GenomeCruzer Clustering Guide

Clustering import

This section explains how to import row and/or column clustering definition into an existing database, to be applied to the database scenes.

Clustering import format

The format to import a clustering definition into an existing database is the same that was used in GenomeCruzer v1.0 scene format.

This file format is a text, tab separated file defined as follows:

1. First header row has *ROWS* or *COLUMS* keyword, followed by the number of clustering levels. For examples, to indicate a row clustering with 3 levels, the first row would be:
   ROWS 3

Second header row has a fixed list of descriptors, with as many *CLUS_ID_n* as the number of clustering levels specified in the first header row. For examples:
UNIQUE_ID CLUS_ID_1 CLUS_ID_2 CLUS_ID_3

2. The following rows describe the clustering. Each line places one gene or sample (identified respectively by EntrezId o sample UniqueId) in the cluster specified by name at each level.

For examples, to place the gene with EntrezId *26155* inside cluster *gene_clust_1* at first level, and in cluster *gene_clust_1.1* at the second level, the row would be:
26155 gene_clust_1 gene_clust_1.1

Any cluster specified at the first level will be a "root" cluster. Any cluster specified at subsequent levels will be a child of the cluster in the previous level.

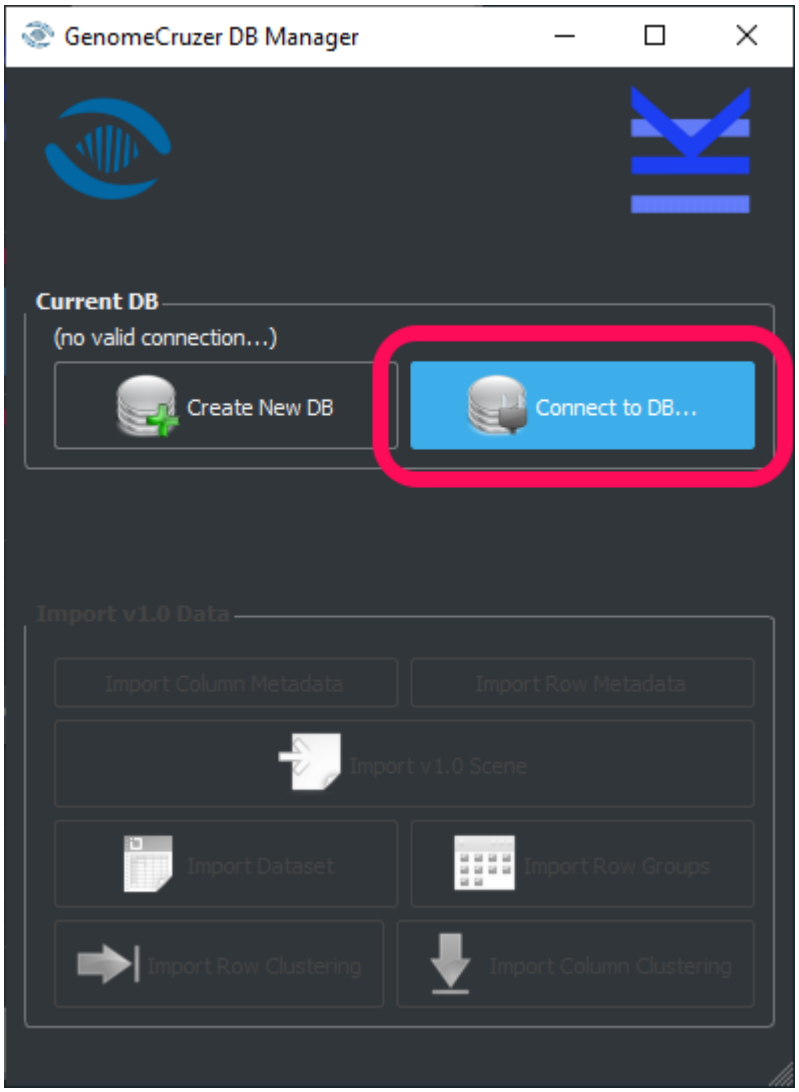
Here follow a sample of a complete row clustering file, where *gene_clust_1.1* and *gene_clust_1.2* are second level clusters, children of the root cluster *gene_clust_1.*

```
ROWS 2
UNIQUE_ID CLUS_ID_1 CLUS_ID_2
26155 gene_clust_1 gene_clust_1.1
339451 gene_clust_1 gene_clust_1.1
84069 gene_clust_1 gene_clust_1.1
57801 gene_clust_1 gene_clust_1.2
9636 gene_clust_1 gene_clust_1.2
375790 gene_clust_1 gene_clust_1.2
401934 gene_clust_1 gene_clust_1.2
```

