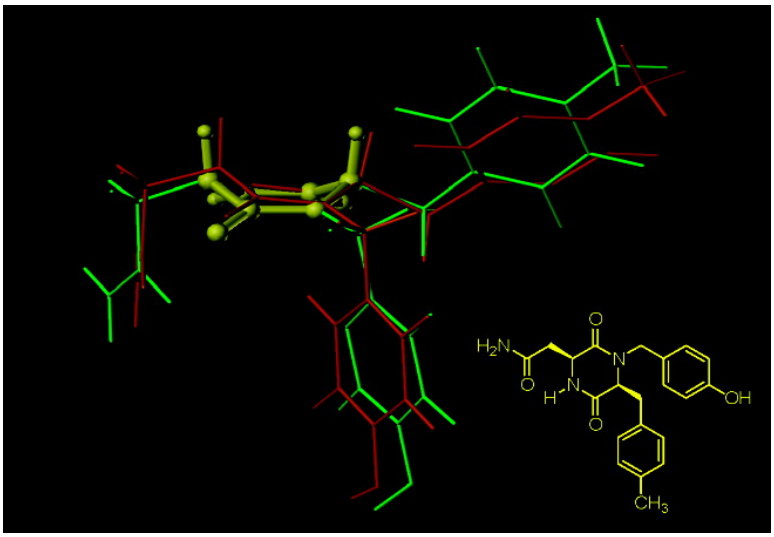
JOURNAL OF combinatorial CHEMISTRY

Article

OptiDock: Virtual HTS of Combinatorial Libraries by Efficient Sampling of Binding Modes in Product Space

Dennis G. Sprous, David R. Lowis, Joseph M. Leonard, Trevor Heritage, Steven N. Burkett, David S. Baker, and Robert D. Clark

J. Comb. Chem., **2004**, 6 (4), 530-539• DOI: 10.1021/cc034068x • Publication Date (Web): 15 April 2004



More About This Article

Additional resources and features associated with this article are available within the HTML version:

- Supporting Information
- Links to the 1 articles that cite this article, as of the time of this article download
- Access to high resolution figures
- Links to articles and content related to this article
- Copyright permission to reproduce figures and/or text from this article

View the Full Text HTML

Journal of Combinatorial Chemistry is published by the American Chemical Society.
1155 Sixteenth Street N.W., Washington, DC 20036

OptiDock: Virtual HTS of Combinatorial Libraries by Efficient Sampling of Binding Modes in Product Space

Dennis G. Sprous,† David R. Lowis, Joseph M. Leonard,‡ Trevor Heritage, Steven N. Burkett, David S. Baker, and Robert D. Clark*

Tripos Inc., 1699 South Hanley Road, St. Louis, Missouri 63144

Received December 5, 2003

Products from combinatorial libraries generally share a common core structure that can be exploited to improve the efficiency of virtual high-throughput screening (vHTS). In general, it is more efficient to find a method that scales with the total number of reagents (Σ growth) rather with the number of products (Π growth). The OptiDock methodology described herein entails selecting a diverse but representative subset of compounds that span the structural space encompassed by the full library. These compounds are docked individually using the FlexX program (Rarey, M.; Kramer, B.; Lengauer, T.; Klebe, G. *J. Mol. Biol.* **1995**, *251*, 470−489) to define distinct docking modes in terms of reference placements for combinatorial core atoms. Thereafter, substituents in R-cores (consisting of the core structure substituted at a single variation site) are docked, keeping the core atoms fixed at the coordinates dictated by each reference placement. Interaction energies are calculated for each docked R-core with respect to the target protein, and energies for whole compounds are calculated by finding the reference core placement for which the sum of corresponding R-core energies is most negative. The use of diverse whole compounds to define binding modes is a key advantage of the protocol over other combinatorial docking programs. As a result, OptiDock returns better-scoring conformers than does serially applied FlexX. OptiDock is also better able to find a viable docked pose for each library member than are other combinatorial approaches.

Introduction

The use of in silico docking of potential ligands into protein binding sites for virtual high-throughput screening (vHTS) has become an integral part of computer-aided drug discovery and development in recent years. The docking programs currently in use can be broadly classified as "rigid body" or "flexible", depending on how they treat rotatable bonds in the ligand (and in some cases, in the protein). Examples of rigid body docking programs include DOCK[2] and FRED.[3] These programs can optimally position a specified ligand conformer in the protein cleft in 0.1−3 s and return a pseudoenergy interaction score for each pose produced. The approach requires the generation, storage, and management of large conformer libraries, which may actually require more CPU time and human intervention than does the docking run itself. It also assumes that the full range of conformations available to a ligand can be adequately represented by a relatively small sample thereof. This assumption often fails, however, in that more fully flexible docking programs generally can outperform their rigid body counterparts.[4−6] Flexible docking programs vary in their underlying protocols, but characteristically operate on a time scale of minutes per ligand. Documented approaches include simulated annealing,[7] Monte Carlo,[8,9] distance geometry,[10] genetic algorithms,[11] and incremental construction.[1,12,13]

Using vHTS to search for new lead compounds requires that large numbers of candidate ligands be passed through the docking program used, which presupposes very efficient processing. The structural redundancy characteristic of combinatorial libraries makes serial docking inherently inefficient, thereby offering an excellent opportunity to reduce CPU demand dramatically. Two programs have been developed in which critical steps in the docking procedure are modified to work with structures in such a way that the number of operations required grows in proportion to the total number of *substituents* ($M = \sum_{j=1}^{q} m_j$,where m_j is the number of substituents considered at variation site R_j, and q is the number of variation points), rather than in proportion to the number of combinatorial *products* ($N = \Pi_{j=1}^{q} m_j$). Both programs start by docking fragments rather than intact products, which can result in some significant limitations.

The CombiDOCK program introduced by Kuntz et al.[14] starts by docking a core structure devoid of substituents to define the potential docking modes subsequently considered. A range of alternative conformations are then considered for individual substituents at each variation site. A binding energy for each product is then calculated by simple summation of contributing substructure interaction energies. This is valid, since the scoring function employed lacks *intra*molecular terms: interactions between atoms in the ligand and in the protein are taken into consideration, but

* To whom correspondence should be addressed. E-mail: bclark@tripos.com.

† Current address: APT Therapeutics, 893 N. Warson Road, St. Louis, MO 63141.

‡ Current address: Abbott Bioresearch Center, Inc., 100 Research Drive, Worcester MA 01605.

10.1021/cc034068x CCC: $27.50
Published on Web 04/15/2004

interactions between different atoms in the ligand are only considered when prohibitive van der Waals bumps are found.

