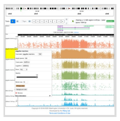


Mar 10, 2020

Visualizing Calling Card Data on the WashU Epigenome Browser

Arnav Moudgil[1], Rob Mitra[1]

[1]Washington University, Saint Louis

1 Works for me dx.doi.org/10.17504/protocols.io.bca8ishw

Transposon Calling Cards

Arnav Moudgil
Washington University, Saint Louis

ABSTRACT

This document explains how to visualize calling card insertions as wells as density and peak tracks on the WashU Epigenome Browser.

MATERIALS TEXT

If you are hosting files on a server, you will require the following software package:

- htslib

BEFORE STARTING

Please make sure you have installed the required software and packages (see Materials section).

This protocol describes how to visualize calling card data, such as raw insertions (in a CCF file), insertion densities (bedgraph file), and peaks (BED file). Ideally, you will be able to store files on a publicly-accessible webserver. This requires enabling Cross-Origin Resource Sharing (CORS). If this is not possible, data files can also be directly uploaded from your computer to the browser. This approach is described in the second half of the protocol.

Introduction

1 During the course of a calling card experiment, or after all the analysis has been said and done, you may want to visually inspect the data. These can be the raw insertions (stored as a CCF file), insertion densities (as a bedgraph file), or the peak boundaries (as a BED file). Instructions for generating these kinds of files can be found in the following protocols:

- Processing Bulk RNA Calling Card Sequencing Data
- Processing Single Cell Calling Card Sequencing Data
- Calling Peaks on piggyBac Calling Card Data

2 The WashU Epigenome Browser natively supports visualizing CCF files as a calling card track. In addition, it supports standard genomic file formats such as BED, bedgraph, bigWig, and HiC. This allows us to compare and contrast calling card data to popular genomic assays like ChIP-seq, DNase-seq, ATAC-seq, and Hi-C.

There are two ways load data onto the browser:

1. Hosting your data on a publicly-accessible server and letting the browser fetch the data.
2. Directly uploading text files to the browser.

The first method is preferred: track information and preferences can be stored in a simple text file, which can then be shared with collaborators. This saves the effort and cost of moving around potentially large datasets. Instead, the data remains in one place and can be accessed by anyone from anywhere.

The second option can be used if a public server is not available to you, or if you want to rapidly look at files that are already on your computer.

3 For this protocol, we will use the following files:
- HCT-116_PBase.ccf, a file of raw calling card insertions
- HCT-116_PBase.bedgraph, a file of calling card insertion densities across the genome
- HCT-116_PBase_peaks.bed, a file of peaks inferred from calling card data

Uploading Data from an External Server

4 BED, bedgraph, and CCF files are all text-based formats. Before being hosted on an external server, these files must be compressed and indexed. This allows for fast, random-access to the data. For this step, make sure you have installed htslib (see Materials).

To compress the file:

```
bgzip HCT-116_PBase.ccf
```

This will compress the original file, creating HCT-116_PBase.ccf.gz

To index the compressed file:

```
tabix -p bed HCT-116_PBase.ccf.gz
```

This will create an index file, HCT-116_PBase.ccf.gz.tbi

Both HCT-116_PBase.ccf.gz and HCT-116_PBase.ccf.gz.tbi should be copied to a publicly-accessible webserver. These steps also apply to bedgraph and BED files.

5 Next, we create a JSON file. JSON is a generic standard describing objects and their properties. Here, we use it to describe each track, its data, and any options we may want to customize. The JSON file is plain text; it is recommended that you use a text editor (e.g. Sublime, Atom, Visual Studio Code, vim/emacs, etc.) instead of a word processor (e.g. Microsoft Word) to create this file.