GenomeCruzer DbManager tool

The tool to import the clustering into an existing database is the DB Manager.
On a Windows installation, a shortcut to tool is available from the Windows start menu.

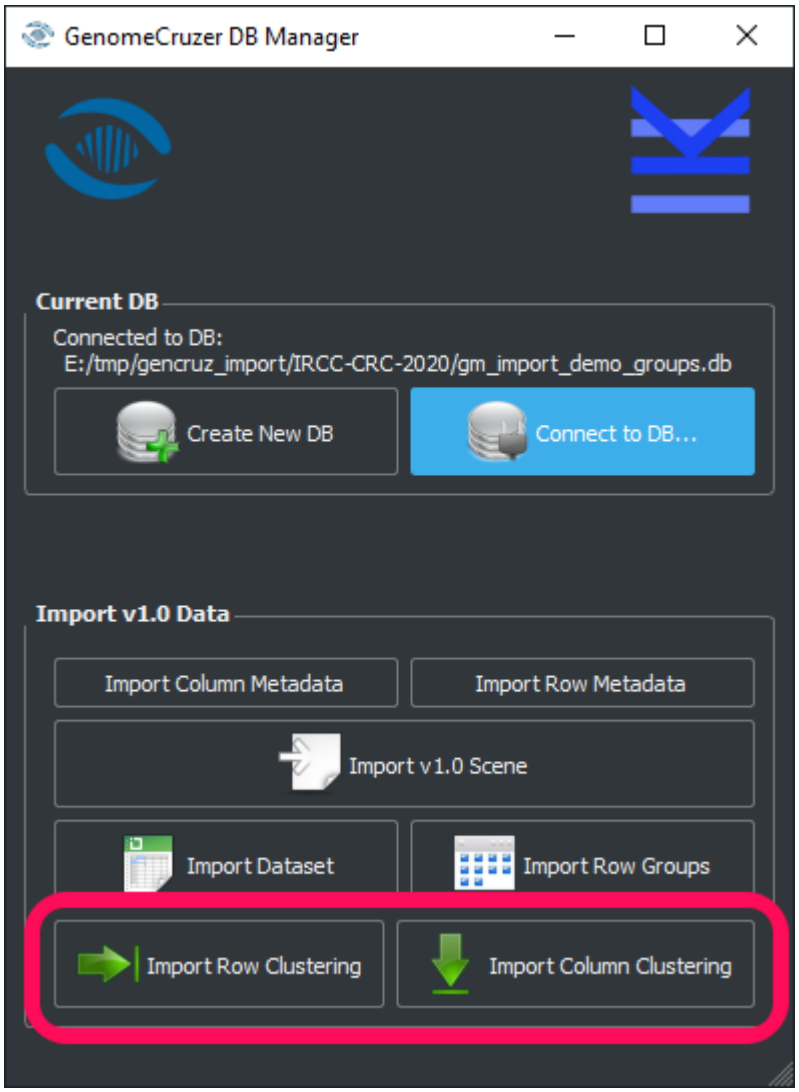
Once opened, click on "Connect to DB" button and select an existing GenomeCruzer database (see figure below):



Once the database is loaded, the other buttons become available.

Press the buttons to import either a Row or Column clustering and select the files containing the clustering definition. (see image below).

A popup message will notify if the operation succeeds or if something goes wrong.



