

---

# **OpenMiChroM**

***Release 1.0.5***

**Antonio B. Oliveira Jr. & Vinícius G. Contessoto**

**Feb 07, 2023**



## GETTING STARTED

<b>1</b>	<b>Installation</b>	<b>1</b>
1.1	Install via conda . . . . .	1
1.2	Install via pip . . . . .	1
<b>2</b>	<b>Introduction</b>	<b>3</b>
<b>3</b>	<b>OpenMiChroM</b>	<b>5</b>
3.1	OpenMiChroM.ChromDynamics . . . . .	5
3.2	OpenMiChroM.Optimization . . . . .	15
3.3	OpenMiChroM.CndbTools . . . . .	20
<b>4</b>	<b>Single Chromosome Simulation</b>	<b>23</b>
4.1	Chromatin Dynamics Simulations on Chromosome 10 of GM12878 Cell Line . . . . .	23
4.2	Trajectory analysis using cndbTools . . . . .	26
<b>5</b>	<b>Multiple Chromosome Simulation</b>	<b>29</b>
5.1	Chromatin Dynamics Simulations on Chromosome 10 and Chromosome 11 of GM12878 Cell Line . . . . .	29
<b>6</b>	<b>Tutorial: Single chromosome optimization using OpenMiChroM</b>	<b>33</b>
6.1	This tutorial enables performing optimization in MiChroM Parameters (Second-order optimization -> Hessian inversion) . . . . .	33
<b>7</b>	<b>Tutorial: Full Inversion Method for a single chromosome using OpenMiChroM</b>	<b>45</b>
7.1	This tutorial enables performing Full Inversion Method using Adam (First-order optimization) . . . . .	45
7.2	Fixing values on the force field . . . . .	77
<b>8</b>	<b>How to cite OpenMiChroM</b>	<b>81</b>
<b>9</b>	<b>References</b>	<b>83</b>
<b>10</b>	<b>License</b>	<b>85</b>
<b>11</b>	<b>Indices and tables</b>	<b>87</b>
	<b>Bibliography</b>	<b>89</b>
	<b>Python Module Index</b>	<b>91</b>
	<b>Index</b>	<b>93</b>



## INSTALLATION

The **OpenMiChroM** library can be installed via [conda](#) or [pip](#), or compiled from [source \(GitHub\)](#).

### 1.1 Install via conda

The code below will install **OpenMiChroM** from [conda-forge](#).

```
conda install -c conda-forge OpenMiChroM
```

---

**Hint:** Often, the installation via conda happens to be stuck. If this is the case, it is recommended to update conda/anaconda using the command line below and try to install **OpenMiChroM** again.

---

```
conda update --prefix /path/to/anaconda3/ anaconda
```

### 1.2 Install via pip

The code below will install **OpenMiChroM** from [PyPI](#).

```
pip3 install OpenMiChroM
```

The **OpenMiChroM** library uses [OpenMM](#) API to run the chromatin dynamics simulations. These requirements can be met by installing the OpenMM package from the [conda-forge channel](#):

```
conda install -c conda-forge openmm
```

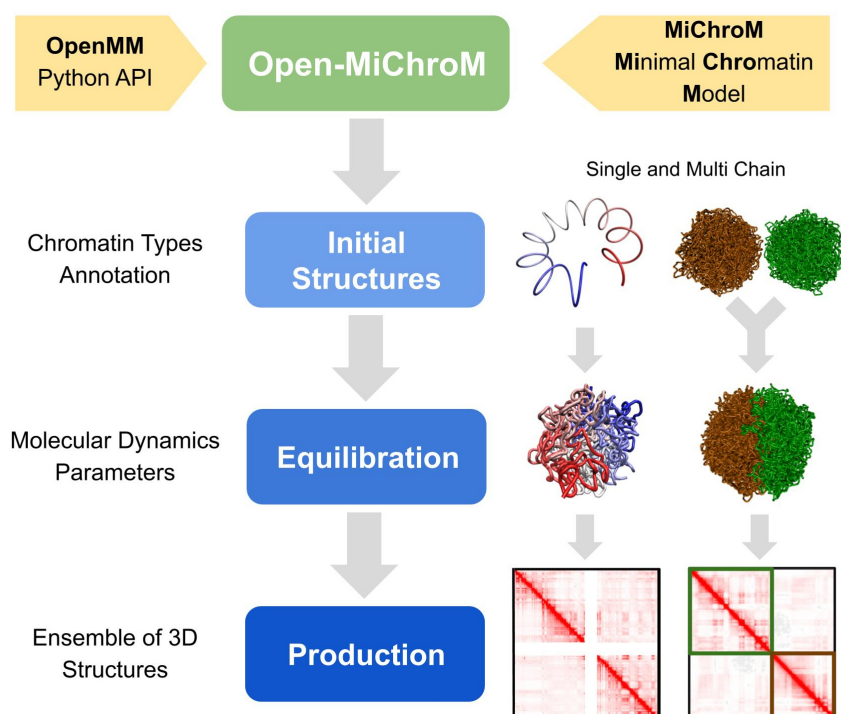
The following are libraries **required** for installing **OpenMiChroM**:

- [Python](#) ( $\geq 3.6$ )
- [NumPy](#) ( $\geq 1.14$ )
- [SciPy](#) ( $\geq 1.5.0$ )
- [six](#) ( $\geq 1.14.0$ )
- [h5py](#) ( $\geq 2.0.0$ )
- [pandas](#) ( $\geq 1.0.0$ )
- [scikit-learn](#) ( $\geq 0.20.0$ )

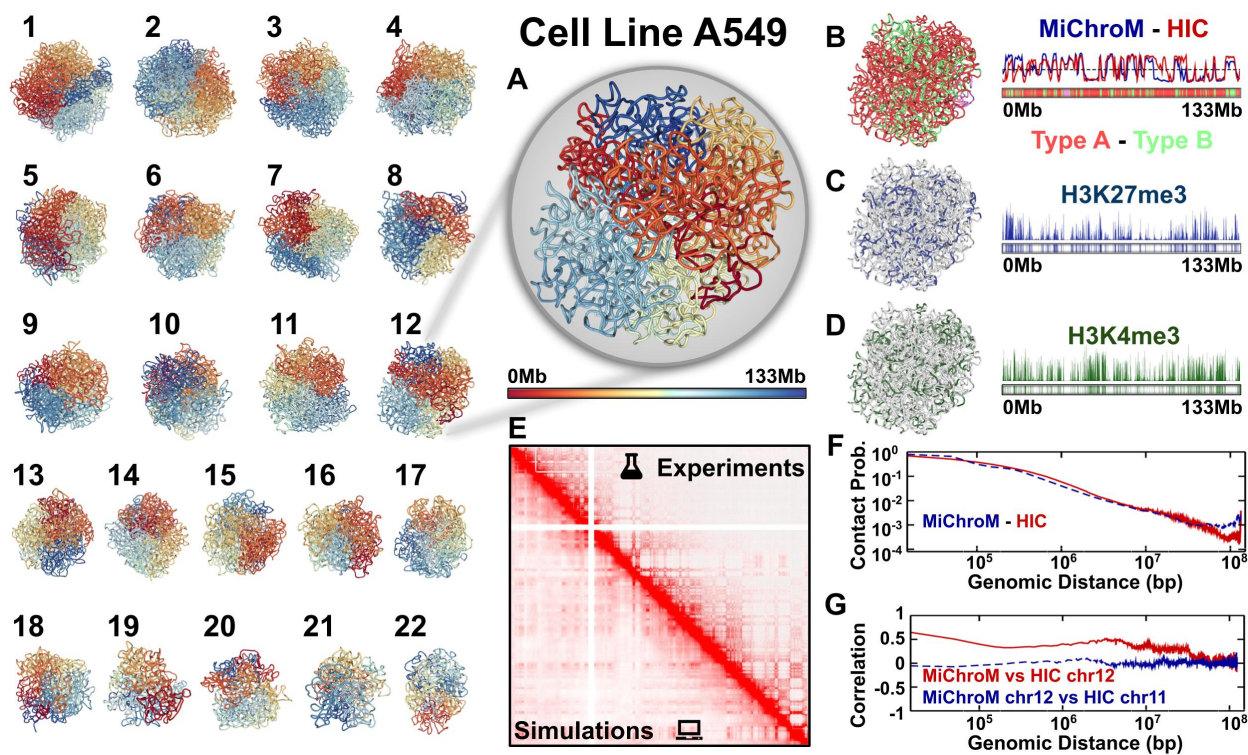


## INTRODUCTION

**OpenMiChroM** is a Python library for performing chromatin dynamics simulations [[1]]. OpenMiChroM uses the **OpenMM** Python API employing the **MiChroM** (Minimal Chromatin Model) energy function [[2]]. The chromatin dynamics simulations generate an ensemble of 3D chromosomal structures that are consistent with experimental Hi-C maps [[3]]. Open-MiChroM also allows simulations of a single or multiple chromosome chain using High-Performance Computing in different platforms (GPUs and CPUs).



The chromatin dynamics simulations can be performed for different human cell lines, cell phases (interphase to metaphase), and different organisms from [DNA Zoo](#). Chromatin subcompartment annotations are available at the [NDB \(Nucleome Data Bank\)](#). OpenMiChroM package receives the chromatin sequence of compartments and subcompartments as input to create and simulate a chromosome polymer model. Examples of running the simulations and generating the *in silico* Hi-C maps can be found [here](#)





## OPENMICHROM

### 3.1 OpenMiChroM.ChromDynamics

The `ChromDynamics` classes perform chromatin dynamics based on the compartment annotations sequence of chromosomes. The simulations can be performed either using the default parameters of MiChroM (Minimal Chromatin Model) or using custom values for the type-to-type and Ideal Chromosome parameters..

```
class OpenMiChroM.ChromDynamics.MiChroM(time_step=0.01, collision_rate=0.1, temperature=1.0,  
                                           verbose=False, velocity_reinitialize=True,  
                                           name='Chromosome', length_scale=1.0, mass_scale=1.0)
```

Bases: object

The *MiChroM* class performs chromatin dynamics employing the default MiChroM energy function parameters for the type-to-type and Ideal Chromosome interactions.

Details about the MiChroM (Minimal Chromatin Model) energy function and the default parameters are described in “Di Pierro, M., Zhang, B., Aiden, E.L., Wolynes, P.G. and Onuchic, J.N., 2016. Transferable model for chromosome architecture. *Proceedings of the National Academy of Sciences*, 113(43), pp.12168-12173.”

The *MiChroM* sets the environment to start the chromatin dynamics simulations.

#### Parameters

- **time\_step** (*float*, *required*) – Simulation time step in units of  $\tau$ . (Default value = 0.01).
- **collision\_rate** (*float*, *required*) – Friction/Damping constant in units of reciprocal time ( $1/\tau$ ). (Default value = 0.1).
- **temperature** (*float*, *required*) – Temperature in reduced units. (Default value = 1.0).
- **verbose** (*bool*, *optional*) – Whether to output the information in the screen during the simulation. (Default value: False).
- **velocity\_reinitialize** (*bool*, *optional*) – Reset/Reinitialize velocities if  $E_{kin}$  is greater than 5.0. (Default value: True).
- **name** (*str*) – Name used in the output files. (Default value: *Chromosome*).
- **length\_scale** (*float*, *required*) – Length scale used in the distances of the system in units of reduced length  $\sigma$ . (Default value = 1.0).
- **mass\_scale** (*float*, *required*) – Mass scale used in units of  $\mu$ . (Default value = 1.0).

**addAdditionalForce**(*forceFunction*, *\*\*args*)

” Add an additional force after the system has already been initialized.

#### Parameters

- **forceFunction** (*function*, *required*) – Force function to be added. Example: `addSphericalConfinementLJ`
- **\*\*args** (*collection of arguments*, *required*) – Arguments of the function to add the force. Consult respective documentation.

**addAngles**(*ka=2.0*)

Adds an angular potential between bonds connecting beads  $i-1, i$  and  $i, i+1$  according to “Halverson, J.D., Lee, W.B., Grest, G.S., Grosberg, A.Y. and Kremer, K., 2011. Molecular dynamics simulation study of nonconcatenated ring polymers in a melt. I. Statics. The Journal of chemical physics, 134(20), p.204904”.

**Parameters** **ka** (*float*, *required*) – Angle potential coefficient. (Default value = 2.0).

**addBond**(*i, j, distance=None, kfb=30*)

Adds bonds between loci  $i$  and  $j$

**Parameters**

- **kfb** (*float*, *required*) – Bond coefficient. (Default value = 30.0).
- **i** (*int*, *required*) – Locus index  $i$ .
- **j** (*int*, *required*) – Locus index  $j$

**addCustomIC**(*mu=3.22, rc=1.78, dinit=3, dend=200, IClist=None*)

Adds the Ideal Chromosome potential using custom values for interactions between beads separated by a genomic distance  $d$ . The parameters  $\mu$  (mu) and  $rc$  are part of the probability of crosslink function  $f(r_{i,j}) = \frac{1}{2} (1 + \tanh[\mu(rc - r_{i,j})])$ , where  $r_{i,j}$  is the spatial distance between loci (beads)  $i$  and  $j$ .

**Parameters**

- **mu** (*float*, *required*) – Parameter in the probability of crosslink function. (Default value = 3.22).
- **rc** (*float*, *required*) – Parameter in the probability of crosslink function,  $f(rc) = 0.5$ . (Default value = 1.78).
- **dinit** (*int*, *required*) – The first neighbor in sequence separation (Genomic Distance) to be considered in the Ideal Chromosome potential. (Default value = 3).
- **dend** (*int*, *required*) – The last neighbor in sequence separation (Genomic Distance) to be considered in the Ideal Chromosome potential. (Default value = 200).
- **IClist** (*file*, *optional*) – A one-column text file containing the energy interaction values for loci  $i$  and  $j$  separated by a genomic distance  $d$ . (Default value: None).

**addCustomTypes**(*name='CustomTypes', mu=3.22, rc=1.78, TypesTable=None*)

Adds the type-to-type potential using custom values for interactions between the chromatin types. The parameters  $\mu$  (mu) and  $rc$  are part of the probability of crosslink function  $f(r_{i,j}) = \frac{1}{2} (1 + \tanh[\mu(rc - r_{i,j})])$ , where  $r_{i,j}$  is the spatial distance between loci (beads)  $i$  and  $j$ .

The function receives a txt/TSV/CSV file containing the upper triangular matrix of the type-to-type interactions. A file example can be found [here](#).

	A	B	C
A	-0.2	-0.25	-0.15
B		-0.3	-0.15
C			-0.35

**Parameters**

- **name** (*string*, *required*) – Name to customType Potential. (Default value = “Custom-Types”)
- **mu** (*float*, *required*) – Parameter in the probability of crosslink function. (Default value = 3.22).
- **rc** (*float*, *required*) – Parameter in the probability of crosslink function,  $f(rc) = 0.5$ . (Default value = 1.78).
- **TypesTable** (*file*, *required*) – A txt/TSV/CSV file containing the upper triangular matrix of the type-to-type interactions. (Default value: None).

#### **addFENEbonds**(*kfb=30.0*)

Adds FENE (Finite Extensible Nonlinear Elastic) bonds between neighbor loci  $i$  and  $i + 1$  according to “Halverson, J.D., Lee, W.B., Grest, G.S., Grosberg, A.Y. and Kremer, K., 2011. Molecular dynamics simulation study of nonconcatenated ring polymers in a melt. I. Statics. The Journal of chemical physics, 134(20), p.204904”.

**Parameters** **kfb** (*float*, *required*) – Bond coefficient. (Default value = 30.0).

#### **addFlatBottomHarmonic**(*kr=0.005*, *n\_rad=10.0*)

Sets a Flat-Bottom Harmonic potential to collapse the chromosome chain inside the nucleus wall. The potential is defined as:  $step(r - r_0) * (kr/2) * (r - r_0)^2$ .

##### **Parameters**

- **kr** (*float*, *required*) – Spring constant. (Default value = 5e-3).
- **n\_rad** (*float*, *required*) – Nucleus wall radius in units of  $\sigma$ . (Default value = 10.0).

#### **addIdealChromosome**(*mu=3.22*, *rc=1.78*, *Gamma1=-0.03*, *Gamma2=-0.351*, *Gamma3=-3.727*, *dinit=3*, *dend=500*)

Adds the Ideal Chromosome potential for interactions between beads separated by a genomic distance  $d$  according to the MiChroM energy function parameters reported in “Di Pierro, M., Zhang, B., Aiden, E.L., Wolynes, P.G. and Onuchic, J.N., 2016. Transferable model for chromosome architecture. Proceedings of the National Academy of Sciences, 113(43), pp.12168-12173”.

The set of parameters  $\{\gamma_d\}$  of the Ideal Chromosome potential is fitted in a function:  $\gamma(d) = \frac{\gamma_1}{\log(d)} + \frac{\gamma_2}{d} + \frac{\gamma_3}{d^2}$ .

The parameters  $\mu$  (mu) and  $rc$  are part of the probability of crosslink function  $f(r_{i,j}) = \frac{1}{2} (1 + \tanh[\mu(r_c - r_{i,j})])$ , where  $r_{i,j}$  is the spatial distance between loci (beads)  $i$  and  $j$ .

##### **Parameters**

- **mu** (*float*, *required*) – Parameter in the probability of crosslink function. (Default value = 3.22).
- **rc** (*float*, *required*) – Parameter in the probability of crosslink function,  $f(rc) = 0.5$ . (Default value = 1.78).
- **Gamma1** (*float*, *required*) – Ideal Chromosome parameter. (Default value = -0.030).
- **Gamma2** (*float*, *required*) – Ideal Chromosome parameter. (Default value = -0.351).
- **Gamma3** (*float*, *required*) – Ideal Chromosome parameter. (Default value = -3.727).
- **dinit** (*int*, *required*) – The first neighbor in sequence separation (Genomic Distance) to be considered in the Ideal Chromosome potential. (Default value = 3).
- **dend** (*int*, *required*) – The last neighbor in sequence separation (Genomic Distance) to be considered in the Ideal Chromosome potential. (Default value = 500).

**addLoops**(*mu*=3.22, *rc*=1.78, *X*=- 1.61299, *looplists*=None)

Adds the Loops interactions according to the MiChroM energy function parameters reported in “Di Pierro, M., Zhang, B., Aiden, E.L., Wolynes, P.G. and Onuchic, J.N., 2016. Transferable model for chromosome architecture. Proceedings of the National Academy of Sciences, 113(43), pp.12168-12173”.

The parameters  $\mu$  (*mu*) and *rc* are part of the probability of crosslink function  $f(r_{i,j}) = \frac{1}{2} (1 + \tanh [\mu(r_c - r_{i,j})])$ , where  $r_{i,j}$  is the spatial distance between loci (beads) *i* and *j*.

---

**Note:** For Multi-chain simulations, the ordering of the loop list files is important! The order of the files should be the same as used in the other functions.

---

**Parameters**

- **mu** (*float*, *required*) – Parameter in the probability of crosslink function. (Default value = 3.22).
- **rc** (*float*, *required*) – Parameter in the probability of crosslink function,  $f(rc) = 0.5$ . (Default value = 1.78).
- **X** (*float*, *required*) – Loop interaction parameter. (Default value = -1.612990).
- **looplists** (*file*, *optional*) – A two-column text file containing the index *i* and *j* of a loci pair that form loop interactions. (Default value: None).

**addMultiChainIC**(*mu*=3.22, *rc*=1.78, *Gamma1*=- 0.03, *Gamma2*=- 0.351, *Gamma3*=- 3.727, *dinit*=3, *dend*=500, *chainIndex*=0)

Adds the Ideal Chromosome potential for multiple chromosome simulations. The interactions between beads separated by a genomic distance *d* is applied according to the MiChroM energy function parameters reported in “Di Pierro, M., Zhang, B., Aiden, E.L., Wolynes, P.G. and Onuchic, J.N., 2016. Transferable model for chromosome architecture. Proceedings of the National Academy of Sciences, 113(43), pp.12168-12173”.

The set of parameters  $\{\gamma_d\}$  of the Ideal Chromosome potential is fitted in a function:  $\gamma(d) = \frac{\gamma_1}{\log(d)} + \frac{\gamma_2}{d} + \frac{\gamma_3}{d^2}$ .

The parameters  $\mu$  (*mu*) and *rc* are part of the probability of crosslink function  $f(r_{i,j}) = \frac{1}{2} (1 + \tanh [\mu(r_c - r_{i,j})])$ , where  $r_{i,j}$  is the spatial distance between loci (beads) *i* and *j*.

**Parameters**

- **mu** (*float*, *required*) – Parameter in the probability of crosslink function. (Default value = 3.22).
- **rc** (*float*, *required*) – Parameter in the probability of crosslink function,  $f(rc) = 0.5$ . (Default value = 1.78).
- **Gamma1** (*float*, *required*) – Ideal Chromosome parameter. (Default value = -0.030).
- **Gamma2** (*float*, *required*) – Ideal Chromosome parameter. (Default value = -0.351).
- **Gamma3** (*float*, *required*) – Ideal Chromosome parameter. (Default value = -3.727).
- **dinit** (*int*, *required*) – The first neighbor in sequence separation (Genomic Distance) to be considered in the Ideal Chromosome potential. (Default value = 3).
- **dend** (*int*, *required*) – The last neighbor in sequence separation (Genomic Distance) to be considered in the Ideal Chromosome potential. (Default value = 500).
- **chainIndex** (*integer*, *required*) – The index of the chain to add the Ideal Chromosome potential. All chains are stored in `self.chains`. (Default value: 0).

**addRepulsiveSoftCore**(*Ecut=4.0*)

Adds a soft-core repulsive interaction that allows chain crossing, which represents the activity of topoisomerase II. Details can be found in the following publications:

- Oliveira Jr., A.B., Contessoto, V.G., Mello, M.F. and Onuchic, J.N., 2021. A scalable computational approach for simulating complexes of multiple chromosomes. *Journal of Molecular Biology*, 433(6), p.166700.
- Di Pierro, M., Zhang, B., Aiden, E.L., Wolynes, P.G. and Onuchic, J.N., 2016. Transferable model for chromosome architecture. *Proceedings of the National Academy of Sciences*, 113(43), pp.12168-12173.
- Naumova, N., Imakaev, M., Fudenberg, G., Zhan, Y., Lajoie, B.R., Mirny, L.A. and Dekker, J., 2013. Organization of the mitotic chromosome. *Science*, 342(6161), pp.948-953.

**Parameters** *Ecut* (*float*, *required*) – Energy cost for the chain passing in units of  $k_bT$ . (Default value = 4.0).

**addSphericalConfinementLJ**(*r='density'*, *density=0.1*)

Sets the nucleus wall potential according to MiChroM Energy function. The confinement potential describes the interaction between the chromosome and a spherical wall.

**Parameters**

- *r* (*float* or *str="density"*, *optional*) – Radius of the nucleus wall. If *r="density"* requires a *density* value.
- *density* (*float*, *required* if *r="density"*) – Density of the chromosome beads inside the nucleus. (Default value = 0.1).

**addTypeToType**(*mu=3.22*, *rc=1.78*)

Adds the type-to-type interactions according to the MiChroM energy function parameters reported in “Di Pierro, M., Zhang, B., Aiden, E.L., Wolynes, P.G. and Onuchic, J.N., 2016. Transferable model for chromosome architecture. *Proceedings of the National Academy of Sciences*, 113(43), pp.12168-12173”.

The parameters  $\mu$  (*mu*) and *rc* are part of the probability of crosslink function  $f(r_{i,j}) = \frac{1}{2} (1 + \tanh[\mu(r_c - r_{i,j})])$ , where  $r_{i,j}$  is the spatial distance between loci (beads) *i* and *j*.

**Parameters**

- *mu* (*float*, *required*) – Parameter in the probability of crosslink function. (Default value = 3.22).
- *rc* (*float*, *required*) – Parameter in the probability of crosslink function,  $f(rc) = 0.5$ . (Default value = 1.78).

**chromRG**()

Calculates the Radius of Gyration of a chromosome chain.

**Returns** Returns the Radius of Gyration in units of  $\sigma$

**createLine**(*ChromSeq*)

Creates a straight line for the initial configuration of the chromosome polymer.

**Parameters**

- **ChromSeq** (*file*, *required*) – Chromatin sequence of types file. The first column should contain the locus index. The second column should have the locus type annotation. A template of the chromatin sequence of types file can be found at the [Nucleome Data Bank \(NDB\)](#).

- **length\_scale** (*float, required*) – Length scale used in the distances of the system in units of reduced length  $\sigma$ . (Default value = 1.0).

**Returns** Returns an array of positions.

**Return type** ( $N, 3$ ) `numpy.ndarray`

**createRandomWalk**(*ChromSeq=None*)

Creates a chromosome polymer chain with beads position based on a random walk.

Args:

**ChromSeq** (**file**, **required**): Chromatin sequence of types file. The first column should contain the locus index. The second column should have the locus type annotation. A template of the chromatin sequence of types file can be found at the [Nucleome Data Bank \(NDB\)](#).

**Returns** Returns an array of positions.

**Return type** ( $N, 3$ ) `numpy.ndarray`

**createSpringSpiral**(*ChromSeq=None, isRing=False*)

Creates a spring-spiral-like shape for the initial configuration of the chromosome polymer.

**Parameters**

- **ChromSeq** (*file, required*) – Chromatin sequence of types file. The first column should contain the locus index. The second column should have the locus type annotation. A template of the chromatin sequence of types file can be found at the [Nucleome Data Bank \(NDB\)](#).
- **isRing** (*bool, optional*) – Whether the chromosome chain is circular or not (Used to simulate bacteria genome, for example). If `bool(isRing)` is `True`, the first and last particles of the chain are linked, forming a ring. (Default value = `False`).

**Returns** Returns an array of positions.

**Return type** ( $N, 3$ ) `numpy.ndarray`

**getLoops**(*looplists*)

Get the loop position (CTFC anchor points) for each chromosome.

---

**Note:** For Multi-chain simulations, the ordering of the loop list files is important! The order of the files should be the same as used in the other functions.

---

**Parameters looplists** (*text file*) – A two-column text file containing the index  $i$  and  $j$  of a loci pair that form loop interactions.

**getPosition**()

**Returns** Returns an array of positions.

**Return type** ( $N, 3$ ) `numpy.ndarray`

**getScaledData**()

Internal function for keeping the system in the simulation box if PBC is employed.

**getVelocities**()

**Returns** Returns an array of velocities.

**Return type**  $(N, 3)$  `numpy.ndarray`

### **initPositions()**

Internal function that sends the locus coordinates to OpenMM system.

### **initStorage(filename, mode='w')**

Initializes the `.cndb` files to store the chromosome structures.

#### **Parameters**

- **filename** (*str*, *required*) – Filename of the `cndb/h5dict` storage file.
- **mode** (*str*, *required*) –
  - 'w' - Create file, truncate if exists. (Default value = w).
  - 'w-' - Create file, fail if exists.
  - 'r+' - Continue saving the structures in the same file that must exist.

### **initStructure(mode='auto', CoordFiles=None, ChromSeq=None, isRing=False)**

Creates the coordinates for the initial configuration of the chromosomal chains and sets their sequence information.

Args:

#### **mode (str, required):**

- 'auto' - Creates a spring-spiral-like shape when a `CoordFiles` is not provided. If `CoordFiles` is provided, it loads the respective type of coordinate files (`.ndb`, `.gro`, or `.pdb`). (Default value = 'auto').
- 'line' - Creates a straight line for the initial configuration of the chromosome polymer. Can only be used to create single chains.
- 'spring' - Creates a spring-spiral-like shape for the initial configuration of the chromosome polymer. Can only be used to create single chains.
- 'random' - Creates a chromosome polymeric chain with beads positions based on a random walk. Can only be used to create single chains.
- 'ndb' - Loads a single or multiple `.ndb` files and gets the position and types of the chromosome beads.
- 'pdb' - Loads a single or multiple `.pdb` files and gets the position and types of the chromosome beads.
- 'gro' - Loads a single or multiple `.gro` files and gets the position and types of the chromosome beads.

**CoordFiles (list of files, optional):** List of files with xyz information for each chromosomal chain. Accepts `.ndb`, `.pdb`, and `.gro` files. All files provided in the list must be in the same file format.

**ChromSeq (list of files, optional):** List of files with sequence information for each chromosomal chain. The first column should contain the locus index. The second column should have the locus type annotation. A template of the chromatin sequence of types file can be found at the [Nucleome Data Bank \(NDB\)](#). If the chromatin types considered are different from the ones used in the original MiChroM (A1, A2, B1, B2, B3, B4, and NA), the sequence file must be provided when loading `.pdb` or `.gro` files, otherwise, all the chains will be defined with 'NA' type. For the `.ndb` files, the sequence used is the one provided in the file.

**isRing (bool, optional):** Whether the chromosome chain is circular or not (used to simulate bacteria genome, for example). To be used with the option 'random'. If `bool(isRing)` is `True`, the first and last particles of the chain are linked, forming a ring. (Default value = `False`).

**Returns** Returns an array of positions.

**Return type**  $(N, 3)$  `numpy.ndarray`

**initVelocities**(*mult=1.0*)

Internal function that set the locus velocity to OpenMM system.

**Parameters** *mult* (*float, optional*) – Rescale initial velocities. (Default value = 1.0).

**loadGRO**(*GROfiles=None, ChromSeq=None*)

Loads a single or multiple *.gro* files and gets position and types of the chromosome beads. Initially, the MiChroM energy function was implemented in GROMACS. Details on how to run and use these files can be found at the [Nucleome Data Bank](#).

- Contessoto, V.G., Cheng, R.R., Hajitaheri, A., Dodero-Rojas, E., Mello, M.F., Lieberman-Aiden, E., Wolynes, P.G., Di Pierro, M. and Onuchic, J.N., 2021. The Nucleome Data Bank: web-based resources to simulate and analyze the three-dimensional genome. *Nucleic Acids Research*, 49(D1), pp.D172-D182.

**Parameters**

- **GROfiles** (*list of files, required*) – List with a single or multiple files in *.gro* file format. (Default value: None).
- **ChromSeq** (*list of files, optional*) – List of files with sequence information for each chromosomal chain. The first column should contain the locus index. The second column should have the locus type annotation. A template of the chromatin sequence of types file can be found at the [Nucleome Data Bank \(NDB\)](#). If the chromatin types considered are different from the ones used in the original MiChroM (A1, A2, B1, B2, B3, B4, and NA), the sequence file must be provided, otherwise all the chains will be defined with ‘NA’ type.

**Returns** Returns an array of positions.

**Return type**  $(N, 3)$  `numpy.ndarray`

**loadNDB**(*NDBfiles=None*)

Loads a single or multiple *.ndb* files and gets position and types of the chromosome beads. Details about the NDB file format can be found at the [Nucleome Data Bank](#).

- Contessoto, V.G., Cheng, R.R., Hajitaheri, A., Dodero-Rojas, E., Mello, M.F., Lieberman-Aiden, E., Wolynes, P.G., Di Pierro, M. and Onuchic, J.N., 2021. The Nucleome Data Bank: web-based resources to simulate and analyze the three-dimensional genome. *Nucleic Acids Research*, 49(D1), pp.D172-D182.

**Parameters** **NDBfiles** (*file, required*) – Single or multiple files in *.ndb* file format. (Default value: None).

**Returns** Returns an array of positions.

**Return type**  $(N, 3)$  `numpy.ndarray`

**loadPDB**(*PDBfiles=None, ChromSeq=None*)

Loads a single or multiple *.pdb* files and gets position and types of the chromosome beads. Here we consider the chromosome beads as the carbon-alpha to mimic a protein. This trick helps to use the standard macro-molecules visualization software. The type-to-residue conversion follows: {‘ALA’:0, ‘ARG’:1, ‘ASP’:2, ‘GLU’:3, ‘GLY’:4, ‘LEU’:5, ‘ASN’:6}.

**Parameters**



- **PDBfiles** (*list of files, required*) – List with a single or multiple files in *.pdb* file format. (Default value: None).
- **ChromSeq** (*list of files, optional*) – List of files with sequence information for each chromosomal chain. The first column should contain the locus index. The second column should have the locus type annotation. A template of the chromatin sequence of types file can be found at the [Nucleome Data Bank \(NDB\)](#). If the chromatin types considered are different from the ones used in the original MiChroM (A1, A2, B1, B2, B3, B4, and NA), the sequence file must be provided, otherwise all the chains will be defined with ‘NA’ type.

**Returns** Returns an array of positions.

**Return type** ( $N, 3$ ) `numpy.ndarray`

**loadStructure**(*filename, center=True, masses=None*)

Loads the 3D position of each bead of the chromosome polymer in the OpenMM system platform.

**Parameters**

- **center** (*bool, optional*) – Whether to move the center of mass of the chromosome to the 3D position  $[0, 0, 0]$  before starting the simulation. (Default value: True).
- **masses** (*array, optional*) – Masses of each chromosome bead measured in units of  $\mu$ . (Default value: None).

**printForces**()

Prints the energy values for each force applied in the system.

**printHeader**()

**printStats**()

Prints some statistical information of a system.

**random\_ChromSeq**(*Nbeads*)

Creates a random sequence of chromatin types for the chromosome beads.

**Parameters** **Nbeads** (*int, required*) – Number of beads of the chromosome polymer chain. (Default value = 1000).

**Returns** Returns an 1D array of a randomized chromatin type annotation sequence.

**Return type** ( $N, 1$ ) `numpy.ndarray`

**randomizePositions**()

Runs automatically to offset the positions if it is an integer (int) variable.

**removeFlatBottomHarmonic**()

” Remove FlatBottomHarmonic force from the system.

**removeForce**(*forceName*)

” Remove force from the system.

**runSimBlock**(*steps=None, increment=True, num=None*)

Performs a block of simulation steps.

**Parameters**

- **steps** (*int, required*) – Number of steps to perform in the block.
- **increment** (*bool, optional*) – Whether to increment the steps counter. Typically it is set False during the collapse or equilibration simulations. (Default value: True).
- **num** (*int or None, required*) – The number of subblocks to split the steps of the primary block. (Default value: None).

**saveFolder(folder)**

Sets the folder path to save data.

**Parameters** **folder** (*str*, *optional*) – Folder path to save the simulation data. If the folder path does not exist, the function will create the directory.

**saveStructure(filename=None, mode='auto', hdf5dictKey='I', pdbGroups=None)**

Save the 3D position of each bead of the chromosome polymer over the chromatin dynamics simulations.

**Parameters**

- **filename** (*str*, *required*) – Filename of the storage file.
- **mode** (*str*, *required*) –
  - 'ndb' - The Nucleome Data Bank file format to save 3D structures of chromosomes. Please see the [NDB - Nucleome Data Bank](#). for details.
  - 'cndb' - The compact ndb file format to save 3D structures of chromosomes. The binary format used the [hdf5 - Hierarchical Data Format](#) to store the data. Please see the NDB server for details. (Default value = cndb).
  - 'pdb' - The Protein Data Bank file format. Here, the chromosome is considered to be a protein where the locus is set at the carbon alpha position. This trick helps to use the standard macromolecules visualization software.
  - 'gro' - The GROMACS file format. Initially, the MiChroM energy function was implemented in GROMACS. Details on how to run and use these files can be found at the [Nucleome Data Bank](#).
  - 'xyz' - A XYZ file format.

**setChains(chains=[(0, None, 0)])**

Sets configuration of the chains in the system. This information is later used for adding Bonds and Angles of the Homopolymer potential.

**Parameters** **chains** (*list of tuples*, *optional*) – The list of chains in the format [(start, end, isRing)]. isRing is a boolean whether the chromosome chain is circular or not (Used to simulate bacteria genome, for example). The particle range should be semi-open, i.e., a chain (0, 3, 0) links the particles 0, 1, and 2. If `bool(isRing)` is `True`, the first and last particles of the chain are linked, forming a ring. The default value links all particles of the system into one chain. (Default value: [(0, None, 0)]).

**setFibPosition(positions, returnCM=False, factor=1.0)**

Distributes the center of mass of chromosomes on the surface of a sphere according to the Fibonacci Sphere algorithm.

**Parameters**

- **positions** ((*Nbeads*, 3) `numpy.ndarray`, *required*) – The array of positions of the chromosome chains to be distributed in the sphere surface.
- **returnCM** (*bool*, *optional*) – Whether to return an array with the center of mass of the chromosomes. (Default value: `False`).
- **factor** (*float*, *optional*) – Scale coefficient to be multiplied to the radius of the nucleus, determining the radius of the sphere in which the center of mass of chromosomes will be distributed. The radius of the nucleus is calculated based on the number of beads to generate a volume density of 0.1.

$$R_{sphere} = factor * R_{nucleus}$$

**Returns**

Returns an array of positions to be loaded into OpenMM using the function [loadStructure](#).

(*Nchains*, 3) **numpy.ndarray**: Returns an array with the new coordinates of the center of mass of each chain.

**Return type** (*Nbeads*, 3) **numpy.ndarray**

**setPosition**(*beadsPos*, *random\_offset=1e-05*)

Sets the 3D position of each bead of the chromosome polymer in the OpenMM system platform.

**Parameters**

- **beadsPos** ((*N*, 3) **numpy.ndarray**) – Array of XYZ positions for each bead (locus) in the polymer model.
- **random\_offset** (*float*, *optional*) – A small increment in the positions to avoid numerical instability and guarantee that a *float* parameter will be used. (Default value = 1e-5).

**setup**(*platform='CUDA'*, *PBC=False*, *PBCbox=None*, *GPU='default'*, *integrator='langevin'*, *errorTol=None*, *precision='mixed'*, *deviceIndex='0'*)

Sets up the parameters of the simulation OpenMM platform.

**Parameters**

- **platform** (*str*, *optional*) – Platform to use in the simulations. Options are *CUDA*, *OpenCL*, *HIP*, *CPU*, *Reference*. (Default value: *CUDA*).
- **PBC** (*bool*, *optional*) – Whether to use periodic boundary conditions. (Default value: *False*).
- **PBCbox** ([*float*, *float*, *float*], *optional*) – Define size of the bounding box for PBC. (Default value: *None*).
- **GPU** (0 or 1, *optional*) – Switch to another GPU. Machines with one GPU automatically select the right GPU. Machines with two or more GPUs select GPU that is less used.
- **integrator** (*str*) – Integrator to use in the simulations. Options are *langevin*, *variableLangevin*, *verlet*, *variableVerlet* and *brownian*. (Default value: *langevin*).
- **verbose** (*bool*, *optional*) – Whether to output the information in the screen during the simulation. (Default value: *False*).
- **deviceIndex** (*str*, *optional*) – Set of Platform device index IDs. Ex: 0,1,2 for the system to use the devices 0, 1 and 2. (Use only when GPU != default)
- **errorTol** (*float*, required if **integrator** = *variableLangevin*) – Error tolerance parameter for *variableLangevin* integrator.

## 3.2 OpenMiChroM.Optimization

The Optimization classes perform the energy function parameters training of the chromosomes based on experimental Hi-C data.

**class** OpenMiChroM.Optimization.**AdamTraining**(*mu=2.0*, *rc=2.0*, *eta=0.01*, *beta1=0.9*, *beta2=0.999*, *epsilon=1e-08*, *it=1*)

Bases: object

The [AdamTraining](#) class performs the parameters training for each selected loci pair interaction.

Details about the methodology are described in “Zhang, Bin, and Peter G. Wolynes. “Topology, structures, and energy landscapes of human chromosomes.” *Proceedings of the National Academy of Sciences* 112.19 (2015): 6062-6067.”

The `AdamTraining` class receives a Hi-C matrix (text file) as input. The parameters  $\mu$  (mu) and  $r_c$  are part of the probability of crosslink function  $f(r_{i,j}) = \frac{1}{2} (1 + \tanh[\mu(r_c - r_{i,j})])$ , where  $r_{i,j}$  is the spatial distance between loci (beads)  $i$  and  $j$ .

#### Parameters

- **mu** (*float, required*) – Parameter in the probability of crosslink function. (Default value = 2.0).
- **rc** (*float, required*) – Parameter in the probability of crosslink function. (Default value = 2.0).
- **eta** (*float, required*) – Learning rate applied in each step (Default value = 0.01).
- **beta1** (*float, required*) – The hyper-parameter of Adam are initial decay rates used when estimating the first and second moments of the gradient. (Default value = 0.9).
- **beta2** (*float, required*) – The hyper-parameter of Adam are initial decay rates used when estimating the first and second moments of the gradient. (Default value = 0.999).
- **it** (*int, required*) – The iteration step

**getLamb**(*Lambdas, fixedPoints=None*)

Calculates the Lagrange multipliers of each pair of interaction and returns the matrix containing the energy values for the optimization step.

#### Parameters

- **Lambdas** (*file, required*) – The matrix containing the energies values used to make the simulation in that step.
- **fixedPoints** (*list, optional*) – List of all pairs (i,j) of interactions that will remain unchanged throughout the optimization procedure.

**Returns** Returns an updated matrix of interactions between each pair of bead.

**Return type** ( $N, N$ ) `numpy.ndarray`

**getPars**(*HiC, centerRemove=False, centrangle=[0, 0], cutoff=0.0*)

Receives the experimental Hi-C map (Full dense matrix) in a text format and performs the data normalization from Hi-C frequency/counts/reads to probability.

#### Parameters

- **HiC** (*file, required*) – Experimental Hi-C map (Full dense matrix) in a text format.
- **centerRemove** (*bool, optional*) – Whether to set the contact probability of the centromeric region to zero. (Default value: `False`).
- **centrangle** (*list, required if centerRemove = True*) – Range of the centromeric region, i.e., `centrangle=[i, j]`, where  $i$  and  $j$  are the initial and final beads in the centromere. (Default value = `[0, 0]`).
- **cutoff** (*float, optional*) – Cutoff value for reducing the noise in the original data. Values lower than the **cutoff** are considered 0.0.

**probCalc**(*state*)

Calculates the contact probability matrix for a given state.

```
class OpenMiChroM.Optimization.CustomMiChroMTraining(ChromSeq='chr_beads.txt', TypesTable=None,
                                                    mu=3.22, rc=1.78, cutoff=0.0, dinit=3,
                                                    dend=200)
```

Bases: object

The *CustomMiChroMTraining* class performs the parameters training employing MiChroM (Minimal Chromatin Model) energy function.

Details about the methodology are described in “Di Pierro, Michele, et al. “Transferable model for chromosome architecture.” Proceedings of the National Academy of Sciences 113.43 (2016): 12168-12173.”

