Journal of
CHEMICAL INFORMATION
AND MODELING

Article

pubs.acs.org/jcim

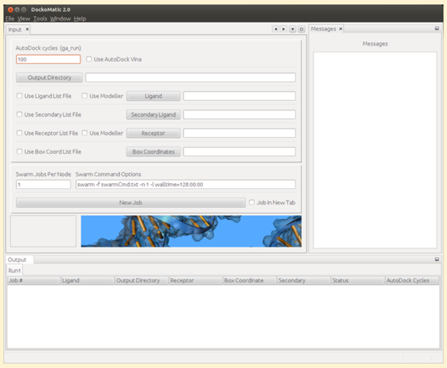
DockoMatic 2.0: High Throughput Inverse Virtual Screening and Homology Modeling

Casey Bullock,[†] Nic Cornia,[†] Reed Jacob,[‡] Andrew Remm,[‡] Thomas Peavey,[‡] Ken Weekes,[‡] Chris Mallory,[‡] Julia T. Oxford,[¶] Owen M. McDougal,*[,‡] and Timothy L. Andersen*[,†]

[†]Computer Science Department, [‡]Department of Chemistry and Biochemistry, and [¶]Department of Biological Sciences, Boise State University, Boise, Idaho 83725, United States

S *Supporting Information*

ABSTRACT: DockoMatic is a free and open source application that unifies a suite of software programs within a user-friendly graphical user interface (GUI) to facilitate molecular docking experiments. Here we describe the release of DockoMatic 2.0; significant software advances include the ability to (1) conduct high throughput inverse virtual screening (IVS); (2) construct 3D homology models; and (3) customize the user interface. Users can now efficiently setup, start, and manage IVS experiments through the DockoMatic GUI by specifying receptor(s), ligand(s), grid parameter file(s), and docking engine (either AutoDock or AutoDock Vina). DockoMatic automatically generates the needed experiment input files and output directories and allows the user to manage and monitor job progress. Upon job completion, a summary of results is generated by Dockomatic to facilitate interpretation by the user. DockoMatic functionality has also been expanded to facilitate the construction of 3D protein homology models using the Timely Integrated Modeler (TIM) wizard. The wizard TIM provides an interface that accesses the basic local alignment search tool (BLAST) and MODELER programs and guides the user through the necessary steps to easily and efficiently create 3D homology models for biomacromolecular structures. The DockoMatic GUI can be customized by the user, and the software design makes it relatively easy to integrate additional docking engines, scoring functions, or third party programs. DockoMatic is a free comprehensive molecular docking software program for all levels of scientists in both research and education.



AutoDock cycles (ga_run): 100

□ Use AutoDock Vina

□ Use Ligand List File □ Use Modeller

□ Use Secondary List File

□ Use Receptor List File □ Use Modeller

□ Use Box Coord List File

Swarm Jobs Per Node: 1

Swarm Command Options: swarm -f swarmCmd.txt -n 1 -l walltime=128:00:00

□ Job in New Tab

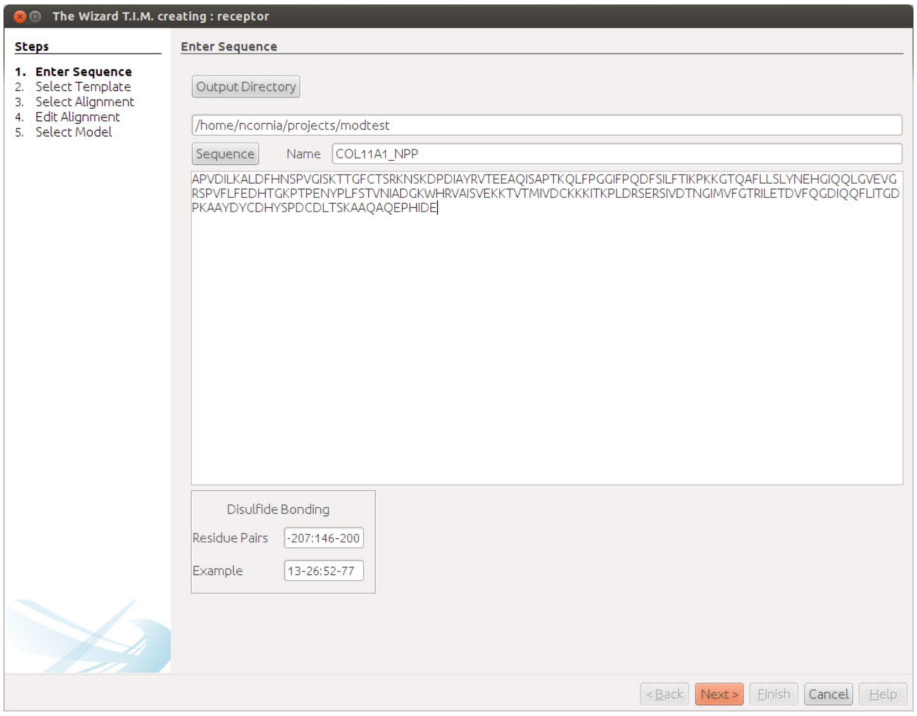
1. BACKGROUND AND MOTIVATION

DockoMatic is an application for virtual peptide ligand creation, molecular docking job setup, initiation and management, and results reporting and analysis.[1,2] Here we report improvements to DockoMatic that allow users to conduct inverse virtual screening (IVS) experiments and to construct and validate homology models for molecular docking. The user may select the computational docking engine used by DockoMatic; currently the program allows access to either AutoDock V 4.2[3] or AutoDock Vina.[4] We also detail the use of Java Netbeans to redesign the graphical user interface (GUI) for user customization of work environment.

High throughput virtual screening (HTVS) is the process by which many ligands are docked to a macromolecular receptor. This is the most common method of computer-aided drug screening and was a feature implemented and evaluated in the first version of DockoMatic.[1] IVS is the complementary process by which one or a few ligands are docked to many receptors.[5] From a computational perspective, this is far more difficult to do because each receptor has a ligand binding domain that must be specified, and the receptor molecules tend to be much larger than ligand files, requiring considerable processing speed, random access memory, and disk storage space for docking studies. Advances in mass spectrometry, X-ray crystallography, and nuclear magnetic resonance (NMR) spectroscopy have led to genomic, proteomic, metabolomic, and other -omics that allow drugs to be screened for activity against desired receptors specific to cell or tissue types. We implemented IVS functionality into DockoMatic and validated its utility by repeating a subset of receptors and ligands originally presented in a study by Lauro et al., where they used a small library of bioactive compounds to screen affinity against protein targets known to be significant in cancer formation.[6] The IVS feature is discussed in greater detail in section 2.1.

Homology model creation is a powerful tool that utilizes information from NMR or X-ray crystallography to predict the structure of biomacromolecules. The most common approach for homology modeling utilizes a combination of the basic local alignment search tool (BLAST) to identify template sequences followed by MODELER to create the homology model from the user-specified template sequence.[7,8] MODELER is a powerful computer program but is difficult to master because of its dependence on Python commands and the use of customized macros to perform basic functions.[9] We have created a GUI interface, which we call the Timely Integrated Modeler (TIM), and incorporated it into DockoMatic to facilitate molecular docking studies where a receptor protein

Received: January 21, 2013



Steps

1. **Enter Sequence**
2. Select Template
3. Select Alignment
4. Edit Alignment
5. Select Model

Enter Sequence

/home/ncornia/projects/modtest

Name: COL11A1_NPP

```
APVDILKALDFHNSPVGISKTTGFCTSRKNSKDPDIAYRVTEEAQISAPTKQLFPGGIFPQDFSILFTIKPKKGTQAFLLSLYNEHGIQQLGVEVG
RSPVFLFEDHTGKPTPENYPLFSTVNIADGKWHRVAISVEKKTVTMIVDCKKKITKPLDRSERSIVDTNGIMVFGTRILETDVFQGDIQQFLITGD
PKAAYDYCDHYSPDCDLTSKAAQAQEPHIDE
```

Disulfide Bonding

Residue Pairs: -207:146-200

Example: 13-26:52-77

Figure 1. Timely integrated modeler (TIM) with the Col (XI) α1 NPP protein. Amino acid sequence provided by the user.

structure does not yet exist.[8] TIM guides the user through all of the steps necessary to construct homology models using BLAST and MODELER. Since MODELER consists of Python modules that are activated through Python scripts, scientists with minimal programming skills face a steep learning curve to customize MODELER's example Python scripts to achieve their model creation goals. For example, challenges that are not easily overcome in MODELER are the creation of disulfide bonds, a fundamental component of protein structure that is often ignored in homology models. The wizard TIM insulates the user from the need to know python and readily addresses disulfide bond creation; exemplified in the broader context of creating the structure of the 223 amino acid amino propeptide domain (NPP) of collagen type XI (Col XI) α1 chain that contains two disulfide bonds.[10] TIM is discussed in greater detail in section 2.2.

