Brian Tsui

PRIME 2011 Osaka University

September 1 2011

Mentors: Dr. Haga and Dr. Date

Abstract:

This experiment focused on using the programs MODELLER and DOCK6 to research the Dual Specificity Phosphatase family. MODELLER was used to predict the models of 23 different proteins in the Dual Specificity Phosphatase family. DOCK6 was used to calculate the ligand-protein binding energies of two proteins.

Introduction:

The Dual Specificity Phosphatase (DSP) family contains 61 different proteins, all of which have many important functions in the body. A shallow and wide active site characterizes this family of proteins; furthermore, the entire family contains the conserved motif of HCXXXXXR (where H is histidine, C is cysteine, R is arginine, and X is any amino acid) as the active site. Many PRIME projects have focused on Slingshot-2 (SSH-2), a protein that is part of the DSP family. SSH-2 modulates the activity of cofilin through dephosphorylation at phosphoserine-3 (Jung et al, 2007). This dephosphorylation activates cofilin, promoting the disassembly of actin filaments. Various studies have linked hyperactive cofilin to Alzheimer's disease, making SSH-2 an attractive target to inhibit (Bamburg, et al., 2010).

Past PRIME projects have focused on finding inhibitors of SSH-2 by using *in silico* screening, which uses computers to screen millions of compounds for a protein. By ranking the energy score outputs, scientists can identify the top compounds that inhibit the protein. These energy scores reveal which compounds bind the best to the ligand because a suitable drug candidate would reduce the energy of the ligand-protein system after the ligand has bound to the protein. While it would be burdensome for scientists to test a few million compounds by hand, this process could be completed in a few days using computers. After identifying the top compounds, scientists can then test them in the laboratory, saving an enormous amount of time. One program that performs *in silico* screening and was used in past PRIME projects is DOCK6, which uses a 3D model of a protein and “docks” a ligand into the active site of the protein (UCSF). DOCK6 rotates the ligand in the active site to find the best conformation between it and the protein (UCSF). This process is repeated for the entire ZINC database, a collection of various small-molecules (UCSF BCIRC, 2009). For further precision, another score, AMBER, is also used in docking. AMBER allows the geometry of the receptor to change as the ligand tries to bind in the active site (Shivakumar, 2011). Once all the AMBER and grid scores are calculated, the scores are used to determine the rankings of how well a ligand binds to the protein.

However, because SSH-2 is part of the DSP family, inhibitors of SSH-2 can also affect the function of other proteins in the DSP family. To make sure that these inhibitors were indeed specific for SSH-2, other PRIME projects have focused on using *in silico* screening to test the same ZINC database against other proteins in the DSP family. An example of a project is the virtual screening of ligands for the proteins PRL-3 and MKP-6, both part of the DSP family (Lo, 2010). Ligands can then be compared across many different proteins, and the ones that only inhibited SSH-2 can be identified. However, scientists have not discovered the 3D structures of all the proteins in the DSP family, which has delayed the testing for the specificity of the various inhibitors.

To solve this problem, MODELLER, a homology protein folding program, can be used to predict the 3D structures of proteins (Eswar, et al., 2006). MODELLER uses information from protein templates that have known 3D structures to predict the structure of other proteins that are closely related. The program can even use information from multiple templates to build models of the protein. After the program has created the protein models, MODELLER evaluates these models using molpdf, DOPE, and GA341 scores. These various scores estimate the energy of the model, giving a good estimate of the quality of the proposed models of the protein. The GA341 score indicates how “native-like” the protein is folded. If a protein is more native-like, then it contains folds that are at least comparable to low-resolution X-ray crystallography structures (Bino & Sali, 2003). However, it is not useful for comparing different models because if there are few errors, all models will have a GA341 close to 1. The molpdf and DOPE scores estimate the energy level of the protein; lower energy levels show that the protein is more stable (Bino & Sali, 2003). This estimates how well the protein is folded because a better fold will result in a lower energy level of the protein. These two scores give a better indication of which models MODELLER folded the best. Furthermore, MODELLER can plot a DOPE energy profile of the entire protein, indicating which locations in the protein have high energy. Once these high-energy locations are identified, they can undergo loop refinement to reduce the energy of those loops. To globally reduce the energy of the model further, CHIMERA, a molecular visualization program, can perform energy minimization on the entire protein (Petterson et al, 2004). Finally, the protein can be prepared and used for ligand-docking studies.

