A Brief Guide to All-Atom/Coarse Grain Simulations 1.1

Contents

1. Introduction

All-atom/coarse grain or AACG representations are used for modeling problems where the detailed behavior for one part of the system necessitates retaining an all-atom (AA) representation while the rest of the system does not and may be represented using a simplified or coarse-grain (CG) model to speed up the calculations by reducing the number of interactions that need to be calculated. In addition, the dynamics for many processes in the CG environment are often inherently faster, in part due to the softer CG potentials, compounding the reduction in CPU time. Of course, this speed up comes at the price of using a more approximate representation (CG vs AA).

Our AACG implementation supports an AA representation for peptides while the environment, comprising the water and lipids, if present, are represented using a CG model. A relatively fine-grain mapping for water, in which one CG water particle represents a single water molecule permits a more detailed representation of the interaction between water and the protein than is typical with CG models. Other components such as the lipids or the cosolvent molecules use a coarser representation in which roughly 3 heavy atoms and any hydrogen atoms bonded to them are represented by a single site. This AACG implementation typically uses between 4 and 5 times less CPU time than the corresponding AA system to simulate a given amount of time. In addition, most diffusion rates through the CG portion of the system are more than a factor of 5 faster than in the corresponding AA system.[1] So, depending on whether the behaviors of interest depend on faster evolution within the CG portion of the system, AACG simulations use between roughly ¼ and 1/25 the CPU time of the corresponding AA system.

While AACG calculations must be run from the command-line, the results may be viewed and analysed in Maestro. All AACG simulations use Desmond and are constructed from AA systems. We support two types of AACG calculations:

1. NVT AACG simulations of peptides in water or with a POPC membrane in water.
2. AACG pocket finder calculations in which mixtures of modified coarse grain solvent molecules are used to identify and induce ligand binding locations including allosteric and/or cryptic pockets.

Both of these calculations are described in this guide.

[1] Protein dynamics also seems faster however the speedup has not been well-characterized and varies considerably.

2. AACG simulations

AACG simulations are run using Desmond in the NVT ensemble. No tools are provided to directly set up AACG simulations. One should construct an AA system using the system_builder and then equilibrate it using an NPT Desmond simulation to relax the system and attain an appropriate volume.[2] This AA system is then converted into an AACG system. After conversion it can be useful to energy minimize the system prior to starting an AACG simulation.

2.1 Capabilities and limitations

AACG technology supports simulating peptides, including oligopeptides, globular proteins and GPCRs, using an all atom representation in a coarse grained environment that can be composed of water, ions and a membrane. For more details please see Table 2.1 below.

Temperature
CG and thus AACG potentials incorporate aspects of the system's free energy. As a result they may not be reliable under simulation conditions distinctly different from those used to construct them. The AACG potentials were parameterized for 300K. While it is expected that simulations within the range 275-325K can be usefully performed, AACG parameters have not been validated at temperatures other than 300K.

Bulk solution at a preset volume
The AACG potentials on their own will not give an appropriate surface area for interfaces or system volume. AACG simulations are run in the NVT ensemble and tend to have a small positive pressure so simulations of liquid-gas or liquid-vacuum interfaces should be avoided.

[2] See the Desmond User Manual for more information.

Table 2.1 AACG force field coverage.

Type of component	Specifics	Additional information
Peptide	The 20 common amino acids	Histidine protonation states (HIS, HIE, HIP), and protonated D and E (ASH and GLH) are also supported.
		Phosphorylated S, T and Y
	Additional amino acids	Selenomethionine
	Termini	ACE, -NH2, -NH3+, NMA, Proline+
Cofactors*	ADP, ATP, GDP, GTP	
Membrane	POPC	16 sites
Water	1 site	
Ions*	Na^+, Cl^-, Ca^{2+}, Mg^{2+}, Mn^{2+}, Zn^{2+}	

*Cofactor and ion parameters are being refined.

