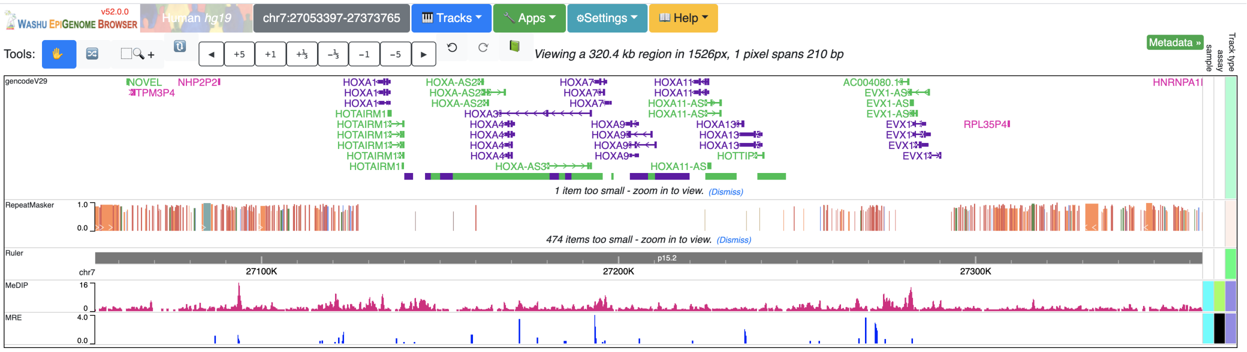


Viewing a 320.4 kb region in 1526px, 1 pixel spans 210 bp

1 item too small - zoom in to view. (Dismiss)

474 items too small - zoom in to view. (Dismiss)

sessionFile link



Viewing a 320.4 kb region in 1526px, 1 pixel spans 210 bp

1 item too small - zoom in to view. (Dismiss)

474 items too small - zoom in to view. (Dismiss)

17.8 Publish with the browser

First thank you very much for considering publishing figures, datahubs and session links using the Browser. For best result, please put all your track files on a permenant web location (like your own web server or Amazon S3), then use *hub* or *sessionFile* URL parameter for browser hub URL. Browser URL with `hub` or `sessionFile` is permenant as long as your track files from your web server stay.

Note: using `session bundle id` is not recommended as session id is suppose to be shared with trusted people, share the session Id in a public environment may result unwanted edits to your session.

CHAPTER

EIGHTEEN

CONTACT US

18.1 Source code and issue tracker

You can find source code at our github repo: https://github.com/lidaof/eg-react

If you find any issue or have any question, please submit an issue here: https://github.com/lidaof/eg-react/issues. Thank you.

Note: Using github issues is our preferred way to commmunicate with users :)

18.2 Get help on social media

You can also get help from social media:

1. Google group: https://groups.google.com/forum/#!forum/epgg
2. Facebook page: https://www.facebook.com/WashUEpiGenomeBrowser/
3. Twitter: https://twitter.com/wuepgg
4. Youtube Channel: https://www.youtube.com/@epgg

18.3 Cite the browser

If you used the browser in your research, please help us by citing the following paper(s):

(the original browser paper)

Daofeng Li, Silas Hsu, Deepak Purushotham, Renee L Sears, Ting Wang, WashU Epigenome Browser update, Nucleic Acids Research, Volume 47, Issue W1, 02 July 2019, Pages W158–W165, https://doi.org/10.1093/nar/gkz348 [PMID: 31165883].

(2022 update)

Daofeng Li, Deepak Purushotham, Jessica K Harrison, Silas Hsu, Xiaoyu Zhuo, Changxu Fan, Shane Liu, Vincent Xu, Samuel Chen, Jason Xu, Shinyi Ouyang, Angela S Wu, Ting Wang, WashU Epigenome Browser update 2022, Nucleic Acids Research, Volume 50, Issue W1, 5 July 2022, Pages W774–W781, https://doi.org/10.1093/nar/gkac238 [PMID: 35412637].

(3D browser)

Daofeng Li, Jessica K. Harrison, Deepak Purushotham & Ting Wang, Exploring genomic data coupled with 3D chromatin structures using the WashU Epigenome Browser, Nature Methods volume 19, pages 909–910 (2022) [PMID: 35864166].

Zhuo, Xiaoyu, Silas Hsu, Deepak Purushotham, Prashant K. Kuntala, Jessica K. Harrison, Alan Y. Du, Samuel Chen, Daofeng Li, and Ting Wang. Comparing Genomic and Epigenomic Features across Species Using the WashU Comparative Epigenome Browser, Genome Research, May 8, 2023, gr.277550.122. https://doi.org/10.1101/gr.277550.122.

(Comparative browser paper)

CHAPTER NINETEEN

INDICES AND TABLES

- genindex
- search