J Comput Aided Mol Des (2011) 25:813–824
DOI 10.1007/s10822-011-9465-6

# VSDMIP 1.5: an automated structure- and ligand-based virtual screening platform with a PyMOL graphical user interface

**Álvaro Cortés Cabrera · Rubén Gil-Redondo · Almudena Perona · Federico Gago · Antonio Morreale**

Received: 15 June 2011 / Accepted: 1 August 2011 / Published online: 9 August 2011


**Abstract** A graphical user interface (GUI) for our previously published virtual screening (VS) and data management platform VSDMIP (Gil-Redondo et al. J Comput Aided Mol Design, 23:171–184, 2009) that has been developed as a plugin for the popular molecular visualization program PyMOL is presented. In addition, a ligand-based VS module (LBVS) has been implemented that complements the already existing structure-based VS (SBVS) module and can be used in those cases where the receptor's 3D structure is not known or for pre-filtering purposes. This updated version of VSDMIP is placed in the context of similar available software and its LBVS and SBVS capabilities are tested here on a reduced set of the Directory of Useful Decoys database. Comparison of results from both approaches confirms the trend found in previous studies that LBVS outperforms SBVS. We also show that by combining LBVS and SBVS, and using a cluster of ~100 modern processors, it is possible to perform complete VS studies of several million molecules in less than a month. As the main processes in VSDMIP are 100% scalable, more powerful processors and larger clusters would notably decrease this time span. The plugin is distributed under an academic license upon request from the authors.

Á. C. Cabrera · F. Gago
Departamento de Farmacología, Universidad de Alcalá, 28871 Alcalá de Henares, Madrid, Spain

Á. C. Cabrera · R. Gil-Redondo · A. Perona · A. Morreale (✉)
Unidad de Bioinformática, Centro de Biología Molecular Severo Ochoa (CSIC-UAM), Campus UAM, c/Nicolás Cabrera 1, 28049 Madrid, Spain
e-mail: amorreale@cbm.uam.es

**Keywords** Docking · Virtual screening · Drug design · Graphical user interface

## Introduction

Many changes in the drug discovery paradigm have emerged in recent years due to major advances in the field of Computer-Aided Drug Design (CADD), which has benefited enormously from astounding improvements in the power of computers and new algorithms. As a result, attempts continue to be made to turn the drug discovery process into a more rational approach that can help design therapeutically relevant New Molecular Entities (NME) with a minimum of synthetic effort. Another important factor that has to be taken into the complex drug making equation is the vast amount of experimental information emanated from genome sequence and structural biology projects, as well as biochemical and biophysical studies, that is stored in publicly accessible databases. Precisely because of this complexity, CADD appears to be placed, at least conceptually, in an excellent position to help reduce the cost and time that it takes to launch a NME onto the market (a thousand million dollars and 15 years on average, respectively [1]). However, despite some signs of promise, the real results still fall far below the expectations [2].

CADD approaches include structure-based (SB) and ligand-based (LB) virtual screening (VS) of chemical (and fragment) libraries, and both methods are widely used in industry and academia. SBVS uses docking tools with the aim of identifying possible hits that can then be subjected to lead optimization. To this end, they are routinely tested for their capacity to (a) reproduce the experimental structures of a series of ligands bound to their receptor targets,

as found in high-resolution X-ray crystal structures, and (b) discriminate between true binders and fake ligands ("decoys") on the basis of a scoring function that, although far from accurately representing the free energy of binding that can be measured experimentally [3], is used to predict the strength of the receptor-ligand association. If the docking engine and the scoring function perform reasonably well in this respect, one can expect some success in the identification and ranking of putative hits in a VS experiment. As an alternative, and particularly in cases where the receptor's 3D structure is not available, it is also possible to use the geometry of one or more ligands that display affinity for this receptor as a query to try and fish out similar molecules from commercially available catalogues or databases. The selected compounds can then be tested experimentally for confirmation of affinity/activity.

Besides the core SBVS and LBVS algorithms, an integrated platform for VS studies needs some other pieces of software, the most important being those required for setting up receptors and ligands at the beginning of the procedure and for processing the results at the end. In addition, large databases are usually filtered according to some custom-made rules. The integration of all of these tools into a common, flexible, and user-friendly platform requires a great deal of effort because a series of *connectors* have to be developed to handle the existing variety of file formats. Besides, an adequate database engine needs to be used to store and process efficiently the massive amounts of data that are generated, in common with trends observed in other computational biology areas [4].

The growing interest in this type of computational platforms that put together all the essential pieces to enable the effortless execution of complex VS protocols has resulted in a number of applications. Some solutions are commercial, like the Schrödinger [5] and Sybyl [6] suites and Pipeline Pilot from Accelrys [7, 8]), but open-source plugins for the popular molecular graphics program PyMOL [9] have also been designed and released, e.g. the intuitive and user-friendly interfaces to widely used software such as AutoDock/Vina [10] or AMBER [11]. Furthermore, other implementations are accessible through a web server (DOCKBLAST [12]), distributed over a grid [13], or endowed with database capabilities [14]. In our lab, VSDMIP [15] was developed to provide the scientific community with a flexible, fully automated computational platform to perform VS experiments and manage every piece of data in an integrated fashion. Significant advantages of this platform are its underlying database, which stores ligand information and every result arising from the different steps of a given VS protocol, and its modular and pluggable architecture, which allows customization of each step of the procedure. However, the original VSDMIP only allowed SBVS to be done and worked through a command-line interface.

In this paper we describe the improvements that have been incorporated into the updated version (VSDMIP 1.5) to overcome these shortcomings: (1) an LBVS module has been built that can be used not only in cases where the receptor's 3D structure is unavailable but also as a complement to SBVS applications, and (2) a graphical user interface (GUI), written in Python programming language, that allows its facile use as a plugin to the popular molecular visualization program PyMOL. VSDMIP 1.5 is compared to similar existing software and its LBVS performance on a subset of the Directory of Useful Decoys (DUD) database [16] is reported. As with the original VSDMIP, we are committed to making this updated and more powerful platform available free of charge to academic and non-profit organizations so that the scientific community, and eventually society at large, can benefit from it.

## VSDMIP extensions

### LBVS: the newly added functionality

#### *Molecular fingerprints*

Molecular fingerprints are strings of bits denoting the presence (1) or absence (0) of certain types of molecular information, typically chemical groups or relevant interaction points. They can be 2D or 3D depending on the structural information encoded. VSDMIP allows the user to work with both 2D (MACCS [17], CATS [18], and chemical groups for filtering) and 3D fingerprints (triplets of interaction points).

*2D fingerprints* The Molecular Design Ltd. (MDL) Molecular ACCess System (MACCS) structural fingerprint is a 166-bit string that indicates whether a predefined substructure or functional group is present or not.

The Chemical Advanced Template Search (CATS) fingerprint is composed of a bit for each possible combination of hydrogen bond donor, hydrogen bond acceptor, positively charged group, negatively charged group, and lipophilic PharmacoPhoric Points (PPP), including aromatic rings, separated by distances between 1 and 10 bonds and totalling a length of 150 bits.

Finally, a group fingerprint is a 306-bit string that denotes the presence or absence of different chemical fragments and functional groups (see Open Babel documentation for details [19]). The use of this fingerprint is indicated as a post-filter after using MACCS or CATS to ensure that the selected compounds do possess the required functional groups.