2.2 AACG commands

Conversion from AA to AACG
The following command converts an AA system into an AACG system:

$SCHRODINGER/run -FROM aacg aa2mr.py <AA_input.cms> <AACG_output.cms>

Both the input and output files should have cms extensions. This command takes about 5 minutes to run for a system containing 50,000 atoms on a typical Linux desktop. If unsupported atoms are present an error message will appear and conversion will be unsuccessful.

Running AACG simulations
The multisim scripts, with an .msj extension, for AACG simulations are available in $SCHRODINGER/aacg-vXXXXX/multisim, namely:
mr_minimization.msj - to minimize the energy

mr_minimization_md.msj - to minimize the energy and then simulate the system.
mr_md.msj - to simulate the system
If any of these are provided with an AA input cms file, they will automatically attempt to convert it into an AACG system before proceeding.

The md versions of these scripts can be modified to:

- adjust the simulation duration, change 50000 (ps) in the setting
  time = 50000
  to the desired time.
- adjust the temperature, change 300.0 in the setting
  temperature = [
  [300.0 0]
  ]
  to the desired temperature.
- turn off randomization of the velocity at the start of the simulation (can be useful when a simulation is being continued) change 0.0 in:
  randomize_velocity = {
  first=0.0
  .
  ,
  ,
  }
  to inf

Note: Glue should not be turned on in AACG simulations nor should the cutoff distance for interactions be modified.

These scripts can be run on CPU based machines with the command:

$SCHRODINGER/utilities/multisim -JOBNAME <jobname> -HOST <CPU_queue_name> -maxjob 1 -cpu <N_CPU> -m <your_msj_file.msj> -description 'My AACG job' <input_cms_file> -mode umbrella -o <output_cmsfile>

The corresponding GPGPU command may vary. On a cluster which has CPU and GPGPU queues the command can take the form:

$SCHRODINGER/utilities/multisim -JOBNAME <jobname> -HOST <CPU_queue_name> -SUBHOST <GPGPU_queue_name> -maxjob 1 -cpu 1 -m <your_msj_file.msj> -description 'My AACG job' <input_cms_file> -set 'stage[1].set_family.md.jlaunch_opt=["-gpu"] -o <output_cmsfile>

2.3 Visualization and analysis

Maestro can display AACG configurations and play AACG trajectories in much the same way as Desmond AA trajectories. Read the <jobname>-out.cms file produced at the end of the AACG simulation into Maestro. In the project table there will be a new entry named <jobname>-out that has a 'T' button in the Title column. Clicking on the 'T' opens up a panel for viewing the trajectory. See the Desmond User Manual for more information. Some simple visual analysis tools developed for AA systems also work for AACG systems within Maestro. For instance, Maestro's tools for measuring distances, angles and dihedral angles as well as dynamically displaying hydrogen bonds all work in AACG systems. See the Maestro manual for more information.

Desmond's Simulation Interactions Diagram tool, which is very useful for characterizing protein structure and dynamics during a simulation, is compatible with AACG systems. In addition, specific geometric but not energy-based measurements in Simulation Event Analysis also work with AACG systems. For more information on these two tools please see the Desmond User Manual.

2.4 AACG simulation licensing requirements

An AACG license is required to convert an AA system into an AACG system. Desmond simulations of AACG systems require both Desmond licenses and an AACG license. AACG simulations only check for the existence of an AACG license so one license is enough for multiple simultaneous use provided that there are sufficient Desmond licenses.

3. AACG pocket finder calculations

The shape of many proteins can vary, altering binding site shape and volume to the extent that binding sites can appear or disappear in different protein conformations. As a result, one or several 3 dimensional representations of a protein structure may not present all of the protein's ligand binding sites. Being able to identify ligand binding sites not present in the starting structure can be very advantageous in developing intellectual property related to the target protein. The goal of the AACG pocket finder is to reliably identify druggable binding sites, including the so-called cryptic binding pockets which in the absence of a bound ligand may appear very infrequently. The AACG pocket finder identifies binding sites by simulating the protein in a mixture of cosolvent molecules whose interactions with each other and with the protein have been modified to induce protein shape changes in relatively short simulations. The procedure uses SiteMap to detect ligand binding sites that are at least transiently present in the simulations. The binding site information is collectively analysed using a procedure calibrated to identify known binding sites while yielding relatively few false positives.

