Marie-France Sagot
Maria Emilia M. T. Walter (Eds.)

LNBI 4643

Advances in Bioinformatics and Computational Biology

Second Brazilian Symposium on Bioinformatics, BSB 2007
Angra dos Reis, Brazil, August 2007
Proceedings

Springer

Marie-France Sagot
Maria Emilia M. T. Walter (Eds.)

Advances in Bioinformatics and Computational Biology

Second Brazilian Symposium on Bioinformatics, BSB 2007
Angra dos Reis, Brazil, August 29-31, 2007
Proceedings

Springer

Volume Editors

Marie-France Sagot
INRIA Rhône-Alpes, UMR 5558 Biométrie et Biologie Évolutive
Université Claude Bernard, Lyon I
43, Bd du 11 novembre 1918, 69622 Villeurbanne cedex, France
E-mail: Marie-France.Sagot@inria.fr

Maria Emilia M. T. Walter
Universidade de Brasília, Instituto de Ciências Exatas
Departamento de Ciência da Computação (CIC)
Campus Universitário – Asa Norte, Brasília, DF, CEP: 70910-900, Brazil
E-mail: mariaemilia@unb.br

Library of Congress Control Number: 2007931833

CR Subject Classification (1998): H.2.8, F.2.1, I.2, G.2.2, J.2, J.3, E.1

LNCS Sublibrary: SL 8 – Bioinformatics

ISSN 0302-9743
ISBN-10 3-540-73730-8 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-73730-8 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12093663 06/3180 5 4 3 2 1 0

Preface

The Brazilian Symposium on Bioinformatics (BSB 2007) was held in Angra dos Reis (Rio de Janeiro), Brazil, August 29-31, 2007, at the Portogalo Suite Hotel. BSB 2007 was the second BSB symposium, although BSB is a new name for the Brazilian Workshop on Bioinformatics (WOB). This previous event had three consecutive editions in 2002 (Gramado, Rio Grande do Sul), 2003 (Macaé, Rio de Janeiro), and 2004 (Brasilia, Distrito Federal). The change from workshop to symposium reflects the increased quality and interest of the meeting. BSB 2007 was co-located with the International Workshop on Genomic Databases (IWGD 2007).

For BSB 2007, we had 60 submissions: 36 full papers and 24 extended abstracts, submitted to two tracks, computational biology/bioinformatics and applications. The second track was created in order to receive and discuss research work with a biological approach, and so to reinforce the participation of biologists in the event. These proceedings contain 13 full papers that were accepted, plus 6 extended abstracts. These papers and abstracts were carefully refereed and selected by an international Program Committee of 48 members, with the help of some additional reviewers, all listed on the following pages. We believe that this volume represents a fine contribution to current research in computational biology and bioinformatics, as well as in molecular biology.

The editors would like to thank: the authors, for submitting their work to the symposium, and the invited speakers Roded Sharan (Tel-Aviv University, Israel), Alberto Martín Rivera Dávila (Fundação Oswaldo Cruz, Brazil) and João Paulo Kitajima (Allelyx Applied Genomics, Brazil); the Program Committee members and the other reviewers for their support in the review process; the General Chair Sérgio Lifschitz and the local organizers Daniel Xavier de Sousa, Cristian Tristão and José Maria Monteiro; the symposium sponsors (see list in this volume); Nalvo Franco de Almeida Jr., João Carlos Setubal, José Carlos Mombach, Marcelo de Macedo Brígido, and again Sérgio Lifschitz, members of the Brazilian Computer Societys (SBC) special committee for computational biology; and Springer for agreeing to print this volume.

August 2007

Marie-France Sagot
Maria Emilia M. T. Walter

Organization

BSB 2007 was organized by the department of Informatics - Pontifical Catholic University of Rio de Janeiro/Brazil.