The *CustomMiChroMTraining* class receive a Hi-C matrix (text file) as input. The parameters  $\mu$  (mi) and  $rc$  are part of the probability of crosslink function  $f(r_{i,j}) = \frac{1}{2} (1 + \tanh[\mu(rc - r_{i,j})])$ , where  $r_{i,j}$  is the spatial distance between loci (beads)  $i$  and  $j$ .

*CustomMiChroMTraining* optimizes the type-to-type (Types) and the Ideal Chromosome (IC) potential parameters separately.

#### Parameters

- **ChromSeq** (*file*, *required*) – Chromatin sequence of types file. The first column should contain the locus index. The second column should have the locus type annotation. A template of the chromatin sequence of types file can be found at the [Nucleome Data Bank \(NDB\)](#).
- **TypesTable** (*file*, *required*) – A txt/TSV/CSV file containing the upper triangular matrix of the type-to-type interactions. (Default value: None).
- **mu** (*float*, *required*) – Parameter in the probability of crosslink function (Default value = 3.22, for human chromosomes in interphase).
- **rc** (*float*, *required*) – Parameter in the probability of crosslink function,  $f(rc) = 0.5$  (Default value = 1.78, for human chromosomes in interphase).
- **cutoff** (*float*, *optional*) – Cutoff value for reducing the noise in the original data. Values lower than the **cutoff** are considered 0.0.
- **dinit** (*int*, *required*) – The first neighbor in sequence separation (Genomic Distance) to be considered in the Ideal Chromosome potential for training. (Default value = 3).
- **dend** (*int*, *required*) – The last neighbor in sequence separation (Genomic Distance) to be considered in the Ideal Chromosome potential for training. (Default value = 200).

**calc\_exp\_phi** (*init*=3, *dmax*=200)

Calculates the contact probability as a function of the genomic distance from the experimental Hi-C for the Ideal Chromosome optimization.

**calc\_exp\_phi\_types** ()

Calculates the average of the contact probability for each chromatin type (compartment annotation) from the experimental Hi-C for the Types optimization.

**calc\_sim\_phi** (*init*=3, *dmax*=200)

Calculates the contact probability as a function of the genomic distance from simulations for the Ideal Chromosome optimization.

**calc\_sim\_phi\_types** ()

Calculates the average of the contact probability for each chromatin type (compartment annotation) from simulation for the Types optimization.

**getBijsim** ()

Normalizes the cross term of the Hessian by the number of frames in the simulation for the Ideal Chromosome optimization.

**getChromSeq(filename)**

Converts the letters of the types/compartments following the rule: 'A1':0, 'A2':1, 'B1':2, 'B2':3, 'B3':4, 'B4':5, 'NA':6.

**Parameters filename** (*file, required*) – Chromatin sequence of types file. The first column should contain the locus index. The second column should have the locus type annotation. A template of the chromatin sequence of types file can be found at the [Nucleome Data Bank \(NDB\)](#).

**Returns** Returns an array of the sequence of chromatin types.

**Return type** (*N, 1*) `numpy.ndarray`

**getHiCSim()**

Calculates the *in silico* Hi-C map (Full dense matrix).

**getHiCexp(filename)**

Receives the experimental Hi-C map (Full dense matrix) in a text format and performs the data normalization from Hi-C frequency/counts/reads to probability.

**getLamb(dmax=200, exp\_map='file.dense')**

Calculates the Lagrange multipliers for the Ideal Chromosome optimization and returns an array containing the energy values for the IC optimization step.

**getLamb\_types(exp\_map, damp=5e-07)**

Calculates the Lagrange multipliers of each type-to-type interaction and returns the matrix containing the energy values for the optimization step.

**getPearson()**

Calculates the Pearson's Correlation between the experimental Hi-C used as a reference for the training and the *in silico* Hi-C obtained from the optimization step.

**getPiPjsim\_types()**

Normalizes the cross term of the Hessian by the number of frames in the simulation for the Types optimization.

**getlambfromfile(filename)**

Receives the Lagrange multipliers of the Ideal Chromosome optimization from a text file.

**probCalculation\_IC(state, dmax=200)**

Calculates the contact probability matrix and the cross term of the Hessian for the Ideal Chromosome optimization.

**probCalculation\_types(state)**

Calculates the contact probability matrix and the cross term of the Hessian for the type-to-type interactions optimization.

**class** OpenMiChroM.Optimization.**FullTraining**(*expHiC, mu=2.0, rc=2.5, cutoff=0.0, reduce=True, pair\_h=2, c\_h=0.1, pair\_l=4, c\_l=0.02*)

Bases: `object`

The **FullTraining** class performs the parameters training for each selected loci pair interaction.

Details about the methodology are described in “Zhang, Bin, and Peter G. Wolynes. “Topology, structures, and energy landscapes of human chromosomes.” *Proceedings of the National Academy of Sciences* 112.19 (2015): 6062-6067.”

The **FullTraining** class receives a Hi-C matrix (text file) as input. The parameters  $\mu$  (mu) and  $rc$  are part of the probability of crosslink function  $f(r_{i,j}) = \frac{1}{2} (1 + \tanh[\mu(rc - r_{i,j})])$ , where  $r_{i,j}$  is the spatial distance between loci (beads)  $i$  and  $j$ .

**Parameters**

- **mu** (*float*, *required*) – Parameter in the probability of crosslink function. (Default value = 2.0).
- **rc** (*float*, *required*) – Parameter in the probability of crosslink function,  $f(rc) = 0.5$ . (Default value = 2.5).
- **cutoff** (*float*, *optional*) – Cutoff value for reducing the noise in the original data. Values lower than the **cutoff** are considered 0.0.
- **reduce** (*bool*, *optional*) – Whether to reduce the number of interactions to be considered in the inversion. If False, it will consider every possible interaction ( $N * (N - 1)/2$ ). If True, it is necessary to give values for the lower and higher cutoffs. (Default value: True).
- **pair\_h** (*int*, required if **reduce** = True) – Loci selection to apply the high-resolution cutoff. If **pair\_h** = 2, the interaction in the high-resolution index grid  $2 : 2 : N^2 : 2 : N$  are subject to a cutoff value **c\_h**, where  $N$  is the total number of monomers interactions (Default value = 2).
- **c\_h** (*float*, required if **reduce** = True) – The the high-resolution cutoff. (Default value = 0.1).
- **pair\_l** (*int*, required if **reduce** = True) – Loci selection to apply the high-resolution cutoff. If **pair\_l** = 4, the interaction in the low-resolution index grid  $1 : 4 : N^1 : 4 : N$  are subject to a cutoff value **c\_l**, where  $N$  is the total number of monomers interactions (Default value = 4).
- **c\_l** (*float*, required if **reduce** = True) – The the low-resolution cutoff. (Default value = 0.02).

**appCutoff**(*pair\_h*, *c\_h*, *pair\_l*, *c\_l*)

Applies the cutoff for low- and high-resolution values.

**createInitialLambda**(*sequenceFile*, *outputPath*='.', *initialGuess*=0.0, *baseLine*=- 0.2)

**getHiCexp**(*filename*, *centerRemove*=False, *centrange*=[0, 0])

Receives the experimental Hi-C map (Full dense matrix) in a text format and performs the data normalization from Hi-C frequency/counts/reads to probability.

#### Parameters

- **centerRemove** (*bool*, *optional*) – Whether to set the contact probability of the centromeric region to zero. (Default value: False).
- **centrange** (*list*, required if **centerRemove** = True) – Range of the centromeric region, *i.e.*, **centrange**=[*i*, *j*], where *i* and *j* are the initial and final beads in the centromere. (Default value = [0, 0]).

**getLambdas**()

Calculates the Lagrange multipliers of each selected interaction and returns the matrix containing the energy values for the optimization step.

**getPearson**()

Calculates the Pearson's Correlation between the experimental Hi-C used as a reference for the training and the *in silico* Hi-C obtained from the optimization step.

**get\_indices**(*hic*)

Receives non-zero interaction indices, *i.e.*, the loci pair *i* and *j* which interaction will be optimized.

**probCalc**(*state*)

Calculates the contact probability matrix and the cross term of the Hessian.

**saveLambdas**(*sequenceFile*, *data*, *outputPath*, *name*)

### 3.3 OpenMiChroM.CndbTools

The `cndbTools` class perform analysis from `cndb` or `ndb` - (Nucleome Data Bank) file format for storing an ensemble of chromosomal 3D structures. Details about the NDB/CNDB file format can be found at the [Nucleome Data Bank](https://nucleome.org/data).

**class** OpenMiChroM.CndbTools.cndbTools

Bases: object

**compute\_Chirality**(xyz, neig\_beads=4)

Calculates the Chirality parameter  $\Psi$ . Details are described in “Zhang, B. and Wolynes, P.G., 2016. Shape transitions and chiral symmetry breaking in the energy landscape of the mitotic chromosome. Physical review letters, 116(24), p.248101.”

**Parameters**

- **xyz** ((frames, beadSelection, XYZ) `numpy.ndarray`, required) – Array of the 3D position of the selected beads for different frames extracted by using the :code: xyz() function.
- **neig\_beads** (int, required) – Number of neighbor beads to consider in the calculation (Default value = 4).

**Returns** Returns the Chirality parameter  $\Psi$  for each bead.

**Return type** `numpy.ndarray`

**compute\_FFT\_from\_Oij**(lowcut=1, highcut=500, order=5)

**compute\_GyrTensorEigs**(xyz)

Calculates the eigenvalues of the Gyration tensor: For a cloud of N points with positions: {[xi,yi,zi]}, gyr tensor is a symmetric matrix defined as,

$$\text{gyr} = (1/N) * \begin{bmatrix} \sum_i (\text{xi-xcm})(\text{xi-xcm}) & \sum_i (\text{xi-xcm})(\text{yi-ycm}) & \sum_i (\text{xi-xcm})(\text{zi-zcm}) \\ \sum_i (\text{yi-ycm})(\text{xi-xcm}) & \sum_i (\text{yi-ycm})(\text{yi-ycm}) & \sum_i (\text{yi-ycm})(\text{zi-zcm}) \\ \sum_i (\text{zi-zcm})(\text{xi-xcm}) & \sum_i (\text{zi-zcm})(\text{yi-ycm}) & \sum_i (\text{zi-zcm})(\text{zi-zcm}) \end{bmatrix}$$

the three non-negative eigenvalues of gyr are used to define shape parameters like radius of gyration, asphericity, etc

**Parameters** **xyz** (:math:`(frames, beadSelection, XYZ) - T \times N \times 3``, required): Array of the 3D position of the selected beads for different frames extracted by using the :code: xyz() function.

**Returns**

**Tx3):** Returns the sorted eigenvalues of the Gyration Tensor.

**Return type** `numpy.ndarray` (dim

**compute\_MSD**(xyz)

Calculates the Mean-Squared Displacement using Fast-Fourier Transform. Uses Weiner-Kinchin theorem to compute the autocorrelation, and a recursion relation from the following reference: see Sec. 4.2 in Calandrini V, et al. (2011) EDP Sciences (<https://doi.org/10.1051/sfn/201112010>). Also see this stackoverflow post: <https://stackoverflow.com/questions/34222272/computing-mean-square-displacement-using-python-and-fft>

**Parameters** **xyz** (:math:`(frames, beadSelection, XYZ) - T \times N \times 3``, required): Array of the 3D position of the selected beads for different frames extracted by using the :code: xyz() function.

**Returns**

**NxT):** Returns the MSD of each particle over the trajectory.

**Return type** `numpy.ndarray` (dim

**compute\_Orientation\_OP**(xyz, chrom\_start=0, chrom\_end=1000, vec\_length=4)



**compute\_RDP**(xyz, radius=20.0, bins=200)

Calculates the RDP - Radial Distribution Probability. Details can be found in the following publications:

- Oliveira Jr., A.B., Contessoto, V.G., Mello, M.F. and Onuchic, J.N., 2021. A scalable computational approach for simulating complexes of multiple chromosomes. *Journal of Molecular Biology*, 433(6), p.166700.
- Di Pierro, M., Zhang, B., Aiden, E.L., Wolynes, P.G. and Onuchic, J.N., 2016. Transferable model for chromosome architecture. *Proceedings of the National Academy of Sciences*, 113(43), pp.12168-12173.

**Parameters**

- **xyz** ((frames, beadSelection, XYZ) `numpy.ndarray`, required) – Array of the 3D position of the selected beads for different frames extracted by using the :code: xyz() function.
- **radius** (float, required) – Radius of the sphere in units of  $\sigma$  to be considered in the calculations. The radius value should be modified depending on your simulated chromosome length. (Default value = 20.0).
- **bins** (int, required) – Number of slices to be considered as spherical shells. (Default value = 200).

**Returns**

Returns the radius of each spherical shell in units of  $\sigma$ . (N, 1) `numpy.ndarray`:

Returns the RDP - Radial Distribution Probability for each spherical shell.

**Return type** (N, 1) `numpy.ndarray`

**compute\_RG**(xyz)

Calculates the Radius of Gyration.

**Parameters** **xyz** (:math:`(frames, beadSelection, XYZ) - TxNx3`), required): Array of the 3D position of the selected beads for different frames extracted by using the :code: xyz() function.

**Returns**

**Tx1**): Returns the Radius of Gyration in units of  $\sigma$ .

**Return type** `numpy.ndarray` (dim

**compute\_RadNumDens**(xyz, dr=1.0, ref='centroid', center=None)

Calculates the radial number density of monomers; which when integrated over the volume (with the appropriate kernel:  $4\pi r^2$ ) gives the total number of monomers.

**Parameters**

- **xyz** (:math:`(frames, beadSelection, XYZ) - TxNx3`), required): Array of the 3D position of the selected beads for different frames extracted by using the :code: xyz() function.
- **dr** (float, required) – mesh size of radius for calculating the radial distribution. can be arbitrarily small, but leads to empty bins for small values. bins are computed from the maximum values of radius and dr.
- **ref** (string) – defines reference for centering the distribution. It can take three values:  
 'origin': radial distance is calculated from the center  
 'centroid' (default value): radial distribution is computed from the centroid of the cloud of points at each time step

'custom': user defined center of reference. 'center' is required to be specified when 'custom' reference is chosen

- **center** (*list of float, len 3*) – defines the reference point in custom reference. required when ref='custom'

#### Returns

the number density

**bins**:class:**numpy.ndarray**: bins corresponding to the number density

**Return type** num\_density:class:**numpy.ndarray**

#### **load**(filename)

Receives the path to **cndb** or **ndb** file to perform analysis.

**Parameters filename** (*file, required*) – Path to cndb or ndb file. If an ndb file is given, it is converted to a cndb file and saved in the same directory.

#### **ndb2cndb**(filename)

Converts an **ndb** file format to **cndb**.

**Parameters filename** (*path, required*) – Path to the ndb file to be converted to cndb.

#### **traj2HiC**(xyz, mu=3.22, rc=1.78)

Calculates the *in silico* Hi-C maps (contact probability matrix) using a chromatin dynamics trajectory.

The parameters  $\mu$  (mu) and rc are part of the probability of crosslink function  $f(r_{i,j}) = \frac{1}{2} (1 + \tanh[\mu(r_c - r_{i,j})])$ , where  $r_{i,j}$  is the spatial distance between loci (beads)  $i$  and  $j$ .

#### **Parameters**

- **mu** (*float, required*) – Parameter in the probability of crosslink function. (Default value = 3.22).
- **rc** – Parameter in the probability of crosslink function,  $f(rc) = 0.5$ . (Default value = 1.78).

#### **xyz**(frames=[1, None, 1], beadSelection=None, XYZ=[0, 1, 2])

Get the selected beads' 3D position from a **cndb** or **ndb** for multiple frames.

#### **Parameters**

- **frames** (*list, required*) – Define the range of frames that the position of the bead will get extracted. The range list is defined by frames=[initial, final, step]. (Default value: :code: [1, None, 1], all frames)
- **beadSelection** (*list of ints, required*) – List of beads to extract the 3D position for each frame. The list is defined by :code: beadSelection=[0, 1, 2, ..., N-1]. (Default value: :code: None, all beads)
- **XYZ** (*list, required*) – List of the axis in the Cartesian coordinate system that the position of the bead will get extracted for each frame. The list is defined by :code: XYZ=[0, 1, 2]. where 0, 1 and 2 are the axis X, Y and Z, respectively. (Default value: :code: XYZ=[0, 1, 2])

**Returns** Returns an array of the 3D position of the selected beads for different frames.

**Return type** ( $N_{frames}, N_{beads}, 3$ ) **numpy.ndarray**

## SINGLE CHROMOSOME SIMULATION

This tutorial should take between 20 to 30 minutes of reading and performing simulations.

### 4.1 Chromatin Dynamics Simulations on Chromosome 10 of GM12878 Cell Line

The first step is to import the **OpenMiChroM** module

```
[ ]: from OpenMiChroM.ChromDynamics import MiChroM
     from OpenMiChroM.CndbTools import cndbTools
```

MiChroM class sets the initial parameters of the simulation:

- `time_step=0.01`: set the simulation time step to perform the integration
- `temperature=1.0`: set the temperature of your simulation

```
[ ]: sim = MiChroM(temperature=1.0, time_step=0.01)
```

There are four hardware platform options to run the simulations:

```
platform="cuda"
platform="opencl"
platform="hip"
platform="cpu"
```

Choose accordingly.

```
[ ]: sim.setup(platform="opencl")
```

Set the directory name in which the output of the simulation is saved:

```
[ ]: sim.saveFolder('output_chr10')
```

The next step is to load the chromatin compartment sequence for chromosome 10 and generate an initial 3D structure to start the simulation. We can use the `createSpringSpiral` function to set the initial configuration of the polymer based in the sequence file.

The first column of the sequence file should contain the locus index. The second should have the locus type annotation. A template file of the chromatin sequence of types can be found [here](#).

The loci positions are stored in the variable **chr10** as a NumPy array  $[N : 3]$ , where  $N$  is the number of beads.

```
[ ]: chr10 = sim.createSpringSpiral(ChromSeq='inputs/chr10_beads.txt', isRing=False)
```

We can check the position of the first five beads:

```
[ ]: print(chr10[:5])
```

The initial structure should then be loaded into the `sim` object.

The option `center=True` moves your system to the origin.

```
[ ]: sim.loadStructure(chr10, center=True)
```

The initial 3D chromosome structure can be saved in `.ndb` file format. The file is stored in the path given in `saveFolder`.

```
[ ]: sim.saveStructure(mode='ndb')
```

The next step is to add the force field in the simulation object `sim`.

In this tutorial, the forces can be divided into two sets:

#### MiChroM Homopolymer (Bonded) Potentials

```
[ ]: sim.addFENEbonds(kfb=30.0)
     sim.addAngles(ka=2.0)
     sim.addRepulsiveSoftCore(Ecut=4.0)
```

#### MiChroM Non-Bonded Potentials

```
[ ]: sim.addTypeToType(mu=3.22, rc=1.78)
     sim.addIdealChromosome(mu=3.22, rc=1.78, dinit=3, dend=500)
```

The last potential adds a spherical constrain to collapse the initial structure.

```
[ ]: sim.addFlatBottomHarmonic(kr=5*10**-3, n_rad=15.0)
```

Run a short simulation to generate a collapsed structure.

```
[ ]: block = 3*10**2
     n_blocks = 2*10**3
```

Two variables control the chromatin dynamics simulation steps:

`block`: The number of time-steps performed in each cycle (or block) `n_blocks`: The number of cycles (or blocks) simulated.

The initial collapse simulation will run for  $3 \times 10^2 \times 2 \times 10^3 = 6 \times 10^5$  time-steps.

```
[ ]: for _ in range(n_blocks):
     sim.runSimBlock(block, increment=False)
```

Details about the output of each simulation block:

- `b1=0`: index number of the simulated block. The parameter `increment=False` is used to ignore the steps counting.
- `pos[1]=[X,Y,Z]`: spatial position for the locus 1.
- `dr=1.26`: average of the loci displacements in each block (in units of `sigma`).
- `t=0`: current simulation time.

- `kin=1.5`: kinetic energy of the system (reduced units).
- `pot=19.90`: total potential energy of the system (reduced units).
- `RG=7.654`: radius of gyration at the end of the simulated block.
- `SPS=12312`: steps per second of each block.

The radius of gyration is a good parameter to check the performance of the collapse. If the chromosome polymer is not collapsed, it is necessary to rerun the initial collapse steps. We can also save the structure for inspection.

```
[ ]: print(sim.chromRG())
sim.saveStructure(mode='ndb')
```

The structure can also be saved using standard file formats used for macromolecules, as the `pdb` and `gro` formats.

```
[ ]: sim.saveStructure(mode='gro')
sim.saveStructure(mode='pdb')
```

The next step is to remove the spherical constrain force to run the production simulation.

```
[ ]: sim.removeFlatBottomHarmonic()
```

If necessary, one could remove any of the forces applied in the system. To see the forces in the system:

```
[ ]: sim.forceDict
```

```
[ ]: # sim.removeForce(forceName="TypeToType")
```

To run the production simulation, it is necessary to initialize the `.cndb` file to save the chromatin dynamics trajectory.

```
[ ]: sim.initStorage(filename="traj_chr10")
```

Set the parameters of the production simulation:

$$block = 5 \times 10^2 \quad n\_blocks = 2 \times 10^3$$

```
[ ]: block = 5*10**2
n_blocks = 2*10**3
```

```
[ ]: for _ in range(n_blocks):
    sim.runSimBlock(block, increment=True)
    sim.saveStructure()
```

Once the simulation is completed, it is necessary to close the `.cndb` file to avoid losing the trajectory data.

```
[ ]: sim.storage[0].close()
```

The simulation should generate the `traj_chr10_0.cndb` trajectory file in the `output_chr10` folder. This file contains 2000 frames (one snapshot per block).

## 4.2 Trajectory analysis using cndbTools

cndbTools is a class that allows analyses in the chromatin dynamics trajectories using the binary format `.cndb` (compact ndb).

```
[ ]: cndbTools = cndbTools()
```

Load the cndb file in the variable `chr10_traj`.

```
[ ]: chr10_traj = cndbTools.load('output_chr10/traj_chr10_0.cndb')
```

```
[ ]: print(chr10_traj) # Print the information of the cndb trajectory.
```

Extract the loci XYZ position over the simulated 2000 frames and save in the variable `chr10_xyz`.

```
[ ]: chr10_xyz = cndbTools.xyz(frames=[1,2001,1], beadSelection='all', XYZ=[0,1,2])
```

The variable `chr10_xyz` allows the `cndbTools` to perform several analyses. In this example, the radius of gyration can be obtained as a function of the simulated frames.

```
[ ]: import matplotlib.pyplot as plt
import matplotlib as mpl

chr10_RG = cndbTools.compute_RG(chr10_xyz)
plt.plot(chr10_RG)
plt.ylabel(r'Radius of Gyration ( $\sigma$ )', fontsize=11)
plt.xlabel(r'Simulation Frames', fontsize=11)
```

`cndbTools` allows the selection of beads to compute the analyses. An example is the Radial Distribution Probability (RDP) for each chromatin subcompartments A1 and B1.

```
[ ]: chr10_A1 = cndbTools.xyz(frames=[1,2001,1], beadSelection=chr10_traj.dictChromSeq['A1'],
↪XYZ=[0,1,2])
chr10_B1 = cndbTools.xyz(frames=[1,2001,1], beadSelection=chr10_traj.dictChromSeq['B1'],
↪XYZ=[0,1,2])
```

```
[ ]: print("Computing RDP...")
r_A1, RDP_chr10_A1 = cndbTools.compute_RDP(chr10_A1, radius=15.0, bins=200)
r_B1, RDP_chr10_B1 = cndbTools.compute_RDP(chr10_B1, radius=15.0, bins=200)
```

```
[ ]: plt.plot(r_A1, RDP_chr10_A1, color='red', label='A')
plt.plot(r_B1, RDP_chr10_B1, color='blue', label='B')
plt.xlabel(r'r ( $\sigma$ )', fontsize=11, fontweight='normal', color='k')
plt.ylabel(r' $\rho(r)/N_{type}$ ', fontsize=11, fontweight='normal', color='k')
plt.legend()
plt.gca().set_xlim([1/200, 15.0])
```

We can also use `cndbTools` to generate the *in silico* Hi-C map (contact probability matrix).

In this tutorial, the trajectory contains 2,000 snapshots of chromosome 10 of the GM12878 cell line. For this set of structures, we expect the *in silico* Hi-C to not be fully converged due to inadequate sampling. To produce a converged map, it is recommended to simulate around 20 replicas with 10,000 frames on each, which generates an ensemble of 200,000 chromosome structures.

```
[ ]: print("Generating the contact probability matrix...")
chr10_sim_HiC = cndbTools.traj2HiC(chr10_xyz)

[ ]: plt.matshow(chr10_sim_HiC, norm=matplotlib.colors.LogNorm(vmin=0.001, vmax=chr10_sim_HiC.max()),
    cmap="Reds")
plt.colorbar()
```

To visualize the chromosome's 3D structures in the standard visualization softwares for macromolecules, there are available scripts for converting the ndb/cndb file format to .pdb and .gro. For details, please check the [Nucleome Data Bank](#).

The ndb plugin for visualizing the chromatin dynamics trajectories in VMD/Chimera/Pymol is under development.





## MULTIPLE CHROMOSOME SIMULATION

This tutorial should take between 20 to 30 minutes of reading and performing simulations.

### 5.1 Chromatin Dynamics Simulations on Chromosome 10 and Chromosome 11 of GM12878 Cell Line

The first step is to import the **OpenMiChroM** module

```
[ ]: from OpenMiChroM.ChromDynamics import MiChroM
     from OpenMiChroM.CndbTools import cndbTools
```

OpenMiChroM allows the simulation of multiples chromosomes. In this tutorial, the multiple chain simulation will be performed using the chromosomes 10 and 11 of the human GM12878 cell line. The system is generated based on the collapsed structure of each chromosome. This collapse step is similar to the one presented in the [single chromosome simulation tutorial](#). For each chromosome, the collapse simulation should be performed individually.

#### 5.1.1 Chromosome 10 Collapse Simulation

```
[ ]: sim_chr10 = MiChroM(name="chr10", temperature=1.0, time_step=0.01)
     sim_chr10.setup(platform="opencl")
     sim_chr10.saveFolder('output_nucleus')
     chr10 = sim_chr10.createSpringSpiral(ChromSeq='inputs/chr10_beads.txt')
     sim_chr10.loadStructure(chr10, center=True)

     sim_chr10.addFENEbonds(kfb=30.0)
     sim_chr10.addAngles(ka=2.0)
     sim_chr10.addRepulsiveSoftCore(Ecut=4.0)
     sim_chr10.addFlatBottomHarmonic()

     sim_chr10.addTypeToType(mu=3.22, rc=1.78)
     sim_chr10.addIdealChromosome(mu=3.22, rc=1.78, dinit=3, dend=500)

     print("Performing chr10 simulation...")
     for _ in range(1000):
         sim_chr10.runSimBlock(500)

     sim_chr10.saveStructure(filename="chr10", mode="ndb")
     del sim_chr10
```

### 5.1.2 Chromosome 11 Collapse Simulation

```
[ ]: sim_chr11 = MiChroM(name="chr11", temperature=1.0, time_step=0.01)
sim_chr11.setup(platform="opencl")
sim_chr11.saveFolder('output_nucleus')
chr11 = sim_chr11.createSpringSpiral(ChromSeq='inputs/chr11_beads.txt')
sim_chr11.loadStructure(chr11, center=True)

sim_chr11.addFENEBonds(kfb=30.0)
sim_chr11.addAngles(ka=2.0)
sim_chr11.addRepulsiveSoftCore(Ecut=4.0)
sim_chr11.addFlatBottomHarmonic()

sim_chr11.addTypetoType(mu=3.22, rc=1.78)
sim_chr11.addIdealChromosome(mu=3.22, rc=1.78, dinit=3, dend=500)

print("Performing chr11 simulation...")
for _ in range(1000):
    sim_chr11.runSimBlock(500)

sim_chr11.saveStructure(filename="chr11", mode="ndb")
del sim_chr11
```

### 5.1.3 Multiple Chromosome Simulation

Once the collapse simulations are done, the collapsed structures of each chromosome should be included in the same simulation system.

```
[ ]: sim_nucleus = MiChroM(name="nucleus", temperature=1.0, time_step=0.01)
```

```
[ ]: sim_nucleus.setup(platform="opencl")
```

```
[ ]: sim_nucleus.saveFolder('output_nucleus')
```

The function `initStructure()` receives a list of files and saves the positions of the collapsed chromosomes in the variable `initial_conf`.

```
[ ]: initial_conf = sim_nucleus.initStructure(
    CoordFiles=['output_nucleus/chr10_0_block1000.ndb',
                'output_nucleus/chr11_0_block1000.ndb']
)
```

`initial_conf` contains the coordinates of all beads for each chromosome. When used to load coordinate's files (`.ndb`, `.pdb`, or `.gro`), the function `initStructure` prints the variable `chains`, which shows the information of each chromosome.

For each chain, we have `(start,end,is_ring)`. `start` is the index of the first bead of the chain, `end` is the index for the last bead and `is_ring` indicates whether the first and last bead are connect or not. `is_ring=0` represents a open chromosome chain and `is_ring=1` represents a circular polymer (used for simulating bacteria genome, for example).

```
[ ]: sim_nucleus.chains
```

Before loading the structures in the simulation context, it is necessary to spatially distribute the chromosome chains. This step is essential to guarantee no overlap between the chains in the initial configuration. It also helps randomizing the initial condition when simulating different replicas.

```
[ ]: initial_conf = sim_nucleus.setFibPosition(initial_conf, factor=1.5)
```

This function distributes the center of mass of each chain in a spherical shell, according to the Fibonacci Sphere Algorithm. The argument `factor` sets the radius of the spherical shell, in comparison to the radius of the nucleus (see function [documentation](#)). For simulations with just a few chromosomes, `factor=1.5` should be fine.

Now we can load the chromosomes with adjusted positions in the simulation context.

```
[ ]: sim_nucleus.loadStructure(initial_conf, center=True)
```

The initial configuration of the chromosome structures can be saved in the `.ndb file format`. The file is stored in the path set in `saveFolder`. We advise saving the structure and double-checking if there is no overlap between the chains.

When having multiple chains in the simulation context, the `saveStructure` function will save each chromosome structure in a different file. Each chain is associated with an index starting from 0. This index follows the order of addition of the chains in the `initStructure` function. In this tutorial, chromosome 10 receives index 0, and chromosome 11 index 1.

```
[ ]: sim_nucleus.saveStructure(mode='ndb')
```

The next step is to add the force field in the simulation object `sim_nucleus`.

In this tutorial, the forces can be divided into two sets:

#### MiChroM Homopolymer (Bonded) Potentials

```
[ ]: sim_nucleus.addFENEbonds(kfb=30.0)
sim_nucleus.addAngles(ka=2.0)
sim_nucleus.addRepulsiveSoftCore(Ecut=4.0)
sim_nucleus.addFlatBottomHarmonic(n_rad=20)
```

#### MiChroM Non-Bonded Potentials

The `addTypeToType` interaction is independent of the chromosome chain and only depends on the chromatin subcompartment annotation for the interacting loci. This potential is added for both chains simultaneously.

```
[ ]: sim_nucleus.addTypeToType(mu=3.22, rc=1.78)
```

On the other hand, for the Ideal Chromosome potential, we add the forces for each chromosome separately. The function `addMultiChainIC` receives the chromosome chain index and adds the IC potential.

The chromosome chain information can be obtained from the aforementioned variable `chains`. Note that the Ideal Chromosome potential was applied from the genomic distance  $d = 3$  to  $d = 500$ . These cutoff values can be adjusted based on the chromosome length and user needs.

```
[ ]: sim_nucleus.addMultiChainIC(chainIndex=0, mu=3.22, rc=1.78, dinit=3, dend=500)
sim_nucleus.addMultiChainIC(chainIndex=1, mu=3.22, rc=1.78, dinit=3, dend=500)
```

#### The simulation setup is complete!

Before running the production simulation, it is necessary to run a collapse and equilibration simulation as the chains are initially apart. The `addFlatBottomHarmonic` function used above includes in the simulation a harmonic potential to drive the chains together.

Set the parameters of the collapse simulation:

## 5.1. Chromatin Dynamics Simulations on Chromosome 10 and Chromosome 11 of GM12878 CelB1 Line

```
[ ]: block = 2*10**3
     n_blocks = 1*10**3
```

```
[ ]: for _ in range(n_blocks):
     sim_nucleus.runSimBlock(block, increment=False)
```

We can save the structure and check if the collapse was successful, with the two chains now interacting with each other.

```
[ ]: sim_nucleus.saveStructure(mode='ndb')
     sim_nucleus.saveStructure(mode='gro')
```

After the collapse, we should remove the harmonic potential and add the nucleus confinement.

```
[ ]: sim_nucleus.removeFlatBottomHarmonic()
     sim_nucleus.addAdditionalForce(sim_nucleus.addSphericalConfinementLJ)
```

The function `addAdditionalForce` adds a force to the system after the system has been already initialized. The initialization happens when the function `runSimBlock` is executed for the first time after the addition of the forces. In this case, we are using the function to add the Lennard-Jones spherical confinement (`addSphericalConfinementLJ`) in the system after removing the harmonic potential.

To run the production simulation, it is necessary to initialize the `.cndb` file to save the chromatin dynamics trajectory. The files will be saved in the output folder set using the `saveFolder` function. Each chromosome will be stored separately according to the chain index, as already mentioned.

```
[ ]: sim_nucleus.initStorage('traj_nucleus')
```

Set the parameters of the production simulations:

```
[ ]: block = 5*10**2
     n_blocks = 5*10**2
```

```
[ ]: for _ in range(n_blocks):
     sim_nucleus.runSimBlock(block, increment=True)
     sim_nucleus.saveStructure()
```

Once the simulation is performed, it is necessary to close the `.cndb` files.

```
[ ]: sim_nucleus.storage[0].close()
     sim_nucleus.storage[1].close()
```

To visualize the chromosome's 3D structures in the standard visualization softwares for macromolecules, there are available scripts for converting the `ndb/cndb` file formats to `.pdb` and `.gro`. For details, please check the [Nucleome Data Bank](#).

The `ndb` plugin for visualizing the chromatin dynamics trajectories in VMD/Chimera/Pymol is under development.

```
[ ]: sim_nucleus.saveStructure(mode="ndb")
     sim_nucleus.saveStructure(mode="gro")
```

See the [single chromosome simulation tutorial](#) for examples on how to use our simulation analysis class `cndbTools` <<https://open-michrom.readthedocs.io/en/latest/OpenMiChroM.html#module-OpenMiChroM.CndbTools>>`\_\_

## TUTORIAL: SINGLE CHROMOSOME OPTIMIZATION USING OPENMICHROM

### 6.1 This tutorial enables performing optimization in MiChroM Parameters (Second-order optimization -> Hessian inversion)

The first step is import the OpenMiChroM modules.

To install OpenMM and OpenMiChroM, follow the [instalation guide](#)

The inputs and apps used in this tutorial can be downloaded [here](#)

Types optimization is available in OpenMichroM version 1.0.5

```
[39]: from OpenMiChroM.ChromDynamics import MiChroM #OpenMiChroM simulation module
      from OpenMiChroM.Optimization import CustomMiChroMTraining #optimization MiChroM_
      ↪ parameters module
      from OpenMiChroM.CndbTools import cndbTools #analysis tools module

      #modules to load and plot .dense file
      import matplotlib.pyplot as plt
      import matplotlib as mpl
      from sklearn.preprocessing import normalize
      import numpy as np
      import pandas as pd
      import h5py
```

*The second step is to have a look on the experimental Hi-C*

A Hi-C file is required for the analysis and training of the MiChroM Potentials (Types and Ideal Chromosome). The file format chosen here is a matrix .txt file (we call it the dense file).

For this tutorial, we will use chromosome 10 from GM12878 in 100 kb resolution.

To extract it from the .hic file we can use `juicer_tools` with this command:

```
java -jar juicer_tools_1.22.01.jar dump observed NONE -d https://hicfiles.s3.amazonaws.com/hiseq/gm12878/in-situ/combined.hic 10 10 BP 100000 input/chr10_100k.dense
```

```
[2]: %bash
      java -jar apps/juicer_tools_1.22.01.jar dump observed Balanced -d https://hicfiles.s3.
      ↪ amazonaws.com/hiseq/gm12878/in-situ/combined.hic 10 10 BP 100000 input/chr10_100k.dense
```

```
WARNING: sun.reflect.Reflection.getCallerClass is not supported. This will impact
↳ performance.
WARN [2022-12-07T11:19:30,430] [Globals.java:138] [main] Development mode is enabled
INFO [2022-12-07T11:19:34,057] [DirectoryManager.java:179] [main] IGV Directory: /home/
↳ antonio/igv
INFO [2022-12-07T11:19:35,068] [HttpUtils.java:937] [main] Range-byte request succeeded
```

This command downloads the .hic from the web and extracts the chromosome 10 in .dense format to the folder “input”.

You can get more information about it at the JuicerTools documentation.

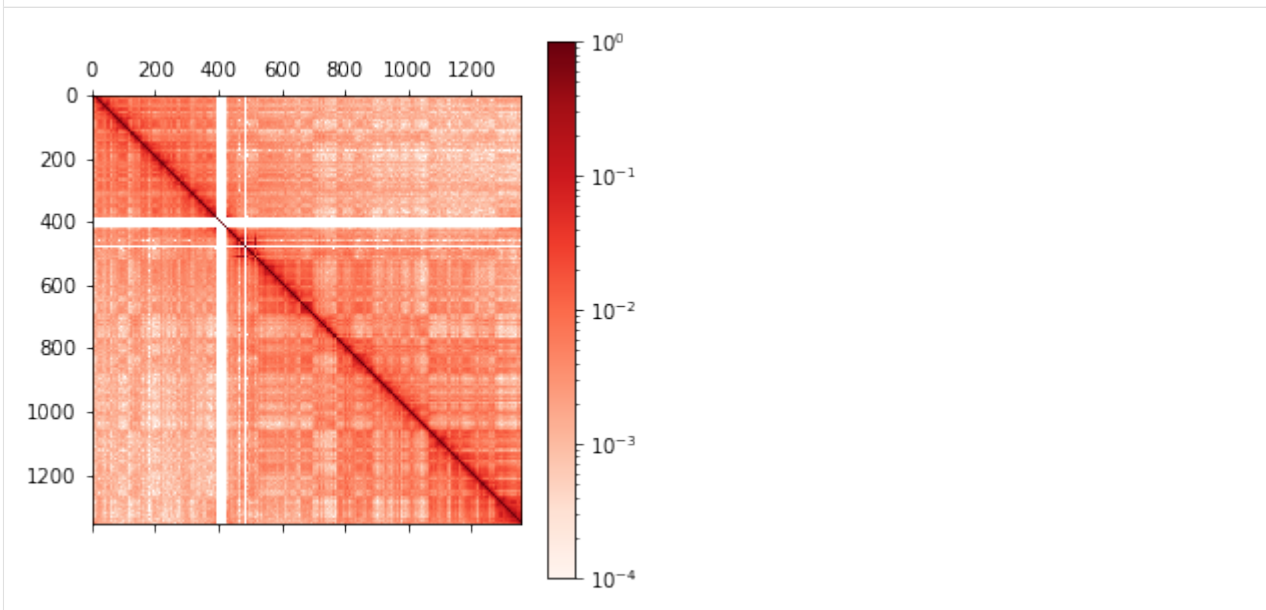
Visualize the .dense file for inspection

```
[3]: filename = 'input/chr10_100k.dense'
hic_file = np.loadtxt(filename)

r=np.triu(hic_file, k=1)
r[np.isnan(r)]= 0.0
r = normalize(r, axis=1, norm='max')
rd = np.transpose(r)
r=r+rd + np.diag(np.ones(len(r)))
print("number of beads: ", len(r))
plt.matshow(r,norm=matplotlib.colors.LogNorm(vmin=0.0001, vmax=r.max()),cmap="Reds")
plt.colorbar()
```

number of beads: 1356

```
[3]: <matplotlib.colorbar.Colorbar at 0x7fea4445bfd0>
```



The Hi-C map has resolution of 100kb per bead, so the chromosome 10 model has a polymer chain with a total of 1356 beads

The next step is to extract the sequence file (A/B sequence) by using the eigenvector decomposition.

Using the juicertools you can extract the eigenvector file. The eigenvector has values both negatives and positives and here we will arbitrary set positives as A1 and negatives as B1.

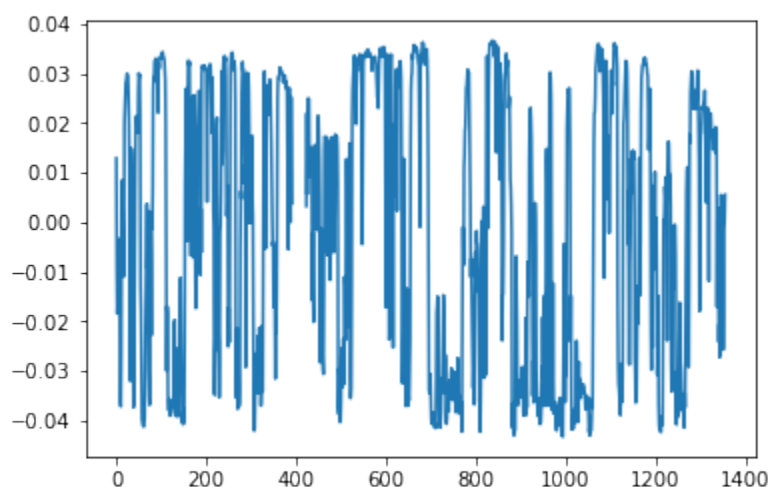
For more details about how it works, take a look on this paper: <https://pubs.acs.org/doi/full/10.1021/acs.jpcb.1c04174>

```
[4]: %%bash
java -jar apps/juicer_tools_1.22.01.jar eigenvector -p Balanced https://hicfiles.s3.
↳amazonaws.com/hiseq/gm12878/in-situ/combined.hic 10 BP 100000 input/chr10_100k.eigen

WARNING: sun.reflect.Reflection.getCallerClass is not supported. This will impact
↳performance.
WARN [2022-12-07T11:23:03,393] [Globals.java:138] [main] Development mode is enabled
INFO [2022-12-07T11:23:05,061] [DirectoryManager.java:179] [main] IGV Directory: /home/
↳antonio/igv
INFO [2022-12-07T11:23:06,008] [HttpUtils.java:937] [main] Range-byte request succeeded
```

```
[6]: eigen = np.loadtxt("input/chr10_100k.eigen")
plt.plot(eigen)
```

```
[6]: [<matplotlib.lines.Line2D at 0x7fe99b2d5090>]
```



From the .eigen file we can create the A/B sequence file.

```
[7]: %%bash
awk '{if ($1 < 0) print "B1"; else print "A1"}' input/chr10_100k.eigen | cat -n > input/
↳seq_chr10_100k.txt
head input/seq_chr10_100k.txt
```

```
1 A1
2 B1
3 B1
4 B1
5 B1
6 B1
7 B1
8 B1
9 B1
10 B1
```

The pipeline to perform the Types Potential Optimization is:

- 1 - Run a long simulation using the homopolymer potential + customTypes potential. The first iteration will start with all parameters equal to zero or a initial guess.
- 2 - Get frames from this simulation to perform the inversion for Types.