AACG pocket finder technology aggressively perturbs the proteins so approximate binding site locations are identified. In tests, the identified binding sites can be displaced from the known binding site location and may have a different shape. Usually, the pockets identified are not of high enough quality for use in structure-based drug design.

3.1 Limitations

AACG pocket finder has similar limitations to AACG as a whole. The models and procedures have not been adapted for membranes so only globular proteins should be used for pocket finder jobs. While it is possible to adjust the temperature from the default (300K) for these jobs this is discouraged as the procedure is highly tuned.

3.2 Running AACG pocket finder calculations

AACG pocket finder jobs are run in three separate parts from the command-line:

1. System setup and relaxation
2. Sampling
3. Analysis

3.2.1 System setup and relaxation

The system setup and relaxation procedure constructs and relaxes an all-atom cosolvent system which is then converted into a mixed resolution cosolvent system with enhanced interactions and relaxed. This procedure requires a Maestro file containing just the protein (and cofactors as well as tightly bound ions, if present) of interest. The protein should be adequately prepared, ideally using the Protein Preparation Wizard. See the Protein Preparation Guide for more information.

To set up and relax a new system on a CPU-based computer use the command:

$SCHRODINGER/utilities/multisim -JOBNAME <jobname> -HOST <CPU_queue_name> -maxjob 1 -cpu <N_CPU> -m cosolvent_setup.msj -description 'My cosolvent setup job' -mode umbrella <input_protein_structure.mae> -LOCAL

The cosolvent_setup.msj is available in $SCHRODINGER/aacg-vXXXXX/multism. N_CPU is the number of CPUs to use for the job. This procedure takes about about ½ day on 8 processors for a typical compact globular protein containing 5,000 atoms. Larger or more extended proteins can require significantly more time to prepare.

Running on a GPGPU can considerably reduce the turnaround time for these calculations, however the corresponding GPGPU command may vary. On a cluster which has CPU and GPGPU queues the command can take the form:

$SCHRODINGER/utilities/multisim -JOBNAME <jobname> -HOST <CPU_queue_name> -SUBHOST <GPGPU_queue_name> -maxjob 1 -cpu 1 -m cosolvent_setup.msj -description 'My cosolvent setup job' -set 'stage[1].set_family.md.jlaunch_opt=["-gpu"]' <input_protein_structure.mae> -LOCAL

3.2.2 Sampling

By default, AACG pocket finder jobs sample the protein's conformations using 5 separate 24ns simulations. To run these jobs on a CPU use the command:

$SCHRODINGER/utilities/multisim -cpu <number_of_processors> -maxjob <number_of_separate_simulations> -JOBNAME <jobname> -m AACG_pocket_finder_multisimulation_jobs.msj -c AACG_pocket_finder_simulation.cfg -HOST <CPU_queue_name> -description 'My cosolvent sampling job' <input_file_name.cms> -LOCAL

where <input_file_name.cms> is the cms file produced using the cosolvent_setup.msj file. A copy of AACG_pocket_finder_multisimulation_jobs.msj is available in $SCHRODINGER/aacg-vXXXXX/multisim. This command will run <number_of_separate_simulations> simulations each on <number_of_processors> processors sequentially or simultaneously if sufficient processors are available. Each simulation will typically run for about 5 days on 8 modern CPUs.

To run these jobs on a GPGPU use the command:

$SCHRODINGER/utilities/multisim -cpu 1 -maxjob <number_of_separate_simulations> -JOBNAME <jobname> -m AACG_pocket_finder_multisimulation_jobs.msj -c

AACG_pocket_finder_simulation.cfg -HOST <CPU_queue_name> -SUBHOST <GPGPU_queue_name> -LOCAL -set 'stage[1].set_family.md.jlaunch_opt=["-gpu"]'

Each simulation will typically run for about 2-4 hours on a GPGPU.

