glue genes Documentation

glue genes is a custom version of the glue software for the visual exploration of genomics data.

Feature Overview

glue genes provides all the core features of glue:

- Interactive, linked statistical graphics
- A user interface for 'glueing' together multiple datasets with complicated relationships
- A highly scriptable and extendable environment

In addition, glue genes provides:

- Data loaders for genomics data file formats such as .bed, .bedgraph, .bedpe, .bigwig, and .hic
- Viewers including: Genome Track Viewer, 2D Heatmap, Dendrogram/Tree Viewer, Network Viewer
- Changes to core-glue to enable easy exploration of genetics data

Getting started with glue genes

Installation

Due to some complicated binary dependencies, the recommended procedure for installing this package is to use conda and intall into a new environment as follows:

```
conda create -n glue-genes python==3.8
conda activate glue-genes
conda install -c glueviz glueviz
```

```
conda install -c bioconda pairix
conda install -c bioconda tabix
pip install glue-genes@git+https://github.com/gluesolutions/glue-genes.git
```

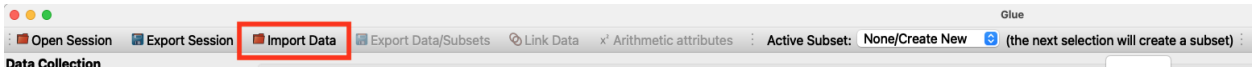
Getting started

If you are entirely new to glue, the Getting Started tutorial from the main glue documentation provides a thorough walk-through of the interface and main features of glue, and is the best place to start learning how to use glue.

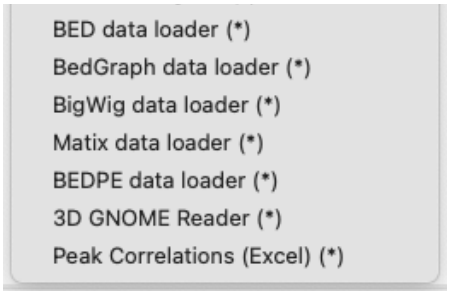
Data Loaders

The first step to analyzing data in glue is to get your data into glue. glue genes contains custom data loaders for a wide variety of genomics data-types, and it is also possible to build your own data loaders for custom data types.

Choose Import Data from the glue toolbar or the File menu and navigate to your datafiles in the window that pops up.



glue recognizes many data file types by file extension, so you can often rely on the default file loader (Auto) to use the correct data loader. If that fails, you can select a specific data loader for your data file. Note that you can select multiple files at once as a convenient way to load in data.



BED data loader

BED (.bed) -- assumes that the first three columns of your bedfile are chr, start, and end and names these automatically while leaving the rest of the column names unspecified. If your dataset features additional columns, you will probably want to give them sensible names by selecting *Data Manager* > *Reorder/rename data attributes* from the menu bar or right-clicking on the Data object in the Data Collection window and selecting *Reorder/rename data attributes*. The first three columns should NOT be renamed, as glue genes assumes these are present and named in this standard fashion when placing genome regions on other datasets.

BedGraph data loader

BedGraph (.bedgraph) -- Continuous-valued data on a genome track. This data loader is very slow the first time it is used on a dataset; glue will seem to hang while it is tiling and indexing the data for quick retrieval in the Genome Track Viewer. In the current version of glue genes, this

data loader creates a custom glue Data object to enable fast retrieval of data from disk. As a consequence, the Data object created can only be displayed in the Genome Track Viewer, and many other operations within glue (such as creating links) will not work in the standard way.

BEDPE data loader

BEDPE (.bedpe) -- Loops and other long-range connections between genome features. This data loader is very slow the first time it is used on a dataset; glue will seem to hang while it is tiling and indexing the data for quick retrieval in the Genome Track Viewer. In the current version of glue genes, this data loader creates a custom glue Data object to enable fast retrieval of data from disk. As a consequence, the Data object created can only be displayed in the Genome Track Viewer, and many other operations within glue (such as creating links) will not work in the standard way.

BigWig data loader

BigWig (.bigwig) -- Indexed binary format with coverage data. This is a fairly naive data loader that loads the full dataset into memory (via pyranges) rather than taking advantage of the indexed data format.

3D GNOME Reader

3D-GNOME model (.stl) -- A model generated and downloaded from [3D-GNOME 2.0]. This software creates a 3D model of genome structure from ChIA-PET or Hi-C like data. The model file (in STL format) must be downloaded from the website with the following parameters under GENERATE STL FILE option:

Line Segments = 10000
Tube Radius = 0
Radial Segments = 3
Save as ASCII STL

This data format does not contain information about the length of the chromosome that has been modelled, which glue needs to know in order to figure out the genomes coordinates along the strand. Currently this is hard-coded in the reader to be chromosome 3, but will eventually be exposed in the loading UI. Once loaded, this data can be visualized using the 3D Scatter Viewer.

