Chemistry Central Journal

Software

Open Access

DOVIS 2.0: an efficient and easy to use parallel virtual screening tool based on AutoDock 4.0

Xiaohui Jiang, Kamal Kumar, Xin Hu, Anders Wallqvist and Jaques Reifman*

Address: Biotechnology HPC Software Applications Institute, Telemedicine and Advanced Technology Research Center, U.S. Army Medical Research and Materiel Command, Fort Detrick, MD 21702, USA

Email: Xiaohui Jiang - xjiang@bioanalysis.org; Kamal Kumar - kamal@bioanalysis.org; Xin Hu - xhu@bioanalysis.org; Anders Wallqvist - awallqvist@bioanalysis.org; Jaques Reifman* - jaques.reifman@us.army.mil

* Corresponding author

Published: 8 September 2008

Chemistry Central Journal 2008, **2**:18 doi:10.1186/1752-153X-2-18

Received: 3 July 2008
Accepted: 8 September 2008

This article is available from: http://journal.chemistrycentral.com/content/2/1/18

Abstract

Background: Small-molecule docking is an important tool in studying receptor-ligand interactions and in identifying potential drug candidates. Previously, we developed a software tool (DOVIS) to perform large-scale virtual screening of small molecules in parallel on Linux clusters, using AutoDock 3.05 as the docking engine. DOVIS enables the seamless screening of millions of compounds on high-performance computing platforms. In this paper, we report significant advances in the software implementation of DOVIS 2.0, including enhanced screening capability, improved file system efficiency, and extended usability.

Implementation: To keep DOVIS up-to-date, we upgraded the software's docking engine to the more accurate AutoDock 4.0 code. We developed a new parallelization scheme to improve runtime efficiency and modified the AutoDock code to reduce excessive file operations during large-scale virtual screening jobs. We also implemented an algorithm to output docked ligands in an industry standard format, sd-file format, which can be easily interfaced with other modeling programs. Finally, we constructed a wrapper-script interface to enable automatic rescoring of docked ligands by arbitrarily selected third-party scoring programs.

Conclusion: The significance of the new DOVIS 2.0 software compared with the previous version lies in its improved performance and usability. The new version makes the computation highly efficient by automating load balancing, significantly reducing excessive file operations by more than 95%, providing outputs that conform to industry standard sd-file format, and providing a general wrapper-script interface for rescoring of docked ligands. The new DOVIS 2.0 package is freely available to the public under the GNU General Public License.

Background

Molecular docking is a computational method that predicts how a ligand interacts with a receptor. Hence, it is an important tool in studying receptor-ligand interactions and plays an essential role in drug design. Particularly, molecular docking has been used as an effective virtual screening tool and successfully applied in a number of therapeutic programs at the lead discovery stage [1].

AutoDock [2,3] is a broadly used docking program. Previously, we developed a Linux cluster-based application termed DOVIS [4], which runs in parallel on hundreds of central processing units (CPUs), uses AutoDock 3.05 as the docking engine, and docks large numbers (millions) of ligands to a target receptor. It automatically partitions input ligands, prepares parameter files for AutoDock, launches parallel AutoDock runs, parses results, and saves a set of top-ranking docked ligands. DOVIS removes many technical complexities and organizational problems associated with large-scale high-throughput virtual screening.

