Title
Tools for ligand discovery and design

Author
Meng, Elaine Chung-su

Publication Date
1993

Peer reviewed|Thesis/dissertation

TOOLS FOR LIGAND DISCOVERY AND DESIGN:

MOLECULAR DOCKING AND STRUCTURAL DATABASES

by

Elaine Ching-su Meng
B.S.Pharm., University of Cincinnati, 1988

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

PHARMACEUTICAL CHEMISTRY

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA

San Francisco

Preface

A revolution is occurring in which disease is being studied and understood on an increasingly microscopic scale. In some cases, conditions are known to result from a single genetic defect; in many others, certain pathological processes but not the disease etiologies have been elucidated at the molecular level. Much progress can be ascribed to recombinant DNA technology, which allows genes to be amplified, detected, sequenced, altered, and expressed in heterologous systems. Not only can this technology increase our understanding of disease genetics, it can yield the quantities of protein necessary for structure determination.

The database of known protein structures has increased rapidly in recent years. Although most conditions cannot be traced to a single molecular defect, there are generally several points at which a pharmaceutical agent can act to alter the course or symptoms of a disease. The targets for action are often proteins. When the structure of a target protein is known, one can, in principle, find or devise a compound that will bind to it and affect its biochemical activity.

I have worked to develop and improve computational tools for structure-based ligand discovery and design. Ligand discovery has traditionally occurred through serendipity or large-volume screening; design has consisted mainly of modifications to known ligands, including substrates. Frequently, the pharmacokinetic properties of known ligands (peptides, nucleotides, and evanescent neurotransmitters, for example) limit their use as therapeutics; typical problems are poor oral absorption, poor distribution to the site of action, and rapid clearance by metabolism or excretion. Solutions include using novel delivery systems, making incremental changes in ligand structure that improve phar-

macokinetic properties, and finally, the discovery of truly novel ligands (compounds chemically dissimilar to known ligands) with acceptable pharmacokinetic properties. The development of consistently useful approaches to any of these would strongly impact biomedical research; I concentrate on the latter.

There is a gap in the computational arsenal that has been used for structure-based ligand design. On the one hand are rigorous statistical-mechanical approaches that can yield relative free energies of binding, in the limits of an accurate force field and thorough sampling of system configurations. These methods are costly, and results are reliable only when the ligands being compared are similar in structure and binding mode. The docking problem, that of identifying probable binding modes, must have already been solved for each system of interest. On the other hand, the docking algorithms available do not evaluate complementarity with much sophistication. Even so, many of the methods are prohibitively slow; the geometric aspects of docking alone generate combinatorial explosions.

A logical plan of attack is to start with a rapid and robust geometric docking algorithm, preferably tunable in terms of the thoroughness of sampling orientation space, and to increase the complexity of scoring (energy evaluation) as much as possible while retaining computational feasibility. Prof. Kuntz and my esteemed predecessors in DOCK development provided the starting point; I focused on new scoring procedures (this is not to say, however, that I had analyzed the problem with such clarity when embarking on the project!).

DOCK versions 2.0 and earlier perform simple contact or "shape" scoring. An important time-saving development by Brian Shoichet was the precalculation and storage of scores for points on a grid. He also initiated the use of electrostatic potential maps.

Using a more sophisticated scoring scheme requires more ligand information. Point charges are necessary for electrostatic scoring, for example, and atom types are necessary for assigning van der Waals parameters. In contrast, for contact scoring, all nonhydrogen atoms are equivalent and charges are not considered. Generation of additional ligand information can be a major task, as the databases used in DOCK searches can contain thousands or even hundreds of thousands of compounds. It is not simply a matter of time; ideally, the process should occur with a minimum of human intervention. When the database in its starting form contains only heavy atom coordinates and atomic numbers, atom types (hybridization states) must be identified to allow hydrogen addition, charge calculation, and the assignment of van der Waals parameters. Manual atom type determination for all the compounds in a large database is not feasible. Chapter 1 describes IDATM (Appendix 4), a program that uses coordinates to discern atom types. Chapter 1 is essentially the publication: E. C. Meng and R. A. Lewis, *J. Comp. Chem.*, **12**, 891 (1991). Richard Lewis encouraged me to develop the algorithm and publish the work.

I used an adaptation of IDATM to make DOCK databases from the crystal structures of small organic compounds (Appendix 3). Even when the atom types are known, however, the remaining steps in database creation can be daunting. I describe database generation from other starting points in Appendices 1 and 2.

Chapter 2 describes the development and testing of a score that approximates molecular-mechanical interaction energies. With single mode docking, I regenerate known complex structures using the conformations present in the complexed state. Use of the complexed conformations simplifies testing, so that any negative results can be ascribed to sampling problems ("correct" binding modes not found) or scoring problems

("correct" binding modes found but not identified). Although it is important to see whether binding modes can be predicted without prior knowledge of the relevant conformations, it is essential to isolate and test aspects of the procedure individually before investigating more complicated situations. It is encouraging that the estimated interaction energy is successful in identifying the experimental binding mode in all four test cases. The other scoring methods tested are less successful. In addition, the time requirements are minimal since sums over receptor atoms are precalculated and stored for points on a grid (program CHEMGRID, Appendix 5). The molecular-mechanical or "force field" score is the major new feature of DOCK 3.0. Chapter 2 is a longer version of the publication: E. C. Meng, B. K. Shoichet, and I. D. Kuntz, *J. Comp. Chem.*, **13**, 505 (1992).

Of course, a correct orientation must be generated before any scoring method can identify it. In Chapter 3, I examine how much sampling is necessary to reproduce experimental geometries, using the same test cases as in Chapter 2. There is a tradeoff between sampling and rigid-body minimization: for the correct orientations to receive the best scores, intensive sampling is necessary, or a combination of moderate sampling and minimization. If sampling is too sparse, however, minimization cannot redeem the situation. It is currently more efficient to sample thoroughly than to combine low-to-moderate sampling with minimization. This may change if a faster minimization algorithm is implemented or if minimization of only a subset of the orientations is performed.

One of the major shortcomings of estimated interaction energies as proxies for free energies of association is neglect of the partial desolvation that occurs upon binding. This approximation is most problematic when different ligands as well as different orientations of the same ligand are being compared, as in a database search. In Chapter 4, I

use ligand atom hydrophobicities and degrees of burial to estimate desolvation contributions to binding. I use atomic contributions to the octanol-water logP and a simple element-based assignment as measures of hydrophobicity, and investigate the use of desolvation terms alone and in combination with the force field score. In the α-chymotrypsin system, desolvation terms improve the correlation between apparent binding energy and score. Somewhat surprisingly, the simple hydrophobicities are more successful than the logP-derived hydrophobicities. The approach needs to be evaluated in more systems, however, as it is unclear whether the improved agreement with experiment is primarily due to a nonspecific selection for greater hydrophobicity.

Sampling and scoring issues in DOCK are far from resolved. Appendix 6 is a case study that reveals some of the difficulties that can arise in a real-life application; I describe the use of DOCK in HIV-1 protease inhibitor discovery and design. The major barriers to prediction in this system have been a lack of knowledge of the complexed conformations of receptor and ligand, preconceptions about which part of the active site would be occupied, and the probable existence of multiple binding modes.

It is simply not possible to sample conformations or orientations exhaustively. Orientational sampling is tunable and fairly robust in DOCK, so conformational sampling is more likely to be limiting. DOCK development continues, and will continue, to focus on dynamic and static ways of including conformational flexibility, general speed-ups involving pruning of the combinatorial matching tree, and scoring methods that will yield better estimates of free energies of binding.

There are many without whom my journey toward a doctoral degree would have been difficult, perhaps impossible. I am first of all grateful to my research advisor, Prof. Irwin Kuntz, who has consistently provided the right mix of encouragement, guidance, and freedom. Profs. Peter Kollman and Fred Cohen have also given generously of their time and advice, as orals and dissertation committee members, and on multiple occasions throughout the years.

Renée DesJarlais, George Seibel, David Pearlman, Richard Lewis, and Andrew Leach were important sources of information and encouragement early on; from my point of view, they were gone too soon. Scott Presnell, Jim Caldwell, and Eric Pettersen have provided technical assistance and levity all along. Baked goods will never be sufficient to repay them.

Many thanks to my DOCK contemporaries, Brian Shoichet and Dale Bodian, for thousands of conversations both serious and frivolous. I hope that my friends and newer officemates Diana Roe, Cindy Corwin, and Dan Gschwend will benefit from the environment as I have. Finally, I could never have survived without my classmates Carolyn Koo, Christine Ring, and Randy Radmer to complain to and laugh with during the various stages of graduate student life, and Jason Rosé to trade and discuss books with.

A different category of gratitude is owed to my family. My father, Hsien-ming Meng, taught me to love books and work earnestly toward my goals. My mother, Linda Reiss, has always made me feel capable of reaching those goals. Lastly, I hope that my talented sister, Annette Meng, is as proud of me as I am of her.

TOOLS FOR LIGAND DISCOVERY AND DESIGN:

MOLECULAR DOCKING AND STRUCTURAL DATABASES

Elaine C. Meng

Dissertation Abstract

The opportunities for *ab initio* drug design are more numerous now than ever before; the molecular bases of a growing number of diseases are known, and the structures of macromolecules are being solved at an accelerating rate.

The ability to propose reasonable ligand-receptor binding geometries is crucial to the success of structure-based design. One approach is to "dock" molecules together in many ways and then "score" or evaluate each orientation. In a database of compounds, those which score well should be more likely to bind to the target macromolecule.

The overarching theme of this dissertation is the development of computational tools to aid structure-based ligand discovery and design.

In Chapter 1, I present a method for determining connectivity and hybridization states given the nonhydrogen atom coordinates of molecules. This is useful for automated parameter assignment within large, heterogeneous databases of organic structures.

Chapter 2 describes the addition of molecular mechanics scoring to a rapid, geometric docking algorithm. Computational costs are mimimal because sums over receptor atoms are precalculated. In four test cases where crystallographically determined complexes are redocked, the "force field score" correctly identifies orientations closest to the experimental geometry; other scoring functions are less successful.

Improving the evaluation of orientations of a single molecule is important for improving the method's ability to find lead compounds in databases.

In Chapter 3, I examine the same systems at various levels of orientational sampling, with and without rigid-body minimization. For the correct orientations to receive the best scores, intensive sampling is required, or moderate sampling combined with minimization. Presently, it is more time-efficient to sample thoroughly than to combine low-to-moderate sampling with minimization.

Serious simplifications include the neglect of flexibility and partial desolvation. In Chapter 4, I describe the use of atomic hydrophobicities to model desolvation. A simple hydrophobicity assignment is apparently as useful as a more complex one based on partitioning.

Appendices 1-3 chronicle the generation of dockable databases starting with different amounts of structural information. Appendices 4 and 5 contain the source code for IDATM (Chapter 1) and CHEMGRID (Chapter 2), respectively. Appendix 6 describes modeling with the HIV-1 protease.

Table of Contents

List of Tables

List of Figures

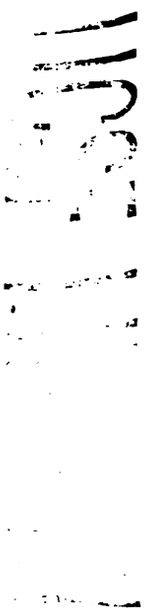
Chapter 4:

Appendix 6:

CHAPTER 1: DETERMINATION OF MOLECULAR TOPOLOGY AND ATOMIC HYBRIDIZATION STATES FROM HEAVY ATOM COORDINATES

Elaine C. Meng* and Richard A. Lewis†

Department of Pharmaceutical Chemistry, School of Pharmacy, University of California, San Francisco, California 94143-0446



**To whom all correspondence should be addressed.*

†Current address: Imperial Cancer Research Fund, Biomolecular Modeling Laboratory, P.O. Box 123, Lincoln's Inn Fields, London WCZA 3PX, England, United Kingdom

ABSTRACT

A method is presented for the derivation of hybridization states and connectivity within molecules from the atomic numbers and coordinates of heavy atoms. The algorithm utilizes bond length data from studies of the Cambridge Structural Database (Allen *et al., J. Chem. Soc. Perkin Trans. II,* S1, (1987)). The program, IDATM, is useful for processing input to hydrogen-adding routines and molecular mechanics programs, as it minimizes the amount of manual preprocessing required. IDATM has been tested on a range of crystallographically determined structures, including poorly determined structures, with a successful assignment of hybridization for over 99% of the atoms in the set.

INTRODUCTION

The availability of coordinates for small organic molecules has increased dramatically in recent years, due to diffraction experiments and the use of coordinate-generating programs such as CONCORD,[1,2] WIZARD,[3] and COBRA.[4] Many experimentally derived structures, however, do not include hydrogen positions; X-ray diffraction is generally unable to yield this information. While some of the lost knowledge may be present in the original references, it may not be readily accessible. Retrieval of hybridization information from the literature becomes impossible when hundreds or thousands of molecules are to be examined.

The term "heavy atom" will be used in this paper to refer to all nonhydrogen atoms. Knowledge of the number of hydrogens bonded to each heavy atom, or equivalently, the hybridization state of each heavy atom, is essential for detailed molecular modeling; otherwise, point charge calculations cannot be performed, and atoms cannot be associated with the appropriate parameters for molecular mechanics studies. We present a geometry-based algorithm, IDATM, for determining automatically the connectivity and hybridization states of atoms within a molecule. IDATM is designed to deal with underdetermined structures, for which data on hydrogen atom positions may be missing. A hierarchical approach is taken; that is, the least ambiguous situations are handled first and used in the determination of the remaining cases.

INCORPORATION OF BOND LENGTH DATA AND DEFINITION OF ATOM TYPES

A tabulation of bond lengths in organic compounds, as determined by X-ray and neutron diffraction, has recently been published.[5] Prior to the publication of this paper,

the only bond length statistics generally available were those in the Chemical Society Special Publications,[6,7] based on data collected before 1960. The newer tabulation, derived from a subset of the September 1985 version of the Cambridge Structural Database (CSD), takes advantage of subsequent increases in the availability and diversity of well-determined structures.

The program IDATM categorizes atoms according to the number of directly attached nonhydrogen neighbors, allowing more efficient use of the bond length data in discerning hybridization states. For example, different cutoffs apply for distinguishing the pairs a) >CH-CH2- versus >C=CH- and b) >CH-CH3 versus >C=CH2. Cutoffs were selected to fall more than two (and often three) standard deviations away from the mean lengths of the bond types being distinguished from one another, whenever the data allowed. This was possible in virtually all cases. An important exception, however, is sp2-hybridized oxygen versus enol sp3-hybridized oxygen; the bond length distributions for these oxygen types overlap significantly.

Table I contains a listing of the atom type names used in the present work, and descriptions of the corresponding atoms. Table II contains geometric criteria used in "typing" the atoms. Four carbon types, seven nitrogen types, three oxygen types, five sulfur types, four phosphorus types, three boron types, two hydrogen types, and two deuterium types are included (recall that hydrogens may be present in the input structures). Other elements are simply typed according to atomic number; for example, there is only one bromine type. It should be noted that aromatic and double-bonded carbons are not distinguished from one another, nor are certain categories of sp2-hybridized nitrogens; there is a continuum of bond lengths involving these atom types, particularly when highly conjugated or poorly determined structures are considered. We do not believe that

Table I. List of atom types.

Name	Description
C3	sp3-hybridized carbon
C2	sp2-hybridized carbon
C1	sp-hybridized carbon
Cac	carboxylate carbon
N3+	sp3-hybridized nitrogen, formal positive charge
N3	sp3-hybridized nitrogen, neutral
Npl	sp2-hybridized nitrogen
N1	sp-hybridized nitrogen
Nox	N-oxide nitrogen
Ntr	nitro nitrogen
Ng+	guanidinium nitrogen, partial positive charge
O3	sp3-hybridized oxygen
O2	sp2-hybridized oxygen
O-	carboxylate or nitro oxygen, partial negative charge
S3+	sp3-hybridized sulfur, formal positive charge
S3	sp3-hybridized sulfur, neutral
S2	sp2-hybridized sulfur
Sac	sulfate sulfur
Sox	sulfoxide or sulfone sulfur
S	other sulfur
Bac	borate boron
Box	other oxidized boron
B	other boron (not oxidized)
Pac	phosphate phosphorus
Pox	P-oxide phosphorus
P3+	sp3-hybridized phosphorus, formal positive charge
P	other phosphorus
HC	hydrogen bonded to carbon
H	other hydrogen
DC	deuterium bonded to carbon
D	other deuterium

Table II. List of geometric criteria.

Description[a]	Value[b]
sp2 versus sp3 angle cutoff	115.0
angle below which the type of an atom with HAV[c] 2 should be reconsidered	122.0
sp versus sp2 angle cutoff	160.0
upper bond length cutoff, HAV 1 C1 to C1	1.22
upper bond length cutoff, HAV 1 C2 to any C	1.41
upper bond length cutoff, HAV 1 C2 to any N	1.37
upper bond length cutoff, HAV 1 N1 to C1	1.20
lower bond length cutoff, HAV 1 N3 to any C	1.38
lower bond length cutoff, HAV 1 N3 to N3	1.43
lower bond length cutoff, HAV 1 N3 to Npl	1.41
upper bond length cutoff, HAV 1 O2 to C2	1.30
upper bond length cutoff, HAV 1 O2 to As	1.685
upper bond length cutoff, HAV 1 S2 to C2	1.76
upper bond length cutoff, HAV 1 S2 to As	2.11
lower bond length cutoff, HAV 2 C3 to any C	1.53
lower bond length cutoff, HAV 2 C3 to any N	1.46
lower bond length cutoff, HAV 2 C3 to any O	1.44
upper bond length cutoff, HAV 2 Npl to any C	1.38
upper bond length cutoff, HAV 2 Npl to any N	1.32
upper bond length cutoff, HAV 2 C2 to any C	1.42
upper bond length cutoff, HAV 2 C2 to any N	1.41
conditional lower bond length cutoff, C3 to C	1.45

[a]See Table I for types.

[b]Angles in degrees, bond lengths in angstroms.

[c]Heavy atom valence, the number of heavy atoms attached.

this is as serious a problem as it may seem, for the following reasons: 1) the geometry of substituent placement about the central atom does not vary within the categories distinguished, and 2) atoms within the groupings are in many cases assigned the same or similar parameters for molecular mechanics calculations.

THE ALGORITHM

The current input format is Brookhaven Protein Data Bank (PDB) standard,[8] although any format containing coordinates and elemental identities could be accommodated with only minor alterations to the program. Molecules are handled singly, although there is no limitation on the number that can be handled during a given run. The output format reflects the input format except that the atom name is replaced by atom type. Again, it would be relatively easy to change the format, for example to include CONECT records, or to meet completely different specifications.

Two data files are read by IDATM: "attyps" contains the names that the user wishes to associate with the atom types (we have used the names given in Table I), and "params" contains geometric criteria (Table II) for distinguishing between atom types. Usage of data files rather than "hard-wired" variables allows additional flexibility and user control. It should be emphasized that the names used here were chosen to be descriptive, and do not necessarily accord with any particular force field. Different force fields and molecular modeling packages have different conventions for naming atom types, and it is more convenient for the user to switch amongst multiple "attyps" files than to keep multiple edited versions of the program itself. Similar arguments apply to the "params" file, although it is less likely that the user will want to alter this file. The maximum number of atoms per molecule and the maximum number of bonds to a given atom are user-

adjustable parameters; when either limit is exceeded, the user is warned accordingly and program execution stops. A planned improvement is to have the the program merely skip oversized structures rather than stop when they are encountered.

The program makes several passes through the constituent atoms of a molecule to determine connectivity and hybridization. The strategy is to identify the most well-determined features of a structure and then to use this information in subsequent iterations through all the atoms, during which deductions are made about the under-specified features of the structure.

Connectivity. Following the work of Allen *et al.*,[9] two atoms are defined to be bonded if the distance between them is less than or equal to the sum of covalent bond radii for the corresponding elements (Table III) plus a tolerance value. The tolerance is an adjustable parameter; a value of 0.4 angstroms was used in the present work. A relatively large tolerance is needed to find all of the bonds in low-resolution structures.

Heavy atom valence. Any hydrogens or deuteriums present are typed during this loop, according to whether they are bonded to carbon or to some other element. In addition, the number of heavy atoms bonded to each heavy atom is determined by subtracting the number of attached hydrogens from the total number of attached atoms. This "heavy atom valence" (HAV) is important for determining how the atom will be treated in subsequent steps.

Fully determined atoms and atoms with HAV > 1. In the main loop, atoms that are typed simply by element (e.g. bromine) are handled first, and the remaining atoms are grouped by HAV. If the HAV equals 4, carbon atoms must be sp3-hybridized; nitrogen atoms must be quaternary or oxidized; phosphorus atoms must be part of a phosphate, P-oxide, or quaternary phosphine group; sulfur atoms must be part of a sulfate, sulfone, or

Table III. Covalent bond radii used in determining connectivity.[a,b]

Ac	1.88	Er	1.73	Na	0.97	Sb	1.46
Ag	1.59	Eu	1.99	Nb	1.48	Sc	1.44
Al	1.35	F	0.64	Nd	1.81	Se	1.22
Am	1.51	Fe	1.34	Ni	1.50	Si	1.20
As	1.21	Ga	1.22	Np	1.55	Sm	1.80
Au	1.50	Gd	1.79	O	0.68	Sn	1.46
B	0.83	Ge	1.17	Os	1.37	Sr	1.12
Ba	1.34	H	0.23	P	1.05	Ta	1.43
Be	0.35	Hf	1.57	Pa	1.61	Tb	1.76
Bi	1.54	Hg	1.70	Pb	1.54	Tc	1.35
Br	1.21	Ho	1.74	Pd	1.50	Te	1.47
C	0.68	I	1.40	Pm	1.80	Th	1.79
Ca	0.99	In	1.63	Po	1.68	Ti	1.47
Cd	1.69	Ir	1.32	Pr	1.82	Tl	1.55
Ce	1.83	K	1.33	Pt	1.50	Tm	1.72
Cl	0.99	La	1.87	Pu	1.53	U	1.58
Co	1.33	Li	0.68	Ra	1.90	V	1.33
Cr	1.35	Lu	1.72	Rb	1.47	W	1.37
Cs	1.67	Mg	1.10	Re	1.35	Y	1.78
Cu	1.52	Mn	1.35	Rh	1.45	Yb	1.94
D	0.23	Mo	1.47	Ru	1.40	Zn	1.45
Dy	1.75	N	0.68	S	1.02	Zr	1.56

[a]All values in angstroms.
[b]Reference 9.

sulfoxide group; boron atoms may be part of borate, another oxidized group, or a reduced group. Distinctions are made on the basis of number of attached oxygens. If the HAV equals 3, the average of the three bond angles around the central atom is calculated and types are assigned using this value and, if appropriate, the number of attached oxygens. Carbons in this group may be sp3-hybridized or sp2-hybridized (possibly part of a carboxylate group); nitrogens may be sp3-hybridized or sp2-hybridized (possibly part of a nitro group); sulfurs may be positively charged and sp3-hybridized, or part of a sulfoxide group; boron atoms may be in a reduced state or an oxidized state. The average bond angle has been found to be a reliable indicator of hybridization status (see Discussion). If the HAV equals 2, carbons and nitrogens may be sp3-hybridized, sp2-hybridized, or sp-hybridized; oxygens and sulfurs must be sp3-hybridized. Only one bond angle can be calculated for atoms in this group, and is not a very reliable indicator of hybridization status. Carbons and nitrogens are assigned a type according to bond angle, but are marked for further examination.

Atoms with HAV = 1. The atoms with HAV equal to 1 are dealt with after completion of the main loop, so that the types of their bond partners as well as the bond lengths can be utilized.

Resolution of ambiguous cases and inconsistencies; identification of charged groups. During the last two passes through the atoms of a molecule, previously assigned types are reexamined. First, the atoms that had been tagged for further consideration are retyped, if necessary, using bond length information. Next, decisions are made regarding the charge states of atoms: 1) sp3-hybridized nitrogens bonded only to sp3-hybridized carbons and/or hydrogens and/or deuteriums are assigned a positively charged type; 2) guanidinium groups are identified, and their nitrogens are typed accordingly; 3) carboxy-

late and nitro group oxygens are identified and typed. In this manner, groups are assigned the charge states that are most probable at physiological pH. Finally, sp2-hybridized carbons bonded to only sp3-hybridized atoms (or other atom types that could only be contributing a single bond) are identified and retyped as sp3-hybridized carbons.

DEVELOPMENT AND TESTING

Since the bond length criteria were derived from experimental data, changes in IDATM during development consisted mainly of small adjustments in bond angle cutoffs and the addition of conditional statements to handle situations not provided for in the original code. For example, the presence of three sp2-hybridized nitrogens bonded to an sp2-hybridized carbon was initially assumed equivalent to the presence of a guanidinium moiety, such as in an arginine side chain. Although successful in identifying guanidinium groups, this assumption led to errors whenever guanidine and similar structures were encountered. In fact, it is necessary to go beyond the nitrogens and check if any of them are bonded to more than one sp2-hybridized carbon; if so, the nitrogens are not given the guanidinium type. Another change involved the section of the code that corrects inconsistencies. As described above, isolated sp2-hybridized carbons are found and retyped as sp3-hybridized carbons, since an sp2-hybridized carbon must have at least one sp2-hybridized bond partner. Originally, the correction was made only if each nearest neighbor was either sp3-hybridized or a hydrogen atom. It was soon realized, however, that additional types could only be participating in a single bond with the atom in question: carboxylate carbon, phosphate phosphorus, sulfate sulfur, and others.

The development set consisted of structures from the CSD (members of the antibiotic class having refcodes starting with 'A' or 'B', cyclic peptides, opiate alkaloids, and

substituted cyclobutadienes) and 25 residues from the Brookhaven file 2alp (alpha-lytic protease), together comprising 1667 nonhydrogen atoms. Antibiotics were chosen for their biological relevance and diversity of structure; nucleotides, peptides, macrocycles, and other organics are represented within this class. The refcode restriction was just an arbitrary way of sampling a subset of the antibiotics. The other molecules and the portions of 2alp were included for additional variety.

Objectives during program development were: exposure of IDATM to a wide range of structural possibilities, discovery of unforeseen circumstances, and derivation of proper rules to use in these new situations. IDATM was neither specifically nor tightly optimized for the particular structures contained in the development set; no high-level pattern recognition or fine-tuning of geometric criteria was employed. We believe that the program adjustments would have been similar had any other heterogeneous development set been used.

Information on the test set is given in Tables IV and V. In addition to the structures from the development set, the test set included antibiotics from the CSD having refcodes starting with 'N', 'O', or 'P', 11 more residues from 2alp, and 54 nucleotides taken from the Brookhaven files 1ana, 1bna, 1zna, 5ana, and 9dna. There was at least one example of each of the 20 standard amino acids, and at least six examples of each standard nucleotide. Whenever present, hydrogens, counterions, and small solvent molecules were removed from the CSD files; this resulted in a total of 3027 atoms to be typed (4435 including the excerpts from Brookhaven files, described above). Of 91 molecules from 81 CSD files, 49 had originally contained hydrogens; the average R-factors were 0.0665 (range 0.0360-0.1720) and 0.0998 (range 0.0450-0.1800) for the hydrogen-containing and non-hydrogen-containing structures, respectively (no R-factor was reported for the

Table IV. Test set.[a,b]

Refcode/entry	R-factor/resln.	Atoms	Errors	Hydrogens originally present
ACTBOL	------	21	0	n
ACTBOL	------	21	0	n
ACTBOL	------	21	0	n
ACTBOL	------	21	0	n
NDMSCN	0.0360	30	0	y
AAGAGG10	0.0373	27	0	y
PILLMA	0.0380	39	0	y
ACMBPN	0.0400	24	0	y
NBPENC	0.0414	22	0	y
ACFUCN	0.0430	14	0	y
OXOFMB	0.0430	20	0	y
NAHACA	0.0440	8	0	y
AAGGAG10	0.0445	27	0	y
APOMRC	0.0450	20	0	n
APOMRC	0.0450	20	0	n
MORPHM	0.0450	21	0	y
MORPHC	0.0460	21	0	y
NONACU	0.0460	52	0	n
TBUCBD10	0.0460	20	0	y
ANTMYC01	0.0480	24	0	y
APYMPR	0.0480	17	1	n
BEVJER10	0.0480	24	0	y
TBUCBD02	0.0480	20	0	y
ANTMYC03	0.0500	24	0	n
NIGERI	0.0500	51	0	n
PIPBCX	0.0500	33	0	n
PIPCIL	0.0500	36	0	y
ACMPXC	0.0510	24	0	y
ACANOB	0.0520	24	0	y
BAMLIK	0.0530	38	0	y
OXYTET01	0.0540	33	0	n
PXMPEN	0.0540	23	0	y
PURMYC10	0.0550	34	0	y
TBUCBD01	0.0550	20	0	y
PODACE	0.0560	24	0	y
AMICET10	0.0600	44	0	y
OXYTET	0.0600	33	0	y
AMOXCT	0.0610	25	0	y
CIMMUG	0.0610	22	0	y
NAHACB	0.0610	8	0	y
AMDMCN	0.0620	14	0	n
ANTETC	0.0620	30	1	n
CIMNAN	0.0620	21	0	y
AGNGEC11	0.0650	51	0	n
ANTROS01	0.0650	60	0	y
NETRSN	0.0670	31	0	y
NAPMYC10	0.0700	30	0	y
AZPCOH	0.0720	8	0	y
OXTETD	0.0720	33	1	y
NEBULR	0.0730	18	0	y
PRPENG	0.0730	17	1	y
PRPENG	0.0730	23	0	y
AOTETC	0.0760	39	1	y
PRMESA	0.0760	17	0	y
OXERTH	0.0770	58	2	y

Refcode/entry	R-factor/resln.	Atoms	Errors	Hydrogens originally present
PENTBH10	0.0770	18	0	y
OXTETK	0.0800	33	3	n
APLASM	0.0840	55	0	n
ABHPTB	0.0850	23	0	n
ABHPTB	0.0850	23	1	n
ANTSUL	0.0866	22	0	y
ANFLCN	0.0870	59	0	n
ANTINA	0.0930	75	0	n
ACTDGU10	0.0940	19	0	n
ACTDGU10	0.0940	19	2	n
NONACT	0.1030	52	0	y
ANSMYC10	0.1050	23	0	n
AMCILL	0.1060	24	0	y
ANTBPE	0.1070	34	0	y
ANTBRN	0.1070	75	2	n
NONAMT	0.1080	52	0	y
ACIGRA	0.1090	45	0	n
AERMYC10	0.1090	53	2	n
OTETCB	0.1170	33	2	n
NONKCS	0.1251	52	0	n
MORPHI	0.1300	21	0	n
PMEPEN	0.1300	24	1	y
PROMYC10	0.1300	16	1	n
PROMYC10	0.1300	7	0	n
PROMYC10	0.1300	84	0	n
PROMYC10	0.1300	84	2	n
PRTYLD	0.1310	28	0	n
PILLBA10	0.1330	29	0	n
HAZMOR	0.1340	24	0	y
AMPIAB10	0.1370	69	0	n
NIVBIO	0.1400	44	1	n
PEANNA	0.1400	66	0	n
PEANAG	0.1500	66	0	n
NONACS	0.1720	52	0	y
NOSHEP10	0.1800	82	5	n
PRMARI	0.1800	62	6	n
2alp	1.70	279	4	n
1ana	2.10	163	0	n
1bna	1.90	486	0	n
1zna	1.60	158	0	n
5ana	2.25	161	0	n
9dna	1.80	161	0	n

[a]Reference 8.

[b]Reference 9.

Table V. Correspondence of molecule names with CSD refcodes.[a,b]

Refcode	Compound
ACTBOL	actinobolin
ACTBOL	actinobolin
ACTBOL	actinobolin
ACTBOL	actinobolin
NDMSCN	nodusmicin
AAGAGG10	cyclo-(Ala-Ala-Gly-Ala-Gly-Gly)
PILLMA	pillaromycin A
ACMBPN	2-amino-N-(3-dichloromethyl-3,4,4A,5,6,7-hexahydro-5,6,8-trihydroxy-3-methyl-1-oxo-1H-2-benzopyran-4-yl) propanamide
NBPENC	*p*-nitrobenzyl-5-pen-2-em-3-carboxylate
ACFUCN	N-acetylfuranomycin
OXOFMB	oxoformycin B
NAHACA	hadacidin
AAGGAG10	cyclo-(Ala-Ala-Gly-Gly-Ala-Gly)
APOMRC	apomorphine
APOMRC	apomorphine
MORPHM	morphine
MORPHC	morphine
NONACU	nonactin
TBUCBD10	tetra-*t*-butylcyclobutadiene
ANTMYC01	anthramycin methyl ether
APYMPR	β-amino-β-(4-amino-6-carboxyamino-5-methylpyrimidin-2-yl) propionic acid amide
BEVJER10	bis-(dimethylammonium)-octacyanotetramethylidenecyclobutanediide
TBUCBD02	tetra-*t*-butylcyclobutadiene
ANTMYC03	anthramycin methyl ether
NIGERI	nigericin
PIPBCX	6-β-phthalimido-6-α-methylpenam-3-α-*p*-bromocarboxanilide β-oxide
PIPCIL	piperacillin
ACMPXC	4-acetyl-3-methyl-7-β-phenoxyacetamido-Cδ-3-cephem
ACANOB	N-acetylactinobolin
BAMLIK	cyclo-(Gly-His-Gly-Ala-Tyr-Gly)
OXYTET01	oxytetracycline
PXMPEN	phenoxymethylanhydropenicillin
PURMYC10	puromycin
TBUCBD01	tetra-*t*-butylcyclobutadiene
PODACE	phenoxymethyl-Cδ-2-desacetoxycephalosporin
AMICET10	amicetin
OXYTET	oxytetracycline
AMOXCT	amoxicillin
CIMMUG	morphine
NAHACB	hadacidin
AMDMCN	amidinomycin
ANTETC	4-deamino-4-hydroxy-4,11A-anhydrotetracycline
CIMNAN	morphine
AGNGEC11	nigericin
ANTROS01	antibiotic A-130A
NETRSN	netropsin
NAPMYC10	naphthyridinomycin
AZPCOH	trans-2-azabicylo[2.1.0]pentane-3-carboxylic acid
OXTETD	oxytetracycline
NEBULR	nebularine
PRPENG	procaine
PRPENG	penicillin G
AOTETC	5,12A-diacetyloxytetracycline
PRMESA	2,4-diamino-5-(3,4-dichlorophenyl)-6-methylpyrimidine

Refcode	Compound
OXERTH	9-deoxy-11-deoxy-9,11-(imino-(2-(2-methoxyethoxy)-ethylidene)-oxy)-erythromycin
PENTBH10	tetrahydropentalenolactone bromohydrin
OXTETK	oxytetracycline
APLASM	aplasmomycin
ABHPTB	5′-anhydro-7-bromo-8-hydroxy-2′,3′-isopropylidenetubercidin
ABHPTB	5′-anhydro-7-bromo-8-hydroxy-2′,3′-isopropylidenetubercidin
ANTSUL	antibiotic 593A
ANFLCN	acanthofolicin
ANTINA	antibiotic K41 *p*-iodobenzoate
ACTDGU10	deoxyguanosine
ACTDGU10	deoxyguanosine
NONACT	nonactin
ANSMYC10	N-acetylbromoanisomycin
AMCILL	ampicillin
ANTBPE	1-amino-1-(4-bromophenyl)-ethane-bis-(dimethyl-(1-ethyl-4-(2-pyrrolocarbonyl)-ethyltetrahydroindanylbutadienyl)-tetrahydropyran-2-acetic acid)
ANTBRN	antibiotic K41 *p*-bromobenzoate
NONAMT	nonactin
ACIGRA	tri-O-acetyl-O-iodoacetylgranaticin
AERMYC10	anhydroerythromycin A cyclic carbonate methiodide
OTETCB	oxytetracycline
NONKCS	nonactin
MORPHI	morphine
PMEPEN	phenoxymethylpenicillin
PROMYC10	picric acid
PROMYC10	toluene
PROMYC10	prolinomycin
PROMYC10	prolinomycin
PRTYLD	protylonolide
PILLBA10	pillaranone monobromoacetate
HAZMOR	6-deoxy-6-azido-14-hydroxydihydroisomorphine
AMPIAB10	N-iodoacetylamphotericin B
NIVBIO	novobiocin
PEANNA	antibiotic A204A
PEANAG	antibiotic A204A
NONACS	nonactin
NOSHEP10	nosiheptide
PRMARI	9-propionylmaridomycin III

[a]Compounds are in the same order as in Table IV.
[b]Reference 9.

CSD file ACTBOL). Considering these two groups together, the average molecular size was 33 heavy atoms, and the average R-factor was 0.0810.

Each molecule or residue was manually checked against the known structure, found by name in a standard reference, or if necessary, in a journal article. 39 errors in atom type and one bond omission were identified. Mistyping of enol oxygens, a known limitation of IDATM, accounted for six of the errors; the nine other enol oxygens in the test set were typed correctly. The remaining atom type errors resulted directly or indirectly from anomolous bond lengths, often accompanied by bond angles unusual for the true atom types. For instance, an abnormally short CD-NE bond in one of the four arginines taken from 2alp (1.382 angstroms; cutoff 1.41 angstroms; other three arginines: 1.431, 1.438, and 1.454 angstroms) led to the carbon being given the sp2-hybridized type, which in turn led to the three neighboring nitrogens not being identified as part of a guanidinium group. The short bond directly caused one error and indirectly caused three additional errors, all in the same residue. This was the only case of propagated error found in the test results. Another point illustrated here is that even well-resolved macromolecular structures may contain significant localized atomic displacements; the 2alp structure has a resolution of 1.70 angstroms and an R-factor of 0.131.

The missed bond involved two sp3-hybridized carbons in a structure with an R-factor of 0.1300 (refcode MORPHI). The distance between them, 1.860 angstroms, obviously falls outside the normal length distribution for single bonds between sp3-hybridized carbons (mean 1.530 angstroms, standard deviation 0.015 angstroms).[5]

Processing of the entire test set, comprising 36 amino acid residues and 54 nucleotides taken from PDB files as well as 91 molecules from the CSD, took less than six minutes of c.p.u. time on a Convex-C1. The algorithm is implemented in Fortran77 and

is compatible with standard UNIX-environment compilers.