Executive Committee

Conference Chair	Sérgio Lifschitz Pontifical Catholic University of Rio de Janeiro Brazil
Local Arrangements	Daniel Xavier de Sousa Cristian Tristao Jose Maria Monteiro Pontifical Catholic University of Rio de Janeiro Brazil

Scientific Program Committee

Program Chairs	Marie-France Sagot INRIA, France *Computational Biology and Bioinformatics*
	Maria Emilia Machado Telles Walter University of Brasilia, Brazil *Applications*

Program Committee

Said S. Adi	(Federal University of Mato Grosso do Sul, Brazil)
Nalvo F. Almeida	(Federal University of Mato Grosso do Sul, Brazil)
Alberto Apostolico	(Accademia Nazionale dei Lincei and Georgia Tech)
Fernanda Baiao	(UNIRIO, Brazil)
Valmir C. Barbosa	(Federal University of Rio de Janeiro, Brazil)
Ana Lúcia Bazzan	(Federal University of Rio Grande do Sul, Brazil)
Marcelo M. Brígido	(University of Brasilia, Brazil)
Edson N. Cáceres	(Federal University of Mato Grosso do Sul, Brazil)
André P. L. F. de Carvalho	(University of São Paulo-São Carlos, Brazil)
Maria Cláudia Cavalcanti	(Military Institute of Engineering, Brazil)
Dominique Cellier	(University of Rouen, France)

Laurent Dardenne	(National Laboratory of Scientific Computation, Brazil)
Alberto M. R. Dávila	(Fiocruz, Brazil)
Zanoni Dias	(University of Campinas, Brazil)
Carlos E. Ferreira	(University of São Paulo, Brazil)
Ana T. Freitas	(Technical University of Lisbon, Portugal)
Richard Garrat	(University of São Paulo-São Carlos, Brazil)
Raffaele Giancarlo	(Universita degli Studi di Palermo, Italy)
Katia S. Guimarães	(NCBI, USA/ Federal University of Pernambuco, Brazil)
David Huson	(University of Tübingen, Germany)
João P. Kitajima	(Alellyx, Brazil)
Gunnar Klau	(Freie Universität Berlin, Germany)
Gad Landau	(University of Haifa, Israel)
Melissa Lemos	(Pontifical Catholic University of Rio de Janeiro, Brazil)
Natalia Martins	(Embrapa/Biological Resources and Biotechnology, Brazil)
Wellington Martins	(Catholic University of Goias, Brazil)
Marta Mattoso	(Federal University of Rio de Janeiro, Brazil)
Alba C. M. A. Melo	(University of Brasilia, Brazil)
Antonio B. Miranda	(Fiocruz, Brazil)
Satoru Miyano	(The University of Tokyo, Japan)
José C. Mombach	(Federal University of Santa Maria, Brazil)
Nadia Pisanti	(University of Pisa, Italy)
Alexandre Plastino	(Federal Fluminense University, Brazil)
Leila Ribeiro	(Federal University of Rio Grande do Sul, Brazil)
David Sankoff	(University of Ottawa, Canada)
Luiz F. Seibel	(Pontifical Catholic University of Rio de Janeiro, Brazil)
João C. Setubal	(Virginia Bioinformatics Institute, USA)
Roded Sharan	(Tel-Aviv University, Israel)
David Sherman	(CNRS, France)
Siang W. Song	(University of São Paulo, Brazil)
Marcílio C. P. de Souto	(Federal University of Rio Grande do Norte, Brazil)
Osmar Norberto de Souza	(Pontifical Catholic University of Rio Grande do Sul, Brazil)
Guilherme P. Telles	(University of São Paulo-São Carlos, Brazil)
Cristina Vieira	(INRIA, France)
Sérgio Verjovski-Almeida	(University of São Paulo, Brazil)
Martin Vingron	(Max Planck Institute, Germany)
Michael Waterman	(University of Southern California, USA)
Fernando von Zuben	(University of Campinas, Brazil)

Additional Reviewers

Christian Baudet
Markus Bauer
Luciano Digiampietri
Alan Mitchell Durham
Cristina G. Fernandes
Ivan Gesteira Costa Filho
Alexandre Paulo Francisco
Ronaldo Fumio Hashimoto
Dennis Kostka
Alair Pereira do Lago
Helena Cristina Gama Leitão
Ana Carolina Lorena
Simone de Lima Martins
Mariá Cristina Vasconcelos Nascimento
Christian Rausch
Christine Steinhoff
Yoshiko Wakabayashi

