

# Cassandra vSurfactant + PostProcessing Documentation

Andrew P. Santos (apsantos@princeton.edu)

September 12, 2017

## Contents

<b>1</b>	<b>Cassandra vSurfactant User Manual</b>	<b>2</b>
1.1	Pair Insertions . . . . .	2
1.2	Interaction tables . . . . .	3
1.3	Added property output . . . . .	4
1.4	Clustering . . . . .	5
1.5	Oligomer cut-off . . . . .	6
1.6	Counterion Degree of Association . . . . .	7
1.7	Histogram file writing . . . . .	7
<b>2</b>	<b>Cassandra PostProcessing</b>	<b>7</b>
2.1	Reading configurations . . . . .	7
2.2	Excluded Volume . . . . .	8

# 1 Cassandra vSurfactant User Manual

## 1.1 Pair Insertions

```
# Prob_Insertion
Real(1)
insertion method
reservoir pair
Integer(i,1) Real(i,2)

# Chemical_Potential_Info
Real(1) ...
pair
Integer(1) Integer(2) Real(3)
.
.
.
```

Defines pairs of species which should be inserted.  
For example,

```
# Chemical_Potential_Info
-103.5 -103.5 -103.5
pair
1 2 -103.5
```

```
# Prob_Insertion
0.225
insertion method
reservoir pair
2 1.0
insertion method
reservoir pair
1 1.0
insertion method
none
```

Would insert pairs of species 1 and 2 (for example an ionic surfactant and its coion) with a chemical potential of -103.5 kJ/mol 100 % of the insertion moves and would not insert the 3rd species (for example water), regardless of the value reported.  
Another example,

```
# Chemical_Potential_Info
-103.5 -103.5 -103.5 -103.5
pair
1 2 -103.5
2 3 -30.5
```

```

# Prob_Insertion
0.225
insertion method
reservoir pair
2 0.5
insertion method
reservoir pair
1 0.5
3 0.5
insertion method
none

```

Would insert pairs of species 1 and 2 (for example an ionic surfactant and its counterion) with a chemical potential of -103.5 kJ/mol 50 % of the insertion moves, species 2 and 3 (for example the counterion and its coion) 50 % of the insertion moves and would not insert the 4th species (for example water), regardless of the value reported.

Another example,

```

# Chemical_Potential_Info
-103.5 -103.5 -103.5 -103.5
pair
1 2 -103.5
2 3 -30.5

```

```

# Prob_Insertion
0.225
insertion method
reservoir pair
2 0.4
insertion method
reservoir pair
1 0.4
3 0.4
insertion method
reservoir

```

Would insert pairs of species 1 and 2 (for example an ionic surfactant and its counterion) with a chemical potential of -103.5 kJ/mol 40 % of the insertion moves, species 2 and 3 (for example the counterion and its coion) 40 % of the insertion moves and would insert the 4th species (for example water) 20 % of the insertion moves, regardless of the value reported.

## 1.2 Interaction tables

```

# Intra_Scaling
table Character(i,2)

```

```
# Mixing_Rule
table Character(1)
```

Gives Information on how inter-molecular and intra-molecular interactions are defined by the user in tables. `Character(i,2)` gives the intra-molecular interactions for every species in the simulation, while `Character(1)` under `# Mixing_Rule` tells the inter-molecular between all species.

For example,

```
# Intra_Scaling
table DS.mix
table Na.mix
# Mixing_Rule
table NaDS.mix
```

The format for the intra-molecular table is as follows”:

```
# Atom_Types
Character(i,1) Integer(i)
# Done_Atom_Types

# Mixing_Values
Integer(i,1) Integer(i,2) {Character(i,3) Real(i,4) ...} ...
# Done_Mixing_Values

# VDW_Style
Integer(i,1) Integer(i,2) Character(i,3) Real(i,4)
# Done_VDW_Style
```

Under `# Atom_Types` `Character(i,1)` is the atom name, as it appears in the corresponding mcf, and `Integer(i)` is the atom type. Values are assumed to be the same as those for inter-molecular interactions with zero 1-4 interactions. If a pair within a species molecule is different define so in `# Mixing_Values`, where `Integer(i,1)` is the atom type one and `Integer(i,2)` is the other atom type, `Character(i,3)` is the interaction type name and `Integer(i,3)` and beyond are the relevant parameters for the interaction type. Multiple interaction types can be defined in one line. `# VDW_Styles` is optional and tells if the style, and cutoff values are different for different pairs.

The format for the inter-molecular table is essentially the same.

### 1.3 Added property output

```
# Property_Info 1
Noligomers
NclustersOlig
NclustersMicelle
MicelleSize
```

Degree\_Association  
Excluded\_Volume

These are new keywords for properties written to “prp”.

- Noligomers is the number of molecules in oligomeric clusters.
- NclustersOlig\* is the number of oligomeric clusters.
- NclustersMicelle\* is the number of micellar clusters (sizes greater than or equal to Oligomer\_cutoff).
- MicelleSize\* is the average size of the micelles at that time.
- Degree\_Association\* is the number of associated species (usually ions) that are bound to the clustered species (usually surfactants).
- Excluded\_Volume\* Excluded volume.