We will create HCT-116_PBase.json for our data. Here is the structure of a JSON file for a single track depicting calling card data:

```
[
    {
        "type":"callingcard",
        "url":"https://htcf.wustl.edu/files/xXl8ZAXy/HCT-116_PBase.ccf.gz",
        "name":"piggyBac insertions",
        "showOnHubLoad":"true",
        "options":{
            "color":"#3182bd",
            "height":100,
            "logScale":"log10",
            "markerSize":3,
            "opacity":[100],
            "show":"all",
            "sampleSize":1000,
        },
    },
]
```

All tracks in the JSON file must be between the square brackets [...]. Curly braces {...} and commas separate invidual tracks, and within a track, another set of curly braces may be used to specify grouped options. Let's consider each of these entries in turn:

```
"type":"callingcard",
```

This specifies the track type, which in this case is a calling card track.

```
"url":"https://htcf.wustl.edu/files/xXl8ZAXy/HCT-116_PBase.ccf.gz",
```

This points to the web address where your data are stored. Note that this must point to the compressed file. The browser will automatically located the index file. (If the index file is not present, it will throw an error).

```
"name":"piggyBac insertions",
```

This is the track label.

```
"showOnHubLoad":"true",
```

This specifies whether the track should be display immediately after the JSON file has been uploaded. The default value is "false."

There are also a number of options that can be specified to customize the track appearance. While all of them can be changed after the tracks have been loaded, specifying them in the JSON file helps to record and reproduce settings. Note that these only need to be set if you wish to override defaults.

```
"color":"#f4916c",
```

This sets the color of the individual calling card markers. The defaul is blue.

```
"height":100,
```

This sets the height of the track in pixels. The default is 40.

```
"logScale":"log10",
```

This transforms the y-axis from a linear scale to a logarithmic scale. By default, the track uses a linear scale, but for calling card experiments we recommed using a logarithmic scale.

```
"markerSize":3,
```

This is the radius of the marker in pixels. The default value is 3.

```
"opacity":[100],
```

This specifies the opacity of the markers, which can help emphasize data-dense regions. This takes an integer value between 0 and 100. The value supplied must be in square brackets. The default value is 100.

```
"show":"all",
"sampleSize":1000,
```

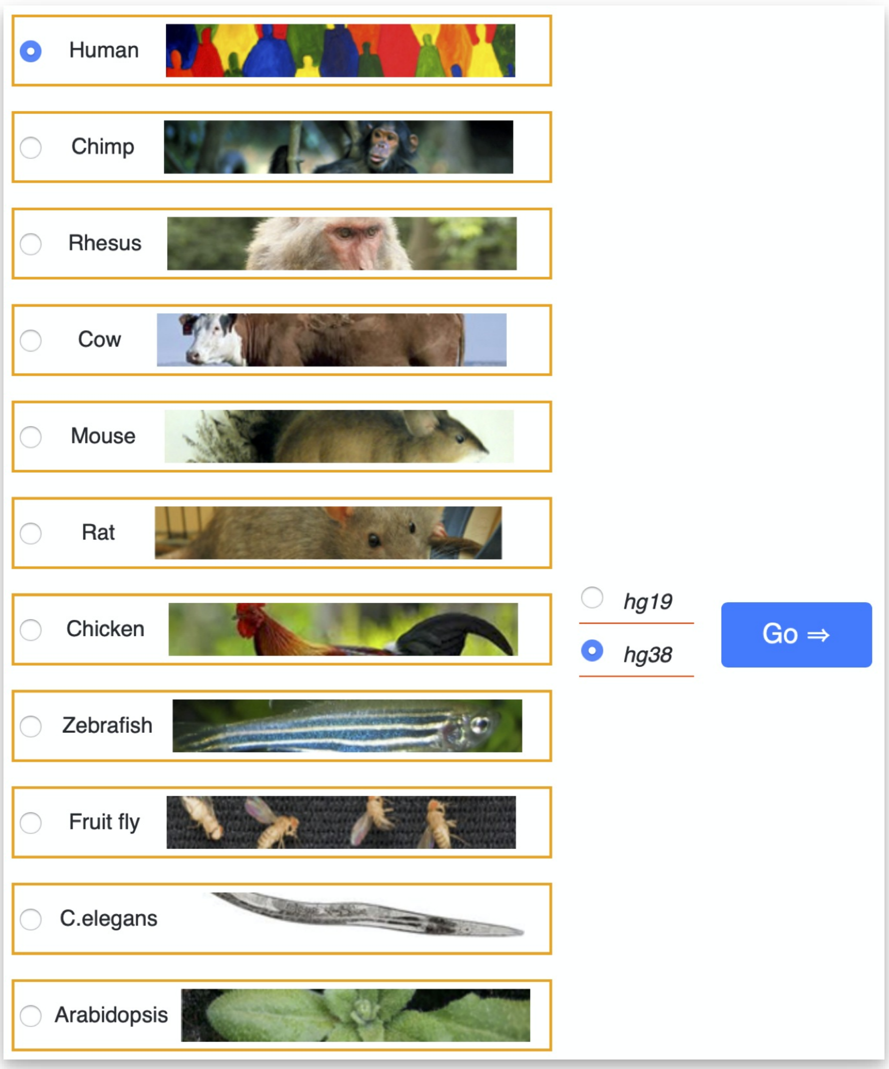
This pair of options specify whether the browser should display all data points, or simply a random subsample. If there are many (e.g. thousands) of data points in view, it can slow down your web browser. Subsampling can help preserve a global representation of your data while still remaining responsive and memory-friendly. The "show" option can take two values: "all" (default) or "sample." The former draws all data points. If the latter is specified, then the "sampleSize" option is applied. This number determines how many points to subsample and is set to 1000 by default.