An enormous amount of computing power is needed for these modeling and docking studies. In order to make these studies feasible, computer clusters from the PRAGMA grid are used. The PRAGMA grid is a network of supercomputers around the world. By splitting these jobs into different slices and sending them to individual nodes on the supercomputers, these studies can be done many times faster than if they were only performed on a single computer. To secure the files and submit jobs, Opal-OP is used. Opal-OP allows the user to wrap scripts and files in a module, which eliminates the need for security and job submission (National Biomedical Computation Resource, 2011). To communicate among the different clusters, Tomcat Jakarta is installed on all the clusters, which allows files to be sent easily among clusters.

Materials and Methods:

Computer clusters from the Supercomputer Center in San Diego, Osaka University in Japan, and University of Zurich in Switzerland were used in this project. The table below shows the number of nodes and CPUs in each of the clusters (PRAGMA).

Name	Number of nodes	Number of CPUs
rocks-200 (SDSC)	17	17
Tea (Osaka University)	40	80
cafe (Osaka University)	20	40
Ocikbpra (University of Zurich)	10	20

Table 1. List of clusters, nodes, and CPUs used in the experiment

DOCK6 and MODELLER were installed on each of the clusters. The amino acid sequences of all the proteins in the DSP family that do not have a known 3D structure were tabulated from Pubmed. Since some proteins had multiple isoforms, the longest isoform of each of the protein was inputted into MODELLER. Because grid computing was used for MODELLER, a workflow was designed based off of a past PRIME project's scripts (Xue, 2010). This diagram is shown in figure 1.

The protein sequence was converted into an ALI file compatible with MODELLER. Next, the protein name in the file "important_info" (figure 3) was changed to the name of the protein currently being modeled. The number of models generated was also written in "important_info" and was set to 600 for all proteins that were modeled. The protein sequence was inputted into SwiftModeller, a Windows suite for MODELLER. Using the build_profile.py script in either SwiftModeller or MODELLER, a list of potential templates for the protein was generated in build_profile.prf. Four to five templates with the highest homology were selected and downloaded from the Protein Databank (PDB). Compare.py from either SwiftModeller or MODELLER then generated a dendrogram that displayed the relative homologies among the templates. The group of templates with the highest homology was then chosen and used as the templates for modeling. The pdb names of the protein templates were written into "model_sequences" as shown in figure 2.

The script "fit_distribute.pl" used the files "important_info," "model_sequences," and the protein ALI file as inputs, sending these files to the different clusters using Opal-OP. “Fit_distribute.pl” also converted parallel-task.py.generic.original into parallel-task.py.generic by renaming the file and adding the PDB name of the protein into the file. Depending on how many templates were used, different alignment scripts were used. If only one template was used, then fit_distribute.pl converted align2d.py.generic.original into align2d.py.generic. If more than one template was used, then fit_distribute.pl converted align2d_mult.py.generic.original into align2d_mult.py.generic. Additionally, fit_distribute.pl would send an additional script, salign.py, to the different clusters. The script read how many models were to be generated and also generated a slice_array file (figure 4), which listed how many different slices should be generated. The number of slices depended on how many templates were used, since the script would generate 600 models of each combination of templates. Finally, the script instructed Opal-OP to send the job to various clusters so MODELLER could be run in parallel.

A second script, “modrun.pl” further converted parallel-task.py.generic into parallel-task.py, a python script that MODELLER could use. Before the job is sent send to the different nodes on each of the clusters, the alignment scripts generate align the templates and the target sequence. First, if there was more than one template, the templates were first aligned using salign.py. Salign.py uses a genetic algorithm to generate structural alignments of the templates (Madhusudhan et al, 2009) and evaluates the alignments with a score. If the alignment score was over 70, then the template alignment was used to align the model sequence using align2d_mult.py. However, if the alignment score was under 70, an iterative version of salign.py was performed before passing the alignment to align2d_mult.py. Both align2d.py and align2d_mult.py aligned the model sequence and the templates; however, align2d.py was used when there is only one template. The script “parallel-task.py” finally instructed MODELLER to build 600 models of the protein per slice.

After all the different slices of the protein were done, the GA341, DOPE, and molpdf scores were tabulated and ranked. The DOPE and molpdf scores were normalized and weighted equally for consensus scoring. The top models of the protein were then chosen for loop refinement and energy minimization in the molecular visualization program, CHIMERA.

