The FlexX Database Docking Environment — Rational Extraction of Receptor Based Pharmacophores

Holger Claussen[†,*], Marcus Gastreich[†], Volker Apelt[†], Jonathan Greene[‡], Sally A. Hindle[†], and Christian Lemmen[†]

[†] *BioSolveIT GmbH, An der Ziegelei 75, 53757 St. Augustin, Germany*

[‡] *Cambios Computing, LLC, 1481 Pitman Avenue, Palo Alto, CA 94301, U.S.A.*

Abstract: We present an integrated docking environment that allows for iterative and interactive detailed analysis of many docking solutions. All docking information is stored in an ORACLE database. New scoring schemes (e.g. target-specific scoring functions) as well as various types of filters can be easily defined and tested within this environment.

As an example application we investigated the validity of the following hypothesis: If a docking procedure can lead to enrichments significantly better than random then a bias towards (partially) correct placements should be detectable. Such bias in terms of a preference for certain interacting groups within the active site can be used to select a set of receptor-based pharmacophore constraints, which in turn might be used to enhance the docking procedure.

As a proof of concept for this approach we performed docking studies on three targets: thrombin, the cyclin-dependent kinase 2 (CDK2) and the angiotensin converting enzyme (ACE). We docked a set of known active compounds with standard FlexX and derived three sets of target-specific receptor-based pharmacophore constraints by statistical analysis of the predicted placements. Applying these receptor-based constraints in a virtual screening protocol utilizing FlexX-Pharm led to significantly improved enrichments.

Keywords: virtual screening, docking, target-tailored scoring, pharmacophore, FlexX-Pharm, Oracle, ACE, CDK2, thrombin.

1. INTRODUCTION

Virtual screening of large libraries of compounds has become a routine task in the drug discovery process [1–4]. The steadily increasing compute power makes it not only possible to apply fast filter criteria, but also to screen many more compounds using molecular docking methods in a reasonable time [5, 6]. Current docking tools such as AutoDock [7], DOCK [8, 9], FlexX [10, 11], FRED [12], Glide [13], and GOLD [14, 15] are able to predict ligand placements with sufficient accuracy and speed. However, the scoring, and therefore rank-ordering, of the predicted placements is still an open problem for docking [5, 6, 16, 17]. Several different scoring functions have been suggested (see [18] for a recent review), which demonstrate advantages for different types of target proteins and, more generally, for different modes of application. Progress has been made by applying filters to rule out obviously wrong placements [19]. Also, techniques that combine different scoring functions simultaneously in a consensus scoring scheme [20–23] or combine terms of different scoring functions [24, 25] were shown to improve performance. All of these approaches try to find a universal scoring method that works for all kinds of protein-ligand complexes. Alternatively, it is also possible to tailor the scoring function to a specific target. Target-specific scoring functions are quite attractive, especially as the available data for particular targets and target families grows within pharmaceutical companies as well as through academic research. Beyond adapting the scoring function, one can incorporate additional knowledge about the target active site [26–29]. Traditionally, pharmacophores are derived from and are defined on the ligand side. Now though, increasing understanding about the nature of active sites allows the derivation of valuable information about interaction counter-groups in the target. There are several approaches for analyzing an active site in order to identify favorable interactions or possible pharmacophore counter-groups (i.e. those groups of atoms in the target, which form interactions with the pharmacophoric groups on the ligand) [16, 30–34]. The most important interactions can be selected as so-called receptor-based pharmacophore constraints and applied as a post-processing filter to eliminate undesirable ligand placements and refine the results of a docking calculation. More effectively, these constraints can be applied during the docking process to reject misplacements of the ligand as early as possible, enabling a speed-up in the calculation and the generation of novel placements [26, 27]. One rational approach to gain sufficient insight necessary for creating a target-specific scoring function or set of receptor-based pharmacophore constraints, is to carry out an iterative and interactive detailed analysis of docking solutions. Since output from docking tools is normally in flat file format, such analysis is tedious if not impossible. A large amount of parsing is required, the process is error prone, and

*Address correspondence to this author at BioSolveIT GmbH, An der Ziegelei 75, 53757 St. Augustin, Germany; Tel: ++49-2241-9736660; Fax: ++49-2241-9736688; E-mail: Holger.Claussen@biosolveit.de

1570-1638/04 $45.00+.00

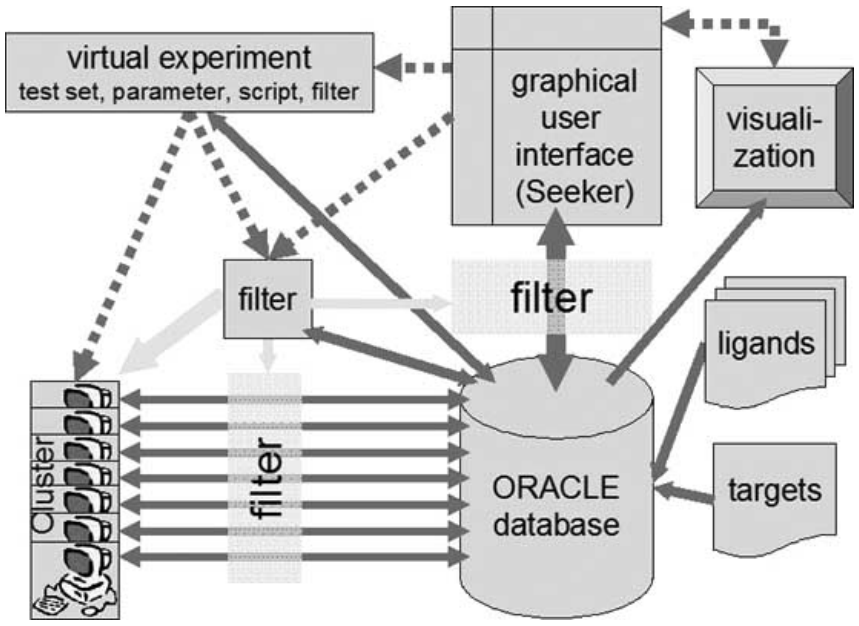

Fig. (1). Overview of Docking Environment. The ORACLE database is the central building block of the docking environment. It contains all input data (ligands, targets), the configuration of the virtual experiments and all docking information. The computing nodes in the cluster access the database directly to fetch their input and to store their results. The Seeker fetches the results *via* Structured Query Language (SQL) queries from the database. The 2D or 3D visualization is launched from the GUI. The definition of virtual experiments and filters can be done interactively from the Seeker. The filters affect the database queries during the interactive analysis. For screening runs they can be applied before storing the data in the database or even during docking (e.g. receptor-based pharmacophore constraints).

consistency is hard to maintain. Therefore, a fully integrated virtual screening platform [35, 36] would certainly be of great benefit. We developed an integrated docking environment, which facilitates the detailed analysis of many predicted placements and which supports rapid prototyping of novel scoring schemes (see Fig. (**1**)). A key step is to store all docking information in a database. The data can be viewed by interactively controllable spread sheets from which 2D and 3D viewers can easily be launched. New combinations of scoring terms as well as various types of filters can be defined and directly employed in this environment. The filters can be made persistent and applied as an integral part of large scale virtual screening runs.

