Docking@Home Newsletter **22-Mar-07 Issue #2**

Welcome to the second issue of the Docking@Home newsletter. It is meant to inform you of the status of the (sub)projects the D@H team is working on.

Table of Contents

1 The Project

The Docking@Home project is the brain-child of Michela Taufer, currently an assistant professor in the Computer Science department of the University of Texas at El Paso (UTEP). Docking@Home is the implementation of the Dynamically Adaptive Protein-Ligand Docking System (DAPLDS) project which involves collaborations among the University of Texas at El Paso (UTEP), The Scripps Research Institute (TSRI), and the University of California at Berkeley. This project enables adaptive multi-scale modeling in a volunteer computing environment and will further the knowledge of atomic details of protein-ligand interactions. By doing so, it will accelerate the discovery of novel pharmaceuticals. The goals of the project are: (1) to explore the multi-scale nature of algorithmic adaptations in protein-ligand docking and (2) to develop cyber-infrastructure based on computational methods and models that efficiently accommodate these adaptations.

Docking@Home

Docking@Home Newsletter **22-Mar-07 Issue #2**

2 Project Status

So what is the current status of D@H and where are we going? Currently the following sub-projects are ongoing in the group:

a) Running CHARMM on architectural different platforms – Memo is working on running experiments that will give us a chance to track down where the divergences in CHARMM occur. He nailed it down to the energy minimization routine, but there is still some work to do to since this routine is approx. 1200 lines of code.

b) Homogeneous Redundancy (HR) – We currently have nine (HR) groups:

- Windows on AMD
- Windows on AMD-K6
- Windows on Intel (PIII and down)
- Windows on Intel (P4 and up)
- Linux on AMD/PII/PIII
- Linux on Intel (P4 and up)
- Linux on AMD-K6
- Mac on Intel
- Mac on PowerPC (no working application version yet)

We are currently working on making these rules more specific and using the new host identification strings that are sent by the newer BOINC client versions: every Intel or AMD host is uniquely identified by a family, model and stepping number, which makes it much easier for us to find the appropriate HR class of a host. The following classes will be likely to appear:

- Windows on Family 15, 6 and 5 (both AMD and Intel)
- Linux on Family 15, 6 and 5 (both AMD and Intel)
- Mac on Family 15 and 6 (Intel)
- Mac on PowerPC

This list is not final yet and is likely to change.

c) Linux ulimit fix – We have found a solution for the problem where the default stack limit of certain Linux distributions is set too low, which made CHARMM crash on these distros. BOINC now distributes the CHARMM executable and a shell script that starts this executable after setting the stack limit to unlimited with the ulimit command.

d) Checkpointing – We have been working a lot on devising a new checkpointing method for CHARMM. The plan is to do this in phases: 1) Make the current checkpointing atomic by using the BOINC critical section functions. 2) Build in a way to reduce the number of checkpoints (currently we create one every conformation and rotation). The first point has been implemented and is currently being tested on our test system. If these tests are successful, we'll deploy the new binaries and input files on D@H. The second point is a lot more challenging, because of the way CHARMM processes data and the fact that we do random seed initiated experiments. This seed needs to be deterministic to make the result predictable so that we can use replica computing.

e) New CHARMM version c33b1 - We are doing tests with the un-BOINC-ified version of CHARMM and it looks like the stack problem has been solved in that release. The divergences among different platforms are still present though, which means we'll still need HR on the D@H system. The BOINC version of c33b1 is still in the works by the team at Scripps and has not been released yet.

f) Fixed credit – We have implemented a new fixed credit scheme based on the estimated number of FLoating point OPerations (FLOPs) that a workunit needs to finish. The formula we are currently using is: 3.3E-12 credit per FLOP, which for the *1tng* workunit we are sending out comes out as 49.5 credit. This means we do not rely on the benchmarks anymore as measured by the BOINC client on your machine. The reason for this is that these benchmarks are presently not completely fair between platforms: i.e. Windows machines gained a lot more credit for the same workunit as a Linux or Mac would receive.

g) D@H Web Site – The web site has not seen too many changes since the last newsletter. A couple of changes are:

- The news now has it's own scrollable section.
- We've added the Community Help link.
- We've added a page to monitor Homogeneous Redundancy status of workunits in the shared memory and database and HR class of active hosts. See for more details the next point **h**).
- An issue with the font size has been resolved.
- We've added a page that gives you an overview of the hardware we run and test D@H on.

h) HR monitoring tool – Memo and Andre have created a tool to monitor the shared memory, database and hosts HR class in real time. A first version that shows shared memory and host distributions is already up on the D@H web site (see screenshot). This tool will basically show how many hosts of a certain HR class are attached to the

project and what type of work is available in the shared memory of the BOINC server. This is useful for researching problems in case people see the dreaded 'Work is available, but has been committed to other platforms' message: if the work queue for your platform shows 0, you know why that message pops up. If there is a lot of unassigned work in the shared memory, this message should not be shown; if it does, please let us know about it via the forums.

