# Data Visualization

**General histogram function:**

hist (FileName, InitialTime, FinalTime, SpeciesName)

Description: This function enables users to obtain a histogram during a certain time period of selected species.

Parameters:

* FileName

Acceptable value: String (Must provide)

Description: It is the path to the ‘.dat’ file, which is usually named as ‘histogram\_complexes\_time.dat’, representing the histogram data to be analyzed.

Sample: FileName = ‘/Users/UserName/Documents/histogram\_complexes\_time.dat’

* InitialTime

Acceptable value: Float (Must provide)

Unit: s

Description: It is the initial time that users desire to examine and plot on the histogram. The acceptable range should not smaller than the starting time or exceed the ending time of simulation.

Sample: InitialTime = 0.5

Note: If the InitialTime and FinalTime both input as -1, the entire time range indicated in the reading(.dat) file will be taken as input.

* FinalTime

Acceptable value: Float (Must provide)

Unit: s

Description: It is the final time that users desire to examine and plot on the histogram. The acceptable range should not smaller than the value of InitialTime or exceed the ending time of simulation.

Sample: FinalTime = 1.0

Note: If the InitialTime and FinalTime both input as -1, the entire time range indicated in the reading(.dat) file will be taken as input.

* SpeciesName

Acceptable value: String (Must provide)

Description: It is the name of species that users want to examine, which should also be identical with the name written in the input (.inp and .mol) files.

Sample: SpeciesName = ‘clat’

**Function for plotting maximum number of monomers in single complex over time:**

max\_complex (FileName, InitialTime, FinalTime, SpeciesName)

Description: This function enables users to obtain a plot indicating maximum number of monomers in single complex molecule during a certain time period.

Parameters:

* FileName

Acceptable value: String (Must provide)

Description: It is the path to the ‘.dat’ file, which is usually named as ‘histogram\_complexes\_time.dat’, representing the histogram data to be analyzed.

Sample: FileName = ‘/Users/UserName/Documents/histogram\_complexes\_time.dat’

* InitialTime

Acceptable value: Float (Must provide)

Unit: s

Description: It is the initial time that users desire to examine and plot on the histogram. The acceptable range should not smaller than the starting time or exceed the ending time of simulation.

Sample: InitialTime = 0.5

Note: If the InitialTime and FinalTime both input as -1, the entire time range indicated in the reading(.dat) file will be taken as input.

* FinalTime

Acceptable value: Float (Must provide)

Unit: s

Description: It is the final time that users desire to examine and plot on the histogram. The acceptable range should not smaller than the value of InitialTime or exceed the ending time of simulation.

Sample: FinalTime = 1.0

Note: If the InitialTime and FinalTime both input as -1, the entire time range indicated in the reading(.dat) file will be taken as input.

* SpeciesName

Acceptable value: String (Must provide)

Description: It is the name of species that users want to examine, which should also be identical with the name written in the input (.inp and .mol) files.

Sample: SpeciesName = ‘clat’

**Function for plotting mean number of monomers in single complex over time:**

mean\_complex (FileName, InitialTime, FinalTime, SpeciesName, ExcludeNum)

Description: This function enables users to obtain a plot indicating mean number of monomers in single complex molecule during a certain time period.

Parameters:

* FileName

Acceptable value: String (Must provide)

Description: It is the path to the ‘.dat’ file, which is usually named as ‘histogram\_complexes\_time.dat’, representing the histogram data to be analyzed.

Sample: FileName = ‘/Users/UserName/Documents/histogram\_complexes\_time.dat’

* InitialTime

Acceptable value: Float (Must provide)

Unit: s

Description: It is the initial time that users desire to examine and plot on the histogram. The acceptable range should not smaller than the starting time or exceed the ending time of simulation.

Sample: InitialTime = 0.5

Note: If the InitialTime and FinalTime both input as -1, the entire time range indicated in the reading(.dat) file will be taken as input.

* FinalTime

Acceptable value: Float (Must provide)

Unit: s

Description: It is the final time that users desire to examine and plot on the histogram. The acceptable range should not smaller than the value of InitialTime or exceed the ending time of simulation.

Sample: FinalTime = 1.0

Note: If the InitialTime and FinalTime both input as -1, the entire time range indicated in the reading(.dat) file will be taken as input.