We have also ported the user interface code into a Netbeans framework to make it easier for programmers to modify the DockoMatic GUI. The Netbeans integrated development environment (IDE) and GUI builder enables customization of the user interface at the source code level.[11] In addition, this also allows ordinary users to customize and save their DockoMatic desktop environment (window layout, locations, etc.) according to their preferences and desired workflow. The move to Netbeans simplified the DockoMatic source code to enable and ease future development of the program. DockoMatic is open source, freely available on sourceforge, and can be customized by researchers that wish to use the program for their specific investigations. The redesign of the GUI and source code is discussed in section 2.3.

2. METHODS

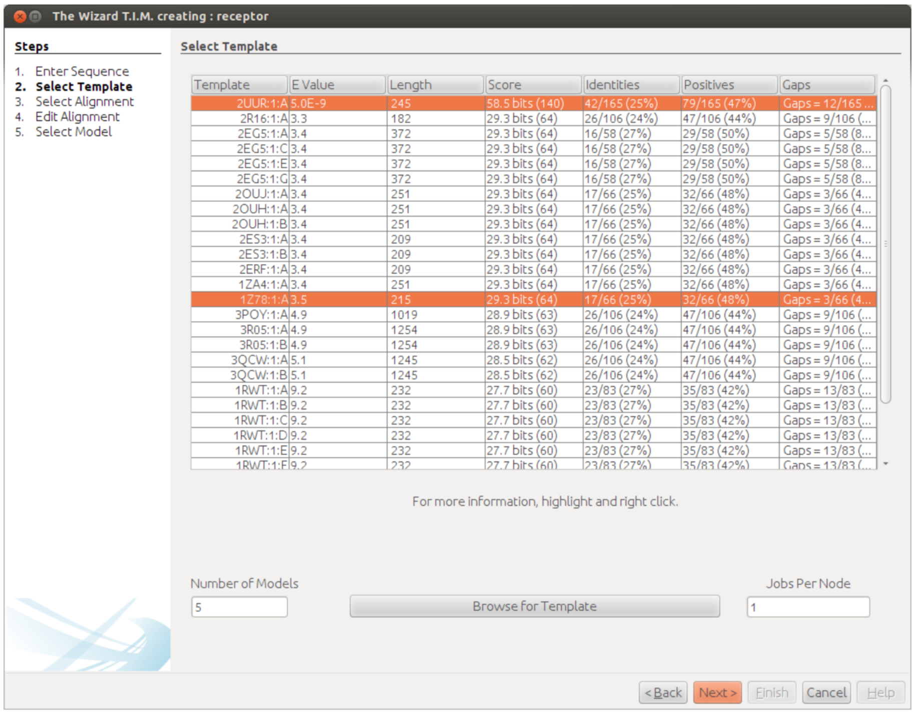
2.1. Inverse Virtual Screening. The HTVS functionality from DockoMatic 1.0 was expanded for IVS in DockoMatic 2.0. To perform HTVS in DockoMatic required the user to provide a ligand list, a receptor protein database (pdb) file, and a grid parameter file (gpf). The new IVS functionality allows the user to perform the following workflow: (1) provide a single ligand pdb file or submit a ligand list with location path to ligand pdb files; (2) enter a receptor list with the location path to the receptor pdb files; and (3) provide a grid box list containing the location path to the corresponding gpf files or text coordinate files associated with each receptor (depending on whether AutoDock or AutoDock Vina is selected, respectively). The user then clicks on the "New Job" button, and DockoMatic populates the docking jobs by performing a cross product of the ligand and receptor/gpf lists. For example, if the user submits two ligands, three receptors, and three gpf's, then a total of six jobs are populated by DockoMatic. If the user mistakenly provides a gpf file that does not correlate with a receptor, DockoMatic will generate an error and reject the jobs.

2.2. Homology Model Integration. A commonly utilized method to build homology models involves the use of BLAST and MODELER in a four step process.

(1) Template search—The user performs a search using the primary amino acid sequence of the protein structure to be created, i.e. the query protein. BLAST compares the sequence of the protein of interest to the sequences of proteins for which structure information is available.[7] The user selects a template structure from the BLAST results based on sequence match, e.g. query coverage, and the likelihood that the sequence randomly matches the template, e.g. *E* value.

(2) Sequence alignment—The MODELER package is then used to generate an initial alignment of the original and the template amino acid sequences using a command line driven script called "align2d", which outputs a protein information resource (PIR) file and a protein alignment parameter (PAP) file. Often, these output files must be manually edited to adjust the initial alignment.

(3) Structure generation—Once the alignment is agreeable, it is used to generate the 3D structure model(s) with



The Wizard T.I.M. creating : receptor

Steps

1. Enter Sequence
2. **Select Template**
3. Select Alignment
4. Edit Alignment
5. Select Model

Select Template

Template	E Value	Length	Score	Identities	Positives	Gaps
2UUR:1:A	5.0E-9	245	58.5 bits (140)	42/165 (25%)	79/165 (47%)	Gaps = 12/165 ...
2R16:1:A	3.3	182	29.3 bits (64)	26/106 (24%)	47/106 (44%)	Gaps = 9/106 (...
2EG5:1:A	3.4	372	29.3 bits (64)	16/58 (27%)	29/58 (50%)	Gaps = 5/58 (8...
2EG5:1:C	3.4	372	29.3 bits (64)	16/58 (27%)	29/58 (50%)	Gaps = 5/58 (8...
2EG5:1:E	3.4	372	29.3 bits (64)	16/58 (27%)	29/58 (50%)	Gaps = 5/58 (8...
2EG5:1:G	3.4	372	29.3 bits (64)	16/58 (27%)	29/58 (50%)	Gaps = 5/58 (8...
2OUJ:1:A	3.4	251	29.3 bits (64)	17/66 (25%)	32/66 (48%)	Gaps = 3/66 (4...
2OUH:1:A	3.4	251	29.3 bits (64)	17/66 (25%)	32/66 (48%)	Gaps = 3/66 (4...
2OUH:1:B	3.4	251	29.3 bits (64)	17/66 (25%)	32/66 (48%)	Gaps = 3/66 (4...
2ES3:1:A	3.4	209	29.3 bits (64)	17/66 (25%)	32/66 (48%)	Gaps = 3/66 (4...
2ES3:1:B	3.4	209	29.3 bits (64)	17/66 (25%)	32/66 (48%)	Gaps = 3/66 (4...
2ERF:1:A	3.4	209	29.3 bits (64)	17/66 (25%)	32/66 (48%)	Gaps = 3/66 (4...
1ZA4:1:A	3.4	251	29.3 bits (64)	17/66 (25%)	32/66 (48%)	Gaps = 3/66 (4...
1Z78:1:A	3.5	215	29.3 bits (64)	17/66 (25%)	32/66 (48%)	Gaps = 3/66 (4...
3POY:1:A	4.9	1019	28.9 bits (63)	26/106 (24%)	47/106 (44%)	Gaps = 9/106 (...
3R05:1:A	4.9	1254	28.9 bits (63)	26/106 (24%)	47/106 (44%)	Gaps = 9/106 (...
3R05:1:B	4.9	1254	28.9 bits (63)	26/106 (24%)	47/106 (44%)	Gaps = 9/106 (...
3QCW:1:A	5.1	1245	28.5 bits (62)	26/106 (24%)	47/106 (44%)	Gaps = 9/106 (...
3QCW:1:B	5.1	1245	28.5 bits (62)	26/106 (24%)	47/106 (44%)	Gaps = 9/106 (...
1RWT:1:A	9.2	232	27.7 bits (60)	23/83 (27%)	35/83 (42%)	Gaps = 13/83 (...
1RWT:1:B	9.2	232	27.7 bits (60)	23/83 (27%)	35/83 (42%)	Gaps = 13/83 (...
1RWT:1:C	9.2	232	27.7 bits (60)	23/83 (27%)	35/83 (42%)	Gaps = 13/83 (...
1RWT:1:D	9.2	232	27.7 bits (60)	23/83 (27%)	35/83 (42%)	Gaps = 13/83 (...
1RWT:1:E	9.2	232	27.7 bits (60)	23/83 (27%)	35/83 (42%)	Gaps = 13/83 (...
1RWT:1:F	9.2	232	27.7 bits (60)	23/83 (27%)	35/83 (42%)	Gaps = 13/83 (...

