

Overview

Dovetail Hi-C data describe which regions of a genome are physically close together. These data can be visualized as a heatmap. Each segment of this heatmap represents how often two regions of the genome were found to be close enough together to be ligated. This heatmap can be viewed at various zoom levels to explore your genome and its three-dimensional shape.

For example, when zoomed out all the way, it is often possible to see that chromosomes are visible as large squares in the heatmap. On the other end of the spectrum, zoomed in tightly in a single chromosome, it is often possible to see much smaller squares of high contact. These squares are often interpreted as topologically associated domains (TADs) – regions of a chromosome that are wadded up such that contacts within a TAD are high and contacts between TADs are low. It may also be possible to see loop contacts and other features in the two-dimensional heatmap data that correspond to specific three-dimensional shapes of the chromosomes.

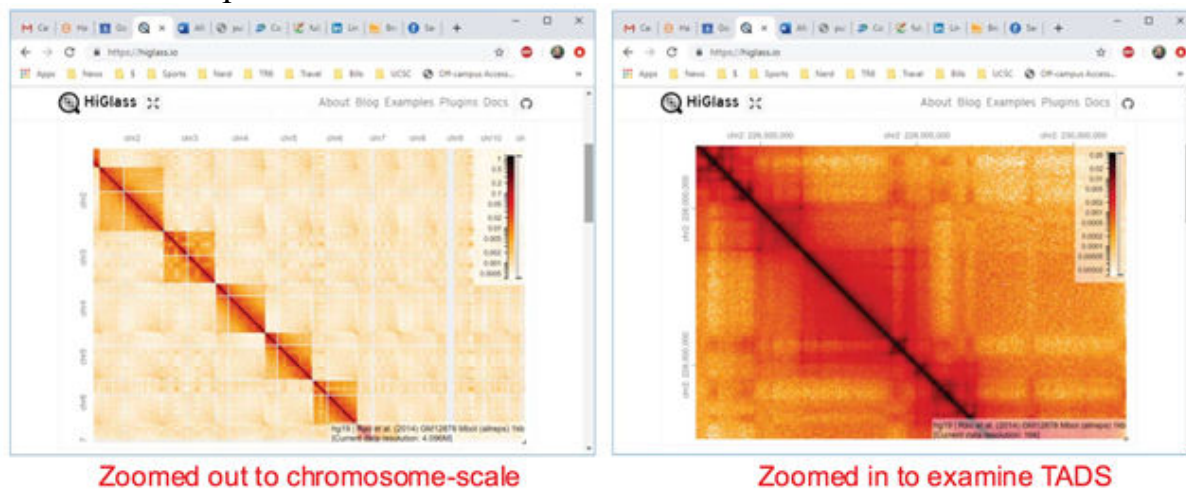


Figure 1 - Dovetail HiC contact heatmap visualized using the HiGlass browser. Regions of close proximity (more observed contacts) are shown in a darker red. Left, HiGlass zoomed out to chromosome-scale showing intra-chromosomal proximity. Right, HiGlass zoomed in show topologically associated domains (TADs) within a section of a chromosome.

The pairwise proximity heatmap data have been calculated using a set of standard procedures described in the report which is included in the delivery package. We have also provided files for visualizing TADs, A/B compartments, isochores, and CTCF binding sites. The rest of this document will describe how to explore these data using the publicly available HiGlass browser and other tools. The instructions here are just the beginning!

Quick Start Guide

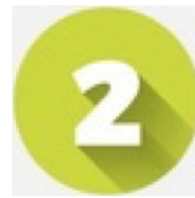
To view and explore the heatmap, you can use the [HiGlass Browser](#). Installing the HiGlass browser on your local machine and loading your data are surprisingly simple procedures. Below are step-by-step instructions for different machine types: UNIX/Linux and Mac. The main concept is that you will install a [docker](#) instance on your machine. Docker is a “virtualization container” - a program that lets you run other programs - in this case the HiGlass server, on any machine. First, we’ll install docker and get it running. Then, we’ll use docker to pull in the [HiGlass](#) docker image. Next, we’ll ingest the Hi-C and other data and have a look!

Note: System requirements for setting up docker, HiGlass, and hosting your data are as follows: The raw docker disk image for Mac is 521Mb, the HiGlass docker image is about 1Gb and a typical distribution of Hi-C and associated data is about 10 Gb. Make sure to perform the HiGlass installation on a computer with enough free disk space for the software and data!



