CHROMSTRUCT v4.3 folder: README

This folder contains 3 files:

- *ChromStruct_4.3.py* (Command-line version of CHROMSTRUCT 4.3)
- *Plot_Energy_Chain_2.0.py* (command-line code to display CHROMSTRUCT results)
- *README.pdf* (this file)

The code files are all self-contained and only need a Python interpreter to run.

ChromStruct_4.3.py

The code can be run from the interactive python dialog or from the console window, by invoking python *ChromStruct_4.3.py.*

The program will ask on the interactive python dialog to insert the names of following files:

- "*File name (.txt) = ?*" **Hi-C square contact frequency matrix** in txt format.
- *"RNAseq file(.txt) = ?"* (Optional) **RNA-seq data** in txt format: binary array at the same resolution and same dimension of the contact matrix. "1" if the bin is interested by expressed genes, "0" if the bin is not.
- *"ChIPseq file(.txt) = ?"* (Optional) **ChIP-seq data associated to H3K27me3 methylation** (or other histone modification associated to repression and compaction) in txt format: binary array at the same resolution and same dimension of the contact matrix. "1" if the bin is interested by H3K27me3 methylation (cut off 300), "0" if the bin is not.
- **CTCF contacts** (Optional) need to be introduced in the contact matrix as additional contacts with high frequency (set equal to the maximum frequency of the matrix by removing the main diagonal).

If the data file is filename.suffix the code produces a number of files with these names:

- **filename_<timestamp>_<level>_BlockSizes.txt**: a list with as many entries as blocks detected at resolution level <level>. <level> is coded as an integer from 0 to *number of detected levels – 1*. <timestamp>, a character string formatted as <yy-mm-dd-hhmm>, is referred to the date and time when the algorithm starts, and is used to identify the files coming from the same data file and the same run.
- **filename_<timestamp>_Log.txt**: a self-explanatory log file.
- **filename_<timestamp>_<level>_<block>_Energy.txt**: a real array with 3 columns and as many rows as accepted annealing updates of the configuration of block <block> (coded as an integer from 0 to *number of detected blocks at level* <level> *– 1*) at the resolution level <level>. The first real in each row is the data fit part of the score function, the second is the constraint part, and the third is the total score. If the related checkbox in the GUI is active, this is plotted as soon as the iteration at each block and level is complete. It can also be plotted by *Plot_Energy_Chain_1.0.py*.
- **filename_<timestamp>_<level>_<block>.txt**: a real array with 4 columns and as many rows as three times the number of beads in block <block> at level <level>. The first of each three rows contains the coordinates (in nm) of the first endpoint of a bead; the second contains the

coordinated of the centroid, and the third contains the coordinates of the second endpoint. Each row is completed with the estimated size (in nm) of the related bead. If the related checkbox in the GUI is active, this structure is plotted as soon as block <block> at level <level> has been computed. It can also be plotted by *Plot_Energy_Chain_1.0.py*.

- **filename_<timestamp>_LastConf.txt**: a real array with the same format as filename_<timestamp>_<level>_<block>.txt, with the final 3D chain configuration. This is plotted at the end of the procedure. It can also be plotted by *Plot_Energy_Chain_1.0.py*.
- **filename_<timestamp>_DistMat.txt**: a real array with the mutual distances between bead centroids, computed from the final estimated structure in filename_<timestamp>_LastConf.txt.

The code also prints the logfile information in the console window (score values and related annealing temperature once in every 1000 cycles). To close the program after the final plot, close the plot window and then the graphical interface. To abort the program, press <ctrl>-c from the keyboard.