The FlexX^C program introduced by Rarey and Lengauer[15] takes a somewhat different approach. With the parent incremental construction program FlexX,[1] the slow step in docking is typically identifying and positioning a good initial base fragment. This is a small substructure from the parent compound that is positioned favorably within the binding site, thereby defining a starting pose from which the remainder of the molecule is subsequently built up. A particular fragment may produce several good starting poses, but the branch and prune logic of the program restricts the total number of such base placements that can be considered. When the act of defining a base placement is repeated for each member of the combinatorial library, the same base placements get "rediscovered" over and over again. FlexX^C achieves a sizable speedup over serial FlexX by allowing the user to specify that base placements are all to be drawn from the substituent list for a single variation site R* (or from the core). These are determined once and saved in memory. When the incremental construction protocol is invoked to start building a new molecule, it pulls an appropriate base placement from memory and proceeds from there. The rate-limiting step of FlexX is then carried out only some small multiple of m^* times rather than N times, so the processing time required exhibits Σ growth with increasing library size rather than the Π growth of serial FlexX.

This improvement in scaling is not without cost, however. FlexX is free to select an initial base fragment from anywhere in the molecule, but FlexX^C requires that the base fragment be selected from the substituent list for the specified variation site. Hence, results obtained from FlexX^C can differ significantly from those obtained from serial FlexX, since different starting points often produce different incremental construction trajectories. In cases in which interaction with substituents at the selected variation site dominates binding, the discrepancy will generally be small, but in other cases, it can be substantial. In particular, some substituent lists may have several members for which no substructure is suitable for use as a base fragment. Any compound bearing such a substituent at the designated base variation site will fail to dock, since there is no possible starting point. This potential shortcoming was noted in the initial FlexX^C publication.[15] In addition, compounds whose docking mode is dominated by interactions with the core or with substituents at variation sites other than R* may dock incorrectly.

Herein, we describe the improved, complementary approach embodied in OptiDock.[16] Our protocol assumes, as do the other approaches, that a relatively modest number T ($\ll N$) of core poses can adequately account for the imporatnt docking modes found for a given combinatorial library and target protein. In contrast to the other methods, however, those core poses are extracted from a set of intact products obtained by efficient sampling based on substructure that have been docked independently of one another. Moreover, it does so in such a way that increasing T progressively increases the accuracy of the results obtained as more reference poses are considered, that is, as T increases. This is accomplished by selecting a structurally diverse but representative subset of compounds from the combinatorial library, docking these compounds, and then extracting diverse and representative core placements from the poses obtained. Core structures are then decorated by adding single substituents drawn from each variation list in turn, and these truncated products (R-cores) are docked using FlexX with the core atoms fixed in place, as specified in each reference placement. Once FlexX has applied the incremental construction protocol to build and score all possible substituent R-groups across all variation points, then whole-compound energies per binding mode can be estimated by simple summation of contributing R-group scores.

The present paper describes the OptiDock protocol in depth. The stages of the OptiDock protocol are explained, and considerations of parameter settings are discussed. Results for three different combinatorial vHTS examples are presented using the method as embodied in version 6.8 of SYBYL.[16] These examples serve to illustrate how CPU usage scales in practice, how reproducible the method is, and how well docking energies correlate with experimentally determined affinities.

Methodology

Library Definition. Combinatorial libraries were input into a compact library definition format known as a CSLN, for Combinatorial SLN, an extension of SYBYL Line Notation.[17] Each CSLN consists of a core 2D structure, with variation sites indicated by Markush atoms, plus appended lists of the substituents found at each site. A very simple, four-compound library consisting of dipeptides bearing glycine or alanine at the first position and valine or aspartic acid at the second is represented by the CSLN,

```
NH2CH(Y1)C(=O)NHCH(Y2)C(=O)OH{Y1:H,CH3}
                             {Y2:CH(CH3)CH3,CH2C(=O)OH}
```

(The chirality flags that would normally be included have been omitted for simplicity's sake). Each compound in a library with q variation sites can then be represented by a vector $\mathbf{V}$, the q elements of which are indices specifying the particular substituent found at the corresponding site. For the dipeptide example given above, for example, $\mathbf{V} = [1\ 1]$ and $\mathbf{V} = [2\ 2]$ represent glycyl valine and alanyl aspartic acid, respectively.

2D Sampling of the Library. 3D structural diversity in docking mode space can be achieved by exploiting a diverse and representative subset of products determined at the 2D level using OptiSim methodology.[18−20] A starting product is chosen on the basis of a random number seed supplied by the user, then a series of subsamples of K candidate compounds are drawn at random from the CSLN. At each iteration, the candidate from the subsample that is least structurally similar to those selected in previous iterations is chosen for addition to the training set.[18−20] The process continues until the prespecified number, S, of compounds has been selected or until no more valid candidates are available. When K is small (typically in the range of 3−5), the subset obtained using this optimizable K dissimilarity (OptiSim) selection procedure is diverse but still representative of the original dataset.[18,19] Redundancy in the selection

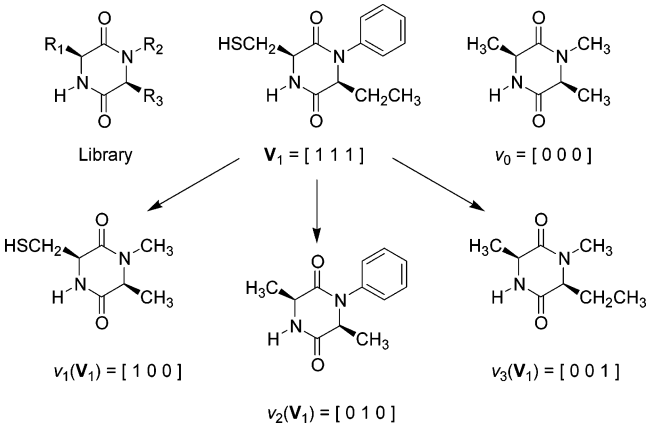


Figure 1. Relation of a compound of an order three combinatorial library to its contributing R-cores.

set is avoided by requiring that no candidate in the subsample be too similar to any compound selected during previous iterations. For the work described here, K was set to 5, and a value of 0.85 was used for the maximum allowed Tanimoto similarity[21] with respect to UNITY substructural fingerprints.[22]

Selection of Reference Core Placements. Each compound in the selected subset is docked independently using FlexX. The initial set of reference core placements is then obtained by stripping away the substituents from the best-scoring pose for each docked product, retaining coordinates for the core atoms. Though the compounds from which they are obtained are structurally diverse, the placements themselves may not be. Hence, another round of OptiSim selection is carried out in Euclidean space to identify T representative and diverse reference placements. In this case, similarity is assessed in 3D space, that is, using the root-mean-square distance (RMSD) between corresponding core atoms in pairs of placements. The initial reference placement chosen is based on a random number seed provided by the user.

