1. **Convert mapped Hi-C reads to hic format file**
2. Purpose: to create a binary hic format file containing contact matrices at different resolutions and normalized by different methods from a text file describing mapped Hi-C reads
3. Input file format: a sample file is executable/sample\_data/GSM1551688\_HIC143\_merged\_nodups.zip (unzip it before use)

The below description is from https://github.com/theaidenlab/juicer/wiki/Pre#file-format.

* + **Medium format**

A whitespace separated file that contains, on each line

<readname> <str1> <chr1> <pos1> <frag1> <str2> <chr2> <pos2> <frag2> <mapq1> <mapq2>

* str = strand (0 for forward, anything else for reverse)
* chr = chromosome (must be a chromosome in the genome)
* pos = position
* frag = restriction site fragment
* mapq = mapping quality score

If not using the restriction site file option, frag will be ignored, but please see above note on dummy values. If not using mapping quality filter, mapq will be ignored. readname and strand are also not currently stored within .hic files.

* + **Short format**

A whitespace separated file that contains, on each line

<str1> <chr1> <pos1> <frag1> <str2> <chr2> <pos2> <frag2>

* str = strand (0 for forward, anything else for reverse)
* chr = chromosome (must be a chromosome in the genome)
* pos = position
* frag = restriction site fragment

If not using the restriction site file option, frag will be ignored, but please see above note on dummy values. readname and strand are also not currently stored within *.hic* files.

* + **Short with score format**

This format is useful for reading in already processed files, e.g. those that have been already binned and/or normalized; this format can be easily used in conjunction with the -r flag to create a *.hic*file that contains a single resolution.

A whitespace separated file that contains, on each line

<str1> <chr1> <pos1> <frag1> <str2> <chr2> <pos2> <frag2> <score>

* str = strand (0 for forward, anything else for reverse)
* chr = chromosome (must be a chromosome in the genome)
* pos = position
* frag = restriction site fragment
* score = the score imputed to this read

If not using the restriction site file option, frag will be ignored, but please see above note on dummy values. readname and strand are also not currently stored within *.hic* files.

* + **Long format**

The long format is used by [Juicer](https://github.com/theaidenlab/juicer) and takes in directly the *merged\_nodups.txt* file.

A whitespace separated file that contains, on each line

<str1> <chr1> <pos1> <frag1> <str2> <chr2> <pos2> <frag2> <mapq1> <cigar1> <sequence1> <mapq2> <cigar2> <sequence2> <readname1> <readname2>

* str = strand (0 for forward, anything else for reverse)
* chr = chromosome (must be a chromosome in the genome)
* pos = position
* frag = restriction site fragment
* mapq = mapping quality score
* cigar = cigar string as reported by aligner
* sequence = DNA sequence

If not using the restriction site file option, frag will be ignored, but please see above note on dummy values. If not using mapping quality filter, mapq will be ignored. readname, strand, cigar, and sequence are also not currently stored within *.hic* files.

* + **4DN DCIC format**

A file that follows the 4DN DCIC format specification ([the 4DN DCIC format specification](https://github.com/4dn-dcic/pairix/blob/master/pairs_format_specification.md)).  
See the link for more information. Briefly, there should be a header with the first seven columns reserved:

## pairs format v1.0

#columns: readID chr1 position1 chr2 position2 strand1 strand2

If the columns line contains (in any field after field 7) both frag1 and frag2, those will also be read in; otherwise Pre will set frag1=0 and frag2=1, so that no reads are discarded. Other fields are ignored.

1. Output: a .hic file containing contact matrices
2. Running: invoke the function from the menu toolbar: *2D-Functions/Convert to HiC*