\* - only available in PostProcessing

## 1.4 Clustering

```
# Clustering
count no_criteria
type no_type_parameters
type_parameters
move no_criteria
type no_type_parameters
type_parameters
exvol no_criteria
type no_type_parameters
type_parameters
```

These keywords giver information on how to determine if clustering is occurring.  
Form \*xvg r example,

```
# Clustering
count 1
skh
5.0 6.25 7.0 ME CC
0.0 0.0 0.0 Na
0 0 0 ME CC
move 0
skh
5.0 6.25 7.0 ME CC
0.0 0.0 0.0 Na
0 0 0 ME CC
exvol 0
```

Would have 5000 energy bins with 50.0 kJ/mol spacing.  
For example,

```
# Clustering
count 1
com
5.0 0.0 0.0
0.0 0.0 0.0
0 0 0
move 1
com
5.0 0.0 0.0
0.0 0.0 0.0
0 0 0
exvol 1
com
0.0 0.0 7.0
0.0 0.0 0.0
7.0 0.0 0.0
```

Would have 5000 energy bins with 50.0 kJ/mol spacing.  
For example,

```
# Clustering
count 1
type 1
1 ME 1 ME 12
move 0
exvol 1
type 1
1 ME 2 ME 12
```

Would have 5000 energy bins with 50.0 kJ/mol spacing.

## 1.5 Oligomer cut-off

```
# Oligomer_Cutoff
Integer(1) Integer(2) ...
```

This keyword lets Cassandra know to cutoff value for cluster size which determines a larger (micellar) or small (oligomeric) cluster for each species. If you are not interested in studying the cluster behavior for a certain species set the value to 0. If you want to study all cluster sizes set it equal to 1. This value typically corresponds to the low-cluster-size local minimum in the cluster size distribution.

For example,

```
# Oligomer_Cutoff
14 0 14
```

## 1.6 Counterion Degree of Association

```
# Degree_Association
Real(1)
Character(i)
```

This keyword lets Cassandra know to calculate and update the degree of counterion association, which is  $\alpha = \sum_i^{N_{\text{clusters}}} \frac{N_{\text{ion}}^{\text{associated}}}{M(i)}$  and is updated to the prp file. **Real(1)** is the distance criteria and **Character(i)** is the atom name for each species which you want to determine is associated, use “NONE” if a species should be excluded. Cassandra assumes that un-clustered species, as defined in **# Clustering\_Info** is the associated species.

For example,

```
# Degree_Association
4.3
HG
Na
NONE
```

Would insert species 3 of 3 into snapshots with an energy criteria of 1000000 kJ/mol.

## 1.7 Histogram file writing

```
# Histogram_Info
Integer(1) Real(2)
```

This keyword lets Cassandra know to calculate and update the histogram of number of particles and energies with **Integer(1)** energy bins that are **Real(2)** kJ/mol wide. As of now, Cassandra assumes that there is only one chemical potential (one insertable component or pair insertionable). The generated file is designed to be interfaced with the **entropy** program suite. For two chemical potential systems one can easily modify the “prp” file with the  $N_i$  and  $U$  data.

For example,

```
# Histogram_Info
5000 50.0
```

Would have 5000 energy bins with 50.0 kJ/mol spacing.

## 2 Cassandra PostProcessing

### 2.1 Reading configurations

#### 2.1.1 xyz

```
# Start_Type
read_xyz
xyz_file
```

Read xyz file written from Cassandra with the following format.

```
<no_atoms>
```

```
<name> <x_pos> <y_pos> <z_pos> <species> <molecule>
```

```
.  
.   
.
```

which can be post-processed if written from a program other than this version of Cassandra.  
For example,

```
# Start_Type  
read_xyz  
prod2_pp.xyz
```

### 2.1.2 dcd

```
# Start_Type  
read_dcd  
xyz_file dcd_file
```

Read dcd written from hoomd or LAMMPS.

### 2.1.3 xtc

```
# Start_Type  
read_xtc  
gro_file ndx_file xtc_file
```

Read Gromacs xtc files. Library installation is necessary. See Gromacs documentation for descriptions of the different files.

For example,

```
# Start_Type  
read_xtc  
prod2.gro n1080_pp.ndx prod2.xtc
```

## 2.2 Excluded Volume

```
# Excluded_Volume  
Integer(1) Integer(2) ...  
method method_parameter
```

This keyword lets Cassandra know to calculate and update the excluded volume by clusters by inserting a species, that which corresponds to the non-zero **Integer** value, **Integer** number of times. The ratio of insertions which are “excluded” is written out to the “prp” file. An insertion is included if one of the following occurs: (i) an insertion becomes part of a cluster,



as defined in the `# Clustering_Info` section, (ii) there is an overlap based on `# Rcutoff_Low` and (iii), if specified, the energy change of the insertion exceeds that which is specified. For example,

```
# Excluded_Volume
0 0 400
energy 1000000
```

would insert species 3 of 3 with 400 different configurations using the fragment library into snapshots with an energy criteria of 1000000 kJ/mol. Or alternatively,

```
# Excluded_Volume
0 0 400
distance
```

would insert species 3 of 3 with 400 different configurations using the fragment library into snapshots with a distance criteria using `# Rcutoff_Low`.

### 3 Cassandra virial\_MC

The same as the rest of the simulations, but with this information

#### 3.1 2nd virial calculation

```
# Virial_Info
Real(1) Real(2) Real(3)
Integer(1) Integer(2)
Integer(1,1) Integer(2,2)
Integer(1,1) Integer(2,2)
```

The first line tells the program to calculate the energy at distances ranging from `Real(1)` to `Real(3)` with spacings of `Real(3)`. The second line tells which 2 species to calculate, can be 1 and 1. The 3rd line tells the number of conformations, from the fragment library of the first and second species should be used at each separation. The final line tells the number of rotations of those conformations to do at each separation.

For example,

```
# Virial_Info
1 0.1 40
1 2
10 20
30 40
```

would calculate the 2nd virial coefficient between species 1 and 2 using 10 conformations of species 1, 20 conformations of species 2, 30 rotations of each species 1 conformation and 40 rotations of each species 2 conformation, for separations from 1.0 Å to 40.0 Å with 0.1 Å spacings.