3 - In the end of inversion, new values to types interactions will be produced.

4 - Calculate the error between the simulated and experimental parameters. If the error is above the threshold, re-do steps 1-3 until reaching the threshold (usually 10% or 15%).

The Types file is a .txt file with a matrix labeled with values for each interaction. In this tutorial, we will be training A1 and B1 type.

Lets create the initial file at this format: A1.B1 0.0 0.0

For this matrix, we have AA AB BA BB interactions.

Save it as lambda\_0.txt

```
[8]: %bash
echo "A1,B1
0,0
0,0" > input/lambda_0

cat input/lambda_0

A1,B1
0,0
0,0
```

With all the required inputs, lets perform a simulation for iteration 0

In MiChroM initiation there are some variables to setup:

**time\_step=0.01** (the time step using for integration, default is 0.01) **temperature=1** (Set the temperature of your simulation) **name='opt chr10 100K'** (the simulation name)

```
[66]: sim = MiChroM(name='opt_chr10_100K', temperature=1.0, time_step=0.01)

*****
**** ** OpenMiChroM-1.0.5 ** ** ** **
****

OpenMiChroM is a Python library for performing chromatin dynamics simulations.
OpenMiChroM uses the OpenMM Python API,
employing the MiChroM (Minimal Chromatin Model) energy function.
The chromatin dynamics simulations generate an ensemble of 3D chromosomal
structures
that are consistent with experimental Hi-C maps, also allows simulations of a
single
or multiple chromosome chain using High-Performance Computing
in different platforms (GPUs and CPUs).
OpenMiChroM documentation is available at https://open-michrom.readthedocs.io

OpenMiChroM is described in: Oliveira Junior, A. B & Contessoto, V, G et. al.
A Scalable Computational Approach for Simulating Complexes of Multiple Chromosomes.
Journal of Molecular Biology. doi:10.1016/j.jmb.2020.10.034.
and
Oliveira Junior, A. B. et al.
Chromosome Modeling on Downsampled Hi-C Maps Enhances the Compartmentalization
Signal.
J. Phys. Chem. B, doi:10.1021/acs.jpcb.1c04174.
```

(continues on next page)





```
[74]: block = 5*10**2
      n_blocks = 10**3
```

We can save the radius of gyration of each block to observe the convergence into the collapsed state (the time required here depends on the size of your chromosome)

```
[75]: rg = []
```

```
[76]: for _ in range(n_blocks):
      sim.runSimBlock(block, increment=False)
      rg.append(sim.chromRG())
```

```
#save a collapsed structure in pdb format for inspection
sim.saveStructure(mode = 'pdb')
```

```
Number of exceptions: 1355
adding force FENEBond 0
adding force AngleForce 1
Add exclusions for RepulsiveSoftCore force
adding force RepulsiveSoftCore 2
Add exclusions for CustomTypes force
adding force CustomTypes 3
adding force FlatBottomHarmonic 4
Positions...
loaded!
potential energy is 31.227323
bl=0 pos[1]=[105.7 -3.8 0.6] dr=1.24 t=0.0ps kin=1.55 pot=31.48 Rg=72.954 SPS=16053
bl=0 pos[1]=[104.7 -3.3 2.7] dr=1.95 t=0.0ps kin=1.67 pot=31.18 Rg=71.676 SPS=17587
bl=0 pos[1]=[102.4 -4.1 4.1] dr=2.00 t=0.0ps kin=1.74 pot=30.82 Rg=70.219 SPS=17438
bl=0 pos[1]=[99.7 -6.1 5.9] dr=1.95 t=0.0ps kin=1.76 pot=30.50 Rg=68.733 SPS=17462
bl=0 pos[1]=[96.7 -5.9 6.4] dr=1.93 t=0.0ps kin=1.89 pot=30.09 Rg=67.232 SPS=17478
bl=0 pos[1]=[93.7 -5.8 7.8] dr=1.98 t=0.0ps kin=1.89 pot=29.68 Rg=65.681 SPS=17308
bl=0 pos[1]=[90.9 -5.0 7.4] dr=1.96 t=0.0ps kin=1.93 pot=29.34 Rg=64.178 SPS=20073
bl=0 pos[1]=[90.9 -5.7 6.8] dr=1.93 t=0.0ps kin=1.90 pot=28.93 Rg=62.698 SPS=20742
bl=0 pos[1]=[90.4 -5.2 7.1] dr=1.88 t=0.0ps kin=1.88 pot=28.57 Rg=61.274 SPS=21052
bl=0 pos[1]=[88.3 -5.7 5.6] dr=1.85 t=0.0ps kin=1.90 pot=28.18 Rg=59.906 SPS=21011
bl=0 pos[1]=[86.5 -6.3 4.5] dr=1.85 t=0.0ps kin=1.86 pot=27.88 Rg=58.504 SPS=20696
bl=0 pos[1]=[85.7 -6.0 4.8] dr=1.86 t=0.0ps kin=1.91 pot=27.55 Rg=57.112 SPS=20802
bl=0 pos[1]=[84.9 -5.2 5.0] dr=1.83 t=0.0ps kin=1.87 pot=27.24 Rg=55.765 SPS=20556
bl=0 pos[1]=[83.2 -3.1 4.6] dr=1.83 t=0.0ps kin=1.85 pot=26.95 Rg=54.445 SPS=20424
bl=0 pos[1]=[80.9 -1.8 4.1] dr=1.80 t=0.0ps kin=1.83 pot=26.61 Rg=53.144 SPS=20322
bl=0 pos[1]=[77.8 -0.5 4.9] dr=1.77 t=0.0ps kin=1.78 pot=26.34 Rg=51.895 SPS=20342
bl=0 pos[1]=[75.9 -0.1 5.4] dr=1.71 t=0.0ps kin=1.76 pot=26.12 Rg=50.727 SPS=19943
bl=0 pos[1]=[74.8 -0.5 6.2] dr=1.71 t=0.0ps kin=1.72 pot=25.86 Rg=49.591 SPS=20255
bl=0 pos[1]=[73.2 -3.1 6.0] dr=1.64 t=0.0ps kin=1.68 pot=25.67 Rg=48.524 SPS=20553
bl=0 pos[1]=[69.8 -3.3 4.6] dr=1.59 t=0.0ps kin=1.69 pot=25.46 Rg=47.505 SPS=20516
bl=0 pos[1]=[67.7 -1.6 2.8] dr=1.58 t=0.0ps kin=1.69 pot=25.25 Rg=46.517 SPS=20445
bl=0 pos[1]=[66.4 -1.7 3.2] dr=1.58 t=0.0ps kin=1.68 pot=25.08 Rg=45.541 SPS=20674
bl=0 pos[1]=[65.8 -1.9 2.3] dr=1.58 t=0.0ps kin=1.74 pot=24.88 Rg=44.574 SPS=20586
bl=0 pos[1]=[64.6 -3.0 2.1] dr=1.63 t=0.0ps kin=1.75 pot=24.74 Rg=43.588 SPS=20882
bl=0 pos[1]=[62.6 -1.6 2.6] dr=1.57 t=0.0ps kin=1.68 pot=24.58 Rg=42.626 SPS=20777
bl=0 pos[1]=[59.4 -0.7 1.9] dr=1.54 t=0.0ps kin=1.71 pot=24.43 Rg=41.684 SPS=21043
bl=0 pos[1]=[56.3 -0.1 2.6] dr=1.59 t=0.0ps kin=1.67 pot=24.23 Rg=40.783 SPS=20794
```

(continues on next page)

(continued from previous page)

```

bl=0 pos[1]=[56.4 1.3 2.5] dr=1.51 t=0.0ps kin=1.66 pot=24.13 Rg=39.934 SPS=20599
bl=0 pos[1]=[56.5 0.9 2.5] dr=1.50 t=0.0ps kin=1.68 pot=23.95 Rg=39.109 SPS=20443
bl=0 pos[1]=[54.5 1.3 1.8] dr=1.50 t=0.0ps kin=1.60 pot=23.83 Rg=38.300 SPS=20511
bl=0 pos[1]=[54.2 2.8 1.1] dr=1.47 t=0.0ps kin=1.55 pot=23.75 Rg=37.514 SPS=20346
bl=0 pos[1]=[51.4 3.3 0.5] dr=1.48 t=0.0ps kin=1.61 pot=23.61 Rg=36.738 SPS=20361
bl=0 pos[1]=[49.0 4.5 0.9] dr=1.46 t=0.0ps kin=1.61 pot=23.49 Rg=36.006 SPS=20278
bl=0 pos[1]=[48.1 5.6 2.6] dr=1.48 t=0.0ps kin=1.64 pot=23.39 Rg=35.257 SPS=21124
bl=0 pos[1]=[47.2 6.5 1.9] dr=1.47 t=0.0ps kin=1.60 pot=23.33 Rg=34.527 SPS=21190
bl=0 pos[1]=[47.2 7.4 3.3] dr=1.48 t=0.0ps kin=1.61 pot=23.21 Rg=33.826 SPS=21369
bl=0 pos[1]=[46.1 6.1 3.3] dr=1.45 t=0.0ps kin=1.61 pot=23.14 Rg=33.181 SPS=21381
bl=0 pos[1]=[44.4 5.7 2.2] dr=1.41 t=0.0ps kin=1.55 pot=23.09 Rg=32.575 SPS=20543
bl=0 pos[1]=[43.5 5.5 0.8] dr=1.42 t=0.0ps kin=1.57 pot=23.02 Rg=31.994 SPS=21699
bl=0 pos[1]=[43.3 4.5 1.3] dr=1.41 t=0.0ps kin=1.53 pot=22.91 Rg=31.410 SPS=22004
bl=0 pos[1]=[43.0 5.8 1.0] dr=1.39 t=0.0ps kin=1.53 pot=22.88 Rg=30.837 SPS=21872
bl=0 pos[1]=[41.0 5.3 1.0] dr=1.37 t=0.0ps kin=1.58 pot=22.84 Rg=30.284 SPS=21754
bl=0 pos[1]=[41.1 6.2 0.2] dr=1.39 t=0.0ps kin=1.58 pot=22.77 Rg=29.678 SPS=21526
bl=0 pos[1]=[38.8 7.6 1.4] dr=1.39 t=0.0ps kin=1.57 pot=22.67 Rg=29.088 SPS=21322
bl=0 pos[1]=[37.9 6.9 0.3] dr=1.39 t=0.0ps kin=1.53 pot=22.60 Rg=28.531 SPS=21037
bl=0 pos[1]=[35.4 6.7 -0.2] dr=1.36 t=0.0ps kin=1.51 pot=22.56 Rg=27.994 SPS=20126
bl=0 pos[1]=[35.2 6.2 -0.6] dr=1.34 t=0.0ps kin=1.54 pot=22.50 Rg=27.529 SPS=20043
bl=0 pos[1]=[35.8 6.1 -0.9] dr=1.37 t=0.0ps kin=1.55 pot=22.42 Rg=27.110 SPS=19473
bl=0 pos[1]=[36.2 4.8 -1.7] dr=1.35 t=0.0ps kin=1.57 pot=22.42 Rg=26.666 SPS=20077
bl=0 pos[1]=[34.2 3.6 -2.7] dr=1.37 t=0.0ps kin=1.53 pot=22.38 Rg=26.215 SPS=19428
bl=0 pos[1]=[32.5 3.4 -1.7] dr=1.37 t=0.0ps kin=1.55 pot=22.35 Rg=25.809 SPS=19760
bl=0 pos[1]=[31.7 4.7 0.5] dr=1.36 t=0.0ps kin=1.58 pot=22.31 Rg=25.365 SPS=19803
bl=0 pos[1]=[31.4 4.6 0.8] dr=1.34 t=0.0ps kin=1.47 pot=22.30 Rg=24.931 SPS=19562
bl=0 pos[1]=[32.0 3.3 1.3] dr=1.29 t=0.0ps kin=1.48 pot=22.20 Rg=24.557 SPS=19850
bl=0 pos[1]=[30.8 4.8 0.4] dr=1.27 t=0.0ps kin=1.48 pot=22.19 Rg=24.223 SPS=19773
bl=0 pos[1]=[30.3 5.9 -0.9] dr=1.27 t=0.0ps kin=1.45 pot=22.17 Rg=23.912 SPS=19531
bl=0 pos[1]=[29.3 6.3 -2.4] dr=1.24 t=0.0ps kin=1.51 pot=22.17 Rg=23.635 SPS=19462
bl=0 pos[1]=[27.8 7.0 -3.2] dr=1.29 t=0.0ps kin=1.52 pot=22.16 Rg=23.327 SPS=19794
bl=0 pos[1]=[26.9 6.3 -4.4] dr=1.34 t=0.0ps kin=1.56 pot=22.08 Rg=22.943 SPS=19187
bl=0 pos[1]=[25.5 6.3 -4.0] dr=1.35 t=0.0ps kin=1.54 pot=22.10 Rg=22.564 SPS=18919
bl=0 pos[1]=[24.4 5.4 -4.3] dr=1.30 t=0.0ps kin=1.54 pot=22.05 Rg=22.241 SPS=18440
bl=0 pos[1]=[25.0 4.8 -5.0] dr=1.30 t=0.0ps kin=1.50 pot=22.01 Rg=21.919 SPS=19122
bl=0 pos[1]=[24.6 4.5 -4.1] dr=1.28 t=0.0ps kin=1.50 pot=21.99 Rg=21.583 SPS=18991
bl=0 pos[1]=[25.8 4.4 -2.9] dr=1.31 t=0.0ps kin=1.54 pot=21.96 Rg=21.229 SPS=18880
bl=0 pos[1]=[25.0 2.8 -2.6] dr=1.30 t=0.0ps kin=1.51 pot=21.96 Rg=20.889 SPS=18808
bl=0 pos[1]=[23.7 1.5 -2.8] dr=1.34 t=0.0ps kin=1.53 pot=21.94 Rg=20.565 SPS=18725
bl=0 pos[1]=[23.0 1.1 -4.6] dr=1.33 t=0.0ps kin=1.58 pot=21.93 Rg=20.253 SPS=18933
bl=0 pos[1]=[23.7 1.2 -6.1] dr=1.30 t=0.0ps kin=1.54 pot=21.93 Rg=19.974 SPS=18705
bl=0 pos[1]=[23.7 2.1 -4.8] dr=1.29 t=0.0ps kin=1.55 pot=21.91 Rg=19.706 SPS=19759
bl=0 pos[1]=[24.8 3.6 -4.2] dr=1.29 t=0.0ps kin=1.59 pot=21.90 Rg=19.438 SPS=20760
bl=0 pos[1]=[23.9 3.7 -4.2] dr=1.30 t=0.0ps kin=1.55 pot=21.93 Rg=19.167 SPS=21955
bl=0 pos[1]=[22.8 4.1 -3.1] dr=1.29 t=0.0ps kin=1.57 pot=21.89 Rg=18.924 SPS=19747
bl=0 pos[1]=[22.8 3.5 -2.3] dr=1.29 t=0.0ps kin=1.54 pot=21.84 Rg=18.697 SPS=18997
bl=0 pos[1]=[22.2 0.7 -3.3] dr=1.30 t=0.0ps kin=1.51 pot=21.83 Rg=18.461 SPS=18783
bl=0 pos[1]=[19.7 -1.3 -2.3] dr=1.32 t=0.0ps kin=1.58 pot=21.83 Rg=18.232 SPS=18798
bl=0 pos[1]=[18.5 -0.1 -0.7] dr=1.30 t=0.0ps kin=1.51 pot=21.81 Rg=18.067 SPS=18970
bl=0 pos[1]=[18.5 1.2 0.3] dr=1.27 t=0.0ps kin=1.52 pot=21.81 Rg=17.905 SPS=18708
bl=0 pos[1]=[19.4 1.4 0.2] dr=1.28 t=0.0ps kin=1.50 pot=21.77 Rg=17.738 SPS=19149
bl=0 pos[1]=[20.1 0.8 0.3] dr=1.29 t=0.0ps kin=1.51 pot=21.80 Rg=17.597 SPS=19050

```

(continues on next page)

(continued from previous page)

```
bl=0 pos[1]=[21.8 0.5 -0.1] dr=1.30 t=0.0ps kin=1.49 pot=21.80 Rg=17.476 SPS=19346
bl=0 pos[1]=[21.3 1.7 0.8] dr=1.29 t=0.0ps kin=1.54 pot=21.78 Rg=17.349 SPS=17803
...bl=0 pos[1]=[8.3 -5.0 6.4] dr=1.26 t=0.0ps kin=1.54 pot=21.58 Rg=11.712 SPS=18526
```

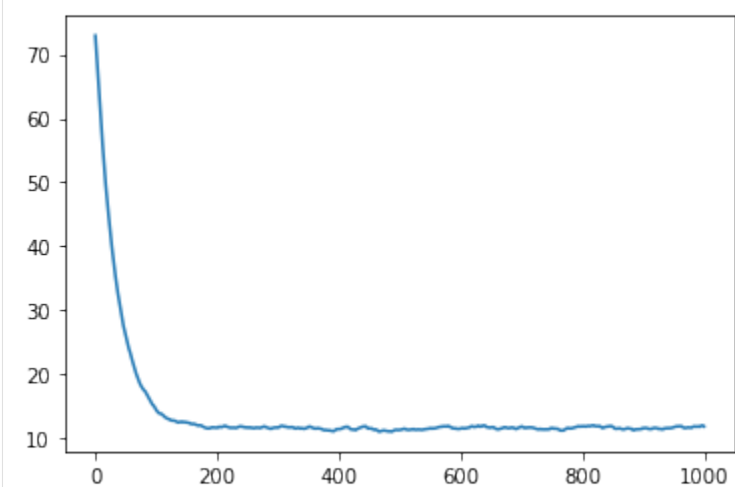
Some details about the output for each block performed:

**bl=0** is the number of each block ran, in this case we set increment=False, so the number of steps is not accounted. **pos[1]=[X,Y,Z]** the spatial position for the first bead. **dr=1.26** show the average positions displacement in each block (in units of sigma). **t=0** the time step. in this case we set increment=False, so the number of steps is not accounted. **kin=1.5** is the kinetic energy of the system. **pot=19.90** is the potential energy of the system. **RG=7.654** is the radius of gyration in the end of each block. **SPS=12312** is the steps per second of each block. A way to look how fast the computations are being performed.

Check the convergence of the radius of gyration:

```
[77]: plt.plot(rg)
```

```
[77]: [<matplotlib.lines.Line2D at 0x7f1325645dc0>]
```



The next step is to remove the restraint force in order to run the sampling simulation

```
[78]: sim.removeFlatBottomHarmonic()
```

Initiate the optimization object. In this tutorial, this object is named “opt”

```
[79]: opt = CustomMiChroMTraining(ChromSeq="input/seq_chr10_100k.txt",
                                   mu=3.22, rc = 1.78)
```

The next step is to perform a long simulation to feed the optimization parameters.

In order to get a good inversion calculation, it is important to have around  $1 \times 10^5$  frames from different replicas. For example, 20 replicas of 5000 saved frames.

This can take some time, so in this tutorial we will use just 1 replica of 5000 frames saved every 1000 steps.

```
block = 1000 n_blocks = 5000
```

```
[80]: block = 1000
      n_blocks = 5000
```

```
[ ]: for _ in range(n_blocks):
    sim.runSimBlock(block, increment=True) #perform 1 block of the simulation
    opt.probCalculation_types(sim.getPositions()) #feed the optimization with the last_
    ↪ position
```

In the end of each replica simulation we need to save some important values required to calculate the inversion.

We are saving these files using the H5 compression because it is faster to write/read.

Note: attention to this step. We have these 4 files for each replica for each iteration step. Be organized!

```
[82]: rep=1

with h5py.File(sim.folder + "/polds_type_" + str(rep)+".h5", 'w') as hf:
    hf.create_dataset("polds_type", data=opt.polds_type)

with h5py.File(sim.folder + "/Bij_type_" + str(rep)+".h5", 'w') as hf:
    hf.create_dataset("Bij_type", data=opt.Bij_type)

with h5py.File(sim.folder + "/Nframes_" + str(rep)+".h5", 'w') as hf:
    hf.create_dataset("Nframes", data=opt.Nframes)

with h5py.File(sim.folder + "/Pold_" + str(rep)+".h5", 'w') as hf:
    hf.create_dataset("Pold", data=opt.Pold)
```

The first part of the optimization is finished. Inside the output folder, for each iteration, there are these 4 files used in next step.

```
[83]: %%bash
ls iteration_0/*.h5

iteration_0/Bij_type_1.h5
iteration_0/Nframes_1.h5
iteration_0/Pold_1.h5
iteration_0/polds_type_1.h5
```

The second part is the inversion, it is quite simple, just feed the optimization object with all replicas and make the inversion to get the new lambda file.

```
[84]: opt2 = CustomMiChroMTraining(ChromSeq="input/seq_chr10_100k.txt",
                                   mu=3.22, rc = 1.78)
```

```
[85]: iterations = "iteration_0"
replicas=[1]
for replica in replicas:
    with h5py.File(iterations + "/polds_type_" + str(replica)+".h5", 'r') as hf:
        opt2.polds_type += hf['polds_type'][:]

    with h5py.File(iterations + "/Bij_type_" + str(replica)+".h5", 'r') as hf:
        opt2.Bij_type += hf['Bij_type'][:]

    with h5py.File(iterations + "/Nframes_" + str(replica)+".h5", 'r') as hf:
        opt2.Nframes += hf['Nframes'][:]
```

(continues on next page)

(continued from previous page)

```
with h5py.File(iterations + "/Pold_" + str(replica)+".h5", 'r') as hf:
    opt2.Pold += hf['Pold'][:]
```

With the parameters fed with all replicas, we calculate the inversion and get the new lambdas

```
[86]: lambdas = opt2.getLamb_types(exp_map="input/chr10_100k.dense")
print(lambdas)

[[823304.95121223 934817.42279105]
 [934817.42279105 398513.55099684]]
```

Now load the old lambda value to calculate the new (lambda\_1, in this case) according the Newton Method:

$$\lambda_1 = \lambda_0 - \delta \times \lambda_{actual}$$

$\delta$  is the learning rate or damp, it can be adjusted in order to get values between [-1:0]. The average value of MiChroM's parameters for Human GM12878 is -0.3

```
[87]: old = pd.read_csv("input/lambda_0", sep=None, engine='python')
lambda_old = old.values
seq = old.columns
print(lambda_old)

[[0 0]
 [0 0]]
```

```
[92]: damp = 3*10**-7
lambda_new = lambda_old - damp*lambdas
```

```
[93]: print(lambda_new)

[[-0.24699149 -0.28044523]
 [-0.28044523 -0.11955407]]
```

Save the new lambda file (lambda\_1.txt) and some files to analyze the inversion

```
[94]: iteration = 0
#prob of A/B in sim and exp
phi_sim = opt2.calc_sim_phi_types().ravel()
phi_exp = opt2.calc_exp_phi_types().ravel()
np.savetxt('phi_sim_' + str(iteration), phi_sim)
np.savetxt('phi_exp', phi_exp)

plt.plot(phi_sim, 'o-', label="sim")
plt.plot(phi_exp, 'o-', label="exp")
plt.legend()

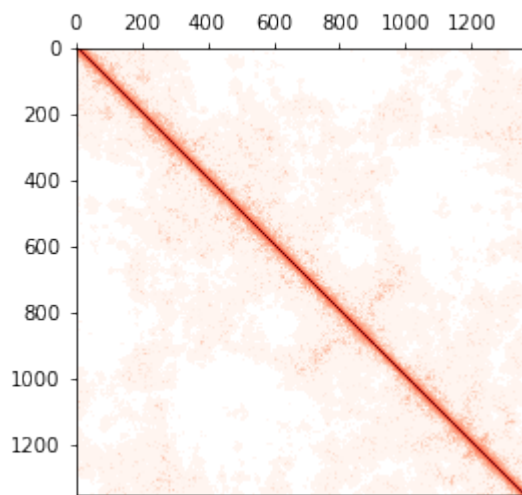
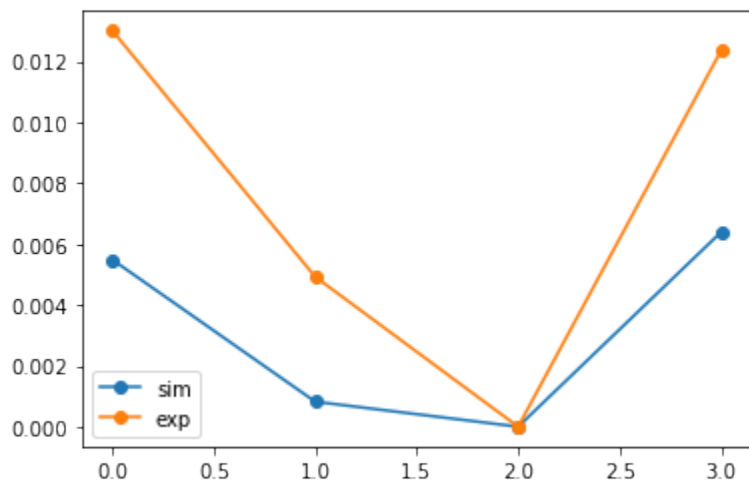
#HiC_simulate
dense_sim = opt2.getHiCSim()
np.savetxt('hic_sim_' + str(iteration)+'.dense', dense_sim)
plt.matshow(dense_sim, norm=mpl.colors.LogNorm(vmin=0.0001, vmax=dense_sim.max()), cmap=
↪ "Reds")

#save the new lambda file
```

(continues on next page)

(continued from previous page)

```
lamb = pd.DataFrame(lambda_new, columns=seq)
lamb.to_csv("input/lambda_1", index=False)
```



Redo these steps using the new lambda file (lambda\_1) as input for Types potential in the next iteration.

The error can be calculate using phi\_sim\_1 and phi\_exp by the equation:  $error = \frac{|\phi_{sim} - \phi_{exp}|}{\phi_{exp}}$

This is appended in the file “error\_and\_pearsonC\_types”

```
[95]: %%bash
cat error_and_pearsonC_types

Error: 0.607774 Pearson's Correlation: 0.814562
Error: 0.615568 Pearson's Correlation: 0.854520
Error: 0.617701 Pearson's Correlation: 0.854046
Error: 0.619801 Pearson's Correlation: 0.854452
Error: 0.621203 Pearson's Correlation: 0.853993
```

We included a folder named “scripts” that has some .py and .slurm files that can be used to run this optimization in parallel using slurm clusters.





## TUTORIAL: FULL INVERSION METHOD FOR A SINGLE CHROMOSOME USING OPENMICHROM

### 7.1 This tutorial enables performing Full Inversion Method using Adam (First-order optimization)

The first step is to import the OpenMiChroM modules.

To install OpenMM and OpenMiChroM, follow the [instalation guide](#)

The inputs and apps used in this tutorial can be downloaded [here](#)

Adam optimization is available in OpenMichroM version 1.0.5

```
[1]: from OpenMiChroM.ChromDynamics import MiChroM # OpenMiChroM simulation module
from OpenMiChroM.Optimization import AdamTraining # Adam optimization module
from OpenMiChroM.CndbTools import cndbTools # Analysis' tools module

# modules to load and plot
import matplotlib.pyplot as plt
import matplotlib as mpl
from sklearn.preprocessing import normalize
import numpy as np
import pandas as pd
import h5py
```

To create a model trained by the Full Inversion Method using Adam optimization 3 files are required:

1. The experimental Hi-C matrix in text format (dense file)
2. The sequence file
3. Initial force field

In the next steps we will create/extract theses files.

#### 1. Extract the experimental Hi-C matrix

A Hi-C file is required for the analysis and training of the Full inversion optimization. The file format chosen here is a matrix .txt file (we will call it the dense file).

For this tutorial, we will use chromosome 10 from GM12878 cell line in 100 kb resolution.

To extract it from the .hic file we can use `juicer_tools` with this command:

```
java -jar juicer_tools_1.22.01.jar dump observed Balanced -d https://hicfiles.s3.amazonaws.com/hiseq/
gm12878/in-situ/combined.hic 10 10 BP 100000 input/chr10_100k.dense
```

```
[5]: %%bash
java -jar apps/juicer_tools_1.22.01.jar dump observed Balanced -d https://hicfiles.s3.
↳amazonaws.com/hiseq/gm12878/in-situ/combined.hic 10 10 BP 100000 input/chr10_100k.dense

WARN [2022-09-14T13:46:43,872] [Globals.java:138] [main] Development mode is enabled
INFO [2022-09-14T13:46:47,853] [DirectoryManager.java:179] [main] IGV Directory: /
↳Users/mm146/igv
INFO [2022-09-14T13:46:48,195] [HttpUtils.java:937] [main] Range-byte request succeeded
```

This command accesses part of the .hic file from the web and extracts the chromosome 10 in .dense format to the folder “input”.

You can get more information about it at the [JuicerTools documentation](#).

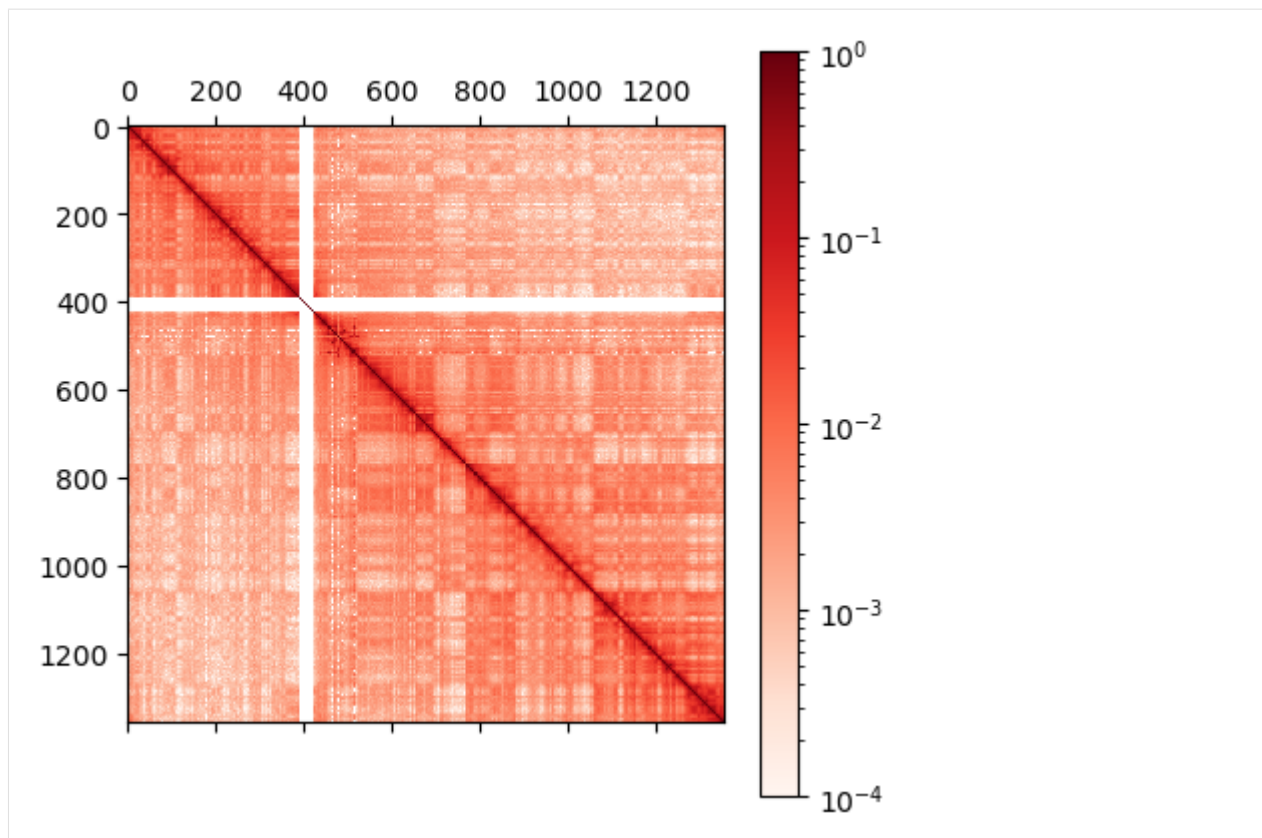
Visualize the .dense file for inspection

```
[6]: filename = 'input/chr10_100k.dense'
hic_file = np.loadtxt(filename)

r=np.triu(hic_file, k=1)
r[np.isnan(r)]= 0.0
r = normalize(r, axis=1, norm='max')
rd = np.transpose(r)
r=r+rd + np.diag(np.ones(len(r)))
print("number of beads: ", len(r))
plt.matshow(r,norm=mpl.colors.LogNorm(vmin=0.0001, vmax=r.max()),cmap="Reds")
plt.colorbar()

number of beads: 1356

[6]: <matplotlib.colorbar.Colorbar at 0x7f81cf0f90d0>
```



The Hi-C map has a resolution of 100 kb per bead, so this chromosome 10 model has a polymer chain with a total of 1356 beads.

## 2. Create the sequence file

The second required input is the sequence file.

The sequence file is a 2 columns text file that labels each bead of your polymer chain.

In OpenMiChroM, each bead must be assigned a chromatin type. In the full inversion method each pair of interaction is trained individually and therefore each bead is unique.

To create the sequence file for chr10 (1356 beads) use the following bash command:

```
[9]: %%bash
rm input/seq_chr10
for i in {1..1356}; do echo "$i Bead$i" >> input/seq_chr10; done
```

This command creates a file inside the folder “input” named seq\_chr10.

The first column is the index for each bead and the second column is the flavor (or type) for each bead. In this tutorial, we chose the names as beadX, where X is the index. You can choose any name for the beads as long as each name is unique. Additionally, you will need to have all names in the force field file (next step).

```
[10]: %%bash
head input/seq_chr10

1 Bead1
2 Bead2
3 Bead3
```

(continues on next page)

(continued from previous page)

```

4 Bead4
5 Bead5
6 Bead6
7 Bead7
8 Bead8
9 Bead9
10 Bead10

```

### 3. Create the initial force field file

Last but not least, we will create the third required file, the initial force field. The force field is represented by a  $N \times N$  matrix, where it pair  $(i, j)$  is the interaction between beads  $i$  and  $j$ .

This file format is a csv-file with 1 header line, delimited by comma (“,”).

To be simple, as a tutorial must be, the initial force field will have all entries equal to 0.

(In the end of this tutorial, we present a smarter method to fix certain interactions and to solve the centromere problem)

```

[11]: #read the sequence file
seq = np.loadtxt("input/seq_chr10", dtype=str)[: ,1]
pol_size = len(seq)
#create a NxN matrix fill with zeros, where N is the chromossome lenght
data = np.zeros((pol_size,pol_size))
#transform it in a Pandas dataframe using the sequence names as header
lamb = pd.DataFrame(data, columns=seq)
#save to a csv file format in the "input" folder
lamb.to_csv("input/lambda_1", index=False)
lamb

```

```

[11]:
      Bead1  Bead2  Bead3  Bead4  Bead5  Bead6  Bead7  Bead8  Bead9  Bead10  \
0      0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
1      0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
2      0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
3      0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
4      0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
...      ...    ...    ...    ...    ...    ...    ...    ...    ...    ...
1351    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
1352    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
1353    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
1354    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
1355    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0

      ...  Bead1347  Bead1348  Bead1349  Bead1350  Bead1351  Bead1352  \
0      ...      0.0      0.0      0.0      0.0      0.0      0.0
1      ...      0.0      0.0      0.0      0.0      0.0      0.0
2      ...      0.0      0.0      0.0      0.0      0.0      0.0
3      ...      0.0      0.0      0.0      0.0      0.0      0.0
4      ...      0.0      0.0      0.0      0.0      0.0      0.0
...      ...      ...      ...      ...      ...      ...
1351    ...      0.0      0.0      0.0      0.0      0.0      0.0
1352    ...      0.0      0.0      0.0      0.0      0.0      0.0
1353    ...      0.0      0.0      0.0      0.0      0.0      0.0
1354    ...      0.0      0.0      0.0      0.0      0.0      0.0
1355    ...      0.0      0.0      0.0      0.0      0.0      0.0

```

(continues on next page)

```
[1356 rows x 1356 columns]
```

|1d2e2c7c29dd4927ba9b78d76a3c9643|

Here we will show how to run a single chromosome simulation and run one iteration of the Adam optimization procedure.

**time\_step=0.01** (the time step used for integration, default is 0.01) **temperature=1** (set the temperature of your simulation) **name='opt\_chr10\_100K'** (the simulation name)

```
*****  
**** OpenMiChroM-1.0.5 ****  
***
```

OpenMiChroM is a Python library for performing chromatin dynamics simulations.  
    OpenMiChroM uses the OpenMM Python API,  
        employing the MiChroM (Minimal Chromatin Model) energy function.

The chromatin dynamics simulations generate an ensemble of 3D chromosomal  
→ structures  
that are consistent with experimental Hi-C maps, also allows simulations of a  
→ single  
or multiple chromosome chain using High-Performance Computing  
in different platforms (GPUs and CPUs).

OpenMiChroM documentation is available at <https://open-michrom.readthedocs.io>

OpenMiChroM is described in: Oliveira Junior, A. B & Contessoto, V, G et. al.  
A Scalable Computational Approach for Simulating Complexes of Multiple Chromosomes.  
Journal of Molecular Biology. doi:[10.1016/j.jmb.2020.10.034](#).  
and  
Oliveira Junior, A. B. et al.

(continues on next page)

(continued from previous page)

Chromosome Modeling on Downsampled Hi-C Maps Enhances the Compartmentalization  
 Signal.  
 J. Phys. Chem. B, doi:10.1021/acs.jpcb.1c04174.  
 Copyright (c) 2022, The OpenMiChroM development team at  
 Rice University  
 \*\*\*\*\*

Now we need to setup the platform that we will use. The options are:

**platform="cuda"** (remember that you need to install CUDA in your system) **GPU="0"** (optional) (if you have more than one GPU device, you can set which gpu you want ["0", "1", ..., "n"]) **platform="cpu"** **platform="opencl"**

```
[3]: sim.setup(platform="opencl")
```

Set the folder name where the output will be saved

```
[4]: sim.saveFolder('iteration_1')
```

The next step is to setup your chromosome sequence and the initial configuration

```
[5]: mychro = sim.createSpringSpiral(ChromSeq="input/seq_chr10")
```

Load the initial structure into the "sim" object

```
[6]: sim.loadStructure(mychro, center=True)
```

Now it is time to include the force field in the simulation object "sim".

Let's separate the forces in two sets:

### Homopolymer Potentials

```
[7]: sim.addFENEbonds(kfb=30.0)
sim.addAngles(ka=2.0)
sim.addRepulsiveSoftCore(Ecut=4.0)
```

### Chromosome Potential

In this tutorial, we will use the CustomTypes potential. Here you need to insert the force field file that contains a matrix of interactions for each bead. For each iteration of the optimization of the parameters, a new force field file will be created.

To check that, you can look on the documentation <https://open-michrom.readthedocs.io/en/latest/OpenMiChroM.html#OpenMiChroM.ChromDynamics.MiChroM.addCustomTypes>

```
[8]: sim.addCustomTypes(mu=3.22, rc = 1.78, TypesTable='input/lambda_1')
```

Note: these values for  $\mu$  and  $r_c$  were calculated for human GM12878 cells and can be changed for other species.

The last potential to be added is the spherical restraint in order to collapse the initial structure

```
[9]: sim.addFlatBottomHarmonic(kr=5*10**-3, n_rad=8.0)
```

Now we will run a short simulation in order to get a collapsed structure.

There are two variables that control the chromosome simulation steps:

**block:** The number of time steps performed in each cycle (n\_blocks) **n\_blocks:** The number of blocks that will be performed.