Sponsoring Institutions

Brazilian Computer Society
CAPES

Table of Contents

Selected Articles

Extended Abstracts

Automating Molecular Docking with Explicit Receptor Flexibility Using Scientific Workflows

K.S. Machado, E.K. Schroeder, D.D. Ruiz, and O. Norberto de Souza

Laboratório de Bioinformática, Modelagem e Simulação de Biossistemas - LABIO
Programa de Pós-Graduação em Ciência da Computação, Faculdade de Informática, PUCRS,
Porto Alegre, RS, Brasil
{kmachado,eschroeder}@inf.pucrs.br,
{duncan,osmar.norberto}@pucrs.br

Abstract. Computer assisted drug design (CADD) is a process involving the execution of many computer programs, ensuring that the ligand binds optimally to its receptor. This process is usually executed using shell scripts which input parameters assignments and result analyses are complex and time consuming. Moreover, receptors and ligands are naturally flexible molecules. In order to explicitly model the receptor flexibility during molecular docking experiments, we propose to use different receptor conformations derived from a molecular dynamics simulation trajectory. This work presents an integrated scientific workflow solution aiming at automating molecular docking with explicit inclusion of receptor flexibility. Enhydra JAWE and Shark software tools were used to model and execute workflows, respectively. To test our approach we performed docking experiments with the *M. tuberculosis* enzyme InhA (receptor) and three ligands: NADH, IPCF and TCL. The results illustrate the effectiveness of both the proposed workflow and the implementation of the docking processes.

1 Introduction

One of the most important features of Bioinformatics is the collection, organization and interpretation of a large amount of information [1]. To carry that out, different computational tools have to be used to manage different data elements in a particular sequence of computational steps. Generally these kinds of experiments, called *in silico*, involve a sequential execution of a number of computer programs, where the output of one is the input for the next.

Usually these programs are executed in a one-by-one basis either manually or by simple shell scripts specially designed for this purpose. Often, one has also to face problems with the heterogeneous and distributed nature of the generated data due to particular input/output formats from the available tools [2]. Furthermore, manual execution of computational programs or the use of shell scripts to execute them usually lead to problems related to computer program diversity of usage, data flow recording and process maintenance. An interesting approach to model these characteristic problems is by using scientific workflows [3]. These workflows involve sequences of analytical steps that can deal with database access, data mining, data analysis, and many other possible steps involving computationally intensive jobs.

M.-F. Sagot and M.E.M.T. Walter (Eds.): BSB 2007, LNBI 4643, pp. 1–11, 2007.

From Biology we know that macromolecules (receptors), such as proteins, enzymes, DNA, and RNA, are not rigid entities in their cellular environment. Therefore, it is highly desirable that this flexibility be explicitly considered during a computer assisted drug design (CADD) process. One important step of the CADD procedure is the molecular docking, where the binding of a small molecule (ligand) to its receptor is computationally tested and evaluated.

Docking experiments can be performed by a number of docking simulation software [4]. Most of them can deal with ligand flexibility, but have difficulties in handling the receptor flexibility. Those capable of handling receptor flexibility do it only in a limited way [5]. In order to overcome this problem and include a more realistic representation of the receptor flexibility during docking experiments, we considered an ensemble of thousands of receptor conformations, generated by molecular dynamics (MD) simulations. For each receptor's possible conformation, one ligand docking experiment has to be performed and analyzed. These steps are currently being executed manually, where the appropriate parameters (such as the protein and ligand names, the names of the files from the MD trajectory, the number of MD snapshots, the docking parameters, etc.) as well as the sequence of execution are defined by the user. Consequently, to re-execute a process with different receptor and/or ligand, the user would probably face serious difficulties to adjust the input parameters and data files. In addition, following and registering all execution processes are not simple tasks.