Get Required Software

- Install Docker
- Install HiGlass



Load Data Into HiGlass

- Download data
- Ingest Coordinate System
- Ingest mcool file
- Ingest TADs
- Add Isochores
- Add CTCF sites
- Add A/B Compartments

Step 1: GET REQUIRED SOFTWARE

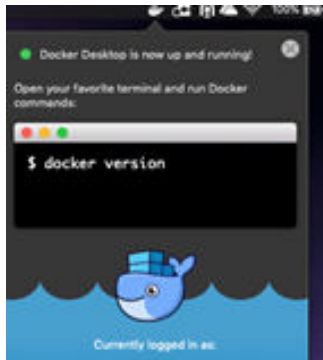
NOTES:

- This is the part of the guide that has different instructions depending on the local machine you are using: Mac or Linux. Step 2 that outlines the instructions to load data into HiGlass is the same for Mac and Linux.
- Throughout this document command lines that need to be copied and pasted into your terminal will be written in **green**

Mac

Install Docker

1. In order to download and install the free, Docker Community Edition software you must register. Visit <https://hub.docker.com/signup> to make a user account and follow the steps for account verification.
2. Download and install [Docker Engine Community](#). Once the Docker.dmg file has downloaded, click it to open. You'll be instructed to drag the Docker image into the Applications Folder. Do that. Then, go to your Applications and launch Docker by clicking it. Follow the instructions for running docker for the first time. Once docker has started successfully, you'll see the little docker icon (figure below) along the top tool bar.



Install HiGlass

1. Open Applications->Utilities->Terminal and use docker to pull (download and install) the HiGlass browser

docker pull gehlenborglab/higlass

```
thomasswale$ docker pull gehlenborglab/higlass
```

Depending on your network connection, this may take a few minutes. This is a good place to grab a cup of coffee.

2. Fire up the HiGlass browser through docker

```
docker run --detach --publish 8888:80 --volume ~/hg-data:/data --  
volume ~/hg-tmp:/tmp --name higlass-container  
gehlenborglab/higlass
```

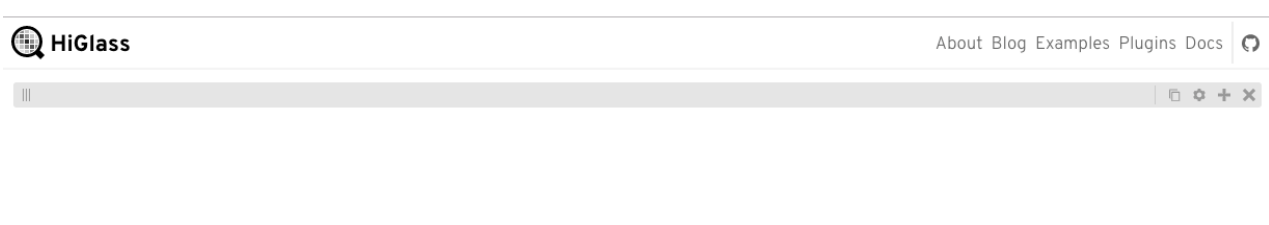
```
(base) Thomas-Swale-MacBook-Pro:~ thomasswale$ docker run --detach --publish 8888:80 --volume  
~/hg-data:/data --volume ~/hg-tmp:/tmp --name higlass-container gehlenborglab/higlass
```

PROGRESS CHECK: Did it work? Open a browser and go to: <http://localhost:8888>

You should see the HiGlass browser, ready to load and explore your data!

This localhost is a special web address that means “my machine”. This means that the website is not coming across the internet. Instead, it’s being served by your own machine.

NOTE: Since this is hosted on your own machine, if you shut down your computer your work will be erased.



- 3. What happened?** We now have HiGlass running under docker. We gave it a name (“higlass-container”, unimaginatively) that can be used to stop it and remove it. The name can be anything. But it’s important to remember what it is. Should you want or need to stop and remove this HiGlass instance, use the commands below. You may want to do this, for example, if you change the location of where the input files are for HiGlass. It is not possible to change the parameters once you start it. The only way to do that is to stop it, remove it, and then restart it with the different parameters.

Note: you don't need to type the “#” characters or what follows after. This # character is just used to allow us to add comments.

```
docker stop higlass-container # replace higlass-container with the  
correct name, if different
```

```
docker rm higlass-container # replace higlass-container with the  
correct name, if different
```

Linux / UNIX

To install, Docker & HiGlass on your UNIX/Linux machine, you'll need root or sudo privileges. Overall, the procedure is straight-forward. The instructions below have been tested on a machine running Ubuntu 18.04.

Install docker

Installing docker requires sudo privileges. The instructions for installing docker are modified from this [docker tutorial](#). There are alternative instructions for various Linux distributions including [CentOS 7](#), [Debian 10](#), and [Ubuntu 16.04](#).

1. From the command line, start the bash shell. These commands work in the bash shell. Other shells have not been tested.

```
bash
```

2. Before we install Docker, make sure your system is up to date.

```
sudo apt update
```



```
ubuntu@ip-172-31-43-219:~$ sudo apt update
Hit:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Get:5 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:6 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 kB]
Get:7 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
Get:8 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]
Get:9 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [710 kB]
Get:10 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [259 kB]
Get:11 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [11.9 kB]
Get:12 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted Translation-en [4156 B]
Get:13 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [997 kB]
Get:14 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [305 kB]
Get:15 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [6636 B]
Get:16 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [3556 B]
Get:17 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [2512 B]
Get:18 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-backports/main Translation-en [1644 B]
Get:19 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [4000 B]
Get:20 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [1856 B]
Get:21 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [477 kB]
Get:22 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [162 kB]
Get:23 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [4976 B]
Get:24 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [2476 B]
Get:25 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [590 kB]
Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [195 kB]
Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [4004 B]
Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [2060 B]
Fetched 17.8 MB in 3s (5587 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
19 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

3. Install curl and some other required packages. Some of these may already be installed on your system.