In the three cases, the fingerprints can be calculated either from a given database or from a file containing a set of Simplified Molecular Input Line Entry Specification (SMILES) strings [20] using Open Babel [19], which is

also integrated within VSDMIP. In the database the fingerprints are stored in the FINGERPRINT table (Fig. 4). Also, a previously saved file containing molecule IDs and fingerprints can be loaded and stored within the database.

*3D fingerprints* 3D fingerprints can be defined using the 3D molecular structure and six types of PPP with the following interaction properties: hydrogen bond acceptor, hydrogen bond donor, positively charged group, negatively charged group, aromatic ring, and lipophilic point. These PPP, associated in triplets, can be automatically calculated for all the molecules in a given database. The generator of triplets (GTP) code recognizes all possible PPP for each conformer in the database and builds the triplets in hexadecimal strings representing the type and the relative disposition of the PPP. This information is stored in the PHARMACOPHORES table (Fig. 4). Additionally, the user can create customized 3D fingerprints by choosing the type of points on the graphical interface and moving them to a desired position. This fingerprint can then be used as a pattern to search for molecules in databases that fulfil these conditions. Finally, the molecules obtained from a search can be incorporated directly into the main workflow of SBVS.

*Fingerprint comparisons*

VSDMIP incorporates three coefficients for fingerprint comparison (Tanimoto, Tversky, and rule-based) as well as two mechanisms to combine queries (hybrid fingerprints and scoring fusion).

*Tanimoto coefficient* Given two objects, A and B, represented as two strings of bits, the Tanimoto coefficient, *Tc*, is defined as the ratio (Eq. 1):

$$Tc = c/(a + b + c) \quad (1)$$

where *a* is the count of *on* bits in object A but not in object B, *b* is the count of *on* bits in object B but not in object A, and *c* is the count of *on* bits in both objects A and B

It can be viewed as the ratio of *on* bits shared by both string representations. The values range between 0 (no similarity at all) and 1 (identical fingerprints).

*Tversky coefficient* It introduces the concept of a *prototype* to which the objects or *variants* are compared to and is defined as the following ratio:

$$T_v = c/(\alpha * a + \beta * b + c) \quad (2)$$

where *a*, *b*, and *c* have the same meaning as before, and $\alpha$ and $\beta$ are weighting factors for the *prototype* and the *variant* so as to customize the relative importance of one *versus* the other. The $T_v$ coefficient is also bound between 0 and 1.

*Rule-based coefficient* This is a ratio calculated as the result of the AND operation between the query string and the comparing counterparts, and therefore takes into account only those bits that are activated in the query string.

*Mechanisms to combine queries: hybrid fingerprints versus scoring fusion*

When more than one query is at hand (i. e., several known actives, to select some specific characteristics, or inactives, to rule out other non-desired properties), two options are available: (a) to combine the queries themselves, or (b) to combine their individual results. For the former, VSDMIP implements the centroids module, which analyses a set of query compounds and generates a fingerprint that concentrates all the information present in the whole set. A bit is activated if it is already present in a certain percentage of the compounds (the cut off is 0.5 by default but can be adjusted manually). Centroids can then be used like a regular fingerprint to query the database. For the latter, once multiple searches have been performed, common fusion scoring schemes are employed over the individual scoring values obtained: maximum, minimum, product, average, and the sum of the scores. These schemes have been implemented via a user-defined function (UDF), and it should be possible to include new operations easily. Finally, combined queries of active and inactive compounds could be useful for detecting ambiguous molecules: those giving good results when looking for actives and also good results when looking for inactives.

## VSDMIP graphical user interface

Considering the advantages, for non-expert users, of incorporating a simple and easy-to-use interface to interact with the VSDMIP platform, we decided to create a plugin for the popular and versatile PyMOL graphics program as a tool to control the complete VS workflow (either LBVS or SBVS, or any combination thereof).

Once the plugin is invoked from the PyMOL plugin interface, it displays a simple window with a menu bar containing six submenus, the description of the program and the version number (Fig. 1). The menu bar includes: SBVS, LBVS, local operations (non-dependent on the database facilities, Local), database information (Info), configuration (Config) and information about the program, the authors, and contact details (About).

Under the SBVS submenu available options are:

(a) Insert new molecules, to incorporate new SMILES strings into the database,
(b) Run Virtual Screening, to create a new VS job and submit it to a cluster or to a multiprocessor machine,

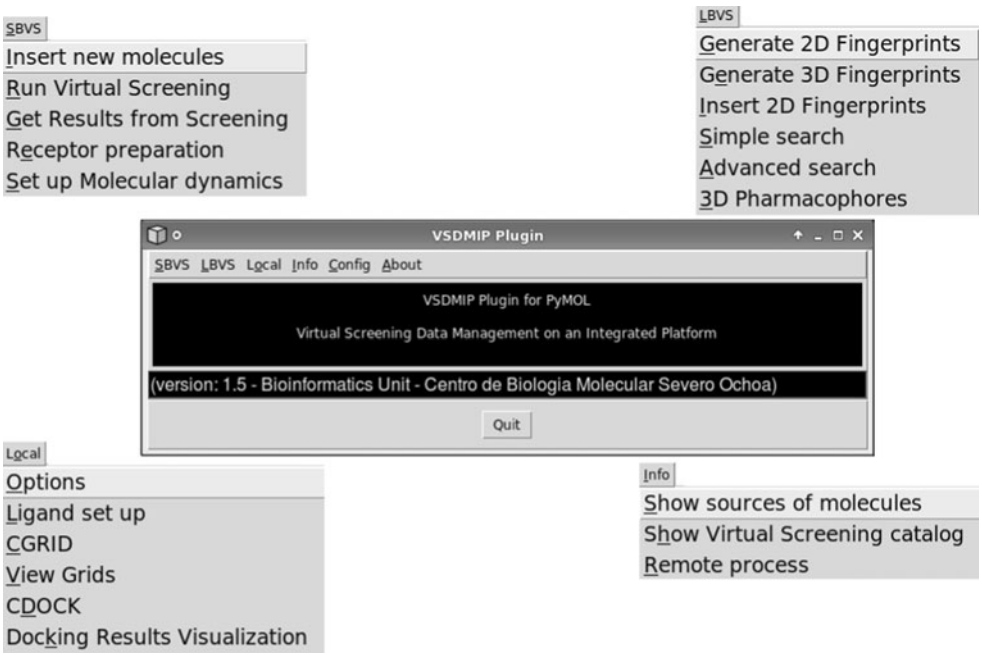


**Fig. 1** VSDMIP's main window (*centre*) as it is launched from PyMOL, and windows that originate from the tabs that give access to the SBVS, LBVS, Local and Info tasks

(c) Get Results from Screening, to extract the results from a previous screening (structures, energy values, and general information about the interactions),
(d) Receptor preparation, to simplify the protein preparation workflow through a local script, and
(e) Set up Molecular Dynamics, to generate the topology [top] and coordinates files [crd] that are necessary to run a molecular dynamics simulation using the AMBER suite (http://ambermd.org/) [21].

The procedure for the latter operation, which makes use of AMBER's antechamber module, involves the mandatory creation of the ligand-related parameter and connectivity files [frcmod] and [prepin], the immersion of the complex in a user-adjustable cube of Transferable Intermolecular Potentials 3 Point (TIP3P) water molecules [22], and the addition of any necessary counterions to achieve electroneutrality.