DISCUSSION

The philosophy behind IDATM is the use of simple geometric criteria to derive atom types. There is no extensive pattern recognition, as the regions considered by the most complex conditional statements extend no further than two bonds from the atom of interest. We believe that this strategy enhances speed and robustness while minimizing the occurrence of propagated errors. In addition, IDATM does not employ an energy function. One way of discerning atom types is to calculate the energy of an observed structure more than once, assuming different hybridization states, in order to identify the states for which the calculated energies are lowest. It must be kept in mind, however, that structures with errors (all experimental structures, to some extent) are not "real", in the sense that their energies are not constrained to fit a Boltzmann distribution. In other words, the relative magnitudes of various kinds of displacements are not necessarily consistent with their relative energy costs. Methods using energy functions may be weighting bond length violations too heavily relative to bond angle violations, since stretching force constants are generally greater than bending force constants. For atoms with HAV equal to 3, the average bond angle is an excellent indicator of hybridization status. When strain such as that introduced by a small ring system, for example, reduces one bond angle about an sp2-hybridized atom, the other two angles compensate by being larger. Our results suggest that more complex algorithms for evaluating planarity, such as the calculation of chiral volumes, are unnecessary.

As with any method of identifying atom types, the number of errors per molecule increases as the input structures become less well-determined. In this test set, the number

of errors per structure did not ever exceed two for structures having R-factors under 0.1800.

The present implementation of IDATM assigns charged types to atoms on the basis of expected ionization states at physiological pH; the intent was to maximize relevance for drug design. It would be relatively simple, however, to alter the way any given functional group is assigned charge. Other feasible alterations include 1) detection of rings and addition of aromatic types of carbon and nitrogen, and 2) recognition of planar nitrogens adjacent to carbonyl groups and addition of an amide nitrogen type.

As mentioned above, detailed molecular modeling requires knowledge of atom types. Only with this knowledge can hydrogens be added, point charges calculated, and molecular mechanics studies performed. IDATM was written specifically for "front-end" processing of molecules, type assignment based on coordinates prior to computations that utilize these atom types. In general, read-in facilities currently provided with molecular modeling packages are poorly equipped to discern the atom types in structures other than standard amino acids and nucleotides. This can be a severe handicap when one is reading in structures for which the traditional chemist's diagrams are not readily available. To illustrate this problem, every tenth CSD structure listed in Table IV, beginning with NDMSCN, was read in using the molecular modeling package SYBYL.[10] IDATM mistyped 4 of these 383 atoms, whereas SYBYL mistyped 43 atoms and disconnected one bond. As with the IDATM trials, the input was in PDB format, hydrogens were omitted, and no distinction was made between sp2 hybridization and aromaticity. The SYBYL and IDATM results for a deoxyguanosine molecule (the first one listed in Table V for refcode ACTDGU10) are shown in Figure 1, along with the correct types. In SYBYL, atom types within nonstandard residues are determined only from the number of bonds to

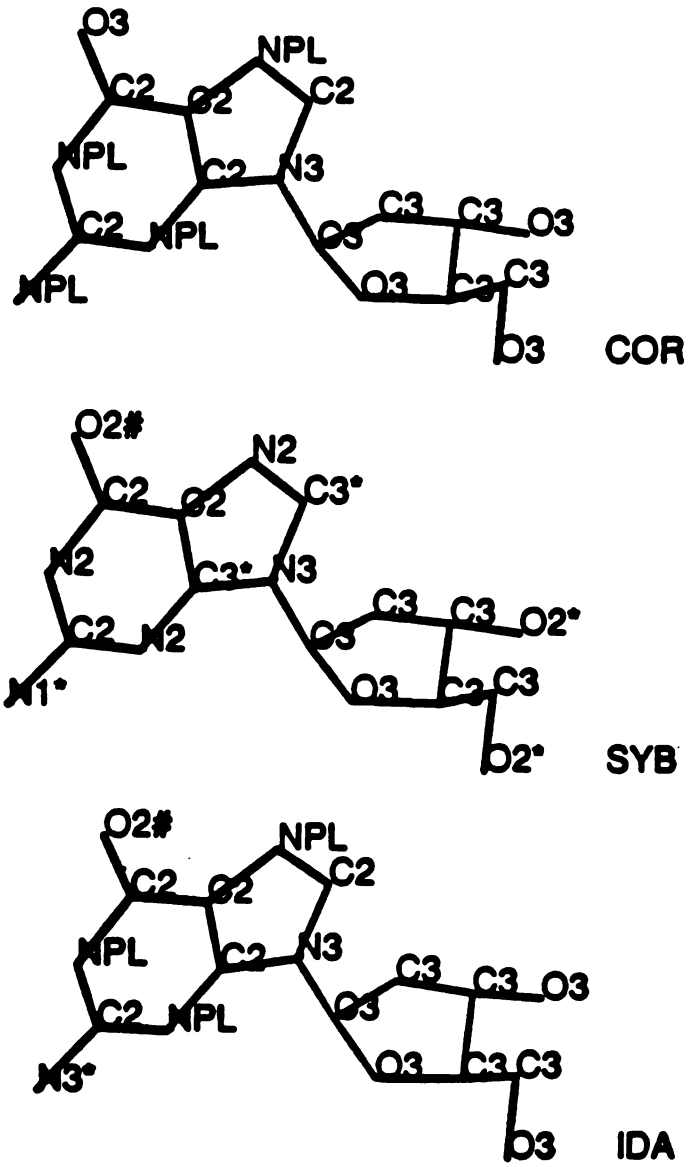


Figure 1. Atom typing results for deoxyguanosine. The coordinates are from the CSD file with refcode ACTDGU10. The correct types according to the IDATM conventions (structure marked COR), the SYBYL results (structure marked SYB), and the IDATM results (structure marked IDA) are shown. Asterisks denote errors; a pound sign signifies an error if deoxyguanosine is assumed to be in the enol form. The types NPL and N2 both represent sp2-hybridized states and are considered equivalent here. Picture generated using UCSF MidasPlus.[11]

each atom; thus, errors occur frequently when hydrogens are missing. Another molecular modeling package, QUANTA,[12] does not attempt to discern types, but relies on the information in the input file and the knowledge of the user (manual editing of atom types is often required). Similarly, MacroModel[13] cannot discern types unless bond orders are supplied. These results suggest that IDATM is a valuable accessory program for drug design and molecular modeling in general. IDATM is available upon request from ECM.

SUMMARY

IDATM, a simple geometric algorithm for determining atom types and molecular topology, has achieved greater than 99% success in processing a wide range of organic structures. Experimental bond length data are utilized. Overall, the method is a rapid and robust way of minimizing the amount of human intervention required between molecule read-in and the implementation of higher-order molecular modeling strategies.

Acknowledgements. ECM gratefully acknowledges support from NIH grants GM-31497 (I. D. Kuntz) and GM-39552 (G. L. Kenyon); RAL would like to thank the Royal Commission of 1851 for a Research Fellowship and the Fulbright Commission for a Senior Scholarship. Thanks are due also to A. R. Leach and I. D. Kuntz for insightful discussions.

References

1. R. S. Pearlman, *Chem. Design Automation News,* **2,** 1 (1987).

2. R. S. Pearlman, A. Rusinko, J. M. Skell, and R. Balducci, 'CONCORD User's Manual,' Tripos Associates, St. Louis, 1988.

3. D. P. Dolata, A. R. Leach, and K. Prout, *J. Comput.-Aided Mol. Design,* **1,** 73 (1987).

4. A. R. Leach and K. Prout, *J. Comp. Chem.,* **11,** 1193 (1990).

5. F. H. Allen, O. Kennard, D. G. Watson, L. Brammer, A. G. Orpen, and R. Taylor, *J. Chem. Soc. Perkin Trans. II,* S1, (1987).

6. L. E. Sutton, 'Tables of Interatomic Distances and Configuration in Molecules and Ions,' *Chemical Society Special Publication No. 11,* Chemical Society, London, 1958.

7. L. E. Sutton, 'Tables of Interatomic Distances and Configuration in Molecules and Ions,' *Chemical Society Special Publication No. 18,* Chemical Society, London, 1965.

8. F. C. Bernstein, T. F. Koetzle, G. J. B. Williams, E. F. Meyer, Jr., M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi, *J. Mol. Biol.,* **112,** 535 (1977).

9. F. H. Allen, S. Bellard, M. D. Brice, B. A. Cartwright, A. Doubleday, H. Higgs, T. Hummelink, B. G. Hummelink-Peters, O. Kennard, W. D. S. Motherwell, J. R. Rodgers, and D. G. Watson, *Acta Crystallogr. Sect. B,* **35,** 2331 (1979).

10. SYBYL Version 5.3, Tripos Associates, St. Louis, MO 63117.

11. Molecular Interactive Display and Simulation, Computer Graphics Laboratory, School of Pharmacy, University of California, San Francisco, CA 94143-0446.

12. QUANTA Version 3.0, Polygen Corporation, 200 Fifth Ave., Waltham, MA 02254.

13. F. Mohamadi, N. G. J. Richards, W. C. Gulda, R. Liskamp, M. Lipton, C. Caufield, G. Chang, T. Hendrickson, and W. C. Still, *J. Comp. Chem.*, **11,** 440 (1990).

CHAPTER 2: AUTOMATED DOCKING WITH GRID-BASED ENERGY EVALUATION

Elaine C. Meng, Brian K. Shoichet, and Irwin D. Kuntz*

Department of Pharmaceutical Chemistry, School of Pharmacy, University of California, San Francisco, California 94143-0446

**To whom all correspondence should be addressed.*

ABSTRACT

The ability to generate feasible binding orientations of a small molecule within a site of known structure is important for ligand design and discovery. We present a method which combines a rapid, geometric docking algorithm with the evaluation of molecular mechanics interaction energies. The computational costs of evaluation are mimimal because the receptor-dependent terms in the potential function are precalculated at points on a three-dimensional grid. In four test cases where the components of crystallographically determined complexes are redocked, the "force field score" correctly identifies the family of orientations closest to the experimental binding geometry. Scoring functions that consider only steric factors or only electrostatic factors are less successful. Improving the evaluation function is crucial for improving the ability of the method to search databases for potential lead compounds.

INTRODUCTION

The opportunities for *ab initio* drug design are more numerous now than ever before. This can be attributed to the discovery of the molecular bases of many diseases and to progress in macromolecular structure determination. A wealth of mechanistic information exists in the atomic coordinates of a macromolecule; in addition, the detailed structure may suggest a means for altering function. Numerous drugs work by specifically binding to a receptor molecule and modulating its biological activity.

The ability to propose reasonable ways of binding a putative ligand molecule to a known receptor site is crucial to the success of structure-based design. One approach is to position or "dock" ligand and receptor molecules together in many different ways, and then "score" each orientation according to an evaluation function of some kind.

Docking methods can be subdivided into manual and automatic approaches. In manual docking, the user is responsible for positioning the molecules; this process may be interactive, with continuous feedback on the energy of the system,[1-3] or each energy determination may require a significant amount of computer time. As with any energy calculation, the time demands increase with increasing complexity of the evaluation function, increasing number of atoms, and increasing number of degrees of freedom within the system. The same considerations apply to automated docking,[4-11] in which the molecules are positioned according to algorithms that vary from exhaustive to stochastic to deterministic. Compared to manual docking, automatic methods are less dependent upon, though certainly not independent of, the preconceptions of the user regarding which areas of the receptor are most important for binding.

The complexity of configuration space for systems involving biomacromolecules leads to high computational costs, especially when realistic potential functions are used.

A balance between thermodynamic accuracy and computational tractability is desirable. A significant reduction in scoring time can be achieved by precomputing terms in the potential function which are sums over receptor atoms. As in the work of Goodford[12] and several of the docking methods,[1-3,10] receptor terms are calculated for each point on a three-dimensional grid. While this approach requires more preparation, it decreases the time needed to evaluate ligand orientations. Any expression in which the ligand and receptor terms are separable can be treated in this manner. We have chosen a set of functions and parameters that approximate the AMBER force field.[13,14] Using this approach, ligand orientations generated with the rigid-body docking algorithm of Kuntz and coworkers[5,15-17] can be ranked according to molecular mechanics interaction energy. The method is rapid and robust; test runs on known complexes suggest that approximate interaction energies are useful for discerning likely binding orientations.

COMPUTATIONAL METHODS

The major steps of the procedure are: characterization of the receptor site, calculation of grids for evaluating docked structures, docking, and evaluation of the resulting ligand orientations. Programs involved in the overall process are shown in Figure 1. The computer programs MS[18,19] and DelPhi[20,21] are distributed independently.

Site Characterization

We characterize the site as described previously.[5,15-17] The Connolly MS algorithm[18,19] is used to generate a molecular surface as defined by Richards.[22] Spheres that fill surface indentations are then calculated with the program SPHGEN.[5] Each sphere touches the surface at two points and is centered along the surface normal at one of the

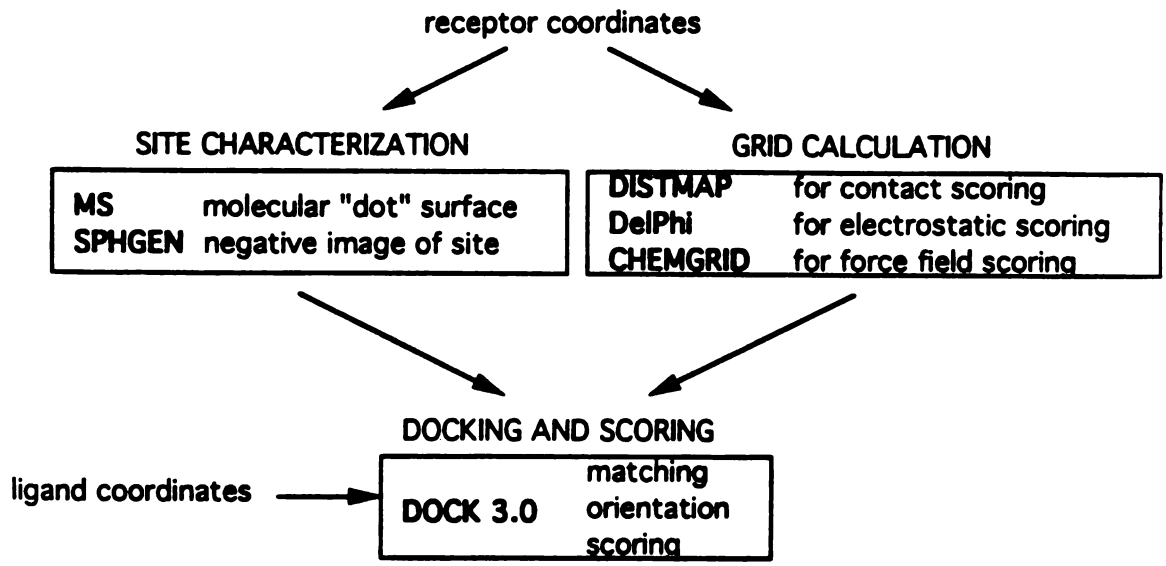


Figure 1. Programs involved in the use of DOCK. MS[18,19] and DelPhi[20,21] are distributed independently.

points. Only one sphere per surface atom, the largest that does not intersect the surface, is generally retained; groups of overlapping spheres are referred to as clusters. The cluster containing the greatest number of spheres tends to occupy the largest indentation of the surface, typically the active site of an enzyme. The user selects one or more clusters for docking.

Calculation of Grids

We use the following means of evaluating molecular complexes: contact score, electrostatic interaction energy, and molecular mechanics interaction energy.

While each option makes use of a cubic lattice, there are differences in the details of implementation. The contact grid is automatically constructed to enclose the input atoms, which may form part or all of the receptor. The electrostatic grid encloses a cubic volume, which, due to the nature of the calculation, should include the entire receptor molecule. The location, size, and resolution of these grids are under user control.

The grid used for force field calculations is also a cubic lattice, but the volume enclosed may have different x, y, and z extents since the data are stored in one-dimensional arrays. All receptor atoms are included in the calculation whether or not they fall within the grid volume. The force field grid may be positioned either by direct specification of its coordinates or by centering within a sphere cluster; in this manner, one can define a box that efficiently encloses the space that docked molecules are likely to occupy.

We next consider how each set of grid values is calculated.

The program DISTMAP[16,17] produces the grid for contact scoring. The user specifies the grid resolution, two "close contact" limits (for receptor polar and nonpolar

atoms, respectively), and a cutoff defining the range of pairwise contacts. For every receptor atom within the contact range, the sum at a grid point is incremented by one, unless a close contact limit is violated, in which case a negative number is added. Hydrogens are not included in the calculation. We note that this is a different contact score than used in the earliest versions of DOCK.[15]

The electrostatic score is an interaction energy based on potentials calculated with the DelPhi program.[20,21] DelPhi uses a finite-difference algorithm to solve the Poisson-Boltzmann equation. The resulting electrostatic potential is thought to be more realistic than those of standard force fields;[23] internal and external dielectrics of different magnitudes, nonzero ionic strength, and ion exclusion effects can be modeled.[20,21] We assume in this implementation that a suitable potential can be calculated using the receptor alone (see Discussion); that is, the potential is not recalculated in the presence of the ligand.

The program CHEMGRID produces the values for computing force field scores. These scores, or molecular mechanics interaction energies, are calculated as a sum of van der Waals and electrostatic components:

$$E = \sum_{i=1}^{lig} \sum_{j=1}^{rec} \left(\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} + 332.0 \frac{q_i q_j}{D r_{ij}} \right) \quad (1)$$

where each term is a double sum over ligand atoms i and receptor atoms j, A_{ij} and B_{ij} are van der Waals repulsion and attraction parameters, r_{ij} is the distance between atoms i and j, q_i and q_j are the point charges on atoms i and j, D is the dielectric function, and 332.0 is a factor that converts the electrostatic energy into kilocalories per mole. Eq. 1 contains the *intermolecular* terms present in the AMBER[13] molecular mechanics function, except for an explicit hydrogen-bonding term. We assume that hydrogen bond energies can largely be accounted for in the electrostatic term.[24]

Grid-based scoring can be accomplished efficiently when the ligand and receptor terms in the evaluation function are separable. This is generally true for the electrostatic part of a potential function. For the van der Waals terms, it is necessary to use a geometric mean approximation:[2,24]

$$A_{ij} = \sqrt{A_{ii}}\sqrt{A_{jj}} \quad \text{and} \quad B_{ij} = \sqrt{B_{ii}}\sqrt{B_{jj}} \tag{2}$$

where the single-atom-type parameters are calculated from van der Waals radius, R, and well depth, ε, according to:

$$A = \varepsilon\left[2R\right]^{12} \quad \text{and} \quad B = 2\varepsilon\left[2R\right]^{6} \tag{3}$$

Using this approximation, eq. 1 can be rewritten as:

$$E = \sum_{i=1}^{lig}\left[\sqrt{A_{ii}}\sum_{j=1}^{rec}\frac{\sqrt{A_{jj}}}{r_{ij}^{12}} - \sqrt{B_{ii}}\sum_{j=1}^{rec}\frac{\sqrt{B_{jj}}}{r_{ij}^{6}} + 332.0q_i\sum_{j=1}^{rec}\frac{q_j}{Dr_{ij}}\right] \tag{4}$$

Three values are stored for every grid point k, each a sum over receptor atoms that are within a user-defined cutoff distance of the point:

$$aval = \sum_{j=1}^{rec}\frac{\sqrt{A_{jj}}}{r_{jk}^{12}} \quad bval = \sum_{j=1}^{rec}\frac{\sqrt{B_{jj}}}{r_{jk}^{6}} \quad esval = 332.0\sum_{j=1}^{rec}\frac{q_j}{Dr_{jk}} \tag{5}$$

These values, with or without interpolation, may subsequently be multiplied by the appropriate ligand values to give the interaction energy.

Input to CHEMGRID includes the grid resolution, location, and dimensions, the form of the dielectric function (constant or distance-dependent), a scaling factor for the dielectric function, a nonbonded cutoff distance, and names of parameter files. Two parameter files are read during a run: a table listing charges and van der Waals (VDW) types for atoms in each of the twenty standard amino acids, and a table containing $\sqrt{A}$ and $\sqrt{B}$ for each VDW type. The receptor parameterization step employs hashing and is very rapid, typically taking less than 1% of the total grid calculation time. The present work uses AMBER united-atom parameters[13] for the receptor, with the exception that all

hydrogens bonded to noncarbon atoms are considered volumeless. We would like to emphasize, however, that it is possible to use other parameter sets without changing the code.

Docking

Orientations are generated by finding sets of ligand atoms that match sets of sphere centers, then performing a least-squares superimposition.[25] As in DOCK 1.0, sets are considered to match if their pairwise internal distances correspond, within some tolerance. Beginning with DOCK 2.0, a modification involving presorting the distances into "bins" has been employed.[16,17] This allows for more systematic searches of orientation space, and for greater user control over the thoroughness of the searches.

Scoring

For contact scoring, each ligand atom is assigned the score of the nearest point on the grid. The total score is the sum of the atomic scores.

The DelPhi-calculated potential at each ligand atom is obtained by trilinear interpolation of the values at the eight surrounding grid points. The potential is multiplied by the ligand atom point charge to give the electrostatic interaction energy, and the total energy is the sum of the atomic energies.

Force field scoring requires the retrieval of three grid values. These may be the sums corresponding to the nearest point, or the results of trilinearly interpolating the values for the eight surrounding points. Substituting eq. 5 into eq. 4, the interaction energy is:

$$E = \sum_{i=1}^{lig} \left[\sqrt{A_{ii}} \left[aval \right] - \sqrt{B_{ii}} \left[bval \right] + q_i \left[esval \right] \right] \qquad (6)$$

Atoms that fall outside the grid, if any, are given interaction energies of zero. Ligand

atoms are associated with parameters at read-in time; the present work uses AMBER all-atom VDW parameters,[14] except that hydrogens bonded to noncarbon atoms are again considered volumeless.

Test Systems and Run Parameters

Four well-determined crystallographic complexes were chosen from the Brookhaven Protein Data Bank[26,27] (Fig. 2 and Table I): 4dfr[28] (dihydrofolate reductase/methotrexate), 6rsa[29] (ribonuclease A/uridine vanadate), 2gbp[30] (periplasmic binding protein/glucose), and 3cpa[31] (carboxypeptidase A/glycyltyrosine). Different aspects of complementarity are evident in these systems, including salt bridge formation, hydrogen bonding, and hydrophobic interactions. In each case, crystallographic waters and ions were removed; the ligand and receptor were separated and hydrogens were added as necessary, in standard geometries. A partial molecular surface was calculated for the receptor, excluding roughly the one-half to two-thirds of the molecule farthest from the site of interest. This surface was used in SPHGEN, and the largest of the resulting sphere clusters was selected for docking. The DISTMAP (contact grid) calculation included atoms contributing to the molecular surface, as well as additional atoms within 5.0 angstroms of any surface atom. Polar and nonpolar close contact limits were 2.3 and 2.8 angstroms, respectively, the maximum distance for a "good" contact was 4.5 angstroms, and the contact grid spacing was 3 points per angstrom. DelPhi runs included the entire receptor, with AMBER united-atom partial charges.[13] Three-step focusing,[21] in which the protein occupied 20, 60, and then 90% of the electrostatic potential grid, was used in order to reduce any errors associated with boundary conditions. Internal and external dielectric constants were 4 and 80, respectively, the ionic strength was 0.145 M, the ion exclusion radius was 2.0 angstroms, and the probe radius was 1.4 angstroms.

Table I. Test systems.

Brookhaven[a] file	resolution[b]	receptor	complexed ligand	docked ligand, formal charge
4dfr[c]	1.7	dihydrofolate reductase	methotrexate	2,4-diamino-6-methylpteridine,[d] +1
6rsa[e]	2.0	ribonuclease A	uridine vanadate	uridine 3′-phosphate,[f] –2
2gbp[g]	1.9	periplasmic binding protein	β-D-glucose	β-D-glucose, 0
3cpa[h]	2.0	carboxypeptidase A	glycyl-L-tyrosine	glycyl-L-tyrosine, 0 (zwitterion)

[a]References 26 and 27.

[b]Angstroms.

[c]Reference 28.

[d]The inflexible part of methotrexate; see text and Figure 3.

[e]Reference 29.

[f]Built from uridine vanadate; see text and Figure 3.

[g]Reference 30.

[h]Reference 31.

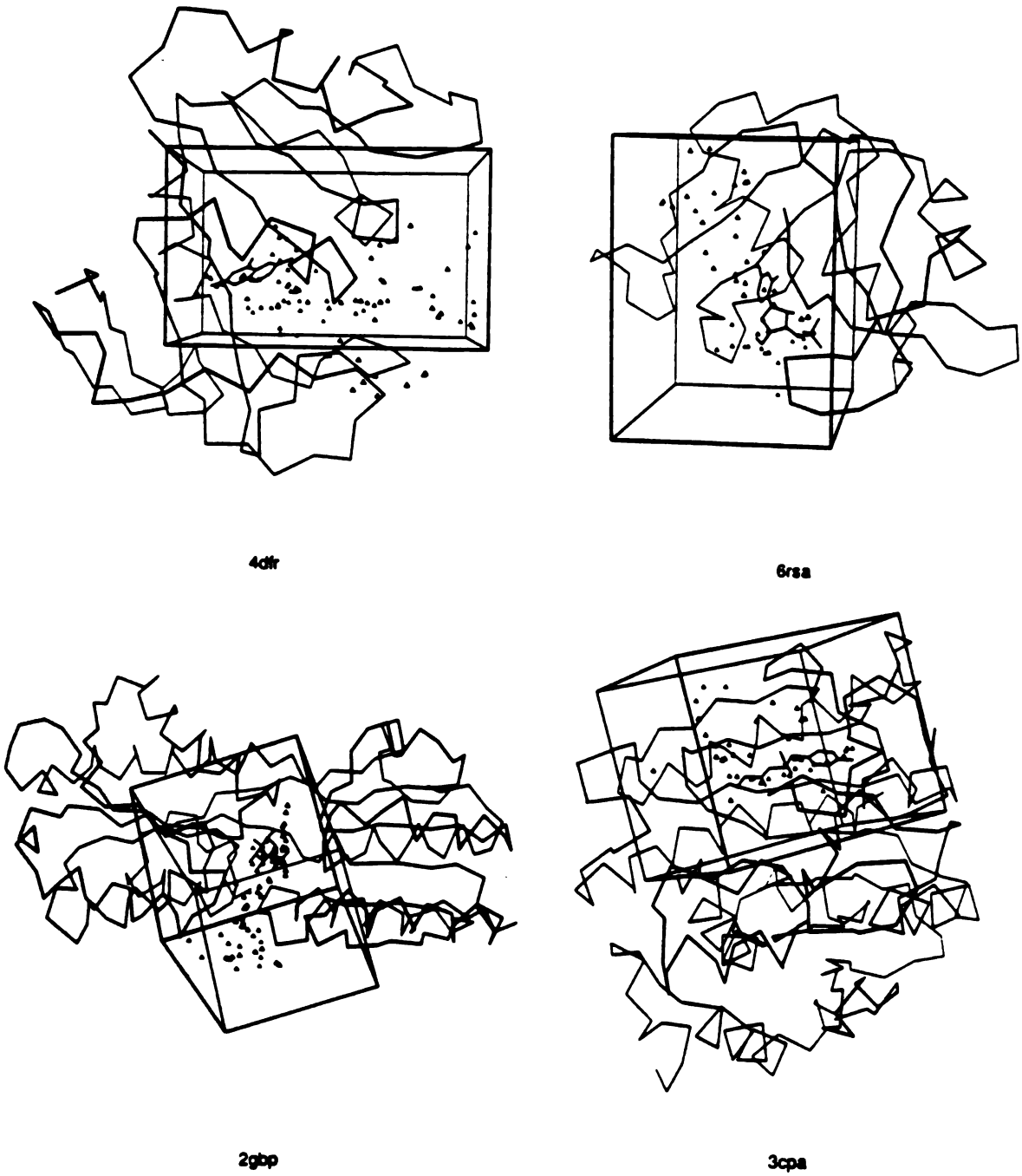


Figure 2. Test systems: Cα representations of the proteins, shown with ligands, sphere centers used for docking (triangles), and boxes outlining the force field grids. Pictures generated with UCSF MidasPlus: Molecular Interactive Display and Simulation, Computer Graphics Laboratory, Department of Pharmaceutical Chemistry, University of California, San Francisco, CA 94143-0446.

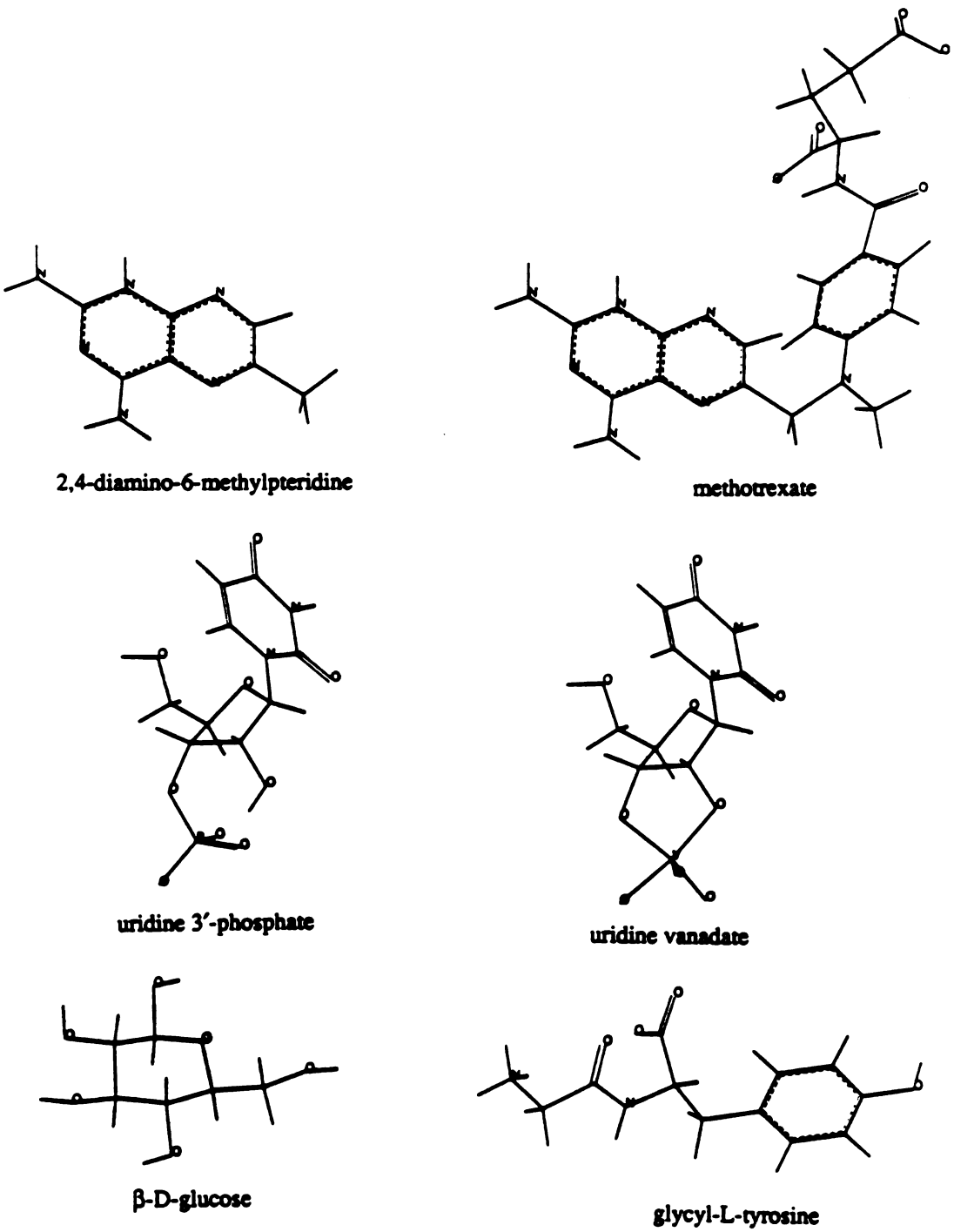


Figure 3. Test system ligands. Pictures generated with SYBYL 5.4: Molecular Modeling System SYBYL, Version 5.4, TRIPOS Associates, Inc., St. Louis, MO 63117.

Force field grids were calculated in CHEMGRID, using the entire receptor, 0.3-angstrom spacing, a 10.0-angstrom cutoff, and $D = 4r$;[32] except where noted below, the use of other parameter sets was not explored. Grid boxes, as well as the sphere centers used for docking, are shown in Figure 2. Test system ligands are shown in Figure 3.

In DOCK, distance matching parameters (Table II) were chosen such that several thousand sterically allowed orientations were found. The close contact limits set in DISTMAP determine which orientations are sterically acceptable. Only those orientations having a score greater than a user-specified cutoff were written out (see legends to Fig. 4 and Figs. 8-10), as it has been observed that orientations with very low contact scores are also unfavorably ranked by other scoring metrics. Molecular mechanics energies were obtained with trilinear interpolation of grid values. Calculations of the root-mean-square deviation (RMSD) from the crystallographic orientation did not include hydrogens.

Time requirements are given in Table III for the pre-docking calculations and in Table IV for the docking runs. All calculations were performed on Silicon Graphics IRIS 4D/25 workstations with 16 megabytes of main memory.

RESULTS

Results of the docking runs are given in Figures 4-10. The RMSD of the ligand from the crystallographically determined position is plotted versus each type of score.

In viewing the plots, it should be noted that there is no reason to expect a simple correlation between RMSD and score, since the most favorable alternative sites are not necessarily those closest to the crystallographically determined binding site; the interaction energy changes monotonically with displacement only in the immediate vicinity of a

Table II. Distance matching parameters (angstroms).[a]

Test system	receptor bin width; overlap	ligand bin width; overlap	matching tolerance
4dfr	1.0; 0.2	1.0; 0.2	1.5
6rsa	1.0; 0.5	1.0; 0.5	1.5
2gbp	1.0; 0.4	1.0; 0.4	1.5
3cpa	1.5; 0.5	1.5; 0.5	2.0

[a]Receptor site sphere-sphere distances and ligand atom-atom distances are sorted into bins before matching is done. Increasing bin width and overlap increases the number of receptor-ligand internal distance comparisons and thus the number of orientations found. The matching tolerance defines how much distances may differ while still being paired with one another. See reference 17 for a discussion of these variables.

Table III. Computational time requirements[a] for pre-docking steps.

Test system	MS[b,c]	SPHGEN	DISTMAP	CHEMGRID[d,e]	DelPhi[f]
4dfr	2:25	72:11	1:21	16:46	5:07
6rsa	1:02	16:32	1:23	18:56	4:33
2gbp	2:13	27:13	1:47	23:31	4:12
3cpa	0:56	3:37	1:02	16:42	4:40

[a]Minutes:seconds on a Silicon Graphics IRIS 4D/25 with 16 megabytes of main memory.

[b]References 18 and 19.

[c]Surface area, square angstroms: 3509 for 4dfr, 1641 for 6rsa, 2108 for 2gbp, and 724 for 3cpa.

[d]0.30-angstrom spacing, 10.0-angstrom cutoff.

[e]Number of grid points: 368,475 for 4dfr, 475,190 for 6rsa, 439,280 for 2gbp, and 279,075 for 3cpa.

[f]References 20 and 21; three-step focusing.

Table IV. Computational parameters and time requirements for docking.

Test system	spheres	atoms[b]	found[c]	written[d]	times[a] Cw[e]	C[f]	C,DE[g]	C,FF[h]
4dfr	86	13	26,121	2617	3:06	2:13	2:06	2:34
6rsa	47	21	6252	3738	4:43	2:43	2:59	3:29
2gbp	75	12	10,926	2265	5:44	4:54	6:13	5:51
3cpa	44	17	76,684	4327	22:52	20:47	22:12	22:37

[a]Minutes:seconds on a Silicon Graphics IRIS 4D/25 with 16 megabytes of main memory.

[b]Nonhydrogen atoms in ligand, those used for matching to spheres.

[c]Sterically allowed orientations found; each of these is scored.

[d]Sterically allowed orientations with contact scores greater than a user-specified cutoff.

[e]Contact scoring only, scores and coordinates written out.

[f]Contact scoring only, scores but not coordinates written out.

[g]Contact and DelPhi scoring, scores but not coordinates written out.

[h]Contact and force field scoring, scores but not coordinates written out.

local minimum. In addition, since the RMSD represents the collapse of three-dimensional information into a one-dimensional descriptor, there may be more than one orientational family having a particular approximate RMSD value. Whether or not this occurs depends on the symmetry and steric restrictiveness of the site.

Below, we consider each system and compare the abilities of the scoring functions to identify the orientations closest to the crystallographic geometry.

Dihydrofolate Reductase

N1-protonated 2,4-diamino-6-methylpteridine (Fig. 3) was chosen as the ligand for docking to dihydrofolate reductase. This is the inflexible portion of methotrexate (Fig. 3), in the protonation state believed to be important for binding.[28] Use of the entire methotrexate molecule proved to be too restrictive for testing scoring methods; relatively few orientations were generated.

STO-3G partial atomic charges[33] were calculated for the ligand and used in docking; 86 spheres were in the cluster of interest, and 2617 orientations were written out. The best (highest) contact score (Fig. 4A) corresponds to a low RMSD, though not the lowest. Other families of orientations that receive high contact scores are the 2.8-angstrom structures, which are barrel-rolled and angled slightly relative to the crystallographic orientation, the 4.0-angstrom structures, which are angled approximately 90°, and the 4.8-angstrom structures, which are flipped end-to-end (so that the 6-methyl group is pointing in the opposite direction) and angled slightly. The 9.0-angstrom and 13.0-angstrom structures do not overlap the experimental orientation. While the 2.8-, 4.0-, and 4.8-angstrom structures are also reasonable according to the DelPhi electrostatic score, the force field score, and the electrostatic component of the force field score, members of the lowest-RMSD family receive the best scores (Fig. 4, B-D).