3.2.3 Analysis

The analysis of the trajectories consists of a number of steps including:

1. Running SiteMap on many frames from the trajectories (default is approximately 60 per simulation)
2. Clustering the SiteMap results
3. Calculating the average cosolvent densities
4. Associating clusters that are in similar locations into Groups
5. Classifying the Groups as Primary, Secondary, Tertiary or other.

These analyses are conducted in the same directory as used to launch the sampling jobs, using the script all_analysis.msj (available in $SCHRODINGER/aacg-vXXXXX/multisim) using the command:

$SCHRODINGER/utilities/multisim -HOST <CPU_queue_name>:<number_of_processors> -maxjob <number_of_processors> -JOBNAME <jobname> -m all_analysis.msj <jobname>_reference_structure.mae -LOCAL

where:

- <jobname> should be the same <jobname> as used for the sampling runs
- <number_of_processors> is the number of processors to use for running SiteMap
- <jobname>_reference_structure.mae is a reference structure (usually the starting structure for the simulations) on which the generated protein structures are superimposed.

This analysis can only be run on CPUs. For a typical AACG pocket finder job approximately 300 SiteMap jobs are launched, each of which takes roughly 10 minutes to run. <number_of_processors> SiteMap jobs are run at the same time. Clustering the SiteMap results uses only one processor and typically takes around 5 hours on a modern CPU. Most other analysis steps are brief. Please note that the analysis process will reserve <number_of_processors> processors for the entire analysis job.

3.3 Visualization and interpretation of AACG pocket finder results

The primary AACG pocket finder results are present in the file:
<jobname>_AACG_pocket_finder-out.mae
along with the ancillary files:
<jobname>_AACG_pocket_finder-out.smap
and a number of files ending in .vis.

To examine the results import <jobname>_AACG_pocket_finder-out.mae into Maestro. The structure of the protein from the end of the last simulation will appear in the Workspace (Figure 1a). If the Project Table is not already open type CTRL+T to open it (Figure 1b). Entries with titles starting with “Group” correspond to potential binding site locations. The classification property for each Group is the most important indication of the importance of a Group. This property can take on values of “Primary”, “Secondary”, “Tertiary” or “other” and the groups appear in the project table in this order. Generally, interesting sites are classified as “Primary” and only occasionally “Secondary”. Within each classification level, groups are ordered by the fraction_present property which is the fraction of the simulated time that SiteMap found a potential binding site that ended up in this group. More often than not Primary sites have been found to correspond to known binding sites.

Figure 1. Images of Maestro’s Workspace and Project table upon importing the output structure file from AACG pocket finder.

a)

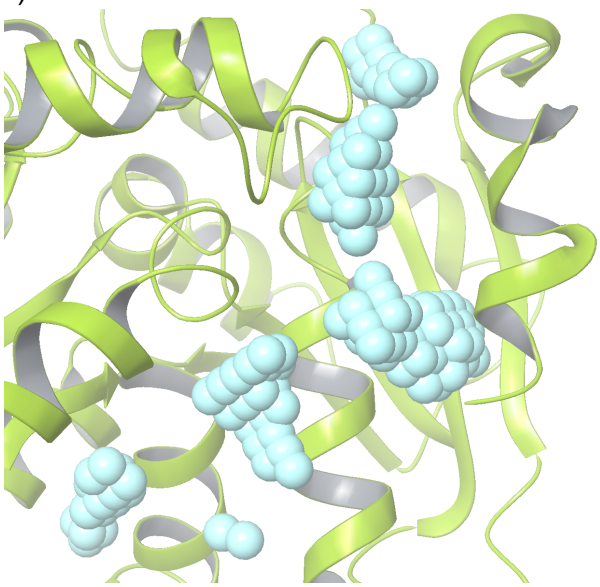
b)