In this example, to perform the collapse we will run  $5 \times 10^2 \times 10^3 = 5 \times 10^5$  time steps

```
[10]: block = 5*10**2
      n_blocks = 10**3
```

We can save the radius of gyration of each block to observe the convergence into the collapsed state (the time required here depends on the size of your chromosome)

```
[11]: rg = []
```

```
[12]: for _ in range(n_blocks):
      sim.runSimBlock(block, increment=False)
      rg.append(sim.chromRG())

      #save the collapsed structure in pdb format for inspection
      sim.saveStructure(mode = 'pdb')
```

```
Number of exceptions: 1355
adding force  FENEBond 0
adding force  AngleForce 1
Add exclusions for RepulsiveSoftCore force
adding force  RepulsiveSoftCore 2
Add exclusions for CustomTypes force
adding force  CustomTypes 3
adding force  FlatBottomHarmonic 4
Positions...
loaded!
potential energy is 31.227323
bl=0 pos[1]=[103.8 -6.9 1.7] dr=1.22 t=0.0ps kin=1.51 pot=31.48 Rg=73.005 SPS=2934
bl=0 pos[1]=[103.3 -6.6 4.0] dr=1.86 t=0.0ps kin=1.61 pot=31.24 Rg=71.826 SPS=14103
bl=0 pos[1]=[101.9 -6.3 5.2] dr=1.91 t=0.0ps kin=1.72 pot=30.91 Rg=70.422 SPS=14505
bl=0 pos[1]=[100.7 -6.0 4.7] dr=1.96 t=0.0ps kin=1.85 pot=30.57 Rg=68.921 SPS=13752
bl=0 pos[1]=[99.3 -4.6 4.8] dr=1.98 t=0.0ps kin=1.89 pot=30.12 Rg=67.367 SPS=14440
bl=0 pos[1]=[97.6 -2.7 6.1] dr=1.97 t=0.0ps kin=1.93 pot=29.76 Rg=65.807 SPS=15452
bl=0 pos[1]=[94.9 -0.6 7.3] dr=1.95 t=0.0ps kin=1.88 pot=29.37 Rg=64.273 SPS=15480
bl=0 pos[1]=[92.4 -0.5 7.3] dr=1.90 t=0.0ps kin=1.90 pot=28.95 Rg=62.808 SPS=17235
bl=0 pos[1]=[90.7 -0.5 8.5] dr=1.88 t=0.0ps kin=1.85 pot=28.62 Rg=61.385 SPS=17314
bl=0 pos[1]=[87.8 -0.5 9.4] dr=1.84 t=0.0ps kin=1.86 pot=28.22 Rg=60.033 SPS=17292
bl=0 pos[1]=[84.3 0.1 9.3] dr=1.85 t=0.0ps kin=1.84 pot=27.90 Rg=58.697 SPS=16504
bl=0 pos[1]=[81.1 -0.8 10.5] dr=1.82 t=0.0ps kin=1.83 pot=27.56 Rg=57.382 SPS=17570
bl=0 pos[1]=[80.2 -0.7 11.4] dr=1.84 t=0.0ps kin=1.83 pot=27.28 Rg=56.065 SPS=17480
bl=0 pos[1]=[76.9 -0.5 12.9] dr=1.81 t=0.0ps kin=1.75 pot=26.99 Rg=54.792 SPS=17399
bl=0 pos[1]=[74.3 -1.3 13.2] dr=1.76 t=0.0ps kin=1.81 pot=26.73 Rg=53.565 SPS=17461
bl=0 pos[1]=[72.6 -2.9 13.8] dr=1.75 t=0.0ps kin=1.77 pot=26.44 Rg=52.331 SPS=17060
bl=0 pos[1]=[71.5 -3.3 13.5] dr=1.75 t=0.0ps kin=1.74 pot=26.15 Rg=51.107 SPS=17400
bl=0 pos[1]=[68.8 -3.8 11.8] dr=1.68 t=0.0ps kin=1.72 pot=25.95 Rg=49.951 SPS=17750
bl=0 pos[1]=[66.9 -4.5 9.5] dr=1.65 t=0.0ps kin=1.72 pot=25.77 Rg=48.878 SPS=17742
bl=0 pos[1]=[64.2 -3.6 8.1] dr=1.65 t=0.0ps kin=1.75 pot=25.55 Rg=47.822 SPS=16747
bl=0 pos[1]=[61.7 -4.1 7.5] dr=1.66 t=0.0ps kin=1.74 pot=25.36 Rg=46.750 SPS=17638
bl=0 pos[1]=[60.6 -6.5 6.7] dr=1.62 t=0.0ps kin=1.71 pot=25.15 Rg=45.736 SPS=17850
bl=0 pos[1]=[60.4 -5.4 5.8] dr=1.59 t=0.0ps kin=1.73 pot=24.94 Rg=44.822 SPS=18043
```

(continues on next page)

(continued from previous page)

```

bl=0 pos[1]=[58.9 -5.2 5.9] dr=1.58 t=0.0ps kin=1.68 pot=24.76 Rg=43.940 SPS=17832
bl=0 pos[1]=[56.6 -6.0 7.6] dr=1.57 t=0.0ps kin=1.69 pot=24.60 Rg=43.033 SPS=17849
bl=0 pos[1]=[55.1 -6.4 9.9] dr=1.57 t=0.0ps kin=1.61 pot=24.47 Rg=42.106 SPS=17763
bl=0 pos[1]=[52.5 -6.8 9.0] dr=1.55 t=0.0ps kin=1.62 pot=24.29 Rg=41.198 SPS=17837
bl=0 pos[1]=[51.0 -6.2 8.0] dr=1.54 t=0.0ps kin=1.67 pot=24.18 Rg=40.336 SPS=17422
bl=0 pos[1]=[48.7 -5.3 7.5] dr=1.56 t=0.0ps kin=1.63 pot=24.06 Rg=39.490 SPS=17423
bl=0 pos[1]=[46.6 -4.3 7.5] dr=1.54 t=0.0ps kin=1.62 pot=23.91 Rg=38.646 SPS=15926
bl=0 pos[1]=[44.4 -3.6 7.2] dr=1.51 t=0.0ps kin=1.61 pot=23.80 Rg=37.834 SPS=16716
bl=0 pos[1]=[42.9 -2.7 8.8] dr=1.51 t=0.0ps kin=1.61 pot=23.65 Rg=36.994 SPS=17139
bl=0 pos[1]=[43.0 -3.7 9.1] dr=1.50 t=0.0ps kin=1.56 pot=23.51 Rg=36.191 SPS=17881
bl=0 pos[1]=[43.1 -3.6 9.2] dr=1.49 t=0.0ps kin=1.60 pot=23.43 Rg=35.413 SPS=16878
bl=0 pos[1]=[43.1 -3.9 10.7] dr=1.47 t=0.0ps kin=1.63 pot=23.29 Rg=34.688 SPS=16727
bl=0 pos[1]=[44.2 -3.5 10.3] dr=1.43 t=0.0ps kin=1.60 pot=23.21 Rg=34.001 SPS=17193
bl=0 pos[1]=[43.0 -3.0 10.9] dr=1.44 t=0.0ps kin=1.53 pot=23.17 Rg=33.324 SPS=17711
bl=0 pos[1]=[41.3 -2.6 11.7] dr=1.41 t=0.0ps kin=1.58 pot=23.07 Rg=32.712 SPS=17386
bl=0 pos[1]=[40.7 -3.8 12.6] dr=1.41 t=0.0ps kin=1.58 pot=23.00 Rg=32.084 SPS=16220
bl=0 pos[1]=[40.2 -5.3 12.9] dr=1.41 t=0.0ps kin=1.53 pot=22.89 Rg=31.436 SPS=16462
bl=0 pos[1]=[39.1 -6.7 13.5] dr=1.41 t=0.0ps kin=1.57 pot=22.81 Rg=30.814 SPS=17130
bl=0 pos[1]=[38.7 -5.6 13.3] dr=1.39 t=0.0ps kin=1.55 pot=22.76 Rg=30.207 SPS=17235
bl=0 pos[1]=[38.4 -4.0 12.6] dr=1.40 t=0.0ps kin=1.56 pot=22.69 Rg=29.637 SPS=16268
bl=0 pos[1]=[36.8 -2.1 13.1] dr=1.38 t=0.0ps kin=1.55 pot=22.68 Rg=29.117 SPS=18108
bl=0 pos[1]=[36.0 0.1 13.2] dr=1.40 t=0.0ps kin=1.57 pot=22.60 Rg=28.609 SPS=17989
bl=0 pos[1]=[33.4 1.6 12.9] dr=1.39 t=0.0ps kin=1.56 pot=22.54 Rg=28.120 SPS=18060
bl=0 pos[1]=[30.6 2.2 14.2] dr=1.37 t=0.0ps kin=1.48 pot=22.50 Rg=27.635 SPS=17033
bl=0 pos[1]=[29.9 2.8 14.4] dr=1.39 t=0.0ps kin=1.55 pot=22.43 Rg=27.117 SPS=16332
bl=0 pos[1]=[29.5 2.4 14.0] dr=1.36 t=0.0ps kin=1.51 pot=22.43 Rg=26.613 SPS=17331
bl=0 pos[1]=[30.8 1.9 13.2] dr=1.32 t=0.0ps kin=1.55 pot=22.38 Rg=26.133 SPS=17662
bl=0 pos[1]=[32.0 1.7 13.8] dr=1.36 t=0.0ps kin=1.58 pot=22.33 Rg=25.700 SPS=17185
bl=0 pos[1]=[31.7 2.6 13.9] dr=1.31 t=0.0ps kin=1.52 pot=22.30 Rg=25.307 SPS=15488
bl=0 pos[1]=[30.2 2.1 15.3] dr=1.34 t=0.0ps kin=1.54 pot=22.25 Rg=24.873 SPS=16077
bl=0 pos[1]=[29.4 1.7 13.5] dr=1.33 t=0.0ps kin=1.52 pot=22.22 Rg=24.431 SPS=15575
bl=0 pos[1]=[31.1 2.0 12.2] dr=1.32 t=0.0ps kin=1.52 pot=22.19 Rg=24.015 SPS=15193
bl=0 pos[1]=[31.7 1.6 12.6] dr=1.30 t=0.0ps kin=1.53 pot=22.19 Rg=23.595 SPS=15574
bl=0 pos[1]=[31.6 1.1 14.4] dr=1.34 t=0.0ps kin=1.50 pot=22.12 Rg=23.179 SPS=16172
bl=0 pos[1]=[29.2 0.6 15.1] dr=1.31 t=0.0ps kin=1.47 pot=22.09 Rg=22.806 SPS=15418
bl=0 pos[1]=[27.2 1.9 12.8] dr=1.28 t=0.0ps kin=1.51 pot=22.07 Rg=22.433 SPS=16059
bl=0 pos[1]=[24.8 3.5 12.5] dr=1.33 t=0.0ps kin=1.55 pot=22.01 Rg=22.041 SPS=15369
bl=0 pos[1]=[23.3 4.5 12.7] dr=1.35 t=0.0ps kin=1.56 pot=21.98 Rg=21.686 SPS=15285
bl=0 pos[1]=[23.1 4.3 12.9] dr=1.34 t=0.0ps kin=1.53 pot=22.04 Rg=21.364 SPS=17384
bl=0 pos[1]=[21.8 4.5 12.6] dr=1.33 t=0.0ps kin=1.55 pot=22.04 Rg=21.049 SPS=16052
bl=0 pos[1]=[20.6 4.8 12.1] dr=1.33 t=0.0ps kin=1.56 pot=21.99 Rg=20.733 SPS=17021
bl=0 pos[1]=[20.1 3.9 10.2] dr=1.31 t=0.0ps kin=1.54 pot=21.91 Rg=20.416 SPS=15706
bl=0 pos[1]=[19.4 1.9 9.6] dr=1.31 t=0.0ps kin=1.47 pot=21.93 Rg=20.113 SPS=17062
bl=0 pos[1]=[19.3 3.0 9.5] dr=1.27 t=0.0ps kin=1.47 pot=21.89 Rg=19.833 SPS=16363
bl=0 pos[1]=[17.4 4.3 9.7] dr=1.28 t=0.0ps kin=1.51 pot=21.91 Rg=19.548 SPS=14951
bl=0 pos[1]=[15.7 4.1 8.9] dr=1.28 t=0.0ps kin=1.53 pot=21.84 Rg=19.320 SPS=16253
bl=0 pos[1]=[15.5 3.3 7.9] dr=1.29 t=0.0ps kin=1.48 pot=21.84 Rg=19.095 SPS=16024
bl=0 pos[1]=[14.2 2.0 8.6] dr=1.25 t=0.0ps kin=1.50 pot=21.78 Rg=18.884 SPS=15265
bl=0 pos[1]=[13.8 1.9 8.8] dr=1.24 t=0.0ps kin=1.45 pot=21.84 Rg=18.707 SPS=16479
bl=0 pos[1]=[14.4 1.5 7.5] dr=1.29 t=0.0ps kin=1.48 pot=21.83 Rg=18.521 SPS=15760
bl=0 pos[1]=[14.1 -0.5 6.7] dr=1.32 t=0.0ps kin=1.50 pot=21.77 Rg=18.294 SPS=15834
bl=0 pos[1]=[14.2 -3.6 7.3] dr=1.34 t=0.0ps kin=1.53 pot=21.77 Rg=18.027 SPS=15507

```

(continues on next page)



(continued from previous page)

```

bl=0 pos[1]=[13.5 -5.0 7.9] dr=1.31 t=0.0ps kin=1.51 pot=21.76 Rg=17.776 SPS=14977
bl=0 pos[1]=[11.7 -6.0 9.5] dr=1.30 t=0.0ps kin=1.52 pot=21.74 Rg=17.512 SPS=15046
bl=0 pos[1]=[12.5 -6.3 9.7] dr=1.28 t=0.0ps kin=1.46 pot=21.76 Rg=17.308 SPS=15968
bl=0 pos[1]=[10.5 -5.7 8.9] dr=1.28 t=0.0ps kin=1.50 pot=21.71 Rg=17.141 SPS=15638
bl=0 pos[1]=[11.2 -5.6 10.2] dr=1.26 t=0.0ps kin=1.49 pot=21.73 Rg=16.963 SPS=14641
bl=0 pos[1]=[9.6 -5.5 10.3] dr=1.26 t=0.0ps kin=1.50 pot=21.72 Rg=16.799 SPS=15726
bl=0 pos[1]=[7.6 -3.7 10.2] dr=1.29 t=0.0ps kin=1.50 pot=21.73 Rg=16.645 SPS=15978
bl=0 pos[1]=[8.0 -3.3 10.3] dr=1.25 t=0.0ps kin=1.48 pot=21.75 Rg=16.468 SPS=14907
bl=0 pos[1]=[8.0 -3.3 9.3] dr=1.26 t=0.0ps kin=1.48 pot=21.76 Rg=16.306 SPS=15538
bl=0 pos[1]=[8.2 -2.9 9.0] dr=1.27 t=0.0ps kin=1.49 pot=21.70 Rg=16.155 SPS=15881
bl=0 pos[1]=[8.3 -2.9 8.4] dr=1.26 t=0.0ps kin=1.45 pot=21.74 Rg=16.036 SPS=15195
bl=0 pos[1]=[9.6 -2.1 8.9] dr=1.28 t=0.0ps kin=1.54 pot=21.76 Rg=15.930 SPS=15667
bl=0 pos[1]=[9.7 -4.0 8.6] dr=1.30 t=0.0ps kin=1.54 pot=21.75 Rg=15.824 SPS=15813
bl=0 pos[1]=[11.9 -1.6 10.9] dr=1.28 t=0.0ps kin=1.53 pot=21.77 Rg=15.702 SPS=15025
bl=0 pos[1]=[10.9 -0.3 9.8] dr=1.26 t=0.0ps kin=1.51 pot=21.74 Rg=15.585 SPS=15269
bl=0 pos[1]=[10.6 -1.0 9.6] dr=1.30 t=0.0ps kin=1.50 pot=21.71 Rg=15.463 SPS=14977
bl=0 pos[1]=[10.5 -1.5 9.9] dr=1.29 t=0.0ps kin=1.54 pot=21.66 Rg=15.330 SPS=15329
bl=0 pos[1]=[9.4 0.1 11.2] dr=1.28 t=0.0ps kin=1.51 pot=21.69 Rg=15.175 SPS=15634
bl=0 pos[1]=[9.3 0.6 13.1] dr=1.25 t=0.0ps kin=1.50 pot=21.70 Rg=15.047 SPS=14532
bl=0 pos[1]=[10.9 0.2 12.8] dr=1.25 t=0.0ps kin=1.52 pot=21.68 Rg=14.976 SPS=15172
bl=0 pos[1]=[9.0 -2.2 12.4] dr=1.26 t=0.0ps kin=1.55 pot=21.71 Rg=14.872 SPS=15249
bl=0 pos[1]=[8.2 -1.7 12.3] dr=1.26 t=0.0ps kin=1.51 pot=21.69 Rg=14.751 SPS=14576
bl=0 pos[1]=[8.6 -2.2 12.1] dr=1.27 t=0.0ps kin=1.50 pot=21.70 Rg=14.626 SPS=15352
bl=0 pos[1]=[9.8 -2.1 13.5] dr=1.28 t=0.0ps kin=1.50 pot=21.67 Rg=14.539 SPS=14347
bl=0 pos[1]=[11.4 -1.8 12.7] dr=1.29 t=0.0ps kin=1.52 pot=21.67 Rg=14.481 SPS=15101
bl=0 pos[1]=[11.1 -1.3 12.1] dr=1.27 t=0.0ps kin=1.55 pot=21.69 Rg=14.404 SPS=15225
bl=0 pos[1]=[10.1 -0.9 11.6] dr=1.28 t=0.0ps kin=1.56 pot=21.66 Rg=14.341 SPS=14536
bl=0 pos[1]=[7.9 -1.6 10.1] dr=1.25 t=0.0ps kin=1.49 pot=21.72 Rg=14.295 SPS=15357
bl=0 pos[1]=[8.2 -3.5 8.5] dr=1.25 t=0.0ps kin=1.52 pot=21.67 Rg=14.239 SPS=14538
bl=0 pos[1]=[8.3 -2.7 5.7] dr=1.25 t=0.0ps kin=1.49 pot=21.68 Rg=14.166 SPS=14819
bl=0 pos[1]=[9.5 -2.0 4.7] dr=1.25 t=0.0ps kin=1.44 pot=21.62 Rg=14.074 SPS=14742
bl=0 pos[1]=[8.7 -1.1 7.0] dr=1.23 t=0.0ps kin=1.50 pot=21.60 Rg=13.981 SPS=14529
bl=0 pos[1]=[9.7 -2.3 6.4] dr=1.25 t=0.0ps kin=1.47 pot=21.61 Rg=13.903 SPS=15026
bl=0 pos[1]=[10.0 -2.8 6.1] dr=1.27 t=0.0ps kin=1.46 pot=21.66 Rg=13.817 SPS=14228
bl=0 pos[1]=[10.7 -4.1 5.2] dr=1.28 t=0.0ps kin=1.46 pot=21.60 Rg=13.703 SPS=14853
bl=0 pos[1]=[10.4 -4.1 5.6] dr=1.23 t=0.0ps kin=1.45 pot=21.65 Rg=13.621 SPS=14234
bl=0 pos[1]=[9.4 -4.5 6.4] dr=1.25 t=0.0ps kin=1.48 pot=21.63 Rg=13.559 SPS=14643
bl=0 pos[1]=[9.4 -3.9 6.1] dr=1.26 t=0.0ps kin=1.49 pot=21.58 Rg=13.454 SPS=14587
bl=0 pos[1]=[9.4 -3.2 5.7] dr=1.27 t=0.0ps kin=1.49 pot=21.68 Rg=13.361 SPS=14174
bl=0 pos[1]=[8.6 -1.6 5.6] dr=1.26 t=0.0ps kin=1.48 pot=21.64 Rg=13.247 SPS=14783
bl=0 pos[1]=[8.9 -2.4 5.0] dr=1.23 t=0.0ps kin=1.48 pot=21.65 Rg=13.137 SPS=13895
bl=0 pos[1]=[8.6 -2.9 5.4] dr=1.29 t=0.0ps kin=1.50 pot=21.63 Rg=13.033 SPS=15268
bl=0 pos[1]=[8.1 -2.7 5.5] dr=1.23 t=0.0ps kin=1.47 pot=21.59 Rg=13.018 SPS=14657
bl=0 pos[1]=[9.0 -3.4 5.6] dr=1.24 t=0.0ps kin=1.50 pot=21.57 Rg=13.020 SPS=15318
bl=0 pos[1]=[10.3 -3.4 5.2] dr=1.26 t=0.0ps kin=1.47 pot=21.62 Rg=12.993 SPS=15619
bl=0 pos[1]=[12.2 -3.1 4.5] dr=1.24 t=0.0ps kin=1.44 pot=21.62 Rg=12.958 SPS=15009
bl=0 pos[1]=[13.1 -4.2 3.2] dr=1.27 t=0.0ps kin=1.47 pot=21.60 Rg=12.973 SPS=16050
bl=0 pos[1]=[15.7 -4.1 2.5] dr=1.22 t=0.0ps kin=1.53 pot=21.61 Rg=12.973 SPS=15933
bl=0 pos[1]=[16.6 -2.7 2.8] dr=1.27 t=0.0ps kin=1.60 pot=21.63 Rg=12.967 SPS=14931
bl=0 pos[1]=[16.6 -2.4 3.0] dr=1.31 t=0.0ps kin=1.58 pot=21.60 Rg=12.976 SPS=15430
bl=0 pos[1]=[16.8 -3.2 3.1] dr=1.27 t=0.0ps kin=1.52 pot=21.59 Rg=12.996 SPS=15985
bl=0 pos[1]=[15.8 -3.3 2.8] dr=1.25 t=0.0ps kin=1.51 pot=21.67 Rg=13.008 SPS=14636

```

(continues on next page)

(continued from previous page)

```

bl=0 pos[1]=[14.6 -2.7 3.2] dr=1.26 t=0.0ps kin=1.53 pot=21.64 Rg=13.001 SPS=14895
bl=0 pos[1]=[15.8 -3.8 3.4] dr=1.24 t=0.0ps kin=1.53 pot=21.60 Rg=13.052 SPS=14783
bl=0 pos[1]=[13.8 -4.0 3.2] dr=1.30 t=0.0ps kin=1.53 pot=21.61 Rg=13.043 SPS=15287
bl=0 pos[1]=[13.0 -4.9 2.2] dr=1.28 t=0.0ps kin=1.54 pot=21.62 Rg=12.968 SPS=16326
bl=0 pos[1]=[14.2 -5.7 2.7] dr=1.31 t=0.0ps kin=1.49 pot=21.63 Rg=12.896 SPS=14765
bl=0 pos[1]=[14.1 -6.7 2.3] dr=1.32 t=0.0ps kin=1.49 pot=21.58 Rg=12.902 SPS=14747
bl=0 pos[1]=[13.1 -7.2 3.1] dr=1.28 t=0.0ps kin=1.47 pot=21.60 Rg=12.843 SPS=15693
bl=0 pos[1]=[13.5 -7.8 3.4] dr=1.25 t=0.0ps kin=1.44 pot=21.59 Rg=12.788 SPS=14657
bl=0 pos[1]=[13.6 -7.7 4.0] dr=1.28 t=0.0ps kin=1.55 pot=21.58 Rg=12.710 SPS=15304
bl=0 pos[1]=[12.0 -6.8 4.8] dr=1.31 t=0.0ps kin=1.53 pot=21.59 Rg=12.667 SPS=14126
bl=0 pos[1]=[11.2 -4.6 3.3] dr=1.30 t=0.0ps kin=1.53 pot=21.62 Rg=12.625 SPS=14725
bl=0 pos[1]=[10.0 -2.3 0.8] dr=1.28 t=0.0ps kin=1.52 pot=21.61 Rg=12.537 SPS=14998
bl=0 pos[1]=[10.1 -1.0 0.0] dr=1.24 t=0.0ps kin=1.49 pot=21.61 Rg=12.500 SPS=14985
bl=0 pos[1]=[9.8 -0.2 -0.1] dr=1.23 t=0.0ps kin=1.51 pot=21.62 Rg=12.467 SPS=15193
bl=0 pos[1]=[8.9 -0.9 0.8] dr=1.28 t=0.0ps kin=1.52 pot=21.60 Rg=12.458 SPS=15583
bl=0 pos[1]=[8.7 -1.6 1.2] dr=1.26 t=0.0ps kin=1.49 pot=21.64 Rg=12.488 SPS=15535
bl=0 pos[1]=[8.2 -2.2 1.7] dr=1.27 t=0.0ps kin=1.49 pot=21.63 Rg=12.516 SPS=15794
bl=0 pos[1]=[8.8 -3.2 1.5] dr=1.27 t=0.0ps kin=1.49 pot=21.61 Rg=12.568 SPS=14251
bl=0 pos[1]=[9.5 -4.4 1.4] dr=1.25 t=0.0ps kin=1.49 pot=21.62 Rg=12.583 SPS=15538
bl=0 pos[1]=[9.6 -3.4 2.9] dr=1.29 t=0.0ps kin=1.52 pot=21.58 Rg=12.574 SPS=15765
bl=0 pos[1]=[9.8 -2.2 3.3] dr=1.26 t=0.0ps kin=1.55 pot=21.60 Rg=12.525 SPS=14635
bl=0 pos[1]=[11.3 -1.9 3.2] dr=1.27 t=0.0ps kin=1.50 pot=21.65 Rg=12.525 SPS=15586
bl=0 pos[1]=[12.3 -1.0 3.6] dr=1.27 t=0.0ps kin=1.53 pot=21.63 Rg=12.543 SPS=15903
bl=0 pos[1]=[13.3 0.2 4.5] dr=1.26 t=0.0ps kin=1.51 pot=21.64 Rg=12.557 SPS=14694
bl=0 pos[1]=[15.1 -0.3 3.5] dr=1.23 t=0.0ps kin=1.50 pot=21.64 Rg=12.550 SPS=15851
bl=0 pos[1]=[15.3 -1.0 4.7] dr=1.24 t=0.0ps kin=1.49 pot=21.59 Rg=12.550 SPS=15151
bl=0 pos[1]=[15.5 -0.2 4.7] dr=1.27 t=0.0ps kin=1.50 pot=21.60 Rg=12.526 SPS=14689
bl=0 pos[1]=[17.1 -0.6 3.6] dr=1.26 t=0.0ps kin=1.50 pot=21.65 Rg=12.487 SPS=15510
bl=0 pos[1]=[17.7 -0.3 2.6] dr=1.26 t=0.0ps kin=1.53 pot=21.58 Rg=12.496 SPS=14952
bl=0 pos[1]=[19.3 -0.2 3.1] dr=1.26 t=0.0ps kin=1.55 pot=21.62 Rg=12.544 SPS=15678
bl=0 pos[1]=[20.2 -1.5 4.5] dr=1.30 t=0.0ps kin=1.53 pot=21.60 Rg=12.615 SPS=15646
bl=0 pos[1]=[19.3 -3.0 4.8] dr=1.25 t=0.0ps kin=1.48 pot=21.63 Rg=12.657 SPS=14483
bl=0 pos[1]=[18.8 -3.5 6.4] dr=1.23 t=0.0ps kin=1.46 pot=21.61 Rg=12.653 SPS=15484
bl=0 pos[1]=[17.9 -4.0 7.2] dr=1.25 t=0.0ps kin=1.47 pot=21.62 Rg=12.643 SPS=15727
bl=0 pos[1]=[16.7 -5.1 7.0] dr=1.26 t=0.0ps kin=1.51 pot=21.60 Rg=12.612 SPS=15453
bl=0 pos[1]=[16.7 -4.6 7.1] dr=1.24 t=0.0ps kin=1.43 pot=21.61 Rg=12.543 SPS=15341
bl=0 pos[1]=[16.9 -4.8 8.0] dr=1.23 t=0.0ps kin=1.48 pot=21.58 Rg=12.479 SPS=15349
bl=0 pos[1]=[16.0 -4.9 9.2] dr=1.25 t=0.0ps kin=1.48 pot=21.59 Rg=12.412 SPS=15213
bl=0 pos[1]=[14.3 -5.5 10.1] dr=1.25 t=0.0ps kin=1.49 pot=21.59 Rg=12.357 SPS=15405
bl=0 pos[1]=[13.5 -5.7 9.7] dr=1.25 t=0.0ps kin=1.50 pot=21.58 Rg=12.295 SPS=15740
bl=0 pos[1]=[13.7 -5.6 10.6] dr=1.25 t=0.0ps kin=1.47 pot=21.57 Rg=12.219 SPS=15298
bl=0 pos[1]=[13.1 -4.0 10.2] dr=1.25 t=0.0ps kin=1.46 pot=21.62 Rg=12.166 SPS=15613
bl=0 pos[1]=[11.7 -2.8 12.3] dr=1.28 t=0.0ps kin=1.50 pot=21.60 Rg=12.104 SPS=15885
bl=0 pos[1]=[11.6 -2.6 12.2] dr=1.27 t=0.0ps kin=1.49 pot=21.60 Rg=12.051 SPS=14927
bl=0 pos[1]=[11.7 -3.1 12.0] dr=1.27 t=0.0ps kin=1.51 pot=21.62 Rg=12.054 SPS=15853
bl=0 pos[1]=[11.7 -3.4 9.4] dr=1.27 t=0.0ps kin=1.48 pot=21.64 Rg=12.082 SPS=14691
bl=0 pos[1]=[12.6 -3.5 10.1] dr=1.27 t=0.0ps kin=1.46 pot=21.58 Rg=12.105 SPS=15485
bl=0 pos[1]=[14.8 -3.0 12.1] dr=1.29 t=0.0ps kin=1.46 pot=21.62 Rg=12.081 SPS=15465
bl=0 pos[1]=[15.5 -3.2 13.2] dr=1.26 t=0.0ps kin=1.48 pot=21.57 Rg=12.013 SPS=14774
bl=0 pos[1]=[16.1 -3.7 12.6] dr=1.27 t=0.0ps kin=1.50 pot=21.60 Rg=11.947 SPS=15412
bl=0 pos[1]=[16.1 -2.8 11.8] dr=1.28 t=0.0ps kin=1.54 pot=21.56 Rg=11.882 SPS=15319
bl=0 pos[1]=[17.6 -2.6 12.5] dr=1.27 t=0.0ps kin=1.46 pot=21.59 Rg=11.782 SPS=14661

```

(continues on next page)

(continued from previous page)

```

bl=0 pos[1]=[17.3 -1.3 11.4] dr=1.26 t=0.0ps kin=1.56 pot=21.57 Rg=11.716 SPS=15855
bl=0 pos[1]=[17.0 -2.1 10.9] dr=1.29 t=0.0ps kin=1.56 pot=21.58 Rg=11.703 SPS=15822
bl=0 pos[1]=[16.1 -3.4 10.3] dr=1.27 t=0.0ps kin=1.52 pot=21.63 Rg=11.698 SPS=15103
bl=0 pos[1]=[14.1 -3.1 9.9] dr=1.25 t=0.0ps kin=1.47 pot=21.63 Rg=11.687 SPS=16030
bl=0 pos[1]=[12.9 -1.7 11.0] dr=1.23 t=0.0ps kin=1.48 pot=21.59 Rg=11.679 SPS=15538
bl=0 pos[1]=[12.9 -1.6 11.4] dr=1.23 t=0.0ps kin=1.48 pot=21.60 Rg=11.692 SPS=14640
bl=0 pos[1]=[13.0 -1.2 11.9] dr=1.27 t=0.0ps kin=1.51 pot=21.60 Rg=11.712 SPS=15223
bl=0 pos[1]=[13.9 0.8 13.1] dr=1.28 t=0.0ps kin=1.50 pot=21.54 Rg=11.732 SPS=15002
bl=0 pos[1]=[13.6 3.4 12.0] dr=1.25 t=0.0ps kin=1.52 pot=21.61 Rg=11.722 SPS=15281
bl=0 pos[1]=[15.0 5.1 10.7] dr=1.28 t=0.0ps kin=1.48 pot=21.62 Rg=11.697 SPS=15488
bl=0 pos[1]=[15.7 5.7 10.3] dr=1.25 t=0.0ps kin=1.55 pot=21.57 Rg=11.701 SPS=15135
bl=0 pos[1]=[15.6 5.2 8.9] dr=1.30 t=0.0ps kin=1.52 pot=21.60 Rg=11.727 SPS=15764
bl=0 pos[1]=[15.0 6.4 7.1] dr=1.28 t=0.0ps kin=1.51 pot=21.59 Rg=11.743 SPS=15725
bl=0 pos[1]=[13.3 5.4 7.0] dr=1.25 t=0.0ps kin=1.49 pot=21.62 Rg=11.696 SPS=14909
bl=0 pos[1]=[12.4 4.5 6.4] dr=1.30 t=0.0ps kin=1.49 pot=21.64 Rg=11.649 SPS=15487
bl=0 pos[1]=[13.3 4.1 6.5] dr=1.26 t=0.0ps kin=1.55 pot=21.62 Rg=11.658 SPS=15225
bl=0 pos[1]=[13.7 1.9 6.5] dr=1.27 t=0.0ps kin=1.56 pot=21.57 Rg=11.669 SPS=14727
bl=0 pos[1]=[13.4 3.0 8.3] dr=1.29 t=0.0ps kin=1.52 pot=21.63 Rg=11.682 SPS=15424
bl=0 pos[1]=[12.9 3.4 7.3] dr=1.27 t=0.0ps kin=1.46 pot=21.60 Rg=11.671 SPS=15406
bl=0 pos[1]=[13.3 2.7 6.8] dr=1.23 t=0.0ps kin=1.51 pot=21.60 Rg=11.657 SPS=15134
bl=0 pos[1]=[13.1 2.2 6.2] dr=1.26 t=0.0ps kin=1.44 pot=21.57 Rg=11.658 SPS=15449
bl=0 pos[1]=[13.6 3.6 6.1] dr=1.26 t=0.0ps kin=1.49 pot=21.59 Rg=11.668 SPS=14817
bl=0 pos[1]=[13.2 4.2 5.8] dr=1.28 t=0.0ps kin=1.55 pot=21.60 Rg=11.695 SPS=15689
bl=0 pos[1]=[11.9 4.5 5.6] dr=1.30 t=0.0ps kin=1.61 pot=21.63 Rg=11.666 SPS=15229
bl=0 pos[1]=[10.6 5.4 5.4] dr=1.31 t=0.0ps kin=1.51 pot=21.60 Rg=11.658 SPS=15060
bl=0 pos[1]=[10.0 3.6 4.8] dr=1.26 t=0.0ps kin=1.52 pot=21.56 Rg=11.637 SPS=15574
bl=0 pos[1]=[9.1 2.6 3.9] dr=1.25 t=0.0ps kin=1.45 pot=21.61 Rg=11.582 SPS=15346
bl=0 pos[1]=[9.8 4.2 3.4] dr=1.25 t=0.0ps kin=1.50 pot=21.58 Rg=11.529 SPS=14922
bl=0 pos[1]=[10.8 4.3 2.7] dr=1.26 t=0.0ps kin=1.53 pot=21.57 Rg=11.521 SPS=15497
bl=0 pos[1]=[12.3 3.7 1.4] dr=1.26 t=0.0ps kin=1.48 pot=21.59 Rg=11.569 SPS=14988
bl=0 pos[1]=[12.3 4.0 0.7] dr=1.27 t=0.0ps kin=1.46 pot=21.59 Rg=11.590 SPS=15306
bl=0 pos[1]=[12.3 5.6 1.2] dr=1.26 t=0.0ps kin=1.48 pot=21.62 Rg=11.660 SPS=14902
bl=0 pos[1]=[11.6 6.7 0.9] dr=1.26 t=0.0ps kin=1.53 pot=21.63 Rg=11.731 SPS=13760
bl=0 pos[1]=[12.2 6.7 0.1] dr=1.26 t=0.0ps kin=1.48 pot=21.62 Rg=11.757 SPS=14693
bl=0 pos[1]=[10.9 7.7 0.8] dr=1.27 t=0.0ps kin=1.51 pot=21.56 Rg=11.752 SPS=14069
bl=0 pos[1]=[8.9 8.3 0.6] dr=1.26 t=0.0ps kin=1.45 pot=21.62 Rg=11.764 SPS=15348
bl=0 pos[1]=[8.1 6.2 1.9] dr=1.25 t=0.0ps kin=1.54 pot=21.53 Rg=11.751 SPS=13892
bl=0 pos[1]=[8.6 6.3 2.6] dr=1.32 t=0.0ps kin=1.55 pot=21.59 Rg=11.687 SPS=15426
bl=0 pos[1]=[6.3 5.8 3.0] dr=1.30 t=0.0ps kin=1.55 pot=21.60 Rg=11.611 SPS=14976
bl=0 pos[1]=[5.8 5.6 4.3] dr=1.29 t=0.0ps kin=1.51 pot=21.61 Rg=11.547 SPS=14934
bl=0 pos[1]=[4.3 7.6 4.5] dr=1.23 t=0.0ps kin=1.47 pot=21.62 Rg=11.528 SPS=15963
bl=0 pos[1]=[5.0 8.9 3.1] dr=1.25 t=0.0ps kin=1.52 pot=21.56 Rg=11.524 SPS=15907
bl=0 pos[1]=[5.5 10.2 2.4] dr=1.29 t=0.0ps kin=1.50 pot=21.59 Rg=11.519 SPS=15570
bl=0 pos[1]=[8.3 9.0 3.5] dr=1.27 t=0.0ps kin=1.50 pot=21.58 Rg=11.496 SPS=15071
bl=0 pos[1]=[8.6 7.2 3.7] dr=1.24 t=0.0ps kin=1.45 pot=21.60 Rg=11.524 SPS=15630
bl=0 pos[1]=[8.4 7.2 4.8] dr=1.24 t=0.0ps kin=1.53 pot=21.58 Rg=11.551 SPS=14633
bl=0 pos[1]=[7.9 8.7 5.0] dr=1.29 t=0.0ps kin=1.51 pot=21.61 Rg=11.574 SPS=15573
bl=0 pos[1]=[8.0 9.1 5.0] dr=1.23 t=0.0ps kin=1.49 pot=21.62 Rg=11.603 SPS=14067
bl=0 pos[1]=[8.4 9.1 4.9] dr=1.25 t=0.0ps kin=1.55 pot=21.61 Rg=11.556 SPS=14884
bl=0 pos[1]=[7.2 8.9 4.0] dr=1.27 t=0.0ps kin=1.55 pot=21.58 Rg=11.530 SPS=15804
bl=0 pos[1]=[7.9 9.4 4.8] dr=1.28 t=0.0ps kin=1.50 pot=21.63 Rg=11.508 SPS=14850
bl=0 pos[1]=[7.4 9.3 6.3] dr=1.28 t=0.0ps kin=1.54 pot=21.62 Rg=11.462 SPS=15185

```

(continues on next page)

(continued from previous page)

```

bl=0 pos[1]=[7.7 9.9 6.9] dr=1.26 t=0.0ps kin=1.50 pot=21.55 Rg=11.455 SPS=15104
bl=0 pos[1]=[7.1 10.0 7.0] dr=1.23 t=0.0ps kin=1.48 pot=21.53 Rg=11.477 SPS=15178
bl=0 pos[1]=[8.0 10.6 7.4] dr=1.21 t=0.0ps kin=1.45 pot=21.58 Rg=11.493 SPS=14900
bl=0 pos[1]=[10.0 8.8 5.8] dr=1.24 t=0.0ps kin=1.49 pot=21.52 Rg=11.521 SPS=14086
bl=0 pos[1]=[8.8 7.7 3.3] dr=1.22 t=0.0ps kin=1.51 pot=21.60 Rg=11.447 SPS=16072
bl=0 pos[1]=[7.7 6.9 3.9] dr=1.24 t=0.0ps kin=1.47 pot=21.57 Rg=11.406 SPS=15798
bl=0 pos[1]=[8.2 8.7 4.4] dr=1.23 t=0.0ps kin=1.44 pot=21.61 Rg=11.421 SPS=14803
bl=0 pos[1]=[10.3 9.9 4.0] dr=1.24 t=0.0ps kin=1.48 pot=21.61 Rg=11.436 SPS=16213
bl=0 pos[1]=[11.0 9.8 3.7] dr=1.24 t=0.0ps kin=1.50 pot=21.56 Rg=11.466 SPS=16050
bl=0 pos[1]=[11.4 9.3 4.0] dr=1.21 t=0.0ps kin=1.46 pot=21.59 Rg=11.544 SPS=15034
bl=0 pos[1]=[13.0 9.9 4.4] dr=1.25 t=0.0ps kin=1.45 pot=21.57 Rg=11.574 SPS=15253
bl=0 pos[1]=[12.6 8.4 5.2] dr=1.26 t=0.0ps kin=1.47 pot=21.61 Rg=11.581 SPS=14861
bl=0 pos[1]=[12.1 7.4 4.3] dr=1.27 t=0.0ps kin=1.50 pot=21.58 Rg=11.566 SPS=15985
bl=0 pos[1]=[11.2 5.6 4.6] dr=1.27 t=0.0ps kin=1.49 pot=21.57 Rg=11.522 SPS=15361
bl=0 pos[1]=[11.7 6.8 3.1] dr=1.26 t=0.0ps kin=1.40 pot=21.56 Rg=11.464 SPS=14883
bl=0 pos[1]=[11.5 6.4 3.6] dr=1.29 t=0.0ps kin=1.47 pot=21.54 Rg=11.401 SPS=15308
bl=0 pos[1]=[10.4 6.8 3.0] dr=1.27 t=0.0ps kin=1.51 pot=21.55 Rg=11.339 SPS=15368
bl=0 pos[1]=[8.7 5.6 2.4] dr=1.27 t=0.0ps kin=1.47 pot=21.56 Rg=11.304 SPS=15015
bl=0 pos[1]=[7.0 6.6 2.8] dr=1.25 t=0.0ps kin=1.47 pot=21.57 Rg=11.287 SPS=15242
bl=0 pos[1]=[4.6 5.8 1.4] dr=1.25 t=0.0ps kin=1.42 pot=21.59 Rg=11.275 SPS=12877
bl=0 pos[1]=[3.6 3.5 2.2] dr=1.24 t=0.0ps kin=1.46 pot=21.55 Rg=11.277 SPS=15548
bl=0 pos[1]=[4.1 2.7 2.5] dr=1.21 t=0.0ps kin=1.45 pot=21.56 Rg=11.260 SPS=15089
bl=0 pos[1]=[4.1 2.5 1.7] dr=1.22 t=0.0ps kin=1.48 pot=21.55 Rg=11.280 SPS=15123
bl=0 pos[1]=[3.6 2.8 3.6] dr=1.25 t=0.0ps kin=1.45 pot=21.56 Rg=11.303 SPS=15307
bl=0 pos[1]=[2.1 3.5 2.8] dr=1.25 t=0.0ps kin=1.45 pot=21.59 Rg=11.310 SPS=15295
bl=0 pos[1]=[1.6 3.1 1.9] dr=1.23 t=0.0ps kin=1.48 pot=21.60 Rg=11.319 SPS=15658
bl=0 pos[1]=[0.1 3.9 2.8] dr=1.25 t=0.0ps kin=1.54 pot=21.59 Rg=11.377 SPS=15241
bl=0 pos[1]=[-0.4 4.4 4.5] dr=1.28 t=0.0ps kin=1.57 pot=21.62 Rg=11.431 SPS=14641
bl=0 pos[1]=[-1.5 4.0 4.2] dr=1.26 t=0.0ps kin=1.49 pot=21.65 Rg=11.480 SPS=15910
bl=0 pos[1]=[-2.0 3.2 4.6] dr=1.27 t=0.0ps kin=1.49 pot=21.59 Rg=11.509 SPS=15875
bl=0 pos[1]=[-0.8 3.3 4.5] dr=1.25 t=0.0ps kin=1.55 pot=21.59 Rg=11.535 SPS=15391
bl=0 pos[1]=[0.9 3.1 5.4] dr=1.27 t=0.0ps kin=1.51 pot=21.61 Rg=11.544 SPS=15960
bl=0 pos[1]=[2.1 3.7 4.8] dr=1.27 t=0.0ps kin=1.50 pot=21.61 Rg=11.526 SPS=14958
bl=0 pos[1]=[1.6 1.2 5.1] dr=1.28 t=0.0ps kin=1.49 pot=21.57 Rg=11.512 SPS=14556
bl=0 pos[1]=[0.0 0.8 5.3] dr=1.26 t=0.0ps kin=1.50 pot=21.58 Rg=11.501 SPS=15425
bl=0 pos[1]=[-0.4 0.5 5.2] dr=1.25 t=0.0ps kin=1.51 pot=21.62 Rg=11.532 SPS=14282
bl=0 pos[1]=[-0.3 -0.2 6.0] dr=1.25 t=0.0ps kin=1.54 pot=21.59 Rg=11.567 SPS=15298
bl=0 pos[1]=[-0.6 -0.1 5.0] dr=1.28 t=0.0ps kin=1.53 pot=21.62 Rg=11.582 SPS=15590
bl=0 pos[1]=[-0.4 -0.2 4.8] dr=1.29 t=0.0ps kin=1.50 pot=21.65 Rg=11.576 SPS=15143
bl=0 pos[1]=[-0.2 -0.9 5.5] dr=1.29 t=0.0ps kin=1.54 pot=21.60 Rg=11.549 SPS=15769
bl=0 pos[1]=[0.2 -0.7 4.5] dr=1.26 t=0.0ps kin=1.53 pot=21.59 Rg=11.504 SPS=15588
bl=0 pos[1]=[0.7 -0.4 4.0] dr=1.26 t=0.0ps kin=1.52 pot=21.57 Rg=11.499 SPS=14497
bl=0 pos[1]=[0.9 1.6 4.1] dr=1.26 t=0.0ps kin=1.49 pot=21.56 Rg=11.502 SPS=16029
bl=0 pos[1]=[1.9 3.2 5.4] dr=1.28 t=0.0ps kin=1.49 pot=21.59 Rg=11.498 SPS=15561
bl=0 pos[1]=[1.3 2.4 5.8] dr=1.25 t=0.0ps kin=1.48 pot=21.61 Rg=11.487 SPS=15137
bl=0 pos[1]=[3.4 2.2 5.7] dr=1.26 t=0.0ps kin=1.47 pot=21.57 Rg=11.499 SPS=15750
bl=0 pos[1]=[4.3 2.5 6.0] dr=1.26 t=0.0ps kin=1.57 pot=21.54 Rg=11.502 SPS=15969
bl=0 pos[1]=[3.2 3.6 5.7] dr=1.27 t=0.0ps kin=1.54 pot=21.58 Rg=11.456 SPS=14584
bl=0 pos[1]=[2.9 3.7 4.8] dr=1.31 t=0.0ps kin=1.52 pot=21.59 Rg=11.421 SPS=15340
bl=0 pos[1]=[3.4 3.8 6.2] dr=1.28 t=0.0ps kin=1.53 pot=21.58 Rg=11.392 SPS=15004
bl=0 pos[1]=[5.7 4.8 7.8] dr=1.26 t=0.0ps kin=1.48 pot=21.58 Rg=11.341 SPS=15638
bl=0 pos[1]=[6.3 7.6 7.8] dr=1.28 t=0.0ps kin=1.53 pot=21.57 Rg=11.351 SPS=15443