A subsample size $K = 5$ was used in the experiments reported here, and the minimum RMSD allowed between reference placements was 0.25 Å unless otherwise indicated. The number of placements was set equal to the training set size, which ensured that OptiSim selection would run until exhaustion, that is, until every core in the docked training set fell close to (here, within an RMSD of 0.25 Å of) some reference core placement.

R-Core Generation and Docking. The potential contribution of each substituent at each variation site to binding is estimated for each distinct reference core placement by constructing and docking a series of R-cores (Figure 1) analogous to those described by Kuntz and co-workers.[14] As used here, an R-core consists of the structural core common to all members of the library with one substitution at a single variation point. The remaining variation sites need to be "capped" so as to minimize spurious effects from open valences, for example, on nitrogen or oxygen bridging atoms. Methyl groups work well for this purpose in FlexX.

Each of the M R-cores is then positioned in the target binding site using the core atom coordinates from each of the T reference placements. Docking then proceeds using FlexX in a constrained mode, such that the core is used as the base fragment and the atoms in it are fixed in place. The end result is then $T \times M$ piecewise binding energies, one for each R-core docked using each reference core placement. This strategy proved adequate for the studies described here, but there is no intrinsic barrier to using multiple good docking configurations for each placement of each R-core.

Figure 2 illustrates the procedure used. Products from the training set are docked independently, and reference placements for each are obtained from the corresponding core atom coordinates (Figure 2A,B). These are then used to position R-cores from which contingent substituent configurations can be determined (Figure 2C); the best-scoring side chain configuration is highlighted in green for one placement in each case.

In addition to the R-cores, OptiDock creates a structure composed of the core with a capping group added at each variation point. This structure (v_0) is passed to FlexX to determine the energy contribution from the core itself.

Estimating Binding Energies for Whole Compounds. OptiDock evaluates docking energies for multiple conformations of each compound. The maximum number of conformations per compound is effectively the number of binding modes successfully determined after elimination of 3D redundancies by OptiSim. This number may be less, however, if atoms are found to be overlapping when whole compounds are assembled and R-group conformations that were created in isolation are found to coexist at the same points in space. Each particular conformation is specifically matched with a docking mode, k. The R-cores representing the substituents on V_i can then be represented by q vectors, $v_j(V_i)$. Since the FlexX energy function lacks conformationally dependent intramolecular terms, such as intramolecular electrostatics or intramolecular Lennard−Jones potential, the energy of V_i for each placement k can be calculated by summing over the contributions from each R-group in compound V_i and the combinatorial core v_0.

$$E_k(V_i) = E_k(V_0) + \sum_{j=1}^{q} E'_k(v_j(V_i)) \qquad (1)$$

The prime symbol in the second term is intended to indicate that this is the docking energy of the R-group rather than of the R-core. The equation above was written to stress that all terms are specific for a unique docking mode k. $E'_k(v_j(V_i))$ (the energy of an isolated R-group at a specific docking mode) is easily calculated as the difference between the energy of the core ($E_k(v_0)$) and the energy of the R-core ($E_k(v_j(V_i))$) for that docking mode.

$$E'_k(v_j(V_i)) = E_k(v_j(V_i)) - E_k(v_0) \qquad (2)$$

Combining eqs 1 and 2 gives us the following:

$$E_k(V_i) = E_k(v_0) + \sum_{j=1}^{q} [E_k(v_j(V_i)) - E_k(v_0)] \qquad (3)$$

Equation 3 can be simplified to

$$E_k(V_i) = \sum_{j=1}^{q} [E_k(v_j(V_i))] - (q-1)E_k(v_0) \qquad (4)$$

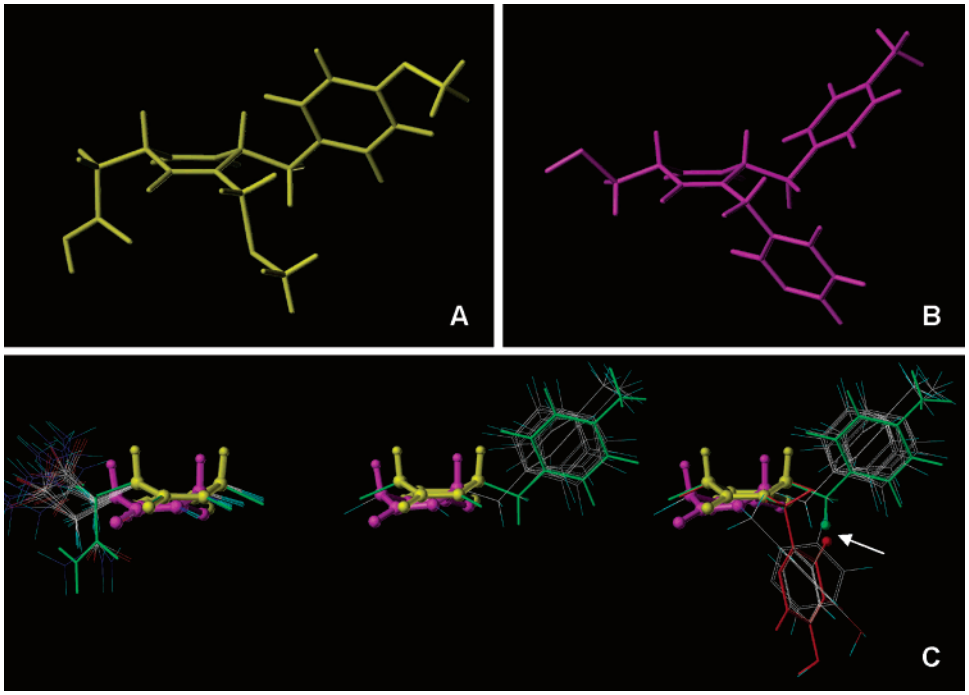


Figure 2. (A and B) Docking configurations for two products from the training set. (C) Docking configurations for one R-core at each variation site, positioned using the reference placements from the poses shown in panels A and B. The arrow highlights a region in which a steric clash exists for some substituent combinations.

Table 1. Combinatorial VHTS Systems Used Here as Examples

name	library source	library dimensions	target protein	pdb code
Sz11K	Szardenning et al.1999[23]	12 × 30 × 30	collagenase	966c[a]
K1K	Kick et al. 1997[25]	10 × 10 × 10	cathepsin D	1lyb[b]
L108	Linusson et al. 2001[27]	6 × 6 × 3	thrombin	1h8d[c]

[a] See ref 24. [b] See ref 26. [c] See ref 28.