The LBVS submenu gives access to the novel options for performing complete VS experiments using fingerprints of different types:

(a) Generate 2D Fingerprints, calculation of 2D fingerprints;
(b) Generate 3D Fingerprints, to generate triplets of PPP using the automated tool runGTP;
(c) Insert 2D Fingerprints, to insert the generated fingerprints into the database;
(d) Simple Search, to perform a single search using simple parameters;
(e) Advanced Search, to use special techniques for performing searches like scoring fusion, substructure search or chemical group filtering; and
(f) 3D Pharmacophores, to generate, within the PyMOL GUI, newly defined PPP to be used in new searches.

The Local submenu allows the user to carry out a complete docking process, from the conformational analysis of the ligand to the final visualization of the results. The available options are:

(a) Options, to configure the paths of the different programs;
(b) Ligand Set Up, to prepare the ligands for ALFA calculations (conformational analysis) and atomic charge assignment;
(c) CGRID, to calculate the energy grids for docking with CDOCK;
(d) View grids, to visualize the grids;
(e) Grid processing, to post-process the interaction energy grids by Boltzmann averaging (as a way to include receptor flexibility for docking [23]) or by calculating a grid as a difference of two other previously calculated grids (for example, on two related targets, as a way to explore selectivity [24]);
(f) CDOCK, to set up the docking process; and
(g) Docking Results Visualization, to analyse docking results (different energy terms and the type of interactions).

The Info submenu contains Show source of molecules and Show VS catalogue, two windows in which the user can look up information regarding the putative ligands and the VS protocol, respectively, as they are stored within the database (to be used later on as part of MySQL queries); and Remote process, to monitor the processes for which

instructions have been issued (the exact command is set up in the Config submenu).

The Config submenu contains the Configure queues tab that gives access to a window for defining the paths for the ssh and scp protocols, remote system and commands, and the MySQL settings.

Finally, the About submenu displays the name and affiliation of the authors, a contact address for further information, and the copyright statement.

Figure 2 illustrates the implementation of the VSDMIP GUI in PyMOL and how results from a docking run with CDOCK [25] can be visualized.

## A case study

### The test set: directory of useful decoys database

Eighteen targets (ACE, MR, HIVPR, P38 MAP, HMGA, PNP, COMT, Thr, TK, fXa, AChE, HSP90, COX1, COX2, AMPC, ALR2, GPB, and $ER_a$) were selected from the DUD database [16] ensuring enough diversity of types. Their bound ligands were downloaded directly from the original site and processed according to our established protocol. In short: a) conversion of all compounds into their isomeric SMILES [26] strings (to meet with the defaults in the VSDMIP protocol as described in the original work) and insertion into the database, which implies their transformation from 2D to 3D with CORINA [27], assignment of atom point charges using the Austin Model 1 electrostatic potential (AM1/ESP) fitting method, as implemented in MOPAC 7 [28], addition of AMBER [21] atomic radii, and conformational analysis with ALFA [29]; and b) calculation of CATS, MACCS and group fingerprints with a modified version of the Open Babel program (decimal output and a parser for CATS were added), and insertion in the extended database. Importantly, this processing means that the original geometry of the bound ligand is lost and that each molecule will be present in the database as a collection of ready-to-dock conformers. For the protein targets PDB2PQR [30] was used to assign AMBER force field atomic radii and charges, while the protonation states of titratable residues were decided on the basis of $pK_a$ calculation carried out with the PROPKA software [31]. The binding site to be explored was delimited in each case by the location of the bound ligand in the X-ray crystal structure using CGRID [25].

### Virtual screening

In LBVS, we have used the searching functions described above to query the database multiple times (as many as the number of actives) and retrieve similar compounds taking the Tanimoto coefficient as the score. The global performance for a given target was evaluated as the mean (over all its actives) of the area under the curve (AUC) values from receiver operating characteristic (ROC) plots, as well as the standard deviation of ligand atoms from their experimentally determined location. Centroid calculations

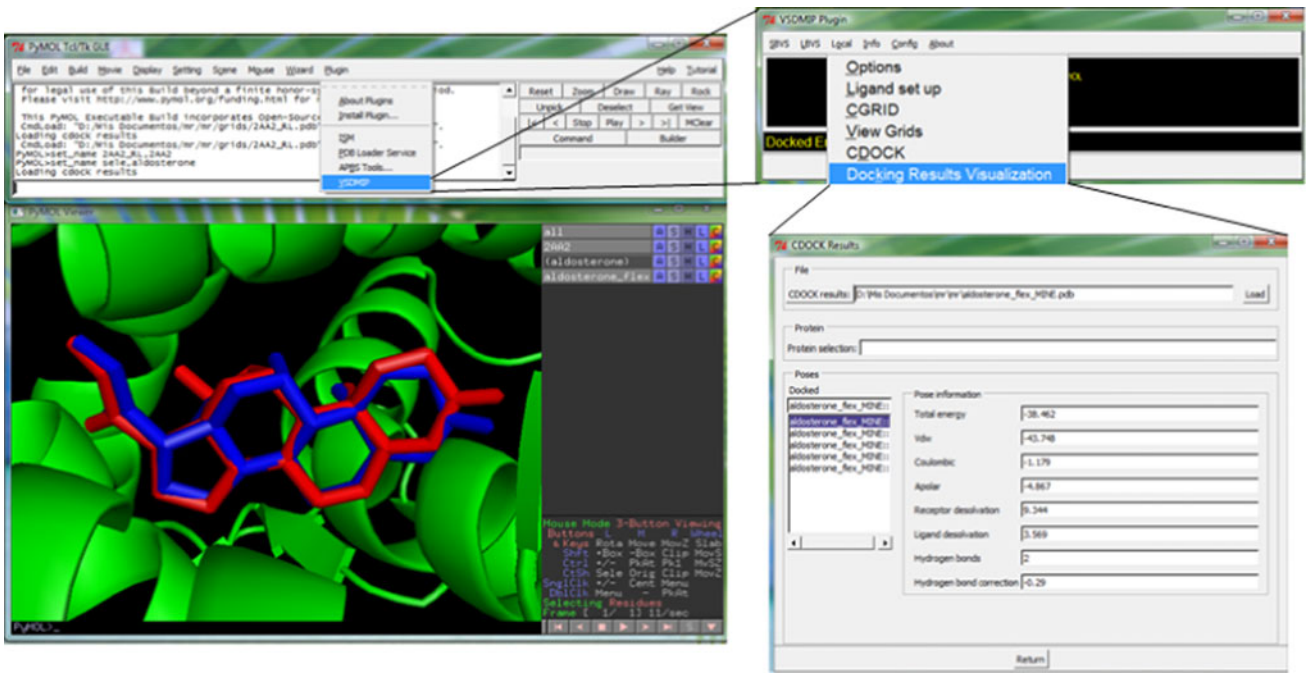


**Fig. 2** Screenshot showing the interface between PyMOL (*left*) and VSDMIP (*right*). A CDOCK window displays the docking results for aldosterone (*blue sticks*) in the ligand-binding site of the mineralocorticoid receptor (*green*) compared to the X-ray crystal structure (*red sticks*). The *lower-right* window lists the binding energy values associated with each pose

were also performed for each target. For fusion calculations, 3 randomly chosen actives were selected in each round, and 3 rounds were performed for each target. Mean values and standard deviations are also reported.