Shared Memory HR Distribution

Linux Intel	11	1%
Linux AMD	31	3%
Linux K6	0	0%
Windows Intel	64	6%
Windows AMD	30	3%
Windows PreP4	68	7%
Windows K6	21	2%
Darwin Intel	247	25%
Darwin PPC	0	0%
Unassigned	528	53%

Host Distribution (Avg. credit > 0.1)

Linux Intel	66	9%
Linux AMD	68	9%
Linux K6	0	0%
Windows Intel	306	41%
Windows AMD	201	27%
Windows PreP4	76	10%
Windows K6	1	0%
Darwin Intel	28	4%
Darwin PPC	0	0%
Other	2	0%

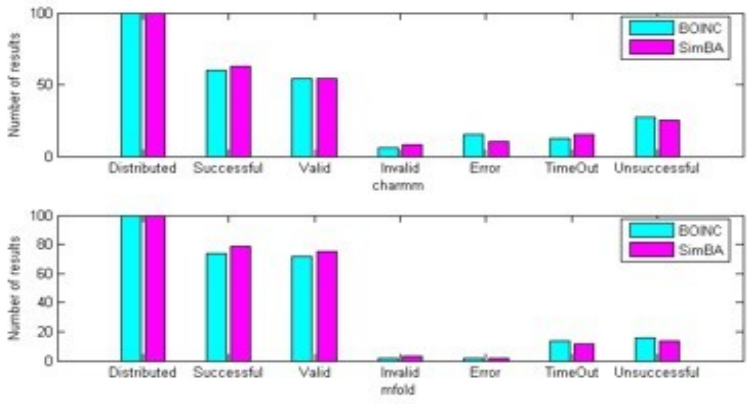
Created Thu Mar 15 18:20:02 UTC 2007

i) Screensaver Graphics – Karina has developed a first version of a screensaver for the Windows application. This was a great chance for her to use her newly gained OpenGL skills in a real project.

The screensaver shows the D@H logo in the background and shows the ligand molecule that is being crunched on, which is currently *1tng*, our alpha test ligand. The atoms are scaled based on the element mass and they are colored according to the

Visual Molecular Dynamics (VMD) color convention (more details, information and links can be found at: http://en.wikipedia.org/wiki/Visual_molecular_dynamics). When the screensaver is running the molecule moves around the screen in different directions. At this time, the screensaver is being worked on in order to improve the molecule representation and read ligand input data from PDB (Protein DataBase) files. We intend to draw the bonds between atoms and to print detailed information about the molecule, i.e. name, min. energy and min. RMSD (Root Mean Square Deviation) found.

j) SimBA – Our Simulator for BOINC Applications, or SimBA, is now able to simulate the scheduling policy based on minimum average credit and maximum average turnaround as used by World Community Grid. Under this policy, results in shared memory are marked as urgent when at least another result from the same work-unit is either invalid or error (including time-outs). Only hosts that have a Recent Average Credit (RAC) above a certain threshold and an average turnaround below a minimum are able to get results marked as urgent. We have run experiments that show that the average difference between the performance of several projects of WCG and the predictions made by SimBA is less than 6%. Future goals include implementation of the infeasibility count for results that have been too long in shared memory and partitioning of the shared memory (as used by the Leiden projects) in order to reduce the number of machines that request work unsuccessfully. Following are a couple of graphs that show some simulation results.



The first graph shows the difference between the results that SimBA acquires against the results obtained from Predictor@home when the FCFS scheduling policy of BOINC is used. It can be seen that the average difference between the simulator and project results is less than 6%. The paper that was accepted in the PADS'07 conference discusses SimBA results more in detail. We will post this paper on the D@H website after the conference.

k) One result per host per work-unit – Since March 1 we are distributing one replica per work-unit per host instead of one replica per work-unit per user. We had to enable this because the work queue of Intel Macs and Intel Linux machines had grown so excessively that nobody else could get work anymore. Andre has written a patch for BOINC to include this functionality as a new config.xml option; this patch has been included in the official BOINC CVS tree.