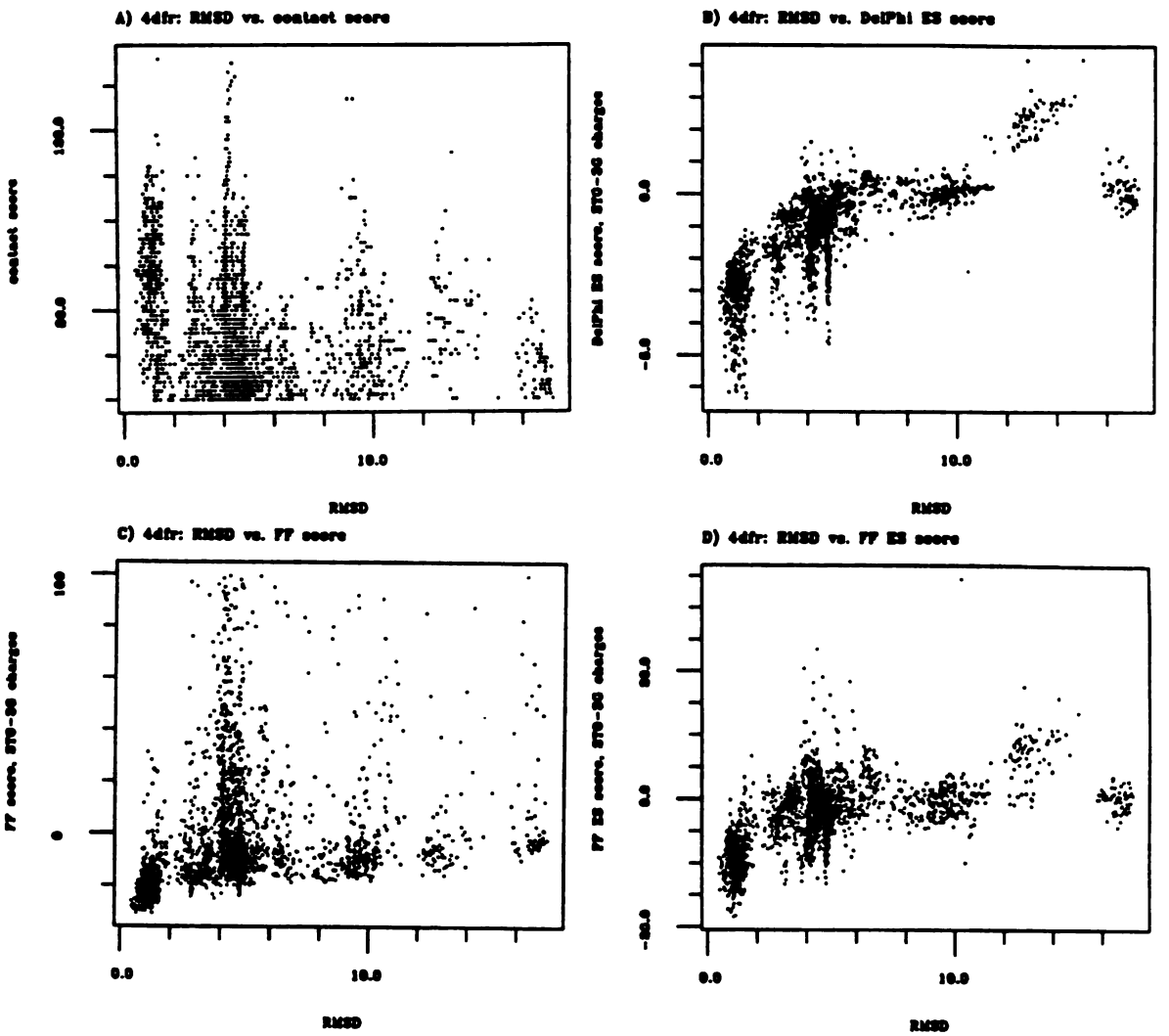


Figure 4. 4dfr test case, using STO-3G charges: RMSD versus score. A) contact score, all 2617 orientations with scores of 60 or greater. B) DelPhi score, all 2617 orientations. C) force field score, the 2404 orientations with energies below 100.0 kcal/mol. D) electrostatic component of the force field score, all 2617 orientations.

The use of a somewhat coarser grid, two points per angstrom, produced very similar results (Fig. 5, A and B), the main effect being an increase in VDW energy for several orientations due to the interpolation approximation. Likewise, using an "infinite" cutoff distance for interactions did not change the output appreciably (Fig. 5, C and D).

Two additional charge sets were generated for the ligand, using the Gasteiger-Marsili[34-36] and Gasteiger-Hückel[34-38] options in SYBYL 5.4.[39] These methods are connectivity-based (independent of conformation) and much faster than molecular orbital calculations. For this test system, the Gasteiger-Marsili results (Fig. 6) are very similar to the STO-3G results (Fig. 4, B-D); there is a reasonable distinction in score between the lowest-RMSD family and other orientations. The Gasteiger-Hückel charges are less successful in this respect (Fig. 7), although members of the lowest-RMSD family again have the best molecular mechanics scores. The solely electrostatic scores using these charges (Fig. 7, B and C) suggest that many orientations with RMSD's close to 3.0 angstroms are at least as favorable electrostatically as the lowest-RMSD structures.

Ribonuclease A

Uridine 3′-phosphate (Fig. 3) was chosen as the ligand for docking to ribonuclease A, so that AMBER all-atom charges[14] could be used. This molecule was constructed from the crystallographic ligand, uridine vanadate (Fig. 3), by changing atom types as necessary and optimizing the phosphate geometry with the Tripos force field.[39]

The cluster for docking contained 47 spheres, and 3738 orientations were written out. The RMSD values are somewhat more diffusely distributed than in the dihydrofolate reductase test case (compare Figs. 4 and 8). Of the eight highest contact scores, six correspond to the lowest-RMSD family of orientations (Fig. 8A); the other two correspond to 9.0- and 10.8-angstrom structures which are translated relative to the

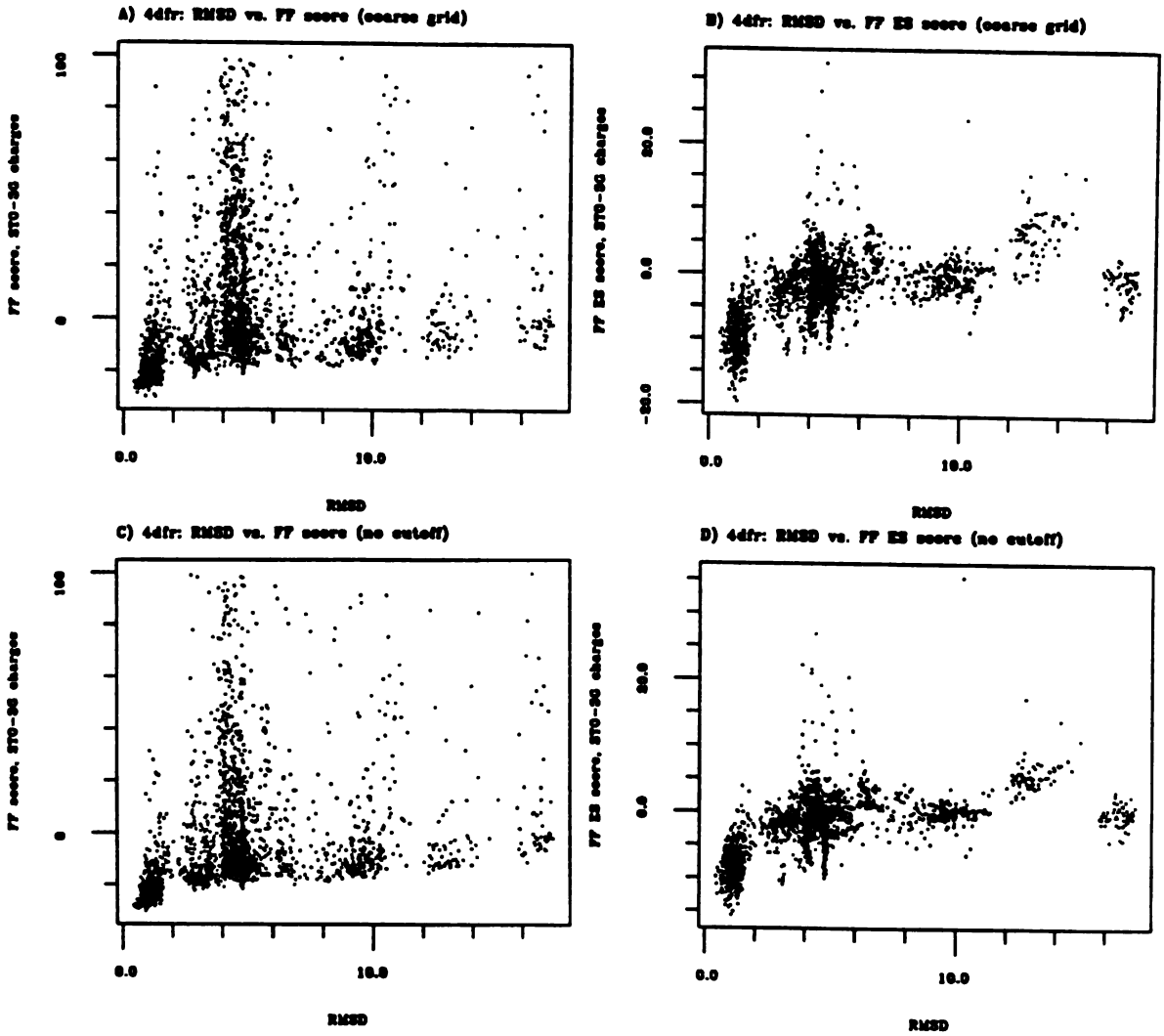


Figure 5. 4dfr test case, using STO-3G charges: RMSD versus score. A) force field score using a coarse grid, the 2288 orientations with energies below 100.0 kcal/mol. B) electrostatic component of the force field score using a coarse grid, all 2617 orientations. C) force field score using an infinite cutoff, the 2401 orientations with energies below 100.0 kcal/mol. D) electrostatic component of the force field score using an infinite cut-off, all 2617 orientations.

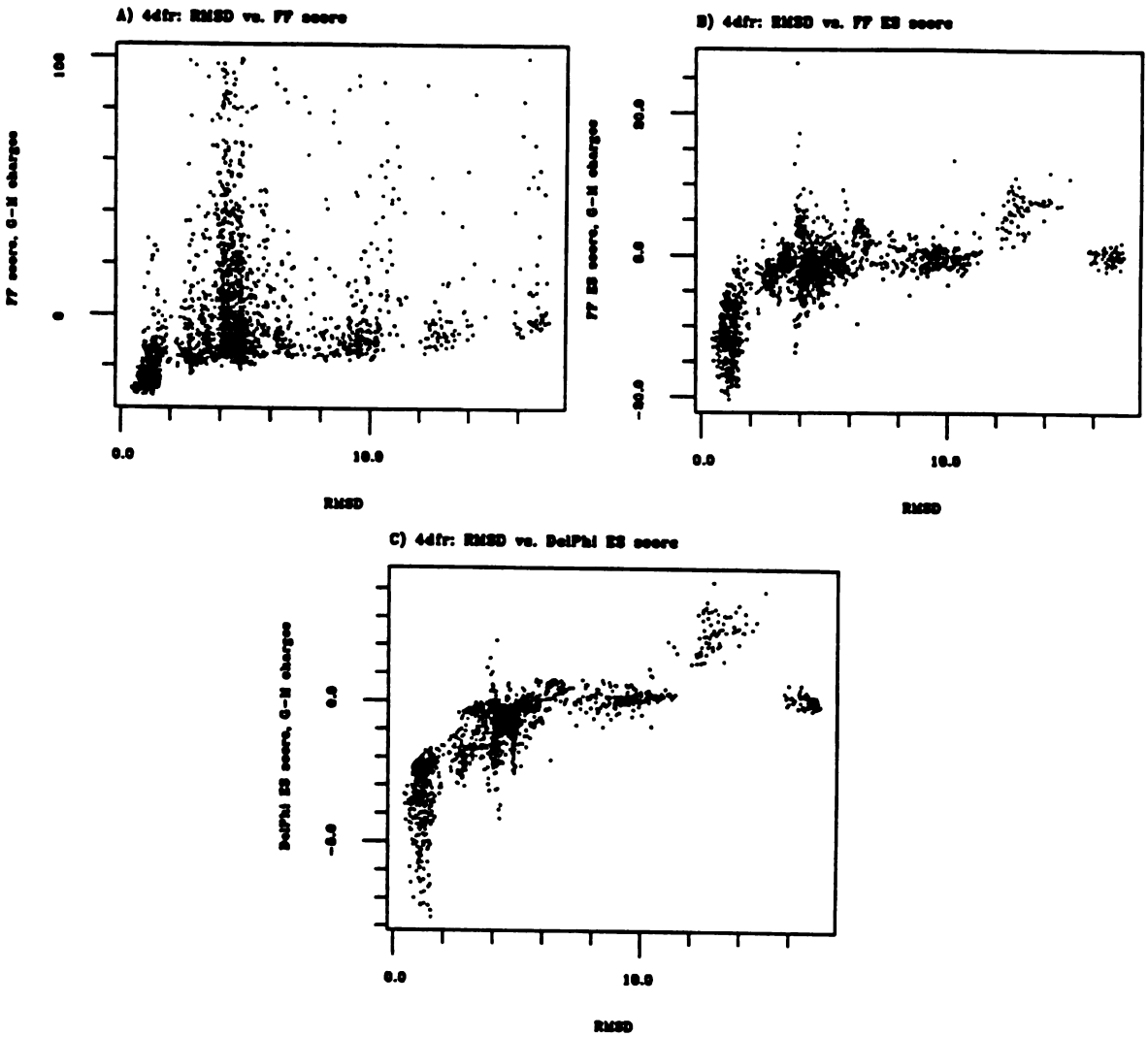


Figure 6. 4dfr test case, using Gasteiger-Marsili charges: RMSD versus score. A) force field score, the 2400 orientations with energies below 100.0 kcal/mol. B) electrostatic component of the force field score, all 2617 orientations. C) DelPhi score, all 2617 orientations.

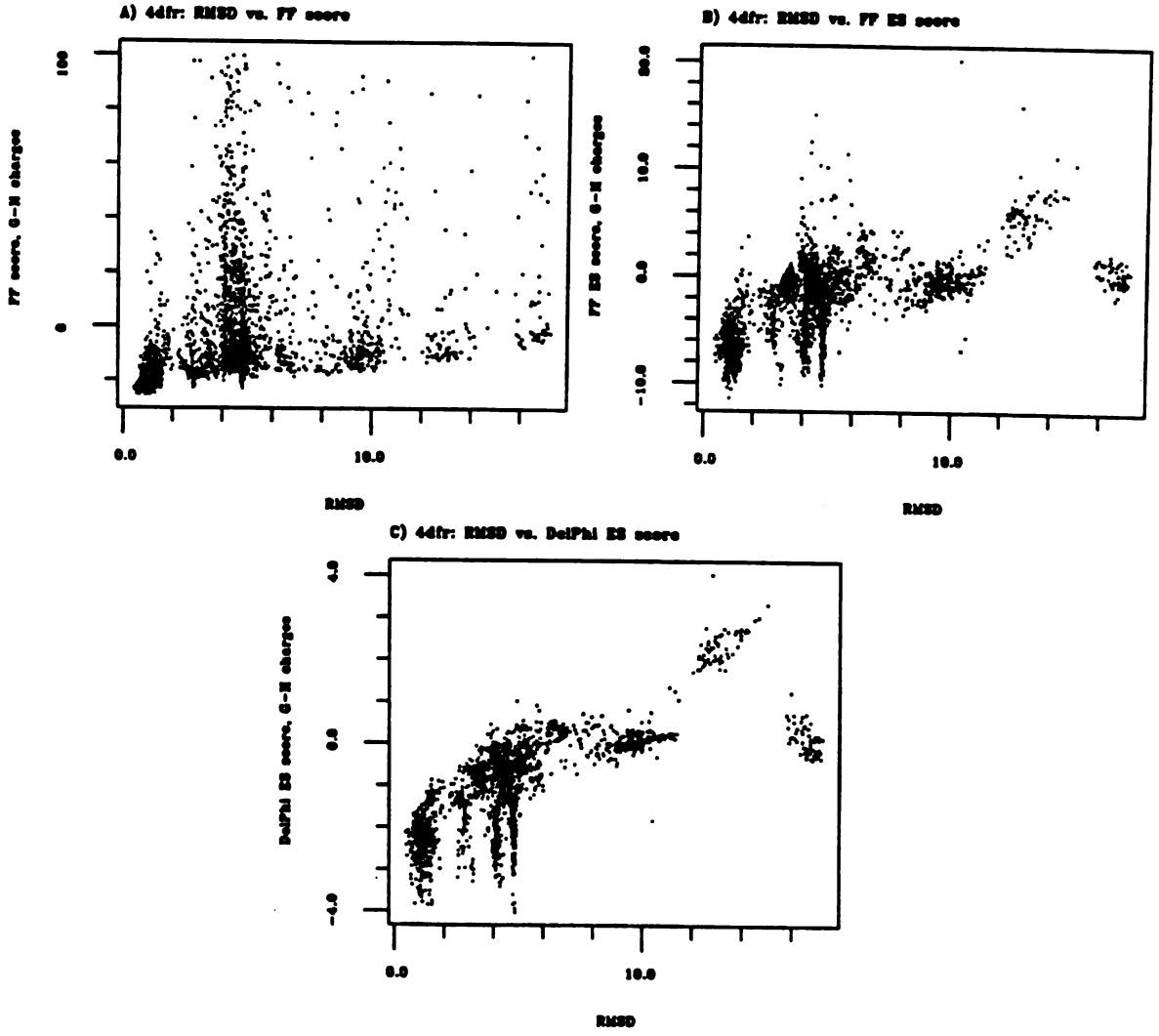


Figure 7. 4dfr test case, using Gasteiger-Hückel charges: RMSD versus score. A) force field score, the 2403 orientations with energies below 100.0 kcal/mol. B) electrostatic component of the force field score, all 2617 orientations. C) DelPhi score, all 2617 orientations.

experimental orientation, but in opposite directions from one another. If the uracil portion is considered the "head" and the ribose is considered the "tail" of the molecule, each of the eight orientations has the same general head-to-tail alignment as the crystal structure ligand. Numerous dockings with opposite head-to-tail alignments were also found; for example, a 6.2-angstrom structure with a contact score of 126 overlaps the true orientation almost completely, but the ribose and uracil positions are switched.

The force field and DelPhi scores (Fig. 8, B and C) are able to distinguish the lowest-RMSD dockings from other orientations, although by fewer than 10.0 kcal/mol. The highest-ranking alternative structures have RMSD's of 4.0-6.0 angstroms and place the phosphate essentially correctly, with the rest of the molecule angled 60-90° relative to the crystallographic orientation. The 10.0-13.0-angstrom structures with favorable scores in Figure 8, B-D are approximately related to the known orientation by a plane of reflection; the true and image phosphates face each other through the nitrogen of a nearby lysine side chain. Such results are evidence of the weight placed upon charge-charge interactions by the scoring function, especially in this test case where the ligand bears a net charge of –2. However, the total force field score is more helpful in discerning the correct binding mode than the electrostatic component alone, which favors a 4.4-angstrom structure (Fig. 8D).

Virtually indistinguishable results were obtained using Gasteiger-Marsili and Gasteiger-Hückel charges for uridine-3′-phosphate (data not shown).

Periplasmic Binding Protein

The complex of periplasmic binding protein and glucose was expected to be a relatively difficult test case. Glucose (Figure 3) bears no net charge and is roughly an oblate ellipsoid. Thus, neither charge nor shape will strongly differentiate among the various

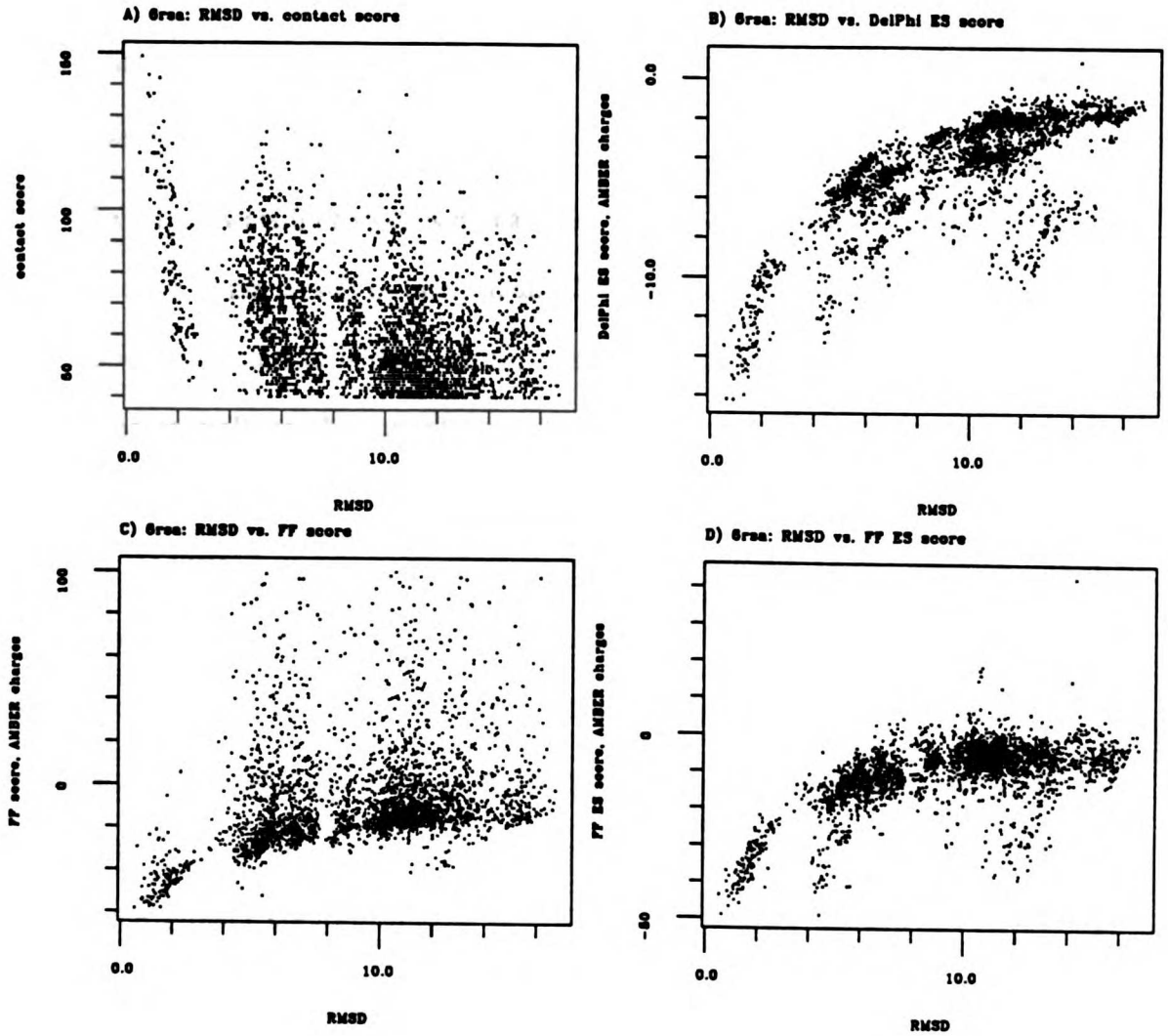


Figure 8. 6rsa test case, using AMBER charges: RMSD versus score. A) contact score, all 3738 orientations with scores of 40 or greater. B) DelPhi score, all 3738 orientations. C) force field score, the 3489 orientations with energies below 100.0 kcal/mol. D) electrostatic component of the force field score, all 3738 orientations.

orientations possible in the context of the site. In addition, periplasmic binding protein has a high affinity for the α- and β-anomers of D-glucose and D-galactose. The structure of the site suggests that any one of these four isomers can participate in thirteen hydrogen bonds with the receptor.[30]

Gasteiger-Marsili charges were calculated for β-D-glucose and used in docking; 75 spheres were in the cluster of interest, and 2265 orientations were written out. There are three obvious clusters of RMSD values, corresponding to a family of dockings very similar to the crystallographic orientation, a group of structures with RMSD's of 3.0-5.0 angstroms, and a group with RMSD's greater than 10.0 angstroms (Fig. 9). The intermediate RMSD's correspond to orientations that overlap the crystal structure ligand but are flipped or rotated in several different ways. The high RMSD's correspond to structures located in either end of the tunnel that traverses the protein (Fig. 2); apparently, there are constrictions that prevent sterically acceptable dockings from being distributed evenly throughout the tunnel. We note that this may pose a problem for methods in which the protein conformation is held constant and the ligand is moved through a representation of real space; a large energy barrier must be surmounted to reproduce the known geometry of the complex.

The simple contact score (Fig. 9A) favors orientations in the correct region of space over the end-of-the-tunnel dockings, but the highest rankings go to 3.0-angstrom structures. As expected, the electrostatic scores alone are not helpful in identifying the lowest-RMSD dockings. The DelPhi score (Fig. 9B) suggests that 3.0-4.0-angstrom structures are the most favored, while the electrostatic component of the force field score (Fig. 9D) does not clearly distinguish the orientational families from one another; the docking favored by this measure has an RMSD of 11.4 angstroms. Only the total force

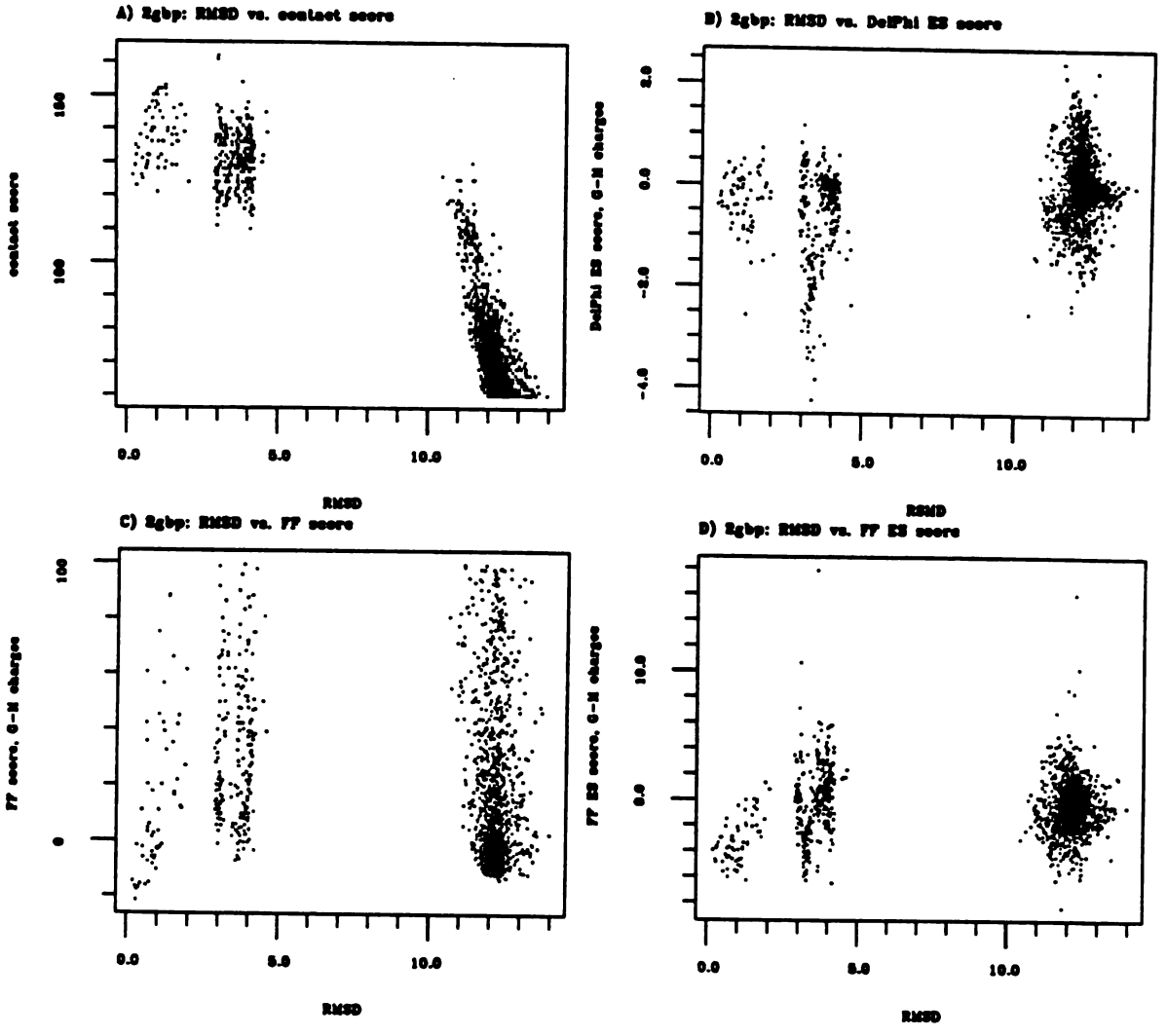


Figure 9. 2gbp test case, using Gasteiger-Marsili charges: RMSD versus score. A) contact score, all 2265 orientations with scores of 60 or greater. B) DelPhi score, all 2265 orientations. C) force field score, the 1680 orientations with energies below 100.0 kcal/mol. D) electrostatic component of the force field score, all 2265 orientations.

field score is successful in identifying structures with RMSD's below 1.0 angstrom; in fact, such structures receive the eight best scores, ranging from -21.5 to -14.6 kcal/mol (Fig. 9C).

Carboxypeptidase A

In the carboxypeptidase A test case, it is not possible to reproduce the experimental complexation geometry exactly without decreasing the close contact limits or allowing them to be violated. The phenolic oxygen of the ligand, glycyl-L-tyrosine (Figure 3), is 2.55 angstroms from the Cα of glycine-253, violating the 2.8-angstrom limit for receptor carbons, and a carboxylate oxygen is 2.23 angstroms from the phenolic oxygen of tyrosine-248, violating the 2.3-angstrom limit for receptor polar atoms. Because a ligand atom receives the attributes of the nearest grid point, however, it is possible for an acceptable orientation to violate the limits by as much as 0.866 × (contact grid spacing), or 0.29 angstroms in the present work.

AMBER all-atom charges[14] were used for glycyl-L-tyrosine. There were 47 spheres in the cluster of interest, and 4327 orientations were written out. Structures with RMSD's below 2.0 angstroms describe essentially the experimental binding mode. Relative to the crystallographic orientation, dockings with RMSD's just above 2.0 angstroms are angled slightly, 3.0-4.0 angstrom structures are barrel-rolled and translated approximately a bond length along the long axis of the molecule, and structures with RMSD's greater than 6.0 angstroms are flipped end-to-end. The lowest RMSD, 0.4 angstroms, corresponds to a structure whose phenolic oxygen is 2.61 angstroms from the Cα of glycine-253 and whose carboxylate oxygens are 2.49 and 4.74 angstroms from the phenolic oxygen of tyrosine-248. This orientation received the second-best force field score. The best contact score corresponds to a member of the lowest-RMSD family, but

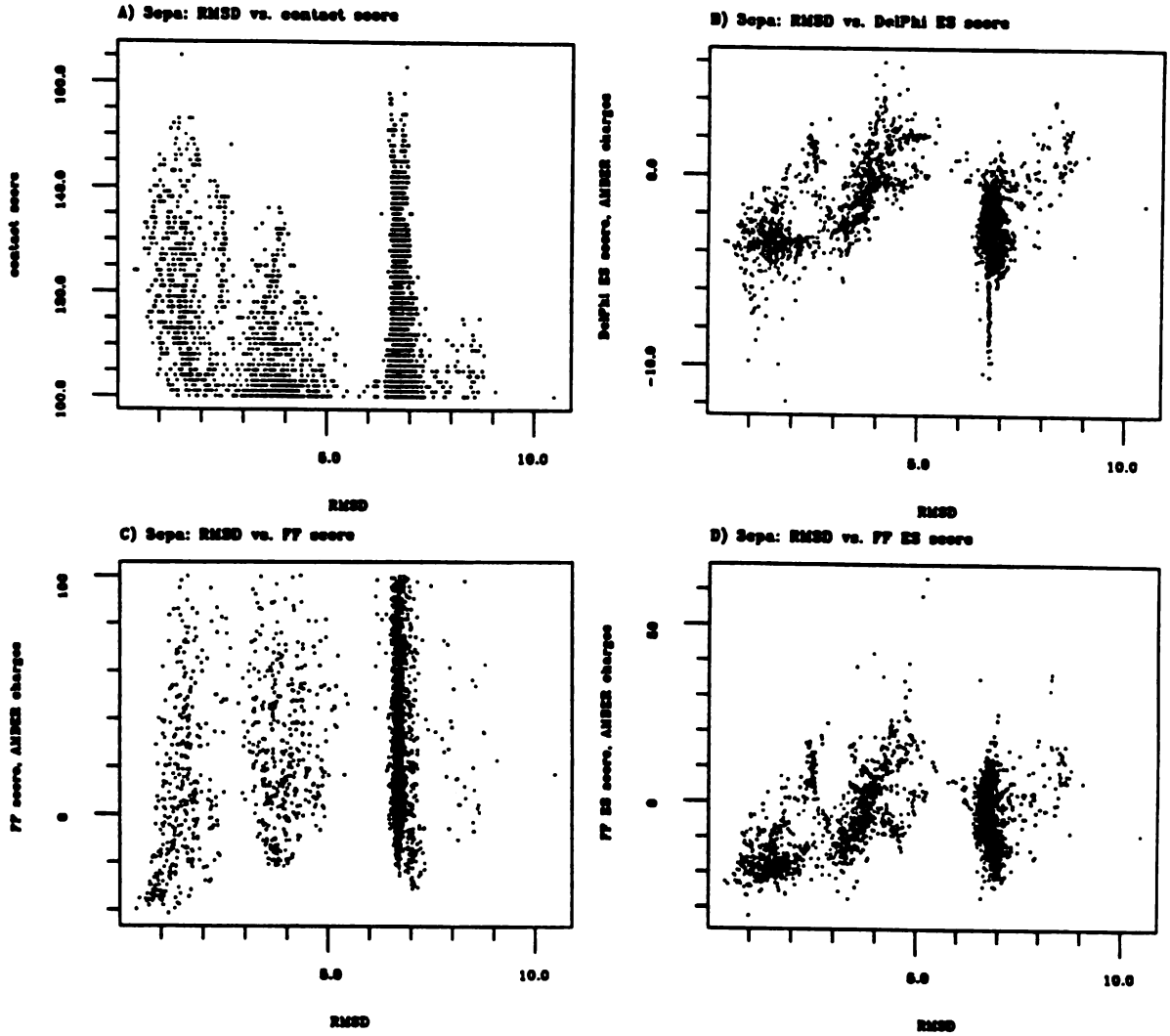


Figure 10. 3cpa test case, using AMBER charges: RMSD versus score. A) contact score, all 4327 orientations with scores of 100 or greater. B) DelPhi score, all 4327 orientations. C) force field score, the 2684 orientations with energies below 100.0 kcal/mol. D) electrostatic component of the force field score, all 4327 orientations.

good scores are also given to several of the end-to-end flipped structures (Fig. 10A). The same could be said of the DelPhi scores (Fig. 10B). Using these scoring methods, the best values are outliers. The force field score, however, clearly selects a low-RMSD cluster (Figure 10C). The 45 dockings with the best force field scores, ranging from -41.3 to -31.6 kcal/mol, all have RMSD's below 2.0 angstroms. The electrostatic component of the force field score is mainly leveling, although a 1.0-angstrom structure is favored (Figure 10D).

Force field score results were essentially the same using Gasteiger-Marsili and Gasteiger-Hückel charge sets for glycyl-L-tyrosine (data not shown).

Summary

In Table V we summarize the results of the docking calculations. For each evaluation function, the score of the experimental complexation geometry is compared with the best value found, and the RMSD of the best-scoring orientation is given.

Several features are worth noting. First, only the molecular mechanics score is successful in identifying orientations close to the crystallographic result in all four of the test cases. The other functions favor alternative modes in one or more of the test cases, and in general, favor dockings with larger RMSD's. Second, the force field identification of a family of orientations near the experimental structure is quite robust. This family is well represented in the set of best scores; depending on the test case, the top 8-112 force field scores correspond to members of the lowest-RMSD family (Table VI). Table VII lists the average RMSD's for the ten best orientations according to contact score and force field score. With one exception, the average RMSD is lower for the configurations favored by the force field score than for those favored by the contact score. The ten best dockings of glucose into periplasmic binding protein, according to the force field score,

Table V. Comparison of crystallographic and best-scoring orientations.

	contact		force field (FF)		FF electrostatic		DelPhi electrostatic	
System:	score	RMSD[a]	score[b]	RMSD[a]	score[b]	RMSD[a]	score[b]	RMSD[a]
4dfr:[c] crystal orientation	81	–	–29.085	–	–11.501	–	–3.001	–
best-scoring orientation	136	1.42	–30.829	0.64	–18.456	1.04	–6.347	1.50
6rsa:[d] crystal orientation	134	–	–61.389	–	–42.446	–	–13.719	–
best-scoring orientation	149	0.65	–58.429	0.56	–49.108	4.42	–16.214	0.92
2gbp:[e] crystal orientation	130	–	–16.995	–	–4.452	–	–0.234	–
best-scoring orientation	162	3.01	–21.483	0.29	–8.397	11.81	–4.252	3.34
3cpa:[d] crystal orientation	94[f]	–	–25.167[g]	–	–19.640	–	–3.565	–
best-scoring orientation	165	1.52	–41.302[h]	1.16	–32.418	0.98	–11.914	1.86

[a]Angstroms, relative to crystal structure orientation.

[b]Kcal/mol.

[c]STO-3G charges for ligand.

[d]AMBER all-atom charges for ligand.

[e]Gasteiger-Marsili charges for ligand.

[f]Does not include the scores of two atoms that violate the close contact limits set in DISTMAP; see text.

[g]AMBER minimization with a 1000.0-angstrom cutoff, allowing only the ligand atoms to move, results in an interaction energy of –57.254 kcal/mol and an RMSD of 0.45 angstroms relative to the unminimized crystal structure orientation.

[h]AMBER minimization with a 1000.0-angstrom cutoff, allowing only the ligand atoms to move, results in an interaction energy of –54.599 kcal/mol and an RMSD of 0.94 angstroms relative to the unminimized crystal structure orientation.