The main components of our docking environment (Fig. (**1**)) comprise an ORACLE database [37] for storing all docking data, a modified version of FlexX [10, 11] that is able to access the database directly, and a graphical user interface (GUI) known as Seeker (for a schematic of the GUI, see Fig. (**2**)). The Seeker not only facilitates interactive analysis of docking results, it also represents the front-end to the database and allows easy access and handling of all database objects that are relevant in this environment. Furthermore, it supports the assembly and control of docking jobs, which may be performed in parallel on a compute cluster. The back-end compute engine constitutes an extension to FlexX, which interfaces to ORACLE and the Seeker, written in Python [38]. We also developed a general purpose Python interface for FlexX, called PyFlexX that provides all FlexX menu commands as functions in a number of Python modules.

Several applications for this integrated docking environment can be envisioned. For example, within the environment the user may readily:

- investigate the specifics of a novel target by analyzing dockings of a set of known active compounds and inactive compounds (training set)
- tailor a scoring function to a specific target (family)
- define and test thresholds for scoring (i.e. define filters)
- derive receptor-based pharmacophore constraints
- specifically analyze combinatorial libraries, aimed at selecting potent sublibraries
- evaluate a supplier database, aimed at selecting auspicious compounds for purchase.

In this paper, as an example application, we focus on the aspect of deriving receptor-based pharmacophore constraints, which may be used as a target-specific filter in virtual screening. The constraints are extracted by a statistical analysis of docking calculations. To this end the following workflow is applied:

						Ranks in subsets			
Target	**Position**	**Residue**	**Number**	**Atom**	**Avg. count**	**Highest**	**Lowest**	**Average**	**Pharm**
a) *Thrombin* (1dwd)	1.	Gly	219	O	8881.4	1	1	1.00	opt.
	2.	Asp	189	OD2	8548.9	2	2	2.00	opt.
	3.	Asp	189	OD1	8041.1	3	3	3.00	opt.
	4.	Gly	216	(N)H	5493.0	4	4	4.00	opt.
	5.	Ser	195	OG	3407.0	5	5	5.00	
	6.	Gly	216	O	3338.0	5	5	5.00	
b) *CDK2* (1di8)	1.	Leu	83	(N)H	3994.7	1	1	1.00	ess.
	2.	Leu	83	O	3039.1	2	2	2.00	opt.
	3.	Glu	81	O	1590.9	3	4	3.10	opt.
	4.	Gln	131	O	1338.4	3	4	3.90	
	5.	Ile	10	O	714.7	5	5	5.00	
	6.	His	84	O	654.8	5	5	5.00	
c) *ACE* (1o86)	1.	Zn	701	Zn	7479.4	1	1	1.00	ess.
	2.	Tyr	523	(O)H	4310.2	2	2	2.00	opt.
	3.	Gln	281	NE2	3381.6	3	3	3.00	opt.
	4.	Lys	511	NZ	2953.6	4	4	4.00	opt.
	5.	His	353	NE2	2826.0	5	5	5.00	
	6.	Tyr	520	(O)H	2754.0	5	5	5.00	

Table I: Frequency-sorted list of target amino acid atoms and hetero-atoms (metal ion in ACE) participating in interactions to active compounds (including FlexX interaction types "metal", "H-donor" and "H-acceptor" only) for (a) thrombin, (b) CDK2, and (c) ACE. Atoms are in PDB notation; a parenthesized (O) is meant to clarify that the oxygen is the phenolic O, a parenthesized (N) signifies that the N-bound hydrogen is involved in the respective H-bonds. In both cases, the hydrogen, not the O or N, respectively, is involved in the interaction. The table shows the six top ranking interactions, which are specified by the *residue* name, the residue *number* and the interacting *atom*. The average number of *counts*, the *highest* and the *lowest* rank and the *average* rank within all ten subsets are listed (see Section 3.3). *Pharm* signifies which category of FlexX-Pharm constraints we have chosen for the respective interaction.

1. A set of known active compounds is docked with FlexX. Information about all generated placements for all active compounds (which includes information about the intermolecular interactions) is stored in the database.

2. A particular query to the database produces statistics regarding individual intermolecular interactions.

3. Based on the statistics, receptor-based pharmacophore constraints are extracted from inspection of the most commonly encountered interactions.

4. A library of compounds is docked with FlexX-Pharm [26, 27] including the extracted constraints.

The motivation for this computational experiment is the following hypothesis: If docking methods do not produce any reasonable placements, it is unlikely that any significant prioritization of active *vs.* inactive compounds can be achieved. However, significant enrichments over random have been reported time and again [4, 24, 39]. Furthermore, reproduction of crystal structures for the majority of cases on benchmark data are commonplace [10, 15, 25, 40]. Therefore, we may assume a certain bias towards reasonable placements. Simple statistics should be able to reveal such bias in docking data.

As an initial test case we docked thrombin inhibitors into the Protein Database (PDB) [41] structure 1dwd and simply counted the hydrogen interactions formed by all the placements generated with FlexX with each particular target atom within the active site (Tab. (**I**)). The residues that form the S_1 specificity pocket (Gly 219, Asp 189, Gly 216) scored highest followed by Ser 195 from the catalytic triad, which is the only amino acid of the three that is in direct contact with the ligand [42–44]. Supported by such initial evidence, we applied the full workflow outlined above to three targets: thrombin, the cyclin-dependent kinase 2 (CDK2) and the angiotensin converting enzyme (ACE). As a proof of concept we compared our findings with the current literature consensus and calculated the enrichments of virtual screening runs *with* and *without* the receptor-based pharmacophore constraints in FlexX-Pharm.

Thrombin is critical in controlling the blood clotting process. The development of thrombin inhibitors is therefore, of great interest for improving treatments for thromboses and following cardiac infarction or surgery.

It has long been known that CDK2 plays an important role in regulating the cell cycle [45]. Due to the relationship between CDK2-activity and the rate of cell division [46, 47],