For more information, highlight and right click.

Number of Models: 5

Jobs Per Node: 1

Figure 2. Results of a BLAST search to identify template structure files.

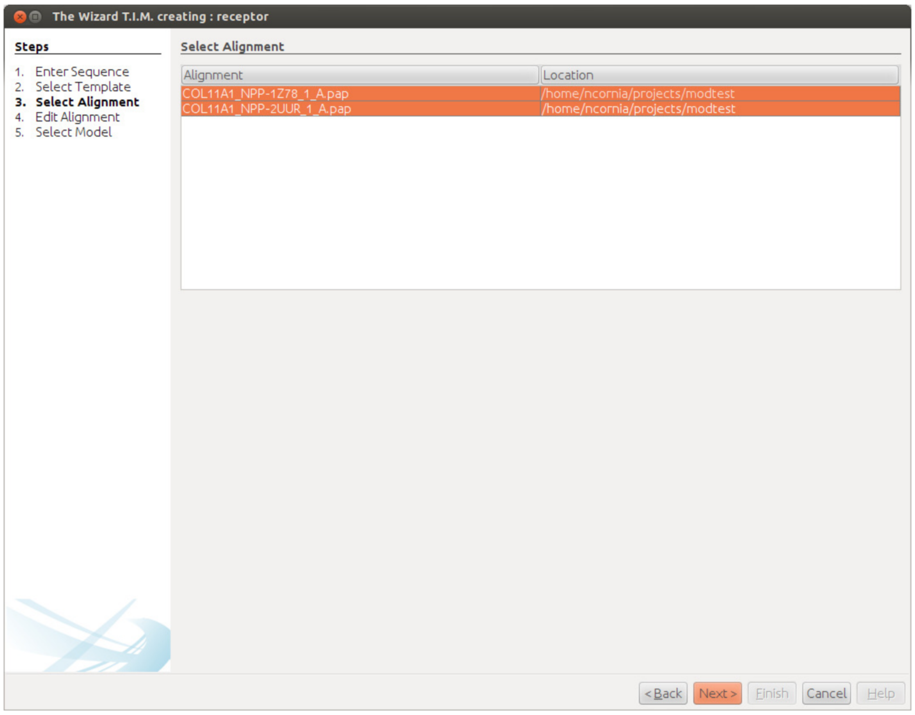
MODELER's automodel class. Typically, several possible 3D models are generated for evaluation by the user.

(4) Structure evaluation—The most suitable model can be determined by the molpdf, discrete optimized protein energy (DOPE), and GA341 output scores generated by MODELER, or by using secondary software packages that assess the 3D structure by means of local environment and geometry of the molecule [http://nihserver.mbi.ucla.edu/SAVES/].[12−15] It is up to the user to select the most desirable model based on these scores.

The "Use Modeler" button in the new DockoMatic GUI generates the receptor structure files required for computational docking experiments. Selection of the "Use Modeler" feature activates the wizard TIM, which guides the user through homology model creation and automatically populates the receptor field with the generated homology model. This allows the user to seamlessly and easily generate desired homology models without having to leave the DockoMatic GUI, and it also shields the user from needing to know the scripts and utilities of yet another software program, in this case MODELER. TIM guides the user through the steps of receptor model creation using five separate windows. This process is described in detail below (sections 2.2.1−2.2.4).

2.2.1. Homology Model Integration: Enter Protein Sequence. The first window in TIM queries the user for the sequence of the target protein. For the purposes of this example, the target protein is the 223 amino acid Col (XI) α1 NPP protein which contains two disulfide bonds between Cys25 and Cys207 and Cys146 and Cys200. The sequence may either be entered directly as single letter amino acid code or by uploading a sequence file in FASTA format (see Figure 1).[16] There are three additional fields: (1) output location; (2) name of sequence; and (3) a field where disulfide bond partners can be specified. Selecting the output location field opens a directory selection window to allow the user to navigate to the desired output directory. The default value for the output location is the current working directory. The sequence name defaults to "myseq" unless the user provides a name for the protein they wish to create. TIM also allows the user to specify disulfide bonds in a homology model by entry of amino acid primary sequence numbers to indicate where disulfide bonds should be located. The user provides input into the disulfide bond dialogue box that includes the first cysteine number, a dash, and then, the second cysteine residue. For example, if the user enters "25−207" then TIM will create a disulfide bond between cysteine 25 and cysteine 207 in the primary sequence of the target protein upon model creation. A colon is used to separate specifications for multiple disulfide bonds. For instance, the structure of Col (XI) α1 NPP contains two disulfide bonds; to specify the location of the disulfide bonds in TIM, the user enters "25−207:146−200". By default TIM does not include disulfide bonds; the inclusion of disulfide bonds into a homology model only occurs if the user explicitly specifies them.

2.2.2. Homology Model Integration: Template Selection. TIM takes the sequence of a peptide or protein for which a structure model is to be built and performs a BLAST search to generate a list of closely matching template structures that have been deposited in the Research Collaboratory for Structural Bioinformatics (RCSB) Protein DataBank (PDB).[17] The list of potential structure templates is displayed with the template name and other parameters to aid in template selection, including *E* value, sequence length, query coverage, amino acid identity, and sequence alignment statistics showing positives and gaps (Figure 2). If more information is desired, a right-click of the mouse provides a popup window with the option to "Open in Browser", which opens the corresponding potential template's information page from pdb.org in the default web browser. For homology models where the protein has quaternary structure, the template will include multiple subunits; generally referred to as A, B, C, etc. For these



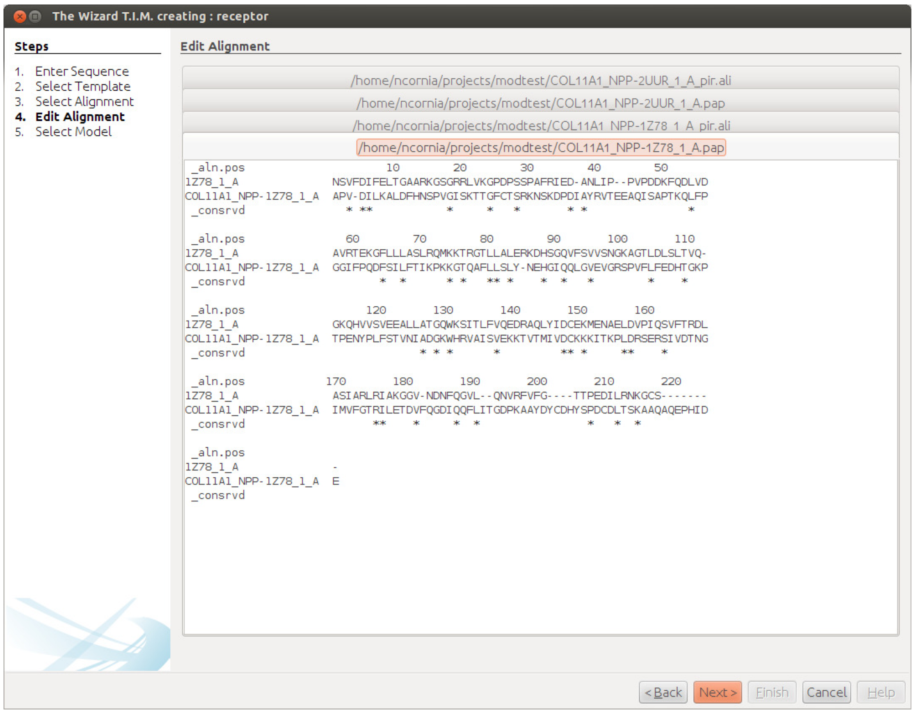
Steps

1. Enter Sequence
2. Select Template
3. **Select Alignment**
4. Edit Alignment
5. Select Model

Select Alignment

Alignment	Location
COL11A1_NPP-1Z78_1_A.pap	/home/ncornia/projects/modtest
COL11A1_NPP-2UUR_1_A.pap	/home/ncornia/projects/modtest

Figure 3. Sequence alignment selection for MODELER structure creation.



