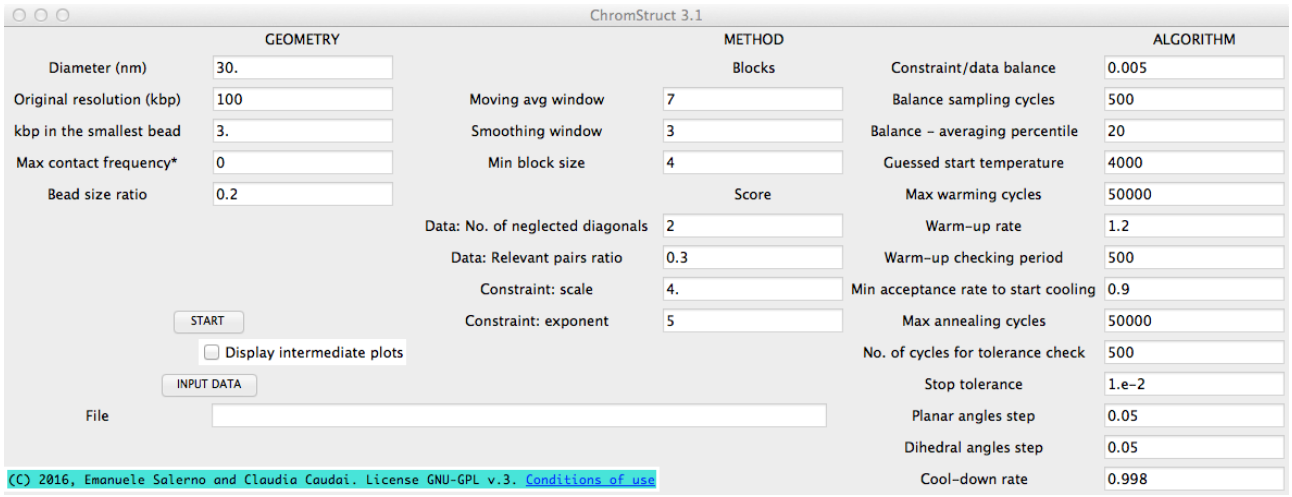
CHROMSTRUCT 3.1 folder: README

This folder contains 5 files:

- `ChromStruct_3.1.py` (commandline version of CHROMSTRUCT)
- `ChromStruct_3.1_GUI.py` (GUI version of CHROMSTRUCT)
- `Plot_Energy_Chain_1.0.py` (commandline code to display CHROMSTRUCT results)
- `100kb_chr1.txt` (example data file, after Caudai *et al.* 2015)
- `README.pdf` (this file)

The code files are all self-contained and only need a Python interpreter to run. The commandline version groups all the tunable parameters in the first code lines. You should simply open the file and modify their values if needed.

The GUI version displays a window where all the parameters are ready to be modified before starting the program (see figure below).



Three groups of quantities are displayed: the first includes geometrical features, the second sets up the TAD extraction and the score function, and the third is only related to the simulated annealing algorithm. We chose to make all the parameters available, but in normal use only a few of them need to be tuned:

- GEOMETRY - Original resolution: this is the genomic resolution of the data to be treated. If mistaken, it only influences the size of the estimated configuration, but not its shape.
- GEOMETRY - Max contact frequency: if set to zero (the default), its value is computed as the maximum of the data matrix. In some cases, for example when different segments of the same chain are being estimated separately, it could be appropriate to set it to some fixed, user-defined value.
- GEOMETRY - Bead size ratio: this allows the user to tune the flexibility of the output chain. The adequacy of its choice can be evaluated *a posteriori*, by considering the biological plausibility of the output.
- METHOD - Blocks - Min block size: this can prevent the program from working with too small submatrices. Use cautiously.

Changing the other parameters can influence the performance of the algorithm in a complicated way. We recommend to be careful when acting on them. To fully understand their significance, please refer to the commented source code and to Caudai *et al.* (2016).

The GUI code can be run from the interactive python dialog or from the console window, by invoking

```
python ChromStruct_3.1_GUI.py
```

After setting the parameters, type the data file name in the File field, or press the INPUT DATA button to choose the data file; its complete path then appears in the File field. Then, to start the algorithm, press the START button. If Display intermediate plots is checked, for each block, the program displays the plots of the score function values during simulated annealing, the number of accepted versus proposed updates during simulated annealing, and the estimated 3D structure of the subchain mapped onto the current data block. To continue execution, the user must first close all the graphical windows.