A last correction needs to be applied to eq 4 for accurate correspondence of OptiDock energies with FlexX energies on a per-compound per-conformation basis. FlexX utilizes a term to model the loss of entropy that occurs when rotatable bonds are frozen on binding of a ligand in a protein cleft. However, the intentional specification of the core as the base placement done as part of the OptiDock protocol eliminates the calculation over all bonds in the combinatorial core, requiring that we calculate this ourselves and add it back into the term. Consideration of this last term, $S(V_0)$, leads to eq 5, which is the form embedded in the OptiDock code base.

$$E_k(V_i) = S(v_0) + \sum_{j=1}^{q}[E_k(v_j(V_i))] - (q-1)E_k(v_0) \quad (5)$$

Note that the entropy term ($S(V_0)$) bears no subscript; it is based solely on the number of rotatable bonds; 3D structure is not considered in the calculation, so docking mode is irrelevant. $S(V_0)$ is the same for each compound and each conformation of an OptiDock run and is strictly a function of combinatorial core bonds that are a constant for a given combinatorial library.

Equation 5 makes it possible to calculate FlexX scores for candidate conformations of a product before generating the coordinates for that product in that particular docking mode. The coordinates for the side chains in the contributing R-core conformations and for the core itself must exist at that specific docking mode, but not the coordinates of the corresponding conformation of the product itself. Once a particular configuration has been selected as interesting − usually meaning it represents the minimum docking energy found by OptiDock − an additional calculation is performed to check for steric clashes (e.g., in the region indicated by the arrow in Figure 2C). If a configuration fails this test, it is eliminated, and the next best conformation is tried.

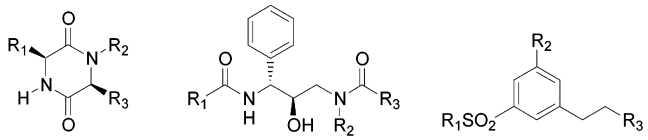


Figure 3. Structures for the combinatorial libraries used in this study.

Results and Discussion

Data Sets. References for the three data sets used herein as examples are laid out in Table 1. The corresponding structures are provided in Figure 3. All involve libraries for which the number of variation sites is $q = 3$. The smallest, the L108 data set, consists of 108 compounds (6 × 6 × 3), of which 12 have firm numeric values for affinities. The small size of this library makes it practical for exploring the

Table 2. Ensemble Properties OptiDock Energies (kJ) from Sz11K Analyses Using Different Training Set Sizes and Correlations (r^2) among Them

T^a	run time (min)	minimum energy	maximum energy	mean energy ± SD	versus $T = 20$	versus $T = 40$	versus $T = 80$
20	81	−40	−3.5	−20 ± 5.2	1.00	0.86	0.86
40	167	−39	−5	−21 ± 5.1		1.00	0.88
80	279	−39	−7.5	−21 ± 4.9			1.00

[a] Number of reference core placements used.

effects of changing various parameters on the results obtained. The second is the K1K data set (1000 compounds, 10 × 10 × 10), employed here just as it was in validating the CombiDOCK program.[14] This data set includes IC50s for seven products. The last (Sz11K) is based loosely on the work of Szardenning et al.[23] and is composed of 10 800 products (30 × 30 × 12).

All data sets are available in CSLN format from the authors.

Accuracy of OptiDock Energies. The ability of eq 5 to reproduce FlexX scores for particular docking poses was assessed using K1K and a subset of compounds from the Sz11K data set. Each CSLN was run through OptiDock to generate a population of docked compounds and associated docking energies. The poses were then embedded in a molecular spreadsheet (MSS) and evaluated using the FlexX scoring function in CSCORE.[29] Least squares linear regression was performed to quantify the correlation between the two sets of energies. The resulting equations and their statistics are given as eqs 6 and 7 for K1K and Sz11K, respectively.

$$E_{\text{OptiDock}} = 2.50 + 0.98\,E_{\text{FlexX}} \quad (r^2 = 0.997,\ F = 108 \times 10^3,\ N = 108) \quad (6)$$

$$E_{\text{OptiDock}} = -1.56 + 0.99\,E_{\text{FlexX}} \quad (r^2 = 0.998,\ F = 203 \times 10^3,\ N = 288) \quad (7)$$

The correlation between the OptiDock energy and the FlexX energy was very high in both cases. Moreover, the slopes found were essentially indistinguishable from unity.

The intercepts in eqs 6 and 7 are small, but they are nonzero. This reflects ambiguity in how FlexX calculates the entropic penalties for rotatable bonds, as well as contributions from the methyl blocking groups used to cap open valences in the R-cores. Both effects are consistent within a particular combinatorial library, however, and so do not affect rank-ordering within that library.

Speed. Docking each of the S compounds in the training set takes the same 1 to 3 min/compound typically required for FlexX. The R-core dockings are much faster, however, taking approximately 5−20 s for each. This is because finding a good base placement is the most time-consuming step in a FlexX run, and that step is bypassed in docking these constructs by specifying the base fragment and its placement. For very large libraries, carrying out the bump checks needed to qualify the poses corresponding to the lowest energies calculated using eq 5 becomes rate-limiting if scores are desired for all products, but the process is still always very much faster than serial docking.

How much faster OptiDock is than serial docking depends on the size of the combinatorial library to which it is being applied, but it depends more strongly on S (the size of the training set) and on T (the number of reference placements). In fact, one can easily make OptiDock slower than serial FlexX by making both unnecessarily large. It is more informative to consider how OptiDock's CPU time growth scales with library size and as program parameters change.

Timings for OptiDock runs on Sz11K using different numbers of reference core placements are shown in Table 2. CPU consumption is roughly proportional to the number of reference placements used: doubling the number of placements doubles the run time. Even with 80 placements, however, the time consumed by the OptiDock run is trivial compared to the 10 days required to dock the 10 800 compounds in Sz11K individually. In this case, OptiDock is 35−105 times faster than FlexX alone.

Efficiency. As noted above, combinatorial docking generally involves some tradeoff between speed and the quality of the results obtained. For the OptiDock approach, this tradeoff is largely determined by how many reference core placements are needed to adequately cover the range of docking modes found in the fully enumerated library. The number of reference placements required is expected to vary depending on the combinatorial library and the target protein of interest. This is, indeed, the case.

Table 2 presents the ensemble properties for OptiDock energies obtained for the Sz11K data set when the number of placements is varied and the correlations among them (in all of the experiments described in this section, the minimum RMSD between core placements was set to 0Å, so none of the core placements produced were eliminated as redundant during the Euclidean OptiSim selection step). It is clearly not necessary to dock all 10 800 compounds to obtain reproducible energies. The differences in ensemble properties among 20, 40, and 80 placements are negligible: though there is a slight drift of maximum energy toward lower values, the minimum and mean energies show very little change. The number of placements required for this system is less than 20, provided that those 20 are diverse and representative enough to saturate the available docking modes.