Steps

1. Enter Sequence
2. Select Template
3. Select Alignment
4. **Edit Alignment**
5. Select Model

Edit Alignment

/home/ncornia/projects/modtest/COL11A1_NPP-2UUR_1_A_pir.ali

/home/ncornia/projects/modtest/COL11A1_NPP-2UUR_1_A.pap

/home/ncornia/projects/modtest/COL11A1_NPP-1Z78_1_A_pir.ali

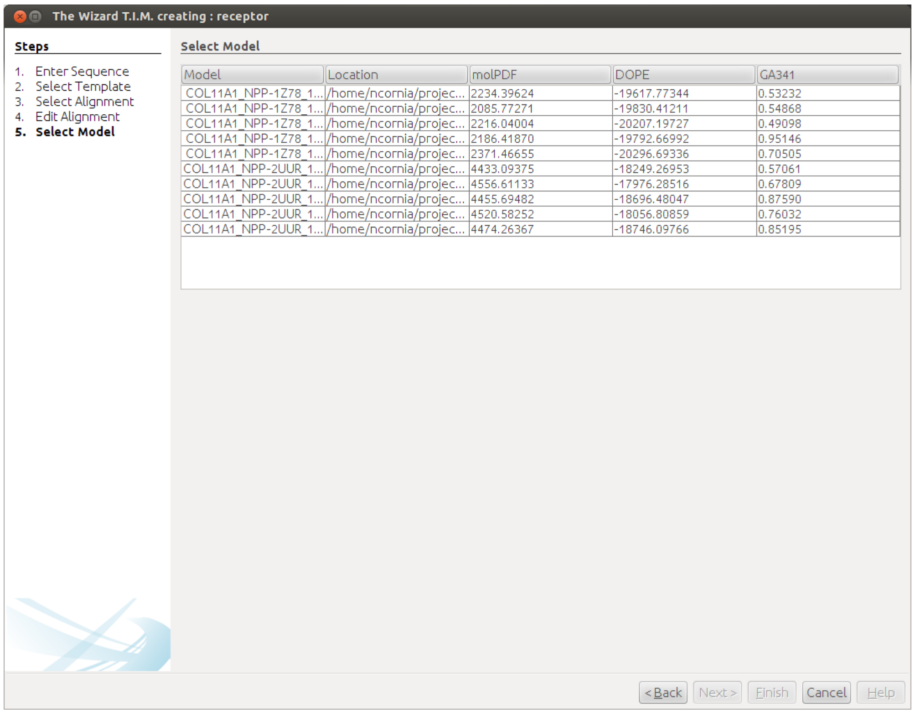
/home/ncornia/projects/modtest/COL11A1_NPP-1Z78_1_A.pap

```
 _aln.pos                     10        20        30        40        50
1Z78_1_A              NSVFDIFELTGAARKGSGRRLVKGPDPSSPAFRIED-ANLIP--PVPDDKFQDLVD
COL11A1_NPP-1Z78_1_A  APV-DILKALDFHNSPVGISKTTGFCTSRKNSKDPDIAYRVTEEAQISAPTKQLFP
 _consrvd               * **           *     *   *       * *               *

 _aln.pos                 60        70        80        90       100       110
1Z78_1_A              AVRTEKGFLLLASLRQMKKTRGTLLALERKDHSGQVFSVVSNGKAGTLDLSLTVQ-
COL11A1_NPP-1Z78_1_A  GGIFPQDFSILFTIKPKKGTQAFLLSLY-NEHGIQQLGVEVGRSPVFLFEDHTGKP
 _consrvd                    *  *      * *   ** *    *  *   *        *    *

 _aln.pos                   120       130       140       150       160
1Z78_1_A              GKQHVVSVEEALLATGQWKSITLFVQEDRAQLYIDCEKMENAELDVPIQSVFTRDL
COL11A1_NPP-1Z78_1_A  TPENYPLFSTVNIADGKWHRVAISVEKKTVTMIVDCKKKITKPLDRSERSIVDTNG
 _consrvd                          * * *      *         ** *     **    *

 _aln.pos            170       180       190       200       210       220
1Z78_1_A              ASIARLRIAKGGV-NDNFQGVL--QNVRFVFG----TTPEDILRNKGCS-------
COL11A1_NPP-1Z78_1_A  IMVFGTRILETDVFQGDIQQFLITGDPKAAYDYCDHYSPDCDLTSKAAQAQEPHID
 _consrvd                   **    *     *  *                * *   *  *

 _aln.pos
1Z78_1_A              -
COL11A1_NPP-1Z78_1_A  E
 _consrvd
```

Figure 4. Sequence alignment editor for MODELER structure creation.

multisubunit proteins, DockoMatic will list the template as many times as there are subunits, specifying a different subunit with each entry to allow the user to select the subunit template they would like aligned with their target protein subunit. In addition to using protein structure file templates from BLAST, the user can browse to an existing template if they have already identified their desired template structure, using the "Browse for Template" button.

When building a homology model with TIM, it is possible to select multiple templates by pressing the control key and selecting the desired rows in the BLAST output table (Figure 2). Each selected template generates a separate MODELER job, which DockoMatic parallelizes across a cluster. The TIM feature allows the user to specify the number of models to be created and the number of processes allowed per cluster node. For example, the default number of homology models is set to



The Wizard T.I.M. creating : receptor

Steps

1. Enter Sequence
2. Select Template
3. Select Alignment
4. Edit Alignment
5. **Select Model**

Select Model

Model	Location	molPDF	DOPE	GA341
COL11A1_NPP-1Z78_1...	/home/ncornia/projec...	2234.39624	-19617.77344	0.53232
COL11A1_NPP-1Z78_1...	/home/ncornia/projec...	2085.77271	-19830.41211	0.54868
COL11A1_NPP-1Z78_1...	/home/ncornia/projec...	2216.04004	-20207.19727	0.49098
COL11A1_NPP-1Z78_1...	/home/ncornia/projec...	2186.41870	-19792.66992	0.95146
COL11A1_NPP-1Z78_1...	/home/ncornia/projec...	2371.46655	-20296.69336	0.70505
COL11A1_NPP-2UUR_1...	/home/ncornia/projec...	4433.09375	-18249.26953	0.57061
COL11A1_NPP-2UUR_1...	/home/ncornia/projec...	4556.61133	-17976.28516	0.67809
COL11A1_NPP-2UUR_1...	/home/ncornia/projec...	4455.69482	-18696.48047	0.87590
COL11A1_NPP-2UUR_1...	/home/ncornia/projec...	4520.58252	-18056.80859	0.76032
COL11A1_NPP-2UUR_1...	/home/ncornia/projec...	4474.26367	-18746.09766	0.85195

Figure 5. TIM output of ten homology models that may be assessed for structure match with the template based on molpdf, DOPE, or GA341 score.

five, and if the user selects two different templates, the end result will be ten models, five for each of the templates.