Peak Correlations

A JAX-specific Excel file (.xlsx) -- giving known correlations or associations between genome regions-of-interest in a comma-separated fashion. The reference dataset is *Example_Peaks_and_Corr.xlsx*. This loader is unlikely to be useful outside of this specific

dataset format, but serves as an example for how to parse complicated Excel datafiles into glue. This Data Loader must be chosen explicitly, otherwise the default Excel data loader is used.

Matrix data loader

Experiment data matrix (RNA-seq or ATAC-seq) (.txt) -- This loader is specifically designed to read in JAX Three Bears RNA-seq and ATAC-seq experimental data and the associated metadata files. The reference datasets are *three_bears_liver_rnaseq_matrix_counts.txt*, *three_bears_liver_rnaseq_matrix_metadata.txt* and *three_bears_liver_rnaseq_geneInfo.txt*. This Data Loader is unlikely to be broadly useful outside of this specific dataset, but serves as an example for how to parse less structured datasets into glue.

Hi-C data loader

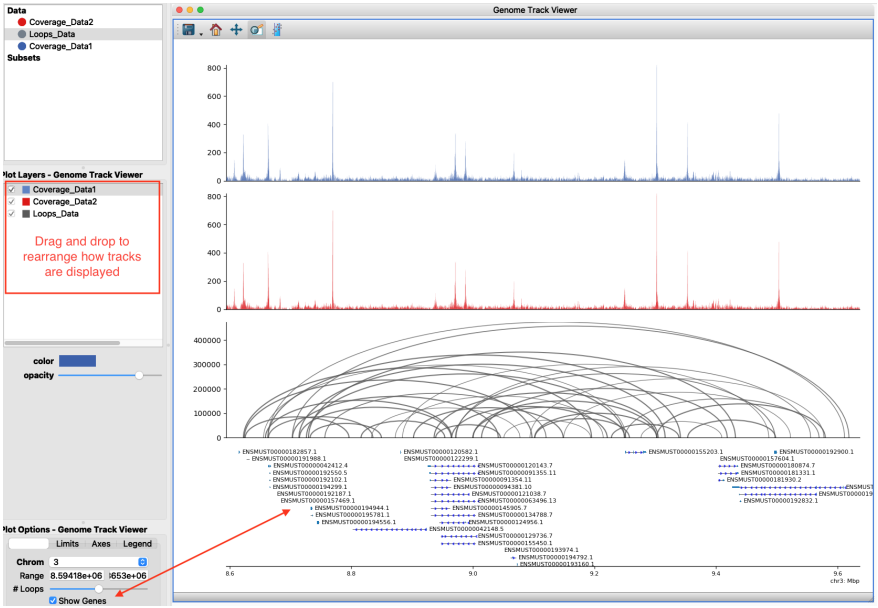
Due to challenges with the installation and distribution of the *straw* package used for reading in Hi-C datasets, this data loader is currently disabled in glue genes.

Data Viewers

Genome Track Viewer

The Genome Track Viewer displays genome-scale datasets. Files of type .bedpe and .bedgraph are read into glue in a special way that allows them to be displayed on Genome Track Viewers (but not other Viewers). Multiple datasets can be dragged into a single Genome Track Viewer to display them all in the same viewport or multiple Genome Track Viewers can be created to see data at multiple resolutions concurrently. Note that the X axis is formatted in kbp/Mbp and displays the chromosome:

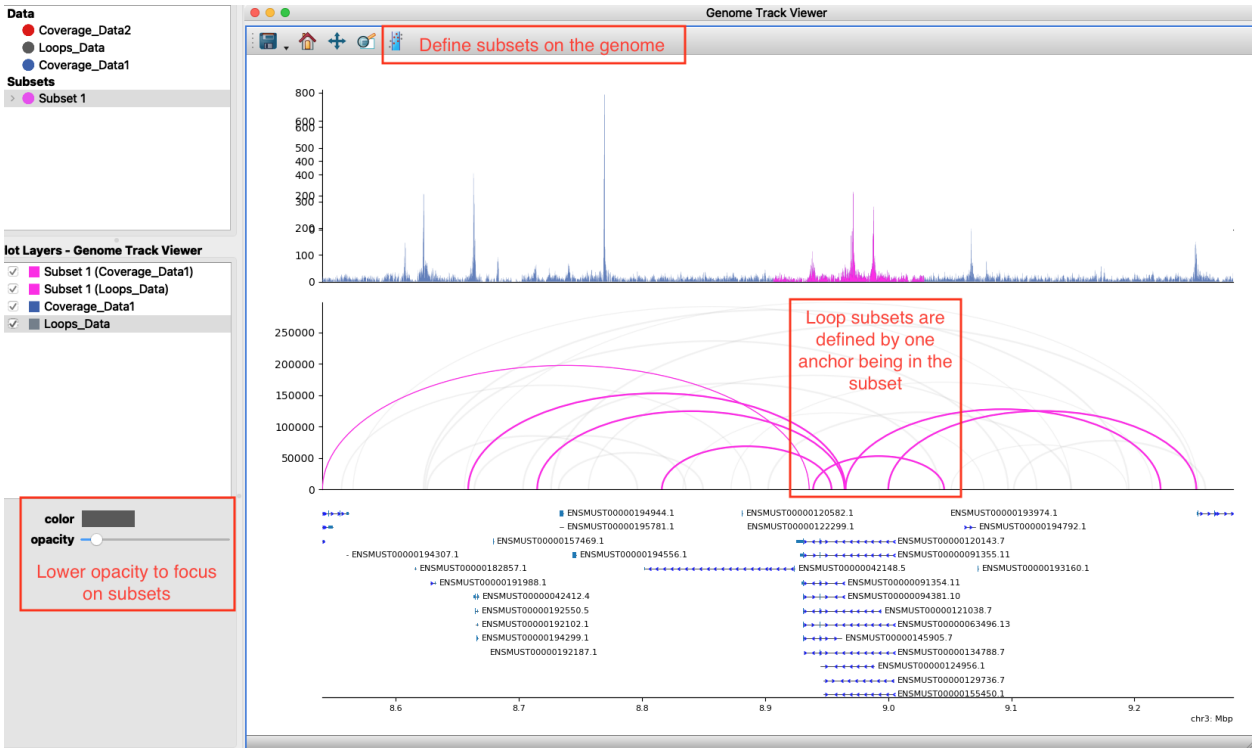
Within a single Genome Track Viewer the order in which datasets are displayed is controlled by re-ordering the datasets in the Plot Layers sidebar. Individual colors and opacity for these datasets can be configured here as well.



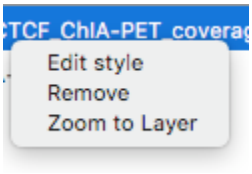
In the prototype, a single annotation dataset (GENCODEvm23) is available for display (these annotations can be toggled on and off), future versions will allow for the display of additional annotation layers.

The Genome Track Viewer displays datasets at multiple (pre-gridded) resolutions. This is most noticeable with loop-type data: loops are only displayed when both ends are present within the current view window, and loops that end and start in the same resolution element are aggregated. In addition, for performance reasons, there is an upper limit on the number of loops that are visualized concurrently (this can be adjusted in the Plot Options), with the weakest loops being dropped to reach the target number of loops. Zooming and panning the loop data can therefore result in slightly surprising results; sometimes it is worth zooming in and out around a region of interest to see how the loop structure changes as the result of changes in resolution.

The Genome Track Viewer can be used to define subsets on the genome. Loop subsets are defined by one anchor (endpoint) being within a genome subset. It is useful to lower the opacity of the full looping dataset to focus on a particular subset of loops as shown below:

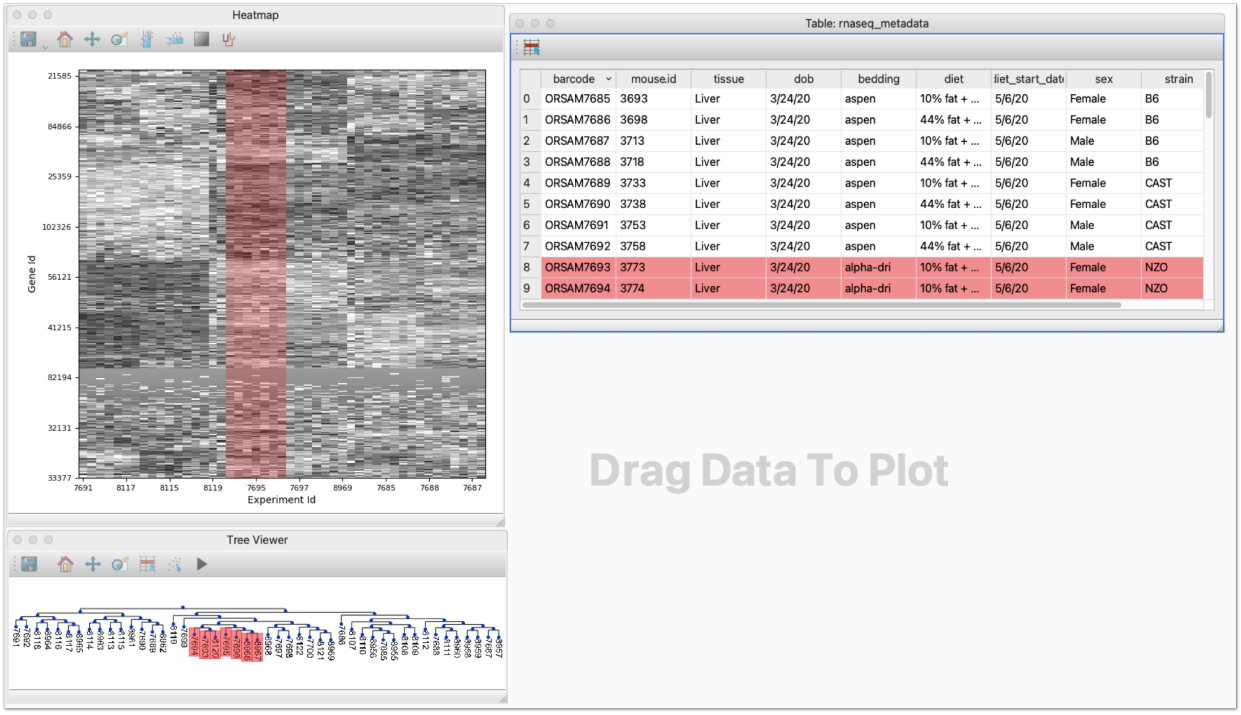


Navigation around the viewer uses the standard matplotlib options for zooming and panning, except that the y-axis tries to adjust to show all the data. Panning+zooming pauses recalculation of the track data layers, to keep that action reasonably smooth. There's an option to right click on a subset in the layer widget, and select "zoom to layer" to update the viewport to the subset's extent.

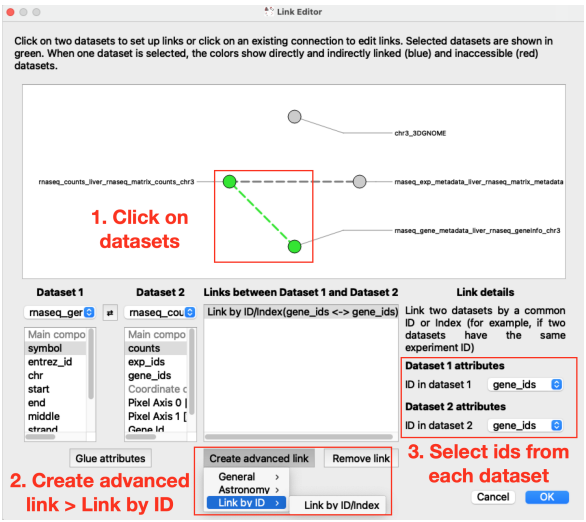


Heatmap Viewer

The Heatmap Viewer in glue genes is built on top of the standard glue 2D Image Viewer, which allows it to remain performant for large data matrices by showing downsampled versions at different resolutions.