* SpeciesName

Acceptable value: String (Must provide)

Description: It is the name of species that users want to examine, which should also be identical with the name written in the input (.inp and .mol) files.

Sample: SpeciesName = ‘clat’

* ExcludeNum

Acceptable value: Int (Optional)

Default: 0

Description: In the generated plot, the number of monomers in the complex that are no larger than this number will be excluded and will not be considered into the average calculation.

Sample: ExcludeNum = 1 (which means the monomer will be excluded)

**Function for converting ‘histogram\_complexes\_time.dat’ file to a data frame:** hist\_to\_df(FileName, SpeciesNameList)

Description: This function enables users to convert the raw .dat file to a data frame in python pandas package for multi-species system. Each column in the data frame includes the simulation time and selected occurrences of species during the simulation; each row is separated by a different simulation time.

Parameters:

* FileName

Acceptable value: String (Must provide)

Description: It is the path to the ‘.dat’ file, which is usually named as ‘histogram\_complexes\_time.dat’, representing the histogram data to be analyzed.

Sample: FileName = ‘/Users/UserName/Documents/histogram\_complexes\_time.dat’

* SpeciesNameList

Acceptable value: List with string elements (Optional)

Default: [] (empty list)

Description: The desired name of species should be contained inside this list. If the list is empty, all occurrences of species during the simulation is contained in the data frame.

Sample: [‘clat: 4.’, ‘clat: 5.’, ‘clat: 6.’] (which means tetramer, pentamer, hexamer will be counted in the data frame)

**Function for creating 3D histogram (histogram with time axis):**

hist\_3d\_time (FileName, InitialTime, FinalTime, SpeciesName, TimeBins)

Description: This function enables users to generate 3D histogram representing the number of monomers in single complex as simulation time develops. The x-axis is the number of monomers, y-axis is the averaged time and z-axis is the relative occurrence probabilities.

Parameters:

* FileName

Acceptable value: String (Must provide)

Description: It is the path to the ‘.dat’ file, which is usually named as ‘histogram\_complexes\_time.dat’, representing the histogram data to be analyzed.

Sample: FileName = ‘/Users/UserName/Documents/histogram\_complexes\_time.dat’

* InitialTime

Acceptable value: Float (Must provide)

Unit: s

Description: It is the initial time that users desire to examine and plot on the histogram. The acceptable range should not smaller than the starting time or exceed the ending time of simulation.

Sample: InitialTime = 0.5

* FinalTime

Acceptable value: Float (Must provide)

Unit: s

Description: It is the final time that users desire to examine and plot on the histogram. The acceptable range should not smaller than the value of InitialTime or exceed the ending time of simulation.

Sample: FinalTime = 1.0

* SpeciesName

Acceptable value: String (Must provide)

Description: It is the name of species that users want to examine, which should also be identical with the name written in the input (.inp and .mol) files.

Sample: SpeciesName = ‘clat’

* TimeBins

Acceptable value: Int (Must provide)

Description: It is the number of bins that users want to divide the selected time period into. The value should be positive integer.

**Function for creating 2D heatmap showing distribution of N-mers vs. time (low contrast option):**

hist\_time\_heatmap (FileName, InitialTime, FinalTime, SpeciesName, TimeBins, ShowNum)

Description: This function enables users to generate 2D histogram of numerical distribution of different N-mers vs. time. The x-axis is the distribution of number of monomers in single complex and y-axis is the time period. The color in each box indicates the number of corresponding N-mers when corresponding time period is reached.

Parameters:

* FileName

Acceptable value: String (Must provide)

Description: It is the path to the ‘.dat’ file, which is usually named as ‘histogram\_complexes\_time.dat’, representing the histogram data to be analyzed.

Sample: FileName = ‘/Users/UserName/Documents/histogram\_complexes\_time.dat’

* InitialTime

Acceptable value: Float (Must provide)

Unit: s

Description: It is the initial time that users desire to examine and plot on the histogram. The acceptable range should not smaller than the starting time or exceed the ending time of simulation.

Sample: InitialTime = 0.5

* FinalTime

Acceptable value: Float (Must provide)

Unit: s

Description: It is the final time that users desire to examine and plot on the histogram. The acceptable range should not smaller than the value of InitialTime or exceed the ending time of simulation.