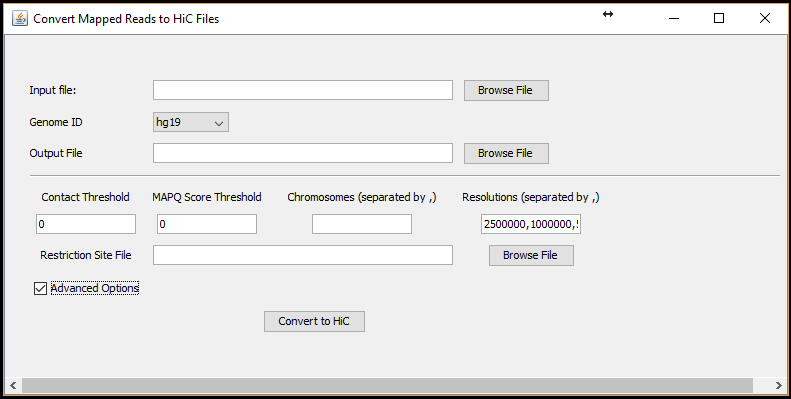


Figure 1 Convert to HiC function

The required fields are: Input file, Genome ID and Output File

Advanced and optional options are:

* + Contact Threshold (default: 0): number of interaction threshold for contacts to be used in creating contact matrices.
  + MAPQ Score Threshold (default: 0): mapping quality score threshold for reads to be considered in creating contact matrices.
  + Chromosomes (default: blank – all chromosomes): chromosomes for which their contact matrices to be created. When left blank, all chromosomes will be considered. Chromosomes must be separated by a comma (,).
  + Resolutions (default: 2500000,1000000,500000,250000,100000,50000,25000,10000,5000): list of resolutions of contact matrices to be created. Resolutions are separated by a comma (,).
  + Restriction Site File (default: blank): each line starts with a chromosome number followed by positions of restriction sites on that chromosome, in numeric order, and ending with the size of the chromosome. When provided, 8 additional fragment-delimited resolutions are added: 500f, 250f, 100f, 50f, 20f, 5f, 2f, 1f

1. **Extract contact matrices from a hic format**
   1. Purpose: to extract a contact matrix from a hic format into a sparse matrix format in a text file
   2. Input: a local path to a hic format or an online link to a hic format. A link to a hic file: https://www.encodeproject.org/files/ENCFF219YOB/@@download/ENCFF219YOB.hic
   3. Output: a contact matrix in sparse matrix format (each line represent a contact by three numbers separated by whitespaces: position1 postion2 interaction\_frequency)
   4. Running: invoke the function from the menu toolbar: 2D-Functions/Extract HiC

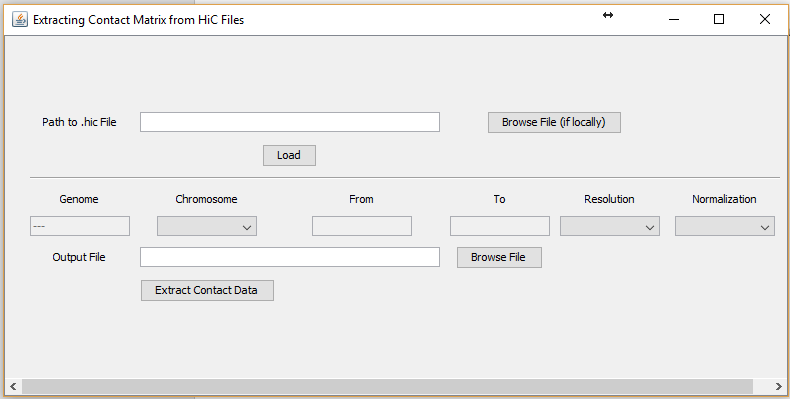
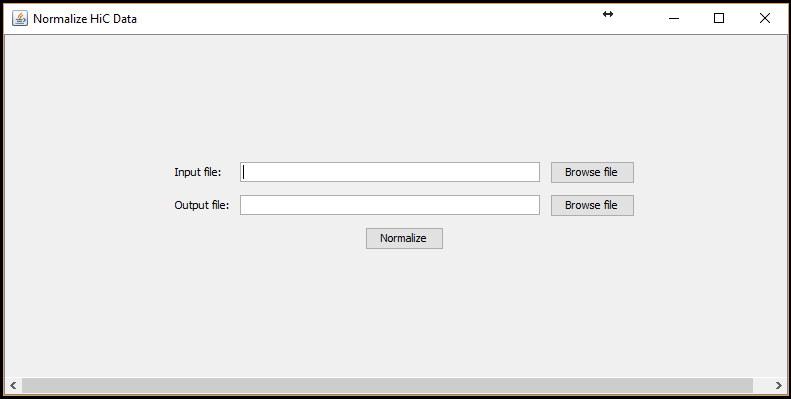