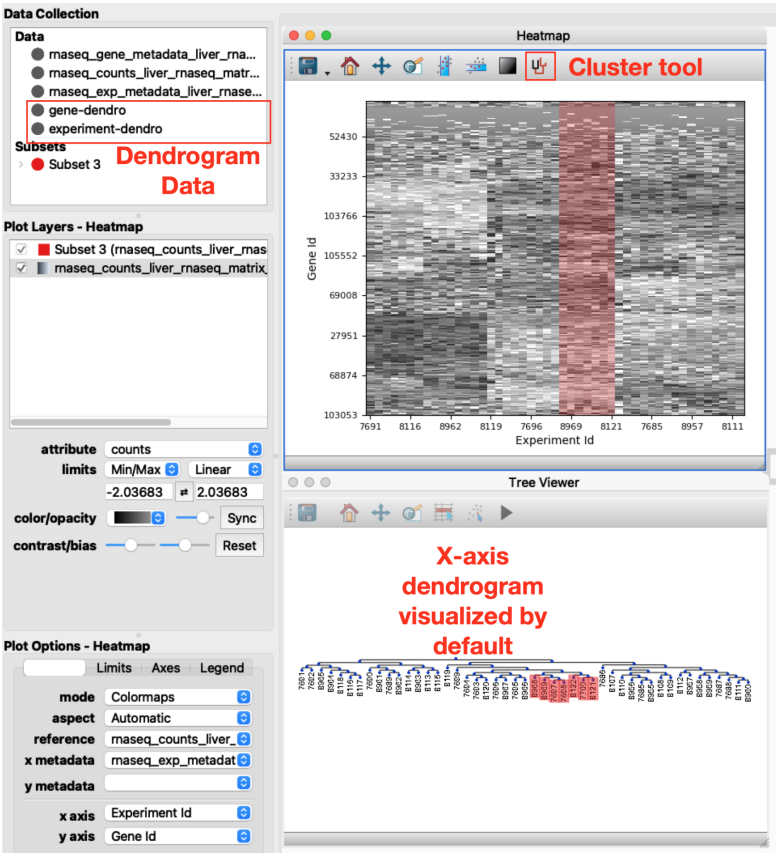


You will often want to link data in the Heatmap Viewer to other metadata about the rows or columns. For instance, if the Heatmap Viewer is showing gene expression versus experimental factors, you could have a separate dataset with the experimental metadata or additional information about the genes. These datasets need to be Linked By ID in glue. Linking By ID (also called join_on_key) is different than traditional gluing/linking of attributes in glue; rather than defining a mathematical relationship between components, Linking By ID says that these elements are the same entities across two different glue data objects, which means that subsets defined on one dataset can always propagate into subsets on the Linked-By-ID dataset.



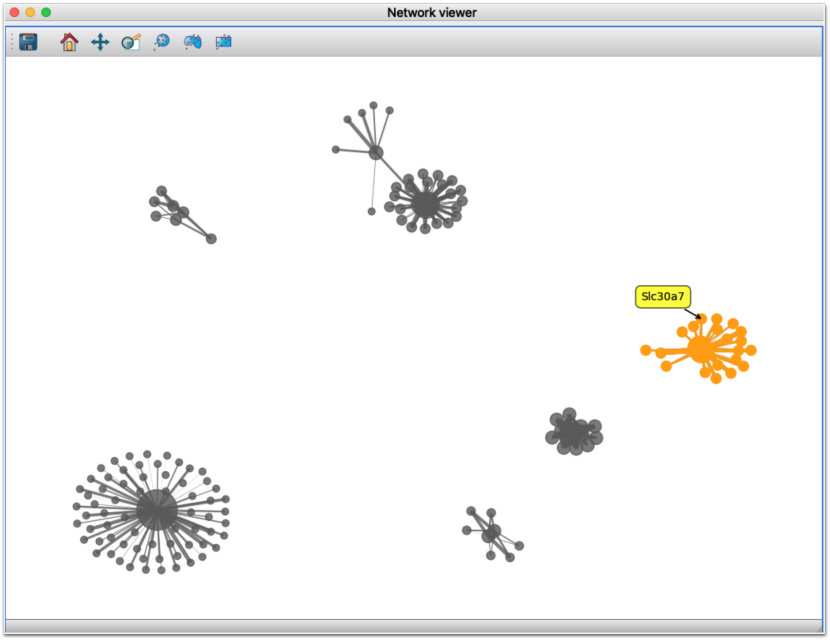
The link manager in glue genes can be used to define Linked By ID type relationships.

The Heatmap Viewer also includes a Cluster Tool to apply hierarchical clustering to the data matrix. This tool also serves to illustrate the process by which additional analysis tools can be integrated into glue. The Cluster Tool applies this clustering and creates two new datasets describing the clustering results as dendrograms. These dendrograms are Linked-By-ID to the original dataset and, by default, the dendrogram over the columns is visualized in the Dendrogram Viewer, facilitating another way to create and modify subsets.



Network Viewer

The Network Viewer visualizes data loaded in by the Peak Correlations (or some other dataset with the same attribute names). Each row in the dataset describes an edge between two nodes. Node names are visible in the viewer by hovering over a node. Subsets can be defined in Network Viewer with standard selection tools or in other viewers of this (or a linked) dataset. Defining new subsets can lead to the main data layer becoming invisible; in this case it is best to create a new viewer.



Dendrogram/Tree Viewer

The Tree Viewer (also known as the Dendrogram Viewer) displays trees of linked nodes. In glue genes this is currently only used as part of the Cluster Tool in the Heatmap Viewer. There are custom display options for this Viewer, and selections can be made either by clicking on specific nodes (which, by default, includes all child nodes) or by brushing across the dendrogram with a line (which selects all nodes where it intersects the connecting branch). Currently the Dendrogram Viewer does not support displaying additional text beyond just the ID in the dendrogram.