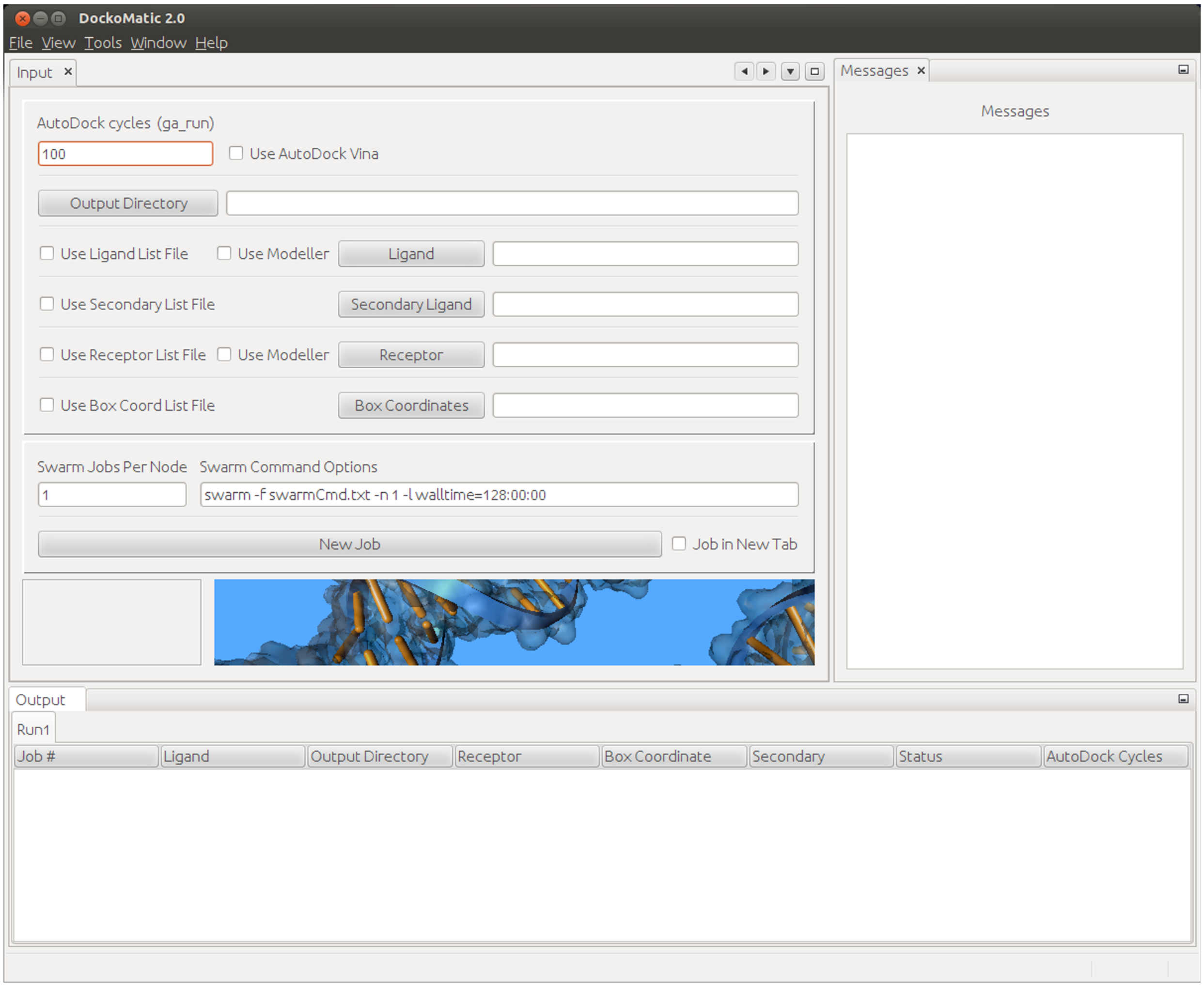
2.2.3. Homology Model Integration: Sequence Alignment. TIM aligns the selected template(s) to the sequences of the desired protein using a modified version of the align2d.py script from MODELER and displays the output (Figure 3). The alignment may be selected for the user to review and edit (Figure 4). TIM allows the user to edit the sequence alignment between the template protein and the desired homology model by manual modification of the PIR and PAP alignment files. TIM's edit feature is accessed through built-in editor tabs (see Figure 4). It is common to adjust the sequence alignment to reflect desired structure elements before generating the final homology model. Generally, the PIR and PAP files are used together to determine what changes to make. For instance, the PAP file is used as a visual reference of the aligned sequences, stacking each sequence on top of the other as they are aligned to easily find locations for adjustments, i.e. addition or deletion of gaps. If a modification is deemed necessary it is important to make the change in the PIR file; the alignment file from which MODELER bases the homology model. It is important to note that any changes to the PAP file are not automatically recorded in the PIR file. If the alignment does not require modification of the PAP or PIR files, the user can simply press the "Next" button to generate the specified number of homology models.

2.2.4. Homology Model Integration: Model Creation and Evaluation. The final step for TIM is the homology model output and comparison to the template (Figure 5). For each of the models created, TIM provides molpdf, DOPE, and GA341 values which may be used to select the best model for study.[18,19] In general, models with lower molpdf scores can be further evaluated for overall spatial quality as assessed by a lower DOPE score. The GA341 value ranges between zero and one, where better models have values closer to one and models that should be discarded have values lower than 0.7.[20] When disulfide bonds are specified in DockoMatic, the models generated by MODELER are output with only the molpdf score. DockoMatic automatically populates the receptor structure input field for molecular docking with the user selected template. The wizard TIM in DockoMatic may be used to create homology models for peptide based ligands and/or receptor molecules.

2.3. New Interface.

The DockoMatic 1.0 GUI was designed for clarity and ease of use, but was limited in functionality to ligand structure creation, efficient job submission to AutoDock, distribution of jobs across a computing cluster, and organization of molecular docking results output from AutoDock.[1−3] DockoMatic 2.0 enables IVS studies, includes the TIM functionality for homology modeling and allows selection of either AutoDock or AutoDock Vina as the docking engine. Design and implementation of these features led to the reorganization, simplification, and modularization of the DockoMatic source code using the Netbeans IDE.[11] The DockoMatic 2.0 interface consists of a main window where the user submits information required for molecular docking calculations, a message window appears to the right side of the screen when jobs are running, and the bottom of the screen shows the output as it is generated from job completion (Figure 6). Thus, the new DockoMatic GUI is segregated into three fields corresponding to input, message/current status of a job, and output.

2.3.1. New Interface: Main Field, Input. Upgrades to the main field were made to enhance the utility of DockoMatic for HTVS and IVS using cluster computing. Components were added to control the number of molecular docking calculations reported, an entry field was added to give direct control to how jobs are distributed across the nodes of a cluster, and addition of the TIM homology modeling wizard. By default, DockoMatic uses "swarm" to submit jobs to a Beowulf cluster. DockoMatic 1.0 could submit one job per cluster node;



AutoDock cycles (ga_run): 100

□ Use AutoDock Vina

□ Use Ligand List File □ Use Modeller

□ Use Secondary List File

□ Use Receptor List File □ Use Modeller

□ Use Box Coord List File

Swarm Jobs Per Node: 1

Swarm Command Options: swarm -f swarmCmd.txt -n 1 -l walltime=128:00:00

□ Job in New Tab

Figure 6. DockoMatic 2.0 graphical user interface consisting of three independent regions: (left) user input fields; (right) running list of job progress reports; and (bottom) output files for jobs that have completed.

DockoMatic 2.0 allows the user to specify, in the "Swarm Jobs Per Node" parameter field, the number of jobs to be sent to each node in the Portable Batch System (PBS) cluster, and there is also space for the user to edit the "Swarm Command Options". In DockoMatic 1.0, it was not possible for the user to choose how many AutoDock job results were generated. AutoDock performs 100 jobs per ligand to receptor binding pair by default. For HTVS or IVS calculations, where hundreds of thousands of jobs may need to be rapidly screened, the amount of data generated is unwieldy. To address this issue, DockoMatic 2.0 permits the user to specify, in the "AutoDock cycles" field, the number of AutoDock jobs to be reported. For example, when performing HTVS/IVS, the user may specify to see only the top ten per 100 most energetically favorable ligand binding poses per trial. To expand the capabilities of DockoMatic, the upgraded version of DockoMatic has additional selection boxes that allow user entry of file lists in the "Ligand", "Receptor", and "Box Coordinates" fields. These upgrades allow for increased experiments such as: multiple docking sites on the entire face of the protein and IVS. The principal improvement to the main input field was the development and implementation of the "Use Modeler" checkbox, which accesses the TIM homology-modeling feature previously described.

2.3.2. New Interface: Message Window. The message window appears to the right side of the screen when jobs are running, and it provides a list of each DockoMatic job that is being processed or has finished. A running report in the message window indicates the population of each job and allows the user to monitor their progress. If there is an error in the job submission or in an individual calculation, the error and a description of the error are displayed in the message window.

2.3.3. New Interface: Output Field. The output field has been relocated to the bottom of the user command box; it has remained largely intact between DockoMatic 1.0 and 2.0 with the exception of the way that jobs are managed. By default, new jobs are populated in the tabbed pane where all input and output information is displayed per job. The user may access an embedded menu for each job by a right mouse click on the job. This produces a menu with the following options: (1) Start; (2) Delete; (3) Analyze; and (4) View Results with Pymol. The pop-up menu was created to minimize the number of extraneous buttons viewed on the control panel.