6 We can also add the bedgraph and BED files to the JSON:

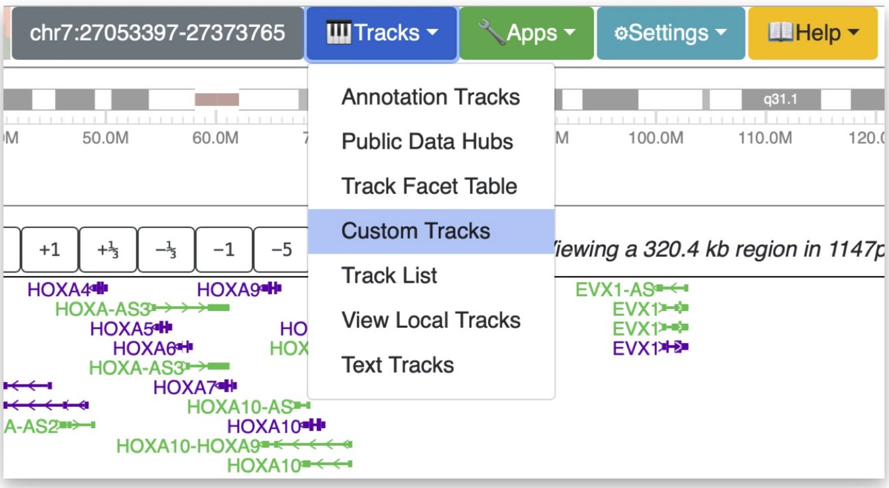
```
[
    {
        "type":"callingcard",
        "url":"https://htcf.wustl.edu/files/xXl8ZAXy/HCT-116_PBase.ccf.gz",
        "name":"piggyBac insertions",
        "showOnHubLoad":"true",
        "options":{
            "color":"#3182bd",
            "height":100,
            "logScale":"log10",
            "markerSize":3,
            "opacity":[100],
            "show":"all",
            "sampleSize":1000,
        },
    },
    {
        "type":"bedgraph",
        "url":"https://htcf.wustl.edu/files/xXl8ZAXy/HCT-116_PBase.bedgraph",
        "name":"HCT-116 PBase density",
        "showOnHubLoad":"true",
        "options":{
            "color":"#3182bd",
            "height":50,
        },
    },
    {
        "type":"bed",
        "url":"https://htcf.wustl.edu/files/xXl8ZAXy/HCT-116_PBase.bed",
        "name":"HCT-116 PBase peaks",
        "showOnHubLoad":"true",
        "options": {
            "color":"#3182bd",
            "height":20,
            "displayMode":"density",
        }
    },
]
```

Many options have the same names and settings as the calling card track. A complete list of options for bedgraph and BED tracks can be found here.

