J Comput Aided Mol Des (2006) 20:717–734
DOI 10.1007/s10822-006-9091-x

ORIGINAL PAPER

Ultrafast de novo docking combining pharmacophores and combinatorics

Marcus Gastreich · Markus Lilienthal · Hans Briem · Holger Claussen

Received: 7 June 2006 / Accepted: 24 October 2006 / Published online: 30 January 2007
© Springer Science+Business Media B.V. 2007

Abstract We report on a successful de novo design approach which relies on the combination of multi-million compound combinatorial docking under receptor-based pharmacophore constraints. Inspired by a rationale by A.R. Leach et al., we document on the unification of two steps into one for ligand assembly. In the original work, fragments known to bind in protein active sites were connected forming novel ligand compounds by means of generic skeleton linkers and following a combinatorial approach. In our approach, the knowledge of fragments binding to the protein has been expressed in terms of a receptor-based pharmacophore definition. The combinatorial linking step is performed in situ during docking, starting from combinatorial libraries. Three sample scenarios growing in size and complexity (combinatorial libraries of 1 million, 1.3 million, and 22.4 million compounds) have been created to illustrate the method. Docking could be accomplished between minutes and several hours depending on the outset; the results were throughout promising. Technically, a module compatibility between FlexX^{C} and FlexX-Pharm has been established. The background is explained, and the crucial points from an information scientist's perspective are highlighted.

Electronic Supplementary Material The online version of this article (doi: 10.1007/s10822-006-9091-x) contains supplementary material, which is available to authorized users.

M. Gastreich (✉) · M. Lilienthal · H. Claussen
BioSolveIT GmbH, An der Ziegelei 75, 53757 St. Augustin, Germany
e-mail: gastreich@biosolveit.de

H. Briem
CDCC/Computational Chemistry, Schering AG, 13342 Berlin, Germany

Keywords Combinatorial chemistry · Docking · FlexX · Pharmacophore · vHTS · Virtual screening

Introduction

The temptation to work out a working de novo design cycle

Docking, the prediction of 3D protein–ligand complex structures, is nowadays an ubiquitous procedure [1–6] despite its drawbacks [7–12]. Its main application fields are to learn about binding modes, and to do selection of promising lead or drug candidates from compound libraries.

In the context of the latter, there have been estimates that the relevant chemical solution space amounts to 10^{18}, 10^{29}, or even more than 10^{60} compounds [13, 14]. Recently, a study has described the generation of a space with 10^{13} virtual compounds [15]. Given these numbers, it is certainly obvious that a structure based, somewhat ''guided'' de novo design approach, i.e., the creation of lead compounds or drug candidates without a template ligand structure, would be a promising idea. (In wet chemistry, too, there are specially devoted journals on the market, and many successful applications of this basic approach have been reported to date (see for example Refs. [15–17] and references therein.)

On the other hand, looking at a number this large, it is also clear that we are faced with two major questions for suitable software: (a) How can we technically tackle 3D searches in such vast spaces; and (b) how can we rank-sort the solutions so that we have the most promising candidates for drug design at the top of the list (the ''scoring problem'')?

The search space has redundancies that can be expressed in terms of combinatorial approaches. It therefore comes natural that combinatorial approaches to medicinal chemistry problems are rapidly growing—as is the number of possible solutions to the docking problem. Yet, is has to be seen that the more possible tentative "solutions" we generate for a problem, the higher is the risk to end up with false positives contaminating our hit lists. It has been demonstrated in the past that the application of constraints or filters during the docking can successfully help in pushing back false positives [18–20]. This also complies with our own experience. One possible approach to realize constraints is to apply pharmacophore-like constraints which are usually spoken of in the context of ligands. We will apply similar constraints (and still term them "pharmacophore constraints" to the receptor [21, 22]. These will then have to be met during the actual docking procedure.[1] Similarly for the scoring problem: while there currently is no such thing as the universal, correct scoring function which can be evaluated rapidly, we try to filter out false positives as good as possible. So, filtering based on pharmacophores has a two-fold effect: the "constraint effect" accelerating a screening, and the "filter-effect" reducing the rate of false positives.

A successful application in the past

A similar approach to combine combinatorial *de novo design* with pharmacophore constraints has been followed in separate steps a couple of years ago. In their publication, Andrew R. Leach and colleagues from GSK, formerly Glaxo Wellcome, depicted the idea of an approach to combine combinatorial chemistry and constrained de novo design [24]. In the publication, the authors placed templates within an "acyclic chain whose length and bond orders" were "systematically varied." The resulting "generic" compounds were checked for their ability to link to fragments which had previously been proven to bind strongly in certain regions of a protein (cp. Fig. 1). In a subsequent step promising generic candidates were used as queries in a search for similar compounds under the constraint to be synthetically accessible. In the original publication, the first part of the protocol was characterized by separate phases:

1. Definition of strongly binding fragments and their orientations in protein binding sites.

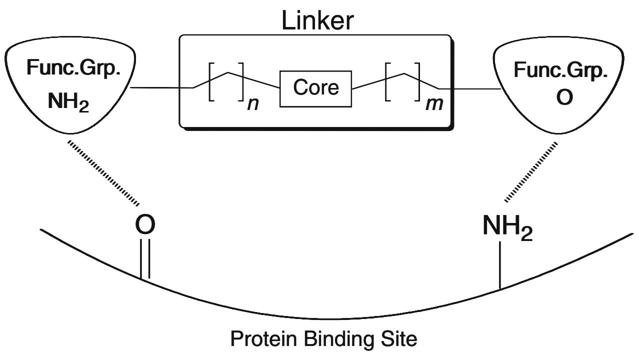


Fig. 1 Schematic display of one way to view the approach by Leach et al. to novel ligand inhibitor compounds (entire top block): varying the distance between so-called "templates" (i.e., fragments equipped with functional groups known to bind tightly in protein active sites) by modifying distances of generic "linkers" (cores plus chains) connecting the templates. We see an analogy between the functional group binding and receptor-based pharmacophore constraint definitions

2. Connection of the fragments by finding and placing generic linkers. A key feature of the approach were "core" templates, e.g., an amide group, imidazoles, etc. to which aliphatic "trial skeletons" were attached. This entailed an exploration of the linkers' torsional space to check for compliance with the connecting atoms of the tight binding fragments.

Another successful guided assembling approach has recently been published by Krier et al.: the authors walked along a similar line to optimize a phosphodiesterase 4 inhibitor. They assembled three "building blocks," a conserved "scaffold," a variable "linker," and a variable "functional group." The protocol has been termed the "Scaffold-Linker-Functional group (SLF) approach" and entailed a check for (ligand based) pharmacophore properties, a synthetic feasibility assessment, and (sequential) FlexX docking. An RMSD-based filter referring to a substructure was used to minimize the number of false positives. The procedure yielded, amongst other active compounds, a subnanomolar inhibitor [19].

Inspired by the underlying rationales, we herein demonstrate how such pharmacophore-guided, combinatorial de novo design can be performed on-the-fly during protein ligand docking. To this end, the flexible docking program FlexX-Pharm has been extended to be compliant with molecular fragments from combinatorial input, i.e., a module compatibility between FlexX-Pharm and FlexX^{C} has been established.

To demonstrate the power of the approach, we shall first describe in detail three application examples. The following paragraphs will then cover more details on

[1] Another, slightly different approach is pursued by Degen and Rarey [23] in their program FlexNovo; the authors employ fragment spaces generated in a chemically sensible way as input to a FlexX-based docking machinery.

the algorithmic side, before we summarize and conclude.

De novo ligands as strongly interacting fragments plus linkers: applications of the idea

At this point, we would like to mention that the examples have been set up to yield proofs of concept estimates for the ability of the method. As mentioned above, after having found generic linkers, Leach et al. continued by transforming the generic compounds into real linkers. This step will be omitted here; yet, it could certainly be conducted in full analogy to what has been described in the paper by Leach and co-workers.

Mimicking the original I: docking a combinatorial library into dihydrofolate reductase (DHFR)

The first example is related to the original work of Leach et al.: given a fragment library (a combinatorial library in our case) and a target (DHFR) for which binding pockets are known, retrieve active compounds. We shall achieve this by docking under pharmacophore constraints. We used FlexX with default settings and read in a binding site description through a receptor description file (RDF) with the following modifications compared to the standard and defaults:

- The active site was cut out such as to include all atoms within a distance of 7 Å of every reference ligand atom
- Water molecules number 403 and 405 were considered in the computations

The receptor-based pharmacophore definition (cp. Fig. 2) was set up to include three amino acid residues. At least one of the Arg 57 N-bonded hydrogens had to form an interaction, and at least one of the Asp 27 carboxylic oxygens had to act as an H-acceptor, and at least one amide oxygen of either Ile 5 or Ile 94 had to form an interaction with an H-donor, i.e., act as an acceptor.