This article aims to model and automate the molecular docking process so as to explicitly include the receptor flexibility, and analyze their results. To accomplish that, we developed a scientific workflow, modeled using the JAWE design tool [6] executed by the Shark workflow engine [7]. The docking software AutoDock3.05 [4] and the PTRAJ module of the AMBER6.0 package [8], herein called PTRAJ only, are executed by scripts and computer programs written to perform each one of the activities described in the workflow.

This article is organized as follows: Section 2 presents a review of some basic concepts, important to understand this work, like the CADD process, molecular docking and MD simulation; Section 3 presents the developed scientific workflow, explaining each activity one by one; Section 4 illustrates the application of the workflow with a case study, and finally, Section 5 discuss possible improvements to the current implementation.

2 CADD, Molecular Docking, MD Simulation and Scientific Workflows

2.1 CADD, Molecular Docking and MD Simulation

New developments in structural and molecular biology and computer simulation tools over the past years have made possible a more accurate rational drug design (RDD) [9]. RDD involves a set of four steps [10]:

1. The target receptor (usually a protein) structure is analyzed using its 3D structure to identify probable binding sites;

2. Based on such probable binding sites, a set of possible ligands is selected and the receptor-ligand interactions can be tested and evaluated by simulations using a docking software;
3. The ligands that theoretically had the best interaction score to the receptor are selected, bought or synthesized, and then experimentally tested;
4. Based on the experimental results, a possible inhibitor is detected or the process returns to step 1.

Steps 1 and 2 constitute the CADD process. During the molecular docking (step 2), the ligand molecule assumes different orientations and conformations inside a defined binding pocket or region of the receptor and their interactions are systematically tested and evaluated (Figure 1a). A large number of evaluations has to be performed in order to identify the best ligand orientation and conformation inside the binding pocket. This information is computed in terms of the free energy of binding (FEB – the more negative, the more effective is the ligand-receptor association).

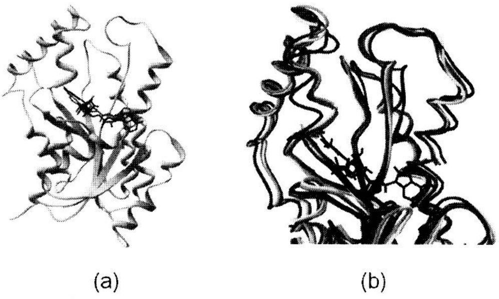

Fig. 1. The docking process. (a) The ligand molecule (in cyan and magenta) in two different orientations inside its InhA receptor (gray) binding pocket. (b) Superposition of five different *Mycobacterium tuberculosis* enzyme InhA conformations (cyan, yellow, magenta, red and green), generated by MD simulations [11], representing the flexibility of InhA bound to NADH (small molecule in blue). See section 3.1.

As ligands are usually small molecules, the different conformations they can assume inside the binding pocket are easily simulated by the docking software [4]. However, the limitation generally occurs when one wants to consider the receptor flexibility. There are a number of alternatives to incorporate at least part of the receptor mobility, but the use of many receptor structures has been characterized as the best alternative [5]. Therefore, one way to simulate the receptor flexibility is to use an ensemble of receptor conformations generated by MD simulations [12].

According to Sali [13], the studies of biological systems were initially limited to observation and interpretation of experimental data. The evolution of experimental techniques has allowed a deeper view of the biological processes by accessing the structural properties of biological macromolecules. These properties, in turn, can be deeply investigated by using the MD simulation methodology which simulates the molecular natural movements of biological molecules in atomic detail [14]. The result of a MD simulation is a series of instantaneous conformations or snapshots denominated the MD simulation trajectory. The InhA enzyme [15] from *Mycobacterium tuberculosis* has been shown to be considerably flexible [11] and was chosen to be our model of receptor (Figure 1b).

2.2 Scientific Workflows

According to the Workflow Management Coalition (WFMC) [16], workflow is "the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules". Although this definition refers to "business process", workflow is not only employed by business applications. Wainer et al. [3] state that workflows can be also classified as ad-hoc workflows and scientific workflows.