Another way to be sure that a broad enough range of docking modes is being considered is to supply several different random number seeds for the OptiSim selections in OptiDock. Each different seed leads to a somewhat different training set being selected, which in turn leads to a somewhat different set of reference core placements. The system is saturated if there is no appreciable difference between runs differing only in the random number seed used when selecting the training set.

Results for such analyses are presented in Table 3 for the K1K and Sz11K data sets for $T = 124$ reference placements.

Table 3. Correlations among OptiDock Energies for Sz11K and K1K Analyses Using 124 Reference Core Placements

	data set					
	K1K			Sz11K		
run[a]	A	B	C	A	B	C
A	1	0.83	0.80	1	0.78	0.76
B		1	0.76		1	0.81
C			1			1

Table 4. Correlation of Compound Energies among Runs for the L108 Data Set

	run[a]						
	10A	10B	10C	10D	10E	108F	serial[b]
10A	1.00	0.92	0.82	0.95	0.93	0.98	0.83
10B		1.00	0.75	0.92	0.91	0.94	0.75
10C			1.00	0.71	0.89	0.87	0.80
10D				1.00	0.86	0.92	0.78
10E					1.00	0.97	0.83
108F						1.00	0.69

[a] Except as indicated otherwise, each run name is based on the number of placements in the reference set (*T*) and by an indicator of the random number seed used. [b] Results from independent application of FlexX to each compound.

The correlations obtained for K1K with fewer reference core placements were unsatisfactory: selection of 31 and 62 reference placements led to poor correlations between successive runs, ~0.40 and 0.5−0.6. At 124 placements, the correlations are high, and the situation is stable (Table 3). The correlations are comparable, in fact, to those for Sz11K, in which the experiment described above demonstrated docking mode saturation (Table 3).

Table 4 presents the results of applying this method to the L108 system. Five runs were performed using different random number seeds but with the maximum number of ligands included in the training set for docking was held at 10 (runs 10A−10E). Differences between successive runs were trivial.

This data set is small enough that two distinct kinds of exhaustive sampling were practical. The first was the case in which all 108 compounds in the fully enumerated library were included in the reference placement set (run 108F in Table 4). These results were as similar to those for the much smaller reference placement sets as they were to each other, which indicates that 10 reference placements were adequate to cover the full range of docking modes for this system.

That docking mode space is more easily saturated for Sz11K and L108 than for K1K probably reflects two factors. First, the core structures for the former libraries are relatively rigid, whereas the core for K1K is much more flexible. As a result, reference placements must sample a range of conformations as well as variation in core rotation and translation. Second, for the Sz11K and L108 libraries, the core atoms are significant in determining the docking mode. This direct participation of the core atoms in binding reduces the number of reference placements required for coverage in those two data sets.

The central limit theorem guarantees that the information content of a sample varies with the square root of the sample size.[30] As a result, the first and second observations in a series are the most cost-effective. A second reasonable breakpoint comes at $T = [N]^{1/2}$, which was used here as a default sample size. However, the above examples have demonstrated that this default is not universally applicable and can be either inadequate (K1K case) or redundantly exhaustive and CPU-wasteful (Sz11K case). After the data is created and interpreted, the reasons for the behavior are obvious. However, this is not the case before the computation is performed. For OptiDock, *adequate* sampling can never be proven. There is always the possibility that some binding mode is missed. However, *inadequate* sampling can be easily recognized. By example, consider the case in which only the random number seed is changed in the OptiDock protocol between runs A and B. Now, the resultant energies per compound between the two runs need to show some reasonable correlation to one another ($R_{A,B}^2 \geq 0.50$). If the two runs are not well-correlated, then sampling was probably inadequate, and a larger number of diverse compounds will be needed.

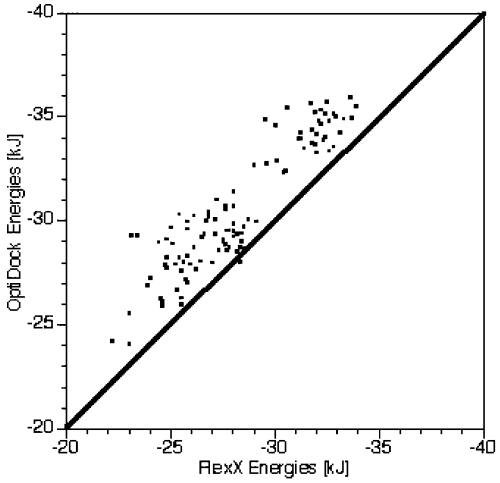


Figure 4. OptiDock energies as a function of serial FlexX energies for the L108 data set. OptiDock energy represents the best energy conformer found for that compound against the best energy conformer found by applying FlexX directly.

OptiDock involves stochastic sampling, so it is possible to use replication to determine whether binding mode sampling has been adequate. For the reasons outlined above, this sampling protocol is best done across multiple seeds to demonstrate a satisfactory degree of coverage of relevant binding modes.

Correlation with Serial Docking. The OptiDock energies for L108 are less well correlated with the results obtained from a serial FlexX run than they are with each other (Table 4). As is shown graphically in Figure 4, the OptiDock energies are almost all more negative in this case than are the corresponding energies from separate FlexX runs. Results for K1K clearly indicate that this is a result of a more thorough docking mode sampling, since the discrepancy between OptiDock and serial FlexX energies drift to progressively more negative (i.e., more favorable) values as more reference placements are used (Figure 5). (Serial FlexX was actually unable to dock about 50 compounds in this case; the corresponding data points have been omitted from Figure 5.)

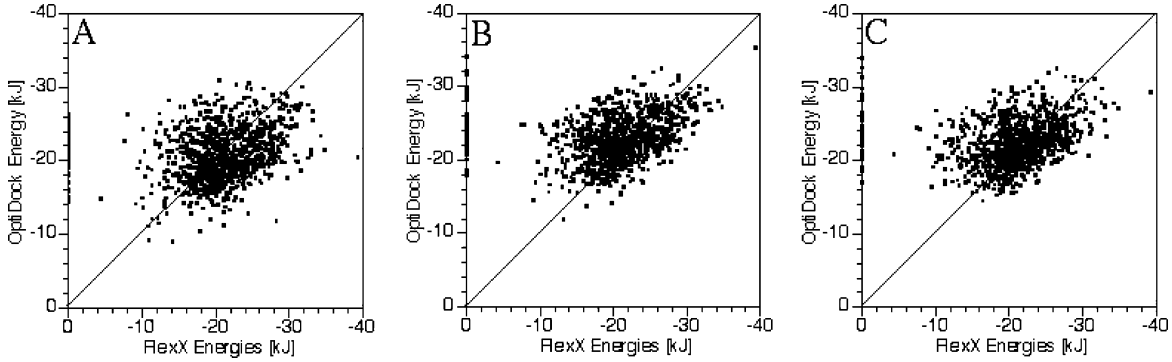


Figure 5. OptiDock energies as a function of serial FlexX energies for the K1K data set: (A) 31 reference placements, (B) 62 reference placements, and (C) 124 reference placements.