The combinatorial library has been assembled as a collection of generic aliphatic linkers (following Leach's approach) plus groups "known to bind in specific (sub-)pockets" including variations, e.g., protomers etc. Guided by the rationale to virtually cleave methotrexate (MTX) at the pterin system (see Fig. 3), at both sides of the phenyl ring, and at one of the carboxyl groups results in five R-groups, two of which

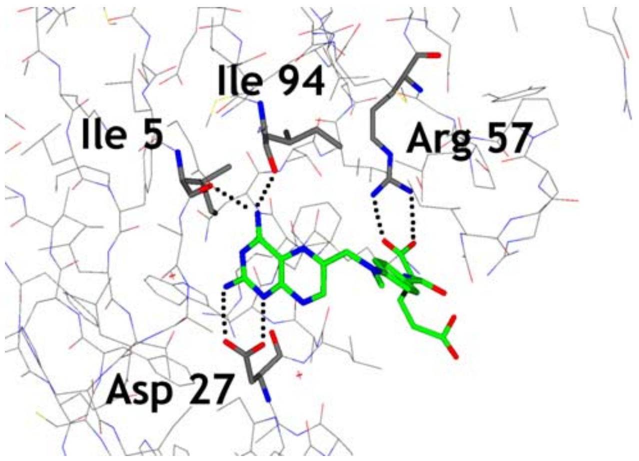


Fig. 2 Receptor-based pharmacophore definition for the DHFR experiment, exemplification of the important interactions to Methotrexate. One of each interaction pair (*dotted lines*) needed to be fulfilled

consist of the same set of aliphatic linker groups. The remaining three correspond to a pterin ring in various protonated and de-protonated forms, a phenyl group (*ortho*, *meta*, and *para*-substituted), and a neutral and deprotonated, delocalized carboxylic group. In addition, the latter group was augmented by various protomers of tetrazole, including N and C-substitution. All in all, if enumerated, there are $10 \times 53 \times 3 \times 53 \times 13 = 1{,}095{,}510$ virtual compounds in the library. Ligands, or fragments respectively, were no longer initialized with the FlexX internal routines but instead left at their respective protonation states etc. (This has been achieved applying the `SELINIT` command in FlexX.) Otherwise the differentiation between the various protomers would have been lost again. 3D-geometries were obtained using Corina from Molecular Networks [25, 26]. For an overview of the fragments, please refer to Fig. 3.

Results

From a total of 132 instances and 1,095,510 virtual compounds docking under pharmacophore constraints led to 694,644 fully assembled molecules, the remaining compounds were not docked. It should be emphasized that this can in principle have two reasons: the first is that the pharmacophore was not fulfilled. The second is due to the incremental build-up procedure being connected to its "*k*-greediness": whenever the next incremental fragment is to be added, FlexX keeps the *k* best scored partial solutions and discards the remainder. With a total of 20,604 min compute time on 20 nodes, the average docking time per ligand amounts to 1.13 s. Because this appeared rather long to us, we investigated

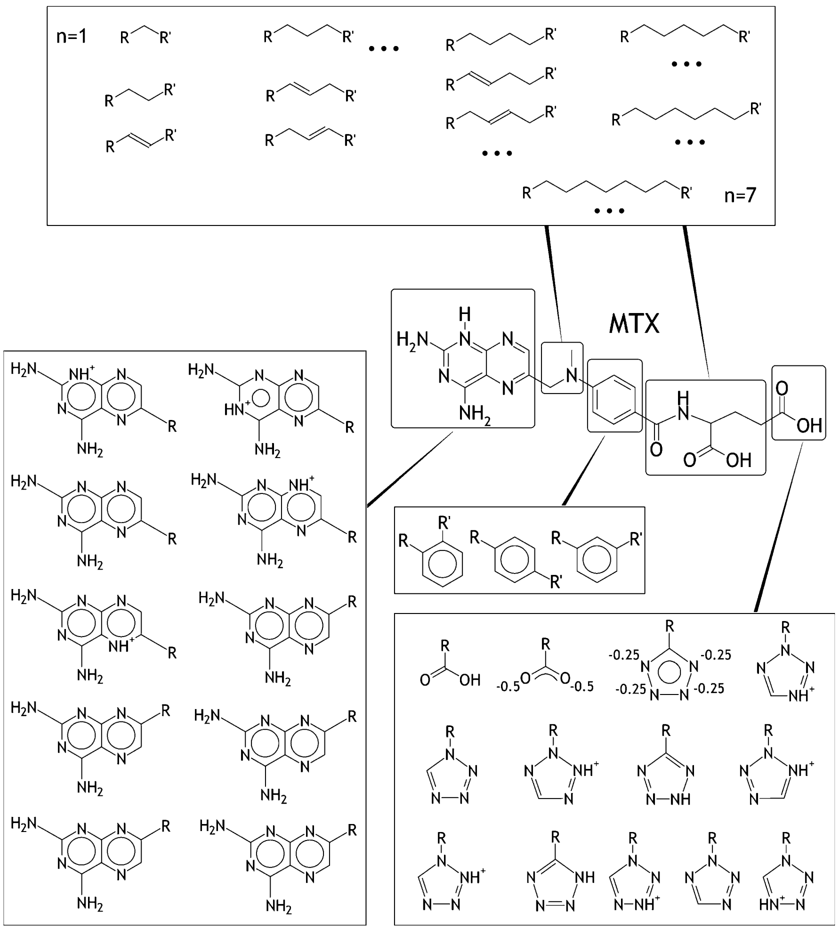


Fig. 3 The combinatorial library of variations of known binding fragments and aliphatic linker groups which extend from *n* = 1 to *n* = 7 with *n* being the number of atoms between two R-group instances. All possible combinations of double bond patterns have been created resulting in 53 aliphatic linker groups

this a little further, and noticed that it occurred with 8,457 placements before the third fragment to be connected that the algorithm did not find any new solutions. This corresponds to 109,941 discardings. For the last R-group, number 4, this occurred 290,925 times, i.e., approximately three times as much. Thus, the decision that the solutions needed to be discarded by either the pharmacophore or the fact that the fragments could not be placed anymore is partly due to a ''late'' decision. With respect to the ranking, a very close analog to MTX which could be constructed from the virtual library occurred on rank 1 (cp. Fig. 4).

Mimicking the original II: mining known DHFR binders in a larger combinatorial library

Encouraged by the results of the first proof of concept, we wanted to construct a more meaningful library and mine actual known binders. Therefore, we introduced 16 known binders from the literature and investigated the effects of the docking results as a function of the build-up sequence. Protein, active compounds, and the ligands of the decoy set have been initialized in the same way as described above. This time, the building blocks of the combinatorial library have been compiled from the Available Chemicals Directory, ACD [27]. It consists of 775 primary and secondary amines for R-group 1 (R1), and 1,750 primary and secondary amines including amino acids for R2. This results in 1,356,250 molecules in the virtual library. The core scaffold to which the instances of R1 and R2 are attached is a benzoyl group. The attachment points are at the carbonyl group and at the *para* position of the phenyl ring. From the set of R-group members, one can construct 16 known binders (cp. Fig. 5). The respective compounds were taken from the literature [28–34]. The pharmacophore has been defined as above.

Quality assessment and procedure refinement

From former experience (cp. below) we know that the build-up order is critical for the quality of the results and that a small, weakly interacting core fragment may be problematic. In this situation it often helps to apply the EXTENDCORE functionality in FlexX, i.e., to extend a small core fragment by instances of one of the

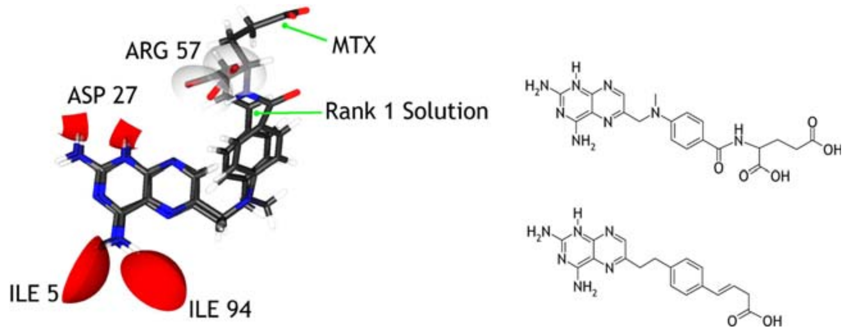


Fig. 4 Methotrexate (MTX) crystal structure and rank 1 docking solution (left). The docking solution corresponds to a very close analog of MTX. The amino acids involved in the definition of the pharmacophore are labeled in the 3D view. (Right: 2D representations of the two molecules.) The compound yielded is the closest analog in the library to MTX from our viewpoint: the pterin ring system is connected to the phenyl ring with an ethylene moiety (substituting the original methylamine function which is not available in the linker pool; further the amide function (not in the linker pool) in MTX is substituted by the (non-rotatable) doubly bonded system. Finally, the propionic acid residue was not present in the pool of linkers, so it could not have been found either

R-groups—although this leads to a longer overall computation time, because this approach violates the combinatorial nature to some extent. Thus, we decided to evaluate the effects of the different build-up orders and the extended core fragment on a shrunk subset of the full combinatorial library first. A three level quality assessment has been undertaken:

1. We have taken a random 150 × 150 (= 22,500) member subset of the 1.3 Mio library to assess (a) the quality of correlation between sequential and combinatorial dockings, and (b) to obtain the best setup of "extended core" combinations (cp. above). This step did *not* involve the pharmacophore constraints.
2. We have applied the pharmacophore constraints as described above to the subset of step 1. Here we evaluated the quality of enrichment and the variation of extended core combinations as above.
3. In the last step, we have applied the derived combination of the extended core and the phar-

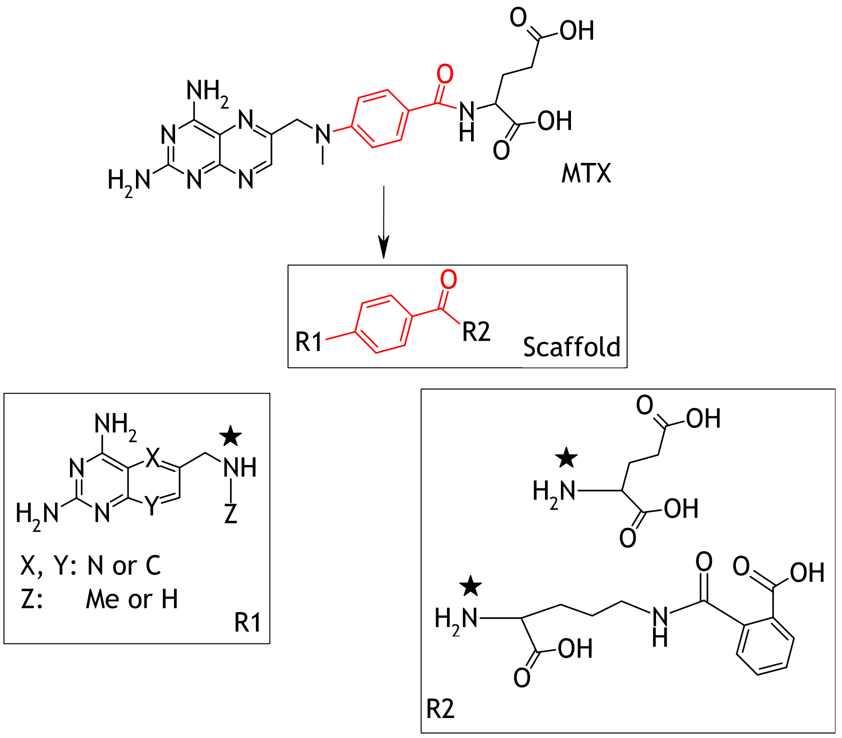


Fig. 5 Connection of the 16 actives mined as R-groups in the 1.3 million member DHFR combinatorial library. As exemplified with Methotrexate (MTX), the benzoyl group sketched in red is the scaffold to which R-groups R1 and R2 are attached. The attachment points on the side of the R-groups are marked with an asterisk. MTX itself can be re-constructed from the combinatorial library and is one member of the group of the 16 actives

macophore constraints to the entire combinatorial library. We then re-assessed the enrichment.