SBVS was performed by the combined use of CGRID and CDOCK [25] as explained in the original VSDMIP publication [15]. Briefly:

(a) for each protein structure, the initial binding site was defined as the space delimited by the axis-parallel box containing the co-crystallized ligand, augmented by 5 Å in each axis direction;
(b) CGRID calculation of protein interaction fields (a 12–6 Lennard-Jones term and an electrostatic term modeled with a sigmoidal dielectric screening function) covering the binding site (0.5 Å spacing in all directions) using common atom probes (C, N, O, S, P, H, F, Cl, Br, and I);
(c) exhaustive exploration by CDOCK of the location and orientation of each molecule within the binding site by positioning their centers of mass on grid points and performing discrete rotations of 27° on each axis;
(d) energy evaluation of each pose by the molecular mechanics force-field scoring function implemented in CDOCK that can additionally include ligand and receptor desolvation energy terms as well as counting of hydrogen bonding interactions; and
(e) selection of the best-scoring pose for each molecule as the docking solution.

Finally, LBVS was employed prior to SBVS to illustrate the connection between both modules and also to test its ability as a filter to reduce docking times using ACE and MR as the targets. To this end, MACCS fingerprints were calculated for ACE and MR active ligands, and the centroids method was used to combine all the information for the actives into single queries that were employed to search the entire DUD database of decoys and retrieve only those with a $Tc > 0.6$. The selected molecules were then docked into their respective targets as explained above.

The results from LBVS and SBVS were evaluated using ROC plots [32], which represent sensitivity ($y$-axis, true positives rate, Eq. 3) versus specificity ($x$-axis, false positive rate, Eq. 4). AUCs were calculated for each ROC plot.

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (3)$$

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}. \quad (4)$$

## Case study results: performance and timing

Using the eighteen targets selected from DUD we compared the performance of LBVS (through topology-based [MACCS] and mixed topological-physico-chemical-based [CATS] fingerprints) and SBVS (using docking and pharmacophore methods) as measured by the AUC values (see Table 1). For MACCS the centroids and fusion approaches were used to handle more than one initial query, whereas for CATS only the centroid approach was employed for testing purposes. In general, and considering the 18-target DUD reduced set, MACCS afforded the best results in terms of averaged AUC values ($0.74 \pm 0.17$), followed by CATS ($0.64 \pm 0.11$) and docking ($0.63 \pm 0.13$) when only one query was used to search the database (only applicable to MACCS and CATS). No significant changes were observed in MACCS ($0.72 \pm 0.14$) or in CATS ($0.65 \pm 0.10$) when different queries were considered via the centroids method. A slight improvement was found for MACCS and the fusion method ($0.79 \pm 0.19$), although only 6 cases were studied. Finally, the PPP method also performed reasonably well, but since it was applied to only two cases no definite conclusions can be drawn. A more detailed analysis can be done by splitting the AUC into three ranges: a lower-bound range where the AUCs are $\leq 0.5$ (worse than random), 0.5–0.7 (above random but with room for improvement), and $\geq 0.7$ (clearly better than random). On the one side, clearly better than random, MACCS afforded 60–70% of the targets with AUCs above 0.7, while these percentages were 30–40% for CATS and 20% for CDOCK. On the other side (worse than random) all the methods performed similarly (6%) although some variability was obtained for MACS depending on the number of starting queries (data not shown). In the middle range ($0.7 \leq \text{AUC} \leq 0.5$), CDOCK yielded the highest value ($\sim 70\%$), then CATS (40–60%), and finally MACCS (20–40%).

As will be discussed below, *analogue bias* can cause artificial enrichment because, if the query molecules are topologically similar to the actives, these will be retrieved more easily. To test this, we calculated similarity matrices using $Tc$ among the actives for each target, and some examples are shown in Fig. 3 (HMGR, TK, COX1, and ALR2).

The use of LBVS (in this case MACCS) as an initial filter to a more computational demanding task such as docking resulted in an important reduction in computer time while maintaining the overall performance. For ACE, the number of docking experiments that had to be done after filtering out with LBVS was reduced from 1791 to just 422, and this meant a reduction of 77% in computation time. The AUC was 0.66, slightly above the value for docking alone. Similar results were also found for the MR target. In this case, a reduction of 74% in computing time was achieved by reducing the number of ligands for docking from 637 down to 163, while the AUC was improved in 0.1 units.

**Table 1** AUC values for the two VS techniques (LBVS and SBVS) and the different methods according to each target studied

| Target | LBVS | | | | | SBVS | |
|---|---|---|---|---|---|---|---|
| | MACCS | | | CATS | | | |
| | Single FP | Centroids | Fusion (average) | Single FP | Centroids | CDOCK | PPP |
| ACE | 0.73 (0.09) | 0.78 | 0.79 (0.48) | 0.61 (0.13) | 0.56 | 0.63 | |
| MR | 0.72 (0.22) | 0.82 | 0.86 (0.03) | 0.51 (0.09) | 0.56 | 0.75 | 0.63 |
| HIVPR | 0.58 (0.13) | 0.62 | | 0.63 (0.12) | 0.46 | 0.25 | |
| P38 MAP | 0.72 (0.16) | 0.87 | 0.81 (0.10) | 0.53 (0.20) | 0.73 | 0.50 | |
| HMGR | 0.85 (0.22) | 0.63 | 0.51 (0.46) | 0.79 (0.09) | 0.77 | 0.26 | |
| PNP | 0.85 (0.06) | 0.71 | 0.91 (0.04) | 0.62 (0.09) | 0.63 | 0.60 | |
| COMT | 0.86 (0.13) | 0.96 | | 0.75 (0.21) | 0.86 | 0.33 | 0.79 |
| Thr | 0.67 (0.15) | 0.76 | | 0.83 (0.11) | 0.62 | 0.33 | |
| TK | 0.88 (0.05) | 0.70 | | 0.81 (0.11) | 0.61 | 0.60 | |
| fXa | 0.77 (0.18) | 0.61 | 0.88 (0.01) | 0.83 (0.17) | 0.60 | 0.55 | |
| AChE | 0.63 (0.10) | 0.59 | | 0.51 (0.07) | 0.55 | 0.73 | |
| HSP90 | 0.77 (0.62) | 0.39 | | 0.63 (0.06) | 0.54 | 0.65 | |
| COX1 | 0.48 (0.16) | 0.63 | | 0.50 (0.04) | 0.67 | 0.53 | |
| AMPC | 0.90 (0.11) | 0.70 | | 0.72 (0.08) | 0.66 | 0.47 | |
| ALR2 | 0.57 (0.08) | 0.62 | | 0.47 (0.11) | 0.61 | 0.33 | |
| COX2 | 0.85 (0.41) | 0.80 | | 0.60 (0.13) | 0.81 | 0.67 | |
| GPB | 0.81 (0.14) | 0.89 | | 0.72 (0.12) | 0.72 | 0.83 | |
| $ER_{agonists}$ | 0.75 (0.12) | 0.92 | | 0.48 (0.05) | 0.70 | 0.63 | |

Numbers in parenthesis are the standard deviations

*ACE* angiotensin-converting enzyme, *MR* mineralocorticoid receptor, *HIVPR* HIV protease, *P38 MAP* P38 mitogen activated protein, *HMGR* hydroxymethylglutaryl-CoA reductase, *PNP* purine nucleoside phosphorylase, *COMT* catechol *O*-methyltransferase, *Thr* thrombin, *TK* thymidine kinase, *fXa* coagulation factor Xa, *AChE* acetylcholinesterase, *HSP90* human heat shock protein 90, *COX-1* cyclooxygenase-1, *AMPC* AmpC $\beta$-lactamase, *ALR2* aldose reductase, *COX-2* cyclooxygenase-2, *GPB* glycogen phosphorylase $\gamma$, $ER_{agonists}$ estrogen receptor (agonist-bound conformation)

To assess the applicability of VSDMIP to large-scale VS projects, based on ligands and/or receptors, we measured the overall performance of VSDMIP when undertaking the major tasks that are common to all the protocols such as inserting the molecules into the database, generating fingerprints, searching within the database, and docking (Table 2).