Sample: FinalTime = 1.0

* SpeciesName

Acceptable value: String (Must provide)

Description: It is the name of species that users want to examine, which should also be identical with the name written in the input (.inp and .mol) files.

Sample: SpeciesName = ‘clat’

* TimeBins

Acceptable value: Int (Must provide)

Description: It is the number of bins that users want to divide the selected time period into. The value should be positive integer.

* ShowNum

Acceptable value: True or False (Optional)

Default: True

Description: If True, the corresponding number will be shown in the center of each box; if false, there will be no number shown.

**Function for creating 2D heatmap showing number of total count of monomers inside N-mers vs. time:**

hist\_time\_heatmap\_mono\_count(FileName, InitialTime, FinalTime, SpeciesName, TimeBins, ShowNum)

Description: This function enables users to generate 2D histogram of total count of monomers in different N-mers vs. time. The x-axis is the number of monomers in single complex and y-axis is the time period. The color in each box indicates the total number of corresponding monomers in N-mers when corresponding time period is reached.

Parameters:

* FileName

Acceptable value: String (Must provide)

Description: It is the path to the ‘.dat’ file, which is usually named as ‘histogram\_complexes\_time.dat’, representing the histogram data to be analyzed.

Sample: FileName = ‘/Users/UserName/Documents/histogram\_complexes\_time.dat’

* InitialTime

Acceptable value: Float (Must provide)

Unit: s

Description: It is the initial time that users desire to examine and plot on the histogram. The acceptable range should not smaller than the starting time or exceed the ending time of simulation.

Sample: InitialTime = 0.5

* FinalTime

Acceptable value: Float (Must provide)

Unit: s

Description: It is the final time that users desire to examine and plot on the histogram. The acceptable range should not smaller than the value of InitialTime or exceed the ending time of simulation.

Sample: FinalTime = 1.0

* SpeciesName

Acceptable value: String (Must provide)

Description: It is the name of species that users want to examine, which should also be identical with the name written in the input (.inp and .mol) files.

Sample: SpeciesName = ‘clat’

* TimeBins

Acceptable value: Int (Must provide)

Description: It is the number of bins that users want to divide the selected time period into. The value should be positive integer.

* ShowNum

Acceptable value: True or False (Optional)

Default: True

Description: If True, the corresponding number will be shown in the center of each box; if false, there will be no number shown.

**Function for creating 2D heatmap showing fractions of monomers forming N-mers vs. time:**

hist\_time\_heatmap\_fraction (FileName, InitialTime, FinalTime, SpeciesName, TimeBins, ShowNum)

Description: This function enables users to generate 2D histogram of fractions of monomers forming different N-mers vs. time. The x-axis is the number of monomers in single complex and y-axis is the time period. The color in each box indicates the fraction of monomers forming corresponding N-mers when corresponding time period is reached.

Parameters:

* FileName

Acceptable value: String (Must provide)

Description: It is the path to the ‘.dat’ file, which is usually named as ‘histogram\_complexes\_time.dat’, representing the histogram data to be analyzed.

Sample: FileName = ‘/Users/UserName/Documents/histogram\_complexes\_time.dat’

* InitialTime

Acceptable value: Float (Must provide)

Unit: s

Description: It is the initial time that users desire to examine and plot on the histogram. The acceptable range should not smaller than the starting time or exceed the ending time of simulation.

Sample: InitialTime = 0.5

* FinalTime

Acceptable value: Float (Must provide)

Unit: s

Description: It is the final time that users desire to examine and plot on the histogram. The acceptable range should not smaller than the value of InitialTime or exceed the ending time of simulation.

Sample: FinalTime = 1.0

* SpeciesName

Acceptable value: String (Must provide)

Description: It is the name of species that users want to examine, which should also be identical with the name written in the input (.inp and .mol) files.

Sample: SpeciesName = ‘clat’

* TimeBins

Acceptable value: Int (Must provide)

Description: It is the number of bins that users want to divide the selected time period into. The value should be positive integer.

* ShowNum

Acceptable value: True or False (Optional)

Default: True

Description: If True, the corresponding number will be shown in the center of each box; if false, there will be no number shown.