For the docking process, the receptor was first prepared according to the DOCK6 tutorial using CHIMERA (UCSF). After the necessary receptor and ligand files for DOCK6 were generated, the script “slice_distribute.pl” sent these files and dock.in.generic to various clusters. “Bigrun.pl” then instructed DOCK6 to docking ligands in the active site of the protein after converting dock.in.generic into dock.in, an input file for DOCK. After DOCK6 docked all the ligands and produced the energy scores, the scores were ranked in terms of energy scores. Then, the script “slice_redistribute.pl” resent the ligand and receptor files for AMBER score calculations. Once AMBER score calculations had been performed on all the ligands, the scores were ranked. The grid score and AMBER score were then weighted equally and used to determine the ligands that bound to the protein the best.

Results:

Using MODELLER, many protein models were generated over 9 weeks. Table 2 lists which proteins were modeled and how many models were generated.

In the last 4 weeks, some AMBER and grid scoring were able to be performed on the protein 3EZZ while only grid scoring was performed on 2Y96. Because consensus scoring cannot proceed without the AMBER scores, there is no data tabulated on these two proteins yet.

Discussion:

From the models that were generated using MODELLER, only the proteins with shorter length were modeled well, according to the DOPE and molpdf scores. Because many of the templates are only around a 100 amino acids long, the alignment between the templates and the model is very short if the model is more than a few hundred amino acids long. This poor alignment does not give MODELLER adequate information on how to fold many parts of the protein, leading to many misfolds and incorrect loops. Additionally, if a low-resolution template is used to model a protein, the energy score of the protein tends to be higher because of conflicting information between it and other templates.

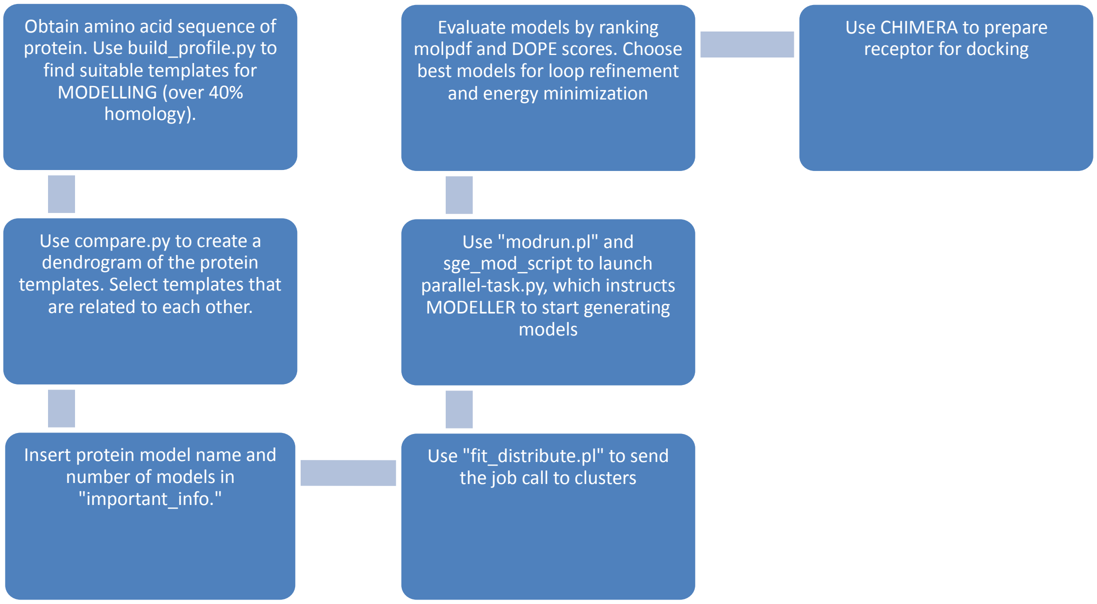
Most importantly, the models generated using MODELLER are only estimates of what the actual 3D model of the protein should look like. The MODELLER documentation claims that if the templates and model have an alignment over 50%, then the model generated will have accuracy comparable to a low resolution (3 angstrom) model (Eswar et al, 2007). In this experiment, it can only be used as a statistic because the 3D structures of our modeled proteins have yet to be discovered. Because the resolution of most crystallography studies is around two angstroms, MODELLER's three angstrom error is acceptable. However, unless the 50% alignment criterion is met, MODELLER may introduce enormous error when the protein is used for docking studies. Because of this uncertainty, many models of the same protein may have to be used in docking to obtain accurate results.