Row	In	Stars	Title	classifition	average num sites	average distance	fraction present	average Dscore	average phobic	average philic	average volume	|charge| per si
	–		1 - my_protein_AACG_pocket_finder-out [49]									
1	■	☆☆☆	**my_protein_11-out-tmp (full system)**									
2	☐	☆☆☆	Group 35 31 34	Primary	96.980		0.909	1.150	2.223	0.768	521.900	0.0
3	☐	☆☆☆	Cluster 35			2.881	0.489	1.141	2.648	0.828	215.900	
4	☐	☆☆☆	Cluster 31			10.059	0.220	1.165	1.846	0.682	1024.700	
5	☐	☆☆☆	Cluster 31 subsite 1				0.220	1.165	1.846	0.682	1024.700	0.0
6	☐	☆☆☆	Cluster 31 subsite 2				0.220	1.165	1.846	0.682	1024.700	0.2
7	☐	☆☆☆	Cluster 34			5.631	0.200	1.155	1.599	0.716	716.900	
8	☐	☆☆☆	Group 23	Primary	87.000		0.760	1.158	1.899	0.580	540.700	0.1
9	☐	☆☆☆	Cluster 23			7.893	0.760	1.158	1.899	0.580	540.700	
10	☐	☆☆☆	Cluster 23 subsite 1				0.760	1.158	1.899	0.580	540.700	0.1
11	☐	☆☆☆	Group 40 32 38	Primary	54.265		0.754	1.154	2.801	0.557	344.200	0.1
12	☐	☆☆☆	Cluster 40			3.679	0.423	1.191	3.607	0.465	165.400	
13	☐	☆☆☆	Cluster 40 subsite 1				0.423	1.191	3.607	0.465	165.400	0.2
14	☐	☆☆☆	Cluster 32			7.184	0.271	1.162	1.840	0.689	664.500	
15	☐	☆☆☆	Cluster 32 subsite 1				0.271	1.162	1.840	0.689	664.500	0.0
16	☐	☆☆☆	Cluster 38			3.588	0.060	0.863	1.465	0.612	158.100	
17	☐	☆☆☆	Cluster 38 subsite 1				0.060	0.863	1.465	0.612	158.100	0.1
18	☐	☆☆☆	Group 6 29 27	Primary	66.107		0.591	1.089	2.639	0.483	294.900	0.0
19	☐	☆☆☆	Cluster 6			3.109	0.311	1.085	3.405	0.307	122.900	
20	☐	☆☆☆	Cluster 6 subsite 1				0.311	1.085	3.405	0.307	122.900	0.0
21	☐	☆☆☆	Cluster 29			7.727	0.191	1.165	1.904	0.703	621.700	
22	☐	☆☆☆	Cluster 29 subsite 1				0.191	1.165	1.904	0.703	621.700	0.0
23	☐	☆☆☆	Cluster 27			2.128	0.089	0.938	1.537	0.626	194.600	
24	☐	☆☆☆	Cluster 27 subsite 1				0.089	0.938	1.537	0.626	194.600	0.0
25	☐	☆☆☆	Group 21 8	Primary	88.160		0.349	1.066	3.118	0.477	521.200	0.1
26	☐	☆☆☆	Cluster 21			10.825	0.200	1.155	1.998	0.634	852.600	
27	☐	☆☆☆	Cluster 21 subsite 1				0.200	1.155	1.998	0.634	852.600	0.1
28	☐	☆☆☆	Cluster 21 subsite 2				0.200	1.155	1.998	0.634	852.600	0.1