# Platonic Solid Self-assembly Input File Writing

Platonic solid self-assembly include 10 models, so that 10 separate functions are needed, which are shown in the following table:

|  |  |  |
| --- | --- | --- |
| Platonic Solid | Center-of-Mass Position | Name of Function |
| Tetrahedron (4-face) | Each Face | tetr\_face(radius, sigma) |
| Tetrahedron (4-face) | Each Vertex | tetr\_vert(radius, sigma) |
| Cube (6-face) | Each Face | cube\_face(radius, sigma) |
| Cube (6-face) | Each Vertex | cube\_vert(radius, sigma) |
| Octahedron (8-face) | Each Face | octa\_face(radius, sigma) |
| Octahedron (8-face) | Each Vertex | octa\_vert(radius, sigma) |
| Dodecahedron (12-face) | Each Face | dode\_face(radius, sigma) |
| Dodecahedron (12-face) | Each Vertex | dode\_vert(radius, sigma) |
| Icosahedron (20-face) | Each Face | icos\_face(radius, sigma) |
| Icosahedron (20-face) | Each Vertex | icos\_vert(radius, sigma) |

Description: This function enable users to generate NERDSS input files (.inp and .mol files) for Platonic solid self-assembly system.

Parameters:

* radius

Acceptable value: Float (Must provide)

Unit: nm

Description: It is the radius of the Platonic solid, which is defined by the distance from the center of Platonic solid to each vertex.

Sample: radius = 40.0

* sigma

Acceptable value: Float (Must provide)

Unit: nm

Description: It is the distance of each interface when reaction takes place.

Sample: sigma = 1.0

# Locating position for certain size of complexes by PDB/restart file

locate\_position\_PDB(FileNamePdb, NumList, FileNameInp, BufferRatio)

Description: This function enables users to locate specific complexes of certain size from a PDB file after simulation. The result will be output as a separated file named “output\_file.pdb” containing only the desired complex.

Note: Reading only the PDB file slows down the function compared to reading the 'restart.dat' file, because the function needs to calculate the distance between all reactive atoms that can be reacted based on the reaction information to determine whether they are bound or not. Therefore, this function is universal but runs slowly in time.

Parameters:

* FileNamePdb

Acceptable value: String (Must provide)

Description: It is the path to the PDB file, which is usually the last frame of simulation.

Sample: FileNamePdb = ‘/Users/UserName/Documents/999999.pdb’

* Numlist

Acceptable value: List with Int elements (Must provide)

Description: It represents the number of individual monomers in the complex that the user needs to locate, and the order of the monomers in the list is determined by the order in which they appear in the PDB file.

Sample: NumList = [12] (This represents the complexes is consist of only one kind of monomer and the desired complexes is a dodecamer, also known as 12-mer)

* FileNameInp

Acceptable value: String (Must provide)

Description: It is the path to the ‘.inp’ file, which usually stores the reaction information.

Sample: FileNameInp = ‘/Users/UserName/Documents/parms.inp’

* BufferRatio

Acceptable value: Float (Optional)

Default: 0.01

Description: When the sigma value is within the interval of 1.0 plus or minus the ratio, the two reaction interfaces can be considered as bonded.

locate\_position\_restart(FileNamePdb, NumList, FileNameRestart)

Description: This function enables users to locate specific complexes of certain size from a PDB file along with ‘restart.dat’ file after simulation. The result will be output as a separated file named “output\_file.pdb” containing only the desired complex.

Note: The advantage of reading the 'restart.dat' file is that the file directly stores the binding information of each complex in the system and can be used directly, so the function runs faster; however, the function is not universal, if the 'restart.dat ' file's write logic changes, then this function will no longer work.

Parameters:

* FileNamePdb

Acceptable value: String (Must provide)

Description: It is the path to the PDB file, which is usually the last frame of simulation.

Sample: FileNamePdb = ‘/Users/UserName/Documents/999999.pdb’

* Numlist

Acceptable value: List with Int elements (Must provide)

Description: It represents the number of individual monomers in the complex that the user needs to locate, and the order of the monomers in the list is determined by the order in which they appear in the PDB file.

Sample: NumList = [12] (This represents the complexes is consist of only one kind of monomer and the desired complexes is a dodecamer, also known as 12-mer)

* FileNameRestart

Acceptable value: String (Optional)

Default: 'restart.dat'

Description: It is the path to the ‘restart.dat’ file.