One method that could control for this uncertainty is to model proteins from the DSP family with known 3D models. Using this method, certain loops or regions that MODELLER has trouble modeling in the DUSP family could be identified. While this process has been done for SSH-2, if this "test" modeling is done for a few more proteins, the accuracy of MODELLER for the DSP family could be better qualified. Furthermore, only the active site of the protein is used in the docking process. If these tests indicate that the active site is modeled correctly most of the time, then a precision in the active site that is higher than three angstroms may be obtainable.

Because the docking part of the experiment was started on the sixth week of PRIME, not all of the molecules have had an AMBER score generated. Once the AMBER scores have been calculated for all of the molecules, a consensus score for all of the ligands will be calculated. These will then be compared to the list of inhibitors for SSH-2, giving a good idea of which molecules are specific only for SSH-2.

The next step after the MODELLER experiments is to choose which protein models are good enough for docking. Before docking, however, energy minimization and loop refinement will be performed to improve the quality of the protein. After docking is done, ligands that bind specifically to SSH-2 can be chosen for further laboratory analysis such as Western blots and actin staining. Because an energy score is only an approximation of how well the ligand binds to the protein, wet-lab tests must be performed on the top candidates to ensure that they inhibit SSH-2.

Appendix:



Obtain amino acid sequence of protein. Use build_profile.py to find suitable templates for MODELLING (over 40% homology).

Use compare.py to create a dendrogram of the protein templates. Select templates that are related to each other.

Insert protein model name and number of models in "important_info."

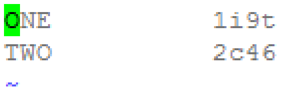
Use "fit_distribute.pl" to send the job call to clusters

Use "modrun.pl" and sge_mod_script to launch parallel-task.py, which instructs MODELLER to start generating models

Evaluate models by ranking molpdf and DOPE scores. Choose best models for loop refinement and energy minimization

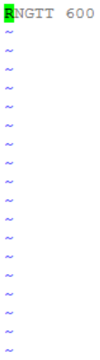
Use CHIMERA to prepare receptor for docking

Figure 1. Summary of MODELLER procedure in a workflow format



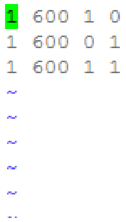
```
ONE           1i9t
TWO           2c46
~
```

Figure 2. Sample "model_sequences" input file. The first column indicates the position of the template. The second column indicates the PDB code of the template.



```
RNGTT 600
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

Figure 3. Picture of the file "important_info." The first column is the name of the protein in the ALI file. The second column indicates how many models should be generated per slice.