Table VI. Separation of best-scoring orientational families in force field score and rank.

System:	Δ [force field score][a] (kcal/mol)	Δ [rank][a]
4dfr:[b]	5.8	113
6rsa:[c]	6.2	23
2gbp:[d]	7.7	9
3cpa:[c]	10.3	46

[a]Difference between the top-ranked orientations of the two best-scoring families.
[b]STO-3G charges for ligand used to calculate force field score.
[c]AMBER all-atom charges for ligand used to calculate force field score.
[d]Gasteiger-Marsili charges for ligand used to calculate force field score.

Table VII. Average RMSD's for the ten best-scoring orientations.

	type of score	
System: average RMSD[a] (range[a])	contact	force field
4dfr:[b]	5.0 (1.4–9.2)	0.9 (0.5–1.3)
6rsa:[c]	2.8 (0.7–10.8)	1.1 (0.6–1.6)
2gbp:[d]	1.7 (0.9–3.8)	2.8 (0.2–12.3)[e]
3cpa:[c]	6.1 (1.5–6.9)	1.0 (0.4–1.4)

[a]Angstroms, relative to crystal structure orientation.
[b]STO-3G charges for ligand used to calculate force field score.
[c]AMBER all-atom charges for ligand used to calculate force field score.
[d]Gasteiger-Marsili charges for ligand used to calculate force field score.
[e]The average and range for the eight best force field scores are 0.4 and 0.2–0.7 angstroms, respectively.

have a relatively high average RMSD; although the best eight have RMSD's less than 1.0 angstrom, the ninth- and tenth-ranked orientations have RMSD's greater than 12.0 angstroms. Third, it is apparent that the experimental orientation does not necessarily receive the best score. This is to be expected for the contact score, which is very simple and does not include electrostatics, and for the solely electrostatic scores, which do not include van der Waals energies. For example, a docked ligand that interacts optimally with charges on the receptor often approaches receptor atoms too closely. In addition, the force field score is simplistic and involves numerous assumptions and approximations (see Discussion). There are factors, however, which could cause even a perfect score to favor "incorrect" orientations: uncertainties in the experimental atomic positions, and uncertainties in the positions of hydrogens added to the structures. Fourth, alternative binding modes with relatively good force field scores are found in each case. These are inherently plausible and may be worth considering in ligand design efforts.

DISCUSSION

The generation of feasible complexation geometries is important at more than one level to the process of structure-based drug design. Above, we have addressed the identification of the preferred mode of binding of a specific conformation of a ligand. Only with this information can one suggest structural modifications intended to form, enhance, or disrupt specific interactions with the receptor. Each cycle of structure-based design requires a model of the binding geometry, which may or may not be falsified during a later cycle. A further application of automated docking is the discovery of ligands by searching through databases of molecules. To be useful in either task, a docking program should: 1) adequately sample the possible configurations of a ligand-receptor sys-

tem, 2) score each configuration accurately, and 3) operate in a reasonable amount of time.

We will discuss each of these issues as they pertain to the present work. We will then consider the limitations of our method and compare it to other approaches to the docking problem.

Sampling Configuration Space

DOCK was designed to sample the six degrees of freedom involved in the relative placement of two rigid three-dimensional objects. We find that of the few thousand configurations that are examined for each test case, tens to hundreds of orientations are close to the experimental result (RMSD's under 1.0 angstrom). This suggests that, in these systems where the ligand and receptor conformations are known, configurational sampling has been performed adequately.

We do not mean to imply that conformational issues are unimportant. In fact, prediction of the conformations of ligand and receptor may be a limiting aspect of the procedure. We have addressed internal degrees of freedom by docking and then linking rigid fragments,[40] by adding multiple conformations of molecules to databases for searching, and by combining docking with conformational search strategies.[41]

Scoring

A primary focus of this paper has been the comparison of methods for assessing orientations of a ligand within a receptor site. We have found that in four diverse systems, a molecular mechanics function consistently identifies configurations resembling the crystal structure of the complex while the other functions tested do not.

Contact scores. A simple contact score, as implemented in DOCK version 2.0,[16,17] is used as a measure of shape complementarity. The highest contact scores are associated with low-RMSD dockings in three out of the four cases. However, the number of low-RMSD/high-score orientations and their separation in score from alternative modes are smaller for contact scores than for the other options tested. These problems are most severe when the ligand has a roughly symmetric shape, such as that of a cylinder (glycyl-L-tyrosine) or oblate ellipsoid (β-D-glucose). In any case, shape fitting is only a part of molecular recognition; even a perfect measure of shape complementarity cannot, in general, be expected to identify the preferred geometry of a ligand-receptor complex. We conclude that contact scoring is most useful for discarding orientations that overlap receptor atoms, and for finding templates for lead compounds. In this view, the docked structures are frameworks with no real chemical sense. Although good steric complementarity is crucial, one must design in the proper atom types and functional groups to convert a template into a lead compound.

Electrostatic scores. We have used two different functions to calculate electrostatic potentials: a simple Coulombic form with a distance-dependent dielectric function, and the linearized Poisson-Boltzmann equation, solved numerically with a finite-difference algorithm as implemented in DelPhi.[20,21] In addition, both quantum-mechanical and connectivity-based partial charge sets have been evaluated.

To a first approximation, the electrostatic scoring options are equally successful when the ligand bears a formal charge. When the ligand is not formally charged, differences are more evident, in part because the connectivity-based charges tend to be smaller than those derived quantum-mechanically.

For calculating the Coulombic electrostatic potential, we feel that the use of $D = 4r$ is reasonable in the absence of explicit solvent.[32] The Coulombic term alone is less helpful than the total force field score, however, being comparable in this respect to the DelPhi score. As expected, the best results are obtained when both sterics and electrostatics are taken into account.

It must be stressed that the way in which we have computed DelPhi electrostatic interaction energies is not rigorously correct, and that using different parameters for calculating the potential map may affect the results. A more rigorous application of DelPhi to calculate electrostatic interaction energies in solution has been described,[42] and involves evaluation of a full thermodynamic cycle including the bound and unbound states of the molecules. This requires the DelPhi program to be run for each ligand-receptor geometry, an option that is not feasible in our application. The practice of calculating a potential map for the receptor alone and then multiplying ligand point charges by the local potentials leads to underestimation of favorable interactions; solvent exclusion by the ligand and dipoles induced upon binding are not modeled.[43] It may be helpful to approximate solvent exclusion by considering the spheres to be regions of low dielectric when calculating the receptor potential (B. K. Shoichet, unpublished results).

Force field scores. Of the options investigated, the force field score is the most successful in identifying ligand-receptor configurations that resemble the experimental geometry (Table V). Several points deserve mention. First, while each DOCK run produced numerous low-RMSD orientations, these receive a wide range of force field scores due to the sensitivity of the 6-12 potential to small displacements. This is not a limitation in practice, as only the most favorable orientations are kept for further study.

Second, we find that the grid-based scoring preserves, to a large extent, the rank-ordering of orientations yielded by continuum (nongrid) calculations. In order to examine the effects of the grid approximation, interaction energies calculated within the AMBER analysis module were compared to grid-based interaction energies (4dfr test system; results not shown). Using grid spacings of 0.5 angstroms or less and trilinear interpolation, electrostatic interaction energies were reproduced to within a kcal/mol. Net favorable VDW energies were also matched reasonably well (within a few kcal/mol); however, net unfavorable (positive) VDW energies differed by as much as several thousand kcal/mol. This is not surprising, as the VDW potential surface rises steeply as contacts become too close. Interpolation preserves the rankings of orientations found with the continuum calculations better than does simply using the values for the grid point nearest to each ligand atom. Overall, it is apparent that the grid approximation does not degrade the results for the most favorable orientations, which are also the most interesting and the most important for the success of the method. The 4dfr trials suggest that as long as the grid spacing is reasonably small and the cutoff reasonably large, variations in these parameters do not significantly alter the results (compare Figs. 4 and 5).

Third, for all cases and for all scoring methods but one, the DOCK procedure found structures that scored better than the crystallographically determined position of the ligand. Although the best-scoring orientation is typically within an angstrom or so of the experimental orientation, this result merits some discussion. Simply stated, free energy determines the binding, but is not completely represented by the scoring functions we have used. As described in the Computational Methods section, the contact score is a rough measure of shape complementarity; charge-charge interactions are not considered. Conversely, the DelPhi score and the electrostatic component of the force field score do

not include steric contributions. The total force field score, while combining these two important aspects of molecular recognition, represents (at best) an estimate of the enthalpy of interaction. The calculation of entropy, and thus of free energy, requires sampling of a statistical ensemble of system configurations, as may be obtained through molecular dynamics or Monte Carlo simulations. Furthermore, a rigorous representation of events in solution must include explicit solvent molecules, and calculation of the free energy of binding requires evaluation of the unbound as well as of the bound structures. These considerations apply to any molecular mechanics study of complexation energetics. Approximations in addition to those of standard force field calculations include discretization of space (the grid approximation) and neglect of intramolecular terms.

There are limitations inherent in the structure of the receptor as well as in the scoring functions. Any crystallographic study yields a structure that contains the effects of static and thermal disorder. Only average atomic positions can be derived from the diffraction data, and these do not necessarily match the coordinates for any single molecule within the crystal lattice. Slight bias may result from the use of potential energy terms during refinement. Finally, the placement of hydrogens in "standard geometries" introduces some uncertainty.

For the reasons stated above, it is important not to overinterpret the scores. Even when the units returned are kcal/mol, the results cannot be converted into absolute or relative binding affinities. For this reason, we prefer to call the calculated quantities "scores" rather than "energies."

Time Requirements

The calculations necessary for docking studies, as described above, can be divided into two phases. The first phase includes calculation of a molecular surface, generation

of site-filling sphere clusters, and the creation of grids for scoring. These steps are done once per receptor, and in our systems, took approximately one hour on a Silicon Graphics IRIS 4D/25 (Table III). The time spent calculating the force field grids depends not only on the number of grid points, but also on the cutoff distance, the total number of receptor atoms, the number of receptor atoms within the cutoff distance of each grid point (that is, the shape of the receptor and the location of the grid box), and whether the dielectric function is distance-dependent or constant. For example, calculating a grid for the 4dfr test case using an "infinite" cutoff took 133 minutes and 5 seconds, nearly eight times as long as the analogous calculation using a 10.0-angstrom cutoff (Table III). In contrast, keeping the cutoff at 10.0 angstroms but increasing the spacing of points from 0.3 angstroms to 0.5 angstroms resulted in a calculation time of only 4 minutes and 8 seconds.

The second phase of the process is docking itself. The times spent in DOCK (Table IV) depend strongly on the size of the ligand, the number of spheres used, and the distance matching tolerances. Notably, the penalty for performing force field or DelPhi scoring in addition to contact scoring is relatively small. Thus, the time costs of more sophisticated calculations can, in part, be shifted to the pre-docking stage and traded for increased usage of physical memory.

Limitations

It is important to point out that the method described here does not address internal degrees of freedom,[1,10,40] pay special attention to hydrogen bonds,[3,10,11] keep track of surface area burial[1,7,11] or solvation energy,[4] or include energy minimization.[1-4,9,10] These capabilities are present to various extents in some of the other docking algorithms, albeit at a computational cost.

Our goal in ligand design applications has been to find lead compounds in an efficient way, rather than trying to find every molecule in the database that might bind to the receptor. Some potential leads may be missed because they are in the wrong conformation for binding.

A related problem is the estimation of degrees of freedom lost upon complexation and their contribution to the free energy of binding. Novotny and coworkers have approximated changes in conformational entropy upon formation of antibody-antigen complexes.[44] This approach could be generalized to provide correction terms for each "ligand" in a database (A. R. Leach and B. K. Shoichet, unpublished results).

Within the molecular mechanics formalism, hydrogen bonds can be treated as special entities, or as an important subset of the primarily electrostatic interactions. We have chosen the latter. In AMBER, the 10-12 potential contributes very little to hydrogen bond energies; it exists mainly to fine-tune hydrogen bond geometries.[13]

Although surface area burial and related solvation energy calculations[45] have proven useful in identifying misfolded protein structures,[45,46] our experience suggests they are less helpful for docking studies.[16]

Full energy-minimization of docked complexes requires parameterization of each "ligand" molecule. This is difficult when large numbers of compounds are to be evaluated. Some docking methods[2-4] employ rigid-body minimization, in which there are no intramolecular degrees of freedom; only nonbonded parameters are required. Although minimization is useful for finding local optima, it adds to the costs of computation. Docking alone can be much faster, but is prone to missing optimal geometries. The algorithm presented here is functionally equivalent to rigid-body mimimization from

multiple starting configurations, as long as the sampling of orientations within the site is sufficiently dense. When thousands of orientations per molecule are produced, as in the test cases above, each is related to many others by very slight rigid-body movements. The optimum can be singled out according to score.

Comparison to Other Methods

The force field grid is not conceptually novel. Goodford's GRID program calculates interaction energies for probes of several types at grid points within a site.[12] Most similar to our method, however, are the programs which use force field grids to evaluate docked structures. We will first address interactive docking algorithms. Pattabiraman *et al.* use the geometric mean approximation and the same evaluation function that we use, allowing for either a distance-dependent or constant dielectric function;[2] an advantage of their approach is that the grid resolution may differ for the electrostatic and steric parts of the calculation. This allows one to grid space more finely for the evaluation of VDW energies, which are very sensitive to small displacements. Tomioka and coworkers[3] use a similar molecular mechanics function, but do not employ the geometric mean approximation for VDW parameters; the interaction energies for multiple types of probe atoms are stored, as in the GRID program. In addition, their method allows ligand bonds to be rotated, and counts the number of intermolecular hydrogen bonds that are formed.

Important practical issues include wide applicability to a number of receptor structures and user control over the grid calculation. Our approach allows the user to specify the grid location, dimensions, and resolution, the cutoff distance for interactions, and the dielectric function, without changing the source code. Because the grid values are stored in one-dimensional arrays, any combination of spacing and x, y, and z extents may be used as long as the total number of points does not exceed the array size (10^6 in the

current work).

Our work to date suggests that the geometric mean approximation is useful, and that it is not necessary to store values for probes of several types. Given the steepness of the VDW potential, memory is better spent on grids with finer spacing. Since hydrogen-bonding groups are generally not allowed to respond to the docked ligand, it seems that it is beyond the resolution of the method to place any particular emphasis on hydrogen bonds. In addition, such calculations would increase computational time since angles as well as distances must be taken into account. As mentioned previously, we feel it is most efficient to perform rigid-body docking, not only because of the time required to consider torsional degrees of freedom, but also because specification of which bonds are rotatable and generation of the corresponding parameters are neither trivial nor easily automated.

Automated docking is preferable to interactive docking for database searching. Furthermore, the results of automated docking are not so strongly dependent on the preconceptions of the user, and in general a greater region of orientation space is explored. Many of the automated methods, however, have only been applied to systems smaller than macromolecule-ligand complexes,[9] or to reduced representations of molecules, so that detail at the level of individual atoms is not considered.[4]

Goodsell and Olson have used Monte Carlo simulated annealing with grid-based energy evaluation to dock molecules automatically.[10] Interaction energies are calculated for different probe types, using the AMBER function[13] but with the 10-12 term scaled by a factor of ten to give a potential well of –4 kcal/mol, and D = 40. Complexes of known structure were examined as test cases. Each simulation began with the ligand in the rough vicinity of the site and proceeded with incremental rigid-body movements and bond rotations. Several simulations were carried out for each complex, with the correct

structure being found and given the best energy in nearly all cases. Advantages of this method include consideration of ligand flexibility, reasonable computational demands, and insights that may be afforded by the simulation trajectories.

There is a fundamental difference between our docking method and the Monte Carlo simulated annealing approach. Our procedure is not carried out within a representation of Cartesian space; instead, it depends only on internal distance matching. Thus, there is no dependence on the starting locations of the molecules, and there are no effects due to steric hindrance or unfavorable charge-charge interactions *en route* to the site. Molecules may be docked successfully even when there is no low-energy pathway from the outside of the protein to the binding site.

Some aspects of the energy function used for the simulated annealing merit discussion. Since a constant dielectric of 40 decreases electrostatic contributions considerably, the 10-12 term was scaled up by a factor of ten to produce reasonable hydrogen bond energies. In addition, although bond rotations are allowed, internal energies are not included in the calculation.

Docking by internal distance matching is quite rapid. While our algorithm may require more computational time than the simulated annealing procedure for tasks that are done once per site, prior to docking, the time per low-energy orientation generated is encouragingly small. The distance-matching algorithm is flexible as well as powerful, in that the user may easily vary the thoroughness of the procedure and the number of sterically allowed orientations that will be found.

CONCLUSION

In summary, we have added molecular mechanics scoring capabilities to a rapid, geometric docking algorithm. Computational costs are kept to a minimum by the precalculation of values on three-dimensional grids. Four crystallographic complexes are used as test cases, in which the small molecule component is docked back into the receptor; the results are encouraging, as the force field score is able to identify the correct family of orientations in each case. Scoring methods that consider solely sterics or solely electrostatics are less successful. Many approximations are inherent in the method; however, we feel that a reasonable balance between rigor and computational tractability has been achieved. Since the results of database searching are highly dependent on the scoring function, improving the evaluation of orientations of a single molecule is an important step in improving the effectiveness of searching for lead compounds.

SPHGEN, DISTMAP, and contact scoring are included in DOCK version 2.0;[16,17] CHEMGRID and force field scoring are included in DOCK version 3.0. DOCK and associated programs are implemented in Fortran77 and available from I. D. Kuntz.

Acknowledgements. The authors gratefully acknowledge support from NIH grants GM-31497 (I. D. Kuntz) and GM-39552 (G. L. Kenyon), DARPA grant MDA-91-J-1013 (F. E. Cohen), and Glaxo Inc.; thanks are due also to K. A. Sharp, A. R. Leach, D. A. Pearlman, P. A. Kollman, R. Langridge, and the UCSF Computer Graphics Laboratory for their advice and assistance. The SYBYL molecular modeling package has been useful in many aspects of our work; we thank Tripos Associates, Inc. (St. Louis), for making this package available to us.

References

1. B. Busetta, I. J. Tickle, and T. L. Blundell, *J. Appl. Cryst.*, **16,** 432 (1983).

2. N. Pattabiraman, M. Levitt, T. E. Ferrin, and R. Langridge, *J. Comp. Chem.*, **6,** 432 (1985).

3. N. Tomioka, A. Itai, and Y. Iitaka, *J. Comp.-Aided Mol. Design,* **1,** 197 (1987).

4. S. J. Wodak and J. Janin, *J. Mol. Biol.*, **124,** 323 (1978).

5. I. D. Kuntz, J. M. Blaney, S. J. Oatley, R. Langridge, and T. E. Ferrin, *J. Mol. Biol.*, **161,** 269 (1982).

6. D. Goodsell and R. E. Dickerson, *J. Med. Chem.*, **29,** 727 (1986).

7. M. L. Connolly, *Biopolymers,* **25,** 1229 (1986).

8. M. Billeter, T. F. Havel, and I. D. Kuntz, *Biopolymers,* **26,** 777 (1987).

9. K. B. Lipkowitz and R. Zegarra, *J. Comp. Chem.*, **10,** 595 (1989).

10. D. S. Goodsell and A. J. Olson, *Proteins,* **8,** 195 (1990).

11. F. Jian and S.-H. Kim, *J. Mol. Biol.*, **219,** 79 (1991).

12. P. J. Goodford, *J. Med. Chem.*, **28,** 849 (1985).

13. S. J. Weiner, P. A. Kollman, D. A. Case, U. C. Singh, C. Ghio, G. Alagona, S. Profeta, Jr., and P. Weiner, *J. Am. Chem. Soc.*, **106,** 765 (1984).

14. S. J. Weiner, P. A. Kollman, D. T. Nguyen, and D. A. Case, *J. Comp. Chem.*, **7,** 230 (1986).

15. R. L. DesJarlais, R. P. Sheridan, G. L. Seibel, J. S. Dixon, I. D. Kuntz, and R. Venkataraghavan, *J. Med. Chem.*, **31,** 722 (1988).

16. B. K. Shoichet and I. D. Kuntz, *J. Mol. Biol.*, **221,** 327 (1991).

17. B. K. Shoichet, D. L. Bodian, and I. D. Kuntz, *J. Comp. Chem.*, **13,** 380 (1992).

18. M. L. Connolly, *J. Appl. Crystallogr.*, **16,** 548 (1983).

19. M. L. Connolly, *Science,* **221,** 709 (1983).

20. I. Klapper, R. Hagstrom, R. Fine, K. Sharp, and B. Honig, *Proteins,* **1,** 47 (1986).

21. M. K. Gilson, K. A. Sharp, and B. H. Honig, *J. Comp. Chem.*, **9,** 327 (1987).

22. F. M. Richards, *Annu. Rev. Biophys. Bioeng.*, **6,** 151 (1977).

23. A. R. Fersht and M. J. E. Sternberg, *Prot. Eng.*, **2,** 527 (1989).

24. A. T. Hagler, E. Huler, and S. Lifson, *J. Am. Chem. Soc.*, **96,** 5319 (1977).

25. D. R. Ferro and J. Hermans, *Acta Crystallogr.*, **A33,** 345 (1977).

26. F. C. Bernstein, T. F. Koetzle, G. J. B. Williams, E. F. Meyer, Jr., M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi, *J. Mol. Biol.*, **112,** 535 (1977).

27. E. E. Abola, F. C. Bernstein, S. H. Bryant, T. F. Koetzle, and J. Weng, in *Crystallographic Databases: Information Content, Software Systems, Scientific Applications,* F. H. Allen, G. Bergerhoff, and R. Seivers, Eds., Data Commission of the International Union of Crystallography, Bonn/Cambridge/Chester, 1987, pp. 107-132.

28. J. T. Bolin, D. J. Filman, D. A. Matthews, R. C. Hamlin, and J. Kraut, *J. Biol. Chem.*, **257,** 13650 (1982).

29. B. Borah, C.-W. Chen, W. Egan, M. Miller, A. Wlodawer, and J. S. Cohen, *Biochemistry,* **24,** 2058 (1985).

30. N. K. Vyas, M. N. Vyas, and F. A. Quiocho, *Science,* **242,** 1290 (1988).

31. Private communication from W. N. Lipscomb.

32. M. Whitlow and M. M. Teeter, *J. Am. Chem. Soc.*, **108,** 7164 (1986).

33. U. C. Singh and P. A. Kollman, *J. Comp. Chem.*, **5,** 129 (1984).

34. J. Gasteiger and M. Marsili, *Tetrahedron,* **36,** 3219 (1980).

35. M. Marsili and J. Gasteiger, *Croat. Chem. Acta,* **53,** 601 (1980).

36. J. Gasteiger and M. Marsili, *Organ. Magn. Reson.*, **15,** 353 (1981).

37. A. Streitweiser, *Molecular Orbital Theory for Organic Chemists,* Wiley, New York, 1961.

38. W. P. Purcel and J. A. Singer, *J. Chem. Eng. Data,* **12,** 235 (1967).

39. Molecular Modeling System SYBYL, Version 5.4, TRIPOS Associates, Inc., St. Louis, MO 63117.

40. R. L. DesJarlais, R. P. Sheridan, J. S. Dixon, I. D. Kuntz, and R. Venkataraghavan, *J. Med. Chem.*, **29,** 2149 (1986).

41. A. R. Leach and I. D. Kuntz, *J. Comp. Chem.*, **13,** 730 (1992).

42. M. K. Gilson and B. Honig, *Proteins,* **4,** 7 (1988).

43. M. E. Davis and J. A. McCammon, *J. Comp. Chem.*, **11,** 401 (1990).

44. J. Novotny, R. E. Bruccoleri, and F. A. Saul, *Biochemistry,* **28,** 4735 (1989).

45. D. Eisenberg and A. D. McLachlan, *Nature,* **319,** 199 (1986).

46. L. Chiche, L. M. Gregoret, F. E. Cohen, and P. A. Kollman, *Proc. Natl. Acad. Sci. USA,* **87,** 3240 (1990).

CHAPTER 3: ORIENTATIONAL SAMPLING AND RIGID-BODY MINIMIZATION IN MOLECULAR DOCKING

INTRODUCTION

While the evaluation of complexation geometries is important in any application of molecular docking, the sampling of orientation space is paramount. Favorable orientations must be found before they can be recognized as such; even a very sophisticated evaluation scheme will be useless without some way of generating orientations of the ligand within the receptor site. The structures generated should include, but not necessarily be limited to, those which will be considered favorable by the evaluation function. In the case of DOCK, it has been shown that experimental geometries can be reproduced quite accurately.[1-4] The degree of sampling necessary to achieve this goal has not been examined systematically, however. It is also unclear whether it is more efficient to sample thoroughly and select the most favorable orientations from the resulting population, or to generate fewer orientations and optimize them within the context of the receptor site. In this chapter, I address these issues, using the same systems employed in previous tests of DOCK 3.0[4] (Chapter 2).

TEST SYSTEMS AND COMPUTATIONAL METHODS

Four well-determined structures of ligand/receptor complexes were selected from the Brookhaven Protein Data Bank:[5,6] 4dfr[7] (dihydrofolate reductase/methotrexate), 6rsa[8] (ribonuclease A/uridine vanadate), 2gbp[9] (periplasmic binding protein/glucose), and 3cpa[10] (carboxypeptidase A/glycyltyrosine). DOCK 3.0 was used to generate multiple orientations of each ligand in the corresponding receptor site, as described in the preced-

ing chapter.[4] Briefly, in each system, all crystallographic waters and ions were removed, the ligand and receptor were separated, and hydrogens were added in standard geometries. Atomic charges were derived as described previously.[4] A molecular surface was created for the receptor region of interest with the MS algorithm,[11,12] then used to calculate spheres for docking. The CHEMGRID and DISTMAP modules of DOCK 3.0 were used to create grids for force field and contact scoring, respectively. Docking and scoring parameters were the same as described previously[4] for the runs with high orientational sampling; however, two additional runs were performed for each system at lower levels of sampling. The docking parameters for each trial are listed in Table I.

The ligand-site matching algorithm is purely geometric and identical to that used in DOCK 2.0.[2,3] Sets of sphere centers that match sets of ligand atoms based on pairwise internal distances are identified and used to generate orientations. The sphere-sphere and atom-atom distances are first sorted into bins; a sphere-atom pairing is only considered when these points are in the corresponding bins. The variables *lbinsz*, *lovlap*, *sbinsz*, and *sovlap* are the ligand bin width, ligand bin overlap, sphere bin width, and sphere bin overlap, all in the units of angstroms. In the present work, a "match" is found when the distances among four spheres are equivalent to the distances among four ligand atoms, within a tolerance *dislim*. *Nmatch* is the total number of matches, or orientations, found. Together with *dislim*, the four bin variables control *nmatch*. *Minwrt* is the number of orientations written; an orientation is written if its contact score is no less than *minscr* and its force field score is no greater than *maxscr*.

Rigid-body minimization was performed on each orientation written from DOCK, using a modified version of Blaney's program RGDMIN.[13] In this algorithm, the receptor molecule is kept stationary while the six degrees of freedom of the ligand are

Table I. Docking variables.[a]

run	*dislim*	*lbinsz*	*lovlap*	*sbinsz*	*sovlap*	*nmatch*	*minscr*	*maxscr*	*minwrt*
4dfr_high	1.5	1.0	0.2	1.0	0.2	67,234	60	100	2406
4dfr_med	1.5	0.4	0.1	0.8	0.2	17,354	60[b]	100	869
4dfr_low	1.5	0.2	0.0	1.0	0.0	7158	60[b]	100	337
6rsa_high	1.5	1.0	0.5	1.0	0.5	77,184	40	100	3492
6rsa_med	1.5	0.4	0.1	0.8	0.2	4785	0	100	266
6rsa_low	1.5	0.2	0.0	1.0	0.0	1693	0	100	105
2gbp_high	1.5	1.0	0.4	1.0	0.4	196,752	60	100	1680
2gbp_med	1.5	0.4	0.1	0.8	0.2	21,037	0	100	849
2gbp_low	1.5	0.2	0.0	1.0	0.0	7773	0	100	389
3cpa_high	2.0	1.5	0.5	1.5	0.5	794,541	100	100	2684
3cpa_med	1.5	0.4	0.1	0.8	0.2	7946	0	100	518
3cpa_low	1.5	0.2	0.0	1.0	0.0	4053	0	100	301

[a]See text for descriptions of the variables.

[b]Set greater than zero to limit the number of orientations written.

manipulated to minimize the calculated interaction energy. A Davidon-Fletcher-Powell subroutine is used;[14] energy-minimization steps are initially steepest-descent and gradually change to Newton-Raphson. Daniel Gschwend, a graduate student in the Kuntz group, modified RGDMIN to easily incorporate the DOCK force field scoring function and parameters, resulting in the program RGDMIN3. In the current work, parameters were set so that the only difference between the DOCK force field scores and the RGDMIN3 interaction energies is that the former use precalculated grid values while the latter are based on the exact ligand atom-receptor atom distances. The two sets of values correspond closely (see Results and Discussion). The minimizations were performed with a maximum translational step size of 0.5 angstroms and a maximum rotational step size of 10°.

All calculations were carried out on a Silicon Graphics Iris 4D/35 workstation; timings are given in Table II. Although the input and results for the high-sampling docking runs are the same as described previously,[4] the runs were faster, for two reasons: time-saving changes were incorporated into DOCK between the two sets of runs, and a faster workstation was used this time (the older runs were done on a Silicon Graphics Iris 4D/25).

RESULTS AND DISCUSSION

For each system and at each level of sampling, the root-mean-square deviation in atomic position (RMSD) of every orientation of the ligand relative to the experimentally observed orientation has been calculated. Hydrogens were not included in the calculation. RMSD is plotted versus DOCK force field (FF) score, before and after rigid-body minimization. Comparisons are made between the RMSD values before and after

Table II. Timings.[a]

run	DOCK (s)	RGDMIN3 (hr:min)	*minwrt*[b]
4dfr_high	103	10:39	2406
4dfr_med	43	3:51	869
4dfr_low	30	1:30	337
6rsa_high	146	9:26	3492
6rsa_med	34	0:39	266
6rsa_low	28	0:15	105
2gbp_high	178	6:37	1680
2gbp_med	60	2:25	849
2gbp_low	48	1:04	389
3cpa_high	766	23:24	2684
3cpa_med	33	3:00	518
3cpa_low	25	1:42	301

[a]All calculations performed on a Silicon Graphics IRIS 4D/35 workstation.

[b]As in Table I; number of orientations written by DOCK and minimized using RGDMIN3.

minimization, and between the RGDMIN3 scores and the DOCK FF scores. Below, I discuss the results for each system, concentrating on the issues of sampling and minimization. Table III summarizes the results in terms of the top-scoring orientations before and after minimization. Scoring methods are discussed and more thorough descriptions of the families of orientations are given in the preceding chapter.[4]

Dihydrofolate Reductase

N1-protonated 2,4-diamino-6-methylpteridine, the rigid part of methotrexate, was used for docking.[4] The lowest-RMSD family of orientations is identified as the most favorable at all levels of sampling; even before minimization, the lowest FF scores correspond to RMSD's below 2.0 angstroms (Figure 1). The main effect of minimization is to collapse the high FF scores down without, for the most part, causing large changes in RMSD (Figures 1 and 2). Apparently, steric conflicts can be resolved by slight rigid-body adjustments of the ligand, as expected from the steepness of the van der Waals potential. There are roughly equal numbers of points above and below the diagonal in Figure 2, indicating that similar numbers of orientations move farther from and closer to the experimental geometry, respectively, during minimization. This is not surprising, since many local minima are being explored; there is no reason to expect a linear or even monotonic relationship between RMSD and interaction energy over such a large region of orientation space. When only the lowest-RMSD structures are considered, however, most of the orientations move toward the crystallographically observed position during minimization.

RGDMIN3 scores are plotted versus FF scores in Figure 3. Note the difference in scale between the two graphs. As stated above, RGDMIN3 scores and FF scores are based on the same parameters and equations. Differences reflect two effects. First, a

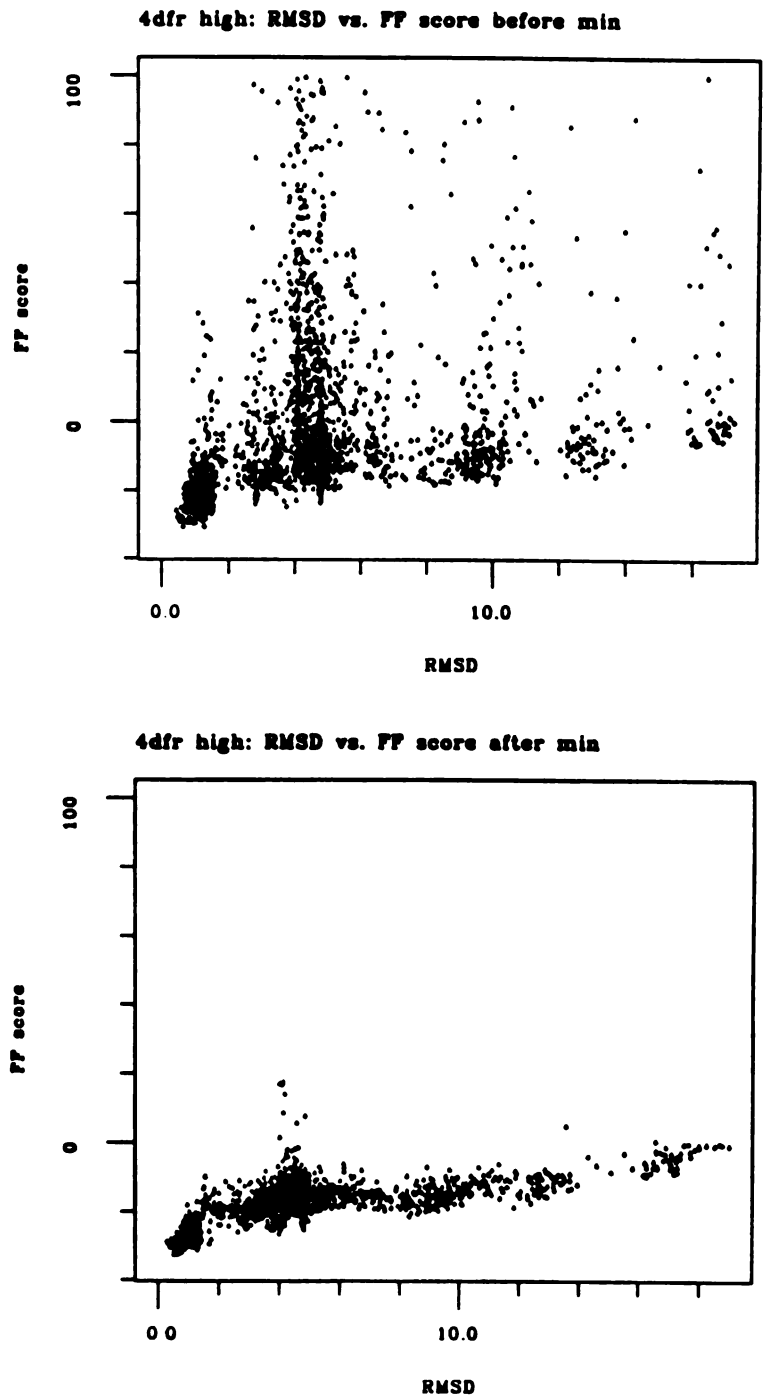


Figure 1A. 4dfr high-sampling run: RMSD versus force field score before and after rigid-body minimization.

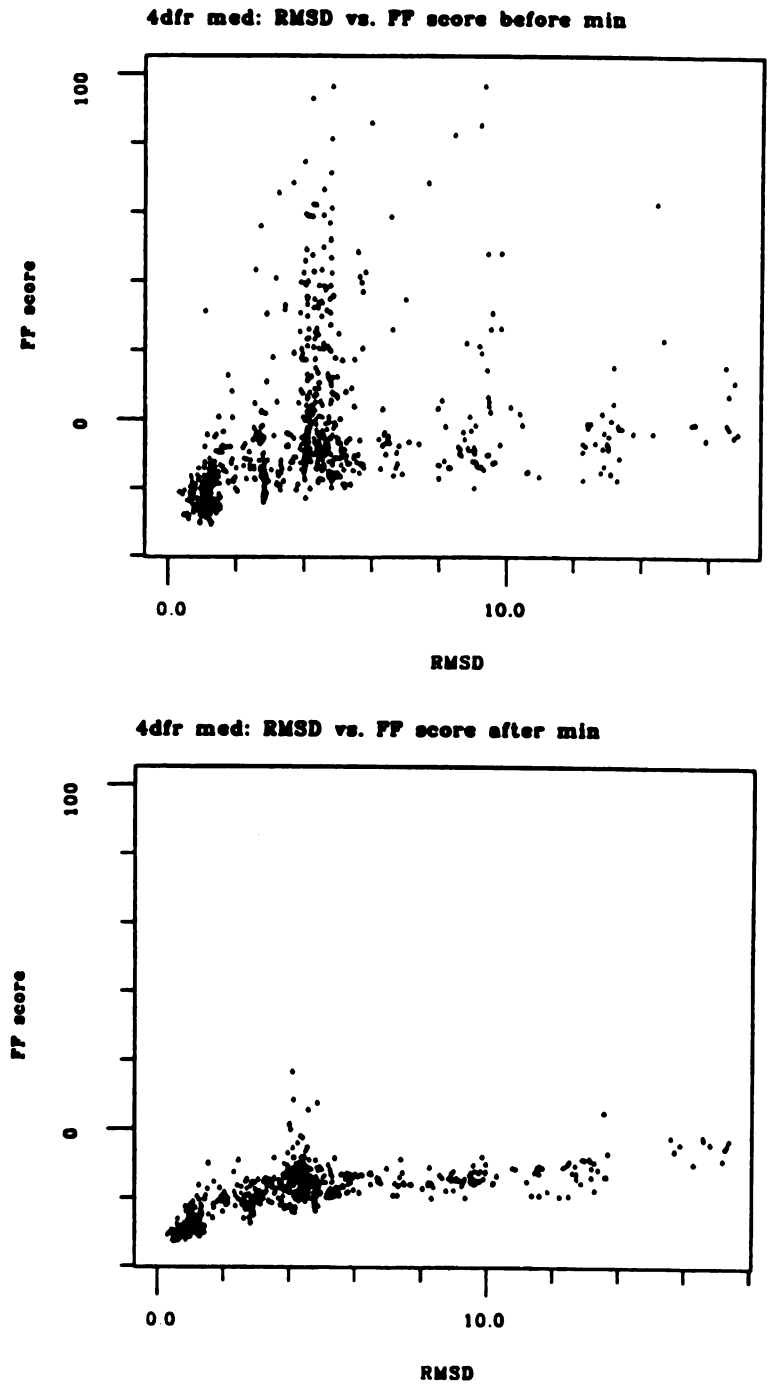


Figure 1B. 4dfr intermediate-sampling run: RMSD versus force field score before and after rigid-body minimization.