If the data file is

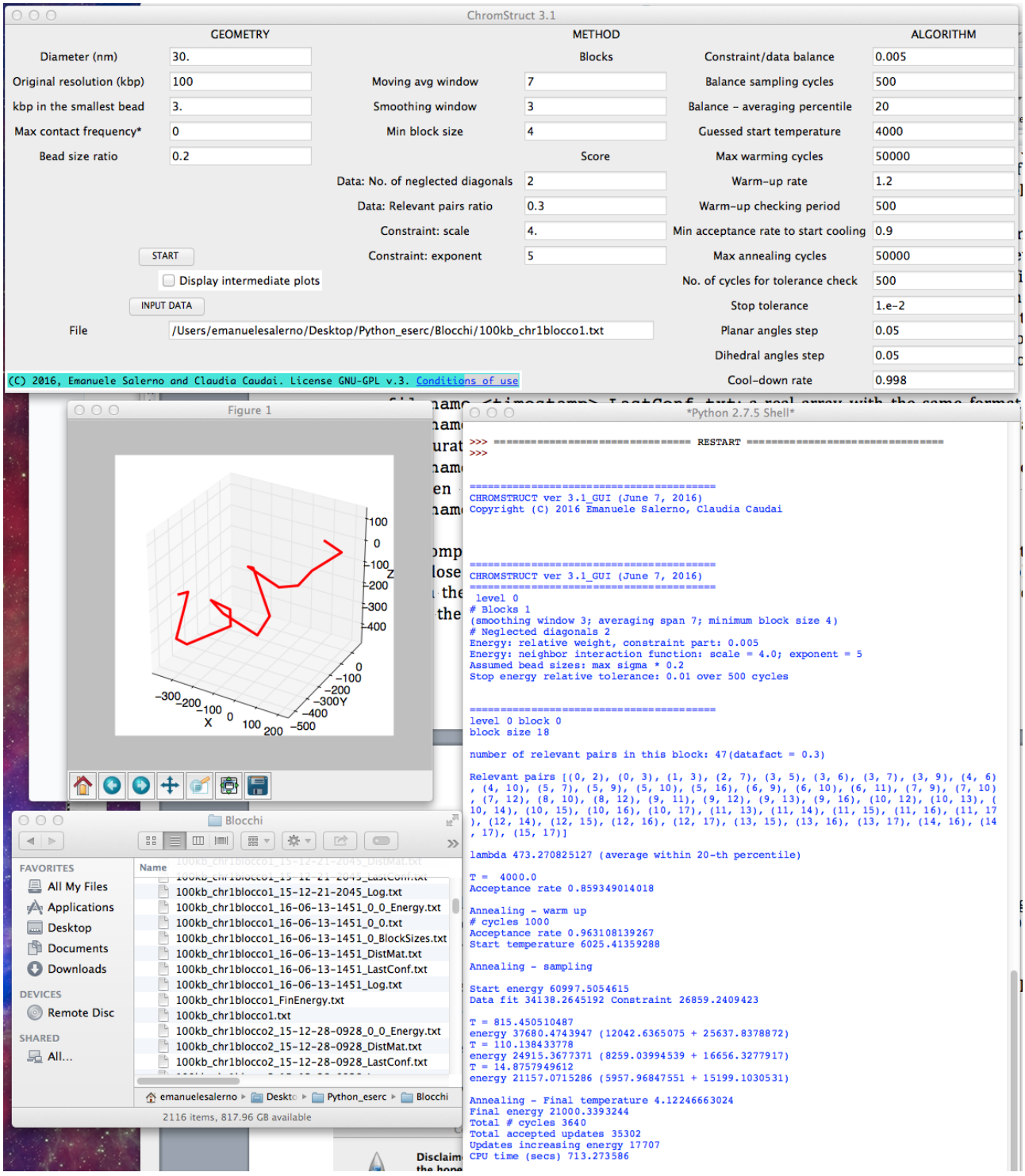
```
filename.suffix
```

(the interactive commandline version inputs the complete filename or, if `.suffix` is `.txt`, only `filename`)

then 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>` 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. It is a string formatted as `<yy-mm-dd-hhmm>`.
- `filename_<timestamp>_Log.txt`: a self-explanatory logfile.
- `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 block 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 so chosen, this is plotted as soon as the annealing iteration at each block and level is complete. It can also be plotted by code `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 so chosen, this structure is plotted as soon as block `<block>` at level `<level>` has been computed. It can also be plotted by code `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.
- `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. The structure plots output by `ChromStruct` display the 3D coordinates of the bead centroids linked by a red ploygonal line. `Plot_Energy_Chain`, conversely, links the centroids with a smooth curve, obtained by cubic spline interpolation.



A screenshot taken during a computation: Top: the CHROMSTRUCT GUI; Bottom - right: the Python interactive window; left: the 3D plot of the reconstructed structure, and the directory list with the data file and the output files, with `<timestamp>` = `16-06-13-1451`.

References

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.

C. Caudai, E. Salerno, M. Zoppè, A. Tonazzini, "3D chromatin structure estimation through a constraint-enhanced score function", ISTI-CNR, Pisa, Italy, Report 2016-B4-003, August 29th, 2016, DOI 10.13140/RG.2.2.30751.97446.

Conditions of use

Please refer to GNU-GPL version 3 or later: <http://www.gnu.org/licenses/gpl-3.0.html>