```
1 600 1 0
1 600 0 1
1 600 1 1
~
~
~
~
~
```

Figure 4. Picture of the file "slice_array." The first column indicates which model number to start modeling. The second column indicates which model number MODELLER should finish. The third and fourth columns tell which models should be used in the slice as inputted from "model_sequences". The "1" indicates the model should be used in the slice while "0"indicates that it should not be used. In this example, there are 2 templates.

Name of proteins modelled	Templates used (PDB codes)	Templates used (names)	Length of protein in amino acids	Number of models generated total
DUSP1	1m3g, 2g6z, 3ezz	DUSP2, DUSP5, DUSP4	367	4200
DUSP7	1mkp, 1zzw, 2hxp	DUSP6,DUSP10, DUSP9	419	4200
DUSP8	1zzw 2oud, 2vsw	DUSP10 (2x), DUSP16 rhodanese domain	625	2400
DUSP16	1zzw 2oud, 2vsw	DUSP10 (2x), DUSP16 rhodanese domain	665	2400
DUSP11	1i9t, 1yn9, 2c46	mRNA enzymes	377	2400
DUSP12	1wrm, 1yz4, 2g6z	DUSP22, DUSP15, DUSP5	340	4200
DUSP13A	2e0t, 2pq5, 2y96	DUSP26, DUSP13B, DUSP27	188	4200
DUSP19	1wrm, 1mkp, 1yz4, 1zzw, 2hxp	DUSP22, MKP3, DUSP15, DUSP10, DUSP9	217	6000
DUSP21	2esb, 2wgp	DUSP18, DUSP14	190	1800
SSH1	2nt2	SSH2	1049	600
SSH3	2nt2	SSH2	659	600
PTP4A2	1r6h, 1x24	PRL-1 and PRL-3	167	1800
CDC14A	1ohe	CDC14B	623	600
PTP9Q22	1fpz	KAP	754	600
TPIP	1d5r	PTEN	522	600
TPTE	1d5r	PTEN	551	600
TNS	1d5r	PTEN	1735	600
TENC1	1d5r	PTEN	1419	600
MTM1	1zsq	MTMR2	603	600
MTMR1	1zsq	MTMR2	665	600
RNGTT	1i9t, 2c46	mRNA enzymes	595	1800
MTMR3	1zsq	MTMR2	1198	600
MTMR4	1vfy, 1zsq, 2yqm	FYVE domain of Vps27p, MTMR2, solutions structure of FYVE	1195	1800

Table 2. Table of proteins modeled. Indicates which templates were used and how many total models were generated from all the slices.

Works Cited

Bamburg, J., Bernstein , B., Davis, R., Flynn, K., Goldsbury, C., Jensen, J., et al. (2010). ADF/Cofilin-Actin Rods in Neurodegenerative Diseases. *Current Alzheimer Research*, 241-250.

Bino, J., & Sali, A. (2003). Comparative protein structure modeling by iterative alignment, model building and model assessment. *Nucleic Acids Research, 31*(14), 3982-3992.

Eswar, N., Marti-Renom, M., Web, B., Madhusudhan, M., Eramian, D., Shen, M., et al. (2006). Comparative Protein Structure Modeling with MODELLER. *Current Protocols in Bioinformatics*, 5.6.1-5.6.30.

Fiser, A., Do, R. K., & Sali, A. (2000). Modeling of loops in protein structures. *Protein Science*, 1753-1773.

Hinsen, K. (2000). The Molecular Modeling Toolkit: A New Approach to Molecular Simulations. *Journal of Computational Chemistry*, 79-85.

Jung, S. K., Jeong, D., Yoon, T., Kim, J., Ryu, S., & Kim, S. (2007). Crystal Structure of human slingshot phosphatase 2. *Proteins: Structure, Function, and Bioinformatics*, 408-412.

Lo, K. (2010, September 2). *PRIME*. Retrieved August 23, 2011, from PRIME Reports 2010: http://prime.ucsd.edu/progress_reports_2010/PRIME_Final_Final/KLo_Final_2010.pdf

Madhusudhan, M., Webb, B. M., Marti-Renom, M. A., Eswar, N., & Sali, A. (2009). Alignment of multiple protein structures based on sequence and structure features. *Protein Engineering, Design & Selection, 22*(9), 569-574.

National Biomedical Computation Resource. (2011, February 8). *NMCR Opal Toolkit*. Retrieved August 28, 2011, from About Opal Toolkit: http://www.nbcr.net/software/opal/

Pettersen, E., Goddard, T., Huang, C., Couch, G., Greenblatt, D., Meng, E., et al. (2004). UCSF Chimera--a visualization system for exploratory research and analysis. *Journal of Computational Chemistry*, 1605-1612.

PRAGMA. (n.d.). *PRAGMA-grid*. Retrieved August 15, 2011, from PRAGMA Computational Grid Resources: http://goc.pragma-grid.net/pragma-doc/computegrid.html

Sali, A., & Blundell, T. (1993). Comparative protein modelling by satisfaction of spatial restraints. *Journal of Molecular Biology*, 779-815.

Shivakumar, D. (2011, July 28). *Amber Score in DOCK6*. Retrieved August 25, 2011, from DOCK6: http://dock.compbio.ucsf.edu/DOCK_6/tutorials/amber_score/amber_score.htm

UCSF BCIRC. (2009, August 6). *ZINC*. Retrieved August 25, 2011, from ZINC: http://zinc.docking.org/

UCSF. (n.d.). *DOCK*. Retrieved July 30, 2011, from Tutorials for DOCK 6.5: http://dock.compbio.ucsf.edu/DOCK_6/tutorials/index.htm

UCSF. (n.d.). *What is DOCK?* Retrieved August 28, 2011, from The Official UCSF DOCK Web-site: http://dock.compbio.ucsf.edu/Overview_of_DOCK/index.htm

Xue, C. (2010, August 20). *PRIME*. Retrieved August 26, 2011, from PRIME 2010 Reports: http://prime.ucsd.edu/progress_reports_2010/PRIME_Final_Final/CXue_Final_2010.pdf