According to the data compiled in Table 2, an average of 370 and 275 molecules can be inserted and docked, respectively, per day and CPU using VSDMIP. The number of inserted molecules showed a very high correlation with the number of conformers per molecule ($r^2 = 0.98$, after exclusion of COMT due to the fact that the ligands for this target are few and very small, and therefore uncharacteristic of the most typical real world scenario). The correlation was more modest ($r^2 = 0.79$) when the number of heavy atoms, the number of conformations per molecule, and the number of valid grid points were simultaneously considered. As expected, LBVS proved to be several orders of magnitude faster than SBVS. In fact, VSDMIP is able to generate around $10^7$ 2D- (molecules) and $10^6$ 3D-fingerprints (conformers) per day and CPU. On the other hand, $10^9$ and $10^6$ comparisons can be performed using 2D- and 3D-fingerprints, respectively, per day and CPU.

## Technical issues

### The PyMOL plugin

The VSDMIP plugin implements a visual interface to manage the most common tasks in VS. The menu bar has three main categories of actions: SBVS, LBVS and Local.

SBVS holds a visual interface for the original workflow implemented in the first release of VSDMIP. It has been extended to use results originating from its LBVS counterpart and to perform docking on the receptor with the selected molecules.

LBVS encompasses database-related operations (such as filters and search tools) to perform similarity calculations on 2D/3D fingerprints (codifications of several molecular features) and does not require the 3D structure of a receptor.

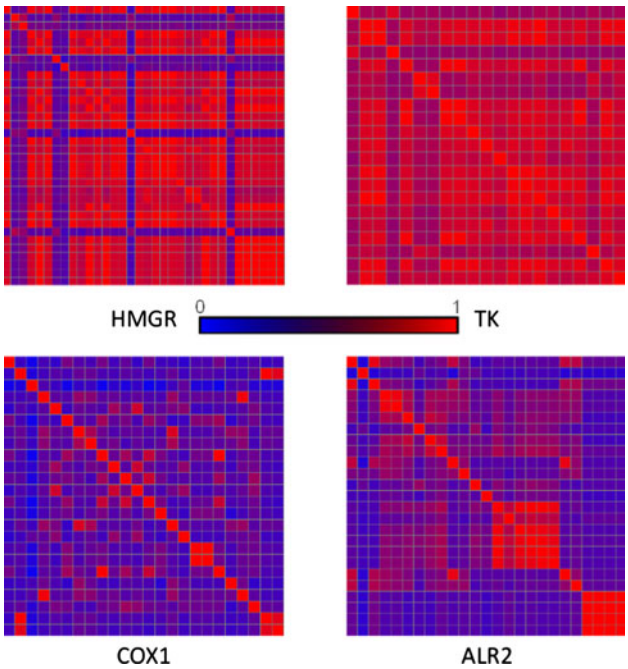


**Fig. 3** Similarity matrices using Tanimoto coefficients for true binders to four different targets. Note that the *analogue bias* effect is clearly shown on HMGR and TK, while it is completely absent in COX1 and ALR2

Local allows the user to execute a complete docking job on a local machine without extending the capabilities of the basic programs (database-related functionalities). The SBVS and LBVS modules are designed to work within a cluster of processors or on a multiprocessor machine. The main node on the cluster communicates with the other calculation nodes through the secure shell (SSH) protocol in a way that is completely transparent to the user.

LBVS techniques have been implemented that extend currently available tools (Open Babel [33], MySQL, and PyMOL) and new programs and interfaces have been designed, e.g. the GTP code to enable the recognition of all possible PPP (see above at the "3D fingerprints" section).

The MySQL Application Programming Interface (API) was used to build User Defined Functions (UDF) in order to hold the methods for comparing and scoring fingerprints, including scoring fusion. The UDF are loaded directly into the main memory whenever the database requires them so that the filtering and searching processes are speeded up.

## Extensions to VSDB

Three new tables have been added to the Virtual Screening DataBase (VSDB) contained in the original VSDMIP: FINGERPRINT, PHARMACOPHORE, and FINGER_TYPE (see Fig. 4). The first and second tables contain 2D and 3D fingerprints, respectively, whereas the third one stores the information and description of the available fingerprint types. The original role played by VSDB in the previous version of VSDMIP (storing molecules and results) has been expanded with filtering and searching tools within the MySQL engine. To this end the MySQL language was complemented with new functions that allow advanced molecular screenings based on similarity calculations to be performed. Backward compatibility with the original VSDMIP is maintained, as the plugin developed here is an optional upgrade.

## Operating system and software/hardware requirements

The client version of the platform is compatible with the Linux and Windows operating systems. However, the

**Table 2** Some ligand- and binding site-related properties of the complexes studied here and overall VSDMIP performance in the main operations

| | Target[a] | ⟨NHA⟩[b] | ⟨Conf⟩[c] | BSGP[d] | Insertion | Docking | | | 2D | 3D |
|---|---|---|---|---|---|---|---|---|---|---|
| SBVS | ACE | 23 | 118 | 18225 | 192 | 336 | LBVS | FP Generation[e] | $9 \times 10^6$ | $1 \times 10^6$ |
| | COMT | 16 | 35 | 7956 | 6600 | 288 | | | | |
| | PDE5 | 30 | 136 | 7182 | 192 | 168 | | | | |
| | AChE | 26 | 133 | 7040 | 200 | 342 | | Search | $2.4 \times 10^9$ | $0.5 \times 10^6$ |
| | PARP | 20 | 16 | 4332 | 1080 | 288 | | | | |
| | Thr | 32 | 136 | 2688 | 192 | 221 | | | | |

For structure-based VS (SBVS), the Insertion and Docking columns display the number of molecules that are processed per day and CPU (either a PIV 32-bit 3.2 GHz or a Xeon 32-bit 3.06 GHz processor). For ligand-based VS (LBVS), the data shown are also molecules per day and the same type of CPU

[a] ACE, COMT, AChE, and Thr have already been defined in Table 1. *PDE5* phosphodiesterase 5, *PARP* poly(ADP-ribose) polymerase