During the execution of a number of large-scale virtual screening campaigns using the DOVIS software, we identified four critical areas in which to enhance the software: 1) improving parallelization efficiency, 2) minimizing file operations on a common file system, 3) interfacing with other modeling programs, and 4) facilitating rescoring of ligand-receptor complexes using third-party software. First, in our original parallelization scheme, ligands are evenly distributed to each CPU, i.e., each available CPU receives a number of ligands to dock that is equal to the total number of ligands divided by the number of CPUs requested in the batch-queuing job. This scheme is not efficient if the queuing system does not make all CPUs available at the same time, as in jobs submitted under the "job-array" queuing option, or if the computational time that it takes to dock a ligand is systematically biased because of compound differences, resulting in unbalanced load distribution for each computer node. Therefore, a scheme that automatically balances computational loads across all available CPUs is highly desirable. Second, during docking of each ligand, the original AutoDock program reads the associated energy grids into the corresponding CPU. For large-scale screening jobs, this scheme generates large amounts of input/output (I/O) file operations, which slows down the entire cluster system. It would be ideal to load all energy grids only once to dock an entire block of ligands instead of repeating the I/O file operation for each ligand. Third, we find that directly interfacing docking results with other programs using the native AutoDock pdbq-format is potentially problematic. Although it is possible to convert a pdbq-formatted file into other file formats by OpenBabel [5] or other programs, the converted structures are not guaranteed to preserve bond order, as this information is not recorded in pdbq files. Preservation of bond order and atom sequence among input and output files would not only keep the integrity of a molecule but also make it easier for users to compare docked structures with results from other modeling programs. Last, we note that enrichment of known ligands from virtual screening can be improved by rescoring the docked ligands with additional scoring functions. Hence, it would be beneficial to have an automated protocol in the DOVIS software that is capable of rescoring docked ligands and selecting top-ranking ligands based on these new scores.

Accordingly, we enhanced the DOVIS software along the four directions discussed above by: 1) developing a new parallelization scheme that automatically balances the computational load during runtime; 2) modifying the latest AutoDock source code to reduce I/O file operations; 3) coding algorithms to output docked ligands and bond information in sd-file format; and 4) providing a wrapper script interface to rescore docked ligands with third-party scoring functions. This last functionality also includes a hierarchical clustering method [6] to cluster and save docked ligands based on user-selected scores.

In addition, we upgraded the DOVIS software to use AutoDock 4.0 [7] as the docking engine. Compared to AutoDock 3.05, AutoDock 4.0 has a wider range of force-field atom types, including different atom types for both polar and non-polar hydrogens. This allows a user to better select the appropriate atomic resolution when studying receptor-ligand interactions or performing large-scale docking campaigns. Therefore, we provide three choices for AutoDock hydrogen models in DOVIS: one all-atom model and two polar-hydrogen models.

The significance of the new DOVIS 2.0 software implementation compared with the previous version lies in its improved performance and usability. The previous version of DOVIS made it possible to run large-scale virtual screening in parallel on Linux clusters. The new version makes the computation highly efficient by automating load balancing, significantly reducing the file I/O operations, providing outputs that conform to industry standard sd-file format, and providing a general wrapper-script interface for rescoring of docked ligands. Finally, the DOVIS 2.0 software, including AutoDock 4.0, is freely available to the public under the GNU general public license (GPL).

Implementation

To manage chemical information in DOVIS 2.0, we applied the C++ libraries from OpenBabel and setup a molecular data structure as a C++ object in our program. This makes handling of molecular structures (e.g., atoms and bonds) transparent between applications and outputs consistent chemical information. We coded the new parallelization scheme, the functions to manage AutoDock runs, and the algorithm to process docking results in C++. We used Perl scripts to control the work flow, compute energy grids, and link third-party scoring programs to DOVIS. Besides AutoDock and OpenBabel, Python scripts in AutoDock Tools [8] are also used by DOVIS to prepare