```
sudo apt install apt-transport-https ca-certificates curl software-properties-common
```

```
ubuntu@ip-172-31-43-219:~$ sudo apt install apt-transport-https ca-certificates curl software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20180409).
ca-certificates set to manually installed.
curl is already the newest version (7.58.0-2ubuntu3.7).
curl set to manually installed.
The following additional packages will be installed:
  python3-software-properties
The following NEW packages will be installed:
  apt-transport-https
The following packages will be upgraded:
  python3-software-properties software-properties-common
2 upgraded, 1 newly installed, 0 to remove and 17 not upgraded.
Need to get 35.3 kB of archives.
After this operation, 153 kB of additional disk space will be used.
Do you want to continue? [Y/n] yes
Get:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 apt-transport-https all 1.6.11 [1692 B]
Get:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 software-properties-common all 0.96.24.32.11 [9996 B]
Get:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 python3-software-properties all 0.96.24.32.11 [23.6 kB]
Fetched 35.3 kB in 0s (3099 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 56638 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_1.6.11_all.deb ...
Unpacking apt-transport-https (1.6.11) ...
Preparing to unpack .../software-properties-common_0.96.24.32.11_all.deb ...
Unpacking software-properties-common (0.96.24.32.11) over (0.96.24.32.9) ...
Preparing to unpack .../python3-software-properties_0.96.24.32.11_all.deb ...
Unpacking python3-software-properties (0.96.24.32.11) over (0.96.24.32.9) ...
Setting up apt-transport-https (1.6.11) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Setting up python3-software-properties (0.96.24.32.11) ...
Processing triggers for dbus (1.12.2-1ubuntu1.1) ...
Setting up software-properties-common (0.96.24.32.11) ...
```

4. Add the GPG key for the official Docker repository so we can download and install it.
`curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

5. Add the Docker repository to the APT sources so that apt can get it, install it, and manage it.

```
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

```
ubuntu@ip-172-31-43-219:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
Hit:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease
Get:4 https://download.docker.com/linux/ubuntu bionic InRelease [64.4 kB]
Hit:5 http://security.ubuntu.com/ubuntu bionic-security InRelease
Get:6 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages [7889 B]
Fetched 72.3 kB in 0s (165 kB/s)
Reading package lists... Done
```

6. Now, an apt update will see Docker, fetch it, and set it up!
`sudo apt update`


```
ubuntu@ip-172-31-43-219:~$ sudo apt update
Hit:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:5 http://security.ubuntu.com/ubuntu bionic-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
17 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

sudo apt-get install -y docker-ce

```
ubuntu@ip-172-31-43-219:~$ sudo apt-get install -y docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  aufs-tools cgroupfs-mount containerd.io docker-ce-cli libltdl7 pigz
The following NEW packages will be installed:
  aufs-tools cgroupfs-mount containerd.io docker-ce docker-ce-cli libltdl7 pigz
0 to upgrade, 7 newly installed, 0 to remove and 17 not upgraded.
Need to get 88.0 MB of archives.
After this operation, 390 MB of additional disk space will be used.
Get:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 aufs-tools amd64 1:4.9+20170918-1ubuntu1 [104 kB]
Get:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 cgroupfs-mount all 1.4 [6320 B]
Get:4 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 libltdl7 amd64 2.4.6-2 [38.8 kB]
Get:5 https://download.docker.com/linux/ubuntu bionic/stable amd64 containerd.io amd64 1.2.6-3 [22.6 MB]
Get:6 https://download.docker.com/linux/ubuntu bionic/stable amd64 docker-ce-cli amd64 5:19.03.1~3-0~ubuntu-bionic [42.5 MB]
Get:7 https://download.docker.com/linux/ubuntu bionic/stable amd64 docker-ce amd64 5:19.03.1~3-0~ubuntu-bionic [22.7 MB]
Fetched 88.0 MB in 2s (50.7 MB/s)
Selecting previously unselected package pigz.
(Reading database ... 56642 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
Selecting previously unselected package aufs-tools.
Preparing to unpack .../1-aufs-tools_1%3a4.9+20170918-1ubuntu1_amd64.deb ...
Unpacking aufs-tools (1:4.9+20170918-1ubuntu1) ...
Selecting previously unselected package cgroupfs-mount.
Preparing to unpack .../2-cgroupfs-mount_1.4_all.deb ...
Unpacking cgroupfs-mount (1.4) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../3-containerd.io_1.2.6-3_amd64.deb ...
Unpacking containerd.io (1.2.6-3) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../4-docker-ce-cli_5%3a19.03.1~3-0~ubuntu-bionic_amd64.deb ...
Unpacking docker-ce-cli (5:19.03.1~3-0~ubuntu-bionic) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../5-docker-ce_5%3a19.03.1~3-0~ubuntu-bionic_amd64.deb ...
Unpacking docker-ce (5:19.03.1~3-0~ubuntu-bionic) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../6-libltdl7_2.4.6-2_amd64.deb ...
Unpacking libltdl7:amd64 (2.4.6-2) ...
Setting up aufs-tools (1:4.9+20170918-1ubuntu1) ...
Setting up containerd.io (1.2.6-3) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Processing triggers for ureadahead (0.100.0-21) ...
Setting up cgroupfs-mount (1.4) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.24) ...
Setting up libltdl7:amd64 (2.4.6-2) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Setting up docker-ce-cli (5:19.03.1~3-0~ubuntu-bionic) ...
Setting up pigz (2.4-1) ...
Setting up docker-ce (5:19.03.1~3-0~ubuntu-bionic) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.24) ...
```

7. Test docker installation using systemctl (requires sudo)