```

(continues on next page)

(continued from previous page)

```

bl=0 pos[1]=[5.3 9.9 6.7] dr=1.25 t=0.0ps kin=1.51 pot=21.59 Rg=11.356 SPS=14598
bl=0 pos[1]=[6.6 9.1 8.0] dr=1.28 t=0.0ps kin=1.54 pot=21.60 Rg=11.377 SPS=15460
bl=0 pos[1]=[7.5 7.8 6.3] dr=1.28 t=0.0ps kin=1.53 pot=21.61 Rg=11.405 SPS=15348
bl=0 pos[1]=[6.3 7.6 5.8] dr=1.26 t=0.0ps kin=1.52 pot=21.60 Rg=11.384 SPS=14498
bl=0 pos[1]=[5.3 8.1 6.3] dr=1.24 t=0.0ps kin=1.57 pot=21.59 Rg=11.346 SPS=15382
bl=0 pos[1]=[6.0 8.5 6.7] dr=1.28 t=0.0ps kin=1.53 pot=21.59 Rg=11.366 SPS=14421
bl=0 pos[1]=[8.4 5.7 7.3] dr=1.30 t=0.0ps kin=1.54 pot=21.59 Rg=11.415 SPS=15602
bl=0 pos[1]=[9.3 6.4 5.1] dr=1.30 t=0.0ps kin=1.55 pot=21.57 Rg=11.450 SPS=15288
bl=0 pos[1]=[9.5 7.0 4.3] dr=1.26 t=0.0ps kin=1.51 pot=21.63 Rg=11.464 SPS=13880
bl=0 pos[1]=[9.8 6.0 3.0] dr=1.28 t=0.0ps kin=1.60 pot=21.57 Rg=11.445 SPS=15141
bl=0 pos[1]=[9.3 6.8 1.5] dr=1.27 t=0.0ps kin=1.47 pot=21.59 Rg=11.437 SPS=14515
bl=0 pos[1]=[8.7 5.4 0.7] dr=1.25 t=0.0ps kin=1.58 pot=21.58 Rg=11.436 SPS=15241
bl=0 pos[1]=[8.0 6.8 1.1] dr=1.29 t=0.0ps kin=1.53 pot=21.58 Rg=11.440 SPS=15042
bl=0 pos[1]=[8.5 6.3 0.8] dr=1.26 t=0.0ps kin=1.50 pot=21.55 Rg=11.446 SPS=14376
bl=0 pos[1]=[9.1 6.4 -0.9] dr=1.25 t=0.0ps kin=1.41 pot=21.55 Rg=11.507 SPS=15507
bl=0 pos[1]=[6.6 4.6 -0.6] dr=1.24 t=0.0ps kin=1.44 pot=21.53 Rg=11.574 SPS=13916
bl=0 pos[1]=[5.4 4.6 -0.8] dr=1.25 t=0.0ps kin=1.44 pot=21.57 Rg=11.607 SPS=15513
bl=0 pos[1]=[4.9 4.7 -0.6] dr=1.27 t=0.0ps kin=1.49 pot=21.55 Rg=11.622 SPS=15355
bl=0 pos[1]=[5.0 4.8 -0.1] dr=1.27 t=0.0ps kin=1.52 pot=21.54 Rg=11.591 SPS=14799
bl=0 pos[1]=[4.9 4.7 1.1] dr=1.25 t=0.0ps kin=1.46 pot=21.59 Rg=11.495 SPS=15607
bl=0 pos[1]=[5.0 5.6 0.9] dr=1.26 t=0.0ps kin=1.46 pot=21.63 Rg=11.421 SPS=15649
bl=0 pos[1]=[5.7 5.7 -0.0] dr=1.23 t=0.0ps kin=1.49 pot=21.60 Rg=11.419 SPS=14767
bl=0 pos[1]=[6.7 4.5 1.6] dr=1.25 t=0.0ps kin=1.51 pot=21.65 Rg=11.421 SPS=15406
bl=0 pos[1]=[8.2 4.3 1.6] dr=1.26 t=0.0ps kin=1.51 pot=21.62 Rg=11.403 SPS=14683
bl=0 pos[1]=[7.8 4.7 0.6] dr=1.24 t=0.0ps kin=1.53 pot=21.60 Rg=11.366 SPS=14926
bl=0 pos[1]=[7.8 6.2 -0.5] dr=1.27 t=0.0ps kin=1.52 pot=21.64 Rg=11.371 SPS=15150
bl=0 pos[1]=[7.2 6.8 -0.8] dr=1.28 t=0.0ps kin=1.50 pot=21.59 Rg=11.383 SPS=14266
bl=0 pos[1]=[6.7 7.1 -0.8] dr=1.24 t=0.0ps kin=1.52 pot=21.54 Rg=11.416 SPS=15577
bl=0 pos[1]=[8.1 6.0 -0.5] dr=1.21 t=0.0ps kin=1.47 pot=21.62 Rg=11.391 SPS=14314
bl=0 pos[1]=[8.5 6.0 0.9] dr=1.28 t=0.0ps kin=1.49 pot=21.58 Rg=11.401 SPS=15167
bl=0 pos[1]=[8.8 6.2 -0.3] dr=1.24 t=0.0ps kin=1.44 pot=21.56 Rg=11.427 SPS=15749
bl=0 pos[1]=[11.1 7.2 2.1] dr=1.23 t=0.0ps kin=1.50 pot=21.54 Rg=11.467 SPS=15318
bl=0 pos[1]=[10.1 7.1 2.2] dr=1.25 t=0.0ps kin=1.47 pot=21.61 Rg=11.549 SPS=15424
bl=0 pos[1]=[11.7 7.4 2.8] dr=1.30 t=0.0ps kin=1.52 pot=21.60 Rg=11.650 SPS=15118
bl=0 pos[1]=[13.9 9.1 2.5] dr=1.28 t=0.0ps kin=1.56 pot=21.60 Rg=11.735 SPS=14509
bl=0 pos[1]=[12.4 8.2 0.8] dr=1.26 t=0.0ps kin=1.52 pot=21.60 Rg=11.769 SPS=15139
bl=0 pos[1]=[11.0 7.8 1.0] dr=1.27 t=0.0ps kin=1.54 pot=21.62 Rg=11.790 SPS=14834
bl=0 pos[1]=[11.1 8.1 1.2] dr=1.26 t=0.0ps kin=1.48 pot=21.58 Rg=11.816 SPS=15916
bl=0 pos[1]=[11.2 8.4 1.8] dr=1.25 t=0.0ps kin=1.52 pot=21.59 Rg=11.844 SPS=15685
bl=0 pos[1]=[10.8 9.6 2.5] dr=1.28 t=0.0ps kin=1.54 pot=21.59 Rg=11.891 SPS=14799
bl=0 pos[1]=[10.5 11.2 1.9] dr=1.25 t=0.0ps kin=1.48 pot=21.61 Rg=11.924 SPS=16088
bl=0 pos[1]=[9.7 9.3 1.3] dr=1.24 t=0.0ps kin=1.46 pot=21.59 Rg=11.955 SPS=15311
bl=0 pos[1]=[9.9 10.0 0.0] dr=1.26 t=0.0ps kin=1.51 pot=21.57 Rg=11.974 SPS=15034
bl=0 pos[1]=[10.9 9.9 -1.3] dr=1.28 t=0.0ps kin=1.48 pot=21.58 Rg=11.985 SPS=15112
bl=0 pos[1]=[10.8 8.9 -0.7] dr=1.27 t=0.0ps kin=1.46 pot=21.55 Rg=11.957 SPS=15244
bl=0 pos[1]=[10.5 7.3 1.5] dr=1.26 t=0.0ps kin=1.48 pot=21.57 Rg=11.921 SPS=15311
bl=0 pos[1]=[10.8 7.2 3.2] dr=1.25 t=0.0ps kin=1.50 pot=21.55 Rg=11.897 SPS=15243
bl=0 pos[1]=[9.7 7.0 3.7] dr=1.27 t=0.0ps kin=1.46 pot=21.60 Rg=11.872 SPS=14756
bl=0 pos[1]=[8.2 5.2 3.1] dr=1.25 t=0.0ps kin=1.51 pot=21.56 Rg=11.863 SPS=16067
bl=0 pos[1]=[8.6 5.9 2.5] dr=1.23 t=0.0ps kin=1.52 pot=21.58 Rg=11.883 SPS=15459
bl=0 pos[1]=[8.3 7.8 1.3] dr=1.24 t=0.0ps kin=1.47 pot=21.56 Rg=11.897 SPS=15477
bl=0 pos[1]=[7.6 10.2 1.1] dr=1.26 t=0.0ps kin=1.50 pot=21.56 Rg=11.899 SPS=15593

```

(continues on next page)

(continued from previous page)

```

bl=0 pos[1]=[7.2 11.3 -1.1] dr=1.25 t=0.0ps kin=1.53 pot=21.57 Rg=11.921 SPS=15478
bl=0 pos[1]=[7.1 11.7 -2.2] dr=1.26 t=0.0ps kin=1.46 pot=21.60 Rg=11.905 SPS=15304
bl=0 pos[1]=[6.5 12.9 -1.4] dr=1.28 t=0.0ps kin=1.50 pot=21.56 Rg=11.928 SPS=15518
bl=0 pos[1]=[6.5 12.6 1.4] dr=1.29 t=0.0ps kin=1.50 pot=21.56 Rg=11.940 SPS=14940
bl=0 pos[1]=[4.9 11.8 1.3] dr=1.26 t=0.0ps kin=1.46 pot=21.58 Rg=11.929 SPS=14451
bl=0 pos[1]=[4.2 12.6 2.8] dr=1.23 t=0.0ps kin=1.47 pot=21.57 Rg=11.928 SPS=15003
bl=0 pos[1]=[4.9 14.6 3.1] dr=1.23 t=0.0ps kin=1.47 pot=21.61 Rg=11.941 SPS=14478
bl=0 pos[1]=[4.8 14.3 2.7] dr=1.24 t=0.0ps kin=1.48 pot=21.54 Rg=11.949 SPS=14911
bl=0 pos[1]=[3.4 14.2 1.6] dr=1.21 t=0.0ps kin=1.46 pot=21.59 Rg=11.951 SPS=14889
bl=0 pos[1]=[1.3 14.7 2.8] dr=1.23 t=0.0ps kin=1.49 pot=21.59 Rg=11.904 SPS=14624
bl=0 pos[1]=[1.9 15.2 4.3] dr=1.27 t=0.0ps kin=1.52 pot=21.62 Rg=11.927 SPS=15737
bl=0 pos[1]=[2.1 14.3 6.6] dr=1.28 t=0.0ps kin=1.52 pot=21.61 Rg=11.933 SPS=14514
bl=0 pos[1]=[3.6 15.4 7.1] dr=1.27 t=0.0ps kin=1.47 pot=21.61 Rg=11.940 SPS=14650
bl=0 pos[1]=[5.0 17.9 6.5] dr=1.27 t=0.0ps kin=1.49 pot=21.59 Rg=11.911 SPS=16222
bl=0 pos[1]=[5.4 18.5 4.6] dr=1.24 t=0.0ps kin=1.54 pot=21.53 Rg=11.840 SPS=14735
bl=0 pos[1]=[5.9 19.0 2.6] dr=1.25 t=0.0ps kin=1.49 pot=21.56 Rg=11.729 SPS=15246
bl=0 pos[1]=[5.8 17.7 4.1] dr=1.25 t=0.0ps kin=1.47 pot=21.57 Rg=11.650 SPS=15573
bl=0 pos[1]=[7.0 18.3 4.9] dr=1.26 t=0.0ps kin=1.51 pot=21.53 Rg=11.623 SPS=14427
bl=0 pos[1]=[7.0 18.3 3.2] dr=1.26 t=0.0ps kin=1.47 pot=21.59 Rg=11.616 SPS=15326
bl=0 pos[1]=[4.7 18.3 3.0] dr=1.22 t=0.0ps kin=1.48 pot=21.59 Rg=11.624 SPS=14654
bl=0 pos[1]=[3.0 18.9 1.5] dr=1.25 t=0.0ps kin=1.53 pot=21.60 Rg=11.624 SPS=15614
bl=0 pos[1]=[3.3 19.1 1.0] dr=1.26 t=0.0ps kin=1.51 pot=21.57 Rg=11.643 SPS=16030
bl=0 pos[1]=[3.7 18.2 0.3] dr=1.27 t=0.0ps kin=1.52 pot=21.59 Rg=11.633 SPS=14643
bl=0 pos[1]=[4.2 17.3 0.6] dr=1.24 t=0.0ps kin=1.44 pot=21.62 Rg=11.594 SPS=15126
bl=0 pos[1]=[3.2 17.5 1.5] dr=1.22 t=0.0ps kin=1.45 pot=21.57 Rg=11.590 SPS=13579
bl=0 pos[1]=[2.3 16.5 1.5] dr=1.23 t=0.0ps kin=1.46 pot=21.58 Rg=11.643 SPS=14483
bl=0 pos[1]=[2.3 16.2 2.3] dr=1.21 t=0.0ps kin=1.48 pot=21.61 Rg=11.668 SPS=14565
bl=0 pos[1]=[0.9 15.4 2.4] dr=1.27 t=0.0ps kin=1.49 pot=21.56 Rg=11.688 SPS=15540
bl=0 pos[1]=[-0.9 14.0 2.6] dr=1.27 t=0.0ps kin=1.47 pot=21.62 Rg=11.678 SPS=15674
bl=0 pos[1]=[-1.3 12.1 3.2] dr=1.24 t=0.0ps kin=1.50 pot=21.60 Rg=11.627 SPS=14832
bl=0 pos[1]=[-1.5 11.1 2.7] dr=1.21 t=0.0ps kin=1.50 pot=21.60 Rg=11.632 SPS=15716
bl=0 pos[1]=[-1.4 11.8 4.0] dr=1.25 t=0.0ps kin=1.53 pot=21.56 Rg=11.715 SPS=16122
bl=0 pos[1]=[0.3 12.8 4.5] dr=1.25 t=0.0ps kin=1.57 pot=21.54 Rg=11.768 SPS=14995
bl=0 pos[1]=[2.5 13.6 4.2] dr=1.30 t=0.0ps kin=1.51 pot=21.57 Rg=11.756 SPS=13949
bl=0 pos[1]=[2.8 13.5 4.5] dr=1.26 t=0.0ps kin=1.49 pot=21.60 Rg=11.756 SPS=14822
bl=0 pos[1]=[2.5 16.1 4.5] dr=1.30 t=0.0ps kin=1.52 pot=21.56 Rg=11.768 SPS=15616
bl=0 pos[1]=[2.8 17.0 4.0] dr=1.26 t=0.0ps kin=1.49 pot=21.57 Rg=11.789 SPS=15202
bl=0 pos[1]=[5.0 15.6 5.8] dr=1.26 t=0.0ps kin=1.49 pot=21.62 Rg=11.818 SPS=15381
bl=0 pos[1]=[6.6 14.4 7.4] dr=1.30 t=0.0ps kin=1.57 pot=21.58 Rg=11.791 SPS=15474
bl=0 pos[1]=[6.1 14.6 9.0] dr=1.26 t=0.0ps kin=1.50 pot=21.58 Rg=11.746 SPS=15067
bl=0 pos[1]=[5.7 13.3 9.1] dr=1.23 t=0.0ps kin=1.48 pot=21.56 Rg=11.727 SPS=14807
bl=0 pos[1]=[5.2 11.7 8.8] dr=1.26 t=0.0ps kin=1.48 pot=21.56 Rg=11.721 SPS=15734
bl=0 pos[1]=[5.9 11.0 7.3] dr=1.23 t=0.0ps kin=1.48 pot=21.56 Rg=11.698 SPS=15824
bl=0 pos[1]=[6.4 11.7 6.7] dr=1.27 t=0.0ps kin=1.49 pot=21.57 Rg=11.685 SPS=14496
bl=0 pos[1]=[5.1 10.6 7.1] dr=1.28 t=0.0ps kin=1.45 pot=21.58 Rg=11.718 SPS=15761
bl=0 pos[1]=[5.9 11.5 7.7] dr=1.28 t=0.0ps kin=1.50 pot=21.65 Rg=11.758 SPS=14670
bl=0 pos[1]=[5.2 12.4 9.5] dr=1.28 t=0.0ps kin=1.58 pot=21.62 Rg=11.786 SPS=15269
bl=0 pos[1]=[4.3 15.6 9.0] dr=1.28 t=0.0ps kin=1.58 pot=21.62 Rg=11.825 SPS=14865
bl=0 pos[1]=[3.1 16.4 8.7] dr=1.27 t=0.0ps kin=1.50 pot=21.63 Rg=11.858 SPS=14738
bl=0 pos[1]=[-0.2 16.2 7.4] dr=1.25 t=0.0ps kin=1.51 pot=21.57 Rg=11.884 SPS=15555
bl=0 pos[1]=[-1.8 15.8 7.6] dr=1.25 t=0.0ps kin=1.53 pot=21.59 Rg=11.942 SPS=15647
bl=0 pos[1]=[-2.7 16.2 7.5] dr=1.25 t=0.0ps kin=1.49 pot=21.61 Rg=12.043 SPS=14928

```

(continues on next page)

(continued from previous page)

```

bl=0 pos[1]=[-1.4 15.3 5.2] dr=1.26 t=0.0ps kin=1.50 pot=21.60 Rg=12.078 SPS=15003
bl=0 pos[1]=[-0.2 16.6 3.1] dr=1.26 t=0.0ps kin=1.54 pot=21.57 Rg=12.096 SPS=14607
bl=0 pos[1]=[-1.1 18.1 1.9] dr=1.26 t=0.0ps kin=1.51 pot=21.62 Rg=12.087 SPS=16222
bl=0 pos[1]=[-1.2 18.5 3.5] dr=1.28 t=0.0ps kin=1.55 pot=21.56 Rg=12.080 SPS=15733
bl=0 pos[1]=[-1.1 19.5 4.9] dr=1.24 t=0.0ps kin=1.49 pot=21.60 Rg=12.058 SPS=14348
bl=0 pos[1]=[0.4 19.3 5.7] dr=1.23 t=0.0ps kin=1.47 pot=21.64 Rg=12.019 SPS=15288
bl=0 pos[1]=[1.3 19.4 5.9] dr=1.21 t=0.0ps kin=1.51 pot=21.58 Rg=12.012 SPS=15142
bl=0 pos[1]=[3.2 19.3 6.5] dr=1.24 t=0.0ps kin=1.47 pot=21.56 Rg=12.011 SPS=14530
bl=0 pos[1]=[4.3 16.5 6.2] dr=1.26 t=0.0ps kin=1.46 pot=21.57 Rg=12.000 SPS=15494
bl=0 pos[1]=[4.4 16.8 7.6] dr=1.27 t=0.0ps kin=1.48 pot=21.60 Rg=11.985 SPS=14728
bl=0 pos[1]=[6.4 17.2 7.9] dr=1.25 t=0.0ps kin=1.58 pot=21.59 Rg=11.979 SPS=14530
bl=0 pos[1]=[6.3 18.8 7.0] dr=1.24 t=0.0ps kin=1.47 pot=21.62 Rg=11.980 SPS=15446
bl=0 pos[1]=[6.3 18.3 6.1] dr=1.25 t=0.0ps kin=1.50 pot=21.60 Rg=11.987 SPS=15151
bl=0 pos[1]=[5.4 16.6 5.5] dr=1.26 t=0.0ps kin=1.49 pot=21.58 Rg=12.003 SPS=15394
bl=0 pos[1]=[5.7 16.3 6.6] dr=1.25 t=0.0ps kin=1.53 pot=21.55 Rg=12.009 SPS=15505
bl=0 pos[1]=[4.9 17.0 6.5] dr=1.24 t=0.0ps kin=1.52 pot=21.57 Rg=11.978 SPS=15430
bl=0 pos[1]=[6.1 17.0 8.5] dr=1.25 t=0.0ps kin=1.50 pot=21.58 Rg=11.906 SPS=15184
bl=0 pos[1]=[5.2 16.0 10.7] dr=1.25 t=0.0ps kin=1.51 pot=21.60 Rg=11.851 SPS=14792
bl=0 pos[1]=[5.8 14.2 10.1] dr=1.25 t=0.0ps kin=1.53 pot=21.64 Rg=11.769 SPS=16055
bl=0 pos[1]=[6.8 14.2 9.2] dr=1.27 t=0.0ps kin=1.52 pot=21.61 Rg=11.732 SPS=15562
bl=0 pos[1]=[8.1 13.6 7.7] dr=1.29 t=0.0ps kin=1.49 pot=21.62 Rg=11.713 SPS=15357
bl=0 pos[1]=[8.7 11.6 6.4] dr=1.28 t=0.0ps kin=1.51 pot=21.64 Rg=11.724 SPS=15244
bl=0 pos[1]=[8.0 10.5 7.2] dr=1.29 t=0.0ps kin=1.52 pot=21.59 Rg=11.728 SPS=15944
bl=0 pos[1]=[8.4 10.5 7.2] dr=1.26 t=0.0ps kin=1.45 pot=21.59 Rg=11.685 SPS=15161
bl=0 pos[1]=[8.5 11.2 6.0] dr=1.26 t=0.0ps kin=1.49 pot=21.58 Rg=11.627 SPS=15916
bl=0 pos[1]=[7.5 12.1 5.7] dr=1.24 t=0.0ps kin=1.49 pot=21.57 Rg=11.596 SPS=15792
bl=0 pos[1]=[9.2 13.8 4.4] dr=1.27 t=0.0ps kin=1.53 pot=21.56 Rg=11.644 SPS=15226
bl=0 pos[1]=[10.3 14.3 5.7] dr=1.30 t=0.0ps kin=1.51 pot=21.57 Rg=11.698 SPS=16051
bl=0 pos[1]=[12.0 14.2 5.9] dr=1.27 t=0.0ps kin=1.53 pot=21.55 Rg=11.748 SPS=15785
bl=0 pos[1]=[12.6 14.4 6.3] dr=1.29 t=0.0ps kin=1.54 pot=21.55 Rg=11.754 SPS=14919
bl=0 pos[1]=[12.7 13.9 5.8] dr=1.26 t=0.0ps kin=1.49 pot=21.59 Rg=11.698 SPS=14968
bl=0 pos[1]=[11.5 13.4 5.2] dr=1.23 t=0.0ps kin=1.45 pot=21.58 Rg=11.655 SPS=15456
bl=0 pos[1]=[10.8 14.4 4.5] dr=1.24 t=0.0ps kin=1.44 pot=21.56 Rg=11.647 SPS=15596
bl=0 pos[1]=[10.5 12.2 3.7] dr=1.25 t=0.0ps kin=1.50 pot=21.55 Rg=11.645 SPS=15459
bl=0 pos[1]=[9.1 11.7 1.9] dr=1.24 t=0.0ps kin=1.53 pot=21.53 Rg=11.595 SPS=15617
bl=0 pos[1]=[7.7 12.9 -0.3] dr=1.26 t=0.0ps kin=1.55 pot=21.60 Rg=11.539 SPS=14885
bl=0 pos[1]=[8.1 13.5 -0.5] dr=1.28 t=0.0ps kin=1.51 pot=21.65 Rg=11.524 SPS=15627
bl=0 pos[1]=[7.3 12.0 -1.2] dr=1.28 t=0.0ps kin=1.56 pot=21.58 Rg=11.514 SPS=13846
bl=0 pos[1]=[6.5 11.7 -1.3] dr=1.29 t=0.0ps kin=1.50 pot=21.57 Rg=11.507 SPS=15385
bl=0 pos[1]=[5.5 10.5 -0.8] dr=1.28 t=0.0ps kin=1.47 pot=21.59 Rg=11.480 SPS=15372
bl=0 pos[1]=[6.0 10.3 0.8] dr=1.27 t=0.0ps kin=1.53 pot=21.56 Rg=11.428 SPS=14014
bl=0 pos[1]=[6.8 11.4 2.5] dr=1.26 t=0.0ps kin=1.50 pot=21.60 Rg=11.403 SPS=14677
bl=0 pos[1]=[6.6 10.9 0.3] dr=1.27 t=0.0ps kin=1.52 pot=21.61 Rg=11.429 SPS=14508
bl=0 pos[1]=[7.6 12.9 -0.2] dr=1.25 t=0.0ps kin=1.48 pot=21.61 Rg=11.502 SPS=15046
bl=0 pos[1]=[9.2 13.5 -1.6] dr=1.27 t=0.0ps kin=1.51 pot=21.57 Rg=11.571 SPS=14682
bl=0 pos[1]=[8.7 13.0 -2.2] dr=1.29 t=0.0ps kin=1.51 pot=21.59 Rg=11.597 SPS=14980
bl=0 pos[1]=[8.8 12.2 -1.9] dr=1.27 t=0.0ps kin=1.46 pot=21.62 Rg=11.592 SPS=14717
bl=0 pos[1]=[9.1 13.0 -3.1] dr=1.26 t=0.0ps kin=1.51 pot=21.58 Rg=11.599 SPS=13925
bl=0 pos[1]=[10.4 13.0 -2.6] dr=1.27 t=0.0ps kin=1.53 pot=21.59 Rg=11.589 SPS=14274
bl=0 pos[1]=[10.0 13.1 -3.1] dr=1.28 t=0.0ps kin=1.52 pot=21.61 Rg=11.542 SPS=14183
bl=0 pos[1]=[9.0 12.2 -3.2] dr=1.26 t=0.0ps kin=1.51 pot=21.59 Rg=11.479 SPS=15679
bl=0 pos[1]=[7.2 12.6 -4.5] dr=1.26 t=0.0ps kin=1.51 pot=21.61 Rg=11.404 SPS=15416

```

(continues on next page)

(continued from previous page)

```

bl=0 pos[1]=[6.2 14.6 -4.4] dr=1.28 t=0.0ps kin=1.52 pot=21.63 Rg=11.348 SPS=16134
bl=0 pos[1]=[6.0 14.5 -4.8] dr=1.27 t=0.0ps kin=1.52 pot=21.56 Rg=11.367 SPS=14773
bl=0 pos[1]=[7.0 13.6 -3.6] dr=1.27 t=0.0ps kin=1.50 pot=21.54 Rg=11.342 SPS=14545
bl=0 pos[1]=[7.7 13.2 -2.9] dr=1.27 t=0.0ps kin=1.51 pot=21.50 Rg=11.309 SPS=13732
bl=0 pos[1]=[8.2 13.7 -4.0] dr=1.26 t=0.0ps kin=1.45 pot=21.57 Rg=11.276 SPS=13899
bl=0 pos[1]=[8.9 14.5 -3.2] dr=1.24 t=0.0ps kin=1.50 pot=21.54 Rg=11.265 SPS=15524
bl=0 pos[1]=[7.9 13.2 -2.2] dr=1.24 t=0.0ps kin=1.47 pot=21.63 Rg=11.261 SPS=14919
bl=0 pos[1]=[9.7 12.0 -1.6] dr=1.26 t=0.0ps kin=1.49 pot=21.59 Rg=11.252 SPS=14731
bl=0 pos[1]=[11.3 12.0 -1.3] dr=1.26 t=0.0ps kin=1.50 pot=21.58 Rg=11.293 SPS=15438
bl=0 pos[1]=[11.5 10.7 -0.3] dr=1.25 t=0.0ps kin=1.51 pot=21.60 Rg=11.355 SPS=14726
bl=0 pos[1]=[12.6 11.8 -0.0] dr=1.27 t=0.0ps kin=1.51 pot=21.61 Rg=11.453 SPS=15509
bl=0 pos[1]=[12.1 13.2 -0.6] dr=1.26 t=0.0ps kin=1.48 pot=21.60 Rg=11.555 SPS=15026
bl=0 pos[1]=[11.3 14.9 0.7] dr=1.24 t=0.0ps kin=1.49 pot=21.56 Rg=11.594 SPS=14180
bl=0 pos[1]=[9.8 14.1 2.2] dr=1.25 t=0.0ps kin=1.50 pot=21.60 Rg=11.619 SPS=15133
bl=0 pos[1]=[10.3 13.5 2.7] dr=1.26 t=0.0ps kin=1.48 pot=21.61 Rg=11.687 SPS=14096
bl=0 pos[1]=[10.3 14.4 3.0] dr=1.23 t=0.0ps kin=1.49 pot=21.58 Rg=11.754 SPS=14581
bl=0 pos[1]=[10.0 14.1 3.9] dr=1.30 t=0.0ps kin=1.52 pot=21.56 Rg=11.824 SPS=15088
bl=0 pos[1]=[9.8 13.6 4.2] dr=1.29 t=0.0ps kin=1.49 pot=21.58 Rg=11.925 SPS=15267
bl=0 pos[1]=[9.2 12.9 4.7] dr=1.28 t=0.0ps kin=1.50 pot=21.58 Rg=12.006 SPS=15473
bl=0 pos[1]=[8.4 13.5 5.3] dr=1.23 t=0.0ps kin=1.47 pot=21.64 Rg=12.017 SPS=14429
bl=0 pos[1]=[7.9 13.6 5.1] dr=1.25 t=0.0ps kin=1.51 pot=21.61 Rg=12.007 SPS=15493
bl=0 pos[1]=[7.7 14.4 5.1] dr=1.22 t=0.0ps kin=1.50 pot=21.54 Rg=12.020 SPS=15450
bl=0 pos[1]=[8.6 13.9 5.7] dr=1.27 t=0.0ps kin=1.50 pot=21.55 Rg=12.052 SPS=14219
bl=0 pos[1]=[9.0 12.8 5.4] dr=1.24 t=0.0ps kin=1.43 pot=21.58 Rg=12.040 SPS=15895
bl=0 pos[1]=[7.7 11.5 5.5] dr=1.25 t=0.0ps kin=1.46 pot=21.56 Rg=12.008 SPS=15294
bl=0 pos[1]=[6.9 10.5 5.1] dr=1.27 t=0.0ps kin=1.48 pot=21.59 Rg=12.040 SPS=14041
bl=0 pos[1]=[5.2 11.6 4.3] dr=1.26 t=0.0ps kin=1.45 pot=21.62 Rg=12.074 SPS=15221
bl=0 pos[1]=[4.8 12.7 4.2] dr=1.23 t=0.0ps kin=1.48 pot=21.62 Rg=12.095 SPS=13564
bl=0 pos[1]=[6.1 14.3 4.9] dr=1.26 t=0.0ps kin=1.57 pot=21.57 Rg=12.159 SPS=15436
bl=0 pos[1]=[6.6 12.5 5.3] dr=1.25 t=0.0ps kin=1.48 pot=21.59 Rg=12.197 SPS=14542
bl=0 pos[1]=[7.1 13.1 4.7] dr=1.22 t=0.0ps kin=1.51 pot=21.59 Rg=12.208 SPS=14816
bl=0 pos[1]=[7.1 13.6 3.9] dr=1.29 t=0.0ps kin=1.52 pot=21.58 Rg=12.203 SPS=15181
bl=0 pos[1]=[6.3 14.7 4.1] dr=1.29 t=0.0ps kin=1.51 pot=21.58 Rg=12.224 SPS=14865
bl=0 pos[1]=[4.8 15.8 3.9] dr=1.27 t=0.0ps kin=1.53 pot=21.56 Rg=12.261 SPS=15235
bl=0 pos[1]=[4.9 16.1 1.9] dr=1.31 t=0.0ps kin=1.50 pot=21.59 Rg=12.343 SPS=14562
bl=0 pos[1]=[4.0 15.8 0.8] dr=1.32 t=0.0ps kin=1.52 pot=21.60 Rg=12.371 SPS=15078
bl=0 pos[1]=[4.6 14.2 -1.1] dr=1.28 t=0.0ps kin=1.57 pot=21.57 Rg=12.331 SPS=15550
bl=0 pos[1]=[6.8 14.2 -2.1] dr=1.28 t=0.0ps kin=1.50 pot=21.61 Rg=12.252 SPS=14613
bl=0 pos[1]=[6.7 15.1 -2.8] dr=1.26 t=0.0ps kin=1.48 pot=21.66 Rg=12.146 SPS=14280
bl=0 pos[1]=[6.8 15.4 -3.2] dr=1.27 t=0.0ps kin=1.49 pot=21.63 Rg=12.076 SPS=13838
bl=0 pos[1]=[7.0 15.6 -2.4] dr=1.24 t=0.0ps kin=1.51 pot=21.63 Rg=12.019 SPS=15198
bl=0 pos[1]=[8.0 15.0 -1.7] dr=1.24 t=0.0ps kin=1.47 pot=21.63 Rg=12.013 SPS=15690
bl=0 pos[1]=[8.9 16.4 -2.3] dr=1.24 t=0.0ps kin=1.54 pot=21.60 Rg=11.993 SPS=15286
bl=0 pos[1]=[9.5 14.9 -1.9] dr=1.28 t=0.0ps kin=1.53 pot=21.63 Rg=11.987 SPS=15493
bl=0 pos[1]=[10.3 13.4 -0.5] dr=1.31 t=0.0ps kin=1.56 pot=21.64 Rg=11.962 SPS=13843
bl=0 pos[1]=[9.8 13.0 -0.7] dr=1.28 t=0.0ps kin=1.53 pot=21.63 Rg=11.964 SPS=14480
bl=0 pos[1]=[9.6 12.9 -0.2] dr=1.26 t=0.0ps kin=1.50 pot=21.62 Rg=11.974 SPS=15336
bl=0 pos[1]=[9.3 13.5 -0.0] dr=1.24 t=0.0ps kin=1.51 pot=21.62 Rg=11.959 SPS=15035
bl=0 pos[1]=[7.9 12.3 -0.3] dr=1.25 t=0.0ps kin=1.52 pot=21.60 Rg=11.874 SPS=15215
bl=0 pos[1]=[9.3 10.4 0.2] dr=1.26 t=0.0ps kin=1.51 pot=21.58 Rg=11.855 SPS=14117
bl=0 pos[1]=[11.6 12.2 -1.0] dr=1.27 t=0.0ps kin=1.51 pot=21.59 Rg=11.853 SPS=15139
bl=0 pos[1]=[12.8 12.8 -3.5] dr=1.26 t=0.0ps kin=1.49 pot=21.59 Rg=11.843 SPS=15094

```

(continues on next page)



(continued from previous page)

```

bl=0 pos[1]=[12.0 11.2 -5.3] dr=1.25 t=0.0ps kin=1.51 pot=21.54 Rg=11.866 SPS=14067
bl=0 pos[1]=[12.7 10.1 -5.1] dr=1.27 t=0.0ps kin=1.50 pot=21.58 Rg=11.896 SPS=15537
bl=0 pos[1]=[14.2 10.6 -3.1] dr=1.23 t=0.0ps kin=1.51 pot=21.58 Rg=11.919 SPS=15163
bl=0 pos[1]=[14.8 10.2 -2.2] dr=1.23 t=0.0ps kin=1.46 pot=21.60 Rg=11.928 SPS=15143
bl=0 pos[1]=[15.8 10.2 -4.5] dr=1.26 t=0.0ps kin=1.49 pot=21.61 Rg=11.903 SPS=15407
bl=0 pos[1]=[14.8 9.7 -2.9] dr=1.24 t=0.0ps kin=1.54 pot=21.60 Rg=11.861 SPS=14401
bl=0 pos[1]=[14.6 9.7 -2.6] dr=1.22 t=0.0ps kin=1.48 pot=21.63 Rg=11.843 SPS=15047
bl=0 pos[1]=[13.2 9.4 -3.2] dr=1.22 t=0.0ps kin=1.53 pot=21.63 Rg=11.825 SPS=13932
bl=0 pos[1]=[13.2 9.2 -3.4] dr=1.28 t=0.0ps kin=1.50 pot=21.63 Rg=11.796 SPS=14650
bl=0 pos[1]=[13.2 8.5 -3.3] dr=1.25 t=0.0ps kin=1.53 pot=21.57 Rg=11.761 SPS=15033
bl=0 pos[1]=[13.8 8.7 -3.5] dr=1.26 t=0.0ps kin=1.56 pot=21.64 Rg=11.664 SPS=14740
bl=0 pos[1]=[14.9 8.4 -3.5] dr=1.29 t=0.0ps kin=1.58 pot=21.61 Rg=11.591 SPS=15200
bl=0 pos[1]=[15.5 9.3 -4.0] dr=1.29 t=0.0ps kin=1.49 pot=21.59 Rg=11.538 SPS=14298
bl=0 pos[1]=[16.5 10.2 -3.8] dr=1.22 t=0.0ps kin=1.46 pot=21.58 Rg=11.537 SPS=15419
bl=0 pos[1]=[16.2 9.0 -3.8] dr=1.27 t=0.0ps kin=1.49 pot=21.54 Rg=11.516 SPS=15181
bl=0 pos[1]=[15.1 8.0 -5.6] dr=1.26 t=0.0ps kin=1.47 pot=21.59 Rg=11.468 SPS=14796
bl=0 pos[1]=[13.1 7.6 -6.1] dr=1.25 t=0.0ps kin=1.48 pot=21.61 Rg=11.479 SPS=14794
bl=0 pos[1]=[13.1 8.3 -5.3] dr=1.23 t=0.0ps kin=1.48 pot=21.59 Rg=11.508 SPS=14382
bl=0 pos[1]=[13.0 9.2 -5.6] dr=1.24 t=0.0ps kin=1.54 pot=21.56 Rg=11.515 SPS=15385
bl=0 pos[1]=[14.1 10.2 -6.4] dr=1.26 t=0.0ps kin=1.53 pot=21.53 Rg=11.566 SPS=15152
bl=0 pos[1]=[14.7 10.4 -7.1] dr=1.24 t=0.0ps kin=1.42 pot=21.62 Rg=11.614 SPS=14194
bl=0 pos[1]=[14.2 10.1 -6.1] dr=1.25 t=0.0ps kin=1.46 pot=21.62 Rg=11.613 SPS=15414
bl=0 pos[1]=[14.6 9.0 -3.5] dr=1.23 t=0.0ps kin=1.49 pot=21.58 Rg=11.600 SPS=14618
bl=0 pos[1]=[14.3 8.3 -2.3] dr=1.23 t=0.0ps kin=1.53 pot=21.59 Rg=11.579 SPS=15365
bl=0 pos[1]=[15.0 9.1 -1.5] dr=1.24 t=0.0ps kin=1.50 pot=21.59 Rg=11.530 SPS=15023
bl=0 pos[1]=[16.4 9.6 -1.4] dr=1.24 t=0.0ps kin=1.51 pot=21.63 Rg=11.517 SPS=14392
bl=0 pos[1]=[17.1 9.0 -1.8] dr=1.28 t=0.0ps kin=1.53 pot=21.63 Rg=11.553 SPS=15165
bl=0 pos[1]=[17.4 10.1 -2.8] dr=1.29 t=0.0ps kin=1.53 pot=21.67 Rg=11.569 SPS=14889
bl=0 pos[1]=[17.3 8.4 -3.9] dr=1.27 t=0.0ps kin=1.52 pot=21.60 Rg=11.567 SPS=15775
bl=0 pos[1]=[16.4 7.5 -3.3] dr=1.25 t=0.0ps kin=1.57 pot=21.60 Rg=11.534 SPS=16002
bl=0 pos[1]=[16.1 7.1 -2.8] dr=1.28 t=0.0ps kin=1.58 pot=21.61 Rg=11.528 SPS=14810
bl=0 pos[1]=[15.8 6.5 -2.3] dr=1.27 t=0.0ps kin=1.56 pot=21.58 Rg=11.512 SPS=16148
bl=0 pos[1]=[16.7 7.0 -1.7] dr=1.27 t=0.0ps kin=1.51 pot=21.62 Rg=11.469 SPS=14941
bl=0 pos[1]=[16.8 8.3 -2.3] dr=1.27 t=0.0ps kin=1.54 pot=21.58 Rg=11.495 SPS=14530
bl=0 pos[1]=[16.7 8.1 0.1] dr=1.26 t=0.0ps kin=1.51 pot=21.61 Rg=11.548 SPS=15569
bl=0 pos[1]=[15.3 8.6 2.5] dr=1.31 t=0.0ps kin=1.54 pot=21.62 Rg=11.606 SPS=15542
bl=0 pos[1]=[13.8 7.5 4.1] dr=1.28 t=0.0ps kin=1.50 pot=21.62 Rg=11.611 SPS=15047
bl=0 pos[1]=[12.5 7.4 4.8] dr=1.26 t=0.0ps kin=1.54 pot=21.63 Rg=11.615 SPS=15173
bl=0 pos[1]=[10.5 7.3 6.9] dr=1.26 t=0.0ps kin=1.49 pot=21.58 Rg=11.616 SPS=15766
bl=0 pos[1]=[9.5 6.5 6.8] dr=1.28 t=0.0ps kin=1.52 pot=21.57 Rg=11.648 SPS=14900
bl=0 pos[1]=[7.7 8.1 5.4] dr=1.27 t=0.0ps kin=1.52 pot=21.57 Rg=11.702 SPS=15392
bl=0 pos[1]=[7.4 10.8 6.7] dr=1.26 t=0.0ps kin=1.46 pot=21.62 Rg=11.756 SPS=15789
bl=0 pos[1]=[6.7 13.2 5.5] dr=1.25 t=0.0ps kin=1.52 pot=21.56 Rg=11.782 SPS=15868
bl=0 pos[1]=[7.1 14.8 4.4] dr=1.24 t=0.0ps kin=1.51 pot=21.59 Rg=11.711 SPS=15578
bl=0 pos[1]=[7.7 15.2 4.9] dr=1.27 t=0.0ps kin=1.45 pot=21.65 Rg=11.679 SPS=15016
bl=0 pos[1]=[6.9 15.1 5.7] dr=1.23 t=0.0ps kin=1.51 pot=21.63 Rg=11.663 SPS=15924
bl=0 pos[1]=[7.0 16.5 6.2] dr=1.27 t=0.0ps kin=1.50 pot=21.61 Rg=11.624 SPS=15485
bl=0 pos[1]=[7.8 15.9 5.2] dr=1.29 t=0.0ps kin=1.49 pot=21.59 Rg=11.553 SPS=15240
bl=0 pos[1]=[7.1 15.6 7.5] dr=1.24 t=0.0ps kin=1.53 pot=21.61 Rg=11.502 SPS=15795
bl=0 pos[1]=[6.8 14.5 7.4] dr=1.24 t=0.0ps kin=1.50 pot=21.67 Rg=11.530 SPS=15946
bl=0 pos[1]=[6.5 14.4 7.2] dr=1.26 t=0.0ps kin=1.55 pot=21.62 Rg=11.534 SPS=14095
bl=0 pos[1]=[6.7 14.8 6.0] dr=1.28 t=0.0ps kin=1.52 pot=21.62 Rg=11.528 SPS=15763