l) CHARMM on Mac PPC – We have been able to compile CHARMM on the PPC platform, but cannot get rid of a dependency on the Fortran libraries. If these libraries are not present on the target system, CHARMM will crash immediately. Since the Intel Mac version of CHARMM does not have this issue, there must be a way to get rid of it using the GNU compiler, but we have not found this yet. A statically compiled executable would be easiest in this case, but Apple does not recommend and support this. Another solution would be to ask our volunteers to download these libraries and install them, but this is not really a good option for later when we might have thousands of volunteers attached to D@H.

m) Credit tags in commandline of CHARMM – We noticed recently that the following tags were supplied as parameter to the CHARMM application: <credit>49.500000</credit>. With the help of Charlie Fenton (BOINC Mac developer) we figured out that this was due to the fact that our credit is fixed and created at workunit creation time. Because we were using a (very) old version of the create_work function call, the credit ended up as a command-line argument to CHARMM! Of course this was easily fixed by using the correct create_work call.

Docking@Home Newsletter **22-Mar-07 Issue #2**

3 Science Status

Introduction to HIV Protease Inhibitors for treatment of HIV/AIDS.

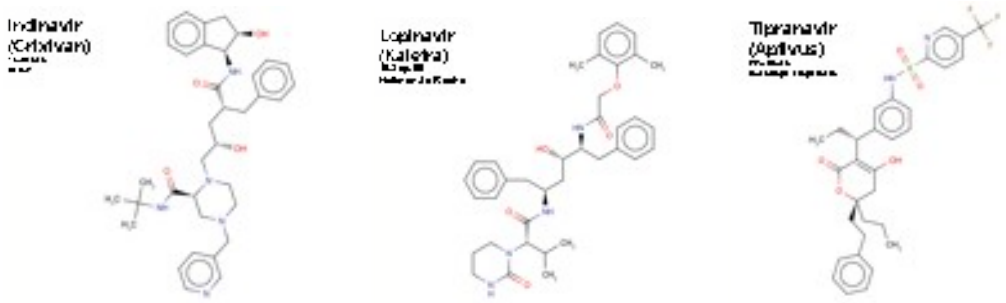
Enzymes are one of the most important functional classes of proteins for drug design. Enzymes participate in active biochemistry by catalyzing a specific biochemical reaction and convert reactants to products. Enzymes are part of complex biochemical pathways. Some enzymes serve as steps in sequential pathways, and other enzymes function to regulate pathways. If an enzyme converts a biochemical reactant (A) to a product (B), it is possible to design an **enzyme inhibitor** that is similar to the reactant (A) but does not allow the reaction to convert the inhibitor to the product (B). Enzyme inhibitors prevent or reduce the formation of product (B). Many common drugs marketed today function in this generic way as enzyme inhibitors. Most enzymes participate in complex biochemical pathways, and certain enzymes can be identified from disease research as **drug targets**.

Proteases are enzymes that perform their function by cleaving a specific recognition sequence in the peptide chain of another protein. The common analogy is that proteases work like a pair of molecular scissors that cut proteins at specific locations. Many important biochemical pathways are regulated by protease activity. **Aspartic proteases** became an important class of enzymes when HIV protease was identified as a drug target from the HIV genome. Inhibiting the HIV protease enzyme inhibits the replication of the virus and the spread of the infection to new healthy cells.

The FDA approval of several potent **HIV protease inhibitors** has revolutionized the treatment of HIV/AIDS. When these HIV protease inhibitors are co-administered with **HIV Reverse Transcriptase Inhibitors** in **combination therapy**, they have been shown to be very effective in reducing the viral load of patients. Examples of some HIV protease inhibitors that have been approved to date include: Saquinavir, Ritonavir, Indinavir, Nelfinavir, Amprenavir, Lopinavir, Atazanavir, Reyataz, Tipranavir, and Darunavir. Each of these inhibitors have various advantages and disadvantages with regards to **dosage** (the number of pills taken per day), **toxicity** (side effects) and the **development of drug resistance** (drug resistant strains of HIV virus). One advantage of including more than one protease inhibitor in combination therapy is that this can lower the total dosage of inhibitors administered, and therefore reduce toxicity by administering a lower dosage of two inhibitors, rather than a larger dosage of a single inhibitor. An additional advantage of including more than one protease inhibitor in combination therapy is that certain combinations have been shown to be more effective at protecting against the development of drug resistance in patients.