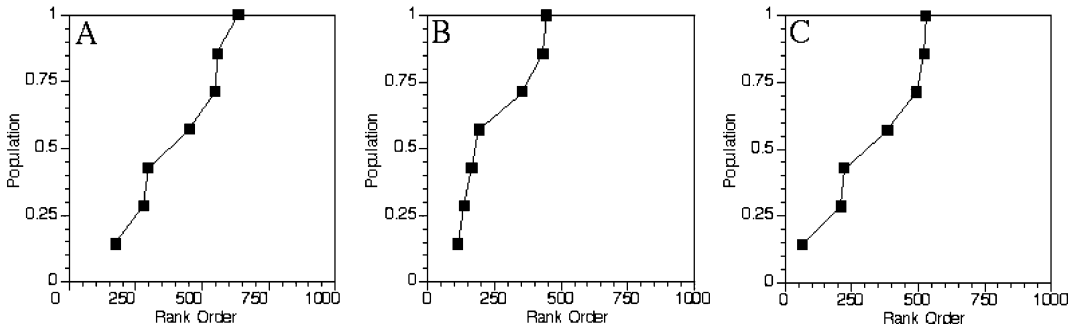


Figure 6. Enhancement plots for K1K as a function of the number of reference placements employed: (A) 31 reference poses, (B) 62 reference poses, and (C) 124 reference poses.

This effect is distinct from the systematic differences seen when the poses generated using OptiDock are rescored using FlexX (eqs 2 and 3). The OptiDock poses actually represent optima in the FlexX scoring function that FlexX itself fails to find, at least with the parameter settings used here. The discrepancy results from FlexX's failing to find globally optimal base placements initially or from eliminating them prematurely during subsequent branch-and-prune phases of the incremental construction process. This should perhaps not be surprising, since the R-cores can effectively "see" the base fragmentations and placements explored by *every* compound in the training set.

OptiDock Energy Versus Binding Affinity. In principle, one might reasonably expect that finding consistently superior docking configurations would automatically lead to better correlation with measured binding affinities. This is true in some cases but not in others.

Most vHTS validation experiments have focused on enrichment, that is, on increasing the number of true actives found among the high-scoring ligands.[14,31] Such enrichment can reduce substantially the number of compounds that need to be assayed for confirmation following a particular vHTS analysis. Since a virtual docking protocol strives to mimic a protein ligand complex, high-quality binding affinity data is valued for docking validation. Unfortunately, real-world applications are often forced to rely on affinity data from HTS screens, where error rates are typically high and extensive accurate affinity data are generally not available.

K1K is just such a data set. Structures and IC50s were reported only for the seven compounds exhibiting the greatest inhibition in the primary screen. Comparison of the primary data provided for these actives with their IC50s indicates that the HTS assay itself has a relatively high error rate. Nonetheless, OptiDock enhanced the number of known actives found consistently within the top 50% of the data set (Figure 6). Increasing the number of reference placements employed, which consistently led to better scoring OptiDock poses (Figure 5), did not yield any qualitative improvement in results. Serial FlexX did not perform nearly as well in this case, in that the enhancement found was actually negative (details not shown).

The L108 library was designed on the basis of a QSAR analysis of known thrombin inhibitors. The final combinatorial design included 108 products. Linusson et al.[27] then selected a diverse subset of 18 compounds which, given the deduced QSAR, were expected to span the full activity range of the data set. Six compounds predicted to have no activity indeed proved inactive when synthesized, whereas the other 12 compounds synthesized were active to a greater or lesser degree. Having these data available made it possible to examine the relationship between OptiDock energy and affinity directly, rather than indirectly via enrichment curves.

The correlation between the FlexX energy obtained by serial docking and measured pIC50 for this data set is quite good ($R^2 = 0.72$; Figure 7A). The plots of OptiDock energy versus pIC50 for the five independent OptiDock L108 runs cited above are qualitatively similar, but are displaced to somewhat more negative (more favorable) energies (Figure 7B). They do, however, exhibit lower correlation coefficients ($R^2 = 0.50-0.66$). That somewhat different outliers are observed in each run suggests that the small differences between runs represent random noise, a notion supported by

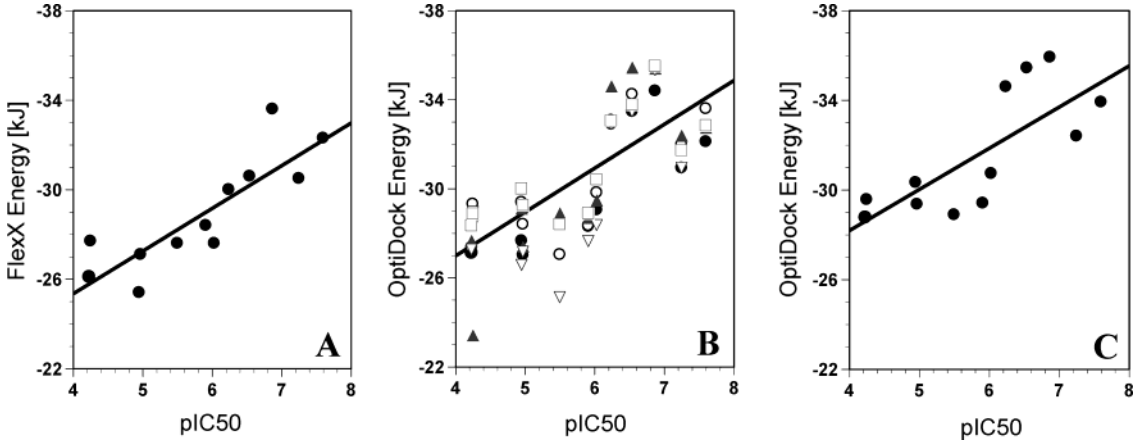


Figure 7. Relationship between pIC50 for inhibition of thrombin and docking energy for the L108 data set. (A) Serial FlexX docking ($R^2 = 0.72$). (B) Results from five OptiDock runs, each using 10 different reference docking modes each ($R^2 = 0.50$ to 0.66). Each symbol type corresponds to a different random number seed. (C) OptiDock results obtained by exhaustively sampling reference poses ($R^2 = 0.58$).