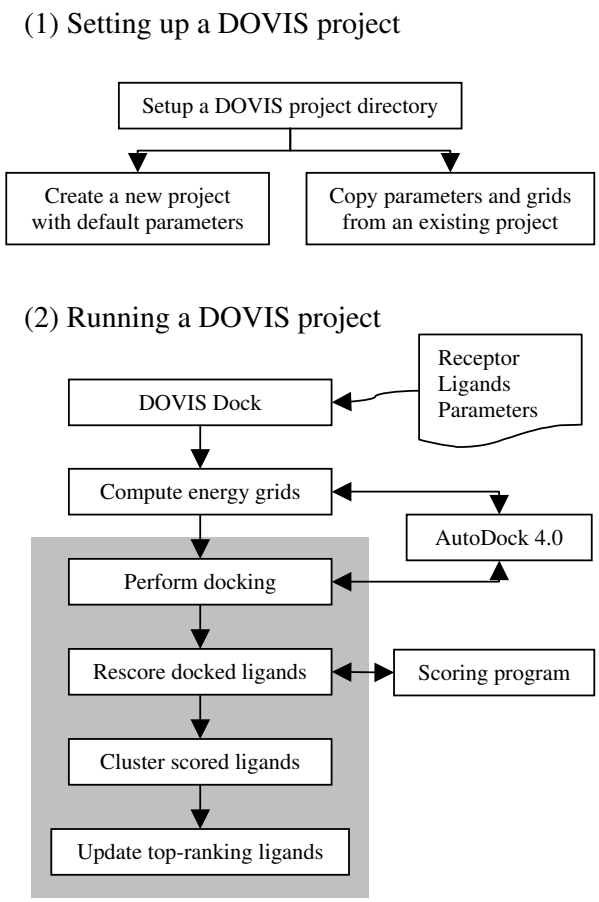

Figure 1
Workflow of DOVIS 2.0. DOVIS is run through a sequential two-step process: (1) setting up a DOVIS project directory and (2) executing a DOVIS docking run. The calculations indicated in the shaded area are run in parallel on multiple central processing units.

receptor and ligand parameter files. Figure 1 summarizes the work flow of DOVIS 2.0.

The DOVIS input file formats for receptors/proteins are either in pdb- or mol2-file format, whereas ligands/small molecules are specified using the sd-file format. The output files contain the docked ligand structure and any auxiliary information in the sd-file format. As shown in Figure 1, a DOVIS run involves a sequential two-step process. Initially, a DOVIS project directory is created that contains subdirectories and appropriate parameter files, either generated with default parameters or cloned from an existing project. Then, the energy grid calculations and the parallel docking processes are launched; the docked ligand-receptor complexes are generated and scored; and lastly, the top-ranking results are saved in the final output.

Detailed usage of DOVIS is documented in the user's manual distributed with the software. Below, we only discuss the new features implemented in DOVIS 2.0.

New parallelization scheme

The components in the shaded area in Figure 1 are run in parallel on the assigned CPUs. We have developed a new parallelization scheme for these components to remove computational bottlenecks identified in the previous version of DOVIS. This scheme implements dynamic job control capabilities and achieves dynamic load balancing without a dedicated master CPU. Thus, before the parallel docking step, input ligands in sd-file format are partitioned into blocks of *N* ligands, where *N* is specified by the user. During parallel docking, each CPU copies the energy grids and other required files to its own temporary directory and requests a block of ligands through a file-lock mechanism. This ensures that each CPU gets one unique set of ligands to work with at a time. After completing a block of ligands, each CPU registers the finished job and updates the top-ranking ligands to the project directory. The CPU then requests another assignment. This process is iterated until all ligand blocks are processed. Finally, an assessment script verifies whether all assignments were successfully completed. Any requested but unfinished ligand block is added back to the original list of blocks to be reprocessed. Users are notified whether all original jobs were successfully completed or a restart is required to complete the docking project.

Since each CPU works on one ligand block at a time and these blocks are continuously requested by the available CPUs during a DOVIS run, by using small block sizes ($N \ll$ total number of ligands) this scheme can provide an effective mechanism for automated, dynamic load balancing. In addition, we provide a mechanism to halt one or more CPUs after their current ligand block is completed. The number of CPUs needed for a restart run can also be changed from run to run.