2.3.4. New Interface: Netbeans Redesign. The DockoMatic 2.0 GUI has been redesigned using the NetBeans IDE.[11] Java NetBeans was chosen as the new development environment for its dedicated integrated development environment that incorporates a powerful GUI builder; its flexibility for GUI design and maintenance, and for the built-in functionality that it

Table 1. IVS Docking Engine Results for Jobs Run in DockoMatic or Independent of DockoMatic Using Either AutoDock or AutoDock Vina as the Docking Engine[a]

aegelinol			clavaminol A			conicaquinone A		
PDB code	AutoDock in DockoMatic	AutoDock	PDB code	AutoDock in DockoMatic	AutoDock	PDB code	AutoDock in DockoMatic	AutoDock
1M61	−4.58 ± 0.107	−5.42 ± 0.036	1M61	−6.18 ± 0.040	−7.15 ± 0.071	1BXL	−8.60 ± 0.030	−8.62 ± 0.044
2FQQ	−4.38 ± 0.506	−5.16 ± 0.020	1Z6T	−6.12 ± 0.046	−6.90 ± 0.032	1M61	−8.39 ± 0.026	−8.45 ± 0.032
1Z6T	−4.28 ± 0.043	−5.00 ± 0.031	1ZY3	−5.89 ± 0.050	−6.74 ± 0.023	1Z6T	−8.22 ± 0.044	−8.33 ± 0.035
1ZY3	−4.20 ± 0.061	−4.78 ± 0.017	cl1bxl	−5.88 ± 0.041	−6.26 ± 0.031	1ZY3	−8.11 ± 0.057	−8.20 ± 0.056
2HYY	−4.16 ± 0.036	−4.67 ± 0.040	2FQQ	−5.67 ± 0.116	−6.12 ± 0.025	2FQQ	−7.84 ± 0.035	−8.00 ± 0.050
2ISI	−4.05 ± 0.068	−4.32 ± 0.035	2HYY	−5.54 ± 0.062	−5.61 ± 0.046	2HYY	−7.71 ± 0.048	−7.62 ± 0.035
1BXL	−3.93 ± 0.071	−4.20 ± 0.021	2ISI	−5.50 ± 0.039	−5.55 ± 0.038	2ISI	−7.29 ± 0.027	−7.44 ± 0.032
2VM6	−3.85 ± 0.044	−4.08 ± 0.025	2VM6	−5.49 ± 0.049	−5.42 ± 0.025	2O21	−7.00 ± 0.025	−6.89 ± 0.046
2W1H	−3.65 ± 0.029	−3.95 ± 0.032	2W15	−5.47 ± 0.067	−5.38 ± 0.057	2VM6	−6.79 ± 0.049	−6.71 ± 0.030
2W15	−3.59 ± 0.056	−3.84 ± 0.040	2W1H	−5.38 ± 0.038	−5.32 ± 0.029	2W1H	−6.68 ± 0.046	−6.54 ± 0.040
2O21	−3.59 ± 0.056	−3.65 ± 0.021	2O21	−5.23 ± 0.040	−5.29 ± 0.021	3C4C	−6.58 ± 0.038	−6.44 ± 0.035
3C4C	−3.32 ± 0.040	−3.29 ± 0.036	3HMI	−5.20 ± 0.062	−5.17 ± 0.055	2W15	−6.51 ± 0.037	−6.33 ± 0.046
3HMI	−3.20 ± 0.045	−3.26 ± 0.038	3C4C	−5.08 ± 0.049	−5.14 ± 0.025	3MVH	−6.44 ± 0.040	−6.32 ± 0.015
3EWT	−3.08 ± 0.039	−3.21 ± 0.063	3EWT	−5.07 ± 0.036	−5.00 ± 0.035	3HMI	−6.27 ± 0.032	−6.24 ± 0.060
3MVH	−2.99 ± 0.035	−3.12 ± 0.055	3MVH	−4.80 ± 0.064	−4.89 ± 0.061	3EWT	−6.22 ± 0.020	−6.12 ± 0.032
PDB code	Vina with DockoMatic	Vina	PDB code	Vina with DockoMatic	Vina	PDB code	Vina with DockoMatic	Vina
2W15	−8.40 ± 0.057	−8.90 ± 0.036	3HMI	−6.40 ± 0.070	−6.34 ± 0.032	3C4C	−9.80 ± 0.052	−10.12 ± 0.032
1Z6T	−8.10 ± 0.078	−8.60 ± 0.060	2HYY	−6.33 ± 0.035	−6.20 ± 0.047	1Z6T	−9.70 ± 0.036	−10.00 ± 0.075
3C4C	−8.00 ± 0.098	−8.12 ± 0.086	3C4C	−5.67 ± 0.035	−5.78 ± 0.040	2W15	−9.40 ± 0.055	−9.70 ± 0.059
3HMI	−8.00 ± 0.095	−7.90 ± 0.085	2W15	−5.63 ± 0.046	−5.61 ± 0.070	1M6I	−8.67 ± 0.015	−9.33 ± 0.046
1M6I	−7.97 ± 0.021	−7.90 ± 0.101	1M6I	−5.50 ± 0.055	−5.52 ± 0.040	2HYY	−8.53 ± 0.031	−9.40 ± 0.055
2HYY	−7.93 ± 0.036	−7.80 ± 0.072	1Z6T	−5.50 ± 0.050	−5.50 ± 0.060	2O21	−8.47 ± 0.015	−8.64 ± 0.025
2ISI	−7.30 ± 0.031	−7.50 ± 0.064	2ISI	−5.50 ± 0.055	−5.45 ± 0.076	3HMI	−8.33 ± 0.031	−8.30 ± 0.056
2W1H	−7.10 ± 0.061	−7.60 ± 0.101	1ZY3	−5.33 ± 0.025	−5.32 ± 0.055	2ISI	−8.10 ± 0.061	−8.10 ± 0.049
2O21	−7.00 ± 0.040	−7.00 ± 0.067	2W1H	−5.13 ± 0.036	−5.00 ± 0.058	2W1H	−8.10 ± 0.067	−8.10 ± 0.082
3EWT	−6.27 ± 0.035	−6.90 ± 0.067	2O21	−5.00 ± 0.045	−5.10 ± 0.066	1ZY3	−6.40 ± 0.059	−6.67 ± 0.045
1BXL	−6.20 ± 0.026	−6.74 ± 0.049	2VM6	−4.37 ± 0.036	−4.54 ± 0.075	2VM6	−6.00 ± 0.061	−6.00 ± 0.055
1ZY3	−5.80 ± 0.047	−5.80 ± 0.072	3EWT	−4.30 ± 0.046	−5.40 ± 0.084	1BXL	−5.80 ± 0.042	−5.54 ± 0.072
2VM6	−5.20 ± 0.053	−5.24 ± 0.042	2FQQ	−3.97 ± 0.065	−3.90 ± 0.080	2FQQ	−5.60 ± 0.049	−5.78 ± 0.061
2FQQ	−5.10 ± 0.040	−4.99 ± 0.045	1BXL	−3.43 ± 0.017	−3.81 ± 0.049	3EWT	−5.50 ± 0.055	−5.40 ± 0.056
3MVH	−3.90 ± 0.063	−3.90 ± 0.114	3MVH	−2.97 ± 0.055	−3.20 ± 0.055	3MVH	−4.50 ± 0.036	−4.40 ± 0.044

[a] Mean binding energy is reported for triplicate trials with standard error values.

provides to programs that are created within its GUI builder framework. The new DockoMatic interface segregates windows by functionality and permits the user to detach, relocate, and resize any window for desktop customization. Users can arrange windows according to their preferences and workflow requirements, and window positioning and size is automatically saved between sessions.

A check box beneath the "New Job" button, when selected, populates the job list into a separate tab. This option is provided to help the user group, track, and organize jobs into related lists. Another enhancement is the specification of how many jobs will be sent to each node in a cluster or single machine. Previously, the default was one job per node; now multiple cores can be utilized, even while running on a single workstation. The number of jobs can be changed using the "Jobs per Node" text field.

DockoMatic submits processing jobs via the swarm command. Swarm is a utility developed at the National Institute of Health for submitting computational jobs on their Beowulf cluster. The swarm utility expects a cluster installed with PBS to be available. However, in order to support users who do not have access to a PBS cluster, we have provided a swarm command file that can be installed alongside DockoMatic for usage on a single workstation.