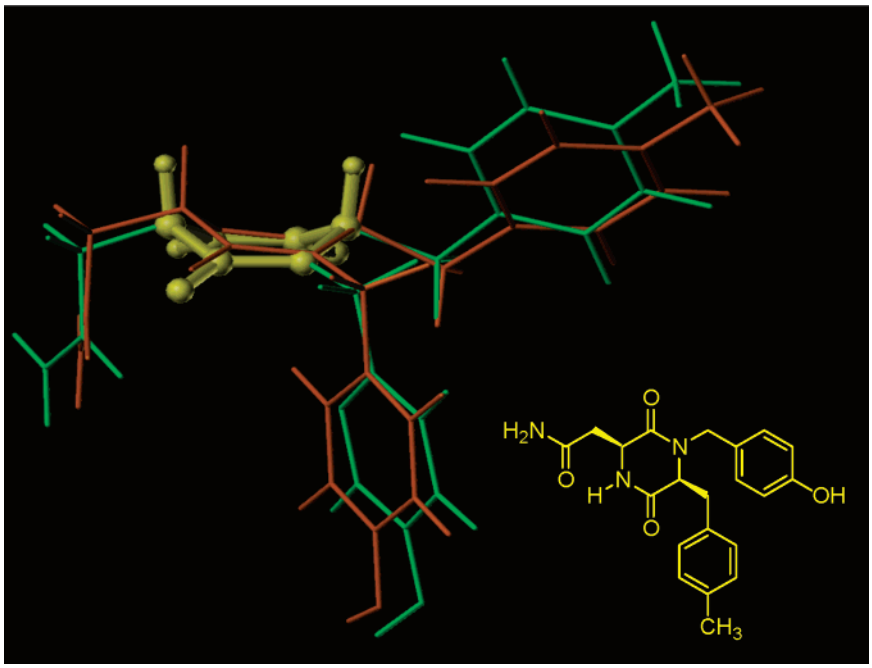


Figure 8. Docking poses for a compound from the Sz11K library obtained from OptiDock (yellow) or directly from FlexX (red). The calculated binding energies were −39.6 and −37.1 kJ/mol, respectively.

the observation that exhaustive sampling of reference core placements ($T = 108$) showed a similar correlation ($R^2 = 0.58$; Figure 7C).

The displacement of OptiDock scores toward more favorable energies in Figure 7 is particularly clear for the lower affinity ligands and probably accounts for the reduced correlation, since these correspond to better docking configurations than those used to parametrize FlexX. Such parametrization entails an implicit assumption that the binding modes found are optimal for the scoring function; at least for these ligands and this target, this is evidently not a safe assumption. Given the prominence of thrombin and related proteases in the FlexX training set, it seems likely that the observed discrepancy is a statistical artifact of overfitting during training and that it could be relieved by reparametrization using OptiDock docking. Fortunately, the effect is small, even within the FlexX training set and is unlikely to be significant outside of it.

In any event, the correlation between vHTS docking score (energy) and experimental affinity is better in all three cases than are those seen for most scoring functions described to date.[4,6,31] The good performance seen here probably reflects the aforementioned presence of thrombin complexes and complexes of related proteases in the training set used to develop the FlexX scoring function.[1] Most scoring functions perform less well when entries from the training set are excluded from the test set, where unrelated compounds are employed and where multiple targets are considered.[4,6,29,31]

Example. Figure 8 shows the OptiDock result for one particular product from the Sz11K library, along with the corresponding pose obtained by applying FlexX directly. The reference placement and side-chain configurations for the

OptiDock pose correspond to those highlighted in yellow and green in Figure 2. The OptiDock energy calculated from eq 5 was better in this case for the reference placement highlighted in purple, but the associated combination of substituent configurations introduced a steric clash in the fully assembled product (indicated by the arrow in Figure 2C), which caused it to be rejected by OptiDock.

Note that the OptiDock and FlexX poses are very similar, despite the fact that the product in question was not among those used to generate reference placements, nor were the substituents it bears present on the products corresponding to its two best reference placements (Figure 2A,B). Note, too, that the pose generated by OptiDock is lower in energy by 2.5 kJ/mol.

Caveat. There is an implicit assumption in the OptiDock strategy that each substituent in a combinatorial product interacts more or less independently with the protein target. This is a reasonable expectation for libraries in which the variation sites are well-separated in space and seems to hold true for the three libraries considered here. The assumption may well break down, however, if the core bears geminal or vicinal variation sites, in part because there will be too many steric clashes among the "best" R-core poses. In such cases, the complementary approach taken in FlexX^C, where such interactions are taken into account directly, may prove more productive.

Discussion

The development of OptiDock was stimulated by perceived inadequacies in the CombiDOCK and FlexX^C protocols. Both make the assumption that fragments of compounds are adequate for determining binding modes, whereas the OptiDock procedure starts from the less restrictive assumption that a representative sampling of whole compounds can adequately define the binding modes. In trials in which either FlexX^C or OptiDock were employed, we found that fragments are rarely adequate to define binding modes. For example, FlexX^C can be used in a manner that only the core is used to define binding modes, akin to the way in which CombiDOCK works. Unfortunately, optimal binding of the core often leaves no room in the binding cleft to accommodate addition of substituents larger than halogens at the variation points. Defining binding modes from the substituents at a specific variation point often leads to similar problems: there was not enough room to add substituents. In general, docking cores only worked well when the cores were relatively large and the substituents small. Docking substituent lists (FlexX^C methodology) likewise was seen to be useful only in cases that one variation site dominated the definition of binding modes. In short, we observed that the most effective way to identify binding modes is by docking complete products.

We have approached combinatorial docking as a sampling problem, working with constructs for which the population grows in proportion to the scale of the reagent space rather than to that of the product space. By extracting reference core placements from poses obtained by docking full products, the OptiDock method is able to efficiently take into consideration a range of critical binding interactions that encompasses the full structural scope of a combinatorial library. In partcular, docking intact products provides a range of "holes" for other substituents to fill. The hole left by one substituent at a particular variation site can guide FlexX to grow the R-core for a *different* substituent along an incremental construction trajectory that, in the end, yields a better interaction with the target protein than would otherwise be found.

Another important benefit of our approach is that failure rates are reduced compared to other combinatorial docking or serial docking methods. Indeed, OptiDock is able to dock each member of the K1K dataset where CombiDock had a 25% failure rate and where FlexX serial failed to dock 5%. Another is that broad sampling can produce better scoring poses than those generated by serial FlexX docking through forced consideration of more diverse docking modes, a result evident in all the datasets we examined. Moreover, this very desirable result is achieved with no loss of CPU efficiency.