```
sudo systemctl status docker
```


This will query the system services on your machine for docker and report its status. If everything is copacetic, you will see information about the Docker Application Container Engine. The output should include a line that says “Active: active (running)”

```
ubuntu@ip-172-31-43-219:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2019-08-19 20:24:11 UTC; 39s ago
     Docs: https://docs.docker.com
    Main PID: 4722 (dockerd)
      Tasks: 9
   CGroup: /system.slice/docker.service
           └─4722 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

Alright! Take a deep breath. You just finished step one. Now docker is installed and running on your system. Docker is a very useful piece of software for running containers. Should you ever want or need to install and run docker containers for other tasks, you won't need to reinstall docker! You've done it. OK, on to installing HiGlass.

Install HiGlass

1. You will use Docker to install (pull) HiGlass

```
sudo docker pull higlass/higlass-docker
```

Depending on your network connection, it might take a few minutes to download everything. This is a good time to get yourself a nice beverage. Once it's all downloaded...

2. Start the HiGlass browser!

```
sudo docker run -detach --publish 8888:80 --volume ~/hg-data:/data
--volume ~/hg-tmp:/tmp --name higlass-container higlass/higlass-
docker
```

This will start an instance of HiGlass on your system. The name of this instance is higlass-container (specified by the --name option). It maps two file locations for you (the --volume options). This is how you map files from your file system into a virtual file system so that docker can see them. The containerized HiGlass instance will see any files in ~/hg-data in /data. It will also see files in ~/hg-tmp in /tmp. This can take a bit of getting used to. But the main idea is that files you put in ~/hg-data, for example, will be in location /data where you communicate with HiGlass.

3. Test the HiGlass browser

HiGlass is now installed and running. Open a browser and type 127.0.0.1:8888 in the address bar. You should see the HiGlass browser, waiting to show you some data. Pretty cool!

4. Stopping the HiGlass docker container

If you want to stop the HiGlass container, it's easy. You may want to do this, for example, to restart it with a different name or other attributes. Just do this:

```
docker stop higlass-container
```

Of course, if you named it something other than higlass-container, then type that other name instead.

Step 2: LOAD DATA INTO HIGLASS

Now that you have the HiGlass browser, we need to put your data into it, i.e., ingest the data. Below are step-by-step instructions for doing this.

NOTE: In this section we provide the command lines to complete these tasks in your terminal. You will need to adjust each command line to tailor it to the name of your output files and species. Directions will be given in a note below each command of what you will need to change.

Below each command line is a picture of the terminal for when we ingested the dog test data for reference.

Download your data

1. Download all the output data files. You would have received instructions from Dovetail Genomics describing where your output data files are located and how to download them.

2. Go to the directory where you have downloaded the output files.

Ingest the coordinate system

1. First, copy the chromosome size file into the hg-tmp folder using the terminal, enter command line:

```
cp chr_sizes.txt ~/hg-tmp/
```

```
thomasswale$ cp chr_sizes.txt ~/hg-tmp/
```

2. Then ingest the file into HiGlass using the script below:

```
sudo docker exec higlass-container python higlass-server/manage.py  
ingest_tileset --filename /tmp/chr_sizes.txt --datatype chromsizes  
--filetype chromsizes-tsv --coordSystem $species_name
```

NOTE: In the script above replace \$species_name with the actual name of your species. This will be the name of the coordinate system in HiGlass.