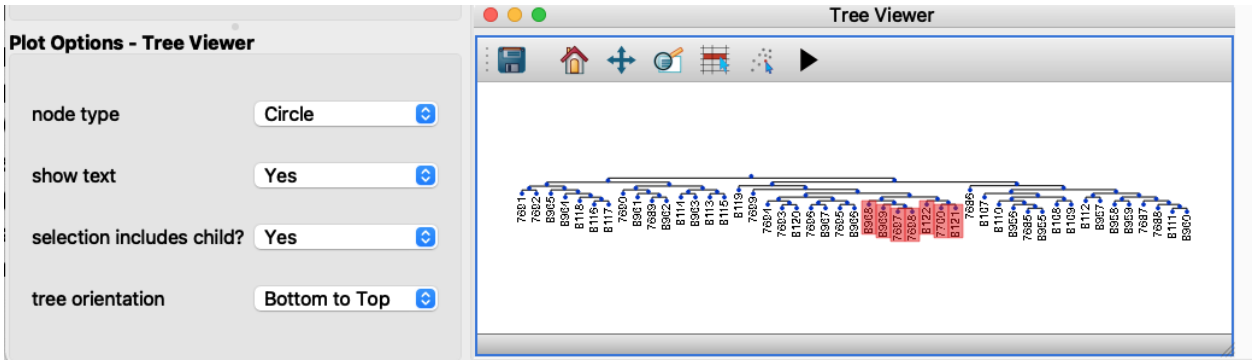


Table Viewer

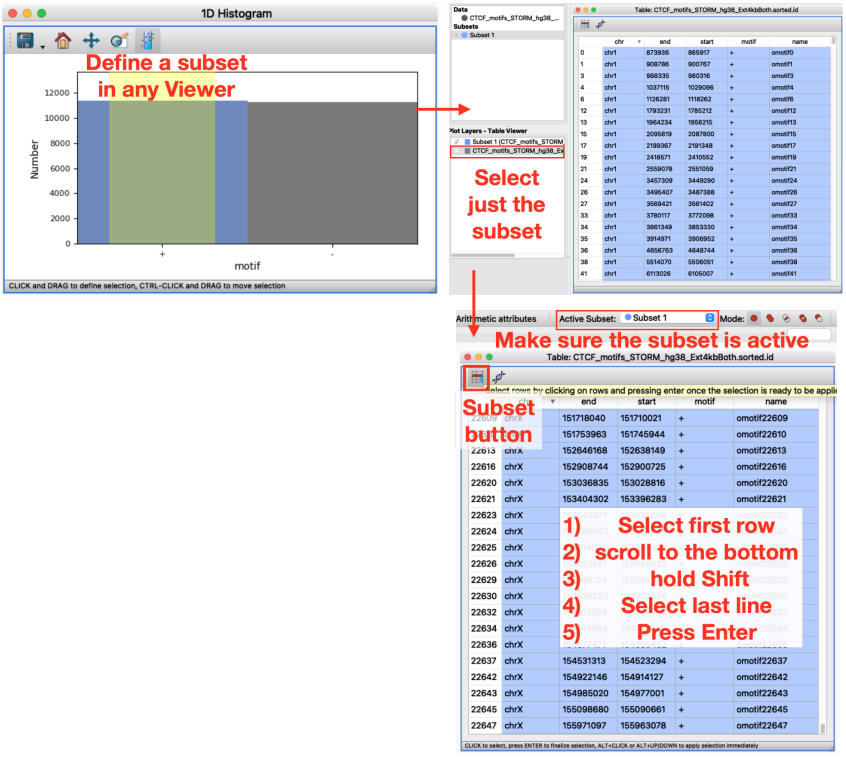
The standard glue Table Viewer contains a few new features in glue genes.

Putting Regions-on-Interest on the Genome

Behind the scenes, a subset defined in the Table Viewer (an ElementSubsetState) knows how to convert itself to GenomeRangeSubset that can be displayed on the Genome Track Viewer. What this means is that subsets defined using the Table Viewer, and ONLY these subsets are sort of dual-natured, existing as both a set of discrete elements in a table AND as a definition of (potentially discontinuous) ranges on the genome. This dual-nature is facilitated by the core assumption that chr/start/end attributes are always present for a dataset that defines genome regions-of-interest.

While it is convenient to have these dual-natured subsets, it is also possible to accidentally break them. For instance, in the following example we are looking at a BED file that defines regions-of-interest on the genome (CTCF motifs). If we define a subset of + motifs using a 1D Histogram Viewer, the subset was not directly created out of the Table Viewer and thus does not have access to the critical chr/end/start information necessary to put this subset on the Genome Track Viewer.