A last crucial strength of approaching docking explicitly as a sampling problem is that doing so makes it possible to determine how well the docking mode space is being explored. In OptiDock, this can be done by comparing results from independent runs that differ only in the random number seed used. Getting significantly divergent results indicates that important binding modes are not being consistently explored and that a broader sampling of products is needed to adequately cover conformational space. This ability to directly evaluate the stability of the methodology is not available in other docking approaches.

Although not discussed in detail here, it should be obvious that training sets and reference core placements can also be specified directly should external data, for example, known affinities or binding modes evident from X-ray crystallographic analyses or NMR studies, be available that make such biasing desirable.

Though we used the FlexX program as the docking engine for the applications described here, the methodology is equally applicable to any flexible docking procedure that uses fitness and scoring functions (e.g., the alternative FlexX function DrugScore[32] or, as noted above, those used in DOCK[2,13,14]) lacking in intramolecular terms, provided that it allows for constrained placement of a fixed core structure while letting substituents flex.

Acknowledgment. This work was underwritten by Pfizer Global Research and Development (Ann Arbor, MI). The authors graciously acknowledge the involvement and support of Drs. Mark Snow and Christine Humblet. D.G.S. thanks Dr. C. Bayly (Merck Frost, Montreal, Quebec) for constructive criticism of the protocol.

References and Notes

(1) Rarey, M.; Kramer, B.; Lengauer, T.; Klebe, G. *J. Mol. Biol.* **1995**, *251*, 470−489.
(2) Kuntz, I. D.; Blaney, J. M.; Oatley, S. J.; Langridge, R.; Ferrin, T. E. *J. Mol. Biol*. **1982**, *161*, 269−288.
(3) FRED is a product of OpenEye Scientific Software, Santa Fe, NM 87507; www.eyesopen.com.
(4) Bissant, C.; Folker, G.; Rognan, D. *J. Med. Chem.* **2000**, *43*, 4759−4767.
(5) Kearsley, S. K.; Underwood, D. J.; Sheridan, R. P.; Miller, M. D. *J. Comput.-Aided Mol. Des.* **1994**, *8*, 565−582.

(6) Knegtel, R. M.; Bayada, D. M.; Engh, R. A.; von der Saal, W.; van Geerestein, V. J.; Grootenhuis, P. D. *J. Comput.-Aided Mol. Des.* **1999**, *13*, 167−183
(7) Goodsell, D. S.; Olson, A. J. *Proteins* **1990**, *8*, 195−202.
(8) Liu, M.; Wang, S. *J. Comput. Aided. Mol. Des.* **1999**, *13*, 435−451.
(9) McMartin, C.; Bohacek, R. *J. Comput.-Aided Mol. Des.* **1997**, *11*, 333−344.
(10) Metaphorics, LLC, Mission Viejo, CA 92691; www.metaphorics.com.
(11) Jones, G.; Wilett, P.; Glen, R. C.; Leach, A. R.; Taylor, R. *J. Mol. Biol.* **1997**, *267*, 727−748.
(12) Welch, W.; Ruppert, J.; Jain, A. N. *Chem. Biol.* **1996**, *3*, 449−462.
(13) Leach, A. R.; Kuntz, I. D. *J. Comput. Chem.* **1992**, *13*, 730−748.
(14) Sun, Y.; Ewing, T. J.; Skillman, A. G.; Kuntz, I. D. *J. Comput.-Aided Mol. Des.* **1998**, *12*, 597−604.
(15) Rarey, M.; Lengauer, T. *Perspect. Drug Discvery Des.* **2000**, *20*, 63−81.
(16) OptiDock is available as part of the CombiDock module in SYBYL and is available exclusively from Tripos Inc., St. Louis, MO 63144; www.tripos.com.
(17) Ash, S.; Cline, M. A.; Homer, R. W.; Hurst, T.; Smith, G. B. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 71−79.
(18) Clark, R. D. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 1181−1188.
(19) Clark, R. D.; Langton, W. J. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 1079−1086.
(20) Clark, R. D. U.S. Patent No. 6,535,819, 2003.
(21) Grower, J. C. In *Encyclopedia of Statistical Sciences*; Kotz, S., Johnson, N. L., Eds.; John Wiley and Sons: New York, 1985; Vol. 5, pp 397−405
(22) Patterson, D. E.; Cramer, R. D.; Ferguson, A. M.; Clark, R. D.; Weinberger, L. E. *J. Med. Chem.* **1996**, *39*, 3049−3059.
(23) Szardenning, A. K.; Antonenko, V.; Campbell, D. A.; DeFranscisco, N.; Ida, S.; Shi, L.; Sharkov, N.; Tien, D.; Wang, Y.; Narve, M. *J. Med. Chem.* **1999**, *42*, 1348−1357.
(24) Lovejoy, B.; Welch, A. R.; Carr, S.; Luong, C.; Broka, C.; Hendricks, R. T.; Campbell, J. A.; Walker, K. A.; Martin, R.; Van Wart, H.; Browner, M. F. *Nature Struct. Biol.* **1999**, *6*, 217−221.
(25) Kick, E. K.; Roe, D. C.; Skillman, A. G.; Liu, G.; Ewing, T.JA; Sun, Y.; Kuntz, I. D.; Ellman, J. A. *Chem. Biol.* **1997**, *4*, 297−307.
(26) Baldwin, E. T.; Bhat, T. N.; Gulnik, S.; Hosur, M. V.; Sowder, R. C., II; Cachau, R. E.; Collins, J.; Silva, A. M.; Erickson, J. W. *Proc. Natl. Acad. Sci. U.S.A.* **1993**, *90*, 6796−800.
(27) Linusson, A.; Gottfries, J.; Olsson, T.; Ornskov, E.; Folestad, S.; Norden, B.; Wold, S. *J. Med. Chem.* **2001**, *44*, 3424−3439.
(28) Skordalakes, E.; Dodson, G. G.; Green, D. S.; Goodwin, C. A.; Scully, M. F.; Hudson, H. R.; Kakkar, V. V.; Deadman, J. J. *J. Mol. Biol.* **2001**, *311*, 549−555.
(29) Clark, R. D.; Strizhev, A.; Leonard, J.; Blake, J. F.; Matthew, J. B. *J. Mol. Graphics Modell.* **2002**, *20*, 281−295.
(30) Mood, A. M.; Graybill, F. A.; Boes, D. C. *Introduction to the Theory of Statistics*, 3rd ed; McGraw-Hill: New York, 1974; p 233.
(31) Charifson, P. S.; Corkery, J. J.; Murcko, M. A.; Walters, W. P. *J. Med. Chem.* **1999**, *42*, 5100−5109.
(32) Gohlke, H.; Hendlich, M.; Klebe, G. *J. Mol. Biol.* **2000**, *295*, 337−356.

CC034068X