Step 1 delivered an enrichment as displayed in Fig. 6. The correlation (cp. below) of sequential and combinatorial docking results can be obtained from Table 1. At 0.7% of the screened library (i.e., after having considered the best docked 158 compounds), all settings lie above an enrichment which would have been obtained when picking compounds randomly. There are several conclusions that can be drawn from the results so far:

- With combinatorial docking, an enrichment comparable to sequential docking can be achieved in this case.
- The best enrichment overall is obtained with the sequence (R1–C)–R2: 4 actives are retrieved in the first 20 compounds. (at 0.09% of the database; enrichment factor of approx. 256; hit rate of 20%)
- Starting with R1 seems to be more promising; it outperforms the enrichment curve for the sequential scenario.
- The correlation between sequential and combinatorial docking scatters around 0.8 with an outlier at 0.67.
- The speed-up achieved ranges from a factor 2 (R2–C)–R1 to 12 (C–R1–R2); extended core base placement is costly (which is understandable because the base placement is algorithmically much more advanced, and the fragment to be placed is larger compared to the naked core).

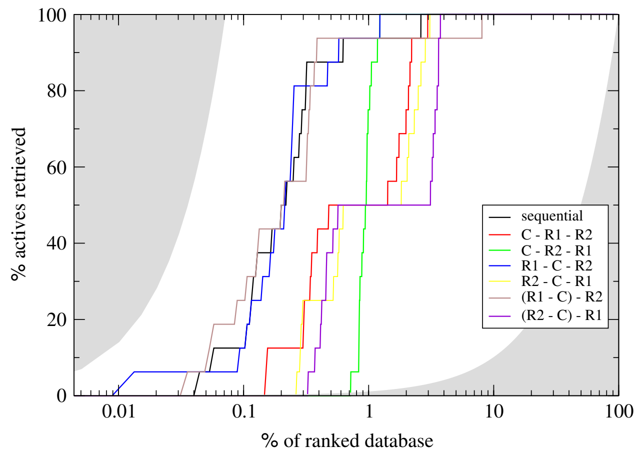


Fig. 6 Logarithmic plot of the enrichment of a 22,500 member subset of the full library. No pharmacophore constraints applied. Assembling more than one fragment to form the base fragment (FlexX^{C}'s EXTENDCORE functionality) is denoted by bracketing. The lower edge of the upper gray area denotes the situation of the top 16 compounds all being active. This would be the optimum reachable in theory; what lies above cannot be. The upper edge of the lower gray area denotes the random result. What lies below would have to be treated as worse than a random picking procedure of test compounds

Step 2: The impact on speed has been added up by both the combinatorial approach and the pharmacophore constraints. The quickest run from the optimization phase exhibits a speed-up of approx. 50 as can be drawn from Table 2. This time, we have omitted the correlation information because it is obvious that correlations can only be computed for the intersection of compounds. Seeing the numbers, the intersection is too small for correlation coefficients to be meaningful.

Also, switching on the pharmacophore constraints for the 22.5K member subset has a pronounced effect on the quality of enrichment as displayed in Fig. 7. The conclusions we draw from this graph are:

- It is confirmed that the EXTENDCORE functionality should be used
- The pharmacophore roughly discards 95% of the compounds irrespective of the remaining settings we altered
- One curve (C–R1–R2) almost touches the optimum: there is 1 hit in the first 3 compounds (at 0.015% of the library, R1–C–R2 and C–R1–R2, resp.); further on the graph there are 4 hits in the first 6 compounds (at 0.025% of the library, C–R1–R2).

Let us look in more detail at the evidently best compromise for the later test on the full combinatorial library (cp. below), namely the combination (R1–C)–R2: Above all, one should keep in mind that, given the mentioned pharmacophore in combination with the setting chosen, one compound could not be retrieved during the optimization phase. In a real scenario, one would possibly weaken the pharmacophore a little so that the known actives are not all ruled out (or at least as weak as needed to get hold of all important scaffolds, i.e., to maintain the diversity of the hits). In the context of this publication though, this would mean an unacceptable imposing of a priori knowledge. For project work on the other hand, it can well be useful to have a stronger pharmacophore definition which usually leads to better enrichments. Usually modelers are interested in the chemistry of the binders, and the respective chemistry may already be reflected in one of the actives found on much higher ranks.

Step 3: After having looked at a significantly smaller subset of the full library, we shall now discuss the final

Table 1 Docking a 22,500 member sub-library of the DHFR experiment *without* pharmacophore application

Order	Avg. time per ligand (s)	Correlation coefficient
Sequential	40.5	(1.00)
C–R1–R2	3.2	0.72
C–R2–R1	8.3	0.81
R1–C–R2	6.4	0.77
R2–C–R1	10.0	0.67
(R1–C)–R2	12.0	0.82
(R2–C)–R1	17.4	0.79

Overview of the correlation between sequential and combinatorial docking as a function of different build-up sequences and EXTENDCORE usage

Table 2 Docking a random 22,500 member sub-library of the DHFR experiment *with* pharmacophore application

Order	Avg. time per ligand (s)	Docked solutions
Sequential	231.6	1,596
C–R1–R2	1.1	750
C–R2–R1	27.5	968
R1–C–R2	2.3	479
R2–C–R1	13.3	322
(R1–C)–R2	4.6	1,177
(R2–C)–R1	32.5	634

Overview of the efficiency of sequential and combinatorial docking as a function of different build-up sequences and EXTENDCORE usage

stage of the analysis of this example, the results for the full set with 1.3 million virtual combinatorial compounds and the best compromise as distilled and discussed above. The enrichment for this large screening is displayed in Fig. 8. (A sequential analog could not be computed for reasons of time, cp. below.) We chose to display two versions, the linear and a logarithmic scale for this figure; because on the one hand, the linear plot is what most scientists will be familiar with, and it allows to obtain a quick visual impression of the quality of an enrichment computation. On the other hand, the important parts of an enrichment occur in the very early regime of a screened database fraction. Therefore we have inset a precise view at this interesting regime in form of an inset. There are several facts we can draw from the graphics:

- The first hit occurs on rank 14.
- There are 4 actives in the top 50 compounds (enrichment of 25% at 0.0035% of the library). This corresponds to an enrichment factor of approx. 7,000.
- There are 9 actives in the top 1,000 compounds.
- The pharmacophore rejects approx. 97.5% of the library by automatic pre- and in situ filtering. The number of truly docked compounds therefore amount to approximately 30,000.
- The speed-up compared to sequential docking can be computed to be approx. 30.
- All but one active could be retrieved. Sometimes, missing actives is due to a pharmacophore which has been defined to restrictive. In our case, though, this cannot be, because the functional groups are the same throughout the set of active compounds. We presume that the reason lies in the greediness of FlexX's search algorithm in combination with a

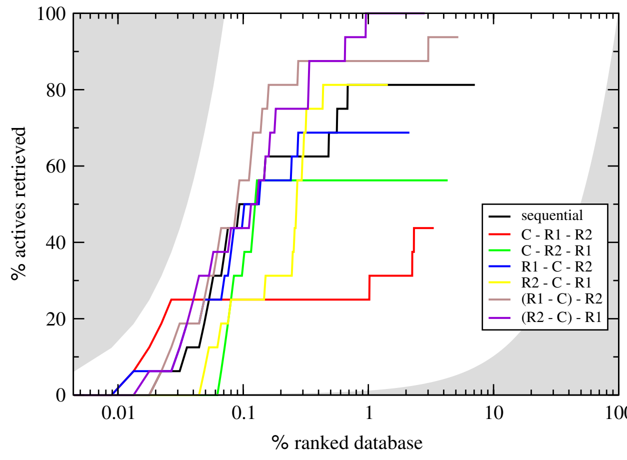


Fig. 7 Enrichment after docking a random 22,500 member subset of the full combinatorial library. A receptor-based pharmacophore definition has been applied as described in the text. Other annotation and explanations as above

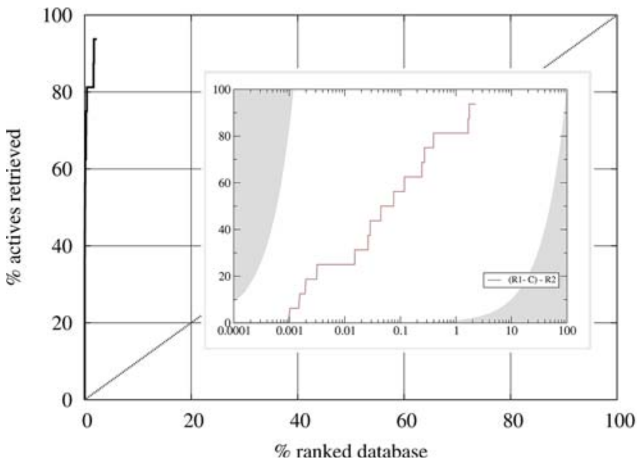


Fig. 8 Linear enrichment plot obtained for the full 1.35 million compound library and build-up sequence (R1–C)–R2 docked with pharmacophore constraints. (Inset: Logarithmic display.) At 0.0025% of the screened database, the enrichment factor amounts to approximately 7,000. All other annotation and explanations as above

limited number of poses taken along the search tree. In other words: if one does not succeed in combining fragments to (partial) poses that are promising enough during the incremental build-up, one risks to not find a final pose for a compound. For one single active compound, this seems to be the case here. A possible way out could be to increase the associated parameters, e.g., the number of partial solutions allowed for every step.