The current release package provides three approaches to start parallel docking calculations: multithreading, secure shell (SSH), and queuing system. Multithreading and SSH are usually used on small Linux clusters without a queuing system. Multithreading is suited for shared-memory Linux clusters whereas SSH is suited for distributed-memory Linux clusters. For the queuing system approach, by design, DOVIS is capable of running under any queuing system. In this release, we provide an integration with the Load Sharing Facilities (LSF), Platform Computing Inc. (Ontario, Canada), queuing system. However, users may use our integration with LSF as an example to code equivalent CPU management functionalities based on other queuing systems. Because queuing systems may impose runtime limits on jobs, we implemented a mechanism to

estimate if there is enough runtime left in a CPU to process another ligand block every time a new block is requested. An assessment script monitors the overall progress of the docking jobs and, should runtime limits be exceeded, it notifies the user that a restart is required.

AutoDock multiple-ligand docking mode

In the AutoDock program, only one ligand is docked at a time in a docking run, and at each time the associated energy grid files corresponding to every atom type in the ligand are loaded into the corresponding CPU. Depending on the size of the energy grid and the number of atom types, approximately 10 Mb of data are loaded at every docking run for each ligand. Most of the energy grids, however, can be reused from ligand to ligand. Hence, to improve runtime efficiency and reduce I/O file operations, we modified the AutoDock 4.0 source code to load the energy grid files only once, while docking multiple ligands in a single run. In this mode, energy grids of all atom types are loaded and a block of *N* ligands are sequentially docked to a receptor.

Ligand file format

In DOVIS 2.0, we use the sd-file format as the ligand I/O format to represent ligand information and associated data. This format increases the portability of the docking results to other modeling software, as most modeling programs are adopting the sd-file format. Using the sd-file format, we can embed the estimated AutoDock binding free energy and our converted AutoDock 4.0 score (the corresponding pK_i value at 298 K) with each docked ligand. The reason we use the molecular data structure, a C++ object, from OpenBabel's C++ library is to be able to manage chemical information and keep the consistency between input and output chemical structures. In contrast, if we were to use the OpenBabel file-conversion program to convert pdb-files into sd-files, the bonds and bond orders would not be guaranteed to be correct in the converted files. This is because bond information typically does not exist in a pdb-file, and because an agreed-upon, reliable method to consistently predict a bond based purely on inter-atomic distance is not available. Instead, we implemented an algorithm that traces input ligand information, processes AutoDock results, and maps docked ligand atoms back to the input ligand structure. Because an output ligand sd-file is based on the information from a corresponding input ligand sd-file, all atomic information, including bond connectivity, is preserved. If an all-atom model is used (see below, Hydrogen Options), the atoms in the input and the output ligand sd-files have identical sequences.

Hydrogen options

In AutoDock 4.0, both polar and non-polar hydrogens are parameterized. Thus, all atoms including non-polar hydrogens can be explicitly modeled during docking. However, the scoring function in AutoDock 4.0 was developed based on receptor-ligand complexes with only polar hydrogens. In principle, it is necessary to reparameterize the scoring function for the all-hydrogen model. To test the tolerance of the provided default parameters, we docked a set of receptor-ligand complexes using all-atom and polar-hydrogen models. The results are similar (see Results and Discussion) for both models. Therefore, we provide both models as options in DOVIS 2.0. In addition, we coded another option that uses the polar-hydrogen model for docking and then adds non-polar hydrogens to docked ligands for rescoring. This option combines the best of using the original AutoDock scoring function and the high resolution of all-atom modeling. This also makes it convenient to compute pairwise root mean square deviation (RMSD) values between docked and reference ligand structures. It is still recommended that users provide ligands with all hydrogens as inputs even when the polar-hydrogen model is used for docking, as the protonation state of a ligand is clearly represented only when each hydrogen atom is specified. When the AutoDock parameter file is prepared for a ligand, non-polar hydrogens are removed, and their partial charges are combined with the charge of the attached heavy atom. Similarly, receptor energy grids are computed based on either the all-atom or the polar-hydrogen model.