```

(continues on next page)

(continued from previous page)

```

bl=0 pos[1]=[7.2 14.4 7.0] dr=1.24 t=0.0ps kin=1.52 pot=21.58 Rg=11.538 SPS=14915
bl=0 pos[1]=[6.5 12.6 7.2] dr=1.28 t=0.0ps kin=1.51 pot=21.63 Rg=11.556 SPS=14796
bl=0 pos[1]=[5.8 11.9 9.8] dr=1.26 t=0.0ps kin=1.50 pot=21.64 Rg=11.574 SPS=15816
bl=0 pos[1]=[4.6 11.5 9.3] dr=1.30 t=0.0ps kin=1.52 pot=21.60 Rg=11.539 SPS=15681
bl=0 pos[1]=[5.6 11.7 7.9] dr=1.28 t=0.0ps kin=1.48 pot=21.63 Rg=11.493 SPS=15412
bl=0 pos[1]=[6.3 10.5 7.6] dr=1.27 t=0.0ps kin=1.55 pot=21.58 Rg=11.470 SPS=15985
bl=0 pos[1]=[5.7 10.3 8.3] dr=1.27 t=0.0ps kin=1.55 pot=21.58 Rg=11.474 SPS=15454
bl=0 pos[1]=[5.9 10.6 7.7] dr=1.26 t=0.0ps kin=1.51 pot=21.59 Rg=11.447 SPS=15298
bl=0 pos[1]=[5.8 12.6 6.0] dr=1.26 t=0.0ps kin=1.49 pot=21.57 Rg=11.393 SPS=14544
bl=0 pos[1]=[4.2 12.8 7.0] dr=1.30 t=0.0ps kin=1.53 pot=21.54 Rg=11.342 SPS=14763
bl=0 pos[1]=[3.6 14.0 7.6] dr=1.28 t=0.0ps kin=1.56 pot=21.56 Rg=11.299 SPS=15452
bl=0 pos[1]=[4.1 15.4 6.7] dr=1.23 t=0.0ps kin=1.52 pot=21.51 Rg=11.274 SPS=15141
bl=0 pos[1]=[5.1 16.5 6.7] dr=1.24 t=0.0ps kin=1.52 pot=21.59 Rg=11.243 SPS=14290
bl=0 pos[1]=[5.0 17.1 6.6] dr=1.27 t=0.0ps kin=1.52 pot=21.53 Rg=11.273 SPS=14552
bl=0 pos[1]=[4.5 16.9 7.0] dr=1.23 t=0.0ps kin=1.41 pot=21.61 Rg=11.266 SPS=14291
bl=0 pos[1]=[3.7 15.2 7.0] dr=1.22 t=0.0ps kin=1.45 pot=21.60 Rg=11.295 SPS=14788
bl=0 pos[1]=[2.9 14.3 5.9] dr=1.24 t=0.0ps kin=1.53 pot=21.58 Rg=11.359 SPS=14975
bl=0 pos[1]=[2.4 14.7 5.7] dr=1.23 t=0.0ps kin=1.49 pot=21.61 Rg=11.396 SPS=15083
bl=0 pos[1]=[2.5 14.1 5.4] dr=1.22 t=0.0ps kin=1.51 pot=21.63 Rg=11.381 SPS=15412
bl=0 pos[1]=[3.0 14.1 5.5] dr=1.25 t=0.0ps kin=1.54 pot=21.58 Rg=11.362 SPS=14164
bl=0 pos[1]=[4.7 13.3 6.3] dr=1.24 t=0.0ps kin=1.48 pot=21.60 Rg=11.344 SPS=15530
bl=0 pos[1]=[7.0 15.1 5.9] dr=1.27 t=0.0ps kin=1.52 pot=21.56 Rg=11.321 SPS=15475
bl=0 pos[1]=[9.4 13.7 4.7] dr=1.29 t=0.0ps kin=1.50 pot=21.53 Rg=11.257 SPS=14924
bl=0 pos[1]=[10.0 14.0 4.4] dr=1.27 t=0.0ps kin=1.48 pot=21.57 Rg=11.220 SPS=15032
bl=0 pos[1]=[9.1 12.9 4.2] dr=1.28 t=0.0ps kin=1.54 pot=21.59 Rg=11.169 SPS=14250
bl=0 pos[1]=[8.0 12.7 1.9] dr=1.29 t=0.0ps kin=1.54 pot=21.58 Rg=11.142 SPS=15110
bl=0 pos[1]=[7.9 13.8 2.8] dr=1.29 t=0.0ps kin=1.50 pot=21.59 Rg=11.149 SPS=15232
bl=0 pos[1]=[9.0 13.2 4.5] dr=1.29 t=0.0ps kin=1.56 pot=21.56 Rg=11.201 SPS=14520
bl=0 pos[1]=[10.8 11.3 3.1] dr=1.27 t=0.0ps kin=1.52 pot=21.56 Rg=11.253 SPS=15080
bl=0 pos[1]=[10.8 10.2 3.0] dr=1.27 t=0.0ps kin=1.52 pot=21.58 Rg=11.294 SPS=14424
bl=0 pos[1]=[10.9 8.9 4.1] dr=1.31 t=0.0ps kin=1.48 pot=21.59 Rg=11.300 SPS=14721
bl=0 pos[1]=[10.4 8.3 5.9] dr=1.22 t=0.0ps kin=1.50 pot=21.57 Rg=11.270 SPS=13729
bl=0 pos[1]=[9.4 9.1 7.3] dr=1.22 t=0.0ps kin=1.52 pot=21.55 Rg=11.234 SPS=14649
bl=0 pos[1]=[10.1 10.0 7.6] dr=1.29 t=0.0ps kin=1.52 pot=21.56 Rg=11.169 SPS=15054
bl=0 pos[1]=[10.6 10.5 6.3] dr=1.29 t=0.0ps kin=1.54 pot=21.55 Rg=11.110 SPS=13729
bl=0 pos[1]=[9.6 10.5 6.4] dr=1.27 t=0.0ps kin=1.49 pot=21.56 Rg=11.100 SPS=14563
bl=0 pos[1]=[9.1 9.7 5.5] dr=1.23 t=0.0ps kin=1.50 pot=21.54 Rg=11.140 SPS=13892
bl=0 pos[1]=[8.5 8.6 4.4] dr=1.24 t=0.0ps kin=1.47 pot=21.53 Rg=11.189 SPS=15372
bl=0 pos[1]=[8.4 7.3 2.7] dr=1.26 t=0.0ps kin=1.47 pot=21.59 Rg=11.219 SPS=14535
bl=0 pos[1]=[6.9 6.1 1.7] dr=1.24 t=0.0ps kin=1.44 pot=21.56 Rg=11.298 SPS=15139
bl=0 pos[1]=[6.2 6.2 2.3] dr=1.20 t=0.0ps kin=1.43 pot=21.58 Rg=11.353 SPS=15267
bl=0 pos[1]=[6.9 5.7 1.9] dr=1.21 t=0.0ps kin=1.53 pot=21.56 Rg=11.377 SPS=14895
bl=0 pos[1]=[7.2 5.4 1.5] dr=1.28 t=0.0ps kin=1.50 pot=21.55 Rg=11.399 SPS=15297
bl=0 pos[1]=[8.7 5.7 1.4] dr=1.22 t=0.0ps kin=1.44 pot=21.58 Rg=11.387 SPS=15510
bl=0 pos[1]=[9.3 8.0 2.9] dr=1.21 t=0.0ps kin=1.52 pot=21.56 Rg=11.358 SPS=15264
bl=0 pos[1]=[8.0 7.3 3.0] dr=1.25 t=0.0ps kin=1.48 pot=21.55 Rg=11.328 SPS=15202
bl=0 pos[1]=[7.3 7.8 4.1] dr=1.25 t=0.0ps kin=1.45 pot=21.60 Rg=11.318 SPS=14731
bl=0 pos[1]=[5.9 7.8 2.6] dr=1.25 t=0.0ps kin=1.51 pot=21.61 Rg=11.329 SPS=15523
bl=0 pos[1]=[4.7 8.4 4.0] dr=1.29 t=0.0ps kin=1.55 pot=21.57 Rg=11.348 SPS=14674
bl=0 pos[1]=[3.7 10.7 5.5] dr=1.27 t=0.0ps kin=1.47 pot=21.56 Rg=11.320 SPS=14519
bl=0 pos[1]=[5.0 11.6 4.6] dr=1.27 t=0.0ps kin=1.49 pot=21.57 Rg=11.254 SPS=14949
bl=0 pos[1]=[4.0 12.1 5.5] dr=1.28 t=0.0ps kin=1.45 pot=21.59 Rg=11.223 SPS=14205

```

(continues on next page)

(continued from previous page)

```

bl=0 pos[1]=[2.5 12.2 5.4] dr=1.29 t=0.0ps kin=1.55 pot=21.55 Rg=11.221 SPS=15084
bl=0 pos[1]=[2.5 11.6 4.8] dr=1.26 t=0.0ps kin=1.48 pot=21.57 Rg=11.180 SPS=15772
bl=0 pos[1]=[1.6 12.1 4.1] dr=1.28 t=0.0ps kin=1.56 pot=21.57 Rg=11.139 SPS=14671
bl=0 pos[1]=[2.5 11.6 4.0] dr=1.28 t=0.0ps kin=1.52 pot=21.61 Rg=11.111 SPS=14798
bl=0 pos[1]=[4.7 13.0 3.0] dr=1.24 t=0.0ps kin=1.51 pot=21.58 Rg=11.084 SPS=14057
bl=0 pos[1]=[6.0 15.3 4.5] dr=1.25 t=0.0ps kin=1.53 pot=21.57 Rg=11.027 SPS=14868
bl=0 pos[1]=[6.6 14.7 7.8] dr=1.24 t=0.0ps kin=1.47 pot=21.62 Rg=10.956 SPS=14636
bl=0 pos[1]=[6.0 14.0 8.8] dr=1.25 t=0.0ps kin=1.51 pot=21.58 Rg=10.903 SPS=15513
bl=0 pos[1]=[5.0 13.6 8.2] dr=1.26 t=0.0ps kin=1.50 pot=21.56 Rg=10.861 SPS=14851
bl=0 pos[1]=[3.5 13.0 8.2] dr=1.24 t=0.0ps kin=1.53 pot=21.59 Rg=10.886 SPS=14377
bl=0 pos[1]=[3.6 11.5 7.5] dr=1.26 t=0.0ps kin=1.52 pot=21.57 Rg=10.917 SPS=12716
bl=0 pos[1]=[3.0 10.0 5.8] dr=1.27 t=0.0ps kin=1.51 pot=21.58 Rg=10.914 SPS=12583
bl=0 pos[1]=[2.1 11.0 5.4] dr=1.27 t=0.0ps kin=1.54 pot=21.54 Rg=10.925 SPS=14218
bl=0 pos[1]=[2.5 10.0 3.3] dr=1.25 t=0.0ps kin=1.51 pot=21.56 Rg=10.973 SPS=14327
bl=0 pos[1]=[2.8 10.2 3.8] dr=1.23 t=0.0ps kin=1.57 pot=21.59 Rg=11.021 SPS=14723
bl=0 pos[1]=[3.2 10.0 3.0] dr=1.27 t=0.0ps kin=1.53 pot=21.63 Rg=11.071 SPS=14693
bl=0 pos[1]=[3.5 10.6 4.3] dr=1.23 t=0.0ps kin=1.57 pot=21.62 Rg=11.169 SPS=14080
bl=0 pos[1]=[3.9 12.5 4.1] dr=1.25 t=0.0ps kin=1.49 pot=21.61 Rg=11.255 SPS=13813
bl=0 pos[1]=[4.7 12.8 6.0] dr=1.25 t=0.0ps kin=1.51 pot=21.63 Rg=11.350 SPS=15194
bl=0 pos[1]=[5.1 14.3 5.9] dr=1.27 t=0.0ps kin=1.52 pot=21.60 Rg=11.433 SPS=15519
bl=0 pos[1]=[6.3 14.8 5.4] dr=1.25 t=0.0ps kin=1.51 pot=21.67 Rg=11.472 SPS=14805
bl=0 pos[1]=[5.9 12.3 3.6] dr=1.26 t=0.0ps kin=1.55 pot=21.58 Rg=11.491 SPS=14567
bl=0 pos[1]=[5.1 12.0 4.3] dr=1.24 t=0.0ps kin=1.47 pot=21.60 Rg=11.509 SPS=14593
bl=0 pos[1]=[7.4 11.7 4.0] dr=1.22 t=0.0ps kin=1.55 pot=21.57 Rg=11.526 SPS=15199
bl=0 pos[1]=[8.9 11.7 3.6] dr=1.24 t=0.0ps kin=1.56 pot=21.56 Rg=11.519 SPS=15570
bl=0 pos[1]=[8.8 11.7 3.4] dr=1.26 t=0.0ps kin=1.53 pot=21.61 Rg=11.507 SPS=14521
bl=0 pos[1]=[8.4 11.6 3.3] dr=1.24 t=0.0ps kin=1.54 pot=21.58 Rg=11.446 SPS=14533
bl=0 pos[1]=[9.8 13.0 3.5] dr=1.21 t=0.0ps kin=1.46 pot=21.56 Rg=11.423 SPS=14008
bl=0 pos[1]=[9.7 13.2 3.7] dr=1.22 t=0.0ps kin=1.48 pot=21.58 Rg=11.414 SPS=14704
bl=0 pos[1]=[9.8 13.5 2.5] dr=1.25 t=0.0ps kin=1.53 pot=21.57 Rg=11.467 SPS=13919
bl=0 pos[1]=[10.3 12.3 3.4] dr=1.28 t=0.0ps kin=1.50 pot=21.58 Rg=11.477 SPS=15753
bl=0 pos[1]=[11.9 13.2 3.0] dr=1.30 t=0.0ps kin=1.47 pot=21.63 Rg=11.499 SPS=15615
bl=0 pos[1]=[11.2 12.8 2.4] dr=1.26 t=0.0ps kin=1.51 pot=21.60 Rg=11.559 SPS=15018
bl=0 pos[1]=[9.0 13.3 2.3] dr=1.26 t=0.0ps kin=1.56 pot=21.57 Rg=11.618 SPS=15626
bl=0 pos[1]=[11.6 12.7 1.7] dr=1.28 t=0.0ps kin=1.54 pot=21.63 Rg=11.612 SPS=15235
bl=0 pos[1]=[10.2 11.8 3.4] dr=1.27 t=0.0ps kin=1.53 pot=21.64 Rg=11.646 SPS=15630
bl=0 pos[1]=[10.0 13.1 4.4] dr=1.24 t=0.0ps kin=1.52 pot=21.67 Rg=11.681 SPS=15562
bl=0 pos[1]=[10.0 14.4 5.3] dr=1.27 t=0.0ps kin=1.52 pot=21.62 Rg=11.703 SPS=15510
bl=0 pos[1]=[10.4 16.3 4.6] dr=1.26 t=0.0ps kin=1.55 pot=21.61 Rg=11.707 SPS=15191
bl=0 pos[1]=[9.4 16.7 2.9] dr=1.30 t=0.0ps kin=1.53 pot=21.56 Rg=11.679 SPS=15225
bl=0 pos[1]=[8.7 17.3 3.1] dr=1.28 t=0.0ps kin=1.49 pot=21.57 Rg=11.697 SPS=14355
bl=0 pos[1]=[9.0 16.5 4.7] dr=1.29 t=0.0ps kin=1.46 pot=21.55 Rg=11.700 SPS=15479
bl=0 pos[1]=[9.3 17.0 5.1] dr=1.27 t=0.0ps kin=1.42 pot=21.52 Rg=11.673 SPS=15146
bl=0 pos[1]=[8.2 18.2 4.6] dr=1.27 t=0.0ps kin=1.46 pot=21.53 Rg=11.650 SPS=14949
bl=0 pos[1]=[7.4 16.1 3.2] dr=1.23 t=0.0ps kin=1.44 pot=21.56 Rg=11.630 SPS=15015
bl=0 pos[1]=[9.5 15.9 4.1] dr=1.22 t=0.0ps kin=1.43 pot=21.57 Rg=11.649 SPS=14963
bl=0 pos[1]=[8.4 16.8 6.1] dr=1.22 t=0.0ps kin=1.47 pot=21.58 Rg=11.638 SPS=15988
bl=0 pos[1]=[8.8 15.5 5.6] dr=1.23 t=0.0ps kin=1.48 pot=21.61 Rg=11.576 SPS=14920
bl=0 pos[1]=[9.1 14.9 6.1] dr=1.29 t=0.0ps kin=1.45 pot=21.58 Rg=11.494 SPS=14556
bl=0 pos[1]=[6.9 14.8 5.3] dr=1.27 t=0.0ps kin=1.48 pot=21.55 Rg=11.450 SPS=15414
bl=0 pos[1]=[6.0 14.1 4.1] dr=1.26 t=0.0ps kin=1.50 pot=21.60 Rg=11.388 SPS=14576
bl=0 pos[1]=[7.5 15.8 3.5] dr=1.25 t=0.0ps kin=1.46 pot=21.60 Rg=11.359 SPS=14722

```

(continues on next page)

(continued from previous page)

```

bl=0 pos[1]=[7.6 16.9 5.0] dr=1.22 t=0.0ps kin=1.45 pot=21.61 Rg=11.368 SPS=14909
bl=0 pos[1]=[8.2 17.5 4.4] dr=1.25 t=0.0ps kin=1.50 pot=21.57 Rg=11.354 SPS=14674
bl=0 pos[1]=[7.2 18.8 4.2] dr=1.28 t=0.0ps kin=1.50 pot=21.55 Rg=11.375 SPS=15420
bl=0 pos[1]=[7.5 19.5 2.7] dr=1.24 t=0.0ps kin=1.50 pot=21.58 Rg=11.387 SPS=15534
bl=0 pos[1]=[6.9 19.5 1.4] dr=1.25 t=0.0ps kin=1.47 pot=21.55 Rg=11.379 SPS=14480
bl=0 pos[1]=[5.2 19.6 1.1] dr=1.27 t=0.0ps kin=1.52 pot=21.57 Rg=11.374 SPS=15908
bl=0 pos[1]=[4.1 19.3 1.4] dr=1.29 t=0.0ps kin=1.53 pot=21.59 Rg=11.356 SPS=14851
bl=0 pos[1]=[3.2 19.6 0.6] dr=1.26 t=0.0ps kin=1.50 pot=21.61 Rg=11.374 SPS=15336
bl=0 pos[1]=[3.4 18.3 0.3] dr=1.30 t=0.0ps kin=1.52 pot=21.58 Rg=11.390 SPS=14647
bl=0 pos[1]=[2.7 17.1 0.6] dr=1.30 t=0.0ps kin=1.54 pot=21.63 Rg=11.451 SPS=14328
bl=0 pos[1]=[2.8 16.7 -0.6] dr=1.26 t=0.0ps kin=1.49 pot=21.63 Rg=11.474 SPS=14921
bl=0 pos[1]=[1.6 17.1 -0.7] dr=1.23 t=0.0ps kin=1.52 pot=21.63 Rg=11.430 SPS=14421
bl=0 pos[1]=[3.9 16.0 -1.8] dr=1.28 t=0.0ps kin=1.51 pot=21.61 Rg=11.425 SPS=15171
bl=0 pos[1]=[2.9 17.4 -1.0] dr=1.25 t=0.0ps kin=1.50 pot=21.62 Rg=11.427 SPS=15637
bl=0 pos[1]=[2.4 17.3 -1.5] dr=1.26 t=0.0ps kin=1.54 pot=21.57 Rg=11.437 SPS=14576
bl=0 pos[1]=[2.1 16.8 -0.6] dr=1.30 t=0.0ps kin=1.52 pot=21.60 Rg=11.467 SPS=15411
bl=0 pos[1]=[2.9 16.8 -1.2] dr=1.26 t=0.0ps kin=1.48 pot=21.60 Rg=11.527 SPS=14686
bl=0 pos[1]=[4.4 16.7 -0.8] dr=1.26 t=0.0ps kin=1.48 pot=21.60 Rg=11.520 SPS=15479
bl=0 pos[1]=[6.1 15.3 0.4] dr=1.26 t=0.0ps kin=1.46 pot=21.55 Rg=11.467 SPS=14861
bl=0 pos[1]=[5.7 15.8 -0.2] dr=1.24 t=0.0ps kin=1.56 pot=21.54 Rg=11.455 SPS=13809
bl=0 pos[1]=[6.5 15.2 -1.1] dr=1.27 t=0.0ps kin=1.49 pot=21.58 Rg=11.511 SPS=15363
bl=0 pos[1]=[6.9 14.0 -1.2] dr=1.22 t=0.0ps kin=1.49 pot=21.61 Rg=11.543 SPS=14826
bl=0 pos[1]=[8.5 11.7 -2.7] dr=1.25 t=0.0ps kin=1.53 pot=21.62 Rg=11.552 SPS=15360
bl=0 pos[1]=[8.3 10.4 -2.8] dr=1.27 t=0.0ps kin=1.54 pot=21.56 Rg=11.548 SPS=15171
bl=0 pos[1]=[7.9 10.4 -4.1] dr=1.26 t=0.0ps kin=1.52 pot=21.58 Rg=11.512 SPS=14282
bl=0 pos[1]=[9.7 10.1 -4.3] dr=1.26 t=0.0ps kin=1.55 pot=21.60 Rg=11.494 SPS=15022
bl=0 pos[1]=[9.0 11.2 -5.0] dr=1.25 t=0.0ps kin=1.50 pot=21.61 Rg=11.499 SPS=14180
bl=0 pos[1]=[9.4 12.3 -5.1] dr=1.24 t=0.0ps kin=1.49 pot=21.58 Rg=11.561 SPS=14573
bl=0 pos[1]=[9.0 14.0 -4.2] dr=1.24 t=0.0ps kin=1.51 pot=21.61 Rg=11.578 SPS=14898
bl=0 pos[1]=[10.6 13.2 -2.2] dr=1.27 t=0.0ps kin=1.52 pot=21.64 Rg=11.520 SPS=14854
bl=0 pos[1]=[7.4 12.1 -0.8] dr=1.23 t=0.0ps kin=1.47 pot=21.61 Rg=11.465 SPS=15288
bl=0 pos[1]=[5.5 10.9 -1.2] dr=1.27 t=0.0ps kin=1.54 pot=21.58 Rg=11.408 SPS=14207
bl=0 pos[1]=[6.3 12.0 -1.1] dr=1.26 t=0.0ps kin=1.48 pot=21.61 Rg=11.358 SPS=13869
bl=0 pos[1]=[5.6 11.3 -1.3] dr=1.27 t=0.0ps kin=1.51 pot=21.60 Rg=11.285 SPS=14367
bl=0 pos[1]=[4.9 10.9 -0.8] dr=1.25 t=0.0ps kin=1.46 pot=21.62 Rg=11.221 SPS=15048
bl=0 pos[1]=[5.2 12.7 0.2] dr=1.27 t=0.0ps kin=1.58 pot=21.57 Rg=11.192 SPS=15402
bl=0 pos[1]=[4.9 14.6 -0.0] dr=1.26 t=0.0ps kin=1.55 pot=21.58 Rg=11.166 SPS=14803
bl=0 pos[1]=[3.5 16.9 -0.7] dr=1.30 t=0.0ps kin=1.52 pot=21.61 Rg=11.177 SPS=15296
bl=0 pos[1]=[2.2 17.7 -1.0] dr=1.27 t=0.0ps kin=1.54 pot=21.62 Rg=11.206 SPS=14397
bl=0 pos[1]=[2.2 18.0 -0.8] dr=1.23 t=0.0ps kin=1.53 pot=21.57 Rg=11.229 SPS=14957
bl=0 pos[1]=[1.8 17.2 -1.6] dr=1.23 t=0.0ps kin=1.50 pot=21.60 Rg=11.248 SPS=15029
bl=0 pos[1]=[2.2 16.2 -1.8] dr=1.23 t=0.0ps kin=1.47 pot=21.61 Rg=11.246 SPS=14561
bl=0 pos[1]=[1.9 15.1 -3.0] dr=1.24 t=0.0ps kin=1.47 pot=21.58 Rg=11.278 SPS=14217
bl=0 pos[1]=[2.1 16.6 -3.9] dr=1.23 t=0.0ps kin=1.52 pot=21.57 Rg=11.280 SPS=14562
bl=0 pos[1]=[4.4 17.7 -4.0] dr=1.29 t=0.0ps kin=1.54 pot=21.57 Rg=11.298 SPS=15042
bl=0 pos[1]=[3.8 16.1 -2.8] dr=1.29 t=0.0ps kin=1.48 pot=21.59 Rg=11.335 SPS=15223
bl=0 pos[1]=[4.2 16.1 -3.2] dr=1.24 t=0.0ps kin=1.45 pot=21.54 Rg=11.359 SPS=14188
bl=0 pos[1]=[3.1 14.6 -2.2] dr=1.29 t=0.0ps kin=1.52 pot=21.53 Rg=11.327 SPS=14593
bl=0 pos[1]=[1.1 14.7 -2.6] dr=1.26 t=0.0ps kin=1.52 pot=21.58 Rg=11.278 SPS=13957
bl=0 pos[1]=[-0.1 15.0 -4.5] dr=1.28 t=0.0ps kin=1.57 pot=21.53 Rg=11.318 SPS=14739
bl=0 pos[1]=[1.2 13.7 -6.3] dr=1.29 t=0.0ps kin=1.52 pot=21.57 Rg=11.369 SPS=14175
bl=0 pos[1]=[4.1 14.1 -6.2] dr=1.26 t=0.0ps kin=1.51 pot=21.55 Rg=11.403 SPS=15117

```

(continues on next page)

(continued from previous page)

```

bl=0 pos[1]=[5.4 12.8 -6.0] dr=1.24 t=0.0ps kin=1.45 pot=21.57 Rg=11.451 SPS=14449
bl=0 pos[1]=[6.5 12.4 -7.1] dr=1.26 t=0.0ps kin=1.50 pot=21.51 Rg=11.451 SPS=14637
bl=0 pos[1]=[5.2 14.5 -7.4] dr=1.30 t=0.0ps kin=1.48 pot=21.57 Rg=11.448 SPS=15339
bl=0 pos[1]=[4.1 14.8 -6.0] dr=1.29 t=0.0ps kin=1.50 pot=21.54 Rg=11.420 SPS=14371
bl=0 pos[1]=[4.9 14.2 -5.1] dr=1.28 t=0.0ps kin=1.52 pot=21.58 Rg=11.393 SPS=14675
bl=0 pos[1]=[4.7 13.0 -5.1] dr=1.27 t=0.0ps kin=1.48 pot=21.62 Rg=11.342 SPS=13986
bl=0 pos[1]=[3.8 13.2 -6.4] dr=1.27 t=0.0ps kin=1.49 pot=21.56 Rg=11.343 SPS=14722
bl=0 pos[1]=[4.1 14.0 -7.0] dr=1.26 t=0.0ps kin=1.52 pot=21.59 Rg=11.372 SPS=13439
bl=0 pos[1]=[3.7 12.9 -8.8] dr=1.24 t=0.0ps kin=1.46 pot=21.56 Rg=11.389 SPS=14546
bl=0 pos[1]=[4.1 12.0 -9.3] dr=1.23 t=0.0ps kin=1.48 pot=21.58 Rg=11.372 SPS=13843
bl=0 pos[1]=[4.6 12.2 -8.9] dr=1.25 t=0.0ps kin=1.46 pot=21.62 Rg=11.355 SPS=15084
bl=0 pos[1]=[4.0 13.2 -7.9] dr=1.23 t=0.0ps kin=1.53 pot=21.59 Rg=11.432 SPS=15213
bl=0 pos[1]=[2.7 11.9 -7.9] dr=1.26 t=0.0ps kin=1.54 pot=21.55 Rg=11.481 SPS=14578
bl=0 pos[1]=[0.7 9.9 -6.6] dr=1.25 t=0.0ps kin=1.45 pot=21.55 Rg=11.528 SPS=15128
bl=0 pos[1]=[0.4 11.7 -9.3] dr=1.22 t=0.0ps kin=1.43 pot=21.56 Rg=11.512 SPS=14552
bl=0 pos[1]=[1.0 12.7 -10.3] dr=1.22 t=0.0ps kin=1.45 pot=21.58 Rg=11.487 SPS=15391
bl=0 pos[1]=[3.0 13.3 -9.1] dr=1.23 t=0.0ps kin=1.47 pot=21.63 Rg=11.491 SPS=15838
bl=0 pos[1]=[4.4 14.5 -7.6] dr=1.25 t=0.0ps kin=1.51 pot=21.59 Rg=11.486 SPS=14833
bl=0 pos[1]=[3.8 13.0 -8.4] dr=1.27 t=0.0ps kin=1.50 pot=21.57 Rg=11.471 SPS=14912
bl=0 pos[1]=[4.1 11.2 -10.1] dr=1.26 t=0.0ps kin=1.48 pot=21.63 Rg=11.423 SPS=13591
bl=0 pos[1]=[4.0 10.7 -11.3] dr=1.24 t=0.0ps kin=1.47 pot=21.60 Rg=11.428 SPS=14599
bl=0 pos[1]=[2.7 10.2 -11.8] dr=1.25 t=0.0ps kin=1.49 pot=21.59 Rg=11.416 SPS=15032
bl=0 pos[1]=[0.7 9.5 -11.9] dr=1.27 t=0.0ps kin=1.54 pot=21.64 Rg=11.419 SPS=14903
bl=0 pos[1]=[0.7 9.0 -10.7] dr=1.22 t=0.0ps kin=1.48 pot=21.63 Rg=11.432 SPS=15718
bl=0 pos[1]=[2.2 8.4 -10.2] dr=1.21 t=0.0ps kin=1.53 pot=21.61 Rg=11.457 SPS=15631
bl=0 pos[1]=[1.8 8.6 -8.8] dr=1.22 t=0.0ps kin=1.49 pot=21.59 Rg=11.491 SPS=15106
bl=0 pos[1]=[1.1 10.6 -8.8] dr=1.28 t=0.0ps kin=1.51 pot=21.57 Rg=11.516 SPS=15416
bl=0 pos[1]=[2.0 9.9 -8.2] dr=1.29 t=0.0ps kin=1.51 pot=21.59 Rg=11.515 SPS=14927
bl=0 pos[1]=[2.5 10.1 -8.0] dr=1.29 t=0.0ps kin=1.52 pot=21.57 Rg=11.513 SPS=15332
bl=0 pos[1]=[4.4 10.6 -6.9] dr=1.27 t=0.0ps kin=1.51 pot=21.59 Rg=11.522 SPS=15445
bl=0 pos[1]=[6.9 12.0 -7.3] dr=1.27 t=0.0ps kin=1.48 pot=21.61 Rg=11.553 SPS=14794
bl=0 pos[1]=[6.1 12.5 -7.3] dr=1.21 t=0.0ps kin=1.48 pot=21.63 Rg=11.570 SPS=15371
bl=0 pos[1]=[5.9 11.9 -5.9] dr=1.24 t=0.0ps kin=1.48 pot=21.59 Rg=11.572 SPS=15672
bl=0 pos[1]=[5.7 14.1 -4.6] dr=1.22 t=0.0ps kin=1.51 pot=21.59 Rg=11.555 SPS=14351
bl=0 pos[1]=[5.0 13.0 -5.4] dr=1.28 t=0.0ps kin=1.53 pot=21.59 Rg=11.530 SPS=14981
bl=0 pos[1]=[5.6 10.7 -5.3] dr=1.29 t=0.0ps kin=1.54 pot=21.57 Rg=11.488 SPS=11741
bl=0 pos[1]=[6.0 9.3 -3.7] dr=1.26 t=0.0ps kin=1.51 pot=21.60 Rg=11.435 SPS=15038
bl=0 pos[1]=[5.1 10.3 -3.5] dr=1.27 t=0.0ps kin=1.60 pot=21.61 Rg=11.431 SPS=12183
bl=0 pos[1]=[4.1 11.3 -3.9] dr=1.28 t=0.0ps kin=1.54 pot=21.63 Rg=11.376 SPS=13150
bl=0 pos[1]=[3.6 11.9 -3.7] dr=1.29 t=0.0ps kin=1.56 pot=21.58 Rg=11.259 SPS=14974
bl=0 pos[1]=[3.8 13.1 -3.9] dr=1.28 t=0.0ps kin=1.53 pot=21.58 Rg=11.116 SPS=15216
bl=0 pos[1]=[4.2 13.0 -2.3] dr=1.29 t=0.0ps kin=1.51 pot=21.63 Rg=11.020 SPS=13723
bl=0 pos[1]=[4.6 14.1 -2.5] dr=1.26 t=0.0ps kin=1.56 pot=21.57 Rg=11.033 SPS=14242
bl=0 pos[1]=[4.0 12.8 -3.6] dr=1.26 t=0.0ps kin=1.51 pot=21.58 Rg=11.030 SPS=14244
bl=0 pos[1]=[4.5 10.3 -2.5] dr=1.27 t=0.0ps kin=1.48 pot=21.61 Rg=11.017 SPS=14515
bl=0 pos[1]=[4.4 10.5 -2.6] dr=1.26 t=0.0ps kin=1.50 pot=21.55 Rg=11.017 SPS=13881
bl=0 pos[1]=[4.5 10.4 -3.4] dr=1.25 t=0.0ps kin=1.49 pot=21.56 Rg=11.006 SPS=14760
bl=0 pos[1]=[5.1 11.8 -2.7] dr=1.25 t=0.0ps kin=1.49 pot=21.54 Rg=11.022 SPS=13977
bl=0 pos[1]=[4.8 9.6 -0.7] dr=1.26 t=0.0ps kin=1.50 pot=21.57 Rg=11.030 SPS=14837
bl=0 pos[1]=[5.5 8.9 -1.1] dr=1.26 t=0.0ps kin=1.47 pot=21.58 Rg=11.040 SPS=15141
bl=0 pos[1]=[6.1 9.1 -1.4] dr=1.24 t=0.0ps kin=1.52 pot=21.59 Rg=11.071 SPS=15265
bl=0 pos[1]=[6.1 10.7 -0.7] dr=1.22 t=0.0ps kin=1.49 pot=21.60 Rg=11.084 SPS=15123

```

(continues on next page)

(continued from previous page)

```

bl=0 pos[1]=[6.4 10.7 -0.7] dr=1.22 t=0.0ps kin=1.44 pot=21.60 Rg=11.095 SPS=13893
bl=0 pos[1]=[6.6 10.5 -2.1] dr=1.25 t=0.0ps kin=1.45 pot=21.55 Rg=11.096 SPS=15398
bl=0 pos[1]=[6.3 10.2 -2.1] dr=1.22 t=0.0ps kin=1.48 pot=21.54 Rg=11.052 SPS=13781
bl=0 pos[1]=[6.7 10.0 -4.4] dr=1.23 t=0.0ps kin=1.47 pot=21.54 Rg=11.002 SPS=14235
bl=0 pos[1]=[7.8 12.6 -6.7] dr=1.27 t=0.0ps kin=1.44 pot=21.60 Rg=10.955 SPS=14924
bl=0 pos[1]=[5.8 13.1 -8.5] dr=1.29 t=0.0ps kin=1.52 pot=21.54 Rg=10.917 SPS=13963
bl=0 pos[1]=[5.4 13.7 -10.1] dr=1.29 t=0.0ps kin=1.52 pot=21.61 Rg=10.894 SPS=14865
bl=0 pos[1]=[6.0 14.3 -10.6] dr=1.29 t=0.0ps kin=1.50 pot=21.57 Rg=10.919 SPS=13846
bl=0 pos[1]=[5.9 14.7 -9.9] dr=1.25 t=0.0ps kin=1.48 pot=21.56 Rg=10.967 SPS=14738
bl=0 pos[1]=[6.2 13.8 -7.8] dr=1.24 t=0.0ps kin=1.41 pot=21.61 Rg=11.023 SPS=13638
bl=0 pos[1]=[7.0 12.0 -6.9] dr=1.26 t=0.0ps kin=1.44 pot=21.56 Rg=11.072 SPS=14812
bl=0 pos[1]=[7.1 12.3 -5.7] dr=1.25 t=0.0ps kin=1.43 pot=21.60 Rg=11.114 SPS=14703
bl=0 pos[1]=[7.0 11.3 -5.6] dr=1.27 t=0.0ps kin=1.53 pot=21.59 Rg=11.139 SPS=14840
bl=0 pos[1]=[7.4 9.8 -4.6] dr=1.26 t=0.0ps kin=1.47 pot=21.58 Rg=11.152 SPS=15275
bl=0 pos[1]=[7.5 9.9 -4.6] dr=1.25 t=0.0ps kin=1.52 pot=21.56 Rg=11.200 SPS=13919
bl=0 pos[1]=[7.9 10.2 -4.9] dr=1.24 t=0.0ps kin=1.44 pot=21.60 Rg=11.313 SPS=15044
bl=0 pos[1]=[8.0 10.5 -4.8] dr=1.26 t=0.0ps kin=1.51 pot=21.58 Rg=11.413 SPS=13203
bl=0 pos[1]=[8.0 12.0 -5.3] dr=1.29 t=0.0ps kin=1.51 pot=21.61 Rg=11.497 SPS=15229
bl=0 pos[1]=[7.2 14.6 -5.3] dr=1.26 t=0.0ps kin=1.54 pot=21.56 Rg=11.532 SPS=14474
bl=0 pos[1]=[6.0 17.7 -6.2] dr=1.30 t=0.0ps kin=1.48 pot=21.60 Rg=11.544 SPS=15012
bl=0 pos[1]=[4.0 18.3 -5.8] dr=1.28 t=0.0ps kin=1.48 pot=21.57 Rg=11.538 SPS=15584
bl=0 pos[1]=[2.3 18.0 -5.8] dr=1.27 t=0.0ps kin=1.49 pot=21.54 Rg=11.532 SPS=14679
bl=0 pos[1]=[2.6 16.8 -4.1] dr=1.25 t=0.0ps kin=1.47 pot=21.57 Rg=11.500 SPS=15210
bl=0 pos[1]=[4.0 18.6 -3.1] dr=1.24 t=0.0ps kin=1.51 pot=21.52 Rg=11.475 SPS=14006
bl=0 pos[1]=[4.8 20.2 -0.8] dr=1.28 t=0.0ps kin=1.52 pot=21.55 Rg=11.460 SPS=15301
bl=0 pos[1]=[4.2 19.7 -0.2] dr=1.25 t=0.0ps kin=1.49 pot=21.55 Rg=11.446 SPS=15187
bl=0 pos[1]=[3.9 19.0 0.1] dr=1.24 t=0.0ps kin=1.49 pot=21.60 Rg=11.412 SPS=14216
bl=0 pos[1]=[3.9 16.6 1.6] dr=1.25 t=0.0ps kin=1.52 pot=21.59 Rg=11.403 SPS=14868
bl=0 pos[1]=[2.4 16.5 1.7] dr=1.24 t=0.0ps kin=1.54 pot=21.58 Rg=11.404 SPS=14819
bl=0 pos[1]=[1.1 16.2 1.8] dr=1.26 t=0.0ps kin=1.49 pot=21.59 Rg=11.413 SPS=15821
bl=0 pos[1]=[0.8 16.2 0.3] dr=1.20 t=0.0ps kin=1.46 pot=21.63 Rg=11.405 SPS=15954
bl=0 pos[1]=[1.8 15.3 -0.2] dr=1.24 t=0.0ps kin=1.48 pot=21.61 Rg=11.387 SPS=14595
bl=0 pos[1]=[2.8 15.4 1.1] dr=1.22 t=0.0ps kin=1.42 pot=21.59 Rg=11.380 SPS=14904
bl=0 pos[1]=[5.2 14.5 0.8] dr=1.22 t=0.0ps kin=1.43 pot=21.56 Rg=11.348 SPS=14873
bl=0 pos[1]=[6.4 15.2 0.6] dr=1.25 t=0.0ps kin=1.46 pot=21.56 Rg=11.305 SPS=15001
bl=0 pos[1]=[5.9 15.4 1.5] dr=1.27 t=0.0ps kin=1.45 pot=21.61 Rg=11.284 SPS=14970
bl=0 pos[1]=[5.4 15.1 1.9] dr=1.25 t=0.0ps kin=1.47 pot=21.60 Rg=11.226 SPS=14288
bl=0 pos[1]=[4.1 14.8 3.0] dr=1.23 t=0.0ps kin=1.47 pot=21.60 Rg=11.239 SPS=14838
bl=0 pos[1]=[3.4 14.4 3.6] dr=1.22 t=0.0ps kin=1.52 pot=21.54 Rg=11.308 SPS=15626
bl=0 pos[1]=[2.8 14.5 2.6] dr=1.22 t=0.0ps kin=1.49 pot=21.56 Rg=11.373 SPS=13935
bl=0 pos[1]=[3.9 16.7 2.9] dr=1.23 t=0.0ps kin=1.51 pot=21.57 Rg=11.393 SPS=14753
bl=0 pos[1]=[4.5 17.1 1.1] dr=1.22 t=0.0ps kin=1.49 pot=21.61 Rg=11.420 SPS=13444
bl=0 pos[1]=[3.6 16.2 1.7] dr=1.27 t=0.0ps kin=1.49 pot=21.59 Rg=11.492 SPS=15691
bl=0 pos[1]=[3.9 16.6 1.6] dr=1.27 t=0.0ps kin=1.54 pot=21.55 Rg=11.569 SPS=14227
bl=0 pos[1]=[4.9 16.1 2.8] dr=1.24 t=0.0ps kin=1.45 pot=21.60 Rg=11.610 SPS=14708
bl=0 pos[1]=[4.2 17.8 2.2] dr=1.26 t=0.0ps kin=1.45 pot=21.63 Rg=11.638 SPS=14888
bl=0 pos[1]=[4.2 18.6 2.4] dr=1.25 t=0.0ps kin=1.50 pot=21.56 Rg=11.664 SPS=13730
bl=0 pos[1]=[2.7 18.2 2.4] dr=1.23 t=0.0ps kin=1.44 pot=21.62 Rg=11.688 SPS=14603
bl=0 pos[1]=[0.8 18.2 2.3] dr=1.25 t=0.0ps kin=1.49 pot=21.57 Rg=11.689 SPS=14354
bl=0 pos[1]=[0.6 17.3 2.3] dr=1.27 t=0.0ps kin=1.49 pot=21.59 Rg=11.690 SPS=15746
bl=0 pos[1]=[-0.7 15.7 2.4] dr=1.27 t=0.0ps kin=1.49 pot=21.59 Rg=11.687 SPS=14272
bl=0 pos[1]=[-0.4 15.2 1.7] dr=1.25 t=0.0ps kin=1.55 pot=21.60 Rg=11.682 SPS=15147