Remarkably, the best scored active indeed exhibits an activity (K_i = 0.35 pM) which is approximately tenfold compared to the one of MTX (K_i = 5.2 pM) which is found as the latest active in the screening experiment.

The real compute time on 60 nodes was 2 days. Computed sequentially it would have run for 60 days. Single node computations would have run 120 days (combinatorially), or almost 10 years (sequentially).

From 22 million compounds back to Gleevec in minutes

This third example has been chosen to illustrate how one could embed Leach's fragment-linker approach in one docking step starting from a huge combinatorial library. One possible test for such an algorithm is to decompose a compound, build-up a combinatorial library from fragments similar to the created ones, and to let the algorithm re-assemble the original compound given receptor-based pharmacophore constraints on a high rank in a realistic computational setup.

Chemistry details and docking setup

Protein kinases represent one of the major target classes in current pharmaceutical research. More than 100 X-ray structures of protein kinases in complex with small-molecule inhibitors have been deposited in the PDB to date, most of which feature the conserved initial Asp–Phe–Gly (DFG) motif of the kinase activation loop in an inward-bound orientation ("DFG-in"). In 2000, Schindler et al. [35] discovered the first instance of the activation loop flipped outward ("DFG-out") by solving the X-ray structure of Novartis' anticancer drug Gleevec in complex with c-Abl kinase. Until now, a few other examples of this inactive state have been reported (Table 3). The conformations of the activation loop depend on inhibitor structures rather than on the protein itself, i.e., a given kinase can adopt both conformations. The hydrophobic pocket opened by a "DFG-out" conformation is highly conserved and features a distinct, well-defined

Table 3 "Gleevec-like" binders in the PDB

Kinase	Ligand	Kinase family	Gatekeeper	PDB code
c-Abl	Gleevec	Tyr	Thr	1iep/1opj
p38-MAPk	BIRB 769	Ser/Thr	Thr	1kv1/1kv2
b-RAF	BAY 43-9006	Ser/Thr	Thr	1uwh
c-KIT	Gleevec	Tyr	Thr	1t46

pharmacophore pattern. For our example study, we have chosen the Gleevec/c-Abl complex (PDB code 1iep). The protein–ligand hydrogen bonds most conserved among DFG-out complexes have been defined as pharmacophore constraints. For c-Abl, these include Asp381-NH, Glu-286-OE2, Ile360-O and the central interaction to the hinge region Met318-NH (Figs. 9, 10). All constraints were defined as "essential" in the FlexX-Pharm jargon, i.e., all solutions had to fulfill all constraints—with the exception for the Glu carboxylate oxygen lone pairs; here, it was sufficient if one of these formed an interaction. It is displayed in Fig. 10.

Further, a focused combinatorial library has been created with the mind set:

- de-compose Gleevec; (since we aim at producing a library with *generic* linkers we have replaced the methylphenyl moiety by an unsubstituted phenyl ring. As a consequence, the closest analog in the library corresponds to a Gleevec which lacks a methyl group).
- re-assemble a combinatorial library from the resulting fragments, with (a) the degree of freedom of different ring substitution patterns, (b) the degree of freedom of different protonation states, (c) a varying "linker" length and bond pattern, and (d) general variations of the building fragments.

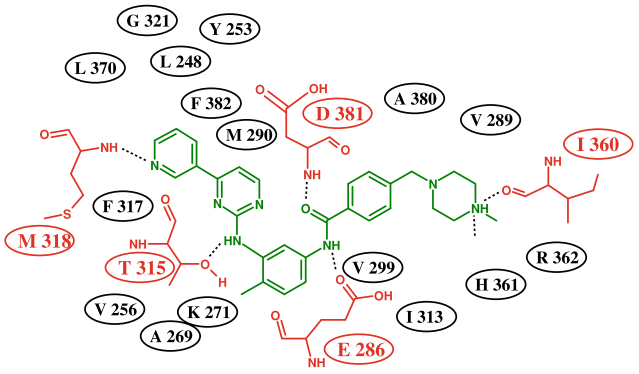


Fig. 9 The interactions with Gleevec (green) and the protein residues in PDB 1IEP. Sketched in red are directly interacting bonding partners (mostly through H-bond interactions); the black assignment signifies spatial vicinity

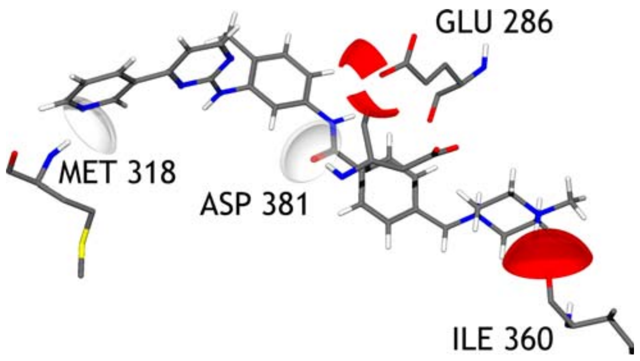


Fig. 10 The pharmacophore for PDB 1IEP. Gleevec is shown with those amino acids which have a (protein-based) pharmacophore feature assigned

For reasons of conciseness we have re-used the set of linker fragments for all fragments except the core fragment (the amide group) and the terminal groups. The resulting number of compounds is $20^4 \times 7 \times 5 \times 4 = 22{,}400{,}000$ compounds. An overview of the generated library is given in Fig. 11. This number should sensibly be boiled down by docking. We decided on having a closer look at the resulting torsional patterns, substructures and Lipinski-like filter criteria.

Torsional freedom assessment

Until now, FlexX is delivered with a parameterization file (`torsion_standard.dat`) which allows the amide bond including H and O, to adopt planar conformations plus the respective settings on a 45° grid. This is mainly to account for all possible scenarios. Yet, knowing about the role of the central amide bond, we constrained the torsional degree of freedom of amides to trans planarity with a tolerance of ±15°.

To avoid this setting being violated during assembly of an amide bond by two fragments, this constraint has been made through two overlapping SMARTS definitions: `[H]-N-^C(=O)` and `C-[C(=O)]-^[NH1]-`*C* (The "–ˆ" corresponds to a FlexX extension of the SMARTS language and matches the Sybyl mol2 definition of an amide bond.) Furthermore, after having consulted the Cambridge Structural Database (CSD) [36] using MOGUL [37], similarly, secondary phenylamines, Ph-C(=O)-residues, *N*-aryl-pyrimidines, and *N*-arylamides (between N and the Phenyl-C) have been constrained to planarity ±15° at their exocyclic carbonyl and/or nitrogen atoms—which would in fact not be the case if default MIMUMBA torsional rules would have been applied. The specification has been made in a torsion*.dat file to which the master configuration file, config.dat, referred. The respective SMARTS patterns are given in the supplementary material.

In situ Lipinski-like filtering

We aimed at a realistic scenario and at reducing the number of compounds to be probable for surviving the entire docking steps. Therefore, a newly written filter facility which reads SMARTS codes and which can work on a number of simple filter criteria such as the molecular mass has been applied. Since this implementation does not yet compute the log*P* values, we have mimicked a Lipinski Rule-of-Five like filter by requesting that

- the molecular weight be below 500
- the number of NH or OH be below or equal to 5 (mimicking donors)
- the number of N or O be below or equal to 10 (mimicking acceptors)

Because the library setup allows an emerging of unwanted aza-compounds, we have excluded these on-the-fly during the docking once formed, i.e., if a partially built-up ligand violates a constraint it is skipped and FlexX continues with the next ligand or fragment (in the combinatorial case), respectively. Besides the aza-filter, at least one violation suffices to discard compounds.

Build-up sequence-related effects

In our case, the core group was represented by two fragments in the center of the largest R-group collection (S and L2, depicted at the bottom of Fig. 11). We used again the extended core approach with two R-groups as the core, because

- we know that small cores are problematic.
- the additional fragment causes a steric hindrance that prevents us from generating too many initial placements to which no further fragments could be attached.
- additional interaction partners in the attached fragment may reach further interaction partners on the protein side and lead to better starting geometries that will be scored favorably.

We started building up the ligand in the center of the compounds for two reasons:

1. We expect the two opposing interactions in the center of the pocket to be a good starting point for the core placement.