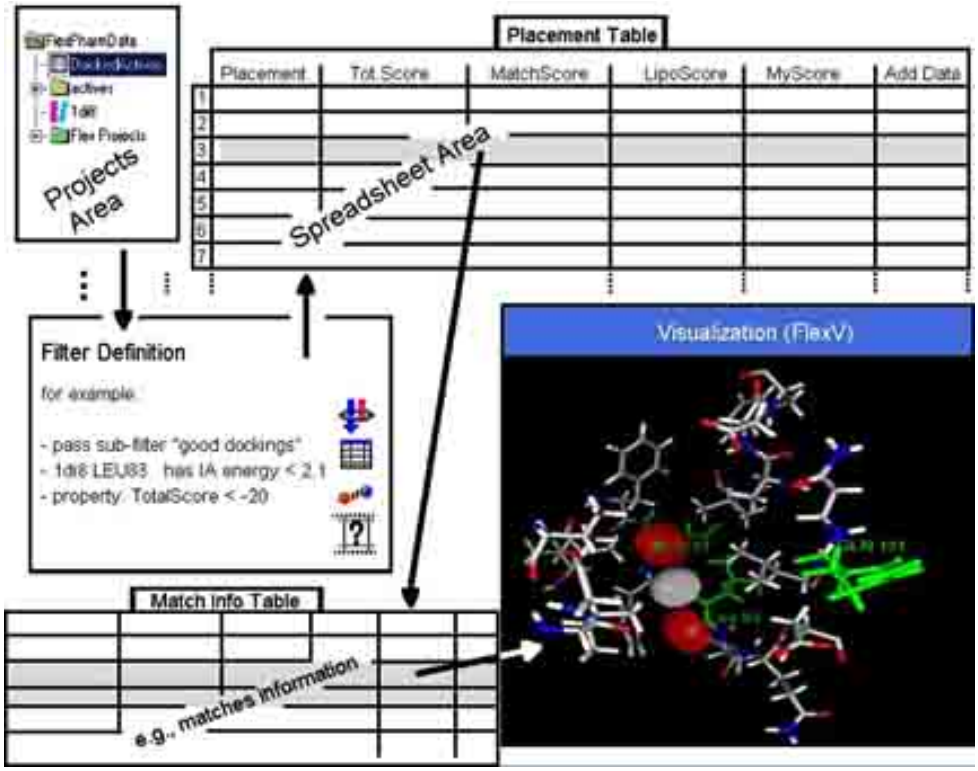

Fig. (2). Schematic of the Seeker GUI: The main seeker window contains the data tree (upper left Area, "Projects Area"), which represents the objects in the database, and a spread sheet that visualizes the docking results (upper right area, "Placement Table"). The columns can be sorted interactively, and new columns may be inserted by combining the already existing columns or by adding own data (not shown). The filter definition part is shown on the left hand side. FlexV visualization facilities (lower right) are used for interactive 3D display; interactions can be viewed in detail. For a full screen shot, please refer to http://www.biosolveit.de/ddb/gui.

medicinal-chemical research around CDK2 is prominent in the field of anticancer treatment.

ACE is known to play a key role in the regulation of cardiovascular and renal functions [48], as well as fertility [49]. Today, research on ACE inhibitors is mostly relevant in the context of the therapy of high blood pressure, because it catalyzes the conversion of angiotensin I to angiotensin II, the latter causing an increase in blood pressure [42].

2. RESULTS AND DISCUSSION

2.1. Method Calibration and Pharmacophore Extraction

For the three target protein structures, sets of pharmacophore constraints have been derived by docking known active compounds and subsequent data analysis. Assuming that the more often an atom in the target forms interactions with atoms from active compounds in the docking solutions, the more relevant this atom is as a pharmacophore counter-group, we analyzed the following: Given a subset of known active compounds, determine the frequencies of interactions with individual pharmacophore counter-groups on the target. In order to ascertain statistical significance, this process has been performed ten times, for ten differently and randomly composed subsets. More details can be found in the experimental section below (see Section 3.3). The interaction frequencies for the six top-ranking interactions for each target can be seen in Tab. (**I**).

a. Thrombin

For thrombin the database query (see the experimental section below) delivered a clearly distinguishable set of three most frequent interactions followed by a fourth slightly less frequent interaction on Gly 216 (Tab. (**Ia**)). Considering the ranks "highest/lowest" for positions 5. and 6., it may appear puzzling that the same entries may occur multiple times in different lines. This is actually due to the fact that the respective interactions are not "served" in every subset of our split-up in groups for the statistical analysis (see Section 3.3 for more details on this). We therefore, ruled out such interactions as not sufficiently relevant for a receptor-based pharmacophore constraint.

For this initial test, we decided to define all interactions as "optional" constraints, but demanded at least two of them must be fulfilled. Further, one has to be aware of the fact that sometimes there is an ambiguity in the assignment of interaction surfaces: For example, the setup of interaction geometries for carboxylate oxygens is such that each oxygen

has *two* surfaces with which to interact — corresponding to the two lone pairs. In FlexX, these surfaces are distinguished by the labels 0 and 1. This fact has to be taken into account when defining the phamacophore constraints. For both oxygen atoms of Asp 189, we chose the surfaces labelled 1, because these were the ones pointing towards the active site.

b. CDK2

Similarly to the case above, we stopped at position 4. when considering putative pharmacophore constraints, because both Ile 10 and His 84 do not make interactions in *all* of our subsets (Tab. (**Ib**)).

Visual inspection also revealed that residue Gln 131 lies at the rim of the binding pocket (see the visualization part in Fig. (**2**)) and it is therefore reasonable to refrain from employing it as a constraint. The most frequently occurring interaction (the H-donor interaction on the Leu 83 nitrogen atom) has been defined as an "essential" pharmacophore constraint. Additionally, interactions on the two Glu 81 and Leu 83 oxygen atoms served as "optional" constraints.

The resulting set of constraints reached using this process has been documented as a useful pharmacophore definition before [27, 50] and is in fact also consistent with crystal data [51, 52]. The CDK2 receptor-based pharmacophore constraints are depicted in Fig. (**3**).

c. ACE

The ligand-based pharmacophore of ACE inhibitors dictates a constellation containing an acidic group, a carbonyl group and a zinc-complexing group [53, 54]. Since the publication of the respective 3D-target structure dates back only a few months [55], it is not yet clear which interacting groups on the target represent the pharmacophore counter-groups.

As for the two preceding cases, we only looked at positions 1. – 4. of our statistics in Tab. (**Ic**) and discarded all subsequent atoms. Positions 2. – 4. in Tab. (**Ic**) are well separated from position 1. which is why we decided to include interactions to them (Tyr 523, Gln 281 and Lys 511) in our pharmacophore constraint set as "optional" constraints. At least two out of those had to be fulfilled. In contrast to these, the average frequency count produced the interaction with zinc as the leading interaction. Hence, we defined position 1. as an "essential" pharmacophore constraint.

The X-ray complex with Lisinopril (LPR) as found in the PDB (code 1o86) reveals that the zinc atom is coordinated by one of the two carboxylic groups of LPR (O3/O2). Let us consider this choice in a bit more detail: Considering the O–O-distance $O(H)_{Tyr523}$–$O3_{LPR}$ is 2.8 Å, it becomes clear that there must be an interaction of LPR with the phenol function of Tyr 523. The Gln 281 forms a hydrogen bridge to one of the LPR carboxylate groups, namely the one connected to the substituted pyrrolidine ring. The second oxygen of the same carboxylate function finally finds a counter-group in an NE2-bound hydrogen from Lys 511. So, the pharmacophore constraints derived above are in accordance with the H-bonding pattern encountered in the only available crystal structure of ACE in the literature so far. The derived pharmacophore constraint set is highlighted in Fig. (**4**) of the receptor in complex with LPR.