```

(continues on next page)

(continued from previous page)

```

bl=0 pos[1]=[0.8 15.5 1.2] dr=1.25 t=0.0ps kin=1.50 pot=21.63 Rg=11.688 SPS=15091
bl=0 pos[1]=[1.6 14.4 0.9] dr=1.23 t=0.0ps kin=1.45 pot=21.59 Rg=11.709 SPS=14505
bl=0 pos[1]=[0.4 16.6 0.2] dr=1.24 t=0.0ps kin=1.51 pot=21.57 Rg=11.750 SPS=14911
bl=0 pos[1]=[0.2 17.9 -0.7] dr=1.25 t=0.0ps kin=1.48 pot=21.58 Rg=11.736 SPS=13680
bl=0 pos[1]=[-1.1 17.6 -0.7] dr=1.25 t=0.0ps kin=1.51 pot=21.58 Rg=11.694 SPS=14813
bl=0 pos[1]=[1.7 16.4 0.2] dr=1.27 t=0.0ps kin=1.55 pot=21.57 Rg=11.654 SPS=14507
bl=0 pos[1]=[2.7 14.1 1.7] dr=1.30 t=0.0ps kin=1.48 pot=21.60 Rg=11.636 SPS=15112
bl=0 pos[1]=[3.2 13.7 1.6] dr=1.27 t=0.0ps kin=1.52 pot=21.60 Rg=11.583 SPS=15066
bl=0 pos[1]=[3.4 14.1 3.0] dr=1.29 t=0.0ps kin=1.52 pot=21.58 Rg=11.501 SPS=14712
bl=0 pos[1]=[2.8 14.1 2.5] dr=1.27 t=0.0ps kin=1.49 pot=21.60 Rg=11.393 SPS=14858
bl=0 pos[1]=[3.4 13.8 2.7] dr=1.28 t=0.0ps kin=1.46 pot=21.62 Rg=11.295 SPS=14184
bl=0 pos[1]=[4.5 13.4 1.8] dr=1.28 t=0.0ps kin=1.49 pot=21.61 Rg=11.212 SPS=15126
bl=0 pos[1]=[4.7 12.4 2.5] dr=1.29 t=0.0ps kin=1.54 pot=21.63 Rg=11.160 SPS=14904
bl=0 pos[1]=[5.3 12.4 4.7] dr=1.29 t=0.0ps kin=1.57 pot=21.61 Rg=11.158 SPS=14567
bl=0 pos[1]=[5.0 12.2 5.1] dr=1.27 t=0.0ps kin=1.51 pot=21.59 Rg=11.132 SPS=14287
bl=0 pos[1]=[3.5 12.4 4.0] dr=1.25 t=0.0ps kin=1.50 pot=21.61 Rg=11.094 SPS=14565
bl=0 pos[1]=[4.1 11.2 4.1] dr=1.27 t=0.0ps kin=1.52 pot=21.58 Rg=11.104 SPS=14916
bl=0 pos[1]=[5.1 11.0 4.2] dr=1.24 t=0.0ps kin=1.49 pot=21.60 Rg=11.150 SPS=13832
bl=0 pos[1]=[4.8 12.2 3.2] dr=1.25 t=0.0ps kin=1.56 pot=21.58 Rg=11.132 SPS=15024
bl=0 pos[1]=[4.7 13.0 3.5] dr=1.25 t=0.0ps kin=1.51 pot=21.63 Rg=11.079 SPS=14950
bl=0 pos[1]=[4.5 12.7 3.0] dr=1.25 t=0.0ps kin=1.52 pot=21.56 Rg=11.079 SPS=15128
bl=0 pos[1]=[4.3 12.4 1.8] dr=1.25 t=0.0ps kin=1.54 pot=21.58 Rg=11.084 SPS=14920
bl=0 pos[1]=[4.4 12.1 2.4] dr=1.25 t=0.0ps kin=1.51 pot=21.59 Rg=11.029 SPS=14472
bl=0 pos[1]=[5.1 11.9 3.3] dr=1.23 t=0.0ps kin=1.47 pot=21.61 Rg=11.022 SPS=15143
bl=0 pos[1]=[5.4 10.7 2.8] dr=1.23 t=0.0ps kin=1.51 pot=21.56 Rg=11.074 SPS=12343
bl=0 pos[1]=[5.8 11.1 1.6] dr=1.25 t=0.0ps kin=1.49 pot=21.56 Rg=11.119 SPS=14593
bl=0 pos[1]=[6.7 10.7 1.3] dr=1.26 t=0.0ps kin=1.54 pot=21.57 Rg=11.147 SPS=14153
bl=0 pos[1]=[5.8 11.1 1.3] dr=1.28 t=0.0ps kin=1.48 pot=21.58 Rg=11.147 SPS=14381
bl=0 pos[1]=[6.0 12.0 0.3] dr=1.27 t=0.0ps kin=1.50 pot=21.55 Rg=11.190 SPS=15121
bl=0 pos[1]=[6.8 13.3 -0.6] dr=1.25 t=0.0ps kin=1.50 pot=21.54 Rg=11.264 SPS=14265
bl=0 pos[1]=[6.6 15.5 -1.2] dr=1.30 t=0.0ps kin=1.48 pot=21.59 Rg=11.285 SPS=14862
bl=0 pos[1]=[5.2 16.3 -1.1] dr=1.28 t=0.0ps kin=1.46 pot=21.58 Rg=11.314 SPS=14936
bl=0 pos[1]=[4.4 16.6 -0.2] dr=1.27 t=0.0ps kin=1.48 pot=21.61 Rg=11.346 SPS=15462
bl=0 pos[1]=[3.6 15.7 -1.3] dr=1.25 t=0.0ps kin=1.55 pot=21.59 Rg=11.394 SPS=14998
bl=0 pos[1]=[4.4 14.2 -3.1] dr=1.24 t=0.0ps kin=1.53 pot=21.61 Rg=11.385 SPS=14996
bl=0 pos[1]=[4.1 14.5 -3.8] dr=1.25 t=0.0ps kin=1.52 pot=21.63 Rg=11.348 SPS=14541
bl=0 pos[1]=[5.3 15.4 -3.1] dr=1.24 t=0.0ps kin=1.49 pot=21.58 Rg=11.309 SPS=13760
bl=0 pos[1]=[3.4 16.3 -3.2] dr=1.23 t=0.0ps kin=1.46 pot=21.59 Rg=11.273 SPS=14258
bl=0 pos[1]=[4.1 15.2 -4.6] dr=1.23 t=0.0ps kin=1.49 pot=21.58 Rg=11.240 SPS=13516
bl=0 pos[1]=[3.7 12.6 -4.3] dr=1.21 t=0.0ps kin=1.46 pot=21.56 Rg=11.167 SPS=14826
bl=0 pos[1]=[1.8 12.1 -3.3] dr=1.24 t=0.0ps kin=1.49 pot=21.55 Rg=11.140 SPS=13501
bl=0 pos[1]=[1.6 13.5 -3.2] dr=1.24 t=0.0ps kin=1.50 pot=21.54 Rg=11.133 SPS=14859
bl=0 pos[1]=[2.0 14.5 -2.4] dr=1.24 t=0.0ps kin=1.50 pot=21.58 Rg=11.132 SPS=14580
bl=0 pos[1]=[2.4 15.4 -2.2] dr=1.24 t=0.0ps kin=1.56 pot=21.58 Rg=11.124 SPS=15186
bl=0 pos[1]=[1.0 14.7 -2.6] dr=1.26 t=0.0ps kin=1.49 pot=21.61 Rg=11.122 SPS=14980
bl=0 pos[1]=[1.4 14.8 -5.0] dr=1.28 t=0.0ps kin=1.45 pot=21.58 Rg=11.073 SPS=14327
bl=0 pos[1]=[1.2 16.0 -5.2] dr=1.23 t=0.0ps kin=1.45 pot=21.58 Rg=11.072 SPS=15121
bl=0 pos[1]=[2.4 16.5 -5.4] dr=1.27 t=0.0ps kin=1.51 pot=21.53 Rg=11.111 SPS=14535
bl=0 pos[1]=[4.1 15.5 -6.5] dr=1.25 t=0.0ps kin=1.47 pot=21.57 Rg=11.136 SPS=14729
bl=0 pos[1]=[5.2 13.6 -6.5] dr=1.24 t=0.0ps kin=1.48 pot=21.57 Rg=11.180 SPS=14774
bl=0 pos[1]=[5.1 13.5 -3.6] dr=1.25 t=0.0ps kin=1.49 pot=21.58 Rg=11.204 SPS=14328
bl=0 pos[1]=[4.6 13.9 -3.2] dr=1.26 t=0.0ps kin=1.53 pot=21.60 Rg=11.242 SPS=14581

```

(continues on next page)

(continued from previous page)

```

bl=0 pos[1]=[4.6 13.9 -4.8] dr=1.28 t=0.0ps kin=1.57 pot=21.64 Rg=11.317 SPS=14719
bl=0 pos[1]=[5.7 13.9 -6.5] dr=1.25 t=0.0ps kin=1.50 pot=21.61 Rg=11.424 SPS=15486
bl=0 pos[1]=[4.0 13.2 -5.1] dr=1.26 t=0.0ps kin=1.49 pot=21.63 Rg=11.441 SPS=15547
bl=0 pos[1]=[3.5 14.1 -4.4] dr=1.27 t=0.0ps kin=1.51 pot=21.61 Rg=11.467 SPS=14295
bl=0 pos[1]=[2.5 15.2 -5.5] dr=1.24 t=0.0ps kin=1.49 pot=21.62 Rg=11.504 SPS=15586
bl=0 pos[1]=[3.5 15.9 -5.6] dr=1.25 t=0.0ps kin=1.54 pot=21.58 Rg=11.545 SPS=14878
bl=0 pos[1]=[4.0 16.3 -5.0] dr=1.29 t=0.0ps kin=1.53 pot=21.57 Rg=11.630 SPS=12155
bl=0 pos[1]=[4.3 14.7 -5.9] dr=1.31 t=0.0ps kin=1.53 pot=21.58 Rg=11.642 SPS=14441
bl=0 pos[1]=[4.5 13.4 -6.3] dr=1.26 t=0.0ps kin=1.53 pot=21.58 Rg=11.608 SPS=15325
bl=0 pos[1]=[5.0 12.1 -8.3] dr=1.31 t=0.0ps kin=1.56 pot=21.59 Rg=11.611 SPS=15282
bl=0 pos[1]=[3.4 13.4 -10.5] dr=1.27 t=0.0ps kin=1.54 pot=21.59 Rg=11.626 SPS=14968
bl=0 pos[1]=[4.7 14.7 -10.7] dr=1.24 t=0.0ps kin=1.42 pot=21.61 Rg=11.628 SPS=15159
bl=0 pos[1]=[6.6 14.4 -10.9] dr=1.19 t=0.0ps kin=1.46 pot=21.58 Rg=11.609 SPS=13759
bl=0 pos[1]=[6.9 12.5 -9.9] dr=1.23 t=0.0ps kin=1.47 pot=21.55 Rg=11.616 SPS=15099
bl=0 pos[1]=[7.6 11.8 -10.4] dr=1.27 t=0.0ps kin=1.46 pot=21.61 Rg=11.640 SPS=16083
bl=0 pos[1]=[8.6 9.9 -11.4] dr=1.28 t=0.0ps kin=1.54 pot=21.56 Rg=11.650 SPS=14773
bl=0 pos[1]=[9.5 8.6 -10.5] dr=1.27 t=0.0ps kin=1.52 pot=21.66 Rg=11.604 SPS=15338
bl=0 pos[1]=[8.7 8.5 -8.9] dr=1.27 t=0.0ps kin=1.55 pot=21.65 Rg=11.590 SPS=14406
bl=0 pos[1]=[11.4 8.8 -8.4] dr=1.27 t=0.0ps kin=1.53 pot=21.63 Rg=11.566 SPS=15060
bl=0 pos[1]=[12.2 9.2 -10.3] dr=1.27 t=0.0ps kin=1.53 pot=21.63 Rg=11.558 SPS=14607
bl=0 pos[1]=[12.1 9.5 -12.0] dr=1.27 t=0.0ps kin=1.52 pot=21.68 Rg=11.550 SPS=14066
bl=0 pos[1]=[12.5 8.4 -11.7] dr=1.27 t=0.0ps kin=1.50 pot=21.65 Rg=11.574 SPS=15142
bl=0 pos[1]=[11.5 8.5 -10.8] dr=1.22 t=0.0ps kin=1.50 pot=21.63 Rg=11.593 SPS=14627
bl=0 pos[1]=[10.6 8.9 -8.6] dr=1.26 t=0.0ps kin=1.53 pot=21.60 Rg=11.580 SPS=14580
bl=0 pos[1]=[11.0 8.8 -6.8] dr=1.26 t=0.0ps kin=1.50 pot=21.62 Rg=11.565 SPS=14483
bl=0 pos[1]=[11.0 7.9 -6.7] dr=1.27 t=0.0ps kin=1.48 pot=21.64 Rg=11.547 SPS=15337
bl=0 pos[1]=[10.8 7.2 -7.0] dr=1.24 t=0.0ps kin=1.50 pot=21.64 Rg=11.501 SPS=14749
bl=0 pos[1]=[10.7 6.6 -7.6] dr=1.28 t=0.0ps kin=1.50 pot=21.59 Rg=11.430 SPS=14549
bl=0 pos[1]=[8.7 5.6 -9.2] dr=1.27 t=0.0ps kin=1.50 pot=21.60 Rg=11.393 SPS=14934
bl=0 pos[1]=[8.5 5.7 -9.5] dr=1.29 t=0.0ps kin=1.49 pot=21.62 Rg=11.392 SPS=14225
bl=0 pos[1]=[9.7 5.2 -8.9] dr=1.26 t=0.0ps kin=1.46 pot=21.60 Rg=11.398 SPS=15503
bl=0 pos[1]=[7.9 4.5 -8.9] dr=1.27 t=0.0ps kin=1.49 pot=21.61 Rg=11.396 SPS=14900
bl=0 pos[1]=[6.0 5.5 -9.7] dr=1.25 t=0.0ps kin=1.53 pot=21.59 Rg=11.412 SPS=14630
bl=0 pos[1]=[4.8 5.6 -10.7] dr=1.26 t=0.0ps kin=1.50 pot=21.61 Rg=11.418 SPS=15524
bl=0 pos[1]=[5.4 5.0 -12.0] dr=1.27 t=0.0ps kin=1.51 pot=21.59 Rg=11.426 SPS=11907
bl=0 pos[1]=[6.8 4.3 -13.2] dr=1.28 t=0.0ps kin=1.51 pot=21.57 Rg=11.415 SPS=15206
bl=0 pos[1]=[7.1 3.8 -13.4] dr=1.29 t=0.0ps kin=1.50 pot=21.56 Rg=11.388 SPS=13823
bl=0 pos[1]=[7.5 4.5 -11.3] dr=1.25 t=0.0ps kin=1.50 pot=21.55 Rg=11.322 SPS=14900
bl=0 pos[1]=[6.7 5.7 -11.7] dr=1.26 t=0.0ps kin=1.50 pot=21.59 Rg=11.285 SPS=14703
bl=0 pos[1]=[7.2 4.2 -12.7] dr=1.27 t=0.0ps kin=1.54 pot=21.61 Rg=11.256 SPS=14417
bl=0 pos[1]=[7.6 6.0 -13.8] dr=1.27 t=0.0ps kin=1.56 pot=21.58 Rg=11.213 SPS=15203
bl=0 pos[1]=[7.0 6.6 -14.1] dr=1.26 t=0.0ps kin=1.52 pot=21.57 Rg=11.157 SPS=14053
bl=0 pos[1]=[8.3 7.4 -11.9] dr=1.29 t=0.0ps kin=1.54 pot=21.55 Rg=11.122 SPS=14993
bl=0 pos[1]=[8.5 7.1 -11.4] dr=1.28 t=0.0ps kin=1.48 pot=21.57 Rg=11.126 SPS=14203
bl=0 pos[1]=[7.8 7.1 -11.6] dr=1.28 t=0.0ps kin=1.56 pot=21.58 Rg=11.173 SPS=14913
bl=0 pos[1]=[9.5 7.1 -11.9] dr=1.29 t=0.0ps kin=1.52 pot=21.58 Rg=11.224 SPS=14177
bl=0 pos[1]=[9.6 8.1 -11.1] dr=1.23 t=0.0ps kin=1.45 pot=21.61 Rg=11.242 SPS=14714
bl=0 pos[1]=[10.5 8.2 -10.3] dr=1.25 t=0.0ps kin=1.47 pot=21.60 Rg=11.281 SPS=14948
bl=0 pos[1]=[10.0 8.4 -9.0] dr=1.24 t=0.0ps kin=1.51 pot=21.57 Rg=11.337 SPS=13961
bl=0 pos[1]=[8.4 9.3 -7.7] dr=1.22 t=0.0ps kin=1.49 pot=21.58 Rg=11.400 SPS=14744
bl=0 pos[1]=[7.8 10.0 -7.8] dr=1.22 t=0.0ps kin=1.52 pot=21.54 Rg=11.425 SPS=14162
bl=0 pos[1]=[7.3 10.8 -7.0] dr=1.24 t=0.0ps kin=1.48 pot=21.59 Rg=11.428 SPS=14925

```

(continues on next page)



(continued from previous page)

```

bl=0 pos[1]=[8.4 9.1 -6.6] dr=1.26 t=0.0ps kin=1.54 pot=21.55 Rg=11.445 SPS=14424
bl=0 pos[1]=[6.5 7.4 -8.5] dr=1.27 t=0.0ps kin=1.50 pot=21.58 Rg=11.461 SPS=15230
bl=0 pos[1]=[4.3 8.0 -8.5] dr=1.25 t=0.0ps kin=1.46 pot=21.62 Rg=11.491 SPS=15375
bl=0 pos[1]=[5.0 6.4 -8.1] dr=1.23 t=0.0ps kin=1.51 pot=21.61 Rg=11.560 SPS=15371
bl=0 pos[1]=[4.4 7.2 -8.0] dr=1.24 t=0.0ps kin=1.51 pot=21.56 Rg=11.603 SPS=14887
bl=0 pos[1]=[4.3 8.2 -8.4] dr=1.26 t=0.0ps kin=1.54 pot=21.60 Rg=11.607 SPS=15105
bl=0 pos[1]=[5.7 8.7 -8.6] dr=1.24 t=0.0ps kin=1.48 pot=21.59 Rg=11.607 SPS=14981
bl=0 pos[1]=[4.2 7.0 -8.3] dr=1.24 t=0.0ps kin=1.56 pot=21.60 Rg=11.580 SPS=15146
bl=0 pos[1]=[4.4 5.3 -6.8] dr=1.30 t=0.0ps kin=1.58 pot=21.59 Rg=11.521 SPS=13328
bl=0 pos[1]=[4.6 5.1 -6.5] dr=1.28 t=0.0ps kin=1.49 pot=21.58 Rg=11.535 SPS=16195
bl=0 pos[1]=[3.6 4.8 -5.3] dr=1.25 t=0.0ps kin=1.48 pot=21.63 Rg=11.554 SPS=15638
bl=0 pos[1]=[3.4 5.6 -4.8] dr=1.24 t=0.0ps kin=1.50 pot=21.58 Rg=11.580 SPS=14816
bl=0 pos[1]=[4.4 6.1 -3.8] dr=1.24 t=0.0ps kin=1.46 pot=21.58 Rg=11.602 SPS=14917
bl=0 pos[1]=[4.4 7.4 -4.3] dr=1.26 t=0.0ps kin=1.48 pot=21.60 Rg=11.552 SPS=12700
bl=0 pos[1]=[4.2 7.8 -4.4] dr=1.21 t=0.0ps kin=1.52 pot=21.61 Rg=11.576 SPS=15436
bl=0 pos[1]=[6.2 9.8 -5.8] dr=1.25 t=0.0ps kin=1.56 pot=21.55 Rg=11.631 SPS=14398
bl=0 pos[1]=[7.2 9.8 -5.7] dr=1.29 t=0.0ps kin=1.50 pot=21.60 Rg=11.655 SPS=15138
bl=0 pos[1]=[7.8 10.7 -4.1] dr=1.28 t=0.0ps kin=1.52 pot=21.59 Rg=11.660 SPS=15052
bl=0 pos[1]=[8.6 9.2 -4.5] dr=1.29 t=0.0ps kin=1.50 pot=21.61 Rg=11.693 SPS=14961
bl=0 pos[1]=[7.2 10.2 -4.2] dr=1.28 t=0.0ps kin=1.55 pot=21.54 Rg=11.746 SPS=15520
bl=0 pos[1]=[8.0 12.6 -5.2] dr=1.29 t=0.0ps kin=1.50 pot=21.59 Rg=11.766 SPS=14401
bl=0 pos[1]=[8.2 12.9 -6.5] dr=1.24 t=0.0ps kin=1.48 pot=21.59 Rg=11.787 SPS=15644
bl=0 pos[1]=[9.7 12.9 -7.9] dr=1.20 t=0.0ps kin=1.50 pot=21.58 Rg=11.788 SPS=15606
bl=0 pos[1]=[12.1 12.0 -8.2] dr=1.23 t=0.0ps kin=1.48 pot=21.59 Rg=11.747 SPS=14639
bl=0 pos[1]=[13.4 12.0 -8.6] dr=1.20 t=0.0ps kin=1.50 pot=21.56 Rg=11.703 SPS=14765
bl=0 pos[1]=[13.7 12.2 -8.6] dr=1.22 t=0.0ps kin=1.45 pot=21.63 Rg=11.680 SPS=14261
bl=0 pos[1]=[14.6 12.7 -8.6] dr=1.22 t=0.0ps kin=1.50 pot=21.59 Rg=11.652 SPS=15184
bl=0 pos[1]=[13.0 13.0 -8.5] dr=1.24 t=0.0ps kin=1.56 pot=21.60 Rg=11.633 SPS=15003
bl=0 pos[1]=[12.3 12.3 -8.9] dr=1.25 t=0.0ps kin=1.52 pot=21.59 Rg=11.578 SPS=14386
bl=0 pos[1]=[11.0 12.9 -8.4] dr=1.26 t=0.0ps kin=1.49 pot=21.59 Rg=11.540 SPS=15033
bl=0 pos[1]=[10.1 12.9 -9.0] dr=1.23 t=0.0ps kin=1.51 pot=21.60 Rg=11.467 SPS=14169
bl=0 pos[1]=[10.8 12.3 -9.9] dr=1.29 t=0.0ps kin=1.52 pot=21.59 Rg=11.401 SPS=13994
bl=0 pos[1]=[10.8 12.1 -10.5] dr=1.29 t=0.0ps kin=1.51 pot=21.62 Rg=11.337 SPS=14157
bl=0 pos[1]=[10.3 13.5 -10.0] dr=1.26 t=0.0ps kin=1.50 pot=21.59 Rg=11.319 SPS=14781
bl=0 pos[1]=[11.9 13.7 -10.6] dr=1.24 t=0.0ps kin=1.55 pot=21.54 Rg=11.295 SPS=14839
bl=0 pos[1]=[11.4 12.6 -10.3] dr=1.26 t=0.0ps kin=1.49 pot=21.60 Rg=11.277 SPS=14477
bl=0 pos[1]=[10.3 12.8 -9.5] dr=1.26 t=0.0ps kin=1.50 pot=21.61 Rg=11.286 SPS=14677
bl=0 pos[1]=[11.8 13.1 -8.9] dr=1.28 t=0.0ps kin=1.57 pot=21.53 Rg=11.312 SPS=13968
bl=0 pos[1]=[11.2 13.8 -6.9] dr=1.27 t=0.0ps kin=1.50 pot=21.59 Rg=11.309 SPS=14679
bl=0 pos[1]=[10.9 14.4 -6.6] dr=1.26 t=0.0ps kin=1.53 pot=21.55 Rg=11.277 SPS=14271
bl=0 pos[1]=[9.6 13.4 -6.9] dr=1.26 t=0.0ps kin=1.47 pot=21.56 Rg=11.218 SPS=14968
bl=0 pos[1]=[9.0 12.7 -9.2] dr=1.27 t=0.0ps kin=1.50 pot=21.55 Rg=11.185 SPS=14328
bl=0 pos[1]=[7.5 13.7 -10.1] dr=1.26 t=0.0ps kin=1.52 pot=21.57 Rg=11.177 SPS=12810
bl=0 pos[1]=[4.7 14.9 -9.6] dr=1.25 t=0.0ps kin=1.47 pot=21.55 Rg=11.184 SPS=14137
bl=0 pos[1]=[4.2 15.6 -7.9] dr=1.22 t=0.0ps kin=1.46 pot=21.56 Rg=11.219 SPS=14580
bl=0 pos[1]=[5.6 16.8 -6.3] dr=1.21 t=0.0ps kin=1.52 pot=21.56 Rg=11.321 SPS=14409
bl=0 pos[1]=[6.3 17.9 -7.7] dr=1.26 t=0.0ps kin=1.49 pot=21.56 Rg=11.400 SPS=15007
bl=0 pos[1]=[4.6 16.6 -8.6] dr=1.22 t=0.0ps kin=1.47 pot=21.57 Rg=11.409 SPS=14983
bl=0 pos[1]=[5.0 15.6 -8.3] dr=1.22 t=0.0ps kin=1.46 pot=21.58 Rg=11.411 SPS=14280
bl=0 pos[1]=[5.9 15.1 -7.9] dr=1.20 t=0.0ps kin=1.45 pot=21.61 Rg=11.401 SPS=15341
bl=0 pos[1]=[6.9 17.6 -7.4] dr=1.25 t=0.0ps kin=1.49 pot=21.57 Rg=11.396 SPS=15287
bl=0 pos[1]=[6.2 17.8 -6.1] dr=1.24 t=0.0ps kin=1.53 pot=21.58 Rg=11.349 SPS=14955

```

(continues on next page)

(continued from previous page)

```

bl=0 pos[1]=[6.8 17.2 -5.6] dr=1.26 t=0.0ps kin=1.48 pot=21.58 Rg=11.303 SPS=14678
bl=0 pos[1]=[7.8 17.2 -5.7] dr=1.24 t=0.0ps kin=1.46 pot=21.61 Rg=11.282 SPS=13784
bl=0 pos[1]=[9.7 16.6 -4.1] dr=1.25 t=0.0ps kin=1.49 pot=21.61 Rg=11.204 SPS=15248
bl=0 pos[1]=[9.8 16.7 -2.8] dr=1.26 t=0.0ps kin=1.55 pot=21.57 Rg=11.155 SPS=14220
bl=0 pos[1]=[9.7 17.2 -3.2] dr=1.25 t=0.0ps kin=1.47 pot=21.60 Rg=11.103 SPS=14423
bl=0 pos[1]=[8.6 17.9 -2.3] dr=1.27 t=0.0ps kin=1.48 pot=21.60 Rg=11.086 SPS=14795
bl=0 pos[1]=[6.9 18.8 -2.6] dr=1.26 t=0.0ps kin=1.51 pot=21.55 Rg=11.090 SPS=14605
bl=0 pos[1]=[5.6 18.1 -2.7] dr=1.24 t=0.0ps kin=1.46 pot=21.62 Rg=11.104 SPS=14324
bl=0 pos[1]=[5.2 18.4 -2.3] dr=1.25 t=0.0ps kin=1.49 pot=21.59 Rg=11.139 SPS=13517
bl=0 pos[1]=[5.6 17.0 -2.8] dr=1.26 t=0.0ps kin=1.51 pot=21.55 Rg=11.183 SPS=15119
bl=0 pos[1]=[4.6 15.4 -1.6] dr=1.23 t=0.0ps kin=1.51 pot=21.57 Rg=11.195 SPS=14360
bl=0 pos[1]=[3.8 15.3 -1.6] dr=1.24 t=0.0ps kin=1.48 pot=21.55 Rg=11.167 SPS=15966
bl=0 pos[1]=[4.7 14.6 -1.9] dr=1.26 t=0.0ps kin=1.45 pot=21.56 Rg=11.132 SPS=15034
bl=0 pos[1]=[5.0 14.4 -2.6] dr=1.25 t=0.0ps kin=1.47 pot=21.58 Rg=11.110 SPS=14345
bl=0 pos[1]=[6.2 15.3 -2.2] dr=1.22 t=0.0ps kin=1.50 pot=21.59 Rg=11.096 SPS=14920
bl=0 pos[1]=[5.3 16.2 -1.3] dr=1.24 t=0.0ps kin=1.53 pot=21.61 Rg=11.095 SPS=14239
bl=0 pos[1]=[6.3 16.6 -1.6] dr=1.28 t=0.0ps kin=1.54 pot=21.62 Rg=11.075 SPS=14562
bl=0 pos[1]=[6.6 18.5 -1.6] dr=1.26 t=0.0ps kin=1.53 pot=21.61 Rg=11.034 SPS=13436
bl=0 pos[1]=[6.6 19.7 -2.9] dr=1.22 t=0.0ps kin=1.50 pot=21.66 Rg=11.010 SPS=14621
bl=0 pos[1]=[7.1 18.9 -5.9] dr=1.25 t=0.0ps kin=1.46 pot=21.61 Rg=10.979 SPS=13823
bl=0 pos[1]=[6.4 16.5 -6.9] dr=1.24 t=0.0ps kin=1.49 pot=21.56 Rg=10.956 SPS=14835
bl=0 pos[1]=[6.4 15.2 -6.5] dr=1.26 t=0.0ps kin=1.50 pot=21.57 Rg=10.915 SPS=14853
bl=0 pos[1]=[4.0 14.4 -4.8] dr=1.26 t=0.0ps kin=1.51 pot=21.60 Rg=10.898 SPS=13865
bl=0 pos[1]=[2.0 13.9 -4.0] dr=1.25 t=0.0ps kin=1.46 pot=21.61 Rg=10.884 SPS=11310
bl=0 pos[1]=[2.1 13.7 -4.3] dr=1.22 t=0.0ps kin=1.48 pot=21.52 Rg=10.878 SPS=14068
bl=0 pos[1]=[4.3 12.2 -4.6] dr=1.26 t=0.0ps kin=1.47 pot=21.62 Rg=10.836 SPS=14063
bl=0 pos[1]=[6.1 12.2 -3.7] dr=1.23 t=0.0ps kin=1.53 pot=21.57 Rg=10.876 SPS=14536
bl=0 pos[1]=[7.3 12.0 -2.6] dr=1.24 t=0.0ps kin=1.46 pot=21.60 Rg=10.955 SPS=14182
bl=0 pos[1]=[8.0 11.1 -2.3] dr=1.22 t=0.0ps kin=1.52 pot=21.58 Rg=11.024 SPS=14548
bl=0 pos[1]=[8.4 11.3 -2.7] dr=1.25 t=0.0ps kin=1.49 pot=21.60 Rg=11.124 SPS=14879
bl=0 pos[1]=[8.9 11.3 -2.2] dr=1.23 t=0.0ps kin=1.53 pot=21.60 Rg=11.177 SPS=14002
bl=0 pos[1]=[9.6 10.5 -3.2] dr=1.22 t=0.0ps kin=1.52 pot=21.60 Rg=11.176 SPS=14717
bl=0 pos[1]=[9.3 10.0 -2.8] dr=1.25 t=0.0ps kin=1.50 pot=21.60 Rg=11.188 SPS=14164
bl=0 pos[1]=[10.1 10.2 -2.6] dr=1.25 t=0.0ps kin=1.51 pot=21.56 Rg=11.209 SPS=14505
bl=0 pos[1]=[8.2 10.0 -3.0] dr=1.26 t=0.0ps kin=1.54 pot=21.59 Rg=11.234 SPS=14060
bl=0 pos[1]=[6.4 11.3 -4.3] dr=1.29 t=0.0ps kin=1.48 pot=21.60 Rg=11.244 SPS=14579
bl=0 pos[1]=[5.9 10.6 -5.0] dr=1.26 t=0.0ps kin=1.54 pot=21.57 Rg=11.286 SPS=14090
bl=0 pos[1]=[5.1 10.1 -6.1] dr=1.27 t=0.0ps kin=1.48 pot=21.59 Rg=11.328 SPS=14923
bl=0 pos[1]=[4.9 9.1 -8.0] dr=1.25 t=0.0ps kin=1.52 pot=21.60 Rg=11.321 SPS=15368
bl=0 pos[1]=[5.7 9.3 -8.6] dr=1.25 t=0.0ps kin=1.51 pot=21.62 Rg=11.362 SPS=13698
bl=0 pos[1]=[5.3 9.6 -9.0] dr=1.26 t=0.0ps kin=1.56 pot=21.59 Rg=11.408 SPS=14979

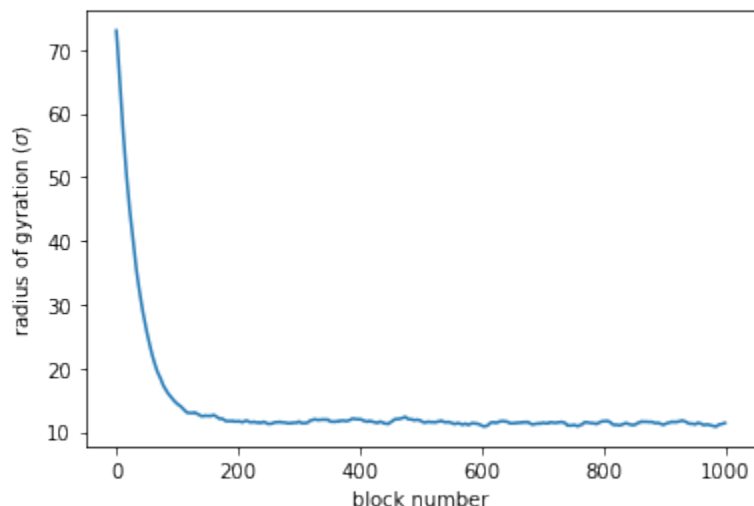
```

Some details about the output for each performed block:

**bl=0** is the number of each simulated block. In this case we set `increment=False`, so the number of steps is not being accounted. **pos[1]=[X,Y,Z]** is the spatial position for the first bead. **dr=1.26** shows the average position displacement in each block (in units of sigma). **t=0** is the total time (?). In this case we set `increment=False`, so the time is not being increased. **kin=1.5** is the kinetic energy of the system. **pot=19.90** is the potential energy of the system. **RG=7.654** is the radius of gyration in the end of each block. **SPS=12312** is the number of steps per second of each block. A measure of how fast the computations are being performed.

Check the convergence of the radius of gyration:

```
[13]: plt.plot(rg)
      plt.xlabel('block number')
      plt.ylabel(r'radius of gyration ($\sigma$)')
[13]: Text(0, 0.5, 'radius of gyration ($\\sigma$)')
```



The next step is to remove the restraining force in order to run the sampling simulation

```
[15]: sim.removeFlatBottomHarmonic()
```

Initiate the Adam optimization object. In this tutorial, this object is named “opt”

```
[16]: opt = AdamTraining(mu=3.22, rc = 1.78, eta=0.01, it=1)
```

Let’s take a look on Adam initialization parameters:

**mu** and **rc** are calculated to fit the experimental Hi-C map. These specific values are for GM12878 (see Di Pierro et al., 2016). **eta** is the learning rate (default value is 0.01, works well to train the pairwise interactions based on the experimental Hi-C maps) **it** is the iteration step.

Load the contact matrix file to the object opt

```
[17]: opt.getPars(HiC="input/chr10_100k.dense")
```

The optimization step uses the contact map from the simulation to update the values in the force field. Therefore, the next step is to perform a long simulation with enough statistics to obtain a converged simulated contact map.

In order to get a good inversion calculation, it is important to have around  $1 \times 10^5$  frames from a set of different simulations. For example, 20 replicas of 5000 saved frames.

This can take some time, so in this tutorial we will use just 1 replica of 5000 frames saved every 1000 steps.