Scientific workflows generally gather and merge data from various experiments, generating data from a computer model or performing data statistical analysis. In addition, scientific workflows can not be completely defined before it starts. While some tasks are being executed, one has to decide the next steps after evaluating the previous one [17]. Thus, the lack of complete knowledge about the processes in scientific applications has implications on modeling scientific workflows. The main assumption is that the models are inherently incomplete or change at any time [18].

As the molecular docking with respect to receptor flexibility is a scientific application, composed by a number of different software tools, in the present work we employed a scientific workflow management system to integrate modeling and execution of docking processes.

3 The Molecular Docking Workflow Model

Before the development of this scientific workflow all of the work had to be manually performed with FORTRAN computer programs and shell scripts. We developed our scientific workflow to model and run the docking processes considering the receptor flexibility explicitly, which is not a trivial task in ligand-receptor docking experiments [5]. The complexity of the receptor is usually the limiting issue. Receptors contain far more atoms than ligands, and therefore a very large number of degrees of freedom must be taken in account.

We adopted the Kua et al. [19] alternative approach to consider the receptor flexibility in docking experiments: to perform a series of dockings using, in each one of them, one different receptor snapshot. In our work, the receptor snapshots were generated by MD simulations [11] with the AMBER6.0 [8] package.

The flowchart of the developed workflow model is schematically shown in Figure 2. The activities in dashed lines correspond to those executed by the user, whilst the ones in solid lines are executed by the system without user intervention. The JAWE [6] and Shark [7] software tools were used to model and execute the workflow, respectively. These software tools are free, and can be used in the Linux environment (as well as AMBER6.0 and AutoDock3.05).

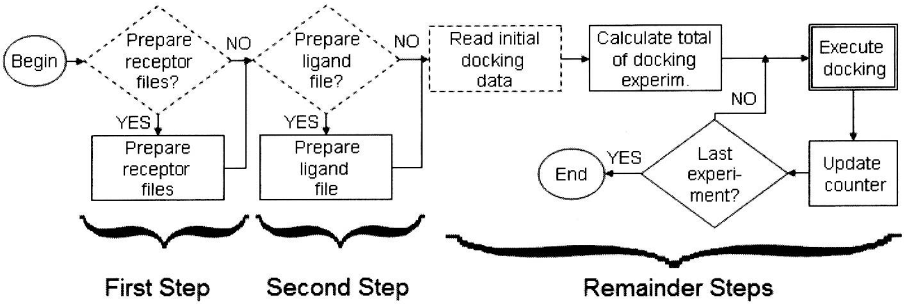

Fig. 2. Flowchart of the proposed molecular docking workflow model

3.1 Step Zero – MD Simulation of the Receptor

The first action in a molecular docking considering the explicit receptor flexibility is the execution of a MD simulation of the receptor, from which a series of receptor snapshots is generated. This step is called step zero because it is not modeled in Figure 2, since it is performed only once for each receptor.

The InhA enzyme from *Mycobacterium tuberculosis* has been shown to be considerably flexible [11] and was chosen to be our model of receptor (Figure 1b) in this work. Its explicit flexibility was obtained from a fully solvated MD simulation trajectory generated by the SANDER module of AMBER6.0 [8] as previously described [11]. MD data were collected for 3,100 ps (1.0 ps = 10^{-12}s) and instantaneous snapshots were saved at every 0.5 ps, in files of 50 ps each. A total of 6,200 receptor snapshots were generated.

3.2 First Step – Prepare Receptor Files

This step has two parts: execution of PTRAJ and selection of snapshots relevant to the docking process. This step may not be necessary if the docking experiment only employs a single receptor.

PTRAJ is a AMBER6.0 utility that converts the trajectory of receptor snapshots generated by MD simulation into the PDB format [20]. A computer program was developed to establish the communication between Shark and PTRAJ. During workflow execution, the user must inform some parameters, such as the number of receptor amino acid residues and the first and last snapshots to be considered. These data, used as input parameters for PTRAJ, are stored into a workspace to be used during workflow execution. Thus, the user can easily change any input parameters and there is no need to modify individual scripts.