Ingest the mcool file (contact matrix) into HiGlass

1. First, copy the mcool file into hg-tmp folder:

```
cp X.mcool ~/hg-tmp/
```

NOTE: in the script above be sure to replace the X with the appropriate file name from your delivery package.

```
thomasswale$ cp dog.mcool ~/hg-tmp/
```

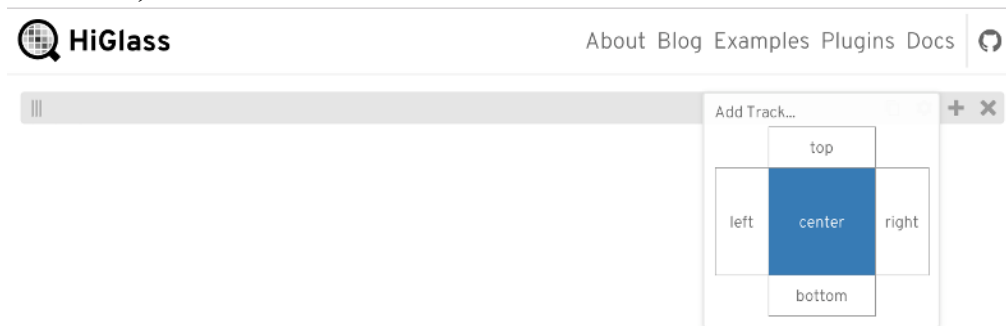
2. Then ingest this file in HiGlass as follows:

```
sudo docker exec higlass-container python higlass-server/manage.py  
ingest_tileset --filename /tmp/X.mcool --datatype matrix --  
filetype cooler --name "X Contact Matrix" --coordSystem  
$species_name
```

NOTE: in the command above be sure to edit the X and \$species_name appropriately before executing.

```
(base) Thomas-Swale-MacBook-Pro:Dog Tads thomasswale$ sudo docker exec higlass-container  
python higlass-server/manage.py ingest_tileset --filename /tmp/dog.mcool --datatype mat  
rix --filetype cooler --name "Dog Contact Matrix" --coordSystem Dog
```

PROGRESS CHECK: You can now open your HiGlass viewer. On the right you will see a “+” sign click this, then select “center”



This will open a subpage with all the files you have ingested

Click the box to the left of the file you want to ingest, in the contact matrix, select heat map, and then hit submit.

Add Track
✕

Select tileset:

>

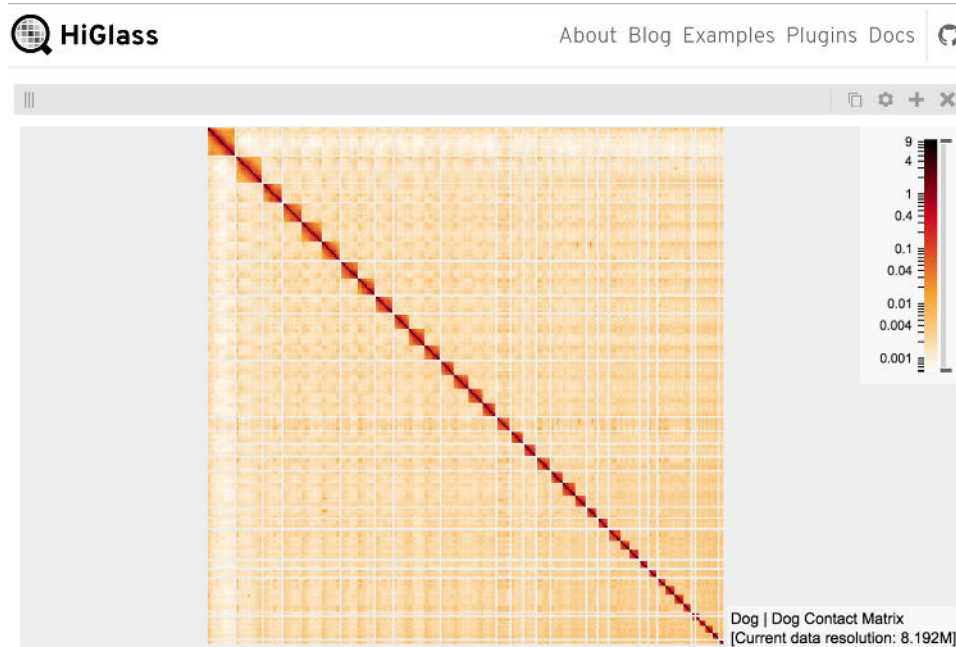
☐ Bonev et al. 2017
☐ chr_sizes.txt
☐ chr_sizes.txt
☐ chr_sizes.txt
☐ Chromosomes
☐ Dixon et al. 2012
☐ Dixon et al. 2015
☐ DOG AB Compartments
☒ Dog Contact Matrix
☐ DOG Isochores

heatmap
 2d-tiles

Search Term

Cancel
Submit

You should now be able to see your heat map:



HiGlass

About Blog Examples Plugins Docs

Dog | Dog Contact Matrix
[Current data resolution: 8.192M]

100 Enterprise Way, Suite A101 • Scotts Valley, CA • 95066 • (831) 713-4465

- 13 -

Ingest TADs (all 3 resolutions)