$block = 1 \times 10^3$   $n\_blocks = 5000$

```
[18]: block = 1000
      n_blocks = 50
```

```
[19]: for _ in range(n_blocks):
      sim.runSimBlock(block, increment=True) #perform 1 block of the simulation
```

(continues on next page)

(continued from previous page)

```
state = sim.getPositions() #get the positions for each block
opt.probCalc(state) #calculate and store the contact map for each block
```

```
bl=1 pos[1]=[5.5 10.3 -7.9] dr=1.88 t=0.0ps kin=1.46 pot=21.58 Rg=11.463 SPS=11324
bl=2 pos[1]=[6.7 11.7 -9.4] dr=1.88 t=0.0ps kin=1.50 pot=21.53 Rg=11.610 SPS=11625
bl=3 pos[1]=[6.7 11.2 -9.9] dr=1.89 t=0.0ps kin=1.50 pot=21.51 Rg=11.857 SPS=13000
bl=4 pos[1]=[6.1 8.8 -10.1] dr=1.93 t=0.0ps kin=1.52 pot=21.51 Rg=12.107 SPS=15198
bl=5 pos[1]=[8.1 8.0 -8.6] dr=1.95 t=0.0ps kin=1.54 pot=21.53 Rg=12.210 SPS=15751
bl=6 pos[1]=[8.0 8.4 -4.9] dr=1.92 t=0.0ps kin=1.52 pot=21.52 Rg=12.351 SPS=15347
bl=7 pos[1]=[4.6 8.0 -7.0] dr=1.89 t=0.0ps kin=1.48 pot=21.56 Rg=12.410 SPS=15001
bl=8 pos[1]=[5.7 11.8 -4.5] dr=1.91 t=0.0ps kin=1.49 pot=21.59 Rg=12.485 SPS=14968
bl=9 pos[1]=[6.4 13.6 -5.2] dr=1.92 t=0.0ps kin=1.52 pot=21.53 Rg=12.653 SPS=14891
bl=10 pos[1]=[4.5 10.4 -4.5] dr=1.91 t=0.0ps kin=1.55 pot=21.49 Rg=12.824 SPS=15702
bl=11 pos[1]=[6.1 9.3 -7.0] dr=1.90 t=0.0ps kin=1.52 pot=21.52 Rg=12.994 SPS=15009
bl=12 pos[1]=[5.9 13.4 -9.1] dr=1.90 t=0.0ps kin=1.53 pot=21.58 Rg=13.158 SPS=14747
bl=13 pos[1]=[5.5 14.2 -9.4] dr=1.93 t=0.0ps kin=1.53 pot=21.49 Rg=13.266 SPS=14975
bl=14 pos[1]=[5.3 10.1 -9.4] dr=1.91 t=0.0ps kin=1.48 pot=21.55 Rg=13.334 SPS=14952
bl=15 pos[1]=[3.8 8.2 -8.7] dr=1.91 t=0.0ps kin=1.53 pot=21.58 Rg=13.469 SPS=14917
bl=16 pos[1]=[3.3 7.3 -8.3] dr=1.83 t=0.0ps kin=1.52 pot=21.49 Rg=13.521 SPS=15231
bl=17 pos[1]=[3.3 6.4 -9.3] dr=1.94 t=0.0ps kin=1.44 pot=21.55 Rg=13.493 SPS=14712
bl=18 pos[1]=[4.3 6.2 -11.1] dr=1.92 t=0.0ps kin=1.49 pot=21.52 Rg=13.548 SPS=15783
bl=19 pos[1]=[7.4 6.9 -12.6] dr=1.87 t=0.0ps kin=1.44 pot=21.56 Rg=13.681 SPS=15239
bl=20 pos[1]=[7.0 8.8 -12.4] dr=1.85 t=0.0ps kin=1.48 pot=21.52 Rg=13.740 SPS=14830
bl=21 pos[1]=[5.1 9.8 -11.0] dr=1.94 t=0.0ps kin=1.50 pot=21.55 Rg=13.804 SPS=15532
bl=22 pos[1]=[4.0 8.0 -10.9] dr=1.89 t=0.0ps kin=1.52 pot=21.52 Rg=13.964 SPS=15286
bl=23 pos[1]=[3.8 10.5 -9.6] dr=1.87 t=0.0ps kin=1.52 pot=21.57 Rg=14.019 SPS=15235
bl=24 pos[1]=[3.5 11.2 -10.9] dr=1.90 t=0.0ps kin=1.43 pot=21.54 Rg=14.053 SPS=16519
bl=25 pos[1]=[6.2 10.7 -11.7] dr=1.93 t=0.0ps kin=1.53 pot=21.56 Rg=14.008 SPS=16787
bl=26 pos[1]=[6.6 7.7 -12.1] dr=1.97 t=0.0ps kin=1.51 pot=21.53 Rg=14.083 SPS=16116
bl=27 pos[1]=[4.4 8.6 -10.6] dr=1.96 t=0.0ps kin=1.49 pot=21.58 Rg=14.146 SPS=16359
bl=28 pos[1]=[4.8 12.2 -10.4] dr=1.95 t=0.0ps kin=1.46 pot=21.55 Rg=14.174 SPS=16202
bl=29 pos[1]=[5.2 12.6 -9.8] dr=1.97 t=0.0ps kin=1.51 pot=21.54 Rg=14.196 SPS=16520
bl=30 pos[1]=[3.7 12.9 -9.9] dr=1.91 t=0.0ps kin=1.45 pot=21.52 Rg=14.144 SPS=15429
bl=31 pos[1]=[5.8 9.6 -7.4] dr=1.88 t=0.0ps kin=1.49 pot=21.54 Rg=14.221 SPS=15646
bl=32 pos[1]=[6.3 10.4 -6.3] dr=1.88 t=0.0ps kin=1.48 pot=21.52 Rg=14.266 SPS=15488
bl=33 pos[1]=[5.6 6.7 -7.0] dr=1.91 t=0.0ps kin=1.47 pot=21.50 Rg=14.302 SPS=16323
bl=34 pos[1]=[6.1 6.5 -8.3] dr=1.94 t=0.0ps kin=1.54 pot=21.54 Rg=14.444 SPS=15755
bl=35 pos[1]=[7.9 7.1 -7.6] dr=1.90 t=0.0ps kin=1.53 pot=21.51 Rg=14.456 SPS=15281
bl=36 pos[1]=[6.2 5.0 -10.2] dr=1.96 t=0.0ps kin=1.52 pot=21.54 Rg=14.557 SPS=15919
bl=37 pos[1]=[5.0 7.4 -10.4] dr=1.96 t=0.0ps kin=1.49 pot=21.52 Rg=14.650 SPS=15861
bl=38 pos[1]=[7.3 9.5 -9.1] dr=1.83 t=0.0ps kin=1.54 pot=21.56 Rg=14.604 SPS=16106
bl=39 pos[1]=[8.0 8.5 -10.2] dr=1.98 t=0.0ps kin=1.51 pot=21.56 Rg=14.597 SPS=16383
bl=40 pos[1]=[6.7 7.6 -8.3] dr=1.91 t=0.0ps kin=1.47 pot=21.53 Rg=14.613 SPS=16118
bl=41 pos[1]=[6.8 7.1 -7.4] dr=1.94 t=0.0ps kin=1.53 pot=21.53 Rg=14.552 SPS=15405
bl=42 pos[1]=[7.9 7.4 -5.8] dr=1.96 t=0.0ps kin=1.54 pot=21.54 Rg=14.560 SPS=15734
bl=43 pos[1]=[8.6 6.8 -6.3] dr=1.94 t=0.0ps kin=1.56 pot=21.56 Rg=14.638 SPS=16091
bl=44 pos[1]=[10.7 8.1 -7.1] dr=1.92 t=0.0ps kin=1.52 pot=21.52 Rg=14.659 SPS=15797
bl=45 pos[1]=[13.2 7.3 -4.7] dr=1.94 t=0.0ps kin=1.50 pot=21.53 Rg=14.690 SPS=15806
bl=46 pos[1]=[12.8 5.8 -6.6] dr=1.91 t=0.0ps kin=1.50 pot=21.56 Rg=14.725 SPS=15625
bl=47 pos[1]=[10.2 4.3 -7.1] dr=1.93 t=0.0ps kin=1.46 pot=21.50 Rg=14.795 SPS=16080
bl=48 pos[1]=[9.6 8.2 -9.8] dr=1.89 t=0.0ps kin=1.46 pot=21.47 Rg=14.775 SPS=15822
bl=49 pos[1]=[7.1 5.9 -9.0] dr=1.94 t=0.0ps kin=1.51 pot=21.48 Rg=14.797 SPS=15511
```

(continues on next page)

(continued from previous page)

```

Statistics for the simulation opt_chr10_100K, number of particles: 1356, number of
↳chains: 1

Statistics for particle position
  mean position is: [-1.23957781  1.7779258  2.78054735]  Rg =  14.796688
  median bond size is  0.9680976445921962
  three shortest/longest (<10)/ bonds are  [0.86936723  0.88400501  0.88756599]  [1.
↳09114228  1.09780597  1.10176906]
  95 percentile of distance to center is:    21.431837046446862
  density of closest 95% monomers is:    0.031240393096327784
  density of the core monomers is:    0.07548378809911002
  min/median/mean/max coordinates are:
  x: -21.97, -1.27, -1.24, 20.78
  y: -18.23, 1.69, 1.78, 28.29
  z: -12.99, 2.12, 2.78, 20.77

Statistics for velocities:
  mean kinetic energy is:  1.512121087969534 should be: 1.5
  fastest particles are (in kT):  [7.25928234  7.28251669  7.57800841  7.84269129  8.
↳89079614]

Statistics for the system:
  Forces are:  ['FENEBond', 'AngleForce', 'RepulsiveSoftCore', 'CustomTypes']
  Number of exceptions:  1355

Potential Energy Ep =  21.483418602507374
bl=50 pos[1]=[5.9 7.4 -7.8] dr=1.97 t=0.0ps kin=1.50 pot=21.44 Rg=14.789 SPS=16144

```

In the end of each simulation replica we need to save some important values required to calculate the Adam step.

We save these values using the H5 compressed files because of the efficiency writing/reading them.

Note: attention to this step. We have two files for each replica for each iteration step. Be organized!

```

[20]: replica=1

with h5py.File(sim.folder + "/Pi_" + str(replica)+".h5", 'w') as hf:
    hf.create_dataset("Pi", data=opt.Pi)

np.savetxt(sim.folder + "/Nframes_" + str(replica), [opt.NFrames])

```

The first part of the optimization is finished. Inside the output folder, for each iteration, there are these 2 files used in next step:

Nframes\_1 Pi\_1.h5

```

[21]: %%bash
ls iteration_1/

Nframes_1
opt_chr10_100K_0_block0.pdb
Pi_1.h5
platform_info.dat
probdist_1

```

The second part is the inversion. It is quite simple, just feed the optimization object with the files from all replicas and make the inversion to get the updated force field file.

```
[22]: opt2 = AdamTraining(mu=3.22, rc = 1.78, eta=0.01, it=1)
      opt2.getPars(HiC="input/chr10_100k.dense")
```

```
[23]: Nreplicas = 1
      path = "iteration_1"
      for rep in range(1,Nreplicas+1):
          print("reading files from replica {}".format(rep))
          with h5py.File(path + "/Pi_" + str(rep)+".h5", 'r') as hf:
              opt2.Pi += hf['Pi'][:]
              opt2.NFrames += int(np.loadtxt(path + "/Nframes_" + str(rep)))
```

```
reading files from replica 1
```

With the contact probabilities from with all replicas, we calculate the inversion and get the updated parameters for the force field.

To calculate the new FF, we need to load the old one.

```
[24]: lamb_new = opt2.getLamb(Lambdas="input/lambda_1")
```

```
[25]: lamb_new
```

```
[25]:      Bead1    Bead2    Bead3    Bead4    Bead5    Bead6    Bead7    Bead8    Bead9    \
0      -0.00999 -0.01000  0.01000 -0.01000 -0.01000 -0.01   -0.01   -0.01   -0.01
1      -0.01000 -0.00999 -0.01000  0.01000 -0.01000 -0.01   -0.01   -0.01   -0.01
2       0.01000 -0.01000 -0.00999 -0.01000  0.01000 -0.01   -0.01   -0.01   -0.01
3      -0.01000  0.01000 -0.01000 -0.00999 -0.01000  0.01   -0.01   -0.01   -0.01
4      -0.01000 -0.01000  0.01000 -0.01000 -0.00999 -0.01    0.01   -0.01   -0.01
...      ...      ...      ...      ...      ...      ...      ...      ...      ...
1351 -0.01000 -0.01000 -0.01000 -0.01000 -0.01000 -0.01   -0.01   -0.01   -0.01
1352 -0.01000 -0.01000 -0.01000 -0.01000 -0.01000 -0.01   -0.01   -0.01   -0.01
1353 -0.01000 -0.01000 -0.01000 -0.01000 -0.01000 -0.01   -0.01   -0.01   -0.01
1354 -0.01000 -0.01000 -0.01000 -0.01000 -0.01000 -0.01   -0.01   -0.01   -0.01
1355 -0.01000 -0.01000 -0.01000 -0.01000 -0.01000 -0.01   -0.01   -0.01   -0.01

      Bead10    ...    Bead1347    Bead1348    Bead1349    Bead1350    Bead1351    Bead1352    \
0      -0.01    ...      -0.01      -0.01      -0.01      -0.01      -0.01      -0.01000
1      -0.01    ...      -0.01      -0.01      -0.01      -0.01      -0.01      -0.01000
2      -0.01    ...      -0.01      -0.01      -0.01      -0.01      -0.01      -0.01000
3      -0.01    ...      -0.01      -0.01      -0.01      -0.01      -0.01      -0.01000
4      -0.01    ...      -0.01      -0.01      -0.01      -0.01      -0.01      -0.01000
...      ...      ...      ...      ...      ...      ...      ...      ...
1351 -0.01    ...      -0.01      -0.01      -0.01      0.01      -0.01      -0.00999
1352 -0.01    ...      -0.01      -0.01      -0.01      -0.01      0.01      -0.01000
1353 -0.01    ...      -0.01      -0.01      -0.01      -0.01      -0.01      0.01000
1354 -0.01    ...      -0.01      -0.01      -0.01      -0.01      -0.01      -0.01000
1355 -0.01    ...      -0.01      -0.01      -0.01      -0.01      -0.01      -0.01000

      Bead1353    Bead1354    Bead1355    Bead1356
0      -0.01000    -0.01000    -0.01000    -0.01000
1      -0.01000    -0.01000    -0.01000    -0.01000
2      -0.01000    -0.01000    -0.01000    -0.01000
```

(continues on next page)

(continued from previous page)

```

3      -0.010000  -0.010000  -0.010000  -0.010000
4      -0.010000  -0.010000  -0.010000  -0.010000
...
1351   -0.010000   0.010000  -0.010000  -0.010000
1352   -0.009999  -0.010000   0.010000  -0.010000
1353   -0.010000  -0.009999  -0.010000   0.010000
1354    0.010000  -0.010000  -0.009999  -0.010000
1355   -0.010000   0.010000  -0.010000  -0.009999

```

```
[1356 rows x 1356 columns]
```

Save the new force field to a file. To keep the organization, we use the same name followed by the number of the new iteration:

```
[26]: lamb_new.to_csv("input/lambda_2", index=False)
```

We can plot the parameters of the force field to check the changes between the 2 steps:

```

[27]: ff_old = pd.read_csv("input/lambda_1")
      ff_new = pd.read_csv("input/lambda_2")

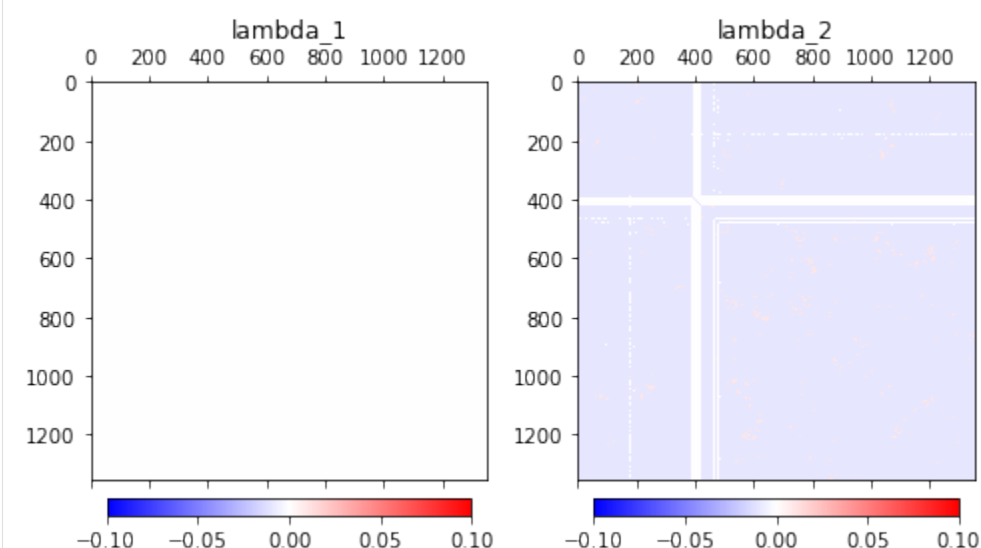
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(8,4))

f1 = axes[0].matshow(ff_old.values, vmin=-0.1, vmax=0.1, cmap="bwr")
axes[0].set_title('lambda_1', loc='center')
fig.colorbar(f1, orientation='horizontal', ax=axes[0], shrink=0.9, fraction=0.046, pad=0.
↪04)

f2 = axes[1].matshow(ff_new.values, vmin=-0.1, vmax=0.1, cmap="bwr")
axes[1].set_title('lambda_2', loc='center')
fig.colorbar(f2, orientation='horizontal', ax=axes[1], shrink=0.9, fraction=0.046, pad=0.
↪04)

```

```
[27]: <matplotlib.colorbar.Colorbar at 0x7f76c41f9a60>
```



We can also check the error between experimental and simulation Hi-C maps:

```
[28]: print(opt2.error)
0.9527989958833853
```

Save/plot the simulation Hi-C and compare to the experimental one

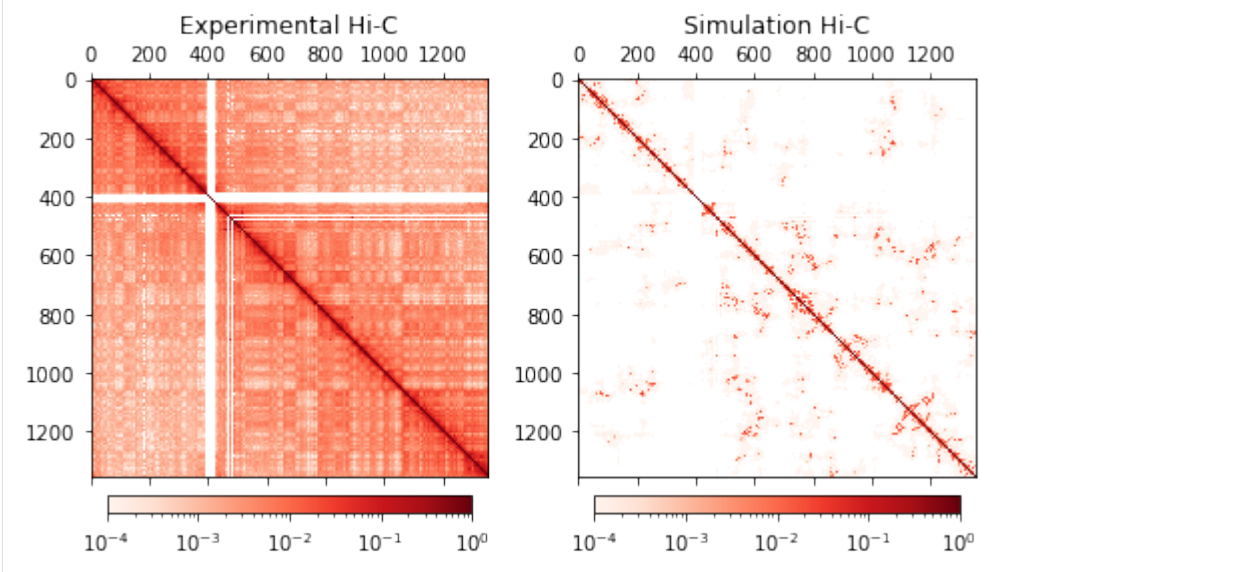
```
[30]: #Save
np.savetxt("iteration_1/probdist_1", opt2.phi_sim)

#Plot
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(8,4))

f1 = axes[0].matshow(opt2.phi_exp,norm=matplotlib.colors.LogNorm(vmin=0.0001, vmax=opt2.phi_exp.
↪max()),cmap="Reds")
axes[0].set_title('Experimental Hi-C', loc='center')
fig.colorbar(f1, orientation='horizontal', ax=axes[0], shrink=0.9, fraction=0.046, pad=0.
↪04)

f2 = axes[1].matshow(opt2.phi_sim,norm=matplotlib.colors.LogNorm(vmin=0.0001, vmax=opt2.phi_exp.
↪max()),cmap="Reds")
axes[1].set_title('Simulation Hi-C', loc='center')
fig.colorbar(f2, orientation='horizontal', ax=axes[1], shrink=0.9, fraction=0.046, pad=0.
↪04)
```

```
[30]: <matplotlib.colorbar.Colorbar at 0x7f76c4103d30>
```



As one can see, the simulation Hi-C map is not converged. This is due to two main reasons: 1 - We just performed a single simulation instead of multiple replicas; 2 - The force field remains almost with zero interactions due to the number of optimization steps.

Redo these steps using the new lambda file (lambda\_2) as input for customTypes potential in the next iteration.



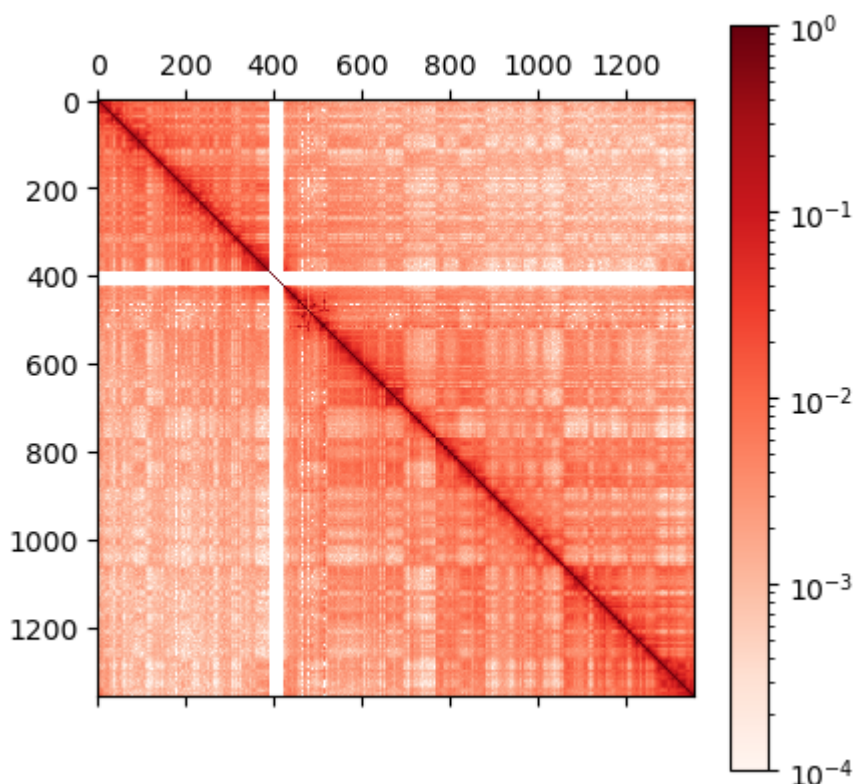
## 7.2 Fixing values on the force field

If we inspect again the experimental Hi-C map you can observe that some regions have a white band. These gaps arise from difficulty of sequencing those regions, often highly repetitive regions. The centromere of the chromosomes is one of them.

```
[70]: filename = 'input/chr10_100k.dense'
hic_file = np.loadtxt(filename)

r=np.triu(hic_file, k=1)
r[np.isnan(r)]= 0.0
r = normalize(r, axis=1, norm='max')
rd = np.transpose(r)
r=r+rd + np.diag(np.ones(len(r)))
plt.matshow(r,norm=matplotlib.colors.LogNorm(vmin=0.0001, vmax=r.max()),cmap="Reds")
plt.colorbar()
```

```
[70]: <matplotlib.colorbar.Colorbar at 0x7f81b07e71f0>
```



So, when we perform the Adam optimization, the parameters regarding the interactions of these regions will not be trained because of the lack of data.

One way to solve this problem is to define some interactions initially and then keep these fixed over iterations.

Find the index position of each white band

```
[71]: #get the first diagonal (first neighbor)
diag1 = np.diag(r, k=1)
```

(continues on next page)

(continued from previous page)

```
#find the index where the first neighbor is equal to zero
centromere_index = np.where(diag1 == 0)[0]
print(centromere_index)

[391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408
 409 410 411 412 413 414 415 416 417 418 419 420 421 422 463 464 466 467
 468 477 478 480 481 482 490]
```

Now we can create the new initial force field parameters' matrix with a specific value for these interactions. Here we chose  $-0.3$  for centromere-centromere interactions and  $-0.2$  for centromere-other beads.

```
[72]: #read the sequence file
seq = np.loadtxt("input/seq_chr10", dtype=str)[: ,1]
pol_size = len(seq)
#create a NxN matrix fill with zeros, where N is the chromossome lenght
data = np.zeros((pol_size,pol_size))

#Set values for centromere regions and NA regions
cent = [391,422]
NAs = [463, 464, 466, 467, 468, 477, 478, 480, 481, 482, 490]
data[cent[0]:cent[1],:] = -0.2
data[:,cent[0]:cent[1]] = -0.2
data[cent[0]:cent[1],cent[0]:cent[1]] = -0.3
for na in NAs:
    data[na,:] = -0.2
    data[:,na] = -0.2

#transform it in a Pandas dataframe using the sequence names as header
lamb = pd.DataFrame(data, columns=seq)
#save to a csv file format in the "input" folder
lamb.to_csv("input/lambda_fixed_1", index=False)
lamb
```

```
[72]:
```

	Bead1	Bead2	Bead3	Bead4	Bead5	Bead6	Bead7	Bead8	Bead9	Bead10	\
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	
1351	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1352	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1353	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1354	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1355	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	
	...	Bead1347	Bead1348	Bead1349	Bead1350	Bead1351	Bead1352	\			
0	...	0.0	0.0	0.0	0.0	0.0	0.0				
1	...	0.0	0.0	0.0	0.0	0.0	0.0				
2	...	0.0	0.0	0.0	0.0	0.0	0.0				
3	...	0.0	0.0	0.0	0.0	0.0	0.0				
4	...	0.0	0.0	0.0	0.0	0.0	0.0				

(continues on next page)

(continued from previous page)

```

...    ...    ...    ...    ...    ...    ...    ...
1351    ...    0.0    0.0    0.0    0.0    0.0    0.0
1352    ...    0.0    0.0    0.0    0.0    0.0    0.0
1353    ...    0.0    0.0    0.0    0.0    0.0    0.0
1354    ...    0.0    0.0    0.0    0.0    0.0    0.0
1355    ...    0.0    0.0    0.0    0.0    0.0    0.0

```

```

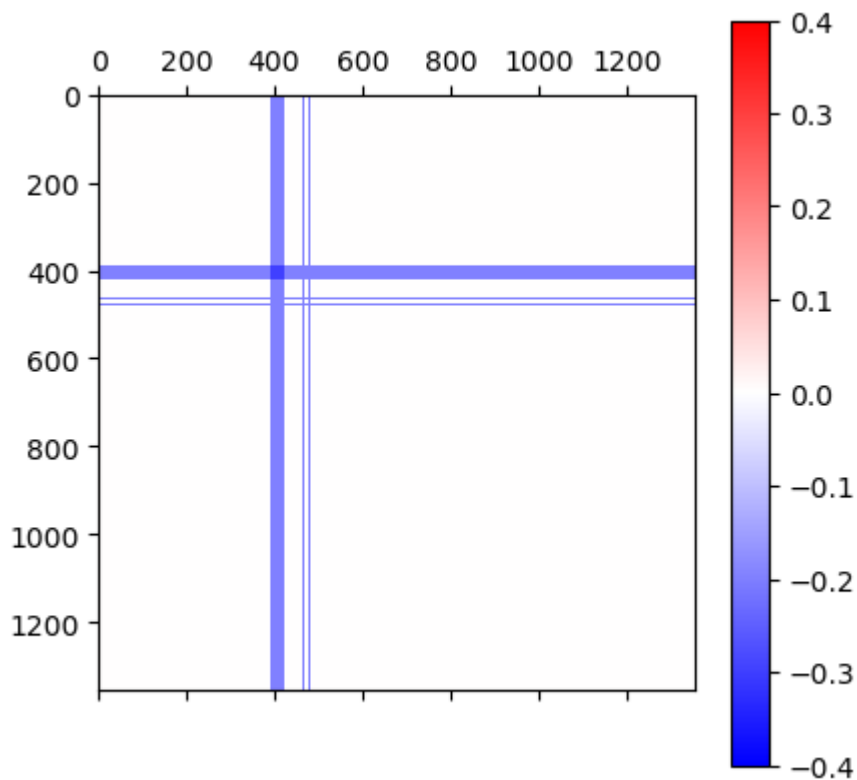
      Bead1353  Bead1354  Bead1355  Bead1356
0          0.0      0.0      0.0      0.0
1          0.0      0.0      0.0      0.0
2          0.0      0.0      0.0      0.0
3          0.0      0.0      0.0      0.0
4          0.0      0.0      0.0      0.0
...    ...    ...    ...    ...
1351      0.0      0.0      0.0      0.0
1352      0.0      0.0      0.0      0.0
1353      0.0      0.0      0.0      0.0
1354      0.0      0.0      0.0      0.0
1355      0.0      0.0      0.0      0.0

```

```
[1356 rows x 1356 columns]
```

```
[73]: plt.matshow(lamb.values, vmin=-0.4, vmax=0.4, cmap="bwr")
      plt.colorbar()
```

```
[73]: <matplotlib.colorbar.Colorbar at 0x7f81b0a783d0>
```



Save the information about the index of the gaps.

When calling the Adam optimization step, we can define fixed points. We provide a list of pair of interactions that will remain unchanged throughout the optimization procedure.

```
[74]: import itertools
      #list of all beads
      N = np.arange(opt2.expHiC.shape[0])

      #use the itertools to create all pair of interactions
      fix_points = list(itertools.product(centromere_index,N))
```

```
[ ]: #Use the following line to update the parameters maintaining the desired interactions,
      ↪fixed
      #lamb_new = opt2.getLamb(Lambdas="input/lambda_1", fixedPoints=fix_points)
```

## HOW TO CITE OPENMICHROM

Please acknowledge the use of this software within the body of your publication for example by copying or adapting the following formulation:

*The chromatin dynamics simulations were performed using Open-MiChroM package[1].*

Open-MiChroM - [1] Oliveira Jr, A. B., Contessoto, V. G., Mello, M. F., & Onuchic, J. N. (2020). A scalable computational approach for simulating complexes of multiple chromosomes. *Journal of Molecular Biology*, 433(6), 166700.

Optimization - [2] Oliveira Jr, A. B., et al. "Chromosome Modeling on Downsampled Hi-C Maps Enhances the Compartmentalization Signal." *The Journal of Physical Chemistry B*, vol. 125, no. 31, Aug. 2021, pp. 8757–67.

CNDB file format - [3] Contessoto, Vinícius G, et al. "The Nucleome Data Bank: Web-Based Resources to Simulate and Analyze the Three-Dimensional Genome." *Nucleic Acids Research*, vol. 49, no. D1, Jan. 2021, pp. D172–82.

The paper are available online at:

[1] [Journal of Molecular Biology \(JMB\)](#).

[2] [Journal of Physical Chemistry B \(JPCB\)](#).

[3] [Nucleic Acids Research \(NAR\)](#).

To cite these references, you can use the following BibTeX entry:

```
@article{OpenMiChroM_2020,
  title={A scalable computational approach for simulating complexes of multiple_
↪chromosomes},
  author={Oliveira Jr, Antonio B and
  Contessoto, Vinicius G and
  Mello, Matheus F and
  Onuchic, Jose N},
  journal={Journal of Molecular Biology},
  volume={433},
  number={6},
  pages={166700},
  year={2020},
  publisher={Elsevier}
  doi = {https://doi.org/10.1016/j.jmb.2020.10.034},
}
```

```
@article{OpenMiChroM_Opt_2021,
  title = {Chromosome {{Modeling}} on {{Downsampled Hi-C Maps Enhances}} the {
↪{{Compartmentalization Signal}}}},
```

(continues on next page)

(continued from previous page)

```

author = {Oliveira Junior, Antonio B. and Estrada, Cynthia Perez and Aiden, Erez,
↪Lieberman and Contessoto, Vin{'i}cius G. and Onuchic, Jos{'e} N.},
year = {2021},
month = aug,
journal = {The Journal of Physical Chemistry B},
volume = {125},
number = {31},
pages = {8757--8767},
publisher = {{American Chemical Society}},
issn = {1520-6106},
doi = {10.1021/acs.jpcc.1c04174},
}

@article{NucleomeDataBank_2021,
  title = {The {{Nucleome Data Bank}}: Web-Based Resources to Simulate and Analyze the
↪Three-Dimensional Genome},
  shorttitle = {The {{Nucleome Data Bank}}},
  author = {Contessoto, Vin{'i}cius G and Cheng, Ryan R and Hajitaheri, Arya and
↪{Doderro-Rojas}, Esteban and Mello, Matheus F and {Lieberman-Aiden}, Erez and Wolynes,
↪Peter~G and Di~Pierro, Michele and Onuchic, Jos{'e} N},
  year = {2021},
  month = jan,
  journal = {Nucleic Acids Research},
  volume = {49},
  number = {D1},
  pages = {D172-D182},
  issn = {0305-1048},
  doi = {10.1093/nar/gkaa818},
}

```

## REFERENCES

- Antonio B. Oliveira Junior, Vinícius G. Contessoto, Matheus F. Mello, and José N. Onuchic. A Scalable Computational Approach for Simulating Complexes of Multiple Chromosomes. *Journal of Molecular Biology*, 433(6):166700, March 2021. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0022283620306185> (visited on 2021-05-15), doi:10.1016/j.jmb.2020.10.034.
- Michele Di Pierro, Bin Zhang, Erez Lieberman Aiden, Peter G. Wolynes, and José N. Onuchic. Transferable model for chromosome architecture. *Proceedings of the National Academy of Sciences*, 113(43):12168–12173, October 2016. URL: <http://www.pnas.org/lookup/doi/10.1073/pnas.1613607113>, doi:10.1073/pnas.1613607113.
- Vinícius G Contessoto, Ryan R Cheng, Arya Hajitaheri, Esteban Dodero-Rojas, Matheus F Mello, Erez Lieberman-Aiden, Peter G Wolynes, Michele Di Pierro, and José N Onuchic. The Nucleome Data Bank: web-based resources to simulate and analyze the three-dimensional genome. *Nucleic Acids Research*, 49(D1):D172–D182, January 2021. URL: <https://academic.oup.com/nar/article/49/D1/D172/5918320>, doi:10.1093/nar/gkaa818.
- Ryan R Cheng, Vinicius G Contessoto, Erez Lieberman Aiden, Peter G Wolynes, Michele Di Pierro, and Jose N Onuchic. Exploring chromosomal structural heterogeneity across multiple cell lines. *eLife*, 9:e60312, October 2020. URL: <https://elifesciences.org/articles/60312>, doi:10.7554/eLife.60312.
- Bin Zhang and Peter G. Wolynes. Topology, structures, and energy landscapes of human chromosomes. *Proceedings of the National Academy of Sciences*, 112(19):6062–6067, May 2015. URL: <http://www.pnas.org/lookup/doi/10.1073/pnas.1506257112>, doi:10.1073/pnas.1506257112.





**LICENSE****MIT License**

Copyright (c) 2020-2023 The Center for Theoretical Biological Physics (CTBP) - Rice ↵  
↵ University

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## BIBLIOGRAPHY

- [1] Antonio B. Oliveira Junior, Vinícius G. Contessoto, Matheus F. Mello, and José N. Onuchic. A Scalable Computational Approach for Simulating Complexes of Multiple Chromosomes. *Journal of Molecular Biology*, 433(6):166700, March 2021. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0022283620306185> (visited on 2021-05-15), doi:10.1016/j.jmb.2020.10.034.
- [2] Michele Di Pierro, Bin Zhang, Erez Lieberman Aiden, Peter G. Wolynes, and José N. Onuchic. Transferable model for chromosome architecture. *Proceedings of the National Academy of Sciences*, 113(43):12168–12173, October 2016. URL: <http://www.pnas.org/lookup/doi/10.1073/pnas.1613607113>, doi:10.1073/pnas.1613607113.
- [3] Vinícius G Contessoto, Ryan R Cheng, Arya Hajitaheri, Esteban Dodero-Rojas, Matheus F Mello, Erez Lieberman-Aiden, Peter G Wolynes, Michele Di Pierro, and José N Onuchic. The Nucleome Data Bank: web-based resources to simulate and analyze the three-dimensional genome. *Nucleic Acids Research*, 49(D1):D172–D182, January 2021. URL: <https://academic.oup.com/nar/article/49/D1/D172/5918320>, doi:10.1093/nar/gkaa818.



## PYTHON MODULE INDEX

### O

OpenMiChroM.ChromDynamics, [5](#)  
OpenMiChroM.CndbTools, [20](#)  
OpenMiChroM.Optimization, [15](#)





## A

AdamTraining (class in *OpenMiChroM.Optimization*), 15

addAdditionalForce() (OpenMiChroM.ChromDynamics.MiChroM method), 5

addAngles() (OpenMiChroM.ChromDynamics.MiChroM method), 6

addBond() (OpenMiChroM.ChromDynamics.MiChroM method), 6

addCustomIC() (OpenMiChroM.ChromDynamics.MiChroM method), 6

addCustomTypes() (OpenMiChroM.ChromDynamics.MiChroM method), 6

addFENEbonds() (OpenMiChroM.ChromDynamics.MiChroM method), 7

addFlatBottomHarmonic() (OpenMiChroM.ChromDynamics.MiChroM method), 7

addIdealChromosome() (OpenMiChroM.ChromDynamics.MiChroM method), 7

addLoops() (OpenMiChroM.ChromDynamics.MiChroM method), 7

addMultiChainIC() (OpenMiChroM.ChromDynamics.MiChroM method), 8

addRepulsiveSoftCore() (OpenMiChroM.ChromDynamics.MiChroM method), 8

addSphericalConfinementLJ() (OpenMiChroM.ChromDynamics.MiChroM method), 9

addTypeToType() (OpenMiChroM.ChromDynamics.MiChroM method), 9

appCutoff() (OpenMiChroM.Optimization.FullTraining method), 19

## C

calc\_exp\_phi() (OpenMiChroM.Optimization.CustomMiChroMTraining method), 17

calc\_exp\_phi\_types() (OpenMiChroM.Optimization.CustomMiChroMTraining method), 17

calc\_sim\_phi() (OpenMiChroM.Optimization.CustomMiChroMTraining method), 17

calc\_sim\_phi\_types() (OpenMiChroM.Optimization.CustomMiChroMTraining method), 17

chromRG() (OpenMiChroM.ChromDynamics.MiChroM method), 9

cndbTools (class in *OpenMiChroM.CndbTools*), 20

compute\_Chirality() (OpenMiChroM.CndbTools.cndbTools method), 20

compute\_FFT\_from\_Oij() (OpenMiChroM.CndbTools.cndbTools method), 20

compute\_GyrTensorEigs() (OpenMiChroM.CndbTools.cndbTools method), 20

compute\_MSD() (OpenMiChroM.CndbTools.cndbTools method), 20

compute\_Orientation\_OP() (OpenMiChroM.CndbTools.cndbTools method), 20

compute\_RadNumDens() (OpenMiChroM.CndbTools.cndbTools method), 21

compute\_RDP() (OpenMiChroM.CndbTools.cndbTools method), 21

compute\_RG() (OpenMiChroM.CndbTools.cndbTools method), 21

createInitialLambda() (OpenMiChroM.Optimization.FullTraining method), 19

createLine() (OpenMiChroM.ChromDynamics.MiChroM method), 9

createRandomWalk() (OpenMiChroM.ChromDynamics.MiChroM method), 10  
 createSpringSpiral() (OpenMiChroM.ChromDynamics.MiChroM method), 10  
 CustomMiChroMTraining (class in OpenMiChroM.Optimization), 16  
**F**  
 FullTraining (class in OpenMiChroM.Optimization), 18  
**G**  
 get\_indices() (OpenMiChroM.Optimization.FullTraining method), 19  
 getBijsim() (OpenMiChroM.Optimization.CustomMiChroMTraining method), 17  
 getChromSeq() (OpenMiChroM.Optimization.CustomMiChroMTraining method), 17  
 getHiCexp() (OpenMiChroM.Optimization.CustomMiChroMTraining method), 18  
 getHiCexp() (OpenMiChroM.Optimization.FullTraining method), 19  
 getHiCSim() (OpenMiChroM.Optimization.CustomMiChroMTraining method), 18  
 getLamb() (OpenMiChroM.Optimization.AdamTraining method), 16  
 getLamb() (OpenMiChroM.Optimization.CustomMiChroMTraining method), 18  
 getLamb\_types() (OpenMiChroM.Optimization.CustomMiChroMTraining method), 18  
 getLambdas() (OpenMiChroM.Optimization.FullTraining method), 19  
 getlambfromfile() (OpenMiChroM.Optimization.CustomMiChroMTraining method), 18  
 getLoops() (OpenMiChroM.ChromDynamics.MiChroM method), 10  
 getPars() (OpenMiChroM.Optimization.AdamTraining method), 16  
 getPearson() (OpenMiChroM.Optimization.CustomMiChroMTraining method), 18  
 getPearson() (OpenMiChroM.Optimization.FullTraining method), 19  
 getPiPjsim\_types() (OpenMiChroM.Optimization.CustomMiChroMTraining method), 18  
 getPositions() (OpenMiChroM.ChromDynamics.MiChroM method), 10  
 getScaledData() (OpenMiChroM.ChromDynamics.MiChroM method), 10  
 getVelocities() (OpenMiChroM.ChromDynamics.MiChroM method), 10  
**I**  
 initPositions() (OpenMiChroM.ChromDynamics.MiChroM method), 11  
 initStorage() (OpenMiChroM.ChromDynamics.MiChroM method), 11  
 initStructure() (OpenMiChroM.ChromDynamics.MiChroM method), 11  
 initVelocities() (OpenMiChroM.ChromDynamics.MiChroM method), 12  
**L**  
 load() (OpenMiChroM.CndbTools.cndbTools method), 22  
 loadGRO() (OpenMiChroM.ChromDynamics.MiChroM method), 12  
 loadWDB() (OpenMiChroM.ChromDynamics.MiChroM method), 12  
 loadPDB() (OpenMiChroM.ChromDynamics.MiChroM method), 12  
 loadStructure() (OpenMiChroM.ChromDynamics.MiChroM method), 13  
**M**  
 MiChroM (class in OpenMiChroM.ChromDynamics), 5  
 module  
 OpenMiChroM.ChromDynamics, 5  
 OpenMiChroM.CndbTools, 20  
 OpenMiChroM.Optimization, 15  
**N**  
 ndb2cndb() (OpenMiChroM.CndbTools.cndbTools method), 22  
**O**  
 OpenMiChroM.ChromDynamics module, 5  
 OpenMiChroM.CndbTools module, 20  
 OpenMiChroM.Optimization module, 15

## P

`printForces()` (OpenMiChroM.ChromDynamics.MiChroM method), 13

`printHeader()` (OpenMiChroM.ChromDynamics.MiChroM method), 13

`printStats()` (OpenMiChroM.ChromDynamics.MiChroM method), 13

`probCalc()` (OpenMiChroM.Optimization.AdamTraining method), 16

`probCalc()` (OpenMiChroM.Optimization.FullTraining method), 19

`probCalculation_IC()` (OpenMiChroM.Optimization.CustomMiChroMTraining method), 18

`probCalculation_types()` (OpenMiChroM.Optimization.CustomMiChroMTraining method), 18

`setup()` (OpenMiChroM.ChromDynamics.MiChroM method), 15

## T

`traj2HiC()` (OpenMiChroM.CndbTools.cndbTools method), 22

## X

`xyz()` (OpenMiChroM.CndbTools.cndbTools method), 22

## R

`random_ChromSeq()` (OpenMiChroM.ChromDynamics.MiChroM method), 13

`randomizePositions()` (OpenMiChroM.ChromDynamics.MiChroM method), 13

`removeFlatBottomHarmonic()` (OpenMiChroM.ChromDynamics.MiChroM method), 13

`removeForce()` (OpenMiChroM.ChromDynamics.MiChroM method), 13

`runSimBlock()` (OpenMiChroM.ChromDynamics.MiChroM method), 13

## S

`saveFolder()` (OpenMiChroM.ChromDynamics.MiChroM method), 13

`saveLambdas()` (OpenMiChroM.Optimization.FullTraining method), 19

`saveStructure()` (OpenMiChroM.ChromDynamics.MiChroM method), 14

`setChains()` (OpenMiChroM.ChromDynamics.MiChroM method), 14

`setFibPosition()` (OpenMiChroM.ChromDynamics.MiChroM method), 14

`setPositions()` (OpenMiChroM.ChromDynamics.MiChroM method), 15