The workflow in this case is to define the subset in whatever viewer you want and then choose to display only that subset in the Table Viewer and use the Table Viewer Subset Creation tool to recreate that subset. Remember to make sure that the Subset you want to define shows up as the “Active Subset” in the menu bar, and that you can select all the rows in the Table Viewer quickly by selecting the first row, scrolling to the bottom of the table and holding down Shift before clicking on the last row. You must then press Enter to finalize the redefinition of the subset.



Future versions of glue genes will probably provide an updated and/or streamlined version of this process.

Interpolating Data onto a 3D Model

The 3D models of chromosome structure supported by glue genes describe an x,y,z position (in arbitrary units) for a fixed number of vertices that are equally spaced along the chromosome. In order to place genome features on this model, we must estimate an interpolating function that gives x,y,z coordinates as a function of genome position. The creation and storage of this interpolating function is done at data load time for the 3D model data.

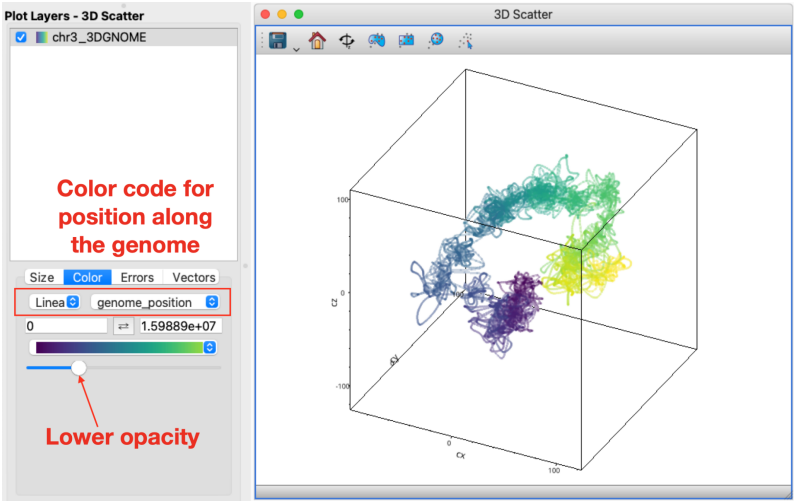
From the Table Viewer, this special button ( ) brings up a dialog box asking you to select the target 3D model dataset, the x,y,z attributes for this dataset and the column in your existing table that gives the genome positions of interest. The dialog attempts to display only relevant 3D datasets as possible targets. From these choices, new x,y,z attributes are added to the dataset in the table and these attributes are linked to the x,y,z attributes in the target 3D model dataset so that the table dataset may be added to the 3D figure.

Currently glue genes only supports the creation of point-like genome features in this fashion. The start or end position is normally fairly equivalent at the scales in question, although for larger genome regions-of-interest it may be worthwhile to define a new attribute for the midpoint of the region.

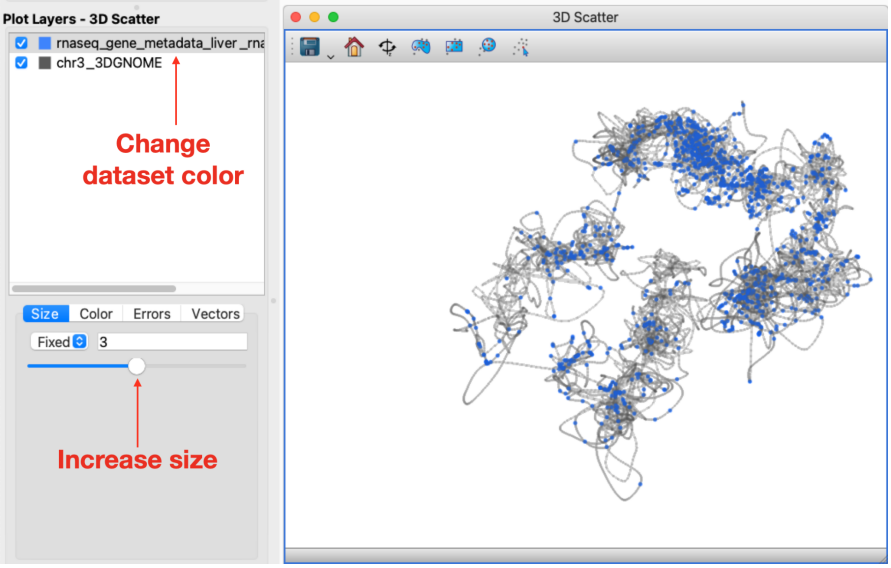
3D Scatter Viewer

The 3D Scatter Viewer present in glue genes is a lightly modified version of the default vispy 3D scatter viewer, which continues to function as normal. When used to plot a Data object (or subset) read in by the 3D GNOME Data Loader, the viewer connects the fixed-resolution points defining the 3D structure of the chromosome with line segments. Both the vertices and the line segments are linked to the same opacity and color selection in the UI. Selections in the 3D Scatter Viewer propagate to datasets displayed in the Genome Tracker Viewer and to genome regions-of-interest in explicitly linked BED-like datasets.