Wrapper interface for scoring program and clustering

It is common practice to apply additional scoring functions to docked ligands after an initial docking run. To facilitate this capability, we provide a wrapper script interface that enables the rescoring of docked ligands with third-party scoring programs. The interface passes a set of predefined parameters, including the working directory, the receptor pdb-file, the docked ligand sd-file, and the name of the scored ligand sd-file to a wrapper script. It also passes customized parameters directly from the DOVIS input file to a wrapper script. This allows users to prepare a wrapper script to drive a scoring program of choice that is linked to DOVIS through the DOVIS input file, where the wrapper script commands are specified.

After rescoring, a separate hierarchical clustering algorithm is available to cluster docked ligand poses based on scores from third-party scoring programs. The clustering algorithm groups docked ligands based on their RMSD values. User-selected scores are employed to determine the cluster centers and to save top-ranking ligands.

Results and discussion

We used eight receptor-ligand complexes from the Protein Data Bank (PDB) [9] to evaluate docking results using both all-atom and polar-hydrogen models with the default AutoDock 4.0 scoring function. Hydrogen atoms

were added to all receptor and ligand input structures. The energy grid center was defined as the geometrical center of the bound ligand in the X-ray receptor-ligand complex. The volume of the energy grid was defined as the volume of the bound ligand in the X-ray complex plus the additional volume mapped out by extending the molecular surface by 4.0 Å in each direction. Other AutoDock parameters were set as follows: five genetic algorithm (ga) runs, each with population size of 150, one million energy evaluations, and a maximum of 27,000 generations per ga run. Pairwise RMSD values were calculated between docked ligand poses and the corresponding X-ray ligand.

Table 1 lists the RMSD values and associated AutoDock 4.0 scores of the docked ligand poses with the lowest RMSD for each of the eight complexes. The RMSD values of the all-atom model and of the polar-hydrogen model are similar for all eight complexes. For all complexes, except 4DFR (RMSD > 5.00 Å), the program found ligand poses close to the experimentally determined ligand pose. Table 1 also indicates that, except for 1RBP, where the score of the all-atom model is much lower (predicting less binding affinity) than the score of the polar-hydrogen model, both models produce similar scores. Figure 2 shows the X-ray ligand and the docked ligand poses from the complexes of 1STP and 1RBP. In the case of 1RBP, we believe that van der Waals (vdW) clashes between the receptor and ligand atoms in the all-atom model caused the difference between the two scores, because, as shown in Figure 2, the binding site of 1RBP is very cramped. In fact, the vdW radii in AutoDock 4.0 were enlarged to compensate for missing non-polar hydrogens. Although our tests suggest that the docking results of both models are generally similar, even when the provided default parameters are used, users should be cautious about using the all-atom model with the default parameters. For a more accurate representation, users should scale down the vdW radii and re-parameterize the AutoDock 4.0 scoring function when applying the all-atom model.

As discussed above, we modified the AutoDock 4.0 source code to implement a strategy where multiple ligands are docked by loading the energy grids only once. The effect on the performance of the file system resulting from this enhancement is dramatic. For a block of 100 ligands, this strategy results in a reduction of more than 95% in data traffic, which is equivalent to a reduction of one Gigabyte of I/O data. This reduction takes place at every CPU. Thus, when hundreds of CPUs are engaged in a virtual screening calculation, the overall impact on transfer rates in the file system is significant. For example, previously on our Linux cluster, we had to use the local disk drive at each computing node to minimize excessive I/O operations. Using the proposed strategy, we may now use more than 256 CPUs of a large cluster to run DOVIS and access data on the cluster's central disk drive without degrading the performance of the file system for other applications. In addition, we tested the time needed for DOVIS to dock a set of 10,000 ligands with varying numbers of CPUs. We performed tests with up to 256 CPUs on a Linux cluster. With the enhancements in the current version of DOVIS, we observed a near-linear speedup up to the 256 CPUs tested using the cluster's central disk drive. This is in contrast with the previous version of DOVIS, which was capable of near-linear speedup to no more than 128 CPUs using the local disk drive.