During the MD simulation, instantaneous snapshots were recorded at every 0.5 ps. Consequently, after 3,100 ps of MD simulation, 6,200 snapshots are generated and, hence, 6,200 PDB files are generated by PTRAJ. However, consecutive snapshots have closely related conformations. In order to trim down the redundancy of conformations we picked up snapshots separated by time intervals larger than 0.5ps. In our case study (see Section 4) we chose 1ps as the time interval to select snapshots and a total of 3100 snapshots were used in the docking experiments.

3.3 Second Step – Prepare Ligand File

This step has two parts. First, the ligand is placed in its initial position within the binding pocket of the receptor. Second, the proper ligand file is generated. This step is performed only once if the docking experiment employs the same ligand-receptor pair.

To place the ligand in its initial position within the binding pocket, the ligand PDB file and the receptor's average structure are automatically opened, by Shark, in the SwissPDBViewer [21]. The ligand is then manually placed by the user in the receptor structure.

Afterwards, the ligand PDB file needs to be transformed into a PDBQ format [4]. A ligand file in MOL2 format needs to be supplied as the input file. This can be basically done in two ways: through proprietary software such as MOE [22], or by downloading a MOL2 file from freely available public databases of small molecules. We developed a computer program that uses such a pre-existing MOL2 file and replaces its ligand coordinates for those correctly positioned in the receptor binding pocket. The module deftors of AutoDock3.05 software is then used to generate the PDBQ file from the MOL2 file.

3.4 Remainder Steps – Execution of the Docking Experiments

After the preparation of the receptor and ligand files, the docking experiments can be executed (Remainder Steps in Figure 2.). The flowchart in Figure 3 details the **Execute docking** subflow from Figure 2.

The docking experiments can be executed using the whole MD trajectory or only part of it. In the workflow the user is asked to inform the initial and final snapshots to be considered and a counter value, which indicates the next experiment to be performed (the counter value was introduced to prevent a restart from the beginning in case of workflow execution failure). As made for ligand and receptor files preparation, computer programs and shell scripts were developed to establish communication between the workflow and each of the AutoDock3.05 modules (Addsol, Mkgpf3, Mkdpf3, Autogrid and Autodock).

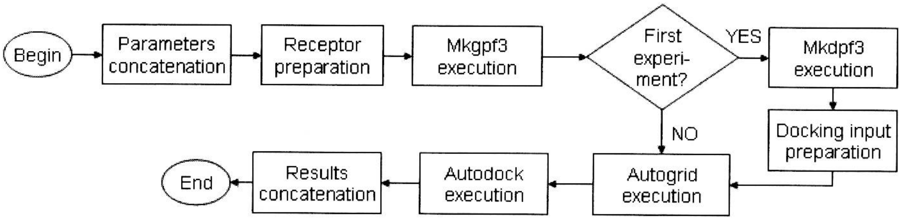

Fig. 3. Flowchart of the subflow Execute docking that executes the docking experiments

In Figure 3, the activity **Parameters concatenation** concatenates the counter value with execution file names from AutoDock3.05. The activity **Receptor preparation** generates the receptor in the PDBQS format using Addsol. **Mkgpf3 execution** generates the "Input.gpf" file, which contains the input parameters for Autogrid execution. **Mkdpf3 execution**, when executed, generates the Autodock parameters file “Input.dpf”. Then, in **Docking input preparation**, “Input.dpf” is edited within a text editor and the user can easily modify the docking parameters. Subsequent executions of the subflow do not need to perform these activities.

The activity **Autogrid execution** executes the Autogrid module where the grid maps are defined for each ligand atom type. During **Autodock execution** the module Autodock is executed to calculate an estimate of the interaction between ligand and receptor in terms of the free energy of binding (FEB). At the end of the docking run, an output file is generated. This file contains information about all the tested ligand conformations, and the results are organized according to the best final docked energy (FEB) and the ligand root mean square deviation (RMSD) from the initial position.

The last activity, **Results concatenation**, collect the current docking energies (FEB) and RMSDs results and stores them in a results’ list. An excerpt of this list is shown in Table 1, where each line represents the results of one docking experiment. This activity also compresses the Autodock output (to save disk space) and deletes the unnecessary files. All computer programs and scientific workflows for flexible receptor docking experiments were executed on Pentium III PCs of 1GHz and 256 MB RAM.