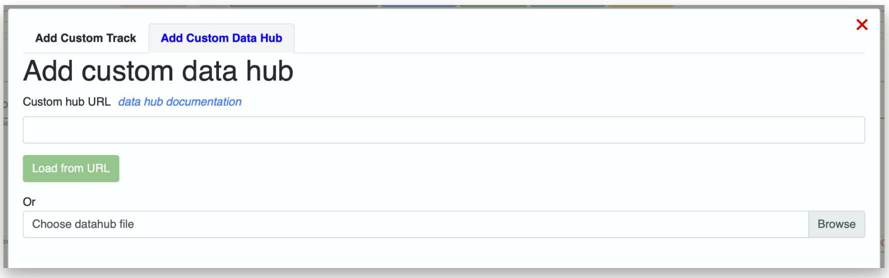
7 Now we can upload this file to the browser and visualize it. Go to the homepage and select the appropriate genome:



8 Once the genome has loaded, click on the blue "Tracks" button and select "Custom tracks."

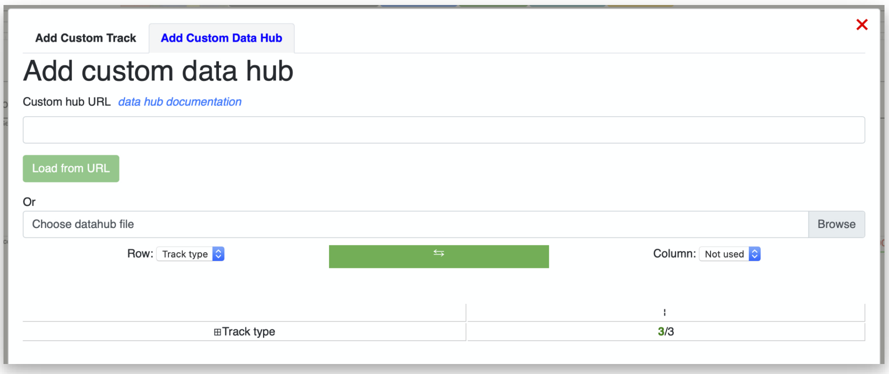


9 Select the second tab, "Add Custom Data Hub."



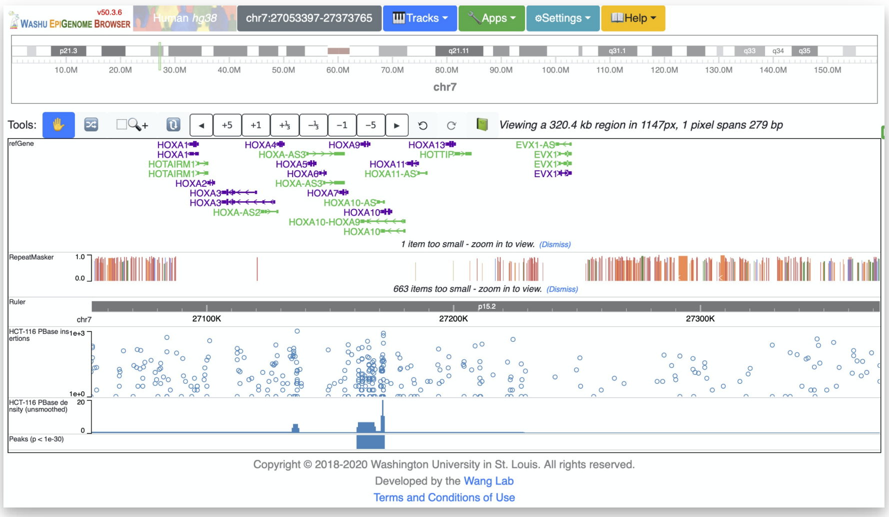
Clicking on the second dialog box ("Choose datahub file") will open a filesystem navigator window. Find the JSON file on your system and upload it.