Limitations

DB Manager

Currently the DB Manager only works for importing clustering defined in the v1.0 format into an existing database.

Clustering definition

At the moment, an externally defined clustering must address only genes and samples **present in the scene** it will be applied to.

For examples let us consider a database in which 2 datasets are defined, and in which the first dataset has 20 samples and the second one has 16 samples in common with the first.

If the imported clustering addresses all the 20 samples, the user will be able to apply it only to the scenes where all the 20 samples are displayed (for example a scene showing only the first dataset, or a scene showing the “union” of the 2 datasets).

If the user tries to apply the clustering to a scene showing for instance only the second dataset (where only 16 samples are available), the Cruzer will display an error message and the clustering won't be applied.

Clustering Layout

The walls representing a hierarchical clustering (which are the genes and samples walls in the Analysis mode) may organize and display their contents in several ways and can be customized with a layout assigned to the clustering itself in the database.

Default Layout: Packing

If no layout is assigned, the default "Packing" placement is used: this visualization type recursively packs all children of every group in a rectangle, attempting to create at every level a shape as close as possible to a square.

The following image shows an example of the packing algorithm used for the genes wall in Analysis mode: genes are packed in sub-bands, sub-bands packed in bands, and so forth…

Custom Layout

A layout may be assigned to any genes and samples clustering in the database, to customize how this is visualized by the corresponding hierarchical wall.

The Layout is a json formatted string which allows to specify how to display each level of the clustering. The following example represent a Layout specification for a 2-levels clustering.

```
[
   {
      "placement": 1,
      "padding": false,
      "center": true,
      "prune": true,
      "elemsPerRow": -1,
      "invert": false
   },
   {
      "placement": 2,
      "padding": true,
      "center": true,
      "prune": false,
      "elemsPerRow": -1,
      "invert": false
   }
]
```

If the target clustering has more levels than those specified by the json layout, default values are used for the remaining levels.

In the following paragraphs the layout properties available for each clustering level are explained in detail.

placement

Integer value in the range [0 - 4] to specify the algorithm used to place the children of the current level. The possible types are:

- **0 – Packing**: explained in the section above.
- **1 – Column**: place children in a single column.
- **2 – Row**: place children in a single row.
- **3 – Fixed Number**: place children in a rectangle, ordered per rows from left to right and from bottom to top. Each row has a fixed number of children.
- **4 – Snake**: same as Fixed Number, but children are ordered following a snake pattern from bottom to top.

padding

Boolean value used only by Column and Row placements. If the current level has 2 children with different size, add a padding to resize the container as if the 2 children had the same dimensions. This might be useful to center rows or column on their midpoint (see *center* property below).

center

Boolean value used only by Column and Row placements. If true, center the children along the placement direction. For instance, if using Column placement, place every child at the center of the column rather than at the leftmost position.

prune

Boolean value. If true, prune from the visualization the cluster nodes which contains no leaves, where leaves are not the direct children of a level, but the leaves at the last level of the graph, which is genes for the genes graph and samples for the samples graph.