Table 1. Example of a list of results for the flexible InhA receptor-IPCF docking

Time (ps)	**Snapshot**	**RMSD (Å)**	**FEB (Kcal/mol)**	**Autogrid Execution Time (min.)**	**Autodock Execution Time (min.)**
1	2	6.3	-9.9	4:50.02	10:22.50
2	4	6.2	-10.2	4:06.81	10:08.61
...	...	...	...	...	...
3099	6198	3.9	-9.9	4:39.90	9:41.25
3100	6200	4.0	-9.7	4:30.58	10:00.94

4 Case Study

The validation of the proposed scientific workflow was carried out by performing docking experiments with the *Mycobacterium tuberculosis* enzyme InhA as the receptor and one large InhA ligand (NADH [15]), and two small ones, IPCF [23] and TCL [24].

4.1 The *M. Tuberculosis* Enzyme InhA

The InhA enzyme from *Mycobacterium tuberculosis* is the bona-fide target for one of the most important drugs (isoniazid) used in tuberculosis treatment. It was shown that this enzyme needs a NADH molecule as a cofactor for enzymatic activity. The activated isoniazid binds to the NADH molecule inside the receptor binding pocket to inhibit its activity [25], leading to mycobacterial death. It was shown that IPCF [23] and TCL [24] also interact with InhA, inhibiting its activity.

Knowing that the InhA enzyme constitutes a flexible receptor [11] and a ligand should bind to it in more than one enzyme conformation, and in order to understand these ligands affinities for the binding site, we developed a scientific workflow that automates the fully flexible molecular docking study to identify the characteristics of those ligand-InhA associations (see Section 3). The ligands molecules (NADH, IPCF and TCL) were docked in a number of different InhA (receptor) conformations previously generated by MD simulations [11].

4.2 Experiments

We performed three experiments, one for each ligand, to validate our implementation of the proposed workflow. The three ligands used in the experiments are shown in Figure 4.

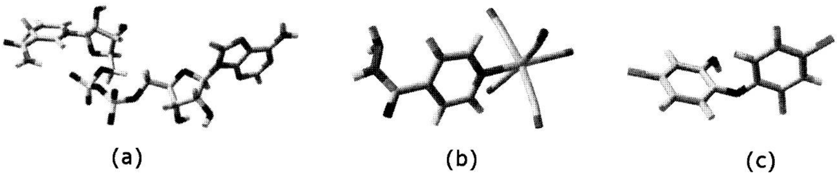

Fig. 4. Stick models of the three-dimensional structure of three InhA ligands: (a) NADH, (b) IPCF and (c) TCL. The ligands atoms are colored by type: Carbon (gray), Nitrogen (blue), Oxygen (red), Hydrogen (cyan), Phosphorus (yellow), Iron (orange), and Chlorine (green).

The same MD simulation trajectory of the receptor was used in all three experiments. Ligand docking to each of the 3,100 receptor snapshots was performed by the simulated annealing protocol including 10 runs with 100 cycles each, a total of 25,000 steps accepted or rejected, with selection of the ligand conformation presenting the minimum FEB. All docking processes and results concatenation were performed, with no human intervention, by the developed scientific workflow. The docking results are reported as in Table 2.

4.2.1 Docking Experiments with the NADH Ligand

After the receptor files preparation from the MD simulation snapshots, the NADH molecule, a large ligand (Figure 4a) presenting 52 atoms was generated through user interaction with the scientific workflow described above. The ligand was initially placed inside the receptor binding pocket, and its PDBQ file was prepared. As all receptor snapshots were superimposed, the initial ligand position, and therefore its PDBQ file, was the same for all 3,100 receptor snapshots considered.

4.2.2 Docking Experiment with the IPCF Ligand

The IPCF ligand molecule, containing 28 atoms, was prepared and initially placed in the receptor binding pocket as described for the NADH ligand in Section 4.2.1. Furthermore, the execution of the first step of the scientific workflow was not necessary since the receptor docking files had already been generated for the NADH docking.