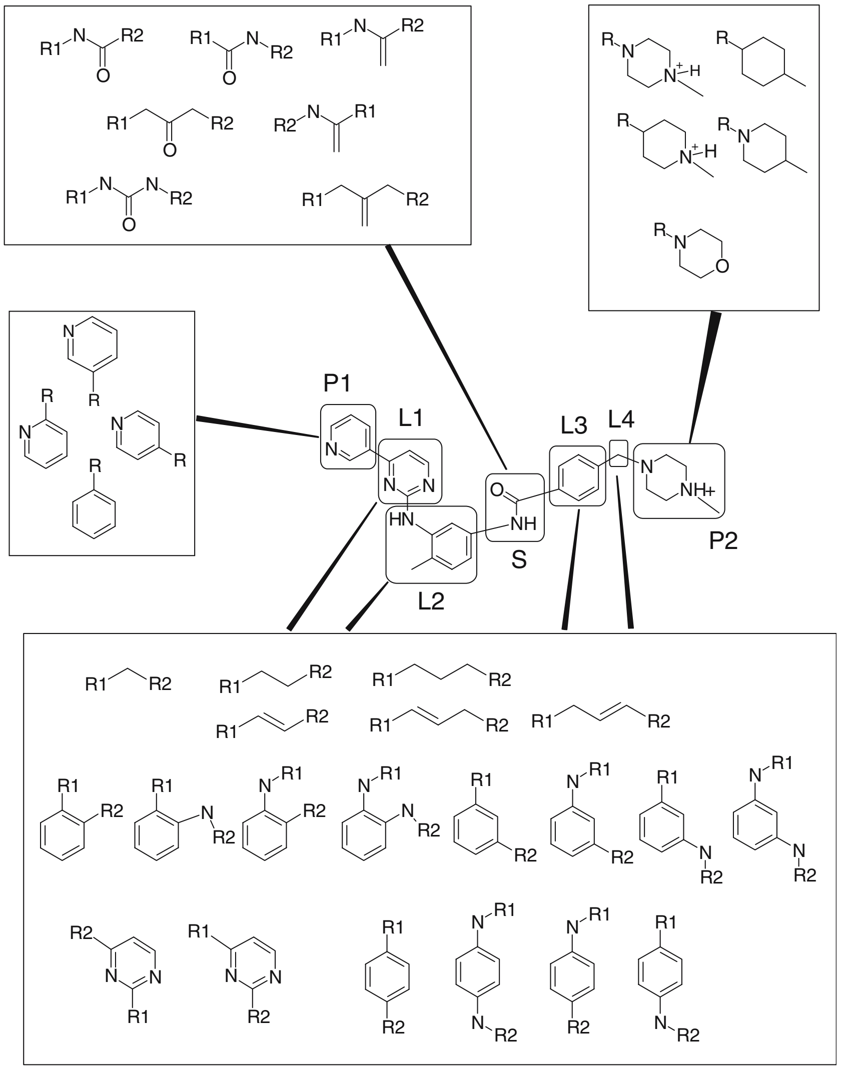


Fig. 11 The combinatorial library generation from Gleevec. Notation of the respective R-groups has been printed in bold above or below the respective Gleevec building blocks. The central amide "scaffold" has been denoted S. The other R-groups are denoted P1 (terminal, on the left), L1, L2, L3, L4, and P2 (terminal, on the right). We have used R1 and R2 if there is more than one attachment point from a fragment. Please mind that the methyl group in L2 has been left out in the building blocks of the combinatorial library we generated

2. We know that the quality of the predicted placements get worse with the number of consecutively attached R-groups. If we start from the center only 2 or 3 fragments have to be attached into both directions instead of 5 from left to right or vice versa.

Re-ranking

To further make the analysis more realistic, especially to minimize the known issues related with over-scaling large ligands [38–42], we have normalized the poses after docking by dividing the FlexX-score by the total solvent accessible surface associated with the respective conformation. This has been accomplished for the docked ligands within a few seconds with another FlexX module (FlexX-Screen) which shall be described elsewhere.

Results

Running this experiment and applying the build-up sequence (L2-S)-L1-P1-L3-L4-P2 (cp. Fig. 11) was completed in 3 hr18 min (198 min). In terms of the enumerated library, this corresponds to a fictitious docking time of 0.001 s per compound and node. In fact, only 70,898 resulting compounds fulfilled the pharmacophore constraints, corresponding to 0.16 s per actually docked compound on 20 nodes, or 3.2 s on a single node for docking including all time required for the protein preparation, I/O, and, as mentioned, discarding non-matching solutions.

The main reason for the fast docking and the rather small number of results is that FlexX can cut whole branches in the search tree very early. Table 4 gives an overview how many branches were cut at which stage and how many solutions are skipped this way. More than 6.7 Mio solutions are skipped because 24 of the 140 extended cores could not be placed (3.8 Mio) or joined (2.9 Mio) due to the filter rules. During the build-up of the ligands another 1 Mio dockings were aborted due to the filter (0.2 Mio) rules and because no valid placements were found (0.8 Mio). The consequence is that, overall, 3.7 Mio and 11.9 Mio final solutions, respectively, were skipped. This impressively documents the potential speed-up that results from the combination of (pharmacophore) filters and combinatorial docking.

The closest analog to Gleevec in the library itself could be found on rank 141 (0.0005% of the full library). It is the compound depicted in red in Fig. 12. The best ranked compound, i.e., the rank 1 solution pose is sketched in Fig. 13. One recognizes the high similarity to Gleevec which is also reflected in a more quantitative analysis:

Analysis of the best-scored compounds from docking. Plotting another descriptor, for example a non-3D similarity value such as the FTrees similarity [43, 44] versus the docking rank is illustrative (Fig. 14). Despite the fact that the closest possible analog to Gleevec is not on rank 1, the average line tells us that there is a compound which is very similar to a known binder on docking rank 1. It exhibits an FTrees similarity to Gleevec of 0.93 (cp. Fig. 13). Between docking rank 1 and rank 141 there are several structures very similar to the known binder, Gleevec. We can see by both the average line (it could not be that high otherwise), and the single point peaks that more than ten compounds exhibit similarities larger than 0.95. Since the average FTrees similarity curve drops with the docking run, we may assume that the docking setup is able to ''pull'' tentative binders, such as the compound most similar to Gleevec and others to high ranks. In this sense, the scoring does work as a rank-ordering machinery.

As another way to check the similarity of the set of best-scored docking solutions to Gleevec, we performed an R-group-based frequency analysis of the 1,000 top-ranked hits. For each R-group, its number of occurrence was counted and sorted by rank. In Fig. 15 the top-ranking fragments as well as the ranks of the

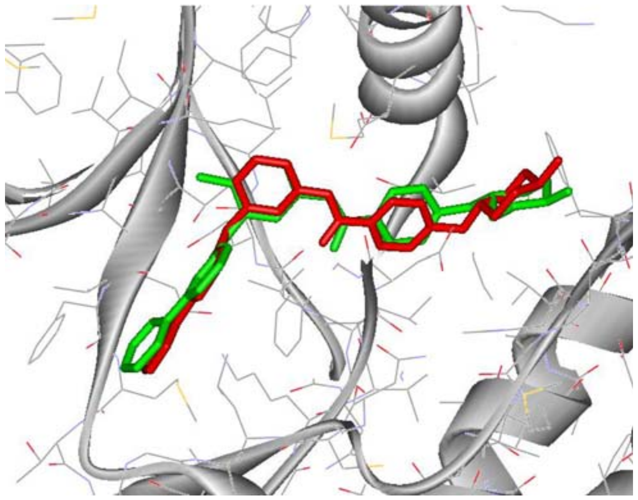


Fig. 12 Docking pose of the closest analog to Gleevec within the 22.4 million member combinatorial library. The Gleevec crystal structure is depicted in green for comparison

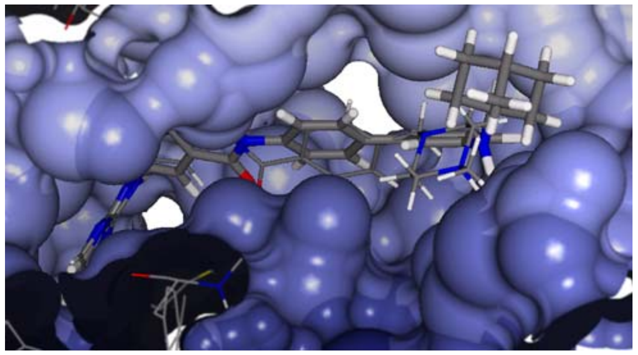


Fig. 13 Docking pose of the rank 1 solution and the Gleevec crystal structure. The Gleevec structure is depicted in thinner stick representation

Table 4 Abortion of combinatorial docking: the table shows how many docking runs were aborted at what stage

R-group	No. of instances	Filtered	No placemt.	Factor	Filtered	No placemt.
(L2-S)	20 × 7	18	24	160,000	2,880,000	3,840,000
L1	20	198	330	8,000	1,584,000	2,640,000
P1	4	0	1,173	2,000	0	2,346,000
L3	20	17,550	51,824	100	1,755,000	5,182,400
L4	20	43,197	233,952	5	215,985	1,169,760
P2	5	165,826	550,131	1	165,826	550,131
Sum	96	266,789	837,434		6,600,811	15,728,291

The R-group names and their number of instances are given in columns 1 and 2. The build-up order is from top to bottom. Columns 3 and 4 show the actual number of dockings that were aborted due to the filter rules, or because no placements were found. Column 5 contains the factor to calculate how many solutions are skipped this way, and finally, columns 6 and 7 show the resulting number of final solutions that are skipped as a consequence

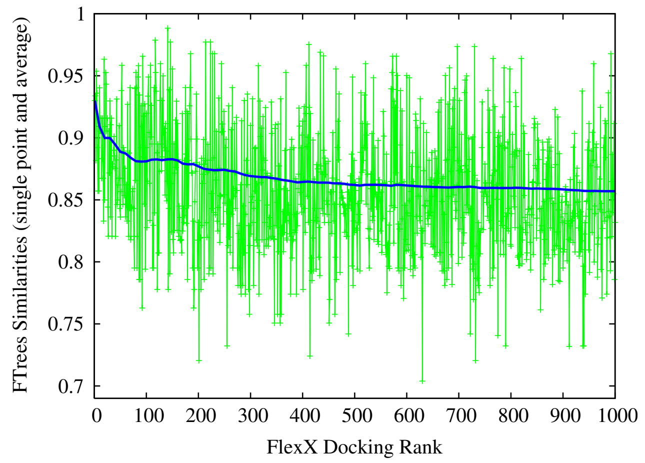


Fig. 14 Cutout of the 22.4 million compound docking result: single point (connected crosses, green) and average (thick line, blue) FTrees similarities for the best 1,000 (according to FlexX docking score) virtual compounds created. The average corresponds to the mean of all similarities from rank 1 to the respective rank on the *x*-axis. The closest analog to Gleevec in the 22.4 million compound combinatorial library is retrieved on rank 141

original Gleevec fragments are listed. In four of the seven instances (P1, L2, S, and L3), the respective Gleevec fragment in fact occurs most frequently in the top 1,000 solutions. In two other instances (L1 and P2) it appears on rank two, with very close analogs at rank 1. Only L4 shows a big discrepancy compared to the respective Gleevec fragment—the methylene group connecting the phenyl and the terminal piperazine ring—ranking only at position 17. A closer look at the docking modes of the top-ranked solutions reveals that there is a fairly large hydrophobic pocket in this region of c-Abl which can accommodate a six-membered aromatic ring well. Therefore, it is not surprising that FlexX favors such a ligand moiety over the methylene group which forms much weaker hydrophobic interactions.