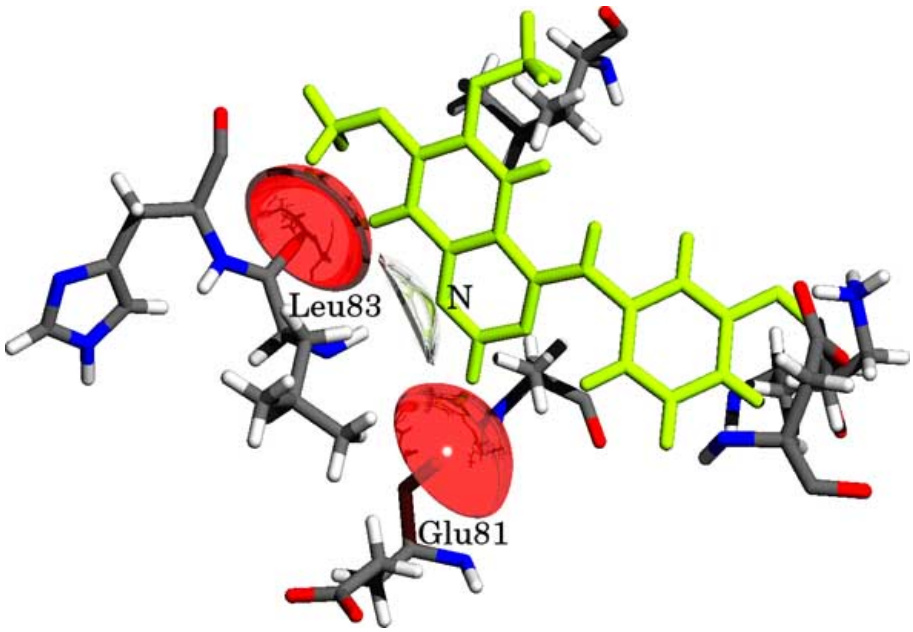

Fig. (3). CDK2: Extracted receptor-based pharmacophore constraints (red: H-acceptors, white: H-donor), the ligand DTQ is plotted in green, Leu 83 forms both donor- and acceptor interactions.

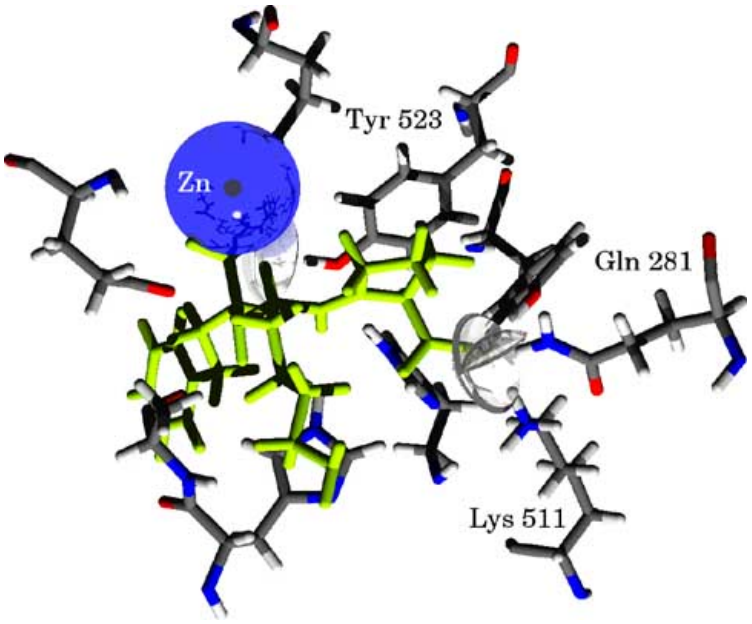

Fig. (4). ACE: Extracted receptor-based pharmacophore constraints (white: H-donor interactions). The metal interaction on the zinc atom is represented by the blue sphere.

2.2. Screening Runs and Enrichments

In all three cases, we registered an improvement of the enrichment between FlexX (without the receptor-based pharmacophore constraints) and FlexX-Pharm (with the constraints), most significantly for thrombin (Figs. (**5**) to (**7**)). Due to the fact that we apply all pharmacophore filtering *during* the actual docking, the ligands which do not comply with the constraints are not placed at all. For all cases there is a small fraction of active compounds which cannot be docked with standard FlexX. Spot checks give a hint that this may be partly due to wrongly defined or assigned ligand atom types. When docking under constraints with FlexX-Pharm, a larger fraction of active compounds cannot be docked (this can for example be seen in the ACE and CDK2 enrichment plots at around 40% of the sorted list of library compounds screened).

The reason for this is simply because these active compounds do not fit the constraints applied. In order to find a set of constraints that all active compounds may fulfill, a stricter, more detailed analysis of the interaction statistics could be carried out. However, care must also be taken not to over-fit the data. It may also be the case that using FlexX, no such set of constraints can be found. This illustrates the risk that must be considered in this process; a balance must be found between the effort involved to derive the required information and the profit gained at the end of the process.

In the following, we will use the term "enrichment factor" $e(x)$

$$e(x) = \frac{\text{accumulated \% of known active compounds in experiment E}(x)}{\text{accumulated \% of known active compounds expected for a random selection}} \quad \textbf{(1)}$$

as the fraction with x being the percentage of the ranked database subset. So, the enrichment factor reflects the gain in the ability to select active compounds compared to the random scenario. The x-axis in the plots has been plotted logarithmically to emphasize the "interesting" regions which lie at the very beginning of an enrichment plot.

a. Thrombin A significant improvement in enrichment between FlexX and FlexX-Pharm was found for this target (see Fig. (**5**)). Especially in the early percentage of the ranked database subset around one percent, the enrichment factor lies at around 55 compared to ≈ 30 before.

b. CDK2 The result is similar but not that pronounced for the CDK2 target. Here, enrichment factors of around 15 are reached early in the sampled ranked database subset at around one percent (see the plot in Fig. (**6**)) while there is clearly a nice improvement up to around 40% of the ranked database subset.

c. ACE Seen from an *a priori* point of view, it appears particularly difficult to improve the already good enrichment for FlexX as seen in Fig. (**7**). For example, at one percent of all screened compounds, the enrichment factor is at about 40. In fact, there is only slight but apparent amelioration *with* the derived set of pharmacophore constraints. Again looking at the one percent mark, the enrichment factor rises to about 42. The higher quality of the enrichment is basically maintained until the cut-off, where no more active compounds were docked.

All in all, the enrichment plots show a consistent improvement obtained with the derived set of pharmacophore constraints. This result lends confidence to the

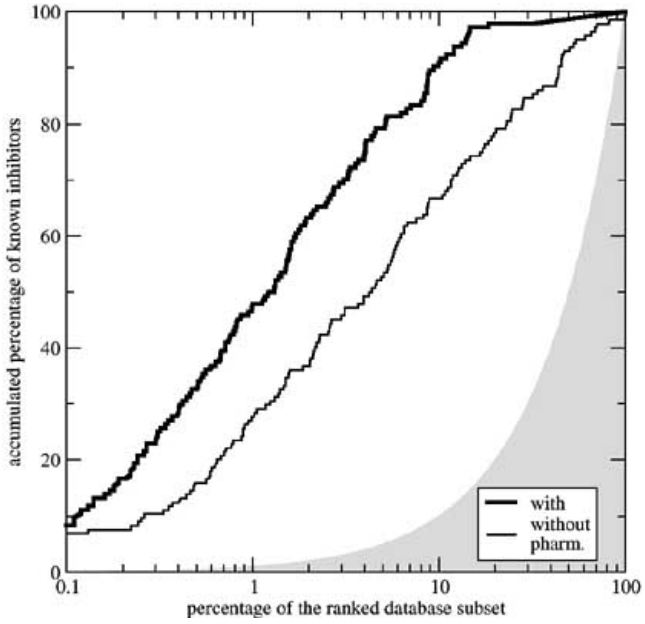

Fig. (5). Thrombin enrichments with and without receptor-based pharmacophore constraints. Plotted is the percentage of active compounds found with respect to percentage of data screened (logarithmic scale).