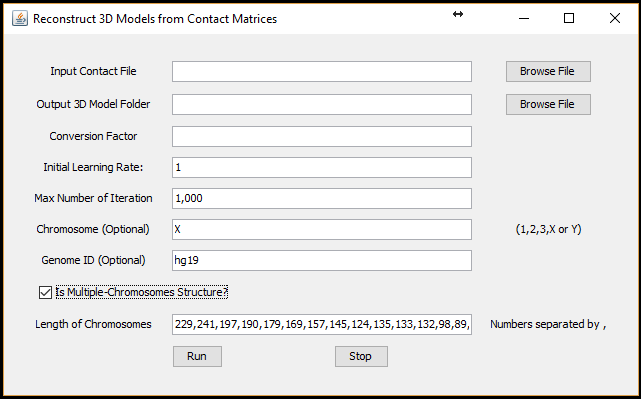
Figure 2 Extract Contact Matrices from a hic file

* Path to .hic File (required): an online link or local path to a hic format file
* Load: clicking this button to fetch information from the header of the hic file.
* Genome: genome version of the hic file
* Chromosomes: list of chromosomes in the hic file to be chosen for contact matrices
* From and To: to specify a fragment to extract its contact matrix
* Resolution: list of resolutions of contact matrices in the hic file
* Normalization: list of normalization methods used to normalize contact matrices

1. **Normalize HiC contact matrices**
   1. Purpose: to normalize contact matrices in sparse matrix format.
   2. Input: a contact matrix in sparse matrix format (each line represents a contact by three numbers separated by whitespaces: position1 position2 interaction\_frequency)
   3. Output: a normalized contact matrix in sparse matrix format. The matrix is normalized by the Iterative Correction and Eigenvector decomposition (ICE) method
   4. Running: invoke the function from the menu toolbar: 2D-Functions/Normalized HiC Data



1. **3D model reconstruction by LorDG**
   1. Purpose: to build 3D chromosomes and genome models
   2. Input: a contact matrix in sparse matrix format
   3. Output: 3D models in .gss format file
   4. Running: invoke the function from the menu toolbar: 3D-Functions/LorDG-3D Modeller



* Conversion Factor (default: 1.0) : in the formula , where is interaction frequency between . When the field is left blank, the program will search for the best value in the range [0.1-3.0] with a step size of 0.1. Users can also specify a range to search by put 2 numbers separated by a hyphen (e.g. 0.5-1.0). During the searching, the right-top corner of the main screen displays information about the current value being tested.
* Initial Learning Rate (default: 1.0): higher learning rate can speed up the reconstruction process but can cause the process to fail as well.
* Max number of Iteration: maximum number of iterations for the optimization.
* Chromosome: chromosome name of the contact matrix in the input. If the input contains contact matrix of the whole genome, please put the list of chromosome names (separated by commas).
* Genome ID: genome version of the contact matrix in the input.
* Is Multiple-Chromosomes Structure? : if the input contains both inter-and intra-chromosomal contacts data, this checkbox should be checked.
* Length of Chromosomes: this field contains a list of lengths of chromosomes in increasing order of chromosome names and separated by commas, if “Is Multiple-Chromosomes Structure” is checked. Please note that these lengths should not contain omitted regions (e.g. centromeres) in the input of chromosomes.
* Run: to start the reconstruction process. The main screen displays how models are being formed from initially random models. The information about the reconstruction is displayed in the top-right corner of the main screen. The conversion factor is being used to build model and the current value of the objective function (higher is better). After the reconstruction is finished, the score of the model is displayed in the top-right corner of the main screen (the lower the value is, the better the model is).
* Stop: during the reconstruction, if this button is pressed, the program will stop and output the currently best structure. If the program is searching for the best conversion factor, it will stop the searching and use the best-found conversion factor to build models.