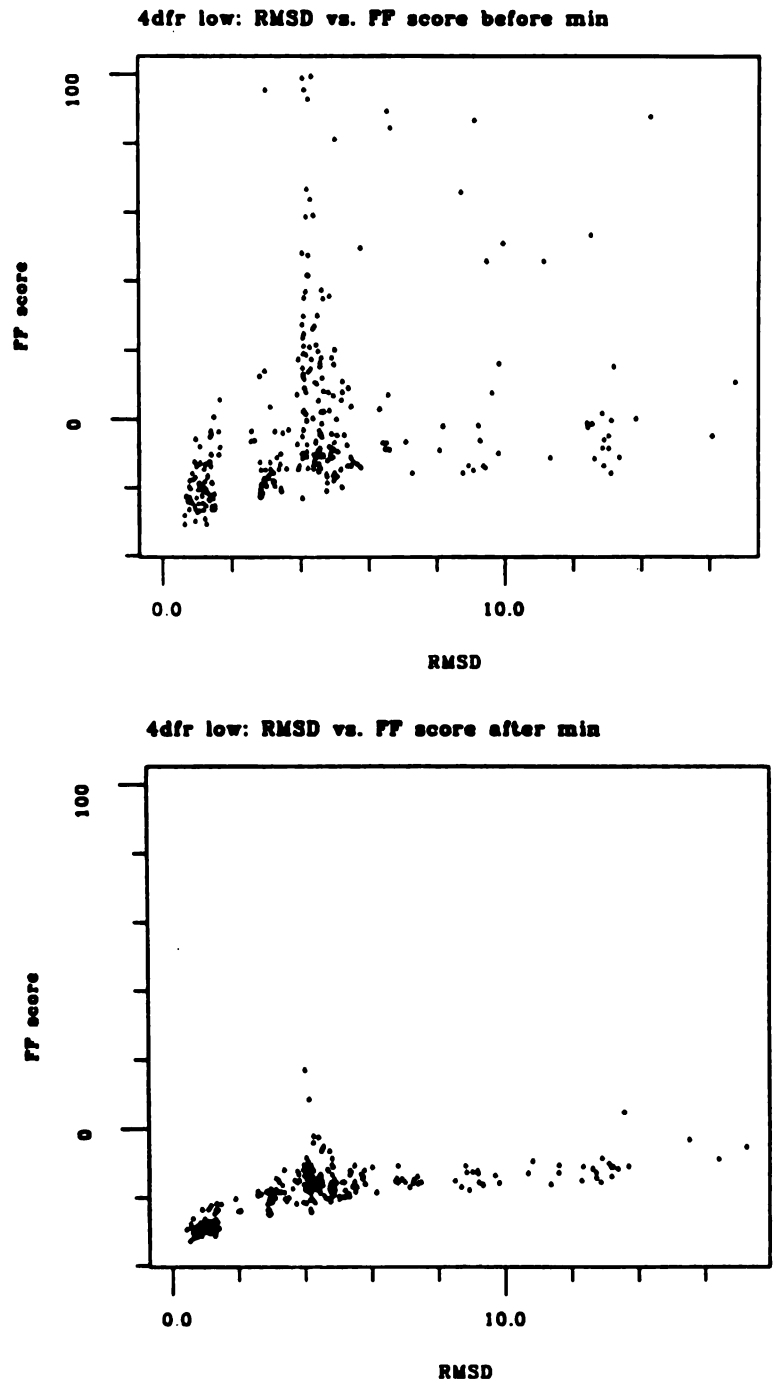


Figure 1C. 4dfr low-sampling run: RMSD versus force field score before and after rigid-body minimization.

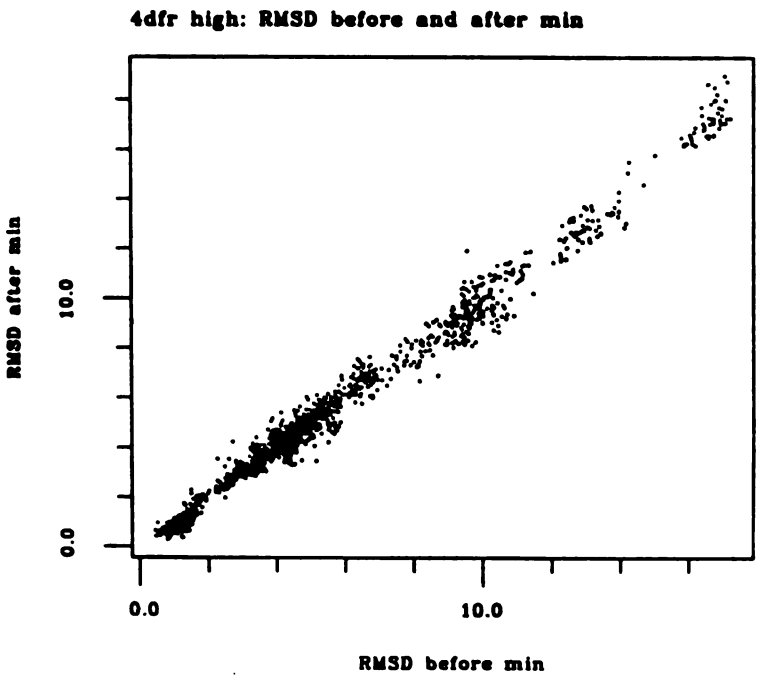


Figure 2. 4dfr high-sampling run: RMSD before and after rigid-body minimization.

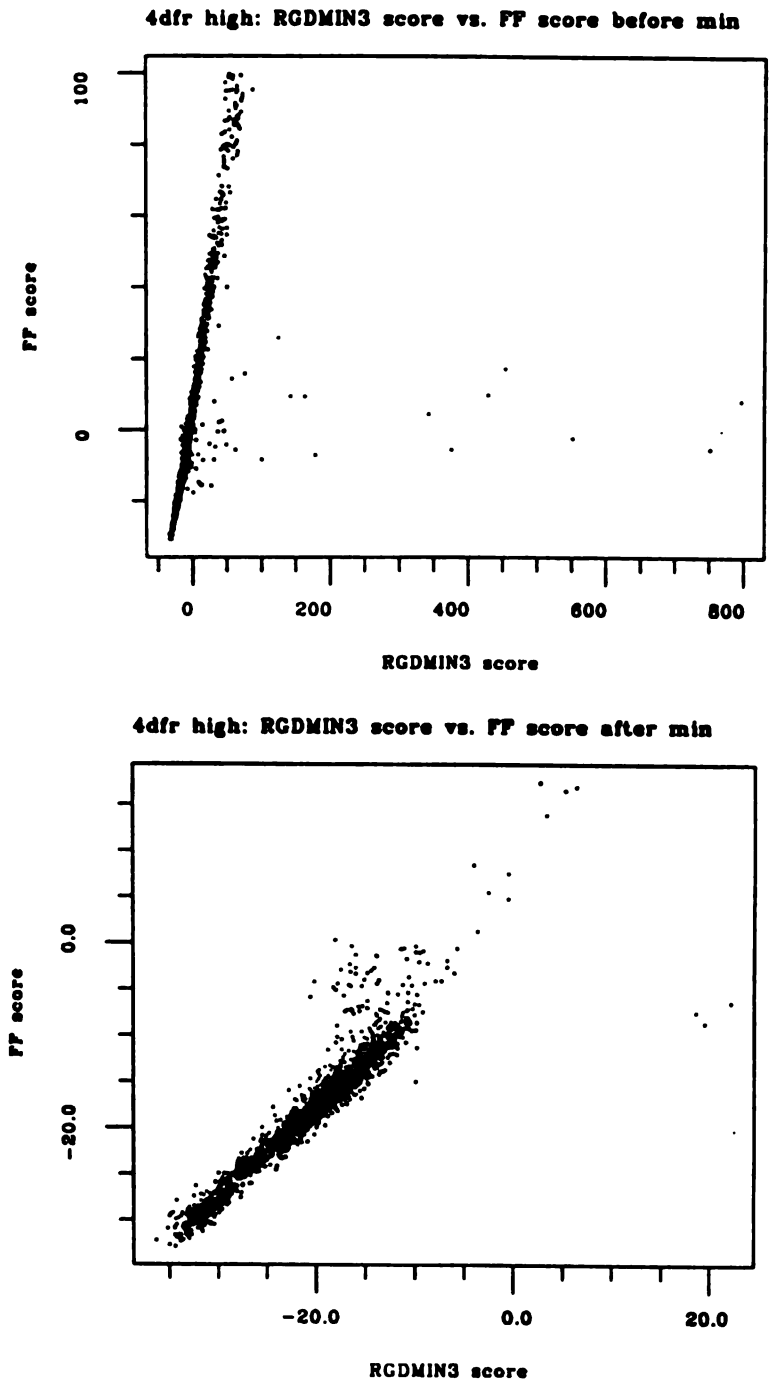


Figure 3. 4dfr high-sampling run: RGDMIN3 score versus force field score before and after rigid-body minimization.

grid approximation is inherent in the FF scores.[4] Trilinear interpolation among the eight surrounding grid points is used to estimate the value of a highly curved potential surface at the location of each ligand atom. There is a general tendency for overestimation; if one of the grid points represents a bad contact, the value stored for that point can be exceedingly large and dominate the interpolated result. This is due to the steepness of the van der Waals component of the interaction potential. Second, it is possible for docked orientations to have atoms that fall outside the FF grid volume. These atoms do not contribute to the FF score, and any bad contacts they make are not detected during docking. They will, however, contribute to the RGDMIN3 score. FF scores much higher than RGDMIN3 scores are usually due to the first effect, whereas FF scores much lower than RGDMIN3 scores are usually due to the second effect. Minimization lowers the RGDMIN3 scores (by definition) and the FF scores while reducing the occurrence of each effect (Figure 3). Before and after minimization, the agreement is best for the most favorable orientations.

Ribonuclease A

Uridine 3′-phosphate was constructed from the crystallographic ligand, uridine vanadate, and used for docking.[4] At the highest level of sampling, the lowest-RMSD family of orientations is identified as the most favorable both before and after minimization (Figure 4A). At the intermediate level of sampling, just two orientations resembling the experimental binding mode are found; they receive the best scores, but only after minimization are they clearly the most favorable (Figure 4B). At the lowest level of sampling, members of the lowest-RMSD family of orientations again receive the best scores, but minimization decreases their separation in score from orientations with RMSD's close to 5.0 angstroms (Figure 4C). The latter have their phosphate groups in

essentially the correct position but are at an angle of about 60° relative to the experimental orientation. This weak discrimination between "correct" and "incorrect" modes by the FF score may reflect an overemphasis on electrostatics, especially when interactions between formal charges are involved.[4]

Interestingly, a greater number of low-RMSD orientations is found in this system with low sampling (Figure 4C) than with intermediate sampling (Figure 4B), using the parameters in Table I. This highlights an important point: results from a higher level of sampling, as quantified by *nmatch* or *minwrt*, will not necessarily include all of the results from a lower level of sampling. Although a great deal of overlap is to be expected, the amount actually obtained depends on the sphere-sphere distances, the atom-atom distances, and the docking parameters used in each run.

As noted above, minimization may move orientations closer to or farther from the crystallographic position; in this case, however, there is a preponderance of orientations for which the RMSD decreases (Figure 5). This is particularly true for orientations that start out in the approximate vicinity of the observed binding mode.

RGDMIN3 score is plotted versus FF score in Figure 6. Again, minimization decreases both kinds of scores and improves their agreement. Both before and after minimization, the agreement is greatest for the most favorable orientations.

Periplasmic Binding Protein

Three families of orientations are produced when β-D-glucose is docked to periplasmic binding protein.[4] The lowest-RMSD structures reproduce the experimental geometry, in which glucose occupies the center of a tunnel traversing the protein. The intermediate RMSD's correspond to orientations that overlap the crystal structure ligand

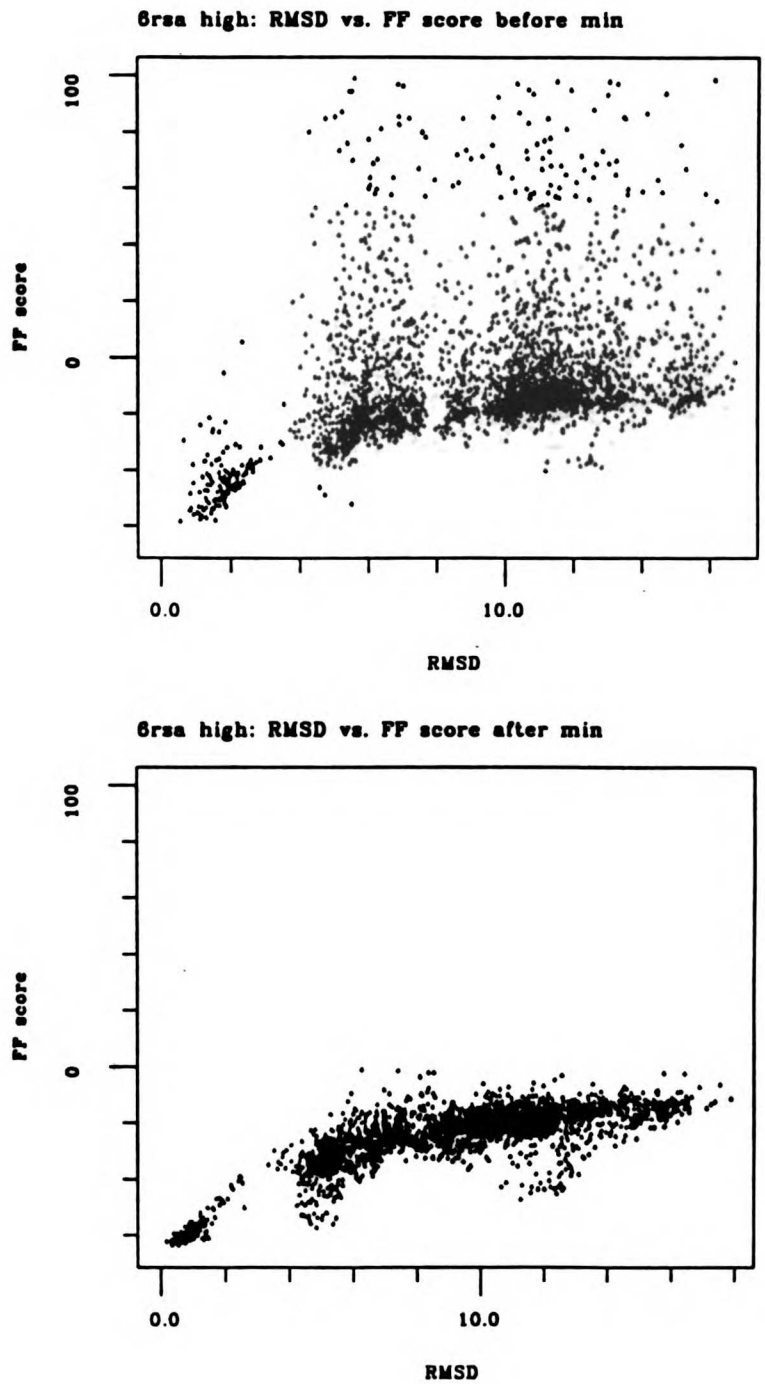


Figure 4A. 6rsa high-sampling run: RMSD versus force field score before and after rigid-body minimization.

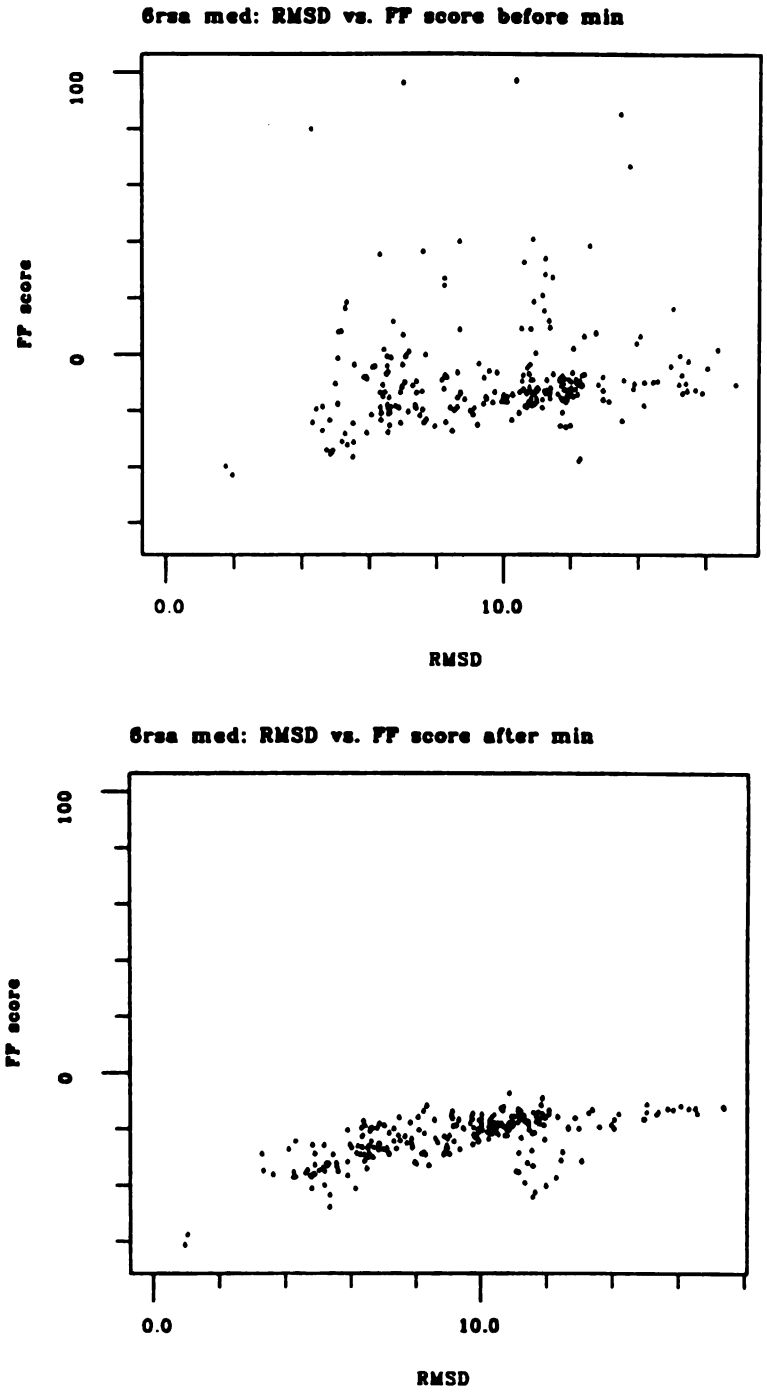


Figure 4B. 6rsa intermediate-sampling run: RMSD versus force field score before and after rigid-body minimization.

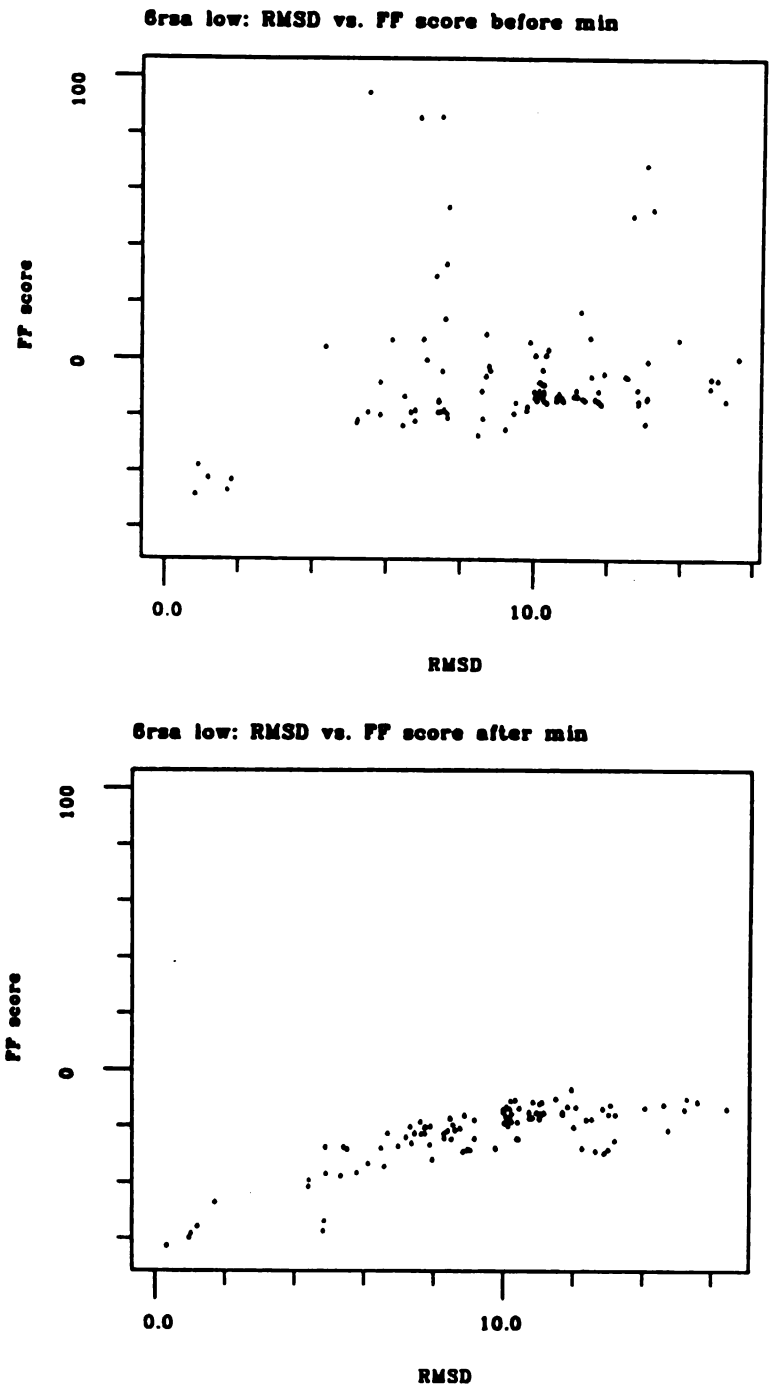


Figure 4C. 6rsa low-sampling run: RMSD versus force field score before and after rigid-body minimization.

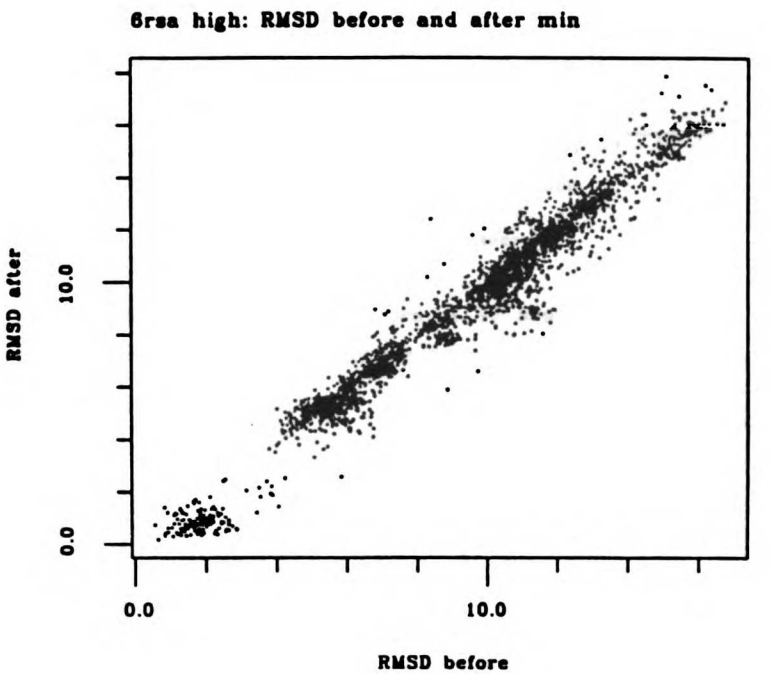


Figure 5. 6rsa high-sampling run: RMSD before and after rigid-body minimization.

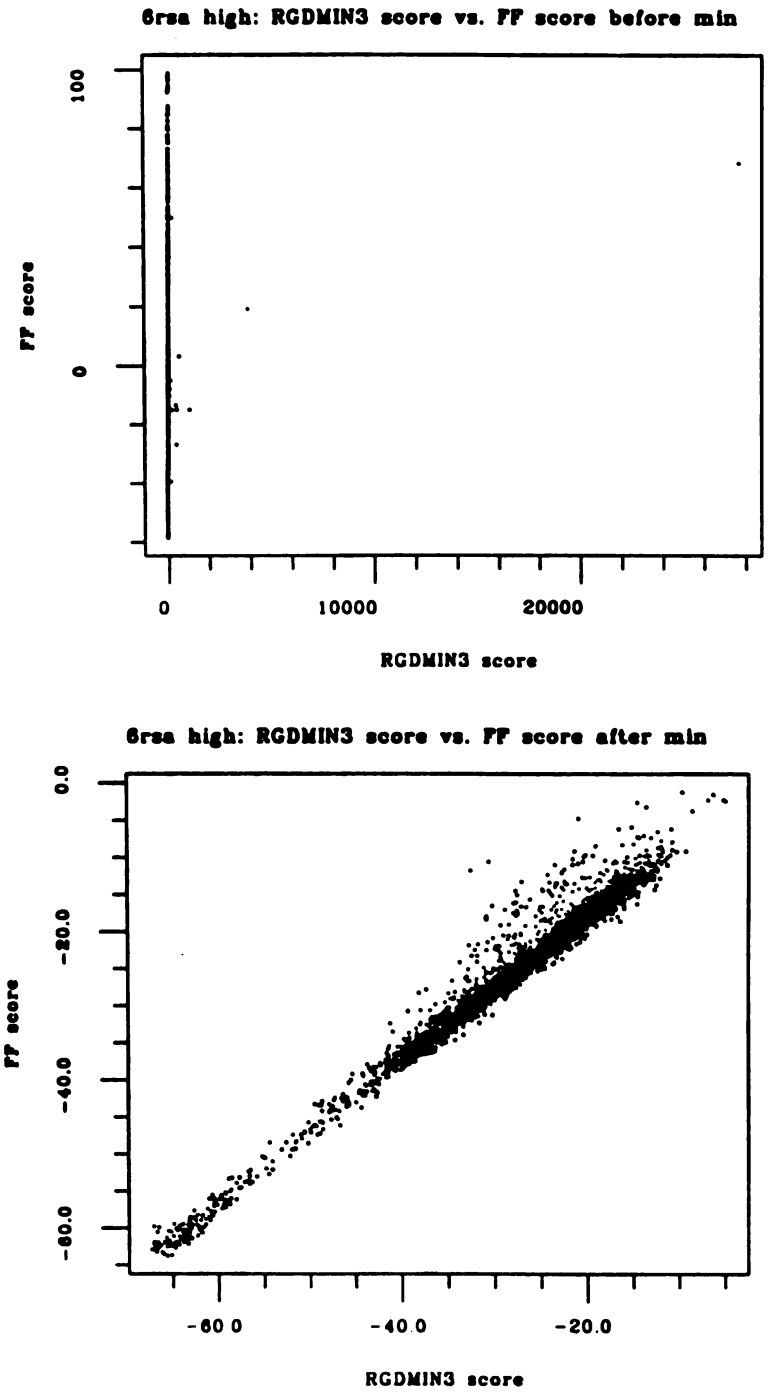


Figure 6. 6rsa high-sampling run: RGDMIN3 score versus force field score before and after rigid-body minimization.

but are flipped or rotated in several different ways. The high RMSD's correspond to structures located in either end of the tunnel.

At the highest level of sampling, the lowest-RMSD family of orientations is identified as the most favorable both before and after minimization (Figure 7A). At the intermediate and low levels of sampling, at least one orientation resembling the observed binding mode is generated, but does not receive the best score until after minimization (Figure 7, B and C). This suggests that if sampling is rather sparse, minimization may be helpful; it cannot, however, salvage a situation in which no orientations close to the true binding mode (whatever that may be) are found. RMSD increases and decreases are seen upon minimization, with decreases predominating in the low- and intermediate-RMSD families of orientations (Figure 8). RGDMIN3 and FF scores respond as expected (Figure 9). Overestimation due to the FF grid approximation remains in some of the minimized structures, and is especially evident in the expanded scale of the postminimization graph.

Carboxypeptidase A

The crystallographic ligand, glycyl-L-tyrosine, was used for docking.[4] The results resemble those from the 2gbp runs: members of the lowest-RMSD family of orientations receive the best scores both before and after minimization when sampling is intensive (Figure 10A), but only after minimization when intermediate or low sampling is performed (Figure 10, B and C). Before minimization, at the low and intermediate levels of sampling, the best scores go to structures flipped end-to-end relative to the experimental binding mode. As in the 2gbp low-sampling run, the 3cpa low-sampling run only generates one orientation similar to the crystallographic orientation. Rigid-body minimization has the expected effects on RMSD's (Figure 11) and scores (Figure 12).

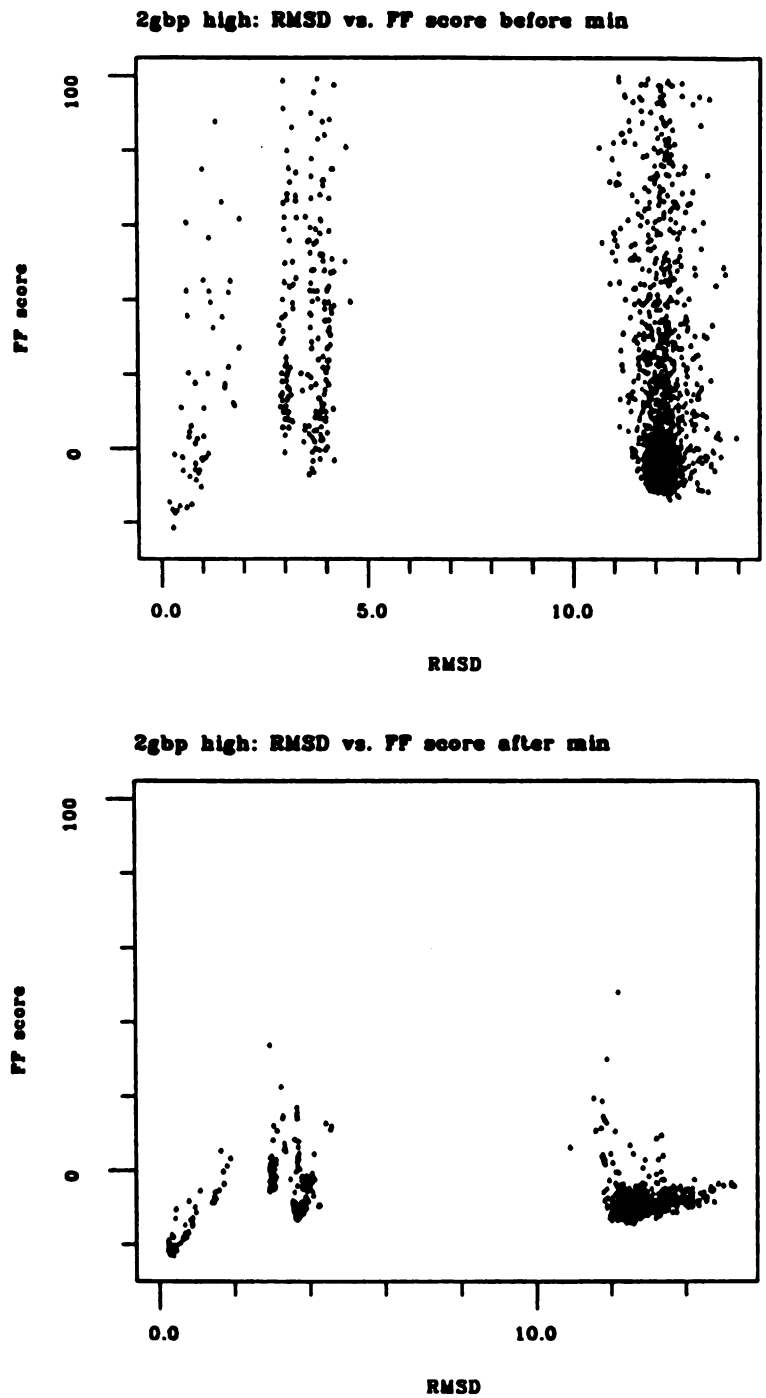


Figure 7A. 2gbp high-sampling run: RMSD versus force field score before and after rigid-body minimization.

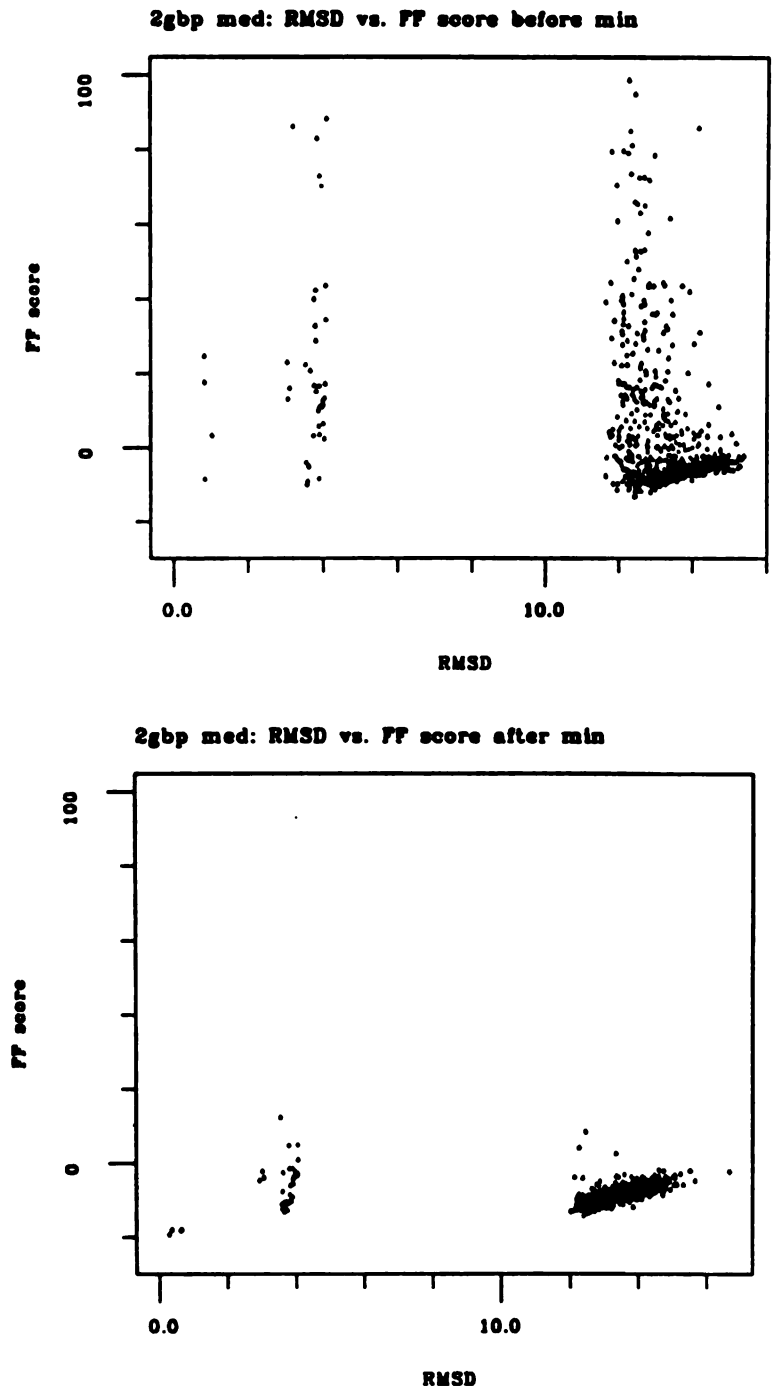


Figure 7B. 2gbp intermediate-sampling run: RMSD versus force field score before and after rigid-body minimization.