3. RESULTS AND DISCUSSION

3.1. IVS Results. To test IVS functionality in DockoMatic 2.0 and demonstrate the ability to use different docking engines, we repeated a portion of a published IVS study using 15 receptors involved in cancer processes (PDB accession numbers: 2W15, 1Z6T, 3C4C, 3HMI, 1M6I, 2HYY, 2ISI, 2WIH, 2O21, 3EWT, 1BXL, 1ZY3, 2VM6, 2FQQ, and 3MVH), and three phenolic natural product ligands (aegelinol, clavaminol A, and conicaquinone A).[6] We ran the IVS study using DockoMatic and performed two trials, one with AutoDock Vina as the docking engine and the second using AutoDock V 4.2. To confirm the validity of the IVS results generated from within DockoMatic, we repeated each study using AutoDock Vina and AutoDock V 4.2 independent of DockoMatic. Each experiment was run in triplicate and the results were ranked according to the relative free energy of binding (Table 1). The triplicate trials demonstrate the degree of variability obtained by even the most established docking engines (AutoDock and AutoDock Vina). Multiple trials are recommended to obtain reproducible qualitative molecular

Table 2. User Time Comparison of the Major Steps for Homology Model Creation for Col α1 (XI) NPP Protein, Manual vs DockoMatic

	BLAST Search	
	manual	DockoMatic
time	5 min	1 min
steps	(1) Enter sequence of target protein	(1) Specify the output directory
	(2) Select Protein Data Bank proteins	(2) Enter (paste) the sequence for the Column α1 (XI) NPP protein into TIM
	(3) Run BLAST	(3) BLAST search
	(4) Select the protein sequence with the lowest "*E* value", e.g. N-Terminal Nc4 Domain Of Collagen IX	(4) Select the protein sequence with the lowest *E* value, e.g. N-Terminal Nc4 Domain Of Collagen IX
	(5) Download and save template structure	
	MODELER	
	manual	DockoMatic
time	45 min	4 min
steps	(1) Download and unzip the template structure file	(1) Specify number of models to create and run job
	(2) Create ".ali" file for the target protein	(2) Highlight model file with the lowest molpdf and DOPE scores
	(3) Edit the align2d.py macro for the target protein sequence name, e.g. 2UUR	(3) Protein sequence data will appear to allow user editing
	(4) Run the align code, mod9.9 align2d.py in MODELER	(4) DockoMatic automatically creates homology models of protein
	(5) Create acceptable file labels and save project files in accordance with tutorial instructions.	
	(6) Initiate model creation	

docking results, since a single search samples an insufficient conformational space to get reliable results with either AutoDock or AutoDock Vina. We empircally chose triplicate trials as the minimum number of stochastic runs to sample sufficient conformational space to get reliable results with either AutoDock or AutoDock Vina (Table 1). Not only can the variability in the result of the docking engine be seen, but the utility of DockoMatic to manage many docking calculations required to get reproducible results in a small basis set was also demonstrated.

The structures for the three ligands, aeginol, clavaminol A, and conicaquinone A, were created in MarvinSketch (Marvin Suite V 5.10.3) and saved in pdb file format. Each of the 15 protein receptors sampled were downloaded from the RCSB Protein Data Bank (rcsb.org) and prepared in AutoDockTools V 1.5.4 (ADT), by removing water molecules and adding polar hydrogen atoms with Gasteiger charges.[21,22] For the IVS study using AutoDock Vina, the grid box coordinates specified by Lauro et al. were copied into a text file according to AutoDock Vina input specifications.[6] Ligand, receptor, and box coordinate list text files were entered into DockoMatic 2.0. The default setting for molecular docking jobs in DockoMatic is 100 cycles; this default setting was used. For the IVS study with AutoDock V 4.2, grid parameter files (gpf) were created using the coordinates as detailed by Lauro et al. using ADT.[6] Ligand, receptor, and grid parameter list text files were entered into DockoMatic 2.0. The docking calculations were performed on the Beowulf cluster at Boise State University, which is comprised of 61 nodes with 122, 2.4 GHz Xeon processors, and a three node development cluster with six 2.4 GHz Xeon processors.

To validate the estimated free binding energy results when AutoDock Vina and AutoDock V 4.2 are selected as docking engines in DockoMatic 2.0, the study was repeated using AutoDock Vina and AutoDock V 4.2 independently. These separate trials were set up as stand-alone experiments using the docking engines and associated programs (e.g., ADT) downloaded from the Scripps Institute (scripps.edu) on a personal computer. The same ligands, receptors, and parameters prepared for use in the IVS study facilitated by DockoMatic were used. The studies using AutoDock Vina and AutoDock V 4.2 were run on a personal computer with Windows 7, 64 bit, AMD Athlon 64 X2 Dual Core Processor 6400+ (3.21 GHz), with 4 Gb of DDR2.

The results of this comparative analysis were consistent between the DockoMatic trials and the use of the docking engines independent of DockoMatic usage (Table 1). The small variability in estimated free energy of binding is due to the nondeterministic nature of the optimization algorithms used in AutoDock Vina and AutoDock V 4.2.[4,23] To obtain consistent results, three trials were performed for each of the four experiments. The average binding energy for the triplicate binding experiments provided a direct correlation between all trials. In all but one trial, the qualitative results matched perfectly (i.e., the order of ligands based on binding energy was consistent). The one result that was inconsistent was within the standard error of the triplicate trials. These results are provided as a basis for repetitive trials to be evaluated when performing molecular docking calculations. When performing many repetitive trials, DockoMatic far exceeds manual methods.

The advantages of DockoMatic are realized in experiment setup time, parallel computing across a cluster, and results analysis (Table 2). The user input time required for entering the files used in our IVS study took less than 1 min when using DockoMatic, as compared to 30 min of input time using the stand alone docking engines for the same study. The total run time for the IVS study to complete on a cluster using DockoMatic with AutoDock Vina was less than 10 min, and DockoMatic with AutoDock V 4.2 was less than 15 min. The total run-time, on a personal computer, using AutoDock Vina and AutoDock was on the order of 60 and 90 min, respectively. DockoMatic compiles the ligand to receptor docking calculations into a single dlg file along with a pdb reference file containing the resulting binding poses of input ligands, from lowest to highest binding energy as opposed to compiling the results manually when analyzing multiple jobs using AutoDock Vina and AutoDock V 4.2 independently.[2]

3.2. Homology Modeling Results. To validate the new TIM homology modeling feature in DockoMatic and the successful implementation of Java Netbeans, we used TIM to construct a model for the Col α1 (XI) NPP domain, a 223 amino acid protein that has no reported experimentally determined structure but does have a computationally reported homology model.[10] The process required for a user to create a homology model for the Col α1 (XI) NPP protein using TIM in DockoMatic was compared to a manual approach using the individual software programs BLAST and MODELER. DockoMatic uses these same programs to construct homology models, but the user is insulated from details of their operation by the DockoMatic GUI. Table 2 shows an abbreviated comparison of the time required to perform each step of the homology model creation process by a first time user of BLAST/MODELER versus DockoMatic. A detailed step-by-step procedure is provided in the Supporting Information (Table S1).

For the first time user, creation of the homology model for the Col α1 (XI) NPP protein with TIM in DockoMatic takes approximately 6 min, compared to 50 when using BLAST and MODELER independently. The TIM feature provides an intuitive and user-friendly interface that dramatically simplifies the creation of homology models by leading the user step-by-step through the homology model creation process. Although DockoMatic calls upon BLAST and MODELER, it does so in a way that does not require the user to prepare programming scripts and be familiar with Python code, enabling beginning users to more efficiently create high quality homology models. It is important to note that an experienced user of MODELER can prepare the NPP homology model in approximately 10 min. The time difference is due to familiarity in preparing Python programming scripts. The prepared scripts are generally saved for reuse on future projects. However, even with these advantages an experienced user will still gain a time advantage using DockoMatic to create homology models as compared to using the standalone utilities.

The homology models of the NPP protein produced by the processes given in Table 2, i.e., the homology model produced by the manual process and the homology model produced by the wizard TIM, are identical as determined by a root-mean-square deviation (rmsd) comparison.

3.3. Performance. DockoMatic is intended for use on a computing cluster, where it can efficiently perform HTVS and, now, IVS and homology modeling. The new interface and Netbeans implementation allow homology model job parsing to separate nodes, giving an *n*-fold speed increase over a single workstation. For instance, we have created 100 models in the same time it took to create one, when tested on the Beowulf cluster at Boise State University. The TIM GUI accessed in DockoMatic facilitates use of BLAST and MODELER by beginning users. In addition, creation of disulfide bonds in model structures does not require additional time in DockoMatic. The creation of disulfide bonds in MODELER is not easily achieved by beginning to novice users.