1. Copy the multires file into the hg-tmp folder.

```
cp X_TADs_10000.multires ~/hg-tmp/
```

NOTE: in the script above be sure to replace the X with the appropriate file name from your delivery package

```
thomasswale$ cp dog_TADs_10000.multires ~/hg-tmp/
```

2. Now ingest this file in HiGlass as follows:

```
docker exec higlass-container python higlass-server/manage.py  
ingest_tileset --filename /tmp/X_TADs_10000.multires --filetype  
bed2ddb --datatype 2d-rectangle-domains --coordSystem  
$species_name --name "X Juicer TADs 10000"
```

NOTE: in the above command replace the X and \$species_name as appropriate before executing.

```
(base) Thomas-Swale-MacBook-Pro:Dog Tads thomasswale$ docker exec higlass-container python  
higlass-server/manage.py ingest_tileset --filename /tmp/Dog_TADs_10000.multires --filetype  
bed2ddb --datatype 2d-rectangle-domains --coordSystem Dog --name "Dog Juicer TADs 10000"
```

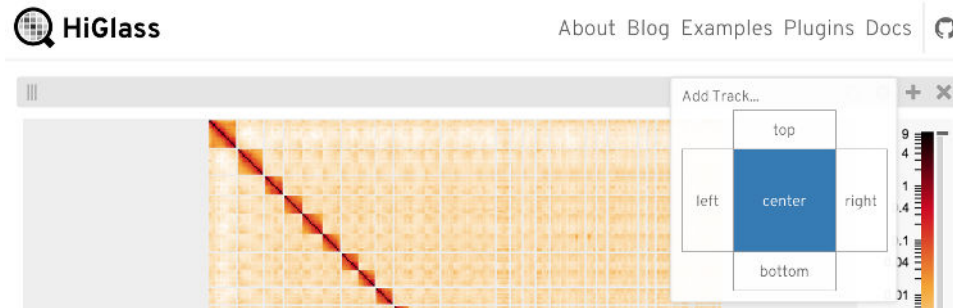
Steps 1 and 2 should be completed for all 3 resolutions in your delivery package: 10000 (10 Kbp), 25000 (25 Kbp), and 50000 (50 Kbp). You will need to adjust the resolution as appropriate. This can be done quickly in the terminal by using the up arrow to retrieve the previous command lines and changing 10000 to 25000 ...

```
(base) Thomas-Swale-MacBook-Pro:Dog Tads thomasswale$ docker exec higlass-container python  
higlass-server/manage.py ingest_tileset --filename /tmp/Dog_TADs_25000.multires --filetype  
bed2ddb --datatype 2d-rectangle-domains --coordSystem Dog --name "Dog Juicer TADs 25000"
```

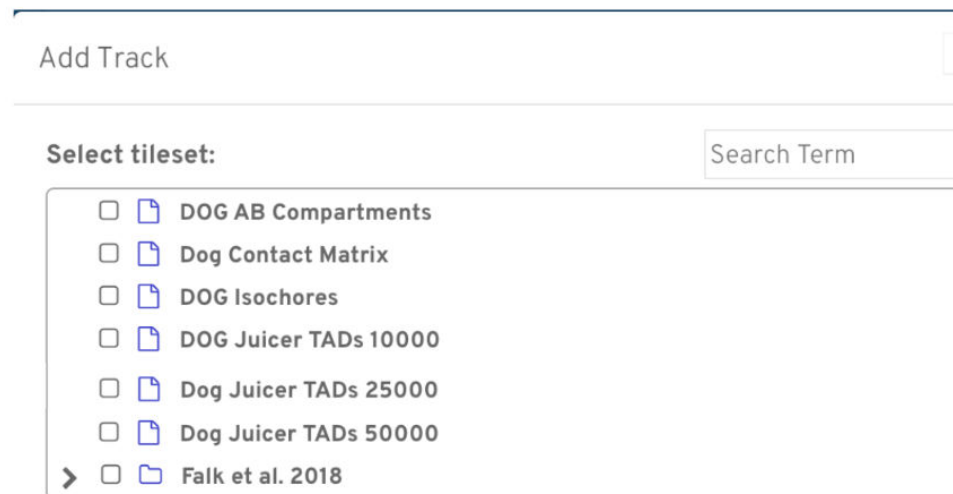
...then repeating for 50000. Be sure to do this for both command lines.

```
(base) Thomas-Swale-MacBook-Pro:Dog Tads thomasswale$ docker exec higlass-container python  
higlass-server/manage.py ingest_tileset --filename /tmp/Dog_TADs_50000.multires --filetype  
bed2ddb --datatype 2d-rectangle-domains --coordSystem Dog --name "Dog Juicer TADs 50000"
```

PROGRESS CHECK: Now when you enter HiGlass and push the “+” sign selecting the center location, you should see all the files you just uploaded.



Select the TADs at the resolution you want to see and 2d-rectangle-domains and hit submit.