As an illustration of the technical performance of DOVIS 2.0, using a total of 256 CPUs, we achieved an average throughput of about 670 ligands/CPU/day. This estimate was derived from a virtual screening calculation of 2.3 million ligands from the ZINC database [10] (version 6) against a 259-amino acid protein target. The size of the

Table 1: Docking results for eight Protein Data Bank (PDB) complexes using DOVIS 2.0.

PDB code	Receptor-ligand complex	All-atom model		Polar-hydrogen model	
		RMSD (Å)	Score[a]	RMSD (Å)	Score[a]
186L	Lysozyme(L99A)/n-butylbenzene	0.59	3.33	0.80	4.03
1ABE	Arabinose-binding protein/α-L-arabinose	2.64	4.58	2.33	4.08
1BR6	Ricin A chain/pteroid acid	0.62	5.40	0.75	5.27
1KIM	Thymidine kinase/deoxythymidine	1.17	3.87	0.73	4.31
1RBP	Retinol-binding protein/retinol	1.98	-18.88	1.54	5.92
1STP	Streptavidin/biotin	0.83	5.16	0.81	5.31
3PTB	Trypsin/benzamidine	0.48	3.37	0.48	3.54
4DFR	Dihydrofolatereductase/methotrexate	5.06	4.59	5.03	4.23

Root mean square deviation (RMSD) values were calculated between the docked ligand poses and the X-ray ligand based on heavy atoms. The entries in the table show the lowest RMSD values and associated AutoDock 4.0 scores for docked ligand poses of eight PDB complexes. Typically, the difference in RMSD between the all-atom and the polar-hydrogen model is very small; the average absolute difference is 0.20 Å. [a] AutoDock 4.0 Score

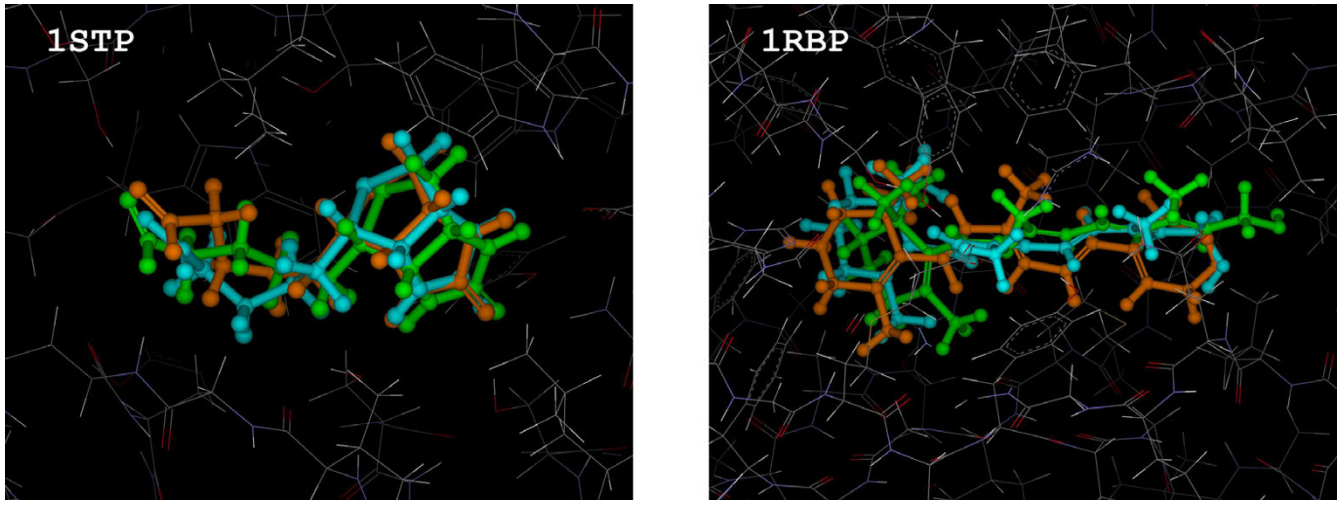

Figure 2
Superimpositions of the X-ray ligand and docked ligand poses to the 1STPand 1RBPcomplexes. The X-ray ligand is shown in green; the docked ligand using the all-atom model is shown in orange; and the docked ligand position derived from the polar-hydrogen model is shown in cyan, where the non-polar hydrogens were artificially added in the representation.

binding site was 28 × 40 × 24 Å. The related AutoDock parameters were set as follows: 10 ga runs, each with population size of 150, 250,000 energy evaluations, and a maximum of 27,000 generations per ga run.