Upon successful upload, you should see a table listing the uploaded tracks:



⊞Track type	3/3

10 After closing the custom track pane (with the red X in the upper right corner), all three tracks should be visible:

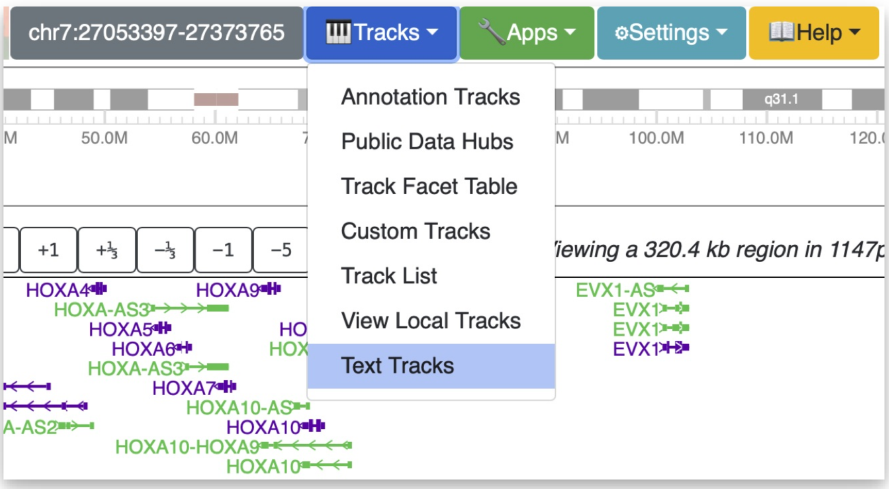


Note that the gencodeV29 track has been removed for this screenshot.

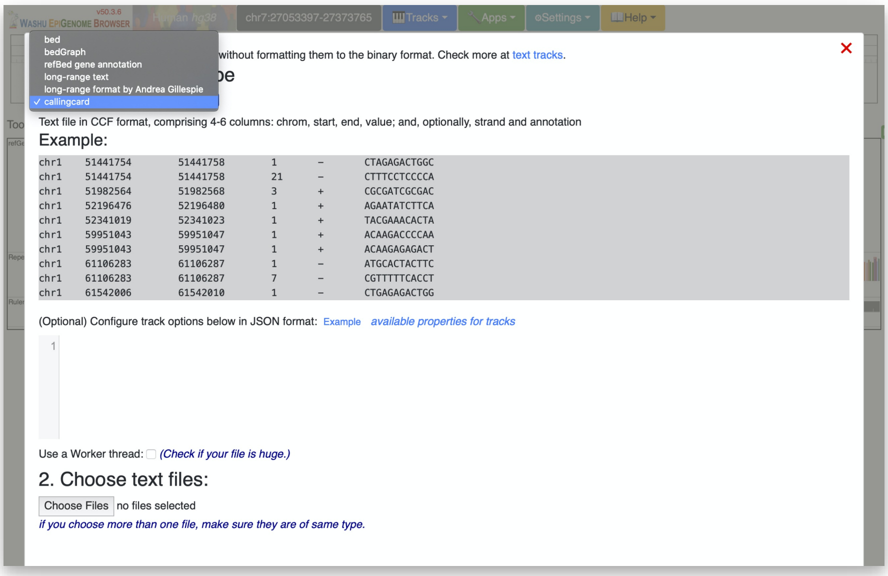
Uploading Local Data Files

11 As an alternative to hosting your data externally, you can directly upload text-based data files, including CCF, bedgraph, and BED. If you choose this option, your files **should not** be compressed. Also, for very large files, this approach may slow down your browser.

12 Click on the blue "Tracks" button and select "Text track."



13 Select "callingcard" from the dropdown menu, then click on the "Choose Files" button. Select one or more CCF files to upload. The browser will then load the tracks; this may take a few moments, depending on the size of the file(s).



without formatting them to the binary format. Check more at text tracks.