Adding CTCF binding sites as a bed file

1. Copy the CTCF sites file into the hg-tmp folder.

```
cp X_CTCF_sites.multires ~/hg-tmp/
```

NOTE: in the script above be sure to replace the X with the appropriate file name from your delivery package

```
thomasswale$ cp dog_CTCF_sites.multires ~/hg-tmp/
```

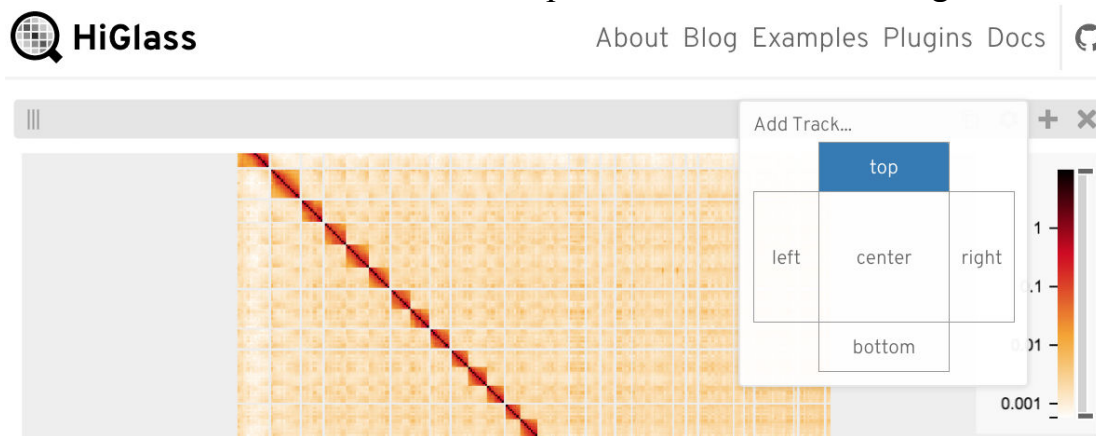
2. Now ingest this file in HiGlass as follows:

```
docker exec higlass-container python higlass-server/manage.py
ingest_tileset --filename /tmp/X_CTCF_sites.bed.multires --
filetype beddb --datatype bedlike --coordSystem $species_name -
-name "X CTCF Sites"
```

NOTE: in the above command replace the X and \$species_name as appropriate before executing.

```
(base) Thomas-Swale-MacBook-Pro:Dog Tads thomasswale$ docker exec higlass-container python
higlass-server/manage.py ingest_tileset --filename /tmp/Dog_CTCF_Sites.multires --filetype
beddb --datatype bedlike --coordSystem Dog --name "Dog CTCF Sites"
```

PROGRESS CHECK: To view in HiGlass open the browser hit “+” sign then select **top**



Select CTCF Sites and bedlike and hit submit

Add Track

Select tileset:

☐ chr_sizes.txt
☐ chr_sizes.txt
☐ chr_sizes.txt
☒ Chromosomes
☐ Dixon et al. 2012
☐ Dixon et al. 2015
☐ DOG AB Compartments
☐ Dog Contact Matrix
☒ DOG CTCF Sites

bedlike

Cancel Submit

Adding Isochores

1. Copy the isochores file into the hg-tmp folder.

```
cp X_isochores.multires ~/hg-tmp/
```

NOTE: in the script above be sure to replace the X with the appropriate file name from your delivery package

```
thomasswale$ cp dog_isochores.multires ~/hg-tmp/
```

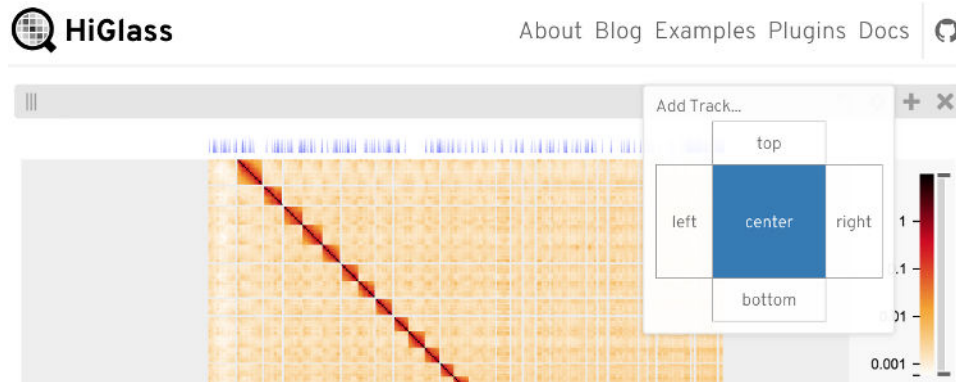
2. Now ingest this file in HiGlass as follows:

```
docker exec higlass-container python higlass-server/manage.py
ingest_tileset --filename /tmp/ X_isochores.multires --filetype
bed2ddb --datatype 2d-rectangle-domains --coordSystem
$species_name --name "X Isochores"
```