Currently, we are developing automated protocols to perform ensemble docking [11] with DOVIS. The enhanced DOVIS software will take multiple conformations of a receptor, automatically set up multiple sets of energy grids, dock ligands into each receptor conformation, and process the results.

Conclusion

We have enhanced the DOVIS software by improving its performance, screening capability, and usability. DOVIS 2.0 incorporates the most-recent release of AutoDock, AutoDock 4.0, implements a more efficient parallelization scheme, allows for rescoring with user-provided scoring programs, and outputs results in the sd-file format. Furthermore, the software comes with an automatic installer and is available to the public under the GNU GPL.

Availability and requirements

Project name: DOVIS

Operating system: Linux

Programming language: C++, Perl, and Python

License: GNU GPL

Project download: http://www.bioanalysis.org/downloads/DOVIS-2.0.1-installer.tar.gz

Authors' contributions

XJ designed, implemented, and tested the software and drafted the original manuscript. KK implemented and tested the software. XH tested the software. JR and AW participated in the software design, project coordination and manuscript revisions. All authors read and approved the final manuscript.

Disclaimer

The opinions and assertions contained herein are the private views of the authors and are not to be construed as official or as reflecting the views of the U.S. Army or of the U.S. Department of Defense. This paper has been approved for public release with unlimited distribution.

Acknowledgements

We thank Drs. M. Lee, M. Olson and G. Kedziora for helpful suggestions. This work was sponsored by the U.S. Department of Defense High Performance Computing Modernization Program, under the High Performance Computing Software Applications Institutes initiative.

References

1. Ghosh S, Nie A, An J, Huang Z: **Structure-based virtual screening of chemical libraries for drug discovery.** *Curr Opin Chem Biol* 2006, **10(3):**194-202.
2. Goodsell DS, Morris GM, Olson AJ: **Automated docking of flexible ligands: applications of AutoDock.** *J Mol Recognit* 1996, **9(1):**1-5.
3. Morris GM, Goodsell DS, Halliday RS, Huey R, Hart WE, Belew RK, Olson AJ: **Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function.** *Journal of Computational Chemistry* 1998, **19(14):**1639-1662.

4. Zhang S, Kumar K, Jiang X, Wallqvist A, Reifman J: **DOVIS: an implementation for high-throughput virtual screening using AutoDock.** *BMC Bioinformatics* 2008, **9:**126.
5. **OpenBabel** [http://openbabel.sourceforge.net]
6. de Hoon MJ, Imoto S, Nolan J, Miyano S: **Open source clustering software.** *Bioinformatics* 2004, **20(9):**1453-1454.
7. Huey R, Morris GM, Olson AJ, Goodsell DS: **A semiempirical free energy force field with charge-based desolvation.** *J Comput Chem* 2007, **28(6):**1145-1152.
8. **AutoDock Tools** [http://autodock.scripps.edu/resources/adt/index_html]
9. Kouranov A, Xie L, de la Cruz J, Chen L, Westbrook J, Bourne PE, Berman HM: **The RCSB PDB information portal for structural genomics.** *Nucleic Acids Res* 2006:D302-305.
10. Irwin JJ, Shoichet BK: **ZINC – a free database of commercially available compounds for virtual screening.** *J Chem Inf Model* 2005, **45(1):**177-182.
11. Cheng LS, Amaro RE, Xu D, Li WW, Arzberger PW, McCammon JA: **Ensemble-Based Virtual Screening Reveals Potential Novel Antiviral Compounds for Avian Influenza Neuraminidase.** *J Med Chem* 2008, **51:**3878-3894.