Text file in CCF format, comprising 4-6 columns: chrom, start, end, value; and, optionally, strand and annotation

Example:

```
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
```

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

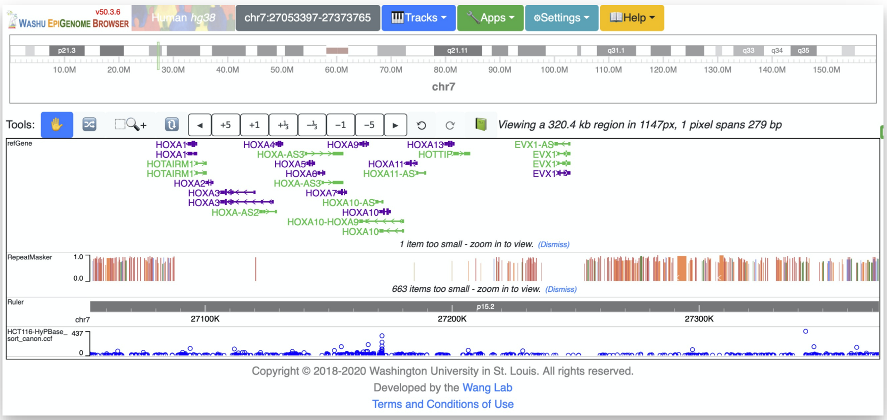
Use a Worker thread: (Check if your file is huge.)

2. Choose text files:

Choose Files no files selected

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

14 After the tracks have been loaded, close the pane by clicking on the red X in the top right corner. When directly uploading text files, tracks will be rendered with default options.



Viewing a 320.4 kb region in 1147px, 1 pixel spans 279 bp

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

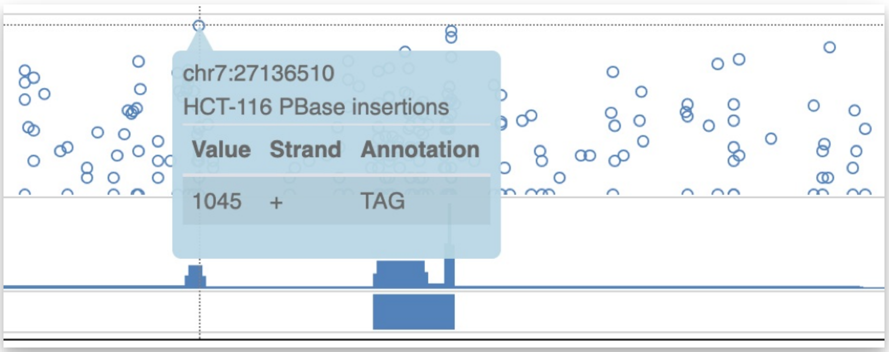
663 items too small - zoom in to view. (Dismiss)

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

These steps can be repeated for BED and bedgraph files.