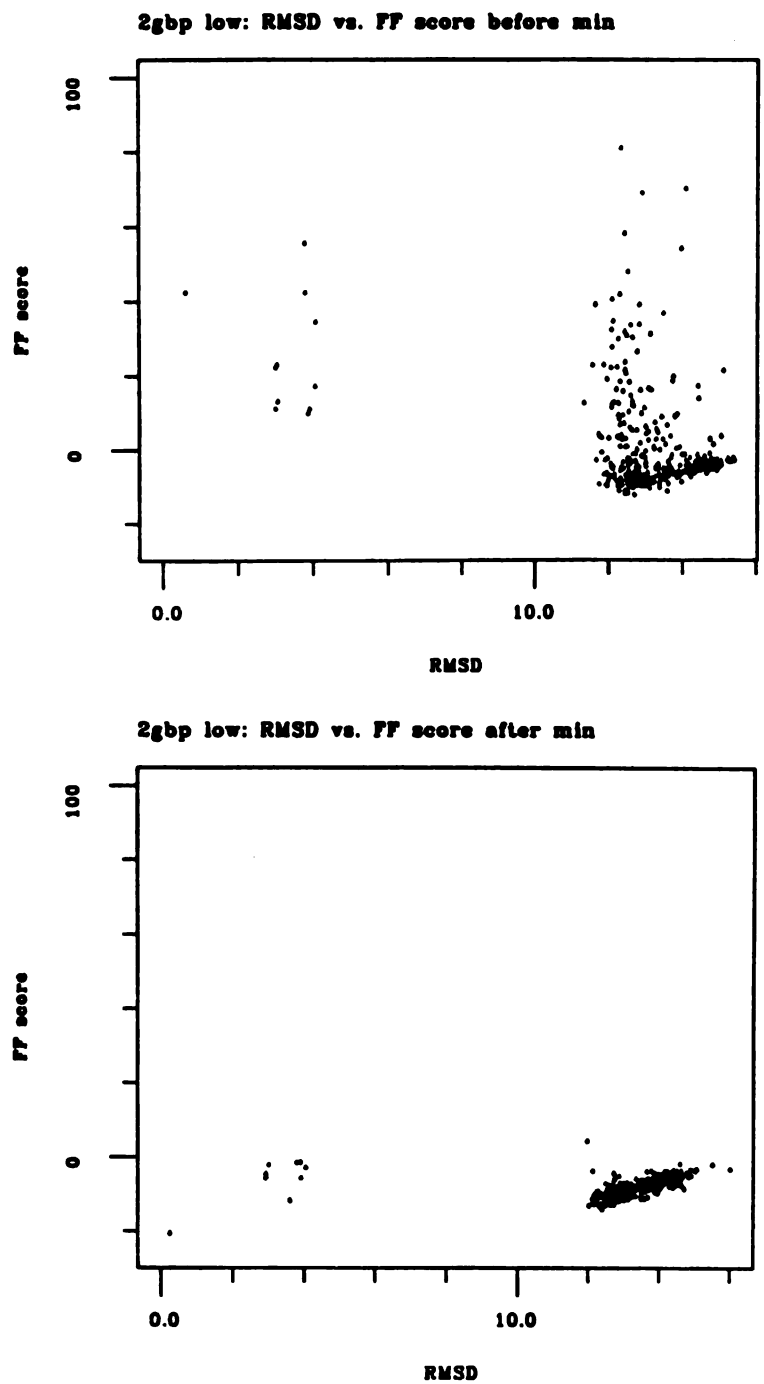


Figure 7C. 2gbp low-sampling run: RMSD versus force field score before and after rigid-body minimization.

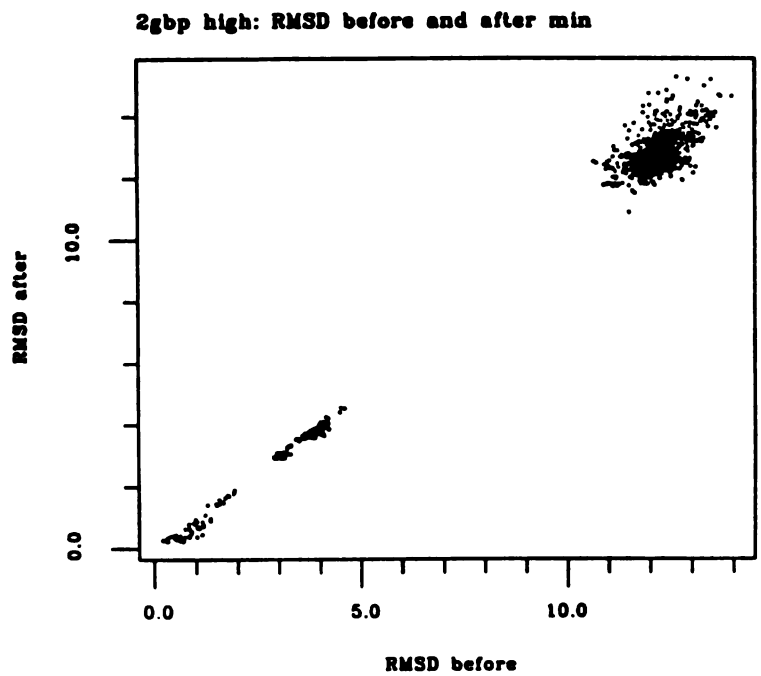


Figure 8. 2gbp high-sampling run: RMSD before and after rigid-body minimization.

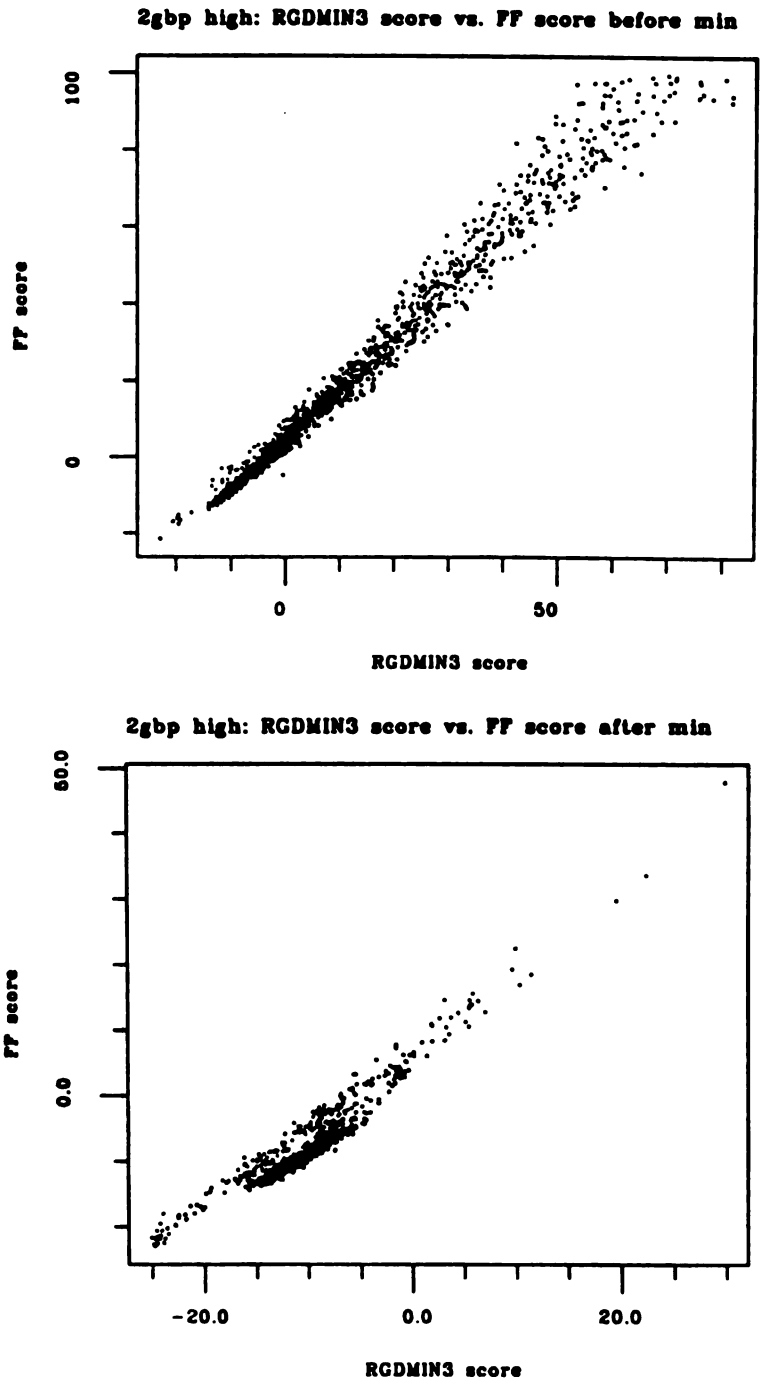


Figure 9. 2gbp high-sampling run: RGDMIN3 score versus force field score before and after rigid-body minimization.

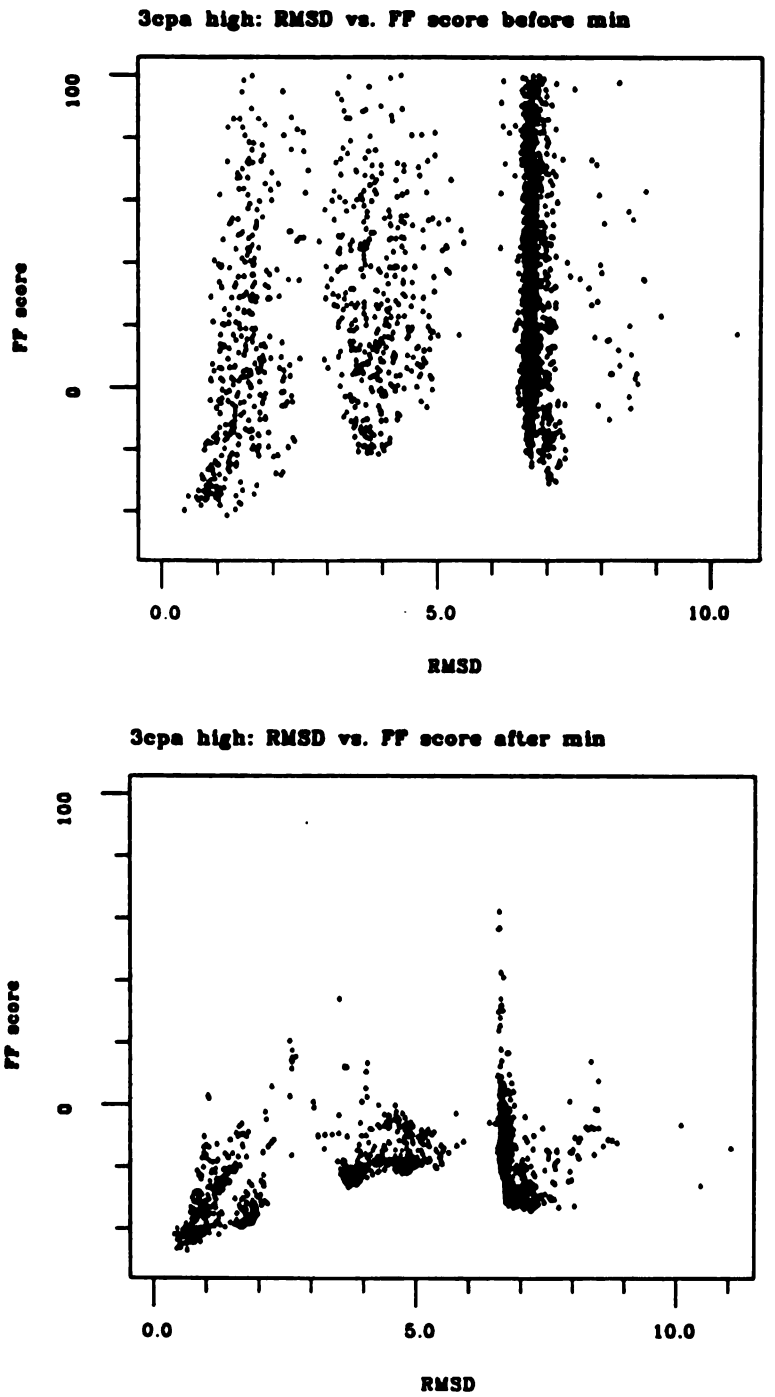


Figure 10A. 3cpa high-sampling run: RMSD versus force field score before and after rigid-body minimization.

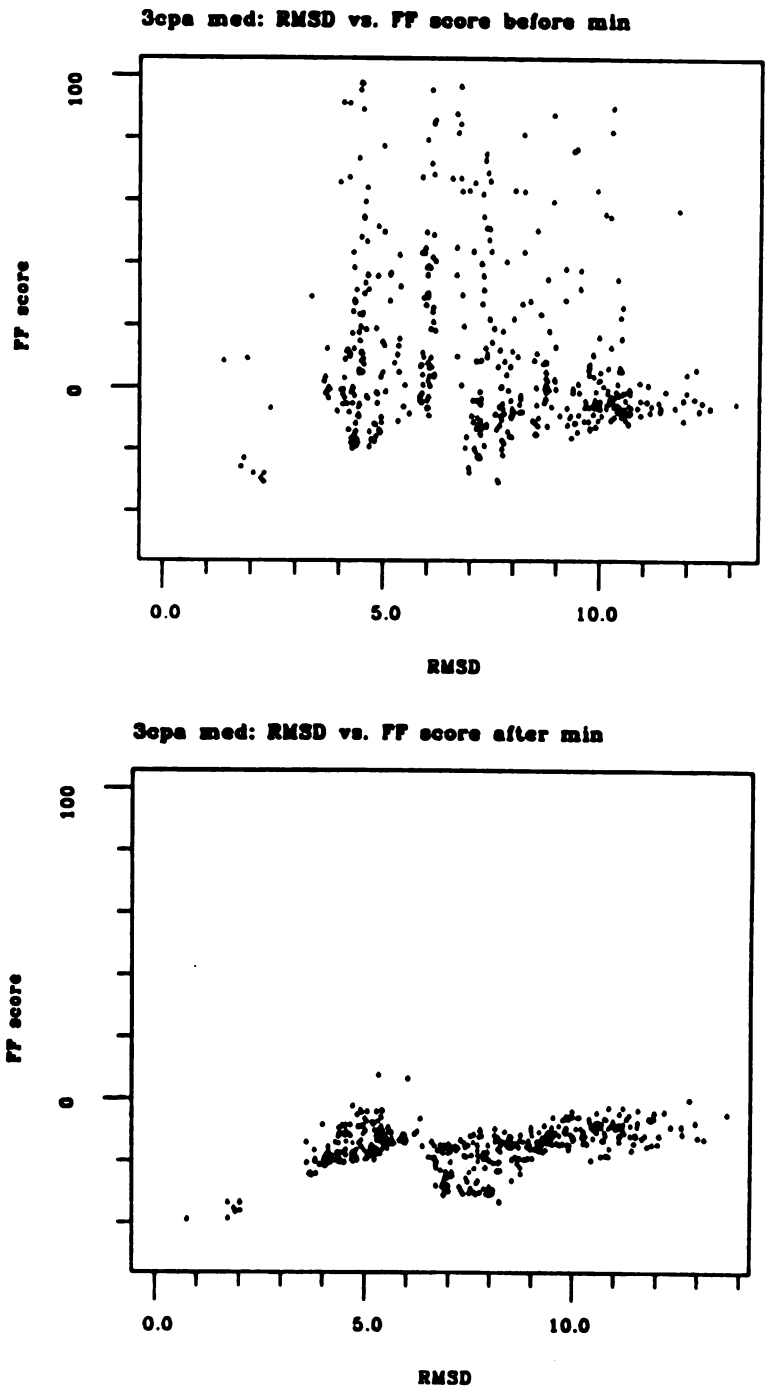


Figure 10B. 3cpa intermediate-sampling run: RMSD versus force field score before and after rigid-body minimization.

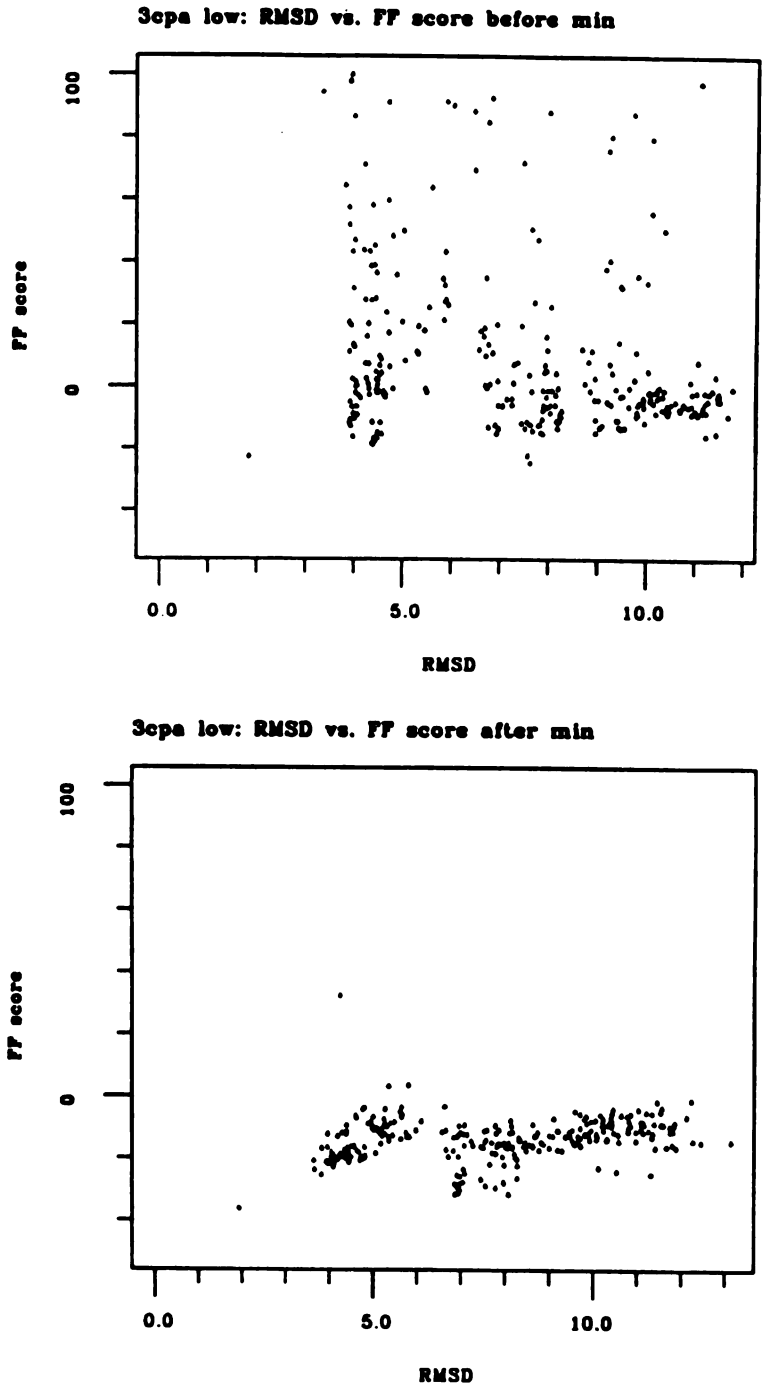


Figure 10C. 3cpa low-sampling run: RMSD versus force field score before and after rigid-body minimization.

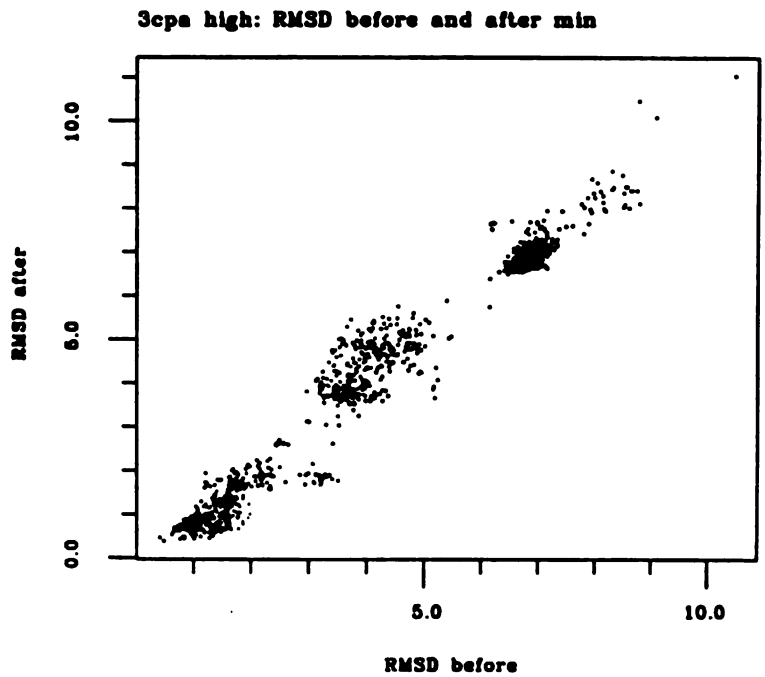


Figure 11. 3cpa high-sampling run: RMSD before and after rigid-body minimization.

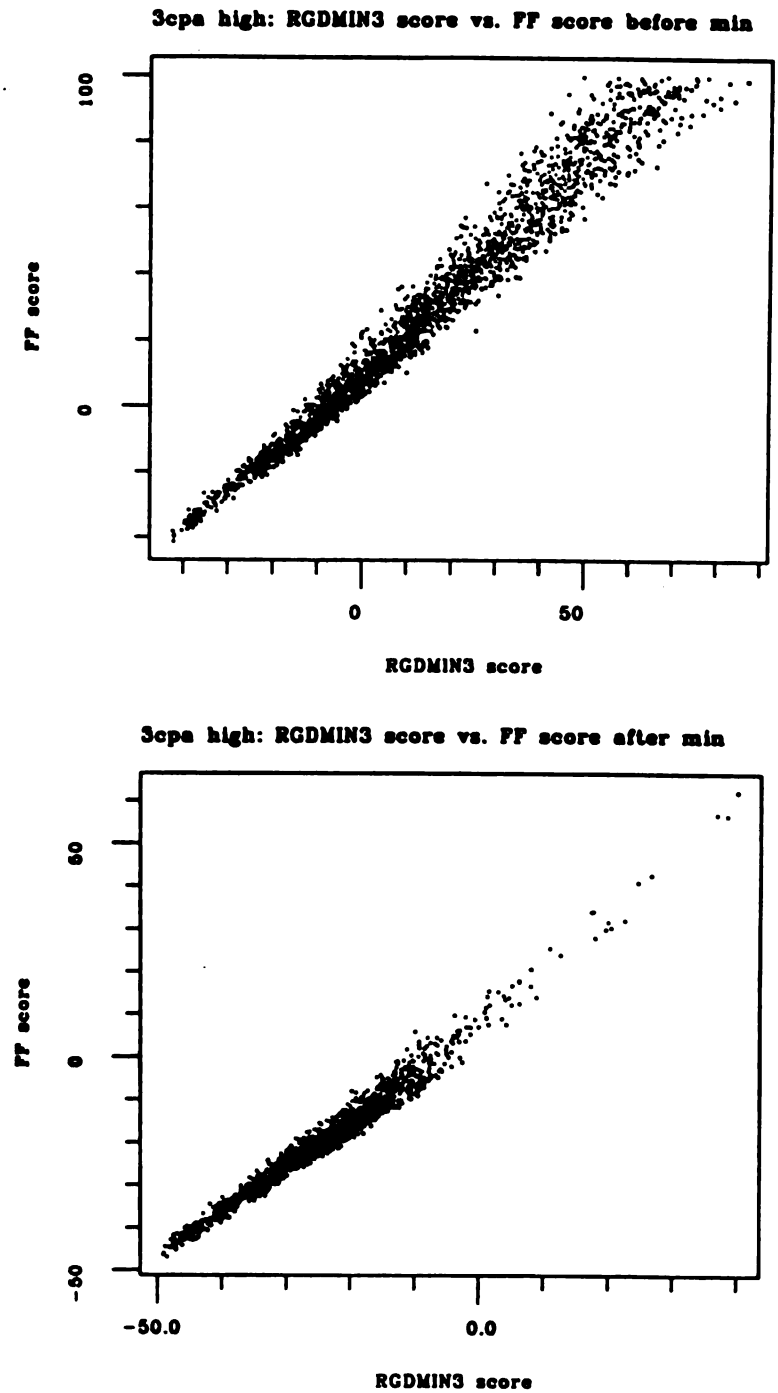


Figure 12. 3cpa high-sampling run: RGDMIN3 score versus force field score before and after rigid-body minimization.

Table III. The top-scoring orientations before and after minimization.

		"correct"[a]		"incorrect"[b]	
run		best FF score[c]	RMSD[d]	best FF score[c]	RMSD[d]
4dfr_high:	before min	−30.828	0.64	−24.998	3.75
	after min	−32.869	0.52	−26.655	3.99
4dfr_med:	before min	−30.773	1.27	−24.269	2.81
	after min	−32.869	0.52	−27.439	2.83
4dfr_low:	before min	−30.828	0.64	−23.126	4.05
	after min	−32.869	0.52	−25.180	2.92
6rsa_high:	before min	−58.437	0.56	−52.256	5.51
	after min	−63.736	0.55	−57.381	4.83
6rsa_med:	before min	−43.017	1.96	−38.122	12.23
	after min	−61.381	0.96	−47.926	5.36
6rsa_low:	before min	−48.551	0.84	−27.459	8.48
	after min	−62.683	0.34	−57.381	4.83
2gbp_high:	before min	−21.485	0.29	−13.752	12.34
	after min	−23.007	0.38	−14.358	12.58
2gbp_med:	before min	−8.537	0.84	−13.327[e]	12.43
	after min	−19.267	0.30	−14.156	12.40
2gbp_low:	before min	42.252	0.62	−12.341[e]	12.70
	after min	−20.630	0.25	−14.156	12.40
3cpa_high:	before min	−41.303	1.17	−30.995	7.03
	after min	−46.929	0.64	−34.154	7.20
3cpa_med:	before min	−25.775	1.78	−30.652[e]	7.66
	after min	−38.935	0.76	−36.055	2.04[f]
3cpa_low:	before min	−22.925	1.84	−24.795[e]	7.63
	after min	−36.221	1.93	−31.594	6.87

[a]RMSD no greater than 2.0 angstroms.

[b]RMSD greater than 2.0 angstroms.

[c]Kcal/mol.

[d]Angstroms.

[e]Note that an "incorrect" orientation receives the best score.

[f]Arguably close to the observed binding mode; the top-scoring obviously incorrect orientation has a FF score of −33.342 kcal/mol and an RMSD of 8.24 angstroms.

CONCLUSIONS

There is a tradeoff between sampling and minimization; for the correct orientations to receive the best FF scores, intensive sampling is required, or moderate sampling combined with minimization. The tradeoff is not complete, however. Whether or not minimization is performed, sampling must be sufficient to find at least one structure in the vicinity of the true binding mode. The four test cases presented here provide valuable information since the binding geometries are known. In most applications, DOCK will be used to postulate geometries in the absence of such knowledge, with the added uncertainty of which molecular conformations are the most appropriate.

The sets of parameters in Table I are somewhat arbitrary, and indeed hardly begin to span parameter space. In general, increasing bin widths and overlaps will increase sampling, as quantified by *nmatch*. Setting the parameters to give an average *nmatch* of 10,000-20,000 is recommended for DOCK search runs. More intensive sampling can be afforded in DOCK single runs. Presently, it is much more time-efficient to sample orientations thoroughly than to combine low-to-moderate sampling with minimization (Table II). This may change if a faster minimization algorithm is implemented or if minimization of only a subset of the orientations is performed.

References

1. I. D. Kuntz, J. M. Blaney, S. J. Oatley, R. Langridge, and T. E. Ferrin, *J. Mol. Biol.*, **161,** 269 (1982).

2. B. K. Shoichet and I. D. Kuntz, *J. Mol. Biol.*, **221,** 327 (1991).

3. B. K. Shoichet, D. L. Bodian, and I. D. Kuntz, *J. Comp. Chem.*, **13,** 380 (1992).

4. E. C. Meng, B. K. Shoichet, and I. D. Kuntz, *J. Comp. Chem.*, **13,** 505 (1992).

5. F. C. Bernstein, T. F. Koetzle, G. J. B. Williams, E. F. Meyer, Jr., M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi, *J. Mol. Biol.*, **112,** 535 (1977).

6. E. E. Abola, F. C. Bernstein, S. H. Bryant, T. F. Koetzle, and J. Weng, in *Crystallographic Databases: Information Content, Software Systems, Scientific Applications,* F. H. Allen, G. Bergerhoff, and R. Seivers, Eds., Data Commission of the International Union of Crystallography, Bonn/Cambridge/Chester, 1987, pp. 107-132.

7. J. T. Bolin, D. J. Filman, D. A. Matthews, R. C. Hamlin, and J. Kraut, *J. Biol. Chem.*, **257,** 13650 (1982).

8. B. Borah, C.-W. Chen, W. Egan, M. Miller, A. Wlodawer, and J. S. Cohen, *Biochemistry,* **24,** 2058 (1985).

9. N. K. Vyas, M. N. Vyas, and F. A. Quiocho, *Science,* **242,** 1290 (1988).

10. Private communication from W. N. Lipscomb.

11. M. L. Connolly, *J. Appl. Crystallogr.*, **16,** 548 (1983).

12. M. L. Connolly, *Science,* **221,** 709 (1983).

13. J. M. Blaney, Ph.D. dissertation, University of California, San Francisco, 1982.

14. R. Fletcher, *Practical Methods of Optimization,* Interscience, 1960.

CHAPTER 4: APPROXIMATING DESOLVATION CONTRIBUTIONS TO BINDING USING ATOMIC HYDROPHOBICITIES

INTRODUCTION

In docking studies, molecular-mechanical interaction energies are useful for identifying reasonable binding modes; orientations that resemble experimental configurations can be singled out according to energy score.[1,2] Such estimates of complementarity are neither quantitative nor reliably predictive, however. Difficulties are especially likely when both species are macromolecules,[1] when the conformations of the molecules in the complexed state are unknown,[1] and when different ligands rather than different orientations of the same ligand are being compared.

The most significant obstacles to prediction in docking are also the fundamental problems encountered in molecular modeling: limitations in sampling and limitations in the accuracy of energy evaluation.[3] Conformational and orientational space must be explored adequately. Using DOCK on known complexes indicates that if the conformations present in the complexed state are known, orientational space can be sampled sufficiently to reproduce the experimental geometry.[1,2,4,5] Thus, conformational sampling has a greater tendency to be limiting than orientational sampling. As for evaluation, although a molecular-mechanical interaction energy is more sophisticated than a contact score, it still involves many assumptions, simplifications, and omissions.[2] Even when "correct" configurations have been generated, it may not be trivial to identify them. Incorrect configurations may be favored by a particular scoring method inasmuch as the method fails to produce accurate free energies. Difficulties are magnified in macromolecule-macromolecule docking, where the complexity of interaction is high, and

when different ligands rather than different orientations of the same ligand are being compared, where a fortuitous cancellation of errors is unlikely to occur.

In this chapter, I address one of the major shortcomings of using molecular-mechanical interaction energies as proxies for free energies of association: neglect of the partial desolvation that occurs upon binding. Accounting for solvation thermodynamics is essential for understanding biochemical processes as they occur *in vivo*; for example, one of two ligands may interact more strongly with a receptor molecule yet bind more weakly due to a greater desolvation cost. Desolvation processes may favor or disfavor binding, depending on whether the surfaces buried are predominantly hydrophobic or predominantly hydrophilic. I explore the use of atomic hydrophobicities to model desolvation contributions to binding energies.

BACKGROUND

The need to consider desolvation energies during scoring has become increasingly evident as DOCK search results become available for more and more target systems. In general, electrostatic scoring tends to favor highly charged molecules over those with few (or zero) formal charges. In a site with a slightly positive electrostatic potential, for example, polyanions receive the best scores rather than structures that match the site charge for charge. Although more reasonable molecules also score well, it is clear from the results and from first principles that including solvation/desolvation effects could improve the performance of DOCK in ranking ligands.

Three types of scores are available in DOCK 3.0:[2] the contact score, the "DelPhi electrostatic score," which uses receptor potential maps from the program DelPhi,[6,7] and the force field score, which approximates an AMBER[8,9] interaction energy. The contact

score represents, roughly, the steric component of the interaction enthalpy. The DelPhi score is purely electrostatic, while the force field score contains both steric and electrostatic terms; in each case, the electrostatic contribution to binding is obtained by multiplying ligand point charges by the local potential due to the receptor. Larger partial charges are clearly favored, as long as they are of the correct sign, no matter how small the potential is. The greater costs of desolvating larger charges are not considered. Nonetheless, these models do include some solvent effects (see below).

Water is a high-dielectric solvent; it influences the electrostatic enthalpy of a system by reducing charge self-energies and screening charge-charge interactions. This can be viewed as a distance-dependent attenuation of the potential due to each point charge. The attenuation, however, is a complex function of the distances among interacting points and the charges and polarizabilities of groups throughout the system. In addition, solvent water is not just a dielectric continuum, but a network of discrete, interacting molecules. Certain statistical contributions to the free energy can only be calculated with explicit water molecules. The hydrophobic effect, for example, is primarily entropic (statistical) at physiological temperatures[10] and plays a significant role in many binding equilibria of biochemical interest.

The continuum dielectric model used in DelPhi includes both screening and self-energy effects. The detailed shape of the solute(s) and, optionally, the presence of ions in solution, are included in the calculation. Self-energy reduction is one way of describing the favorable interactions between water and polar or charged solute groups. The unfavorable interactions between water and nonpolar solute groups are not described, however; this hydrophobic effect arises from the discrete nature of water. The most rigorous treatment within the continuum model involves the full thermodynamic cycle,

where electrostatic energies are determined for each docked complex as well as the solvated receptor alone and the solvated ligand alone.[11] The DelPhi score in DOCK employs the further assumption that the receptor electrostatic potential need only be calculated once, in the absence of ligands.[2] DOCK studies have been performed with potential maps of either the completely solvated receptor or the receptor plus the docking spheres, which are considered chargeless regions of low dielectric.[12] The former approach overestimates solvent screening in the docked complexes, while the latter underestimates it. Unfortunately, the errors are not simply those of scale (which could be corrected with a multiplicative factor).

Brian Shoichet has corrected for desolvation by subtracting the entire electrostatic solvation energy of each ligand from its DelPhi score.[12] He used a modified Born equation[13] to calculate the solvation energies. Qualitative success was attained; compounds that matched the site approximately charge for charge received the top scores, rather than small, multivalent ions. Nevertheless, I found this approach unsatisfying for several reasons. First, it would be more consistent to use the same algorithm, namely DelPhi, to calculate the potential maps and solvation energies. The use of different algorithms increases uncertainty about the relative scaling of terms. Second, it is crude to subtract the entire solvation energy from the score regardless of the geometry of the complex. This can never help distinguish among different orientations of the same ligand. Furthermore, even buried atoms can interact significantly with the solvent;[11] subtracting some fraction of the solvation energy should be more appropriate than subtracting the entire value, even when the ligand is completely engulfed by the receptor. Third, important components of binding energies are not included in the DelPhi score, corrected or uncorrected: van der Waals interactions and the hydrophobic effect.

In force field scoring, a dielectric function that depends linearly on the distance between interacting charges can be used to model screening due to the solvent. There is only a tenuous physical rationale for this practice,[14,15] yet it is convenient and in some cases produces good agreement with experiment.[16,17] Notably, a linear distance dependence of $4r$ or $4.5r$ corresponds closely to the sigmoidal function of Mehler and Eichele[18,19] at separations under 15-20 angstroms. Self-energies are ignored within the molecular mechanics formalism, so favorable interactions between water and polar or charged groups are not accounted for. Also, as in the continuum dielectric model, unfavorable interactions between water and nonpolar groups are not represented.

Ideally, a correction to the force field score would include desolvation contributions from nonpolar, polar, and charged groups alike; it should reflect the costs of burying specific atoms or groups, allowing different orientations of the same ligand to be distinguished. The method should be computationally expedient (for example, allowing separation of receptor and ligand terms so that the receptor component can be precalculated) and applicable to a wide range of structures. Finally, the correction should not include any ligand-receptor interaction terms, since they are already present in the force field score.

Solvation terms for use in combination with standard molecular mechanics equations, in lieu of explicit water molecules, have been proposed.[20,21] Other approaches use atomic solvation parameters derived from experiment.[22,23] Unfortunately, each method fails to meet one or more of the preceding criteria.

Gilson and Honig[20] propose a term that penalizes charges for being buried, as self energies are most unfavorable in a low-dielectric medium. The term does not model

screening of charge-charge interactions; presumably, it would be used in combination with standard molecular mechanics terms and a distance-dependent dielectric. The necessary parameters depend on environment as well as atom type, so extensive parameterization is required. Finally, receptor and ligand terms are not separable, and the hydrophobic effect is not included.

Still *et al.*[21] add two terms to a standard molecular mechanics equation. One is surface-area-based and intended to account for the hydrophobic effect and cavitation energy. There is no consideration of whether the atoms buried are polar or nonpolar. The other term is essentially the generalized Born equation; it encompasses the self-energy reduction and charge-charge screening effects of a high-dielectric solvent. Again, however, receptor and ligand components are not separable, and parameterization could be difficult. The Born radii are dependent on conformation and need to be recalculated as atomic positions change.

Eisenberg and McLachlan have developed atomic solvation parameters from the octanol/water partitioning behavior of amino acid side chain analogs.[22] Self-energy and hydrophobic effects are implicit since these are determinants of partitioning behavior. The parameters are multiplied by changes in the corresponding atomic solvent-accessible surface areas to obtain the change in solvation energy between two states of a system. As in my proposal (see below), the assumption is that the tendency to partition into octanol from water parallels the tendency for burial. Drawbacks are that few model compounds were used in the parameterization, so the values are intended for proteins and peptides only, and that surface area calculations are necessary for each configuration of the system.

Horton and Lewis take a similar approach, but their parameters are derived from the dissociation constants of 24 protein-protein complexes.[23] In addition, the parameters depend on whether or not an atom is involved in a hydrogen bond or salt bridge. The equation is meant to be complete, that is, to encompass interactions as well as solvation effects. As with the method of Eisenberg and McLachlan,[22] the parameters are based on and meant for use with proteins and peptides only, and surface area calculations are necessary for each configuration.

Numerous methods exist for calculating solvation energies, but they do not decompose the energies into contributions from specific atoms or groups. Since solvation energy is not an additive property, such a breakdown is actually a mental construct: a simplified representation of a system that may be useful or instructive. A widely used construct, for example, is the partial atomic charge model of molecules.

Besides solvation energy, another relevant observable is the partition coefficient (P), the ratio of concentration in an organic solvent to concentration in water of a compound equilibrated between the two phases. Data are available on the octanol-water partitioning behavior of a wide variety of organic molecules, and several methods have been developed for estimating logP values from structures.[24-27] The program CLOGP[28] uses the fragment-based algorithm of Hansch and Leo.[24] Furthermore, and key for my purposes, the related program HINT[29,30] decomposes the predicted logP into atomic contributions. Assuming that solutes interact with octanol in a primarily nonspecific manner, differences in logP should reflect differences in interactions with water, that is, differences in solvation energy. There is a definite precedent for relating the octanol-water partitioning behavior of groups to their effects on binding; the logP-derived descriptor π is commonly used in quantitative structure-activity relationships (QSAR's).[24] I use

HINT-calculated atomic contributions to logP as well as a simple scale based on element alone to represent atomic hydrophobicity.