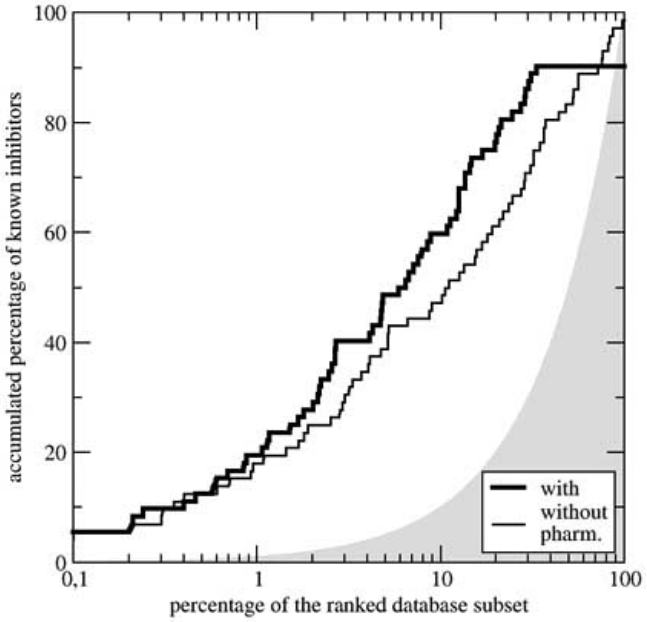

Fig. (6). CDK2 enrichments with and without pharmacophore definitions. Plotted is the percentage of active compounds found with respect to percentage of data screened (logarithmic scale).

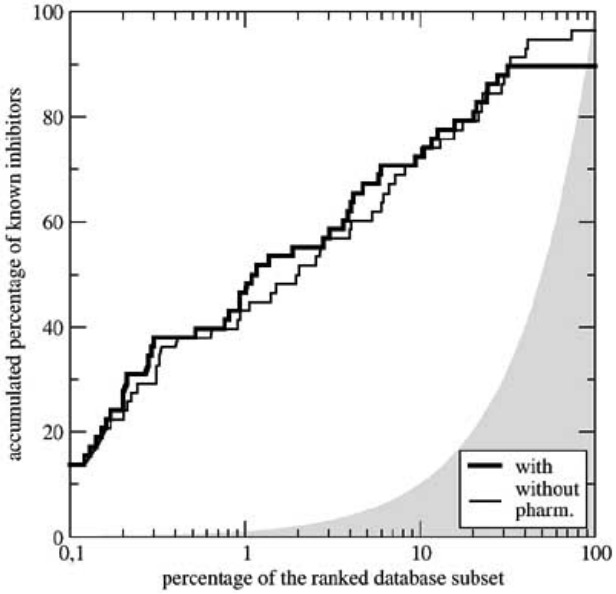

Fig. (7). ACE enrichments with and without pharmacophore definitions. Plotted is the percentage of active compounds found with respect to percentage of data screened (logarithmic scale).

pharmacophore constraint selection procedure and also validates the underlying assumption that docking results are actually biased towards more physically meaningful placements.

3. METHODS

3.1. Docking

3.1.1. Background and Current Problems

Molecular docking tools are still not reliable enough to be used as black box filters. A lot of manual preparation has to be done before a successful virtual screening may be expected. Typically, after having prepared the target structure, a small test set of ligands (known active compounds plus randomly selected compounds) is docked to optimize various parameters and to potentially tune the scoring, e.g. by re-scoring or post-filtering. The obvious aim of this process is to learn about the specifics of the particular target and to improve the enrichment in subsequent screening runs. This is an interactive and iterative process requiring considerable analysis and experience.

Various tools are used not only to prepare the input and run the docking, but also to visualize and analyze the results. Also re-scoring and post-filtering of solutions is often done externally with separate tools. In addition, the amount of data involved in such a setup may become rather large and complex. Files are oftentimes the only means of data exchange yet originate in different locations. The ligands might be fetched from a library whereas target information, scripts, and output are frequently stored in other separate locations. The docking predictions are usually dumped as flat file output requiring a lot of parsing in order to be at all accessible to analysis. Even with the most meticulous bookkeeping, the risk of inconsistency in the data in such a scenario is enormous and, even worse, makes the whole process rather error prone.

3.1.2. Integrated Environment

In order to overcome the above problems, and with the further aim of enabling the detection of Structure Activity Relationships (SAR) and relevant features for ligand binding from docking data, we developed an integrated environment that supports the entire virtual screening workflow (see Fig. (**1**)). The environment consists of three main components which are dealt with in detail here; the database, a modified FlexX version and the Seeker, all "glued" together with Python [38] as an interfacing language.

a. The Database

An ORACLE database [37] provides the central building block of the docking environment. The advantages of using a central database are manifold: First of all, the approach guarantees consistency by unique reference. Then, due to the structured storage of the docking information in the database schema, no parsing is necessary when analyzing the computed placements. The restriction to only a few top-ranking solutions becomes unnecessary. All information about all placements including all scoring parameters and the actual 3D coordinates of the placements are stored in the database. Moreover, reasonable solutions but ones that originally ranked low percolate up to the top in re-scoring experiments frequently. Also, the computational results are all reproducible, because the complete setup, down to the version number of the executable is stored as well. This is especially helpful if additional ligands need to be added at a later date.

The database scheme has been optimized to gain efficient access to the information about the predicted placements; for

example an overview of which intermolecular interactions are formed may be neatly arranged in several tables. In contrast, the input files are stored *en bloc* without further decomposition. Often databases are used to store edited data persistently in the long term. Note that we utilize the database here in a slightly different manner, since we store raw data for later analysis and use the database mainly for bookkeeping, structured deposition, and integrated analysis. Detailed docking results are expected to be kept only for the life time of a particular project. As a consequence, the rate of data generation during the docking, and the amount of data that has to be scanned for analysis is rather large. This was taken into account in the database setup, mainly via a provision of large main memory of the database server and fast hard disk access. In addition, the data distribution in the database tables may change rapidly due to its more temporary nature. This causes problems for an automated optimization of queries to the database. This effect is still under investigation.

b. The Database Interface

The database interfaces are written in Python [38]. Python is an open-source, cross-platform, full-featured programming language, which can be used for both high level object-oriented programming and simple scripting. For the latter reason it was found to be a suitable replacement for the current proprietary FlexX scripting language. To this end, we developed a Python module, called PyFlexX, that provides all FlexX commands as Python functions. The PyFlexX functions are wrapped around the original C-functions using shared memory between Python and C. This allows for easy combination of the complete FlexX functionality with other Python modules without losing any of the speed of the computation. PyFlexX is part of the current FlexX release. Using Python, it is very easy to combine FlexX functionality with additional external modules, which are numerous in the area of chemical informatics (see http://www.chempython.org for a comprehensive overview). Finally, PyFlexX forms an ideal basis for rapid prototyping of workflows.