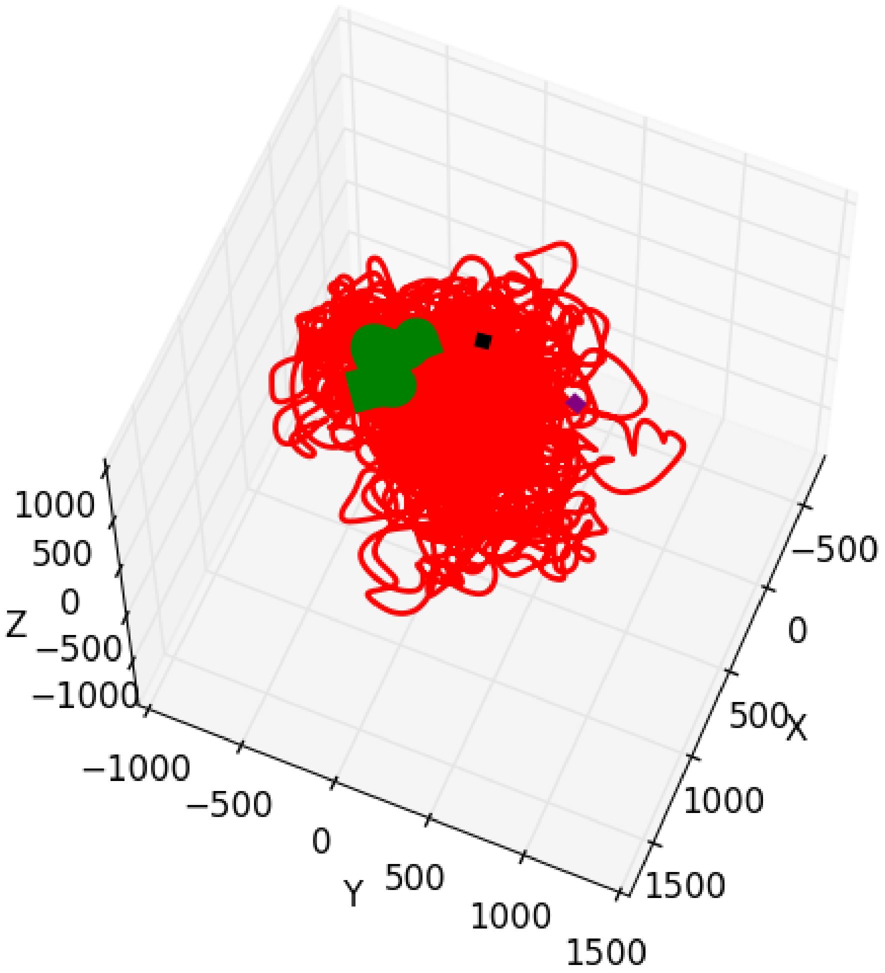
Plot_Energy_Chain_2.0.py

The code can be run from the interactive python dialog or from the console window, by invoking python *Plot_Energy_Chain_2.0.py.*

The program will ask on the interactive python dialog to insert the names of following files:

- "*Chain: file name = ?*" **filename_<timestamp>_LastConf.txt** produced by *ChromStruct_4.3.py*: the final configuration is represented by a red smooth curve linking the centroids of the bins (smoothing is obtained by cubic spline interpolation).
- *"Chain: centromeres file name = ?"* (Optional) ***filename_<timestamp>_centromere.txt*** *produced by ChromStruct_4.3.py*: the centromere region is emphasised in green.
- *"Chain: telomeres file name = ?"* (Optional) ***filename_<timestamp>_telomeres.txt*** *produced by ChromStruct_4.3.py*: the telomeric regions are emphasised in blue.
- Sphere envelops are set with default "not" (*ball='n')*. To display each bin as a sphere set *ball='y'* at line 108.

Output of Chromosome 16 at 25 kb resolution (GEO GM06990) produced by *Plot_Energy_Chain_2.0.py :* smooth red line represents final configuration produced by *ChromStruct_4.3.py,* centromere is emphasised in green.



References

[1] C. Caudai, E. Salerno, M. Zoppè, A. Tonazzini, "Inferring 3D chromatin structure using a multiscale approach based on quaternions", BMC Bioinformatics, Vol. 16, 234, 2015, DOI: 10.1186/s12859-015-0667-0.

[2] Caudai, C.; Salerno, E.; Zoppè, M.; Tonazzini, A. A statistical approach to infer 3D chromatin structure. In Mathematical Models in Biology, Springer International Publishing ed.; Zazzu, V: Switzerland, 2015; pp. 325 161–171.

[3] Caudai, C.; Salerno, E.; Zoppe, M.; Merelli, I.; Tonazzini, A. ChromStruct 4: A Python Code to Estimate the Chromatin Structure from Hi-C Data. IEEE/ACM Transactions on Computational Biology and Bioinformatics 2018, pp. 1–1. doi:10.1109/TCBB.2018.2838669.

[4] Caudai, C.; Salerno, E.; Zoppe, M.; Tonazzini, A. Estimation of the Spatial Chromatin Structure Based on a Multiresolution Bead-Chain Model. IEEE/ACM Transactions on Computational Biology and Bioinformatics 2019, 16, 550–559. doi:10.1109/TCBB.2018.2791439.

Previous versions of ChromStruct

CHROMSTRUCT v3.1 - Reconstruction of 3D chromatin structure from chromosome conformation capture data. E. Salerno, C. Caudai.
Software, Release 3.1, cnr.isti/2016-SW-031, 2016, DOI: 10.13140/RG.2.2.35785.13923.

CHROMSTRUCT v4.2 - Reconstruction of 3D chromatin structure from chromosome conformation capture data. E. Salerno, C. Caudai.
Software, 2018, CNR-ISTI, Pisa, 2018-388694 DOI: 10.13140/RG.2.2.26123.39208.