At the molecular level, the differences in the molecular structures of inhibitors lead to

different profiles of drug resistance. Some of the inhibitors when administered as the only protease inhibitor in combination therapy allow the HIV virus to develop drug resistant mutations. Other inhibitors such as **Tipranavir** seem to be much more effective at treating previous drug resistant strains of the virus (this is known as salvage therapy for patients with some drug-resistance). At the molecular level this is because the HIV protease enzyme seems to require multiple mutations to develop resistance to **Tipranavir**. The chemical structure of most of the FDA approved protease inhibitors are **peptidomimetic** (this means that they resemble the chemical structure of the peptide, that is the natural substrate for the HIV protease reaction - cleaving the peptide chain). Tipranavir is an HIV protease inhibitor that is non-peptide based inhibitor, and the basic backbone of this molecule does not resemble a peptide. The non-peptide structure of Tipranavir confers some of its properties for being more protective against the formation of drug-resistant strains of the virus. Unfortunately, Tipranavir is also more toxic, and shows more severe side effects in certain patients than other protease inhibitors. Therefore and ongoing goal of development of new classes of protease inhibitors is to develop new non-peptide inhibitors that have low levels of toxicity and may also have favorable properties with regards to the development of drug-resistance. It is also an ongoing goal to reduce the toxicity of inhibitors similar to Tipranavir.



Recent work on HIV Protease Inhibitors at Docking@Home.

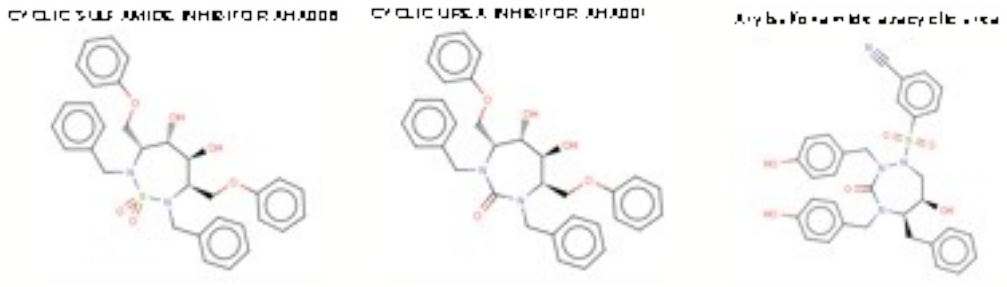
In order to perform our CHARMM-based molecular docking calculations on a protein-ligand complex, one of the most challenging preliminary steps is to develop a reasonable potential energy function for both the protein and the ligand. Over the years much work has been done by many researchers in the CHARMM community to develop and verify various all-atom potential functions for proteins. However, much work still needs to be done to develop, improve, and verify a generalized potential function for small molecule ligands that is compatible with the CHARMM potential function for proteins. Changes to the **potential energy function** for the small molecule ligands and protein will effect the accuracy of docking results. Therefore we can validate changes to the potential function by **docking accuracy**.

Docking@Home Newsletter **22-Mar-07 Issue #2**

During the last few months we have been working with our new test-set of 33 HIV protease protein-ligand complexes. The ligands in this test set cover a range of chemically diverse classes of inhibitors including derivatives of many FDA approved inhibitors such as Indinavir and Tipranavir. For this test set, we have been examining the performance of different changes to the potential energy function. Each time we change the potential function, we can test the performance of this change by looking at the accuracy of the predicted binding geometry (of the ligand bound to the protein) compared to the experimental structure. We have recently implemented an **implicit solvent model** for use in our calculations that allows us to include the effects of water without explicitly representing solvent atoms (which is very computationally expensive). This implicit solvent model has been extensively tested for use with CHARMM, but had not yet been applied to our docking simulations. Using this implicit solvent model, we can predict the **free energy of binding** for a ligand, and then compare these predicted values to experimental values. The free energy of binding is a thermodynamic property can be thought of as the strength of the protein-ligand interaction. This property is important for predicting if a ligand may be a potent inhibitor or not. We have been assessing our predictions for the free energy of binding for the HIV protease inhibitor test set, both from the experimentally known binding geometry, and also from predicted binding geometries from our docking simulations. Preliminary results with this test set suggests that we will be able to use our implicit solvent model to improve the accuracy of our docking geometries. These preliminary results also suggest that our predictions from this model may be good enough to rank inhibitors in a virtual screen of new inhibitors where there is no experimental information about the binding geometry. This is crucial for the discovery of novel classes of inhibitors (different **chemotypes**) with novel or different chemical connectivity between atoms.