Methods: docking, scoring, and the pharmacophore filtering engine

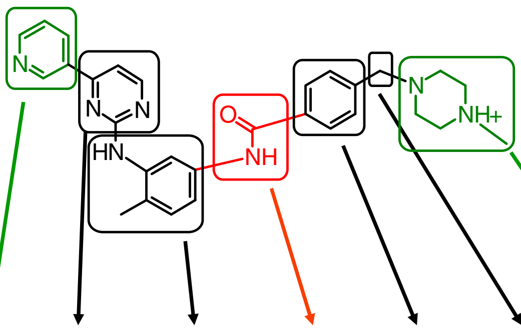
The base module: FlexX

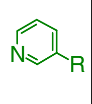
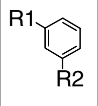
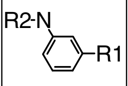
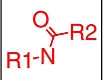
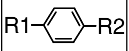
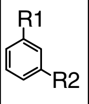
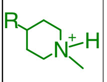
One of the key issues in docking is to treat the compound to be docked (commonly spoken of as "the ligand") as a flexible molecule. That means, rotatable bonds should not be considered as rigid. This can be achieved in two ways: by enumeration of all (or reasonable) conformers prior to docking, or on-the-fly [45, 46].

FlexX [47–50] uses the second way and de-composes the ligands at rotatable bonds before docking. During the build-up process within the active site, torsions are sampled according to certain rules (grid-based or MIMUMBA-derived), and a certain amount of solutions is taken along to the next step of re-building the ligand. Those (incremental) placements that score best are taken further. Finally, the result is the sum of all solutions that survived the entire re-building process.

The first step of all placement steps is a special step. It is referred to as the "base placement." For the placement of the first fragment (the "base fragment"), triangles formed of "interaction centers" on ligand and within the active site are searched for mutual matches. This step is usually the most time-consuming step. To speed-up the base placement, triangles, once generated, are stored in a hash table. Commonly, generating

Fig. 15 Overview of the fragment-wise occurrence counts in comparison to the Gleevec structure. The respective Gleevec fragments appear on high ranks except the L4 analog which is on rank 17. This is explainable, refer to text. For fragment L2, there is a non-methylated fragment in the combinatorial library only



	P 1	L 1	L 2	S	L 3	L 4	P 2
Rank of Gleevec fragment	1	2	1	1	1	17	2
Top-ranked fragment							

the list of solutions is expressed in terms of navigating through a solution tree. Keeping this in mind will help in understanding what follows further below. FlexX's standard scoring is based on the LUDI approach [51] by Boehm et al., yet, in principle any available scoring scheme can be applied by the user. The version we used for this work is FlexX 2.1 in a pre-release version. Except where noted otherwise, all computations have been run on the BioSolveIT 20 node compute cluster equipped with AMD Athlon MP 2000+ processors, 2GB RAM, and Linux SuSE 9.1 as the operating system.

Combinatorial docking: FlexX^{C}

FlexX^{C} [52] is the combinatorial extension to FlexX: due to the fact that the ligands are de-composed and re-built during the docking process, it lies at hand that one can substitute a certain fragment by its combinatorial substitute on-the-fly. In other words, one can re-use the fragments that have already been placed and continue with the docking of a combinatorial substitute at this very step. Figure 16 explains this pictorially.

It becomes obvious that in general this procedure should be much faster compared to sequential docking, i.e., the full enumeration of all possible ligands and their respective docking: re-using is faster than generating a solution from the beginning. What is not so obvious is that sequential and combinatorial docking may not lead to the same placements (i.e., scores) for given compounds. Let us compare the pathways of sequential and combinatorial docking: as an example, we assume that a ligand has five fragments split at and separated by rotatable bonds:

```
A-B-C-D-E
```

The sequential run is free to start building up the ligand in the active site with any of the fragments. In fact FlexX starts with up to six different base fragments in parallel. This choice is made on the grounds of a very simplistic scoring pattern, which basically only counts the number of hydrogen donors and acceptors. The selected base fragments undergo a comparably computationally expensive base placement procedure. So, in this example FlexX could start with all fragments in parallel. The combinatorial approach, however, has to start with a base fragment that is part of the core instance. Let us assume that our ligand is part of a combinatorial library, where C is the core fragment, [A-B] is in R1, and [D-E] are in R2. Square brackets denote what belongs to the R-groups:

```
[A–B]–C–[D–E]
  R1–C–R2
```

In this case, FlexX has to start with C as base fragment and cannot use the other ones. Thus, FlexX will gen-

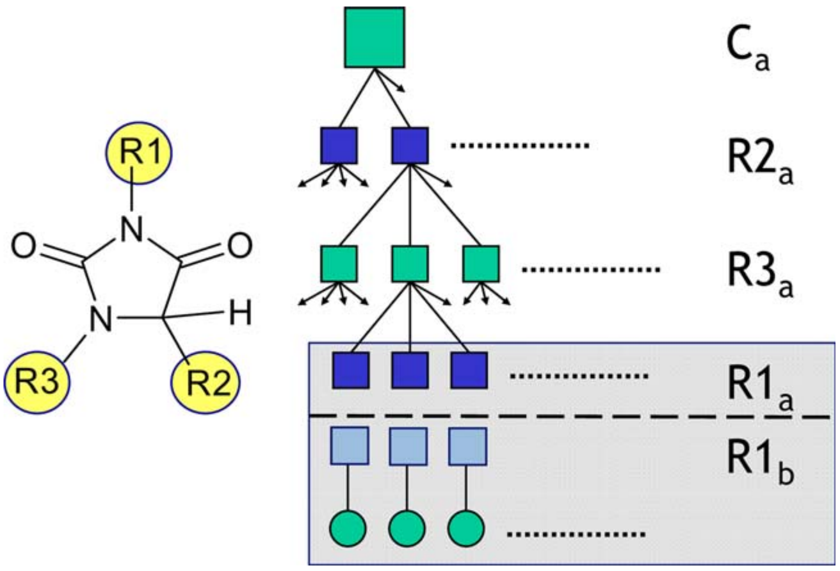


Fig. 16 Schematic sketch of the FlexX^{C} approach for combinatorial docking. A combinatorial library is created from a core fragment with R-groups (R1–R3, on the left). R-groups (and in principle also cores) may have several instances, denoted by indices *a*, *b* etc. In our example, the docking is started with instance *a* of the core, then the instance *a* of R2 is attached etc. Once R3$_{a}$ has been placed, the already placed fragments are re-used for the placement of all instances of R1 and subsequent steps (gray area). This drastically accelerates the docking at the expense that the degree of freedom of the build-up sequence is pruned compared to sequential docking, see text

erate more and different placements for C and the other starting points are not available. So, FlexX navigates through the search space in a different way resulting in overall different poses. By no means the core fragment C is guaranteed to be the best choice as a base fragment. In fact it may even happen that FlexX finds no placements for C or no base placements that can be further extended with additional fragments. In this case the combinatorial approach does not find any solution at all.

In summary, there is the possibility of multiple base fragments, and FlexX will pick the ''best possible'' choice. In contrast, FlexX$^{\mathbf{C}}$ is forced to start with what is declared to be the so-called ''core'' group. Thus, we do not expect a perfect correlation between sequential and combinatorial docking. Experience has shown in fact, that the correlation can be quite bad in some cases:

It turned out that the main reason for bad correlation scenarios lies in the occurrence of small core instances with rather unspecific interaction properties; for example, a phenyl ring which does not exhibit directed interactions (in the LUDI/FlexX sense of interactions). In such cases, the outcome of solutions obviously strongly depends on the initial phase of the docking, and a difference ''created'' at the start carries through up to the end of the docking. To overcome this, the possibility to start with a core and an additional instance during the base placement phase has been implemented. We have termed this feature ''EXTENDCORE'' and denote it in the following by brackets; ''(C–R1)'' then means: ''the base placement is carried out with the core and an instance of R-group 1 as the base fragment.'' Coming back to our example this would read:

$$\begin{array}{c}([\mathrm{A{-}B}])-\mathrm{C}-[\mathrm{D{-}E}] \\ (\mathrm{R1}-C)-\mathrm{R2}\end{array}$$

In this situation FlexX can at least use fragments A, B, and C in parallel for the core placement. However, this strategy requires more compute time because now not only one core fragment has to be placed, but for each instance of the adjoint R-group a separate base placement has to be performed. But the better placements and score justify this extra effort in almost all cases. The functionality is now available in FlexX$^{\mathbf{C}}$.