1. **Chromatin loop identification**
   1. Purpose: to identify chromatin loop in 3D models
   2. Input: a 3D model
   3. Output: a list of chromatin loops in a bed format file and highlighted in the 3D model
   4. Running: invoke the function from the menu toolbar: 3D-Functions/Loop Detection

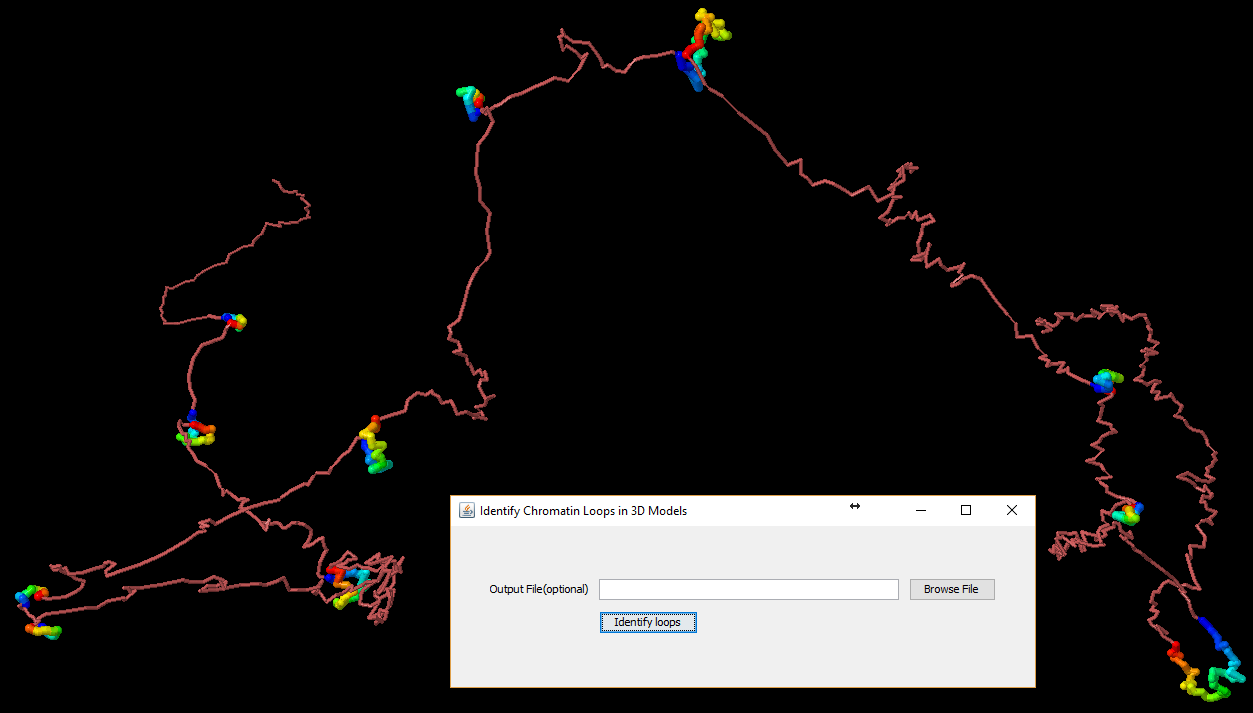


Figure 3 Chromatin loops

The function identifies chromatin loops and highlights them in the 3D model. The loops can also be outputted into a bed format file specified in the Output File field. The top-right corner of the main screen displays the number of chromatin loops identified.

Loops are colored in spectrum (from blue to red). To highlight loops better, color the model by a single color( right-click on the main screen, choose color/structure/chain)

1. **Model annotation**
   1. Purpose: annotate 3D models with genomic elements
   2. Input: a 3D model (e.g. in executable/sample\_data/models) and genomic elements in bed format files (e.g. in executable/track\_files)
   3. Output: 3D model is annotated with data from bed format files
   4. Running: invoke the function from the menu toolbar: 3D-Functions/Model Annotation