We have also started to investigate the role of **protein flexibility** in docking simulations by performing **cross-docking simulations** with our HIV protease inhibitor test set. When a ligand binds to a protein, there is some "**induced-fit**" effect, where a certain ligand will preferentially bind to a specific conformation of the protein's binding pocket. Therefore, the binding pocket may look slightly different from one ligand to another, and often greater difference in the structure of the ligand will result in different conformations of the protein receptor. One theoretical problem in **structure-based-drug-design** and in **virtual screening** is how to deal with this type of flexibility in the protein binding pocket. In the current implementation of our docking simulations the protein conformation is rigid while the ligand is allowed to be flexible. We aim to develop a method to include protein flexibility in our docking predictions. In order to do this, a good place to start is to take our test set, and then dock each ligand in the test set, into each conformation of the binding pocket (this is known as cross-docking). What we can learn from cross-docking is how sensitive our results are

to slight changes in protein conformation. From these simulations we can see how to use experimental protein conformations to develop a minimal model of protein flexibility that can hopefully improve our results. Initial cross-docking results suggest that using multiple protein conformations can improve our results for both docking accuracy and prediction of free energies of binding. In the future, we intend to develop computational models to include protein flexibility *a priori*, and then compare the performance of these models to the benchmark of cross-docking into experimentally determined structures.



We have also been working on our first example **virtual screen** with a series of 60 non-peptide HIV inhibitors (like Tipranavir) that are based on a cyclic urea and cyclic sulfamide chemical scaffolds. No HIV inhibitor that is based on a cyclic urea has yet gained approval by the FDA. However, several inhibitors of this type (such as DMP 450 or Mozenavir) have been shown to be save and effective in class I and class II clinical trials, so it is possible that an inhibitor of this class could yet be found with more improved properties. In a virtual screen, we do not know the binding geometry of all of the compounds, but in this case, we have examples of 5 inhibitors of this general class in our test set of experimental protein-ligand complexes, so we have some idea of what binding geometries can be expected. Our initial results from these tests are promising and show that in most cases we are able to predict a reasonable binding geometry and also perform fairly well at ranking the free energy of binding. We can also use these virtual screens as a test for changes to the potential function, and also to test methods of including protein flexibility in a virtual screen. We intend to continuing to add more and more potential HIV inhibitors to this virtual screen as we go. In our next rounds we are planning to add up to 1000 more inhibitors with known values for the experimental binding free energy, including a series of Tipranavir derivatives.

Docking@Home Newsletter **22-Mar-07 Issue #2**

4 Publications

The following D@H-related papers have been accepted in peer-reviewed conferences (there might be overlap with the first newsletter):

- M. Taufer, A. Kerstens, T. Estrada, D.A. Flores, and P.J. Teller: SimBA: a Discrete Event Simulator for Performance Prediction of Volunteer Computing Projects. To appear in Proceedings of the International Workshop on Principles of Advanced and Distributed Simulation 2007 (PADS'07), June 2007, San Diego, California, USA.

  M. Taufer, A. Kerstens, T. Estrada, D.A. Flores, R. Zamudio, P.J. Teller, R. Armen, and C.L. Brooks III: Moving Volunteer Computing towards Knowledge-Constructed, Dynamically-Adaptive Modeling and Scheduling. To appear in Proceedings of the First Workshop on Large-Scale, Volatile Desktop Grids (PCGrid'07), in conjunction with IPDPS'07 March 2007, Long Beach, California, USA.

Docking@Home Newsletter **22-Mar-07 Issue #2**

5 Some Interesting Statistics

Here are some details gathered from the Docking@Home server.

Volunteers – 365

Active volunteers (total credit > 0) – 333

Hosts – 1709

- Windows – 1245
  - Intel – 831
  - AMD – 411
  - Cyric – 3
- Linux – 405
  - Intel – 253
  - AMD – 152
- Macintosh – 59
  - Intel – 45
  - PPC – 14

Active hosts (total credit > 0) – 1167

Generated workunits – 35364

Valid workunits – 28023

Generated results – 121104

Valid results – 78129

Total credit – 3,431,147

Total Macintosh credit – 430,182

Total Windows credit – 2,520,152

Total Linux credit – 480,812

Docking@Home Newsletter **22-Mar-07 Issue #2**

6 What will the Future bring?

As you have seen in the previous sections, lots is going on in the project and we could not have done it without your help. When we have resolved most of our pressing problems, reliable checkpointing being the most important, we will go to the next Beta phase and open up the project for more volunteers.

Many thanks from the Docking@Home project team!

Contact us at: dockingadmin@utep.edu