The FlexX database interface is also based on PyFlexX. We have extended the capabilities of FlexX to read its input directly from and store the results directly into the ORACLE database [37]. In principle, the data output of the FlexX command DOCKING/LISTALL is stored in the database; this comprises all scoring terms, the total score, the different overlapping volumes, and all necessary detail about formed interactions that enable the calculation of a score for the interaction. Besides the terms of the original FlexX scoring function [11], additional scoring terms related to the functions ChemScore [56], DrugScore [57], PLP [58, 59] and terms of the Tripos force field [60] are also provided.

c. The Seeker

The Seeker, which is written in Python too, has several functions: Firstly it forms the front end for the database. In this capacity it enables viewing, importing, and exporting data to and from the database while relieving the user of the burden of directly formulating lengthy SQL statements. No detailed knowledge about the database scheme is necessary. Secondly, Seeker is the GUI for the docking program. The user can assemble docking experiments interactively, view the current status of docking runs, and control these jobs. Thirdly, it is the browser for docking results. The data can be analyzed by interactive spread sheets that allow for sorting, showing and hiding of arbitrary columns. Also, new columns can be added by combining other columns (see for example the placement table in the top right of the schematic in Fig. (**2**)). Filters can be applied to scoring terms and interaction patterns. In this way rapid prototyping of novel scoring functions derived from the given raw docking data is possible. Visualization is provided by 2D and 3D viewers (see lower right of Fig. (**2**)). FlexV functions as our 3D viewer, particularly as it is capable of highlighting certain interactions of interest.

3.2. Data Preparation

3.2.1. Protein Structures

Docking with FlexX requires the user to define the active site. This may be done based on the coordinates of a complexed ligand. Further definition of protonation and tautomeric states of the amino acids, based on default templates is also required. Finally, the torsional angles of hydroxyl groups of serine and threonine, and the hydrogen position of histidine side chains have to be provided by the user. For the cases presented here chemically reasonable hydrogen bonding patterns were obvious. Side chains of lysine and arginine are protonated, asparagine and glutamine carboxylate groups are deprotonated [10].

For all targets, the active sites were determined by including all atoms within a distance of 6.5 Å from the atoms of the complexed ligand, no water was included, whereas metal atoms were always kept in the protein description. We shall give further target-specific details below.

Thrombin As an initial test we took the already prepared input data for thrombin from the FlexX-200 test set [10]. Redocking was successful with the best-ranked solution exhibiting an RMS value of 1.7 Å, and the placement with smallest RMS (0.8 Å) at rank 2.

CDK2 Though there is a variety of CDK2 structures available from the PDB, we have arbitrarily selected the structure with PDB code 1di8. This is a complex structure with DTQ (4-[3-hydroxyanilino]-6,7-dimethoxyquinazoline). Redocking DTQ into CDK2 does not pose problems: The best-ranked solution exhibited an RMS value of 1.3 Å, the best RMS appears on rank 9 with 0.9 Å.

ACE Only recently the first ACE structure (in complex with the inhibitor Lisinopril (LPR)) has been deposited in the PDB (code: 1o86) [55]. It exhibits a tube-shaped cavity as its active site. All ligand-related operations in this context will refer to LPR. When redocking LPR, the base fragment placement was restricted to a pocket of 4.5 Å radius around LPR atoms to avoid unreasonable placements at the rim of the pocket. For the same reason we removed the Arg 522 residue from the pocket definition. It too is located at the rim of the pocket and because of its three charged guanidino functions, docking solutions would tend to cluster around this residue. This is a commonly recognized problem with Arg residues when docking in FlexX. Explicitly defined tautomeric forms of histidines have been used for His 383

and His 387 after visual inspection, such that the histidines correctly complex the Zn ion. Finally, two hydrogen-related torsional angles have been set as to optimize the hydrogen bond pattern: Tyr 520 and Ser 516. All other amino acid templates have been left in their respective default states. LPR was extracted from the PDB structure and cleaned up manually. The subsequent optimization of the conformation prior to docking was computed with the Tripos force field [60] as implemented in FlexX. To check the validity of all preprocessing, we redocked Lisinopril into the 3D structure. The top ranked ligand conformation exhibited an RMS value of 1.0 Å. The best RMS was found at rank 2 with 0.9 Å. All three re-dockings confirm that the targets are well-specified for docking and the applied algorithms and the scoring function are appropriate.

3.2.2. Ligands

All information on active compounds were collected from the literature. The set of thrombin inhibitors [61–63] taken as the set of active compounds has already been used in former docking studies [4, 24, 61]. We obtained 72 known inhibitors as the active compounds for CDK2 from the literature [64–69]. The known ACE inhibitors were taken from ref. [70] and used as the active compound set. Similarly to thrombin, this set had been employed in earlier virtual screening studies [4].

As the library of diverse compounds considered to be active, we used the Bionet library [71], which is freely available and consists of approximately 34000 screening compounds. The protonation of all Bionet ligands was made according to FlexX defaults (acids are deprotonated, only tertiary amines are protonated). Prior to all screening we have ensured zero overlap between the Bionet library and the respective set of known active compounds for each case. All compounds were subject to 3D-coordinate generation with Corina version 3.0 [72].

3.3. Statistical Analysis

FlexX predicts several placements for each docked ligand, and the placement with minimum RMSD is not always top-ranked. Frequently however, many reasonable placements lie somewhere among the ones produced. We assume that a slight bias towards good (partial) placements should be reflected by a bias to match individual important interactions. Therefore, we counted how often (in how many of the all the generated placements) a particular interaction with a target atom was formed. The counting was done for a number of known active compounds for the target and iterated for every interacting atom in the active site of the target and over *all* predicted solutions. In order to ascertain statistical significance, we randomly selected subsets of active compounds and performed the analysis ten times with different subsets. In Tab. (**I**), the output is listed by the average counts resulting in the respective positions; comparing all subsets we have traced the highest and lowest ranks within all ten subsets. If an interaction did not occur at all in a subset, we did not consider it as a pharmacophore constraint. Note that, in such a way, all highest/lowest ranks for two interactions can emerge the same. *A priori*, i.e. without having taken into account any results beforehand, only hydrogen donors/acceptors and metal interactions (as defined in FlexX) were taken into account. The former are included because they are "spatially selective". Because metals are of crucial importance for coordination chemistry, we included this type of interaction as well. All other interaction types were neglected in the analysis. The subsets each contained 58 out of 68 active compounds for thrombin, 60 out of 72 in the case of CDK2, and 50 out of 58 in the case of ACE, respectively. This choice offered a balanced trade-off between information content and diversity within the subsets.

3.4. Docking and Screening

For all dockings we used the Python module of FlexX version 1.12.3. FlexX-Pharm is an integrated module of this FlexX version, and allows the inclusion of receptor-based pharmacophore constraints in the active site. FlexX-Pharm determines, which partially built docking solutions can potentially obey these constraints. Solutions that will not obey the constraints are deleted as early as possible, often even before actually docking the ligand. This decreases the calculation time and enables new docking solutions to emerge. More technical details about the algorithms used in FlexX [11, 61, 73, 74] and FlexX-Pharm [26] have been described.

Throughout the docking procedures standard parameters were used. The ligands were provided in multi-mol2 format. Formal charges were assigned automatically by FlexX and distributed across delocalized systems. Corina version 3.0 [72] was used to generate multiple ring conformations during docking. Empirical scoring functions were used to score the placements. The standard FlexX function [11], which is closely related to the Böhm function [75], was used for the initial test with Thrombin and for ACE. For CDK2 we used the ScreenScore [24] because is known to work better for rather hydrophobic targets like CDK2.