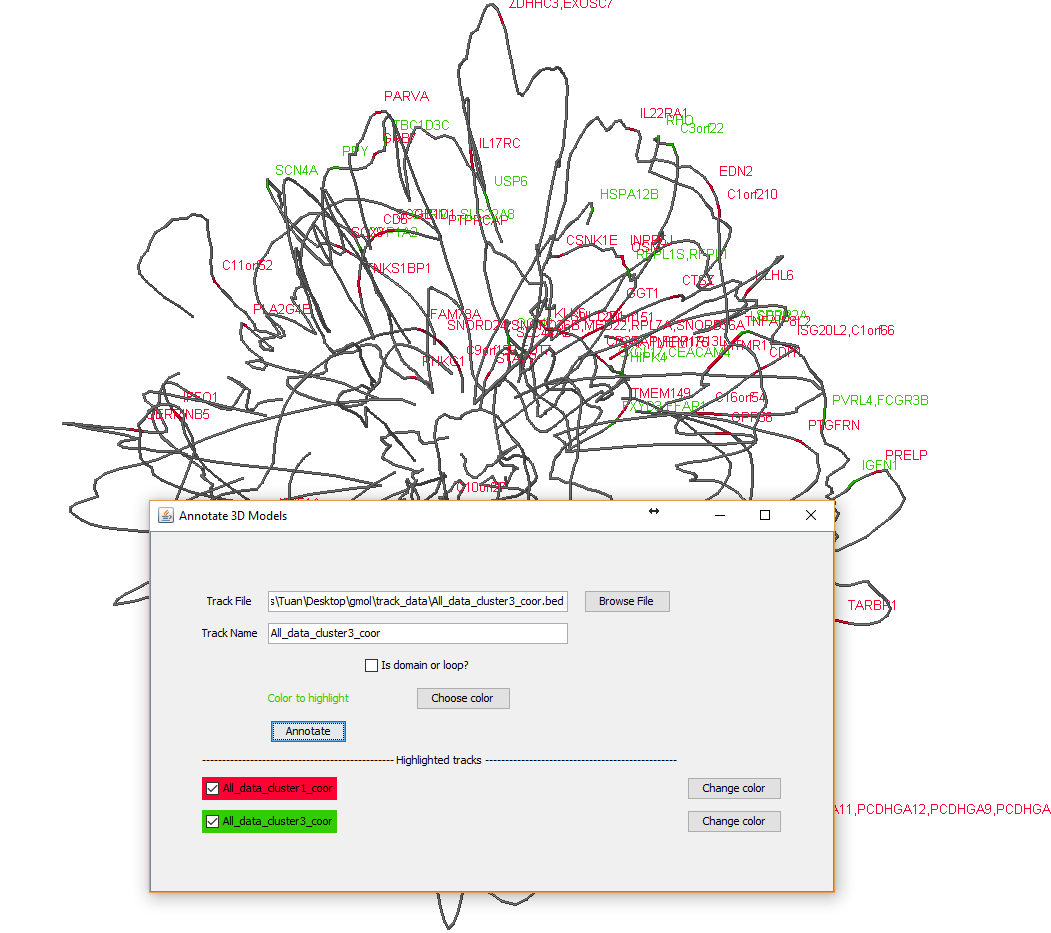


Figure 4 Function to annotate 3D models

To better highlight track data, change the color of the model to a sing color (right-click on the main screen, Color/Structure/Reset). The background can be changed to white to (Color/Background/White)

To get to

* Track file: a file in bed format (see executable/track\_files for example)
* Track name: a unique name associated with the above input file
* Choose color: to choose a unique color for the track data

Checking corresponding track names will display or hidden the content of tracks.

To get the genomic coordinate of a point, left-click or mouse-over to the point as shown in **Figure 5.**

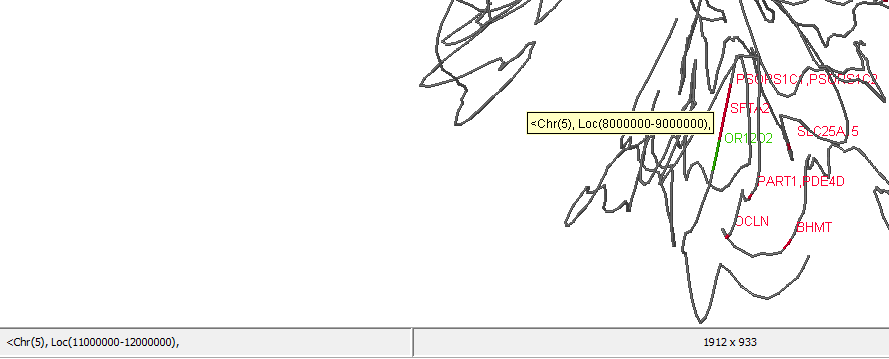


Figure 5 Coordinate of a point in the model