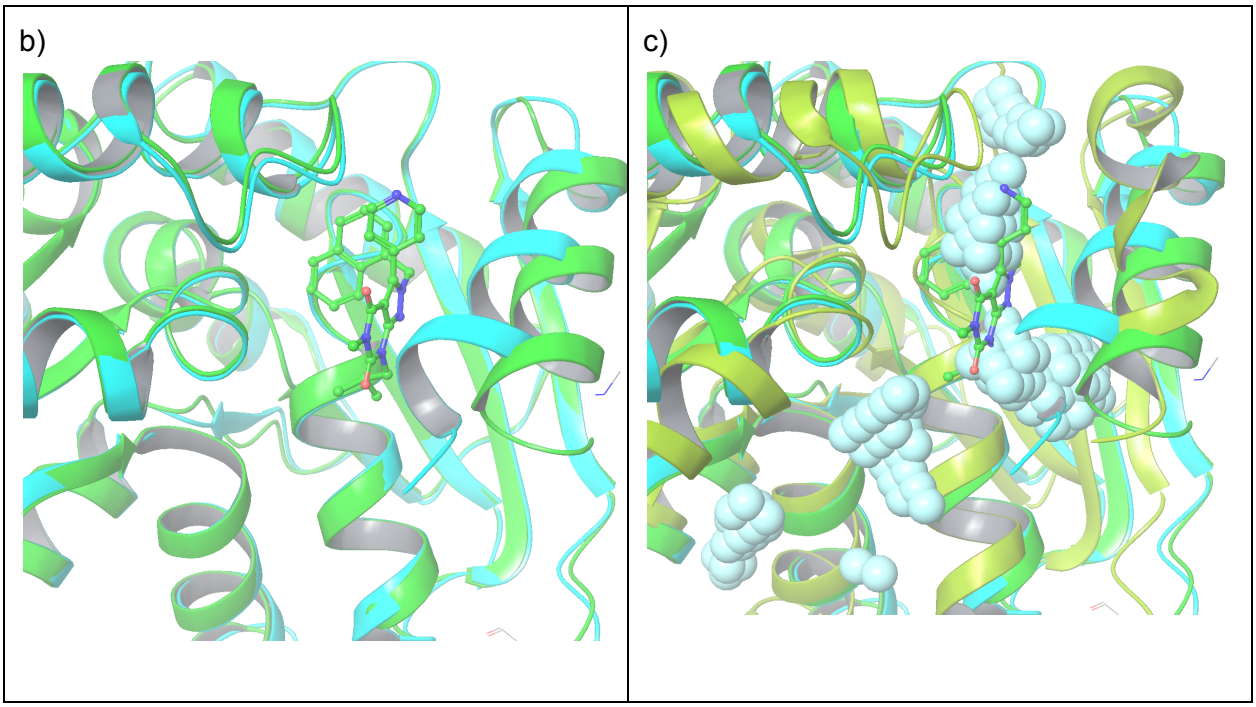
Each group comprises one or more separate clusters of SiteMap results. The individual cluster numbers are listed in the Group Title. Separate Entries for each cluster of SiteMap results follow each Group Entry. Locations of high cosolvent density within or near each of these clusters may also be present if there were 4 or more such locations for a given site. Other useful Group properties include “average DScore”, “average phobic”, “average volume” (in $Å^3$), and “|charge| per site” (average absolute charge across the high density locations).

Figure 2. The Primary group corresponding to the known allosteric site in muri-hp.

a)



a. The 5th Primary group identified by AACG pocket finder.
b. The crystal structure without the allosteric site (cyan, 2JFY), which was used as the starting structure for the AACG pocket finder calculation, and a crystal structure with a ligand in the binding site (bright green, 2JFZ). Note the shift in the alpha helix to the right of the ligand.
c. The crystal structures with the output structure from AACG_pocket_finder and the 5th primary site. Note the good overlap of the primary site with the ligand and the shift in the alpha helix to the right of the ligand in the AACG result.



In this example for muri-hp, there are 5 Primary sites with the remainder being classified as “other”. Three of the primary sites correspond to the known catalytic site present in the initial structure. One corresponds to a location for which a binding site is not known and thus may be not be an actual site. The 5th Primary site is a known allosteric site (see Figure 2) not present in the starting structure.

3.4 AACG pocket finder licensing requirements

An AACG license is required to run each of the three parts of an AACG pocket finder job. AACG pocket finder jobs only check for the existence of an AACG license so one license is enough for multiple simultaneous runs. Please note that Desmond licenses are required, as they are for AA simulations, for system setup and equilibration as well as sampling. In addition, SiteMap licenses are required for the analysis portion of AACG pocket finder jobs.