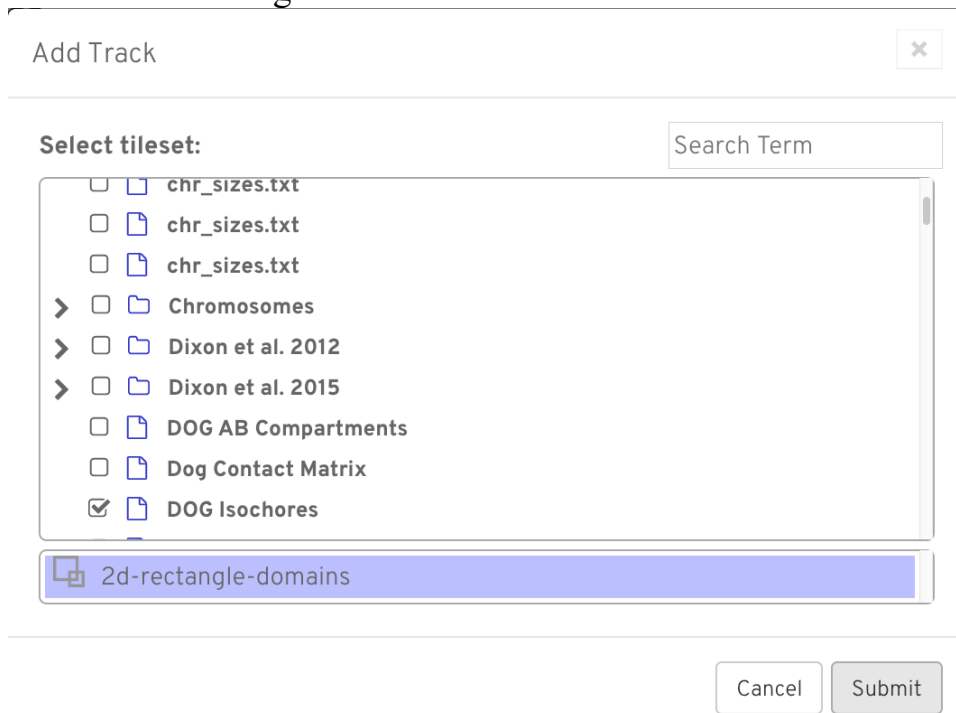
NOTE: in the above command replace the X and \$species_name as appropriate before executing.

```
(base) Thomas-Swale-MacBook-Pro:Dog Tads thomasswale$ docker exec higlass-container python
higlass-server/manage.py ingest_tileset --filename /tmp/dog_isochores.multires --filetype
bed2ddb --datatype 2d-rectangle-domains --coordSystem Dog --name "Dog Isochores"
```

PROGRESS CHECK: To view in HiGlass open the browser hit “+” sign then select Center



Select Isochores and 2d-rectangle-domains and hit submit



Add Track

Select tileset:

Search Term

- ☐ chr_sizes.txt
- ☐ chr_sizes.txt
- ☐ chr_sizes.txt
- ☐ Chromosomes
- ☐ Dixon et al. 2012
- ☐ Dixon et al. 2015
- ☐ DOG AB Compartments
- ☐ Dog Contact Matrix
- ☒ DOG Isochores

2d-rectangle-domains

Cancel Submit

Adding AB Compartments

1. Copy the AB compartments file into the hg-tmp folder:
`cp X_AB_compartments.bedpe.multires ~/hg-tmp/`

NOTE: in the script above be sure to replace the X with the appropriate file name from your delivery package

```
thomasswale$ cp DOG_AB_compartments.multires ~/hg-tmp/
```

2. Now ingest this file in HiGlass as follows:
`docker exec higlass-container python higlass-server/manage.py ingest_tileset --filename /tmp/X_AB_compartments.bedpe.multires --filetype bed2ddb --datatype 2d-rectangle-domains --coordSystem $species_name --name "X AB Compartments"`

NOTE: in the above command replace the X and \$species_name as appropriate before executing.

```
(base) Thomas-Swale-MacBook-Pro:Dog Tads thomasswale$ docker exec higlass-container python higlass-server/manage.py ingest_tileset --filename /tmp/DOG_AB_compartments.multires --filetype bed2ddb --datatype 2d-rectangle-domains --coordSystem Dog --name "Dog AB Compartments"
```

PROGRESS CHECK: You can now open your HiGlass viewer. On the right you will see a “+” sign click this, then select “center”





Select AB Compartments, and then hit submit.


Add Track


Select tileset:


Search Term


> ☐  Alavattam et al. 2019


> ☐  Bonev et al. 2017


☐  chr_sizes.txt


☐  chr_sizes.txt


☐  chr_sizes.txt

> ☐  Chromosomes

> ☐  Dixon et al. 2012

> ☐  Dixon et al. 2015

☒  DOG AB Compartments

 2d-rectangle-domains

Cancel

Submit

100 Enterprise Way, Suite A101 • Scotts Valley, CA • 95066 • (831) 713-4465

- 20 -