For all targets, the active compounds were merged with the Bionet library [71] and all ligands were docked into the respective target structures. The compounds were ranked according to their docking score and the accumulated percentage of known inhibitors was plotted against the percentage of the ranked database subset (see (Figs. (**5**) to (**7**)). The screening was done both with and without the derived receptor-based pharmacophore constraints in order to evaluate their effect on the enrichment.

All calculations were performed on a 20-node LinuX cluster (10 Athlon MP 2000+ double processor machines, 2GB, SuSE 8.0). For the ORACLE server we employed an Athlon MP 1800+ double processor machine with 4 GB RAM using SuSE 8.1 [76], ORACLE 9.2i [37], and a RAID 1 system with a net capacity of 240 GB. All machines are connected via a 100 MBit fast Ethernet and access the same NFS.

4. SUMMARY

This work describes recent extensions to the molecular docking program FlexX. Based on the FlexX python module, PyFlexX, which allows for easy handling and rapid prototyping of automatized workflows, we implemented an integrated docking environment for FlexX.

The central element of this environment is an ORACLE database, in which all necessary input data, the configuration of the docking runs, and the complete information about all predicted placements are stored. This structured storage of the solutions allows for easy detailed interactive and iterative analysis of the results. The docking environment facilitates rapid experimentation with various filters and different scoring terms in order to improve the enrichment on a given test set. The derived target-specific scoring function can be made persistent and used for subsequent large virtual screening runs.

As an example application, we extracted receptor-based pharmacophore constraints for the targets thrombin, CDK2, and ACE. We used the integrated docking environment to derive target-specific constraints by statistical analysis of ligand-target interaction counts based on the generated placements. By means of this relatively simple statistical analysis, we found valuable insights in the binding behavior and were able to extract pharmacophore constraints that further significantly improved the enrichment in the virtual screening run. The advantage of this procedure becomes clear for the ACE target. Here, only one complex has been crystallized so far, but a considerable number of active inhibitors are known. Using these inhibitors, a receptor-based constraint hypothesis could be derived that in fact improved the already good enrichment obtained from screening. This set of pharmacophore constraints is not based on comparison of several crystal structures (which are not available in case of ACE) but on a virtual docking study and a simple statistical analysis. Summarizing, the process of extracting target-specific information for enhancing virtual screening may be achieved in a more rational, efficient, and consistent fashion using the new FlexX database docking environment.

ACKNOWLEDGEMENTS

This work is part of a cooperation project with Bayer Cropscience AG, Germany. The authors thank Robert Klein, Gudrun Lange and Jürgen Albrecht for many helpful discussions. We also express our gratitude to Andreas Steffen (University of Marburg, Germany) for the collection of the known CDK2 inhibitors and to Hugo Kubinyi for various comments and suggestions on this work.

REFERENCES

[1] Abagyan, R.; Totrov, M. *Curr. Opin. Chem. Biol.* **2001**, *5*, 375–382.
[2] Davis, A.; Teague, S.; Kleywegt, G. *Angew. Chem. Int. Ed.* **2003**, *42*(24), 2718–2736.
[3] Lyne, P. D. *Drug. Discov. Today* **2002**, *7*, 1047–1055.
[4] Stahl, M.; Rarey, M.; Klebe, G. In *Bioinformatics - From Genomes to Drugs*; Lengauer, T., Ed., Vol. II; Wiley-VCH, Heidelberg, 2001; pages 137–170.
[5] Rarey, M. In *Bioinformatics – From Genomes to Drugs*; Lengauer, T., Ed., Vol. I; Wiley-VCH, Heidelberg, 2001; pages 315–360.
[6] Schneider, B.; Böhm, H. *Drug Disc. Today* **2002**, *7*(1), 64–70.
[7] Morris, G.; Goodsell, D.; Halliday, R.; Huey, R.; Hart, W.; Belew, R.; Olson, A. *J. Comput. Chem.* **1998**, *19*, 1639–1662.
[8] Makino, S.; Kuntz, I. *J. Comput. Chem.* **1997**, *18*(4), 1812–1825.
[9] Meng, E.; Shoichet, B.; Kuntz, I. *J. Comput. Chem.* **1992**, *13*(4), 505–524.
[10] Kramer, B.; Rarey, M.; Lengauer, T. *Proteins* **1999**, *37*, 228–241.
[11] Rarey, M.; Kramer, B.; Lengauer, T.; Klebe, G. *J. Mol. Biol.* **1996**, *261*(3), 470–489.
[12] OpenEye Scientific Software, Santa Fe, NM, USA, http://www.eyesopen.com/fred/index.html.
[13] Schrödinger, LLC, New York, NY, USA, http://www.schrodinger.com/Products/glide.html.
[14] Jones, G.; Willett, P.; Glen, R. *J. Mol. Biol.* **1995**, *245*, 43–53.
[15] Jones, G.; Willett, P.; Glen, R.; Leach, A.; Taylor. R. *J. Mol. Biol.* **1997**, *267*, 727–748.
[16] Gohlke, H.; Klebe, G. *J. Med. Chem.* **2002**, *45*(19), 4153–70.
[17] Tame, J. *J. Comput.-Aided Mol. Des.* **1999**, *13*, 99–108.
[18] Gohlke, H.; Hendlich, M.; Klebe, G. *Angew. Chem. Int. Ed.* **2002**, *41*, 2644–2676.
[19] Stahl, M.; Böhm, H. *J. Mol. Graph. Mod.* **1998**, *16*, 121–132.
[20] Bissantz, C.; Folkers, G.; Rognan, D. *J. Med. Chem.* **2000**, *43*(25), 4759–67.
[21] Charifson, P.; Corkery, J.; Murcko, M.; W.P., W. *J. Med. Chem.* **1999**, *42*, 5100–5109.
[22] Paul, N.; Rognan, D. *Proteins* **2002**, *47*, 521–533.
[23] Wang, R.; Lu, Y.;Wang, S. *J. Med. Chem.* **2003**, *46*(12), 2287–2303.
[24] Stahl, M.; Rarey, M. *J. Med. Chem.* **2001**, *44*, 1035–1042.
[25] Verdonk, M.; Cole, J.; Hartshorn, M.; Murray, C.; Taylor, R. *Proteins* **2003**, *52*(4), 609–623.
[26] Hindle, S.; Rarey, M.; Buning, C.; Lengauer, T. *J. Comput.-Aided Mol. Des.* **2002**, *16*, 129–149.
[27] Hindle, S.; Stahl, M.; Rarey, M. In *Proceedings of Euro-QSAR 2002*, Oxford, UK, in press. Blackwell Publishing Ltd.
[28] Schulz-Gasch, T.; Stahl, M. *J. Mol. Mod.* **2003**, *9*(1), 47–57.
[29] Thomas IV, B.; Joseph-McCarthy, D.; Alvarez, J. In *Pharmacophore Perception, Development and Use in Drug Design*; Güner, O., Ed.; International University Line: La Jolla, California, USA, 2000; pages 351–367.
[30] Bruno, I.; Cole, J.; Lommerse, J.; Rowland, R.; Taylor, R.; Verdonk, M. *J. Comput.-Aided Mol. Des.* **1997**, *11*, 525–537.
[31] Goodford, P. *J. Med. Chem.* **1985**, *28*, 849–857.
[32] Masukawa, K.; Carlson, H.; McCammon, J. In *Pharmacophore Perception, Development, and Use in Drug Design*; Güner, O., Ed.; International University Line: La Jolla, California, USA, 2000; pages 409–427.
[33] Venkatachalam, C.; Kirchho., P.; Waldman, M. In *Pharmacophore Perception, Development, and Use in Drug Design*; Güner, O., Ed.; International University Line: La Jolla, California, USA, 2000; pages 339–350.
[34] Verdonk, M.; Cole, J.; Taylor, R. *J. Mol. Biol.* **1999**, *289*, 1093–1108.
[35] Magnet: A tool to analyze database docking results and generate tailor-made scoring functions. Jansen, J. M.; Martin, E. J. 4th European workshop on Drug Design, Certosa di Pontignano, May 2003, oral communication.
[36] Watson, P.; Verdonk, M. Hartshorn, M. *J. Mol. Graph. Mod.* **2003**, *22*, 71–82.
[37] Oracle version 9.2i (9.2.0.1.0), Oracle Corporation, Redwood Shores, CA, USA, http://www.oracle.com/.
[38] Python version 2.2.1, http://www.python.org.
[39] Knegtel, W.; Wagner, M. *Proteins* **1999**, *37*, 334–345.