[b] Average Number of Heavy Atoms in the ligand set for each target

[c] Average number of conformations in the ligand set for each target

[d] Binding Site Grid Points

[e] Fingerprints generation

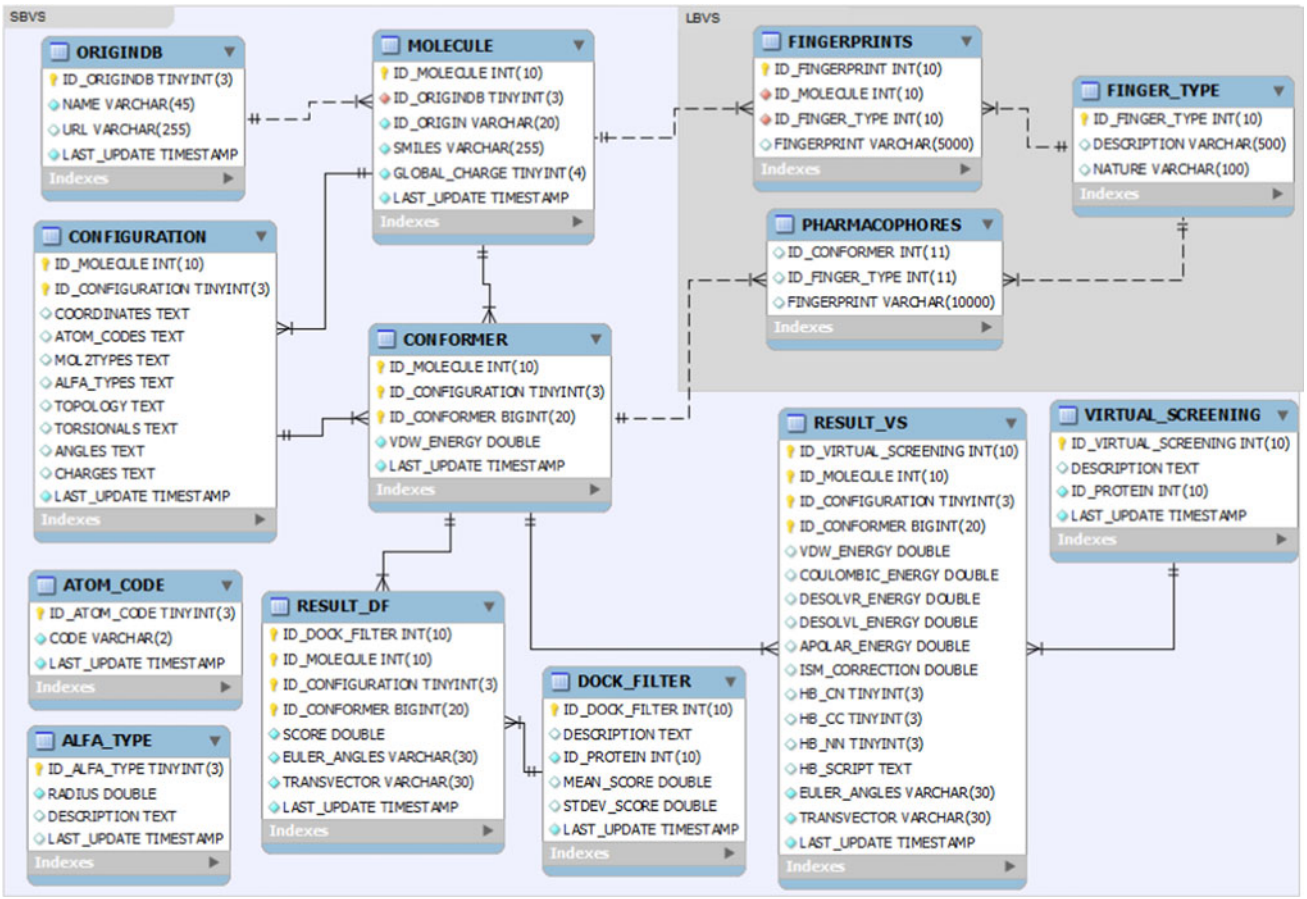


**Fig. 4** Entity-relationship scheme showing the database (VSDB) used by VSDMIP. The *dark grey* shaded region corresponds to the newly added capabilities

server modules are only available for Linux with the OpenPBS/Torque queue system 2.x or above. In addition, the VS modules require an MySQL database engine and the client libraries installed on the system (mysql-server, mysql++ [34], and libmysqlclient).

The GUI front end uses PyMOL software version 1.2 or higher and requires the mysql-python (version $\geq$ 2.4) and NumPy (version $\geq$ 1.3) modules. It can be executed on Windows and Linux operating systems. The back end can be used in a wide variety of hardware architectures, from single personal computers or laptops to a cluster of processors or grid-like systems. The minimum recommended amount of main memory per processor is 1 GB. The database needs $\sim$45 GB of hard disk space to store $4 \times 10^6$ molecules together with their properties and calculated conformers. In addition, $\sim$850 and $\sim$285 MB more are needed to store information related to 2D or 3D fingerprint types, respectively. Of note, all this information is entered only once and can be reutilized in every project. Finally, $\sim$750 MB of extra storage space would be needed for the outcome of filtering/screening the entire database.

## VSDMIP in the context of existing VS platforms

When VSDMIP was originally released, a limited number of similar platforms were available, namely Pipeline Pilot [7, 8] from Accelrys, alternative implementations from Schrödinger [5] and Tripos [6], and a proprietary web-based platform from Astex Therapeutics [35]. Other non-commercial solutions include the Data Management System for Distributed Virtual Screening (DVSDMS) [14], the SOMA workflow multiplatform [36], the KNIME modular environment [37], and the public access web-based DOCK Blaster platform [12]. On the other hand, the MoStBioDat database [38] was designed to store and manipulate ligand and receptor data and allows LBVS to be performed. A much wider view of the drug design cycle in terms of the implemented features is represented by OSIRIS [39] and PFAKT [40]. The former is defined as a *drug discovery informatics system* […] *and contains a complete coverage of the drug discovery process by custom tailored applications* whereas the latter *is a suite of integrated services* […] *that facilitate the medicinal chemistry design cycle* […] *and provides a unified data analysis and collaboration environment*.

Some other applications have been launched with a different goal in mind, namely the need to lower the technical skill barriers so that a wider range of researchers can benefit from them. Examples are BDT [41], DOVIS [42, 43], VSDocker [44], AMMOS [45], iGEMDOCK [46], VSM-G [13], and a PyMOL plugin for AutoDock/Vina [10]. VSDMIP 1.5 represents an attempt to provide the scientific community with a customizable and comprehensive VS platform that is managed from a friendly GUI and also integrates an underlying database.

We believe, therefore, that the capabilities added to VSDMIP (namely the GUI, the LBVS module, and the interconnectivity between LBVS and SBVS modules) place version 1.5 within the *state-of-the-art* automatic platforms that perform VS experiments. Having all the tools integrated in a single application facilitates the complex task of building VS protocols and analyzing the results (Fig. 5). Besides, VSDMIP 1.5 allows the user to work with the programs individually (single docking and grid visualization, database searches, analysis of interactions, conformational analysis…) taking full advantage of the easy-to-use PyMOL interface. From the technical point of view, and although some computer skills are still required to properly configure the application for high performing computer architectures, the GUI and the configuration files provided as a guide (plus the support given by the development team) makes VSDMIP installation and maintenance relatively straightforward. VSDMIP can also be run on a desktop computer with or without the database environment, employing all the available cores in a small cluster, a user-defined number of them, or even just one. VSDMIP has been fully tested on a Linux cluster using a Linux- or Windows-running computer as the interface to the cluster. Modularity (individual tasks can be connected in different ways to allow the user to customize his/her VS protocols) and flexibility (other software pieces can be easily added to the platform and configured through user-configurable extensible markup language [XML] files) are still retained in this new version as they are considered the main cornerstones that differentiate VSDMIP from other VS platforms described to date. Detailed information related to installation, configuration, possible extensions, as well as examples, can be found within the User's Guide at http://ub.cbm.uam.es/software/vsdmip/doc/.