COMPUTATIONAL METHODS

The octanol-water partition coefficient is an equilibrium constant, and is related to the free energy of transfer from water to octanol as follows:

$$\Delta G_{transfer} = -RT(lnP) = -2.303RT(logP) \quad (1)$$

R is the gas constant and T is the absolute temperature. Assuming that the logP of a molecule can be broken down into atomic contributions α,

$$logP = \sum_{i=1}^{atoms} \alpha_i \quad (2)$$

$$\Delta G_{transfer,i} = -2.303RT\alpha_i \quad (3)$$

and

$$\Delta G_{transfer} = \sum_{i=1}^{atoms} \Delta G_{transfer,i} = -2.303RT \sum_{i=1}^{atoms} \alpha_i \quad (4)$$

The free energy of binding, neglecting changes in conformational energies, depends on the specific interactions between the molecules and the costs of partially desolvating them:

$$\Delta G_{bind} = \Delta G_{bind,interaction} + \Delta G_{bind,desolvation} \quad (5)$$

Using partitioning from water into octanol to approximate the desolvation that occurs upon binding (but not the interactions between the complexing species),

$$\Delta G_{bind,desolvation} \approx -2.303RT \sum_{i=1}^{atoms} f_{buried,i}\alpha_i \quad (6)$$

where $f_{buried,\,i}$ is the fraction of the interaction of atom i with water lost upon complexation. Since this desolvation term is empirically derived from partitioning rather than solvation processes, the appropriate scaling factor is uncertain; the coefficient merely

converts it into a partitioning energy, which should be a reasonable starting point for investigation.

The DOCK 3.0 force field score has been used to approximate $\Delta G_{bind,\ interaction}$.[2] It consists of van der Waals (VDW) and electrostatic terms:

$$\Delta G_{bind,interaction} \approx \sum_{i=1}^{lig} \sum_{j=1}^{rec} \left[\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} + 332.0 \frac{q_i q_j}{D r_{ij}} \right] \quad (7)$$

Each term is a double sum over ligand atoms i and receptor atoms j, A_{ij} and B_{ij} are VDW repulsion and attraction parameters, r_{ij} is the distance between atoms i and j, q_i and q_j are the point charges on atoms i and j, D is the dielectric function, and 332.0 is a factor that converts the electrostatic energy into kcal/mol. Combining eqs. 5, 6, and 7 yields an approximate total free energy of binding:

$$\Delta G_{bind} \approx \sum_{i=1}^{lig} \sum_{j=1}^{rec} \left[\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} + 332.0 \frac{q_i q_j}{D r_{ij}} \right] - 2.303RT \sum_{k=1}^{atoms} f_{buried,k} \alpha_k \quad (8)$$

I also examine the use of simple, element-based hydrophobicities β:

$$\Delta G_{bind} \approx \sum_{i=1}^{lig} \sum_{j=1}^{rec} \left[\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} + 332.0 \frac{q_i q_j}{D r_{ij}} \right] - \sum_{k=1}^{atoms} f_{buried,k} \beta_k \quad (9)$$

Within this general framework for scoring, determining f_{buried} is the technical bottleneck. There is no measure that clearly represents the extent of an atom's interaction with water. Although the solvent-accessible surface area is often used, it is not the definitive measure; atoms without exposed surface area can still interact significantly with the solvent.[11] I use another imperfect measure, related to intermolecular dispersion interactions. During the calculation of the force field grids, three values are stored for every grid point m, each a sum over receptor atoms:

$$aval = \sum_{j=1}^{rec} \frac{\sqrt{A_{jj}}}{r_{jm}^{12}} \quad bval = \sum_{j=1}^{rec} \frac{\sqrt{B_{jj}}}{r_{jm}^{6}} \quad esval = 332.0 \sum_{j=1}^{rec} \frac{q_j}{Dr_{jm}} \tag{10}$$

The steric parameters A_{jj} and B_{jj} for receptor atom j are calculated from the VDW radius R and well depth ε according to:

$$A = \varepsilon \left[2R\right]^{12} \quad \text{and} \quad B = 2\varepsilon \left[2R\right]^{6} \tag{11}$$

The grid values, with or without interpolation, are multiplied by the appropriate ligand values to give the force field score:

$$\Delta G_{bind,interaction} \approx \sum_{i=1}^{lig} \left[\sqrt{A_{ii}} \left[aval\right] - \sqrt{B_{ii}} \left[bval\right] + q_i \left[esval\right] \right] \tag{12}$$

I use the quantity *bval* to describe the degree of burial of a ligand atom:

$$f_{buried} = \begin{cases} \dfrac{bval}{blim} & \text{for } bval < blim \\ 1 & \text{for } bval \geq blim \end{cases} \tag{13}$$

where *blim* is a cutoff value defining complete burial of a ligand atom. I use *blim* = 0.10 based on *bval* calculated for ligand atoms in experimental complex structures; values of 0.10 or greater are obtained for ligand atoms that are completely engulfed by the receptor. This measure of burial is convenient since *bval* is evaluated in the course of calculating force field scores. Using *bval* is preferable to using intermolecular dispersion energy since it is independent of ligand atom type (though dependent on receptor atom types), and preferable to using the total VDW interaction energy since, for each pair of interacting atoms, it is a monotonic function of the distance between them (and thus more appropriate for use with a cutoff value).

There is no similarly expedient descriptor for the burial of receptor atoms by the ligand. As a first approximation, I only correct for ligand desolvation.

DOCK 3.0[2] was slightly modified to use HINT-calculated atomic hydrophobicities according to eq. 8 ("DOCK3HINT") and simple hydrophobicities according to eq. 9

("DOCK3SIMP"). Each version allows independent scaling of the VDW, electrostatic, and desolvation components of the score. HINT-calculated atomic hydrophobicities were incorporated into a modified DOCK 3.0 database format; simple hydrophobicities were included in an altered DOCK 3.0 VDW parameter file: 1.0 for C, −1.5 for N and O, and 0.0 for all other atoms.

TEST SYSTEM

One goal of including a solvation/desolvation term is improving the ability of DOCK to rank compounds according to their binding affinities for a receptor. Although it is unreasonable to expect any computational method (with the possible exception of free energy perturbation) to make fine distinctions, significant improvements in DOCK rankings may be possible, especially in comparing ligands with widely varying affinities and differing numbers of formal charges.

Chymotrypsin was chosen as the test receptor for multiple reasons: its structure is known, hydrophobicity is important in ligand binding, affinities have been reported for numerous ligands and range over several orders of magnitude, and many of the ligands are rigid. Finally, inspiration came from the application of DOCK 1.1 to this system by the Burroughs Wellcome group.[31]

The Brookhaven Protein Data Bank[32,33] (PDB) structure 4cha[34] has been refined at 1.68 angstroms resolution. Two independent molecules of α-chymotrypsin are contained in the asymmetric unit, and each consists of three chains produced by the excision of residues 14-15 and 147-148 from the 245-residue zymogen. Residues 12-13 of molecule "A" and residues 11-13 of molecule "B" are disordered and not included in the structure. The molecule designated "A" was used. Cynthia Corwin calculated a molecular

surface[35] with the Connolly MS algorithm[36,37] and used it in SPHGEN[4] to yield a 97-sphere cluster in the active site. With the program CLUSTER[1,5] and manual editing, she generated a smaller cluster of 47 spheres. Figure 1 shows the α-carbon trace of molecule "A" together with the sphere centers for docking and the box outlining the force field scoring grid.

The various scoring grids are described at length in Chapter 2,[2] so only the essential parameters are given here. Contact-scoring grids were created in DISTMAP with a spacing of three points per angstrom, a cutoff for interactions of 4.5 angstroms, and a polar contact limit of 2.3 angstroms. Two different grids, "CONT1" and "CONT2," were generated with nonpolar contact limits of 2.6 and 2.8 angstroms. For the force field grid (CHEMGRID) calculation, hydrogens were added to 4cha in standard geometries. I modeled the N-termini at residues 1 and 149 as positively charged and the C-termini at 146 and 245 as negatively charged; histidine side chains were protonated to the positively charged state. The calculation used the entire receptor with AMBER united-atom partial charges and VDW parameters,[8] 0.3-angstrom spacing, a 10.0-angstrom cutoff, and $D = 4r$. Close contact limits in CHEMGRID were 2.3 and 2.6 angstroms for receptor polar and nonpolar atoms, respectively.

Affinities of over a hundred aromatic compounds for α-chymotrypsin have been determined.[39] With a combination of name and substructure searching, Cynthia Corwin retrieved more than 70 of the compounds from the Fine Chemicals Directory[40,41] (FCD version 89.2). Structures for these compounds had been generated with CONCORD.[42,43] I restricted the ligand database to 74 highly rigid compounds (Table I): 58 structures from the FCD and 16 others built using the modeling package SYBYL.[44] Multiple con-

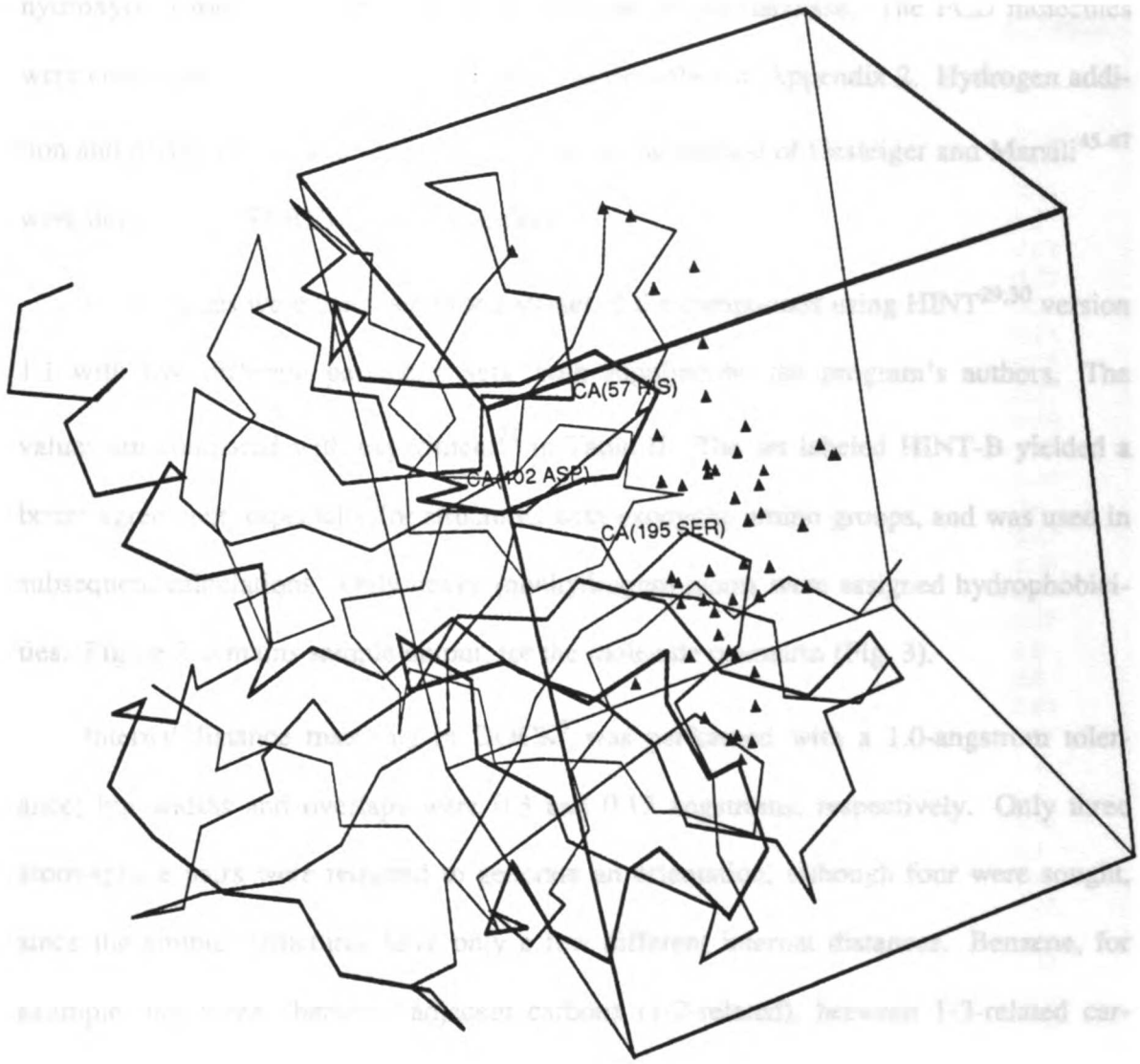


Figure 1. The 4cha[34] test system: An α-carbon trace of molecule "A," the sphere centers used for docking (triangles), and a box outlining the force field grid. The catalytic triad residues are labeled; in this view, the lowest sphere centers occupy the specificity pocket. Picture generated with UCSF MidasPlus:[38] Molecular Interactive Display and Simulation, Computer Graphics Laboratory, Department of Pharmaceutical Chemistry, University of California, San Francisco, CA 94143-0446.

formations were constructed for a few of the compounds, mainly those with rotatable hydroxyls; a total of 90 structures were included in the database. The FCD molecules were converted into SYBYL MOL2 format as described in Appendix 2. Hydrogen addition and partial charge calculations according to the method of Gasteiger and Marsili[45-47] were done within SYBYL for all structures.

LogP values were calculated for a subset of the compounds using HINT[29,30] version 1.1 with two different parameter sets, both supplied by the program's authors. The values are compared with experiment[24] in Table II. The set labeled HINT-B yielded a better agreement, especially for structures with exocyclic amino groups, and was used in subsequent calculations. Only heavy (nonhydrogen) atoms were assigned hydrophobicities. Figure 2 contains sample output, for the molecule coumarin (Fig. 3).

Internal-distance matching in DOCK[5] was performed with a 1.0-angstrom tolerance; bin widths and overlaps were 0.3 and 0.15 angstroms, respectively. Only three atom-sphere pairs were required to generate an orientation, although four were sought, since the simpler structures have only a few different internal distances. Benzene, for example, has three: between adjacent carbons (1-2-related), between 1-3-related carbons, and between 1-4-related carbons. One bad contact per orientation was allowed, and one level of zooming was specified. Ten different variations on force field scoring were applied (Table III): two with DOCK 3.0, four with DOCK3HINT, and four with DOCK3SIMP. All-atom descriptions of the ligands were used; force field grid values were interpolated. Each run on the database of 90 structures took between 25 and 30 minutes on a Silicon Graphics Iris 4D/25 workstation. Two additional DOCK 3.0 runs were done with contact scoring only; these took approximately 10 minutes on the same workstation.

Table I. Chymotrypsin inhibitors and inhibitory constants.[a]

Compound	K_i (mM)
BENZO[F]QUINOLINE	0.063
ACRIFLAVINE	0.08
PROFLAVINE	0.13
1-NAPHTHOL	0.2
ACRIDINE	0.22
2-AMINOACRIDINE	0.22
BENZO[C]QUINOLINE	0.23
3-AMINOACRIDINE	0.23
BIPHENYL-4-OL	0.25
BETA-NAPHTHYLAMINE	0.25
1-NAPHTHYLAMINE	0.30
ISOQUINOLINE	0.32
1-AMINOACRIDINE	0.34
QUINOLINE	0.6
COUMARIN	0.67
BENZO[H]QUINOLINE	0.70
7-METHYLQUINOLINE	0.7
8-QUINOLINOL	0.77
1-METHYLINDOLE	0.8
INDOLE	0.8
2-QUINOLINOL	0.87
1-METHYL-2-INDOLINONE	0.87
4-AMINOQUINOLINE	1.1
2-AMINOQUINOLINE	1.3
7-AZAINDOLE	1.33
NAPHTHORESORCINOL	1.4
2-NAPHTHOIC_ACID	1.4
PHTHALIDE	1.42
2-METHYLQUINOLINE	1.5
2-NAPHTHALENESULFONATE	1.84
1-INDANONE	1.88
PHTHALIMIDINE	2.02
3-AMINOQUINOLINE	2.3
4-METHYLQUINOLINE	2.3
1,3-INDANDIONE	2.4
NINHYDRIN	2.7
4-AMINOPYRIDINE	2.9
PHTHALAZONE	2.95
BENZIMIDAZOLE	3
NN-DIMETHYLANILINE	3.4
FORMANILIDE	3.9
CRESOL_RED	4.67
1-NAPHTHYLAMINE-6-SULFONATE	4.8
QUINOXALINE	5
BENZIMIDAZOLE-2-CARBOXYLATE	5.4
N-METHYLANILINE	6.3
PHENOL	6.4
ANILINE	6.6
ANISOLE	8.4
QUINOLINE-4-CARBOXAMIDE	8.4

Compound	K_i (mM)
2-AMINOPYRIDINE	9.4
2,4,6-TRIMETHYLPYRIDINE	10
BENZAMIDE	10
FLUORESCEIN	10.2
2-NAPHTHYLAMINE-6-SULFONATE	11
3-AMINOPYRIDINE	12.3
TOLUENE	13
ACETANILIDE	13
1,10-PHENANTHROLINE	15.1
BENZENE	25
PYRIDINE	28
1-NAPHTHYLAMINE-5-SULFONATE	31
2-NAPHTHYLAMINE-1-SULFONATE	41
IMIDAZOLE	45
BENZENESULFONIC_ACID	70
4-QUINOLINECARBOXYLIC_ACID	104
2-PYRIDOL	110
2,6-NAPHTHALENEDISULFONATE	130
BENZOIC_ACID	150
8-QUINOLINESULFONIC_ACID	177
1-NAPHTHYLAMINE-4-SULFONATE	185
2,7-NAPHTHALENEDISULFONATE	400
1,5-NAPHTHALENEDISULFONATE[b]	
1,6-NAPHTHALENEDISULFONATE[b]	

[a]Reference 39.

[b]Inactive.

Table II. Comparison of HINT-calculated and experimental logP values.

Compound	HINT-A[a]	HINT-B[a]	experiment[b]
PHENOL	1.460	1.460	1.46
BIPHENYL-4-OL	3.330	3.330	3.20
BENZENE	2.130	2.130	2.15
FORMANILIDE	0.386	0.386	1.12
1,3-INDANDIONE	0.684	0.684	0.61
NINHYDRIN	−1.781	−1.781	0.65
1-NAPHTHOL	2.620	2.620	2.98
ACRIDINE	3.405	3.405	3.40
2-AMINOPYRIDINE	−0.813	0.411	0.58
3-AMINOPYRIDINE	−2.019	−0.328	0.20
4-AMINOPYRIDINE	−1.724	−0.384	0.26
QUINOXALINE	0.926	0.926	1.32
QUINOLINE	2.030	2.030	2.03
2-METHYLQUINOLINE	4.580	4.580	2.59
3-AMINOQUINOLINE	−0.749	0.942	1.63
7-METHYLQUINOLINE	4.580	4.580	2.47
8-QUINOLINOL	1.681	1.681	1.96
COUMARIN	1.860	1.860	1.39
ISOQUINOLINE	1.815	1.815	2.09
ANILINE	−2.325	0.590	0.90
BENZAMIDE	−0.687	−0.687	0.64
N-METHYLANILINE	1.525	1.525	1.82
NN-DIMETHYLANILINE	2.977	2.977	2.31
ANISOLE	2.180	2.180	2.08
TOLUENE	2.790	2.790	2.73
ACETANILIDE	0.780	0.780	1.16
PYRIDINE	0.655	0.655	0.64
2-AMINOQUINOLINE	0.478	1.702	1.87
4-AMINOQUINOLINE	−0.640	0.701	1.63
INDOLE	1.007	0.980	2.00
1-AMINOACRIDINE	0.415	2.106	2.47
2-AMINOACRIDINE	−0.093	2.035	2.62
3-AMINOACRIDINE	0.464	1.980	2.19

[a]HINT version 1.1 was used with two different parameter sets.

[b]Reference 24.

Drug: COUMARIN

Atom	Hydrophobicity	Polar Hydrophob.	Slvnt Accss. Srf. Area
1 C (602)	0.125	0.000(0)	24.61
2 C (602)	0.225	0.000(0)	21.47
3 O (802)	-0.685	0.000(7)	29.96
4 C (607)	0.355	0.000(0)	40.89
5 C (607)	0.355	0.000(0)	40.89
6 C (607)	0.355	0.000(0)	40.89
7 C (602)	1.980	-0.620(7)	25.94
8 C (607)	0.355	0.000(0)	39.21
9 C (607)	0.355	0.000(0)	40.89
10 C (607)	0.355	0.000(0)	39.21
11 O (801)	-1.915	0.000(7)	40.78
TOTAL:	1.860		

Figure 2. Sample output from HINT.[29,30] The α values are in the column labeled "Hydrophobicity." "TOTAL" indicates the calculated logP of the molecule, coumarin.

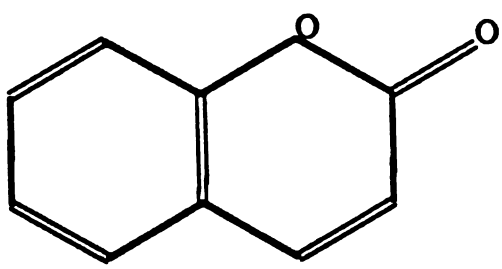


Figure 3. The structure of coumarin. Figure generated with SYBYL.[44]

RESULTS AND DISCUSSION

The correlation between apparent binding energy and score was examined graphically (Figs. 4-15) and by linear regression (Table III). Only the best-scoring conformation of a compound is included in each graph and regression analysis.

There is essentially no correlation between activity and score using the standard force field (FF) score (score 1, Fig. 4) or just its VDW component (score 2, Fig. 5). Combining the HINT term with the FF score improves the correlation (score 3, Fig. 6), especially as the term is doubled and tripled (score 4, Fig. 7, and score 5, Fig. 8). The HINT term alone (score 6, Fig. 9) yields a correlation greater than the FF plus 1HINT score but smaller than the FF plus 2HINT or 3HINT scores. A correlation coefficient of nearly 0.40 is obtained with the FF plus 3HINT score (Table III, score 5).

The simple hydrophobicity term (SIMP) gives surprisingly high correlations, either in combination with the FF score (scores 7-9, Figs. 10-12) or alone (score 10, Fig. 13). Alone, this term yields a correlation coefficient of 0.60; the FF plus 1SIMP, 2SIMP, and 3SIMP scores give correlation coefficients of 0.43, 0.62, and 0.68, respectively (Table III).

It must be emphasized that all of the results in Table III depend on the force field grids, even the results for scoring functions that do not include FF terms. Orientations that make more than one bad contact (as defined by the limits set in CHEMGRID and subject to the grid approximation) are thrown out. The number of bad contacts allowed is specified in the input to DOCK. Although FF score alone does not correlate with apparent binding energy in this system, the information in the FF grids is crucial for ruling out orientations that intersect the receptor.

Table III. The correlation between apparent binding energy[a] and score using different scoring functions.

	scale factor of term in the score				results of linear regression		
	ES[b]	VDW[c]	HINT[d]	SIMP[e]	slope	intercept	*r*
1	1	1	0	0	−0.0431	−22.0	−0.01608
2	0	1	0	0	0.0938	−21.3	0.03641
3	1	1	1	0	0.703	−21.2	0.2316
4	1	1	2	0	1.39	−20.6	0.3460
5	1	1	3	0	2.05	−20.3	0.3966
6	0	0	1	0	0.363	−1.71	0.3029
7	1	1	0	1	1.48	−22.5	0.4307
8	1	1	0	2	3.06	−23.0	0.6188
9	1	1	0	3	4.57	−23.9	0.6788
10	0	0	0	1	1.25	−2.94	0.6026

[a]$RT\ln(K_i) = 0.592424\ln(K_i)$ kcal/mol at 298.15 K.

[b]The electrostatic part of the force field score (the third term in eq. 7).

[c]The VDW part of the force field score (the first two terms in eq 7).

[d]The desolvation term using HINT atomic hydrophobicities (the last term in eq. 8).

[e]The desolvation term using simple atomic hydrophobicities (the last term in eq. 9).

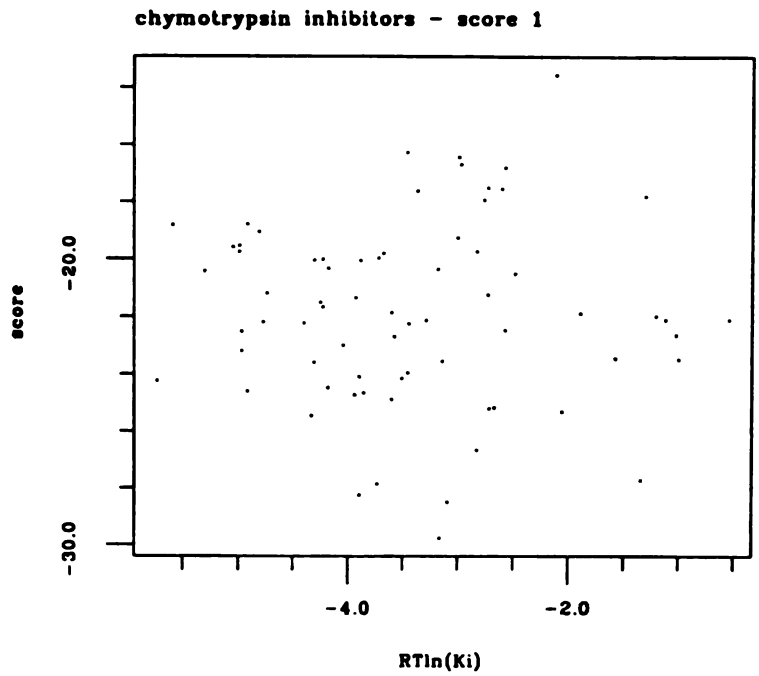


Figure 4. RTln(K_i) versus score 1, ES + VDW (the FF score). Linear regression yields slope –0.0431, intercept –22.0, and r = –0.01608.

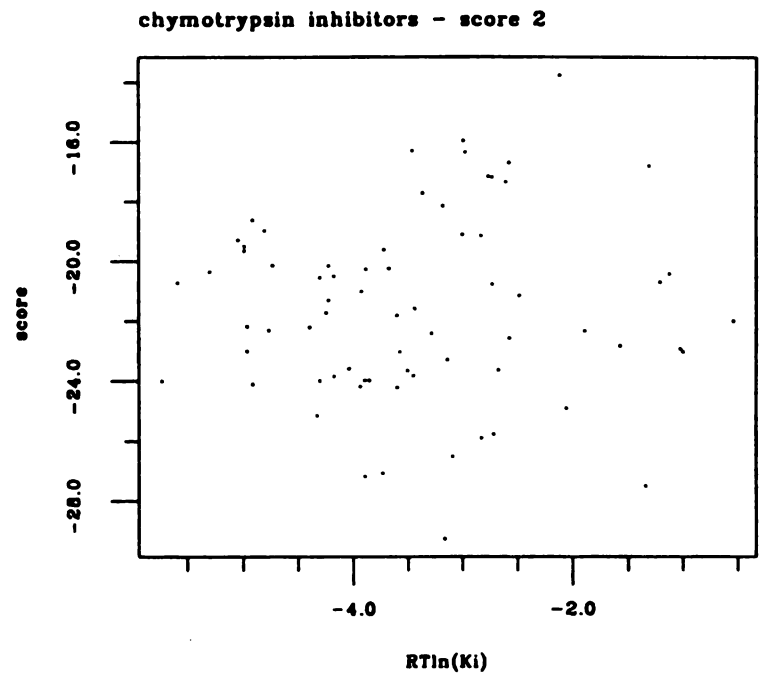


Figure 5. RTln(K_i) versus score 2, VDW. Linear regression yields slope 0.0938, intercept –21.3, and r = 0.03641.

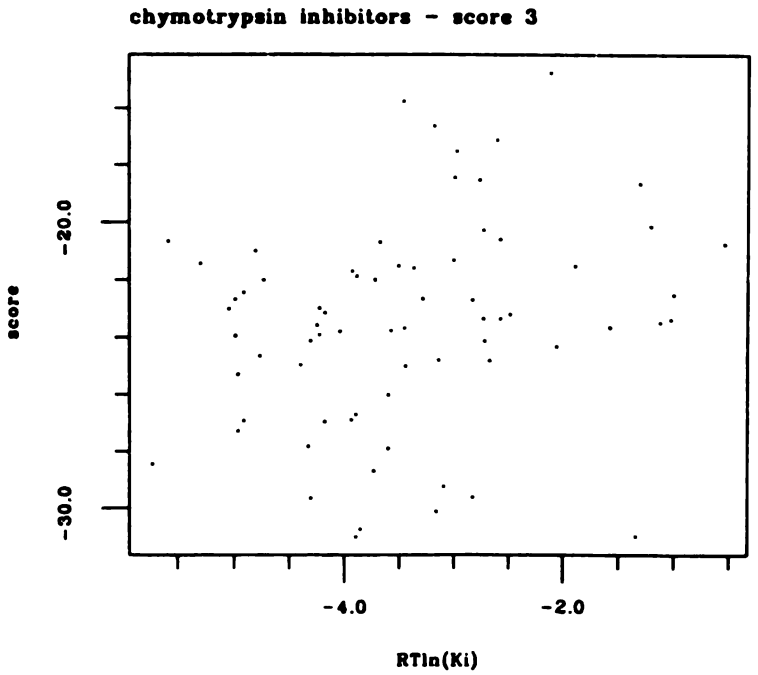


Figure 6. RTln(K_i) versus score 3, ES + VDW + HINT. Linear regression yields slope 0.703, intercept –21.2, and $r = 0.2316$.

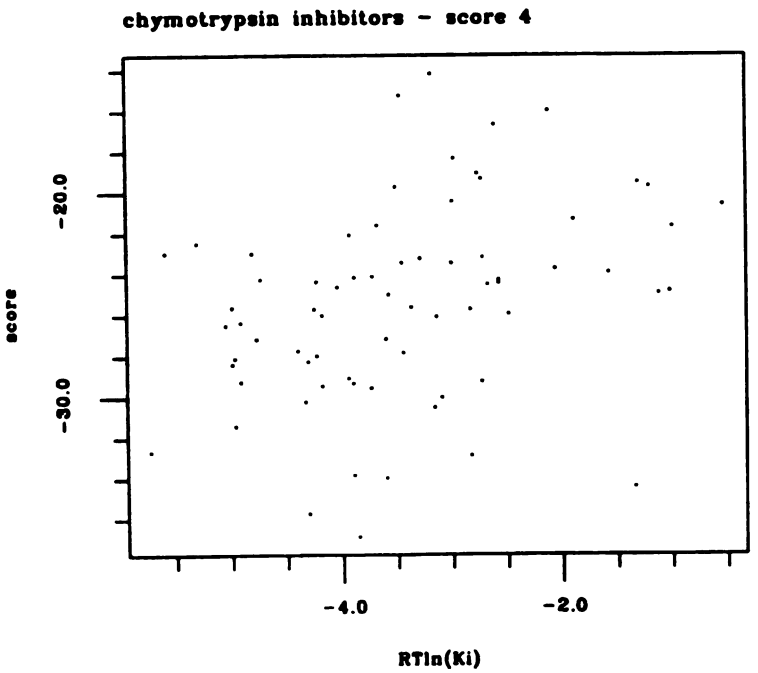


Figure 7. RTln(K_i) versus score 4, ES + VDW + 2HINT. Linear regression yields slope 1.39, intercept –20.6, and $r = 0.3460$.

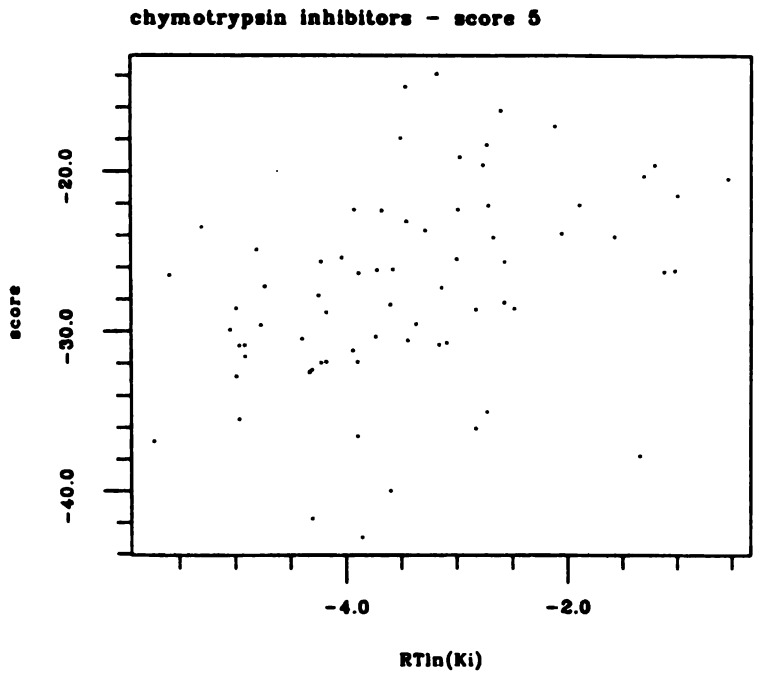


Figure 8. RTln(K_i) versus score 5, ES + VDW + 3HINT. Linear regression yields slope 2.05, intercept −20.3, and $r = 0.3966$.

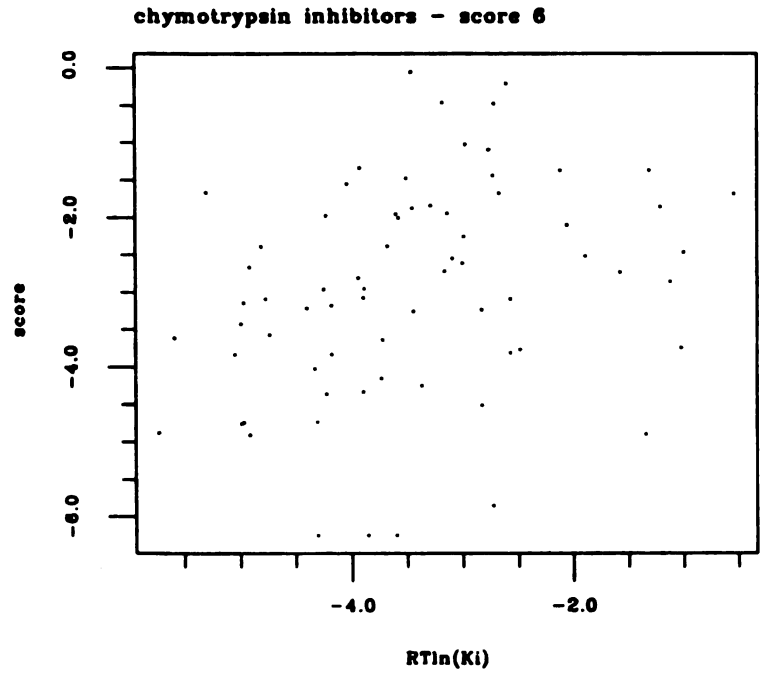


Figure 9. RTln(K_i) versus score 6, HINT. Linear regression yields slope 0.363, intercept −1.71, and $r = 0.3029$.

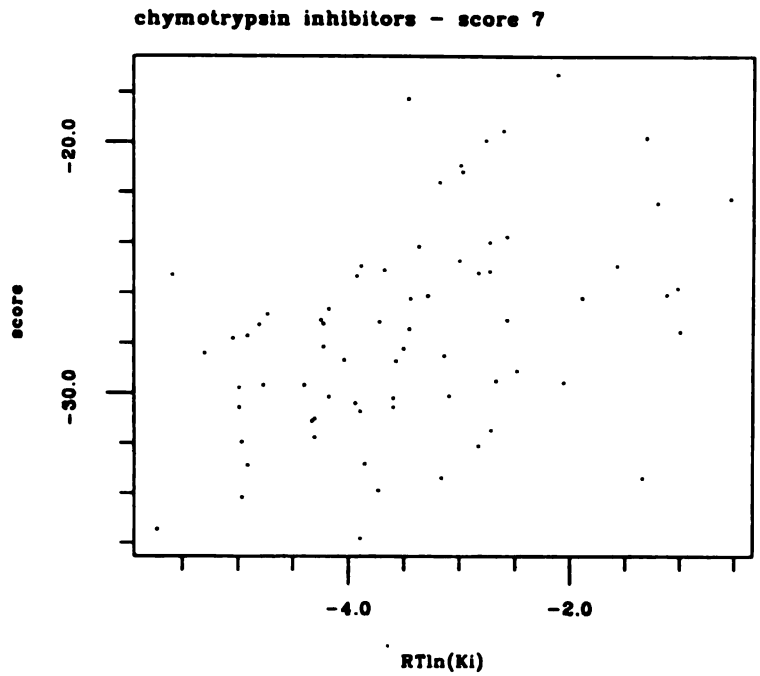


Figure 10. RTln(K_i) versus score 7, ES + VDW + SIMP. Linear regression yields slope 1.48, intercept −22.5, and $r = 0.4307$.

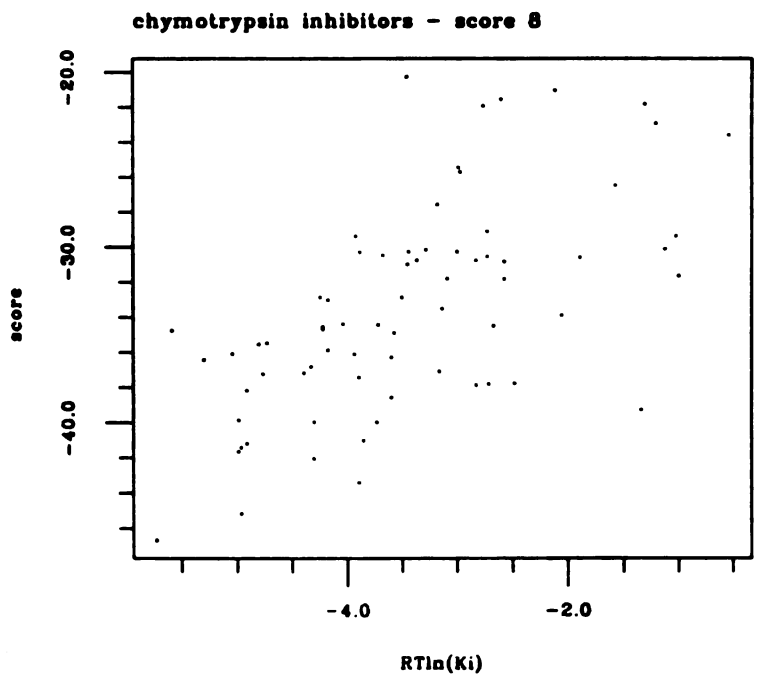


Figure 11. RTln(K_i) versus score 8, ES + VDW + 2SIMP. Linear regression yields slope 3.06, intercept −23.0, and $r = 0.6188$.