elemsPerRow

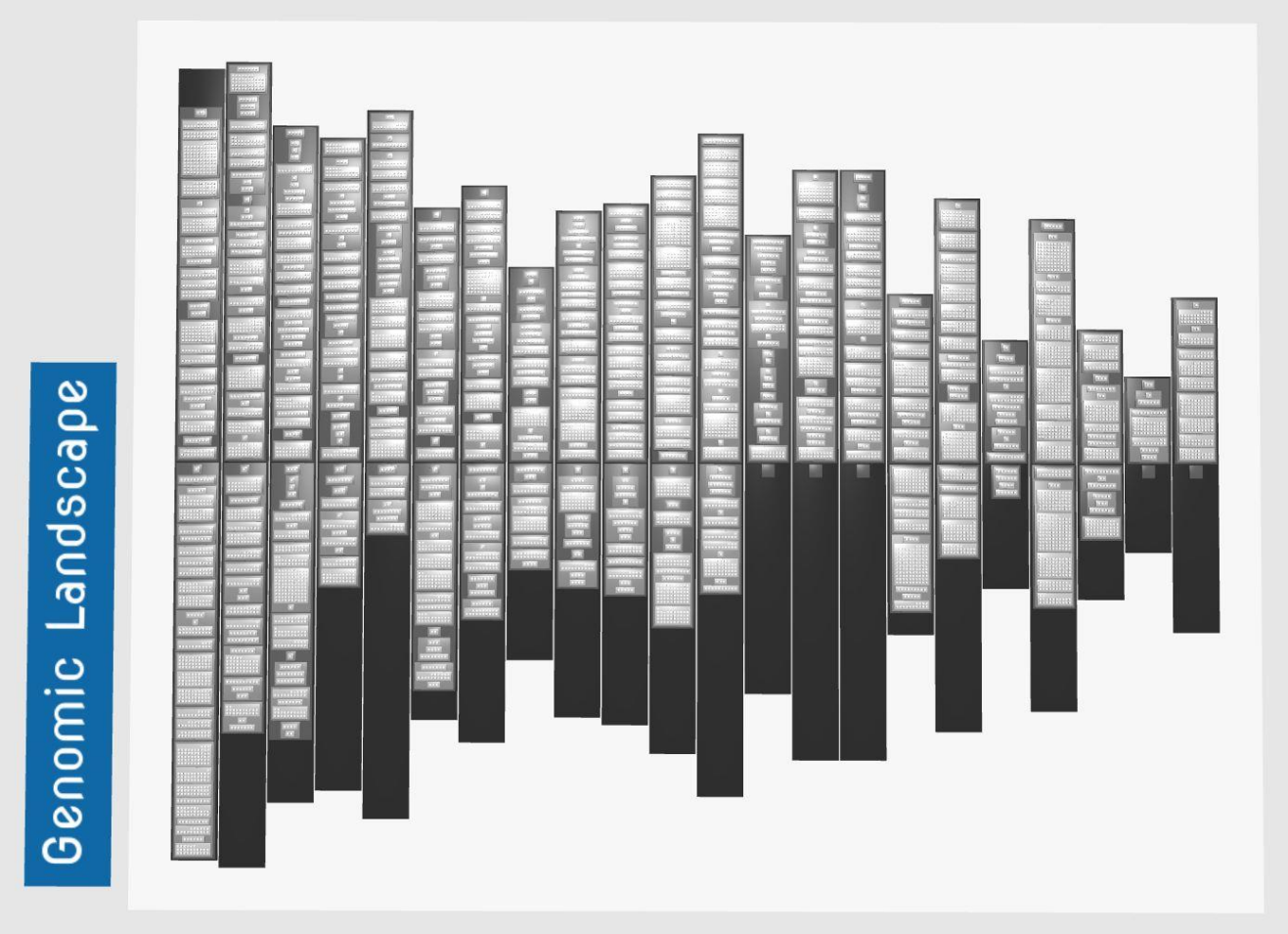
Integer value used by Fixed Number and Snake placements. Specifies the number of children to place in each row.

invert

Boolean value. Invert the order of children with respect of how they are found in the data graph.

The following example shows the use of a custom layout applied to the same data displayed above with the default packing algorithm.

```
[
   {
      "placement": 1,
      "padding": false,
      "center": true,
      "prune": true,
      "elemsPerRow": -1,
      "invert": false
   },
   {
      "placement": 2,
      "padding": true,
      "center": true,
      "prune": false,
      "elemsPerRow": -1,
      "invert": false
   },
   {
      "placement": 2,
      "padding": false,
      "center": true,
      "prune": true,
      "elemsPerRow": -1,
      "invert": true
   },
   {
      "placement": 2,
      "padding": false,
      "center": true,
      "prune": true,
      "elemsPerRow": 7,
      "invert": true
   },
   {
      "placement": 4,
      "padding": false,
      "center": false,
      "prune": true,
      "elemsPerRow": 10,
      "invert": true
   }
]
```


Genomic Landscape