## VSDMIP performance

The numerical results shown here for a reduced subset of targets from the DUD database follow the trend commented above [47], i.e., in general, LBVS (fingerprints) outperforms SBVS (pharmacophores and docking). On the other hand, our in-house CGRID/CDOCK docking tool ($\langle AUC \rangle \approx 0.6$) performs as well as DOCK, FlexX, ICM, and PhDOCK, and slightly worse than GLIDE and Surflex ($\langle AUC \rangle \approx 0.7$), or eHiTS ($\langle AUC \rangle \approx 0.9$) [48]. Thus, there is clearly room for improvement. Although we found these results reasonable, we are aware of the possible *analogue bias* that might be introduced during the construction of the DUD database, which leads to artificial enrichments (see Table 1) in the case of LBVS methods [49]. A key ingredient to achieve success in retrospective VS experiments is to count with a well defined database (actives + decoys) of complexes with known 3D structures and information about their activity. The term "well defined" refers to the fact that a good VS method should differentiate actives from decoys on the basis of interaction features only. The molecular properties of the selected decoys should resemble those of the active ligands and at the same time, to avoid artificial enrichments, they should

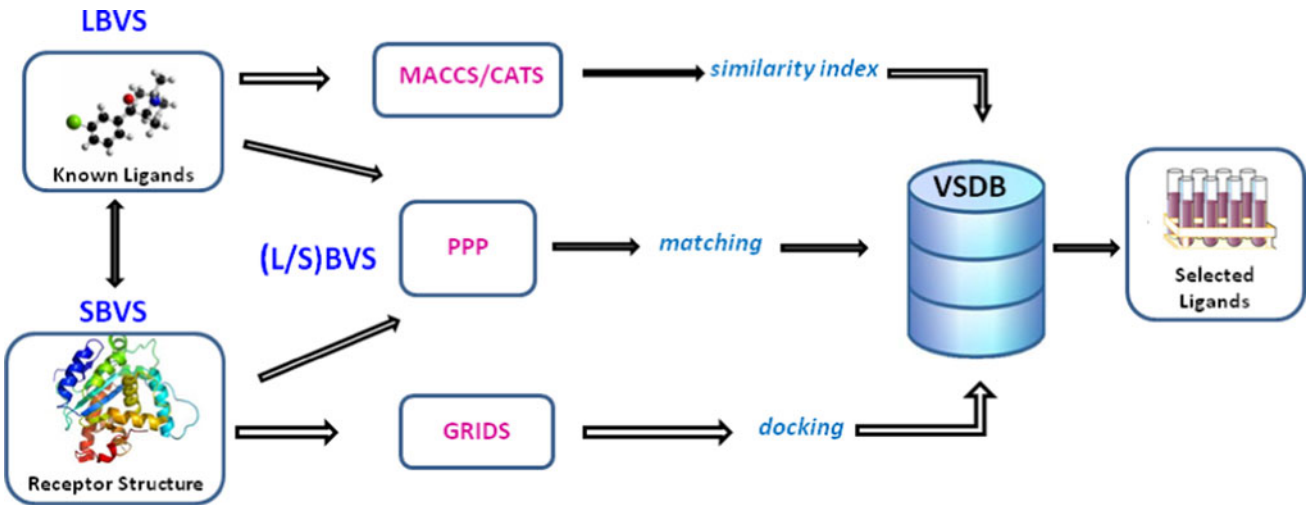


**Fig. 5** VSDMIP flowchart

be structurally different. To comply with this requirement the DUD database was created as a specific benchmark to test docking methods, and has been widely accepted within the docking and VS communities to the extent that it is generally considered as the reference database. However, some caveats have been pointed out [50], in particular overfitting effects, its incomplete sampling of chemical space, and its inadequacy to be applied to LBVS methods. In relation to its applicability in LBVS, as reported here, recent results have shown that the DUD decoy sets are robust enough for LBVS [51]. Furthermore, it was found that 2D fingerprints outperformed 3D shape-based approaches as VS tools, in agreement with a previous study where fingerprints, pharmacophores, and docking were compared as VS engines [47]. This behavior might be linked to the *analogue bias* effect, so called because chemical diversity is not warranted in the DUD database and many decoys share a reduced set of common scaffolds resembling the actives. If by any chance one of the most populated scaffolds is selected as the query, artificial enrichment comes into play. *Analogue bias* is more prone to occur with the use of fingerprints than with 3D methods based on shape, pharmacophores or docking simply because the former represent molecular similarity whereas the latter involve chemical interactions. This observation was confirmed by us in some of the targets. We calculated similarity matrices among all the actives for each target using the *Tc* (see Eq. 1), coloring them from blue (low similarity, $Tc = 0$) to red (high similarity, $Tc = 1$). Selected results are shown in Fig. 3. For HMGR and TK the active sets are very similar (predominant red color in both matrices) with MACCS AUCs $\sim$ 0.8–0.9, CATS AUCs $\sim$ 0.8, and CDOCK AUCs $\sim$ 0.6. On the other hand, more difficult cases are represented by COX1 and ALR2, where the active sets are highly dissimilar, as shown by the prevailing blue color. In these cases, MACCS and CATS AUCs drop to 0.5–0.6, while CDOCK AUCs are less sensitive and only small variations are observed.

## Benchmarking

With a medium-size cluster ($\approx$100 processors) and a database of the order of several million compounds ($4 \times 10^6$ in VSDMIP), it would take 3.6 months to insert all the molecules. Although this can be considered a very long time, most of it is spent on calculating the point charges, and we recall here that the molecules need to be inserted only once and then can be reutilized as many times as desired. Docking the complete database would take around 5 months. Nevertheless, and given the high speed obtained in LBVS, several of these runs can be easily completed in an affordable time span before undertaking more computationally intensive SBVS. LBVS would then serve as a filter, leading to an important reduction in the number of molecules to be docked, as shown above for the ACE and MR targets, and as a result, to an important optimization of the computer resources. By the time the results presented here were collected, we acquired some new Xeon Core2 64-bit 2.5 GHz processors, and preliminary tests yielded a speed-up of 2.2-fold relative to the old ones. Therefore, it should now be possible to insert $4 \times 10^6$ SMILES strings in less than 2 months, and dock the entire database in 2.2 months using 100 processors of this type. In view of these figures, and taking into account the ever-increasing computer power and the fact that docking (in the way it has been implemented in VSDMIP) is 100% scalable (the more processors available, the less time required to complete the tasks), we believe that VSDMIP is a computational platform capable of performing VS experiments in perfectly affordable time schedules in an environment accessible to a large number of researchers.

## Conclusions

VSDMIP 1.5 allows an inexperienced user to execute both SBVS and LBVS protocols, or any combination of the two, by means of an easy-to-learn and friendly GUI implemented in the commonly used PyMOL molecular graphics program. We have tested its ability to conduct VS protocols and compared the efficiency of different methods. Good agreement with results from previous studies was found but we also realized that the *analogue bias* effect in the DUD database can lead to artificial enrichment for LBVS. In terms of computer time, we show that VSDMIP can indeed cope with the current demand of performing VS experiments in weeks rather than in months. This version of the platform is distributed to the scientific community upon request from the authors as a bundled package including the scripts and necessary SQL files to create the database structure and the XML configuration files. The programs implemented in the platform (except those that need to be purchased for a modest prize, such as CORINA or AMBER) are either free for academics (MOPAC, DOCK, FRED, AutoDock) or will be released under a scientific/academic non-profit and non-commercial license as is the case for ALFA, CGRID, CDOCK, and ISM.