Interacting with the Calling Card Track

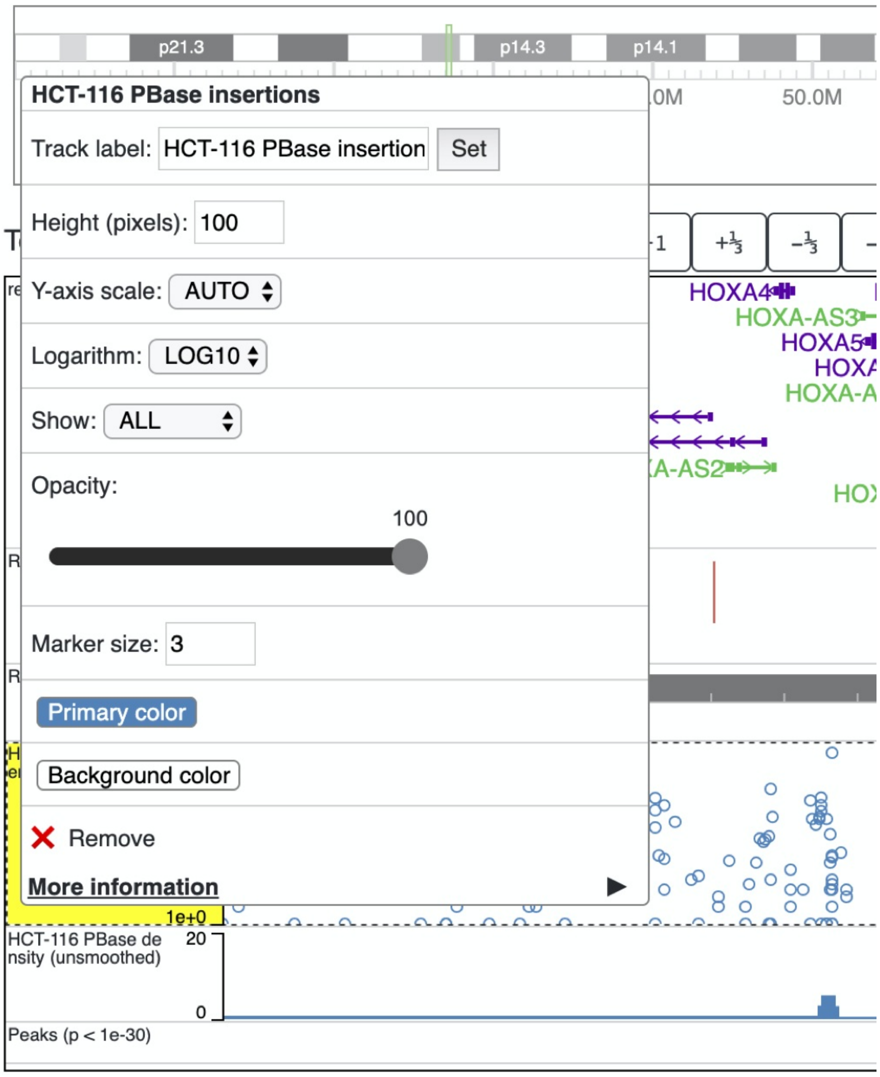
15 The calling card track was designed for interactive exploration. This is chiefly accomplished by a rollover box that appears as the mouse cursor nears a data point. An approximate location is at the top of the pane, while the bottom lists the read count, strand, and barcode for each insertion:



chr7:27136510
HCT-116 PBase insertions

Value	Strand	Annotation
1045	+	TAG

16 Right clicking on the calling card track will bring up a preference pane, where tracks can be dynamically customized:


HCT-116 PBase insertions

Track label: HCT-116 PBase insertion Set

Height (pixels): 100

Y-axis scale: AUTO

Logarithm: LOG10

Show: ALL

Opacity: 100

Marker size: 3

Primary color

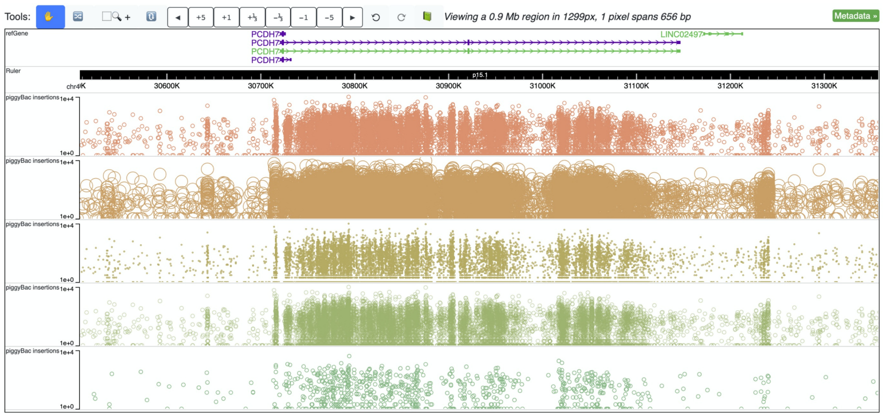
Background color

Remove

More information

SImilar preference panes exist for the BED and bedgraph tracks.

17 This example showcases all the different options available for calling card tracks, such as color, marker size, opacity, and subsampling:



Tools:
Viewing a 0.9 Mb region in 1299px, 1 pixel spans 656 bp
Metadata »
refGene
PCDH7
LINC02497
Ruler
p15.1
chr4
30600K
30700K
30800K
30900K
31000K
31100K
31200K
31300K
piggyBac insertions
1e+4
1e+0