Do combinatorial docking scores correlate with sequential results?

In order to assess the effects associated with different build-up orders and the extended core approach, we have docked eight combinatorial libraries into two different targets. Targets and libraries were provided by Bayer CropScience. The first target which we cannot disclose here, is known to be a tough task for FlexX because of its very hydrophobic binding site. The other one (4dfr) is known to be well suited for docking with FlexX.

Table 5 shows the correlation coefficients c between sequential and combinatorial docking runs for different build-up orders. The correlation coefficients have been computed according to the definition

$$c(X,Y) := \frac{E(XY) - E(X)E(Y)}{\sqrt{V(X)V(Y)}} \qquad (1)$$

with E being an expectation value and V the variance of the respective variables X and Y. The sequence of combinatorial docking is denoted as A–B–C; as mentioned, extended cores are signified by bracketing, so, for example, ''(R1–C)–R2'' means that we started the base placement with R1 instances attached to the core, and then docked the R2 fragments. Correlation coefficients better (larger) than 0.6 are highlighted in light gray and better than 0.75 highlighted in dark gray. Obviously, all correlations depend on the build-up order. Analyzing the numbers, it becomes clear that it is not advisable to start with the core alone. Instead, one should employ another R-group in this case. Yet, the best results can be achieved if one starts with an extended core. The overall correlation is better for target 2 (4dfr), which is also better suited for docking. For this target even some good correlations are found when starting with the core fragment alone. To test the hypothesis of libraries that can be well docked to also result in good correlations between sequential and combinatorial docking we docked a 100×100 subset of the library of Methotrexate analogs (cp. Fig. 5) into 4dfr (Table 5, the row labeled ''Metho''), and in fact with this library we already obtain a good correlation when starting with the core alone. If we start with an extended core using R-group R2 we end up with a correlation coefficient of nearly 0.9.

The average runtimes are given in the last row. Starting with the core is—as expected—the fastest setup, starting with an R-group is slightly lower, and the extended core takes the most time. But even the extended core calculation is on average more than ten times faster than the sequential run.

For reasons of completeness, the last column analyzes the correlation between the two EXTENDCORE lines of experiments. Both experiments are constrained to start with a combined core plus R-group. It can be presumed that the probability of coming to the same result

Table 5. Correlation coefficients between sequential and combinatorial docking and the two EXTENDCORE computation lines (last column). Correlation coefficients above 0.6 in light grey, above 0.75 in dark grey. The best correlations are obtained with the EXTENDCORE functionality. The timing of 33.1 s in the lower left refers to the sequential calculation.

Target 1		Start with Core		Start with R Group		Start Core + R-Group		Corr. of (R1-C)-R2
Library	Size	C-R1-R2	C-R2-R1	R1-C-R2	R2-C-R1	(R1-C)-R2	(R2-C)-R1	vs. (R2-C)-R2
Lib 1	50×100	0.50	0.42	0.66	0.58	0.68	0.70	0.40
Lib 2	30×30	0.29	0.29	0.56	0.55	0.62	0.50	0.25
Lib 3	30×30	0.46	0.32	0.33	0.44	0.59	0.62	0.43
Lib 4	30×30	0.54	0.49	0.46	0.50	0.41	0.56	0.13
Lib 5	30×30	0.15	0.24	0.59	0.48	0.61	0.46	0.30
Lib 6	30×30	0.53	0.56	0.47	0.51	0.49	0.48	0.26
Lib 7	84×100	0.30	0.34	0.68	0.70	0.71	0.54	0.37
Lib 8	100×84	0.01	0.26	0.72	0.56	0.76	0.53	0.36

Target 2 (4dfr)		Start with Core		Start with R Group		Start Core + R-Group		Corr. of (R1-C)-R2
Library	Size	C-R1-R2	C-R2-R1	R1-C-R2	R2-C-R1	(R1-C)-R2	(R2-C)-R1	vs. (R2-C)-R2
Lib 1	50×100	0.56	0.54	0.73	0.73	0.77	0.78	0.64
Lib 2	30×30	0.74	0.67	0.78	0.52	0.78	0.70	0.56
Lib 3	30×30	0.70	0.68	0.76	0.42	0.77	0.68	0.55
Lib 4	30×30	0.26	-0.01	0.59	0.68	0.60	0.61	0.21
Lib 5	30×30	0.40	0.46	0.66	0.60	0.68	0.76	0.46
Lib 6	30×30	0.19	0.28	0.65	0.72	0.69	0.75	0.38
Lib 7	84×100	0.55	0.50	0.71	0.74	0.77	0.71	0.57
Lib 8	100×84	0.52	0.26	0.72	0.62	0.80	0.63	0.49
Metho	100×100	0.69	0.61	0.65	0.76	0.74	0.87	0.64
Avg. time	33.1 s	0.9 s	1.1 s	1.5 s	1.6 s	2.6 s	2.5 s	–

will decrease when one compares the two: an unconstrained sequential docking has the freedom to start with any base fragment, whereas the EXTENDCORE lines of experiments are constrained to start with subsets of base fragments which will usually be different from each other (the core is very small in our examples). Coming back to our example from above, the sequential runs can start with any of [A] to [E], whereas the (R1−C) run would have to start off with [A] to [C] (and (R2−C) with [C] to [E], respectively). So, the EXTENDCORE lines of experiments have each some identical starting points with sequential docking, but the sequential docking runs may find some additional (and better scoring) solutions. But, apart from the core, the two experiments cannot have any common base placements at all. Since the correlation between these two experiments and sequential docking is not ideal in neither of the cases, it is no surprise that the correlation between these two experiments themselves is even worse. In other words, the deviations for the two EXTENDCORE calculations add up in a way. This is indeed confirmed by correlation factors down to 0.13 in the last column. On the other hand, if both correlations with the sequential runs are promising, the probability to end up with a better inter-correlation of the EXTENDCORE calculations is higher. As a confirmation, the example of the two highest values (libraries "Metho" and "Lib 1," both 0.64) can serve: they have to be seen in the context of the high correlations of 0.88/0.78, and 0.74/0.87, respectively.

In summary, the correlation between sequential and combinatorial docking score depends on the target and the build-up order, and it is therefore a good idea to evaluate the best build-up order prior to screening a large combinatorial library.

Pharmacophore constraints during docking: FlexX-Pharm

Traditionally, [53] pharmacophores are defined as a ligand-based property, namely as the "spatial arrangement of functional ligand groups required for bonding to the protein." In contrast to this, in FlexX-Pharm, Hindle et al. [54, 55] have implemented pharmacophore definition possibilities from the receptor side. In general, the modeler will have a certain knowledge of the target protein at hand. For example, in kinases, it is well known where the ATP-binding site, the so-called hinge region is. Modelers can make use of such knowledge during docking by imposing constraints to the docking procedure. FlexX-Pharm offers the following types of constraints:

- Interaction constraints: the modeler defines an interaction such as a hydrogen acceptor interaction and assigns it to a certain amino acid atom or group.
- Spatial constraints: a sphere is defined into which certain ligand atoms have to fall if the solution shall be valid. New to the current release (2.x) is that `SMARTS` patterns [56] can be applied to fuzzily define the ligand atoms that can match.

The interactions *can* (optional constraints) or *must* (essential constraints) be met. Optionally, the combinations of all these is possible with the help of so-called logical expressions, that is, the user assigns `A, B, C, D...` to the constraints and then uses logical expression syntax to define what is allowed and what is not. An example for this would be `(A or B) and (C or D)`

User-defined constraints are checked in three steps prior *and* during the actual docking:

1. Countergroup check: this checks whether the ligand (or its remaining fragments) at all carry interaction groups which comply with the given set of constraints. For example, if we request a hydrogen acceptor interaction for a donor functionality at the receptor, but there is none in the ligand, there is no need to start or continue with the docking.
2. Distance check: this check controls whether constraints at given distances from the current starting point can be reached with the given fragments. For example, if there are only two bonds in the last fragment, yet, the constraint to be matched is 10 Å away, we can be sure this constraint will not be met. The distance check operates on a maximum distance look-up table (MDLT) in which estimates for maximum reachable distances are stored. The principle of filling the MDLT is based on the addition of increments: for atoms separated by 2 and 3 bonds, there is no degree of freedom, because there are no torsional angles for such fragments. The respective distances can be computed directly and are stored in the MDLT. For four bonds, the value of a maximum distance at 180° is stored. All subsequent distances are computed by addition of the respective distances.
3. Directed tweak check: this check has first been authored by [57]. It proceeds by calculating the derivative of distances with respect to torsion angles of rotatable bonds. With the respective result, FlexX-Pharm checks whether it is possible to meet a distance constraint given the discrete set of torsion angles allowed.

Whenever at least one of the checks fails, the docking of the respective compound is aborted or proceeds with either another part of the solution branch which did not fail in the test. In virtual screening the docking continues with the next molecule.

The characteristics of the procedure are two-fold: usually the docking is much faster because of rejected solutions. Second, the solutions by design fulfill the constraints, and this is why generally the solution set is of higher quality and made up of novel solutions compared to unconstrained docking. The latter holds because by filtering away parts of the solution tree, the algorithm is ‘‘forced’’ to fill up the set of solutions with other, i.e., novel placements.

Combining pharmacophores and combinatorial input algorithmically

The key feature of FlexX-Pharm is the ability to look forward and to see in advance whether the remaining fragments are still able to fulfill the constraints or not. In order to do this the numbers of different counter groups of the remaining fragments are summed up and their maximum distance look-up (MDLT) tables are combined. So, the major challenge in combining the combinatorial approach with receptor-based pharmacophores is to adapt this looking forward approach to sets of various linkers and R-groups.