**Acknowledgments** The authors thank Dr. Eva M[a] Priego and Dr. Alberto Gómez for testing the application and valuable comments, as well as the rest of members of the Bioinformatics Unit at CBMSO and the Molecular Modeling group at UAH for encouragement and fruitful discussions. This work was supported by grants from Ministerio de Ciencia e Innovación (MICINN) BIO2008-04384 (to A. M.) and SAF2009-13914-C02-02 (to F. G.), and Comunidad Autónoma de

Madrid (CAM) S-BIO-0214-2006. A. M. acknowledges CAM for financial support through the AMAROUTO program to the Fundación Severo Ochoa, R. G. -R. thanks MICINN for a contract from "Programa de Personal Técnico y de Apoyo 2008", and A. C. thanks Ministerio de Educación for the FPU grant AP2009-0203. We are grateful to OpenEye Scientific Software, Inc. for providing us with an academic license for their software. The technical support and advice from the Bioinformatics Facility at CBMSO is gratefully acknowledged, as well as the computer resources, technical expertise and assistance provided by the Barcelona Supercomputing Center—Centro Nacional de Supercomputación.

## References

1. Munos B (2009) Nat Rev Drug Discov 8(12):959
2. Jorgensen WL (2004) Science 303(5665):1813
3. Zhou HX, Gilson MK (2009) Chem Rev 109(9):4092
4. Ivanov AS, Veselovsky AV, Dubanov AV, Skvortsov VS (2006) Methods Mol Biol 316:389
5. Maestro (2011) Maestro, version 9.2. Schrödinger, LLC, New York
6. SYBYL-X 1.2. (2011) Tripos International, 1699 South Hanley Rd., St. Louis, Missouri, 63144, USA
7. Schulz T, Pleiss J, Schmid R (2000) Protein Sci 9(6):1053
8. Hassan M, Brown RD, Varma-O'brien S, Rogers D (2006) Mol Divers 10(3):283
9. DeLano WL (2002) The PyMOL molecular graphics system. Schrodinger Inc, New York
10. Seeliger D, de Groot BL (2010) J Comput Aided Mol Des 24(5):417
11. Lill MA, Danielson ML (2011) J Comput Aided Mol Des 25(1):13
12. Irwin JJ, Shoichet BK, Mysinger MM, Huang N, Colizzi F, Wassam P, Cao Y (2009) J Med Chem 52(18):5712
13. Beautrait A, Leroux V, Chavent M, Ghemtio L, Devignes MD, Smail-Tabbone M, Cai W, Shao X, Moreau G, Bladon P, Yao J, Maigret B (2008) J Mol Model 14(2):135
14. Zhou T, Caflisch A (2009) J Chem Inf Model 49(1):145
15. Gil-Redondo R, Estrada J, Morreale A, Herranz F, Sancho J, Ortiz AR (2009) J Comput Aided Mol Des 23(3):171
16. Huang N, Shoichet BK, Irwin JJ (2006) J Med Chem 49(23):6789
17. Murray CW, Baxter CA, Frenkel AD (1999) J Comput Aided Mol Des 13(6):547
18. Schneider G, Neidhart W, Giller T, Schmid G (1999) Angew Chem Int Ed 38(19):2894
19. Open Babel: The open source chemistry toolbox; 2011
20. Weininger D (1988) J Chem Inf Comput Sci 28(1):31
21. Case DA, Cheatham TE III, Darden T, Gohlke H, Luo R, Merz KM Jr, Onufriev A, Simmerling C, Wang B, Woods RJ (2005) J Comput Chem 26(16):1668
22. Jorgensen WL, Chandrasekhar J, Madura JD, Impey RW, Klein ML (1983) J Chem Phys 79:926
23. Zacharias M, Luty BA, Davis ME, McCammon JA (1994) J Mol Biol 238(3):455
24. Pastor M, Cruciani G (1995) J Med Chem 38(23):4637
25. Perez C, Ortiz AR (2001) J Med Chem 44(23):3768
26. Weininger D (1988) J Chem Inf Model 28(1):31
27. Corina. Computerchemie Langemarckplatz 1, Erlangen, Germany: Molecular Networks GmbH; 2000
28. Stewart JJ (1990) J Comput Aided Mol Des 4(1):1
29. Gil Redondo R (2006). Master thesis. UNED, Madrid
30. Dolinsky TJ, Nielsen JE, McCammon JA, Baker NA (2004) Nucleic Acids Res 32(suppl 2):W665
31. Li H, Robertson AD, Jensen JH (2005) Protein Struct Funct Bioinforma 61(4):704
32. Triballeau N, Acher F, Brabet I, Pin JP, Bertrand HO (2005) J Med Chem 48(7):2534
33. Guha R, Howard MT, Hutchison GR, Murray-Rust P, Rzepa H, Steinbeck C, Wegner J, Egon L, Willighagen O (2006) J chem inf model 46(3):991
34. MySQL++. A MySQL API for C++: Tangensoft
35. Watson P, Verdonk M, Hartshorn MJ (2003) J Mol Graph Model 22(1):71
36. Lehtovuori PT, Nyronen TH (2006) J Chem Inf Model 46(2):620
37. Preisach C, Burkhardt H, Schmidt-Thieme L, Decker R, Berthold MR, Cebron N, Dill F, Gabriel TR, Kötter T, Meinl T, Ohl P, Sieb C, Thiel K, Wiswedel B (2008) KNIME The Konstanz information miner. In: Bock HH, Gaul W, Schader M et al (eds) Data analysis. machine learning and applications. Springer, Berlin Heidelberg, p 319
38. Bak A, Polanski J, Kurczyk A (2009) Molecules 14(9):3436
39. Sander T, Freyss J, von Korff M, Reich JR, Rufener C (2009) J Chem Inf Model 49(2):232–246
40. Brodney MD, Brosius AD, Gregory T, Heck SD, Klug-McLeod JL, Poss CS (2009) J Chem Inf Model 49(12):2639
41. Vaque M, Arola A, Aliagas C, Pujadas G (2006) Bioinformatics 22(14):1803
42. Zhang S, Kumar K, Jiang X, Wallqvist A, Reifman J (2008) BMC Bioinformatics 9:126
43. Jiang X, Kumar K, Hu X, Wallqvist A, Reifman J (2008) Chem Cent J 2:18
44. Prakhov ND, Chernorudskiy AL, Gainullin MR (2010) Bioinformatics 26(10):1374
45. Pencheva T, Lagorce D, Pajeva I, Villoutreix BO, Miteva MA (2008) BMC Bioinformatics 9:438
46. Hsu KC, Chen YF, Lin SR, Yang JM (2011) BMC Bioinformatics 12(Suppl 1):S33
47. Modest von Korff JF, Sander Thomas (2009) J Chem Inf Model 49(2):209
48. Cross JB, Thompson DC, Rai BK, Baber JC, Fan KY, Hu Y, Humblet C (2009) J Chem Inf Model 49(6):1455
49. Verdonk ML, Berdini V, Hartshorn MJ, Mooij WT, Murray CW, Taylor RD, Watson P (2004) J Chem Inf Comput Sci 44(3):793
50. Irwin JJ (2008) J Comput Aided Mol Des 22(3–4):193
51. Venkatraman V, Perez-Nueno VI, Mavridis L, Ritchie DW (2010) J Chem Inf Model 50(12):2079