When displaying just the 3D chromosome structure, it is nice to color code the lines and vertices by position along the genome and to lower the opacity of the lines/points a bit, as shown here:



When displaying other datasets on the genome, it is normally necessary to alter some parameters to get a good visualization. Since full datasets are gray by default, it is helpful to change the color of one of the datasets. In addition, it is helpful to increase the size of the markers for the datasets you want to display on the genome.



When displaying multiple *continuous* regions (visualized with lines connecting vertices), it is helpful to have a low opacity on the layers and is normally best to drag subsets to be the lowest layer in the plot layers.

Other Features

Behind-the-Scenes Linking

In regular glue, all links between different data sets have to be made explicitly. In glue genes, it is assumed that all data-sets are defined on the same genome, which streamlines analysis where this assumption is correct. This is not a correct assumption if, for instance, you bring mouse and human data together into a single glue session and it is up to the user to determine if a particular dataset makes sense to display in a particular Viewer.

Custom Data Objects

Many genomics datasets use slightly non-standard glue data objects.

The most unusual are the data objects created by the Bedgraph and BEDPE data loaders and utilized by the Genome Track Viewer. These data objects only have an empty placeholder attribute/component, and define custom functions to retrieve data from disk in the context of the Genome Track Viewer.

The data objects created by using the 3D GNOME data loader are standard glue data objects but have specific metadata describing the distance on the genome between the fixed vertices that describe the model (*genome_stepsize*) and an interpolating function (*interp_function*) that can be used to place other datasets into the x,y,z coordinates of this 3D model. The custom 3D viewer and Table Viewer use the presence of this metadata to identify 3D model datasets and perform various functions on them.

Experimental data arrays that display in the Heatmap Viewer use two slightly unusual facets of glue datasets. First, because the Heatmap Viewer is based on the base glue Image Viewer, we define a Coordinate System for the data matrix in the Heatmap Viewer that simply maps each row and column to the appropriate label. Second, we store full 2D arrays for the quantities displayed on the rows and the columns alongside whatever value is being represented in the heatmap (e.g. counts). This would normally be represented as a Coordinate Component derived from the Coordinate System, but Coordinate Components have to be continuous transformations in glue, and the 1:1 mapping is not continuous. This has the side effect that subsets defined on the Heatmap Viewer are defined on all linked datasets as well. For instance, in an array of gene expression versus experimental factors, selecting a range of experimental factors will define a valid subset over the genes (the subset encompassing all genes).

Caveats and Known Limitations

glue genes is prototype quality code, and is still under active development. This section contains a summary of some of the most important caveats and limitations.

It is helpful to load glue with the -v flag: `glue -v` to see errors/warnings at the command line.

If a viewer starts misbehaving, it may be necessary to close the viewer and recreate it.

The Network Viewer has buggy behavior wherein defining a new subset in an existing viewer causes the main underlying dataset to not be shown. The workaround is to close that viewer and start a new one.

Many viewers and options have only been robustly tested on datasets for a single chromosome and may not work as expected across multiple chromosomes.

Very large genome region-of-interest files can cause performance issues. The best workaround is to load in only the required subset of a datafile for analysis.

Under some circumstances datasets that are Linked By ID are not unlinked when the link is removed in the UI Link Manager. This can be fixed at the terminal by clearing the private dictionary `_key_joins` for a given dataset.

glue session files (.glu) are not guaranteed to work within glue genes.

Developing for glue genes

Code Organization

The main glue genes package is just a meta-package that installs all the component packages necessary for glue genes to work. Currently these packages are:

- glue-genomics-viewers: Custom Viewers (Genome Track Viewer, Heatmap Viewer, and Network Viewer), as well as the special data classes and subset types for dealing with Genome Track Viewer data.
- glue-genomics-data: Custom Data Loaders for genomics data
- glue-tree-viewer: The Tree/Dendrogram Viewer
- glue-vispy-viewers: A custom version of the vispy viewers (which were already a separate plug-in package for glue)
- glue-core: The core glue package with modifications to support glue genes. Major changes include customizations to the standard Table Viewer, ElementSubsetState support for GenomeRangeSubsets, and support for Linking-by-ID in the UI.

The best reference for developing glue genes is the core glue documentation. The Introduction to customizing/extending glue is a good place to start. Other helpful part of the documentation are Working with Data objects, The selection/subset framework, The communication framework, The linking framework, and Code organization.