[40] Nissink, J.; Murray, C.; Hartshorn, M.; Verdonk, M.; Cole, J.; Taylor, R. *Proteins* **2002**, *49*, 457–471.
[41] Berman, H.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T.; Weissig, H.; Shindyalov, I.; Bourne, P. *Nucleic Acid Res.* **2000**, *28*, 235–242.
[42] Böhm, H.-J.; G., K.; H., K. *Wirkstoffdesign;* Spektrum Akademischer Verlag GmbH: Heidelberg, **1996**.
[43] Dullweber, F. *Strukturelle und physikochemische Charakterisierung der Protein-Ligand-Wechselwirkung am Beispiel der Serinproteasen Thrombin und Trypsin*, PhD thesis, University of Marburg, Marburg, Germany, **2000**.
[44] Karlson, P.; Doenecke, D.; Koolman, J. *Biochemie für Mediziner und Naturwissenschaftler;* Georg Thieme Verlag, 1994.
[45] Grim, J. E.; Clurman, B. E. *Trends Cell Biol.* **2003**, *13*(8), 396–399.
[46] Malumbres, M.; Hunt, S. L.; Sotillo, R.; Martin, J.; Odajima, J.; Martin, A.; Dubus, P.; Ortega, S. *Adv. Exp. Med. Biol.* **2003**, *532*, 1–11.
[47] Stead, E.; White, J.; Faast, R.; Conn, S.; Goldstone, S.; Rathjen, J.; Dhingra, U.; Rathjen, P.; Walker, D.; Dalton, S. *Oncogene* **2002**, *21*(54), 8320–8333.
[48] Niu, T.; Chen, X.; Xu, X. *Drugs* **2002**, *62*(7), 977–93.
[49] Hagaman, J.; Moyer, J.; Bachman, E.; Sibony, M.; Magyar, P.; Welch, J.; Smithies, O.; Krege, J.; O'Brian, D. *Proc. Nat. Acad. Sci. U.S.A.* **1998**, *95*(5), 2552–2557.
[50] Hecker, E. A.; Duraiswami, C.; Andrea, T. A.; Diller, D. J. *J. Chem. Inf. Comput. Sci.* **2002**, *42*(5), 1204–11.
[51] Moliner, E. D.; Brown, N. R.; Johnson, L. N. *Eur. J. Biochem.* **2003**, *270*, 3174–3181.
[52] Young, S. S.; Keefer, C. E. In *Chemical Data Analysis in the Large, May 22nd -26th 2000, Bozen, Italy*, pages 78–82. Beilstein-Institut, 2000.
[53] Bersuker, I. B.; Bahceci, S.; Boggs, J. E. *J. Chem. Inf. Comput. Sci.* **2000**, *40*(6), 1363–1376.
[54] Mayer, D.; Naylor, C. B.; Motoc, I.; Marshall, G. R. *J. Comput.-Aided Mol. Design* **1987**, *1*, 3–16.
[55] Natesh, R.; Schwager, S.; Sturrock, E.; Acharya, K. *Nature* **2003**, *421*, 551.
[56] Eldridge, M.; Murray, C.; Auton, T.; Paolini, G.; Mee, R. *J. Comput.-Aided Mol. Des.* **1997**, *11*, 425–445.
[57] Gohlke, H.; Hendlich, M.; Klebe, G. *J. Mol. Biol.* **2000**, *295*, 337–356.
[58] Gehlhaar, D.; Verkhivker, G.; Rejto, P.; Sherman, C.; Fogel, D.; Fogel, L.; Freer, S. *Chem. & Biol.* **1995**, *2*, 317–324.
[59] Gelhaar, D.; Verkhivker, G.; Rejto, P.; Fogel, D.; Fogel, L.; Freer, S. In McDonnell, J., Reynolds, R., Fogel, D., Eds., *Proceedings of the Fourth Annual Conference on Evolutionary Programming*, pages 615–627, 1995.
[60] Clark, M.; Cramer, R.; Opdenbosch, N. *J. Comput. Chem.* **1989**, *10*, 982–1012.
[61] Rarey, M. *Rechnergestützte Vorhersage von Rezeptor-Ligand-Wechselwirkungen*, Vol. 268 of *GMD-Berichte;* R. Oldenbourg Verlag, 1996.
[62] Sanderson, P.; Nayer-Olsen, A. *Curr. Med. Chem.* **1998**, *5*, 289–304.
[63] Wiley, M.; Fisher, M. *Exp. Opin. Ther. Pat.* **1997**, *7*, 1265–1282.
[64] Gray, N.; Detivaud, L.; Doerig, C.; Meijer, L. *Curr. Med. Chem.* **1999**, *6*(9), 859–875.
[65] Sielecki, T.; Boylan, J.; Benfield, P.; Trainor, G. *J. Med. Chem.* **2000**, *43*(1), 1–18.
[66] Hardcastle, I.; Golding, B.; Griffin, R. *Ann. Rev. Pharm. Tox.* **2002**, *42*, 325–48.
[67] Kim, K. S.; Kimball, S. D. *et al. J. Med. Chem.* **2002**, *45*(18), 3905–3927.
[68] Knockaert, M.; Greengard, P.; Meijer, L. *Trends Pharmacol. Sci.* **2002**, *23*(9), 417–25.
[69] Personal communication. Steffen, A. University of Marburg, Germany.
[70] Fink, C. *Exp. Opin. Ther. Pat.* **1996**, *6*, 1147–1164.
[71] Bionet Screening Compounds Database, Key Organics Limited UK, http://www.keyorganics.ltd.uk/screenin.htm.
[72] Sadowski, J.; Gasteiger, J. *Chem. Rev.* **1993**, *93*, 2567–2581.
[73] Rarey, M.; Kramer, B.; Lengauer, T. *J. Comput.-Aided Mol. Des.* **1997**, *11*, 369–384.
[74] Rarey, M.; Kramer, B.; Lengauer, T. *Bioinformatics* **1999**, *15*, 243–250.
[75] Böhm, H.-J. *J. Comput.-Aided Mol. Des.* **1994**, *8*, 243–256.
[76] SuSE Linux AG, Nuremberg, Germany http://www.suse.com/.