3.4. Dependencies. DockoMatic integrates third party applications in a concerted effort to manage molecular docking. In order to be fully utilized, DockoMatic is dependent on the following: AutoDock Vina,[4] AutoDock, and AutoGrid 4.0,[4] from MGLTools prepare_ligand4.py, prepare_receptor4.py, prepare_gpf4.py, and prepare_dpf4.py,[25] from OpenBabel suite, babel and obconformer,[26] SWARM,[27] MODELER,[28] and PyMol.[12]

4. CONCLUSIONS

DockoMatic 2.0 has been redesigned and enhanced for use in IVS and HTVS studies, homology model creation with the GUI TIM, and it has a new user-customizable architecture created with Netbeans.

We have repeated a published IVS study to demonstrate the ability of DockoMatic to generate consistent results as individual docking engines, AutoDock or AutoDock Vina.[6] This experiment also allowed us to measure the time savings for use of the DockoMatic GUI to automate the process of molecular docking when multiple receptors (15) are bound to several ligands (3) using individual grid parameter files for each receptor. The time savings is significant for experiment setup time, computational processing time across a computing cluster, and result output analysis using DockoMatic 2.0.

We have demonstrated the utility of the DockoMatic TIM GUI to successfully query protein sequences using BLAST, identify potential template molecules, and generate homology models by accessing MODELER. The homology model creation utility is invaluable to a broad base of users desiring to perform traditional HTVS or IVS molecular docking experiments. We have shown linear speedup with regards to the number of models that can be generated for a given experiment or group of experiments. The principle benefits demonstrated for this latest release of DockoMatic 2.0 are ease of use by a first time user and the increase in speed required to generate homology models beginning with the sequence of a protein of interest. Additional bonuses include the flexibility built into the new interface that allows the user to undock and resize windows for customization of their work environment.

Availability and requirements

- Project Name: DockoMatic
- Project home page: https://sourceforge.net/projects/dockomatic
- Operating System: Linux
- Programming Languages: Java, Perl
- License: LGPL

■ ASSOCIATED CONTENT

Ⓢ Supporting Information

Detailed step-by-step procedure of the homology model creation process. This material is available free of charge via the Internet at http://pubs.acs.org.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: OwenMcDougal@boisestate.edu (O.M.M.); tim@cs.boisestate.edu (T.L.A.).

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported by the Defense Threat Reduction Agency under contract number W81XWH-07-1-000, DNA Safeguard. This publication was made possible by NIH Grant P20 RR0116454 and P20 GM103408, the Idaho IDeA Network of Biomedical Research Excellence, Research Corporation Cottrell College Scholars program, and Mountain States Tumor Medical Research Institute. We wish to thank

Drs. Mark Rudin and Greg Hampikian for their support of this project over the years.

REFERENCES

(1) Bullock, C. W.; Jacob, R. B.; McDougal, O. M.; Hampikian, G.; Andersen, T. Dockomatic - automated ligand creation and docking. *BMC Res. Notes* **2010**, *3*, 289.

(2) Jacob, R.; Bullock, C.; Andersen, T.; McDougal, O. DockoMatic: automated peptide analog creation for high throughput virtual screening. *J. Comput. Chem.* **2011**, *32*, 2936−41.

(3) Morris, G. M.; Goodsell, D. S.; Halliday, R. S.; Huey, R.; Hart, W. E.; Belew, R. K.; Olson, A. J. Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. *J. Comput. Chem.* **1998**, *19*, 1639−1662.

(4) Trott, O.; Olson, A. J. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J. Comput. Chem.* **2010**, *31*, 455−461.

(5) Steffen, A.; Thiele, C.; Tietze, S.; Strassnig, C.; Kämper, A.; Lengauer, T.; Wenz, G.; Apostolakis, J. Improved cyclodextrin-based receptors for camptothecin by inverse virtual screening. *Chemistry* **2007**, *13*, 6801−6809.

(6) Lauro, G.; Romano, A.; Riccio, R.; Bifulco, G. Inverse Virtual Screening of Antitumor Targets: Pilot Study on a Small Database of Natural Bioactive Compounds. *J. Nat. Prod.* **2011**, *74*, 1401−1407.

(7) Johnson, M.; Zaretskaya, I.; Raytselis, Y.; Merezhuk, Y.; McGinnis, S.; Madden, T. L. NCBI BLAST: a better web interface. *Nucleic Acids Res.* **2008**, *36*, 5−9.

(8) Marti-Renom, M. A.; Stuart, A. C.; Fiser, A.; Sánchez, R.; Melo, F.; Sali, A. Comparative protein structure modeling of genes and genomes. *Annu. Rev. Biophys. Biomol. Struct.* **2000**, *29*, 291−325.

(9) van Rossum, G.; de Boer, J. Interactively Testing Remote Servers Using the Python Programming Language. *CWI Q.* **1991**, *4*, 283−303.

(10) McDougal, O. M.; Warner, L. R.; Mallory, C.; Oxford, J. T. Predicted Structure And Binding Motifs Of Collagen a1(Xi). *GSTF Int. J. Bioinf. Biotechnol.* **2012**, *1*, 43−48.

(11) Boudreau, T.; Glick, J.; Greene, S.; Spurlin, V.; Woehr, J. *NetBeans: The Definitive Guide*; Definitive Guide Series; O'Reilly: Sebastopol, CA, USA, 2003.

(12) *The PyMOL Molecular Graphics System*, Version 1.3r1; Schrödinger, LLC: New York, 2010.

(13) Pettersen, E. F.; Goddard, T. D.; Huang, C. C.; Couch, G. S.; Greenblatt, D. M.; Meng, E. C.; Ferrin, T. E. UCSF Chimera−a visualization system for exploratory research and analysis. *J. Comput. Chem.* **2004**, *25*, 1605−1612.

(14) Laskowski, R. A.; MacArthur, M. W.; Moss, D. S.; Thornton, J. M. PROCHECK: a program to check the stereochemical quality of protein structures. *J. Appl. Crystallogr.* **1993**, *26*, 283−291.

(15) Arnold, K.; Bordoli, L.; Kopp, J.; Schwede, T. The SWISS-MODEL workspace: a web-based environment for protein structure homology modelling. *Bioinformatics* **2006**, *22*, 195−201.

(16) Lipman, D. J.; Pearson, W. R. Rapid and sensitive protein similarity searches. *Science* **1985**, *227*, 1435−1441.

(17) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28*, 235−242.

(18) Shen, M.-Y. Y.; Sali, A. Statistical potential for assessment and prediction of protein structures. *Protein Sci.* **2006**, *15*, 2507−2524.

(19) Eramian, D.; Shen, M.-y. Y.; Devos, D.; Melo, F.; Sali, A.; Marti-Renom, M. A. A composite score for predicting errors in protein structure models. *Protein Sci.* **2006**, *15*, 1653−1666.

(20) Melo, F.; Sanchez, R.; Sali, A. Statistical potentials for fold assessment. *Protein Sci.* **2002**, *11*, 430−448.

(21) Morris, G. M.; Goodsell, D. S.; Huey, R.; Olson, A. J. Distributed automated docking of flexible ligands to proteins: Parallel applications of AutoDock 2.4. *J. Comput.-Aided Mol. Des.* **1996**, *10*, 293−304.

(22) Saquib, Q.; Musarrat, J.; Siddiqui, M. A.; Dutta, S.; Dasgupta, S.; Giesy, J. P.; Al-Khedhairy, A. A. Cytotoxic and necrotic responses in human amniotic epithelial (WISH) cells exposed to organophosphate insecticide phorate. *Mutat. Res., Genet. Toxicol. Environ. Mutagen.* **2012**, *744*, 125−134.

(23) Scripps. http://www.scripps.com (accessed February 2013).

(24) Boise State Beowulf computing cluster. http://cs.boisestate.edu/~amit/research/beowulf/ (accessed April 2013).

(25) *MGLTools*, Version 1.5.2; The Scripps Research Institute: La Jolla, CA, 2010.

(26) Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An open chemical toolbox. *J. Cheminf. [online]* **2011**, *3*, No. Article 33, ; http://www.jcheminf.com/content/3/1/33 (accessed Nov 19, 2011).

(27) Swarm. http://biowulf.nih.gov/apps/swarm.html (accessed February 2013).

(28) Sali, A.; Blundell, T. L. Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.* **1993**, *234*, 779−815.