The basic idea to solve this problem is to introduce another maximum estimate. We simply take the sum of the interaction possibilities within the FlexX interaction model as an upper limit for the number of counter groups of a particular type within a set of R-groups. The maximum distance through a set of linkers is estimated as the maximum of all maximum distances through the particular linkers in the set. As a first step local MDLTs for all particular instances of all R-groups of a combinatorial library are computed. For each R-group a master MDLT is created that basically contains the distances from the anchor atom to all atoms, to all potential interactions, and especially to the outgoing linker atom. Two MDLTs for different R-groups or particular instances of R-groups can easily be combined, because the MDLT overlap with each other: the anchor atom of fragment A is part of the MDLT of fragment B and vice versa. During the incremental build-up of a ligand of a combinatorial library there are three different parts of the ligand that can be handled differently:

1. the part of the ligand already placed
2. the particular instance of an R-group that shall be added next
3. the remaining fragments of R-groups which have not yet been placed.

The number of counter groups, the coordinates, and distances are fixed for the part of the ligand already placed. So, there is no need to use an MDLT for this estimate. For the particular R-group instance a the local MDLT can be used, which can be combined with the master MDLTs for the remaining R-groups in order to estimate whether the particular R-group can still fulfill the constraints with respect to the whole sets of remaining R-groups.

If this is not the case the build-up of the particular ligand and of the rest of the library can be aborted immediately. The earlier a build-up can be aborted the larger the part of the search tree that can be cut. So, combining the combinatorial approach with receptor-based pharmacophores speeds up the docking in two ways. First it prevents—as in the sequential approach—from building up ligands completely that do not meet the constraints, but secondly and even more importantly, it prevents us from exploring the entire combinatorial library space. Thus, this approach is much faster than combinatorial docking with post filtering even though some extra time is needed for the additional preparation and the checks before adding the next fragment.

Conclusions and perspectives

We have reported on an on-the-fly protocol for very fast compound docking which combines both pharmacophore filtering and combinatorial fragment search. Application fields one could think of are fragment-based de novo design, gap-bridging between multiple known pockets, scanning combinatorial libraries for matching fragments, etc.

The method is based on an integration of FlexX-Pharm and FlexX^{C}. It could be demonstrated that enrichments could be efficiently and very significantly improved—synonymous to a reduced number of false positives. The speed-up by the combinatorial approach (in the order of 10) could be multiplied by 5 through application of pharmacophore constraints in our study. The correlation between sequential and combinatorial docking can be ameliorated by the ‘‘extended core’’ approach, a procedure which involves base placements with a core *and* an R-group instance. The reported functionality will be present as module compatibility in FlexX Release 2.1 modules.

Acknowledgments We gratefully appreciate comments and hints from Hugo Kubinyi. Thanks also to Robert Klein (Bayer CropScience in Frankfurt, Germany) who helped us during our correlation investigations. We extend our gratitude to Birte Seebeck at University of Hamburg for compiling combinatorial libraries. This work has been partly funded by Schering AG, Berlin, Germany.

References

1. Kitchen DB, Decornez H, Furr JR, Bajorath J (2004) Nat Rev Drug Discov 3:935
2. Shoichet BK, McGovern SL, Wei B, Irwin JJ (2002) Curr Opin Chem Biol 6(4):439
3. Tame JR (1999) J Comput-Aided Mol Des 13(2):99
4. Rarey M, Degen J, Reulecke I (2006) In: Lengauer T (ed) Bioinformatics – from genomes to therapies. Wiley-VCH, Weinheim, Germany
5. Rarey M (2001) In: Lengauer T (ed) Bioinformatics – from genomes to drugs, vol. I. Wiley-VCH, Heidelberg, p 315
6. Abagyan R, Totrov M (2001) Curr Opin Chem Biol 5:375
7. Schulz-Gasch T, Stahl M (2004) Drug Discovery Today: Technologies
8. Wang R, Lu Y, Wang S (2003) J Med Chem 46:2287
9. Wang R, Lu Y, Fang X, Wang S (2004) J Chem Inf Comput Sci 44:2114
10. Marsden PM, Puvanendrampillai D, Mitchell JBO, Glen RC (2004) Org Biomol Chem 47:3032
11. Gohlke H, Hendlich M, Klebe G (2000) J Mol Biol 295:337
12. Stahl M, Rarey M (2001) J Med Chem 44:1035
13. Dobson CM (2004) Nature 432:824
14. Lipinski C, Hopkins A (2004) Nature 432:855
15. Nikitin S, Zaitseva N, Demina O, Solovieva V, Mazin E, Mikhalve S, Smolov M, Rubinov A, Vlasov P, Lepikhin D, Khachko D, Fokin V, Queen C, Zosimov V (2005) J Comput-Aided Mol Des 19:47
16. Jahnke W, Flörsheimer A, Blommers MJJ, Paris CG, Heim J, Nalin CM, Perez LB (2003) Curr Topics Med Chem 3:69
17. Banerjee AL, Swanson M, Roy BC, Jia X, Haldar MK, Mallik S, Srivastava DK (2004) J Am Chem Soc 126:10875
18. Lyne PD, Kenny PW, Cosgrove DA, Deng C, Zabludoff S, Wendoloski JJ, Ashwell S (2004) J Med Chem 47(8):1962
19. Krier M, de Araújo-Júnior JX, Schmitt M, Duranton J, Justiano-Basaran H, Lugnier C, Bourguignon J-J, Rognan D (2005) J Med Chem 48(11):3816
20. Mestres J, Fradera X (2004) Curr Top Med Chem 4(7):687
21. Güner OF (ed) (2000) Pharmacophore perception, development and use in drug design. International University Line, La Jolla, California, USA
22. Claussen H, Gastreich M, Apelt V, Greene J, Hindle SA, Lemmen C (2004) Curr Drug Discov Technol 49–60
23. Degen J, Rarey M (2006) Chem Med Chem 854–868 (FlexNovo can be obtained through BioSolveIT GmbH, St. Augustin, Germany, http://www.biosolveit.de)
24. Leach AR, Bryce RA, Robinson AJ (2000) J Mol Graph Model 18(4–5):358
25. Corina v 3.10. Molecular Networks GmbH, http://www.mol-net.de
26. Sadowski J, Gasteiger J (1993) Chem Rev 93:2567
27. Available Chemicals Directory (ACD). MDL Information Systems Inc., http://www.mdli.com/products/experiment/available_chem_dir
28. Kisliuk RL (2003) Curr Pharm Des 9(31):2615
29. Rosowsky A (1999) Curr Med Chem 6(4):329

30. Wright JE, Yurasek GK, Chen YN, Rosowsky A (2003) Biochem Pharmacol 65(9):1427
31. Vaidya CM, Wright JE, Rosowsky A (2002) J Med Chem 45(8):1690
32. Wright JE, Vaidya CM, Chen Y, Rosowsky A (2000) Biochem Pharmacol 60(1):41
33. Rosowsky A, Wright JE, Vaidya CM, Forsch RA (2000) Pharmacol Ther 85(3):191
34. Rosowsky A, Wright JE, Vaidya CM, Bader H, Forsch RA, Mota CE, Pardo J, Chen CS, Chen YN (1998) J Med Chem 41(26):5310
35. Schindler T, Bornmann W, Pellicena P, Miller WT, Clarkson B, Kuriyan J (2000) Science 289:1938
36. Allen FH (2002) Acta Cryst B 58:380
37. Bruno IJ, Cole JC, Kessler M, Luo J, Motherwell WDS, Purkis LH, Taylor R, Smith BR, Cooper RI, Harris SE, Orpen AG (2004) J Chem Inf Comput Sci 44(6):2133
38. Huang N, Nagarsekar A, Xia GJ, Hayashi J, MacKerell AD Jr (2004) J Med Chem 47:3502
39. Krämer O, Hazemann I, Podjarny AD, Klebe G (2004) Proteins Struct Funct Genet 55:814
40. Pegg SC, Haresco JJ, Kuntz ID (2001) J Comput-Aided Mol Des 15:911
41. Muegge I, Martin YC, Hajduk PJ, Fesik SW (1999) J Med Chem 42:2498
42. Pan Y, Huang N, Cho S, MacKerell ADJ (2003) J Chem Inf Comput Sci 43:267
43. Rarey M, Dixon JS (1998) J Comput-Aided Mol Des 12:471
44. Rarey M, Stahl M (2001) J Comput-Aided Mol Des 15:497
45. Kramer B, Rarey M, Lengauer T (1999) Proteins 37:228
46. Rarey M (2001) In: Lengauer T (ed) Bioinformatics, vol. 1. Wiley-VCH, Weinheim, p 315
47. Rarey M, Kramer B, Lengauer T, Klebe G (1996) J Mol Biol 261(3):470
48. Rarey M, Kramer B, Lengauer T (1997) J Comput-Aided Mol Des 11:369
49. Rarey M, Kramer B, Lengauer T (1999) Bioinformatics 15:243
50. BioSolveIT GmbH http://www.biosolveit.de/FlexX. Address: An der Ziegelei 75, St. Augustin, Germany. BioSolveIT obtained exclusive rights to further develop and market the Flex* suite of programs
51. Böhm H-J (1992) J Comput-Aided Mol Des 6:61
52. Rarey M, Lengauer T (2000) Persp Drug Dis Des 20:63
53. Böhm H-J, Klebe G, Kubinyi H (1996) Wirkstoffdesign. Spektrum Akademischer Verlag GmbH, Heidelberg
54. Hindle SA, Rarey M, Buning C, Lengauer T (2002) J Comput-Aided Mol Des 16:129
55. Hindle SA, Stahl M, Rarey M (2003) In: Proceedings of Euro-QSAR 2002. Blackwell Publishing Ltd, Oxford UK
56. SMARTS is a molecular substructure language invented by Daylight Chemical Information Systems, Inc. http://www.daylight.com/learn
57. Hurst T (1994) J Chem Inf Comput Sci 34:190