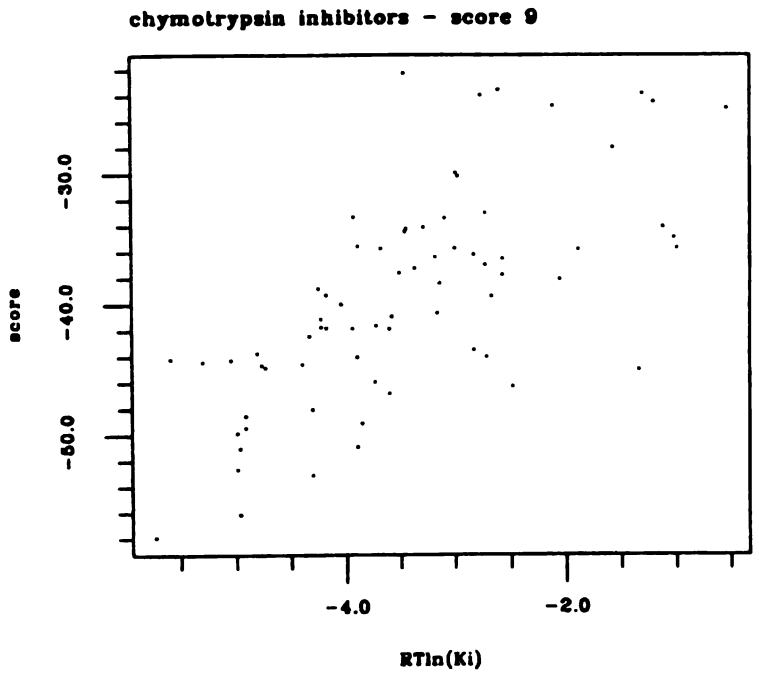


Figure 12. RTln(K_i) versus score 9, ES + VDW + 3SIMP. Linear regression yields slope 4.57, intercept −23.9, and $r = 0.6788$.

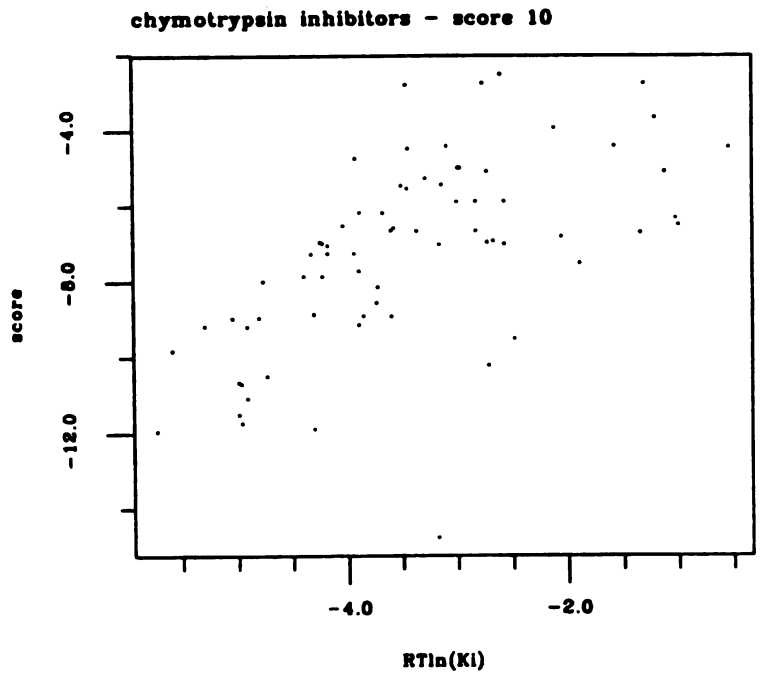


Figure 13. RTln(K_i) versus score 10, SIMP. Linear regression yields slope 1.25, intercept −2.94, and $r = 0.6026$.

The CONT1 and CONT2 shape scores (Figs. 14 and 15) yield better correlations than the FF score alone (score 1) or its VDW component (score 2), but worse correlations than nearly all scores involving a hydrophobicity term (Table III). Bad contacts found in the shape-scoring runs are defined by the limits set in DISTMAP and subject to the grid approximation. Although low in both cases, the correlation is higher using CONT2 than using CONT1, possibly because 2.8 angstroms is a more realistic nonpolar close contact limit than 2.6 angstroms. Whereas scores 1-10 give roughly equal numbers of false negatives and false positives, points in the upper left and lower right corners of the graphs, respectively (Figs. 4-13), the shape scores give many more false positives than false negatives. Shape scores favor the tightest intermolecular packing possible without violations of the limit on the number of bad contacts. As the limiting distances are fairly low and no penalty is imposed for the allowed bad contacts, this setup may be overly permissive of ligands that are slightly too large. Furthermore, contact scoring will not detect bad electrostatic interactions or rule out compounds that are too polar.

The two compounds listed as inactive at the end of Table I are not included in the graphs and regressions; it is interesting to see how they are ranked by the various scores. 1,5-naphthalenedisulfonate is ranked first by each of the contact scores! 1,6-naphthalenedisulfonate is ranked 22nd by the CONT1 contact score, 37th by the FF score alone (score 1), and 48th by the VDW component of the FF score (score 2). In all other cases, these compounds are not among the top 50.

If the experimental data were highly accurate, the bound conformations of both species were known, and a scoring method had been devised to accurately produce free energies of association under the same conditions as the experiments, a linear regression of score versus $RT\ln(K_i)$ would give slope = 1, intercept = 0, and $r = 1$. The simple

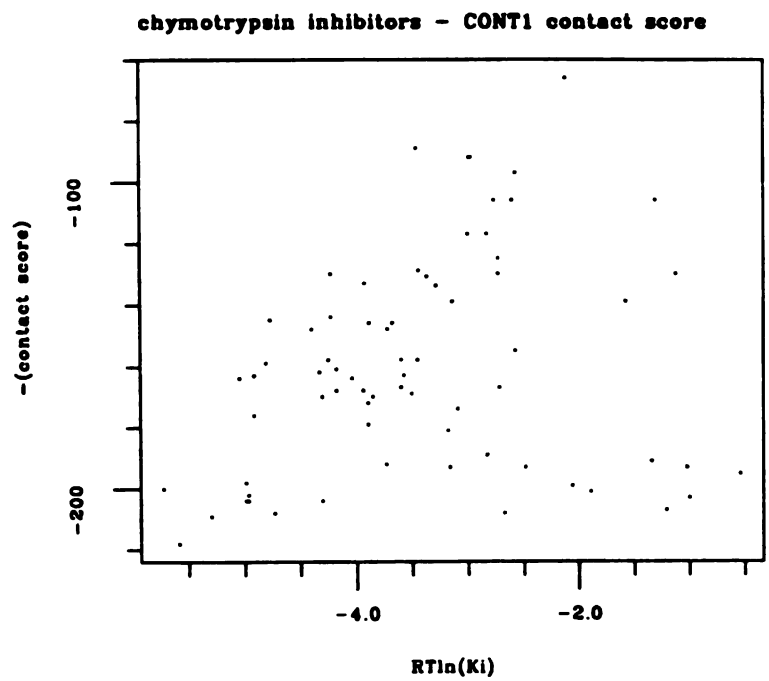


Figure 14. RTln(K_i) versus –(contact score) using CONT1. Linear regression yields slope 6.28, intercept –139, and $r = 0.2134$.

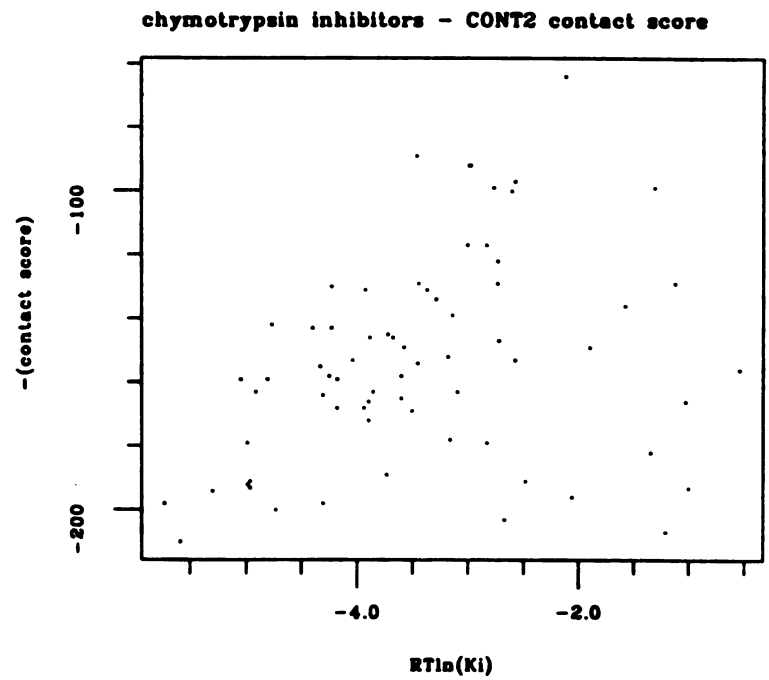


Figure 15. RTln(K_i) versus –(contact score) using CONT2. Linear regression yields slope 7.96, intercept –126, and $r = 0.2964$.

schemes tested here obviously fall far from this ideal (Table III). The correlation coefficient is the most important of the regression results, since it reflects the ability of a score to rank compounds by affinity; slope and intercept have little or no meaning without a correlation. My goal is to improve correlations between affinity and score, rather than to calculate the free energy of binding *per se*. Considering the approximations made, including the lack of explicit solvent, it is unreasonable to expect free energies of association to be obtained. Rather, I find it encouraging and sometimes even surprising that geometric docking combined with rapid energy estimation has been so helpful, both qualitatively and semiquantitatively.

Where there appears to be a correlation, slopes range from less than 1 to greater than 4 and intercepts range from –2 to –24 (Table III). Scores 6 and 10, which do not include force field terms, yield fairly low slopes and intercepts close to 0. There are several reasons why regression results might differ from the ideal values given above. Overestimating mainly the highest (worst) energies and underestimating mainly the lowest (best) energies would increase the slope, whereas overestimations or underestimations "across the board" would not affect the slope but would raise or lower the intercept, respectively. Overestimations mixed with underestimations would increase the noise and decrease any apparent correlation. The HINT and SIMP terms are very rough, and even when they are included, several factors in binding are still neglected: conformational energies, desolvation of the receptor, and entropy loss upon binding, to name a few. The use of a distance-dependent dielectric function may introduce errors of either sign in electrostatic contributions,[15] and the VDW part of the FF score overestimates the steric costs of forming borderline close contacts.[2] Any borderline close contacts may be *artifactual*, resulting from an incorrect binding mode or incorrect conformations, or

actual, where the ligand in question is not capable of optimally fitting into the site. Examination of the docked orientations shows that they occupy the region expected based on biochemical evidence, namely the active site and in particular the hydrophobic specificity pocket (Fig. 1). Naturally, the sphere cluster used for docking was constructed to occupy this region. Conformational uncertainty was decreased by restriction of the ligand database to rigid compounds.

The test system and experimental data are not without flaws. Although orders of magnitude in affinity are spanned by the ligands in the database, not one has a submicromolar inhibitory constant (K_i);[39] nonspecific binding or at least multiple binding modes may be significant in the inhibition of chymotrypsin by these compounds. In addition, few data points per compound (sometimes only one) were used in estimating the K_i values. However, no other systems with as many attributes favorable for testing a desolvation/hydrophobicity term have come to my attention. It is rare to have experimental affinities for so many rigid molecules that bind to a receptor of known structure, especially ligands that vary significantly in affinity and number of formal charges.

The success of the SIMP term raises some questions. First, does the term merely favor carbons over polar atoms in a nonspecific way? In light of the possibility of nonspecific inhibition, both reality and the calculations may be favoring the same thing: hydrophobicity. This could also apply to scores including the HINT term. The various desolvation corrections need to be tested in more systems, including those in which hydrophobicity is not the dominant factor in molecular recognition. Second, is the HINT calculation unneccessarily complicated for this application? Correlations are significantly higher for scores including the SIMP term than for scores including the HINT term (Table III). A comparison of the SIMP term alone (score 10) and the HINT

term alone (score 6) demonstrates that the larger correlations are not merely due to the SIMP term being more effective than the HINT term at overwhelming the force field contribution. The greater complexity of generating HINT atomic hydrophobicities as compared to the SIMP assignment, besides taking more time and effort, increases opportunities for the introduction of errors and artifacts. The HINT results depend on the handling of polar proximity effects[29,30] and on the SYBYL atom type assignments. Agreement with experimental logP values has been encouraging, for the most part, but important uncertainties remain about how a logP should be split up amongst atoms. As stated earlier, such a breakdown is a mental construct; what decomposition is the most appropriate depends on how the atomic values will be used. The surface atoms are the most important in this application and probably should account for most of the logP, but this is not always true of the HINT-calculated values. Third, how were the SIMP assignments chosen, and would using a different set change the results significantly? The assignments were chosen to roughly resemble the HINT-calculated values in terms of signs and relative magnitudes; no systematic optimization was attempted. Results were similar for similar assignments (data not shown; the studies were performed on a different system, using a database of 200 structures, predominantly aromatic cyclic compounds and their derivatives).

CONCLUSIONS

One of the major shortcomings of estimated interaction energies as proxies for free energies of association is neglect of the partial desolvation that occurs upon binding. I have used ligand atom hydrophobicities and degrees of burial to estimate desolvation contributions to binding. Atomic contributions to logP calculated with HINT[29,30] as well as a simple element-based assignment have been investigated as measures of hydrophobicity. In the α-chymotrypsin system, desolvation terms improve the correlation between apparent binding energy and score. The simple hydrophobicities are more successful than the HINT-derived hydrophobicities in this implementation. However, it is unclear whether the improved agreement with experiment is primarily due to a nonspecific selection for greater hydrophobicity. The approach needs to be evaluated in different systems where hydrophobic interactions do not necessarily dominate molecular recognition. It is a challenge to find systems for which the receptor structure is known, binding affinities are available for scores of diverse ligands, and the occurrence of confounding uncertainties such as ligand flexibility is low.

Acknowledgements. I would like to thank Cynthia Corwin for pioneering the chymotrypsin work and helping to acquaint me with the system. Thanks are due also to Glen Kellogg and Donald Abraham for providing HINT and generous amounts of assistance and advice.

References

1. B. K. Shoichet and I. D. Kuntz, *J. Mol. Biol.*, **221,** 327 (1991).

2. E. C. Meng, B. K. Shoichet, and I. D. Kuntz, *J. Comp. Chem.*, **13,** 505 (1992).

3. W. F. van Gunsteren and H. J. C. Berendsen, *Angew. Chem. Int. Ed. Engl.*, **29,** 992 (1990).

4. I. D. Kuntz, J. M. Blaney, S. J. Oatley, R. Langridge, and T. E. Ferrin, *J. Mol. Biol.*, **161,** 269 (1982).

5. B. K. Shoichet, D. L. Bodian, and I. D. Kuntz, *J. Comp. Chem.*, **13,** 380 (1992).

6. I. Klapper, R. Hagstrom, R. Fine, K. Sharp, and B. Honig, *Proteins*, **1,** 47 (1986).

7. M. K. Gilson, K. A. Sharp, and B. H. Honig, *J. Comp. Chem.*, **9,** 327 (1987).

8. S. J. Weiner, P. A. Kollman, D. A. Case, U. C. Singh, C. Ghio, G. Alagona, S. Profeta, Jr., and P. Weiner, *J. Am. Chem. Soc.*, **106,** 765 (1984).

9. S. J. Weiner, P. A. Kollman, D. T. Nguyen, and D. A. Case, *J. Comp. Chem.*, **7,** 230 (1986).

10. K. A. Dill, *Biochemistry*, **29,** 7133 (1990).

11. M. K. Gilson and B. Honig, *Proteins*, **4,** 7 (1988).

12. B. K. Shoichet, R. M. Stroud, D. V. Santi, I. D. Kuntz, and K. M. Perry, *Science*, **259,** 1445 (1993).

13. A. A. Rashin, *J. Phys. Chem.*, **94,** 1725 (1990).

14. N. K. Rogers, *Prog. Biophys. Molec. Biol.*, **48,** 37 (1986).

15. A. R. Fersht and M. J. E. Sternberg, *Prot. Eng.*, **2,** 527 (1989).

16. M. Whitlow and M. M. Teeter, *J. Am. Chem. Soc.*, **108,** 7163 (1986).

17. R. W. Pickersgill, *Prot. Eng.*, **2,** 247 (1988).

18. E. L. Mehler and G. Eichele, *Biochemistry,* **23,** 3887 (1984).

19. E. L. Mehler and T. Solmajer, *Prot. Eng.*, **4,** 903 (1991).

20. M. K. Gilson and B. Honig, *J. Comp.-Aided Mol. Design,* **5,** 5 (1991).

21. W. C. Still, A. Tempczyk, R. C. Hawley, and T. Hendrickson, *J. Am. Chem. Soc.*, **112,** 6127 (1990).

22. D. Eisenberg and A. D. McLachlan, *Nature,* **319,** 199 (1986).

23. N. Horton and M. Lewis, *Protein Science,* **1,** 169 (1992).

24. C. Hansch and A. J. Leo, *Substituent Constants for Correlation Analysis in Chemistry and Biology,* John Wiley & Sons, New York, NY, 1979.

25. R. F. Rekker and H. M. de Kort, *Eur. J. Med. Chem.*, **14,** 479 (1979).

26. G. Klopman, K. Namboodiri, and M. Schochet, *J. Comp. Chem.*, **6,** 28 (1985).

27. T. Suzuki and Y. Kudo, *J. Comp.-Aided Mol. Design,* **4,** 155 (1990).

28. A. Leo, D. Weininger, and A. Weininger, CLOGP, Medicinal Chemistry Project, Pomona College, Claremont, CA 91711.

29. F. C. Wireko, G. E. Kellogg, and D. J. Abraham, *J. Med. Chem.*, **34,** 758 (1991).

30. G. E. Kellogg, G. S. Joshi, and D. J. Abraham, *Med. Chem. Res.*, **1,** 444 (1992).

31. K. D. Stewart, T. A. Fairley, J. A. Bentley, C. W. Andrews, and M. Cory, *Med. Chem. Res.*, **1,** 439 (1992).

32. F. C. Bernstein, T. F. Koetzle, G. J. B. Williams, E. F. Meyer, Jr., M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi, *J. Mol. Biol.*, **112,** 535 (1977).

33. E. E. Abola, F. C. Bernstein, S. H. Bryant, T. F. Koetzle, and J. Weng, in *Crystallographic Databases: Information Content, Software Systems, Scientific Applications,* F. H. Allen, G. Bergerhoff, and R. Seivers, Eds., Data Commission of the International Union of Crystallography, Bonn/Cambridge/Chester, 1987, pp. 107-132.

34. H. Tsukada and D. M. Blow, *J. Mol. Biol.,* **184,** 703 (1985).

35. F. M. Richards, *Annu. Rev. Biophys. Bioeng.,* **6,** 151 (1977).

36. M. L. Connolly, *J. Appl. Crystallogr.,* **16,** 548 (1983).

37. M. L. Connolly, *Science,* **221,** 709 (1983).

38. T. E. Ferrin, C. C. Huang, L. E. Jarvis, and R. Langridge, *J. Mol. Graph.,* **6,** 13 (1988).

39. R. A. Wallace, A. N. Kurtz, and C. Niemann, *Biochemistry,* **2,** 824 (1963).

40. Molecular Design Limited, San Leandro, CA 94577.

41. B. D. Christie, D. R. Henry, O. F. Güner, and T. Moock, *Online Information 90,* 137 (1990).

42. R. S. Pearlman, *Chem. Design Automation News,* **2,** 1 (1987).

43. R. S. Pearlman, A. Rusinko, J. M. Skell, and R. Balducci, 'CONCORD User's Manual,' Tripos Associates, St. Louis, 1988.

44. Molecular Modeling System SYBYL, TRIPOS Associates, Inc., St. Louis, MO 63117.

45. J. Gasteiger and M. Marsili, *Tetrahedron,* **36,** 3219 (1980).

46. M. Marsili and J. Gasteiger, *Croat. Chem. Acta,* **53,** 601 (1980).

47. J. Gasteiger and M. Marsili, *Organ. Magn. Reson.,* **15,** 353 (1981).

APPENDIX 1: PEPTIDE SHAPE DATABASES

FROM PROTEIN STRUCTURES

Unlike organic molecules in general, peptides are inherently modular and amenable to automated synthesis. They can be constructed with relatively small investments in time and effort and are thus attractive as first-generation ligands in receptor-structure-based design. My earliest work as a member of the Kuntz group was to create databases of peptide fragments from some of the protein structures in the Brookhaven Protein Data Bank[1,2] (PDB). I used the database format appropriate for "shape" or "contact" scoring by DOCK 1.1[3] since other scoring options were not yet available. The data set consisted of 17 of the 18 structures that Ponder and Richards used to construct their side-chain rotamer library[4] (format conversion problems arose with one file), as these are well-determined and diverse. In some cases, 14 additional well-determined structures were used. Names of the PDB files contributing to each database are given below.

Several programs were used in the conversion process. First, internal coordinates were written for each structure using the Midas[5] command "midas.out." Each protein was examined with a sliding window two, three, or four peptide units wide, depending on the size of the fragments being generated. A descriptor line including the residues' identities and torsional classifications was written out for each window position (programs BINIT, BINIT3, BINIT4). The descriptor lines were sorted and redundancies were removed (programs SCREEN and ORDER). Finally, the peptide fragments corresponding to the remaining lines were excised from the protein structures (program GETPDB) and converted to shape database format (programs MKSDB, MKSDB3, MKSDB4). Some shape databases were converted to their mirror image databases using the program

INVERT. Additional peptide databases were created from molecules in the Cambridge Structural Database (CSD),[6] using the program MKDBV11 (obtained from Renée Des-Jarlais, a Kuntz group member).

The following information is in a file, DBINFO, in the same directory as the databases.

ECM November 1989

The following DOCKable databases are available -- please feel free to use them and refer any problems or questions to me.

1) **dipep.db** -- 1937 conformationally unique dipeptides, representing 368/400 of the standard dipeptides. Conformational "uniqueness" was achieved by classifying backbone and side chain torsions into "bins" and discarding a dipeptide if it duplicated another with respect to amino acid constituents and every torsion bin. The source of the dipeptides was the Ponder and Richards set of PDB files (minus 1ppt due to problems with format interconversion): 1bp2, 1crn, 1ins, 1lz1, 1mbo, 1nxb, 1ppd, 1sn3, 2alp, 2app, 2hhb, 2rhe, 2sga, 3sgb, 5cyt, 5pti, and 5rxn. The actual PDB fragments are in "dipdb" (in the same directory).

bins: phi -180 to 0, 0 to +180

psi -180 to -60, -60 to +60, +60 to +180

each chi +120 to -120, -120 to 0, 0 to +120

Naming convention: PDB file, one-letter code for the first residue, one-letter code for the second residue, number of times this particular dipeptide has been encountered + 10. Examples:

1bp2KK11 Lys-Lys from 1bp2, first occurrence of Lys-Lys in the database

3sgbGQ14 Gly-Gln from 3sgb, fourth occurrence of Gly-Gln in the database

2) **tripep.db** -- 2682 tripeptides from the same source as dipep.db. These fragments were not screened for uniqueness.

Naming convention: PDB file, one-letter code for the first residue, one-letter code for the second residue, one-letter code for the third residue, number of times this particular tripeptide has been encountered. Example:

5cytALV1 Ala-Leu-Val from 5cyt, first occurrence of Ala-Leu-Val in the database

3) **csdpep.db** -- 171 peptide-like molecules obtained by a connectivity search of the Cambridge Structural Database. Hydrogens were retained whenever they were present.

Naming convention: CSD refcode.

ECM November 1989

4) **csdaas.db** -- all structures in Cambridge Database class 48 (amino acids, including nonstandard; linear peptides; cyclic peptides) that were amenable to conversion and that are not already present in csdpep.db. In other words, the 794 (to be exact) structures include standard amino acids, nonstandard amino acids, cyclic peptides, and a few linear peptides that were for some reason not picked up by the connectivity search used in generating csdpep.db. Hydrogens were retained whenever they were present. The bibliographic references for all of class 48 are in "csdbib." There is a bound hard copy of "csdbib" in S-955.

Naming convention: CSD refcode.

5) **invdi.db, invtri.db** -- the mirror images of the structures in dipep.db and tripep.db, respectively.

Naming convention: the character*8 code for the original structure written backwards.

ECM April 1990

6) **pepaas.db** -- concatenation of csdpep.db and csdaas.db, 965 structures.

7) **invpepaas.db** -- the mirror images of the structures in pepaas.db.

Naming convention: the character*8 code for the original structure written backwards.

ECM June 1990

8) **tripep.db2** -- 2927 more tripeptides (different from those in tripep.db) taken from 14 well-determined structures (resolution no greater than 2.0 angstroms; 2ca2, 5cpa, 8dfr, 1gcr, 3grs, 1il1b, 2rnt, 1rn3, 3tln, 1ton, 3wrp, 1ubq, 1utg, and 2cpp).

Naming convention: the same as for tripep.db.

9) **tripepall.db** -- 5609 tripeptides, concatenation of tripep.db and tripep.db2.

Naming convention: the same as for tripep.db.

10) **tetpep.db** -- 5571 tetrapeptides taken from 31 well-determined structures, of which the first 17 are listed under 1) and the remaining 14 are listed under 9).

Naming convention: PDB file, sequence in one-letter codes. Example:

1crnTCCP Thr-Cys-Cys-Pro from 1crn

11) **invtriall.db, invtet.db** -- the mirror images of the structures in tripepall.db and tetpep.db, respectively.

Naming convention: the character*8 code for the original structure written backwards.

References

1. F. C. Bernstein, T. F. Koetzle, G. J. B. Williams, E. F. Meyer, Jr., M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi, *J. Mol. Biol.*, **112,** 535 (1977).

2. E. E. Abola, F. C. Bernstein, S. H. Bryant, T. F. Koetzle, and J. Weng, in *Crystallographic Databases: Information Content, Software Systems, Scientific Applications,* F. H. Allen, G. Bergerhoff, and R. Seivers, Eds., Data Commission of the International Union of Crystallography, Bonn/Cambridge/Chester, 1987, pp. 107-132.

3. R. L. DesJarlais, R. P. Sheridan, G. L. Seibel, J. S. Dixon, I. D. Kuntz, and R. Venkataraghavan, *J. Med. Chem.*, **31,** 722 (1988).

4. J. W. Ponder and F. M. Richards, *J. Mol. Biol.*, **193,** 775 (1987).

5. T. E. Ferrin, C. C. Huang, L. E. Jarvis, and R. Langridge, *J. Mol. Graph.*, **6,** 13 (1988).

6. F. H. Allen, O. Kennard, W. D. S. Motherwell, W. G. Town, and D. G. Watson, *J. Chem. Doc.*, **13,** 119 (1973).

APPENDIX 2: DOCK 3.0 DATABASES

FROM MACCS-3D DATABASES

BACKGROUND

The earliest versions of DOCK measure complementarity in terms of shape only.[1-3] The implementation of more advanced scoring functions[4] requires parallel advances in database format, as atomic charges are needed for evaluating electrostatic complementarity and atom types are required for calculating van der Waals interaction energies.

The ability to search databases with DOCK was introduced in version 1.1.[2] The original approach was to use contact scoring, a rough measure of shape complementarity, to identify structures that fit into a target site particularly well. These were to be viewed as "templates," frameworks with little or no chemical sense; it was left to the user to decide which atom types and functional groups would be most conducive to binding. Several developments led to the consideration of atom types and partial charges during molecular docking. One development was the increased availability of huge databases of organic structures. With a larger selection of molecules, more stringent screening is practicable. More compounds are winnowed out when detailed electrostatic and steric complementarity, rather than just rough shape fitting, is required. The structures that remain are then candidate lead compounds requiring little or no chemical modification. The investment of time and money is minimized by this approach, especially if the databases consist of compounds that are commercially available. Another development was the realization that grid-based scoring methods are extremely time-efficient, such that increasing the complexity of the scoring function would not necessarily cause a drastic increase in computational time. The grid-based contact scoring in DOCK 2.0[3] is much faster than the

original contact scoring method,[2] for example, and electrostatic scoring using a potential grid precalculated with DelPhi[5,6] adds little to the total DOCK run time.[4] Indeed, the calculation of grid-based molecular mechanical interaction energies is quite affordable.[4] Finally, the availability of faster workstations and greater amounts of disk space for storing databases and grid files facilitated these trends.

In 1990, the Kuntz group acquired three databases from Molecular Design Limited (MDL):[7] Comprehensive Medicinal Chemistry (CMC), MACCS-II Drug Data Review (MDDR), and the Fine Chemicals Directory (FCD). The CMC database contains structures of common therapeutic agents as listed in Pergamon Press' *Drug Compendium*, in volume six of *Comprehensive Medicinal Chemistry*;[8] the MDDR database contains structures of compounds covered in recent patent applications, as listed in issues of the Prous journal *Drug Data Report* since mid-1988; and the FCD contains structures from the catalog listings of over 65 international chemical suppliers.[9] When initially acquired, CMC contained 4817 structures, MDDR contained 12,048 structures, and the FCD contained 57,128 structures. Coordinates had been generated for these molecules using CONCORD.[10,11]

The steps taken to convert the MDL databases into DOCK 3.0 databases are described below. The process is not merely an exercise in reformatting, because the DOCK databases contain more information than the MDL databases.

METHODS

An SDfile was generated from each database by Cynthia Corwin and Brian Shoichet, using the MDL MACCS software.[7] SDfiles are essentially the same as MACCS MOLfiles[7] except they include some additional data not important for conversion to DOCK format. The relevant data for a given compound are: number of atoms, number of bonds, elemental identities, coordinates, connectivities, bond orders, registry number, and sometimes the compound name. Hydrogens were generally not included. Although CONCORD can generate hydrogen coordinates, molecules become protonated to a neutral state in the process (with the exception of quaternary amines and the like). For drug design purposes, it is preferable that functional groups be in the charge states most likely to occur at physiological pH in aqueous solution: carboxylates and phosphates negatively charged, aliphatic amines positively charged, and so on.

The task at hand was to add the appropriate hydrogens to and calculate partial charges for these thousands of molecules, all in an automated fashion. The SYBYL molecular modeling package[12] contains hydrogen addition and charge calculation capabilities. Considering the number of molecules that were to be processed and the approximate nature of the DOCK force field scoring function, rapidity was deemed more important than a high level of accuracy for the charge calculations. Instead of a quantum-mechanical (QM) method, the SYBYL Gasteiger-Marsili option[13-15] was used. This algorithm is connectivity-based (independent of conformation) and significantly faster than any QM calculation. The resulting values are qualitatively and semiquantitatively reasonable, even superior in some cases to QM charges in reproducing the experimental dipole moments of small organic compounds (Maria Longuemarie, unpublished results).

Although SYBYL can read MOLfile format, preprocessing is necessary for two main reasons. First, atom types need to be assigned such that the desired formal charge states will result. Simply reading in the MOLfiles results in protonation to neutrality whenever possible (positively charged quaternary amines but neutral carboxyl groups and aliphatic amines, for example). Second, the atom types available in SYBYL do not include all the elements. Unrecognized atoms are assigned the dummy atom type, which is ignored during all subsequent operations on the molecule, including hydrogen addition and charge calculation. This treatment may not always be appropriate, and could result in the incorporation of meaningless or spurious structures into the database.

Finally, special problems are presented by the need to process a large and heterogeneous collection of structures automatically. Files must be broken down into manageable segments, and the programs and scripts must be able to handle a wide variety of molecules without choking.

The conversion involves the following steps:

- Using the program SMTM (smart MACCS to MOL2 converter), which reads in an SDfile, recognizes functional groups that should have formal charges, assigns the appropriate SYBYL atom types, and writes out manageable (in terms of size) files in SYBYL MOL2 format.[12] Oxygens that are to receive a formal negative charge are temporarily assigned the SYBYL lone pair type, so that they will not get protonated in the following step. Nitrogens that are to receive a formal positive charge are recognized and assigned the proper types.

Formal charge decisions:

Nitrogen. If sp3-hybridized, a nitrogen will be considered positively charged if it is (1) bonded to four nonhydrogen atoms OR (2) NOT alpha to any double or triple bond

AND NOT bonded to anything other than hydrogens and sp3-hybridized carbons (exception: the terminal nitrogen in a hydrazine; bonded to only one other nonhydrogen atom which is also an sp3-hybridized nitrogen) AND NOT two bonds from any oxygen. If double-bonded, a nitrogen will be considered positively charged if it is bonded to three nonhydrogen atoms. Positive charge is added to amidinium and guanidinium groups at a later stage (in MOL2DB3, see below).

Oxygen. If single-bonded to one nonhydrogen atom only, an oxygen will be considered negatively charged if it is part of a carboxylate, sulfate, phosphate, or nitro group (also sulfonate, phosphonate, etc.). These groups are recognized by the presence of another oxygen double-bonded to the central atom. Nitro groups are present in the SDfiles as the charge-separated resonance structures. The charges on oxygens that are equivalent by resonance in these functional groups are equalized at a later stage (in MOL2DB3, see below).

Structures with atoms not recognized by SYBYL are funneled to a separate file, as are structures containing sp-hybridized nitrogens. These nitrogens were not handled well by my original set of scripts in conjunction with SYBYL version 5.3, so I had to treat them separately with a different macro. Molecules with atoms not recognized by SYBYL were not processed further.

- Running a SYBYL script, which calls a macro in SYBYL programming language (SPL), for each file generated in the preceding step. Each molecule in the file is read in, hydrogens are added, lone pairs are changed back to sp3-hybridized oxygens, and charges are calculated. The atoms are then assigned unique names. After all the molecules in a file have been processed, they are written out in SYBYL MOL2 format, including the new information. A different script and macro are necessary for processing

structures with sp-hybridized nitrogens, at least in SYBYL 5.3.

• Using the program MOL2DB3 to convert from SYBYL MOL2 format to DOCK 3.0 database format. MOL2DB3 is similar to the program MOL2DB, a reformatting program distributed with DOCK 3.0, but performs some additional operations. These are adjustments of the partial charges in certain functional groups: equalization of charges on oxygens that are equivalent by resonance (in carboxylates, phosphates, sulfates, etc.) and addition of a net positive charge to amidinium and guanidinium groups (the charge is spread over the two or three nitrogens, respectively). Only the largest bonded group per reference code is written out; counterions are discarded.

Each compound's registry number is carried through the all the steps, as well as its name, if available. The MACCS database name and registry number become the reference code of the structure in the DOCK 3.0 database ("FCD 55652" would be the reference code of the compound in the FCD with the registry number 55652). The compound name is also included in this format, on a separate line. Unfortunately and inadvertently, the *internal* registry number was used, rather than the *external* registry number. Only the *external* registry number remains constant between database releases.

The relevant code, scripts, and sample input and output are given below.

---

program SMTM

```
c                    PROGRAM SMARTMTM              E. Meng  9/4/91
c
c       This program converts MACCS molfiles to SYBYL MOL2 files,
c     making decisions about charge:  1) carboxylates are found,
c     and one oxygen in each such group is marked to be negatively
c     charged--this is done by temporarily giving the oxygen the
c     SYBYL LP type, so that a hydrogen is not added when 'addh' is
c     invoked; subsequently to hydrogen addition, the LP is changed
c     back to an oxygen.  2) nitrogens are marked to be positively
c     charged (are given the N.4 type) if they are either a) bonded
```

```
c     to sp3 carbons (C.3 atoms) only, or b) bonded to only one atom
c     and this atom is an sp3 nitrogen (N.3); this makes monosub-
c     stituted hydrazines positively charged (the pKa of hydrazine
c     is approximately 8).
c     9/5/90 error fixed so that S.O and S.O2 can now be identified.
c     11/15/90 error fixed so that phosphates and sulfates will not
c     get protonated (oxygens temporarily given the LP type, as
c     mentioned above for carboxylates)
c     9/4/91 error fixed so that all double-bonded nitrogens are
c     are made into N.2, NOT N.pl3, even if 3-coordinate.  Gives
c     N-substituted pyridinium molecules and nitro groups the correct
c     net charge.
c
c
c
      integer maxats, maxbds, maxlks
      parameter (maxats=255)
      parameter (maxbds=255)
      parameter (maxlks=8)
      character*80 line
      character*79 header, molnam, bdlin(maxbds)
      character*30 coords(maxats)
      character*3 name(maxats)
      character*5 type(maxats)
      character*4 mol2a
      integer i, j, k, l, m, a1, a2, bo
      integer ccode(maxats), links(maxats,maxlks), nlinks(maxats)
      integer maxbo(maxats)
      integer ocnt, molcnt, totcnt, molend, unitno
      character*1 numlet
c
      open (unit=1, file='maccs', status='old')
      mol2a='cmca'
      open (unit=2, file=mol2a, status='new')
      open (unit=3, file='spns', status='new')
      open (unit=4, file='other', status='new')
c
c  --initialize everything
c
      totcnt=1
    5 molcnt=0
   10 do 20 i=1, maxats
        coords(i)(1:10)='          '
        name(i)='   '
        type(i)='     '
        ccode(i)=0
        maxbo(i)=0
        nlinks(i)=0
        do 15 j=1,maxlks
          links(i,j)=0
   15   continue
   20 continue
      do 30 i=1, maxbds
        bdlin(i)(1:10)='          '
   30 continue
      unitno=2
c
c  --read in next molecule
c
      molcnt=molcnt + 1
      read (1, '(A79)', end=900) molnam
      read (1, '(A79)', end=900) header
      read (1, '(A80)', end=900) line
      read (1, '(2I3)') nats, nbds
```

```
      do 130 i=1, nats
        read (1, 1000) coords(i), name(i), ccode(i)
1000    format (A30, A3, 3x, I3)
 130  continue
      do 160 i=1, nbds
        read (1, '(A79)') bdlin(i)
        read (bdlin(i), '(3I3)') a1, a2, bo
        nlinks(a1)=nlinks(a1) + 1
        nlinks(a2)=nlinks(a2) + 1
        if (nlinks(a1) .gt. maxlks .or. nlinks(a2) .gt. maxlks)
     &  then
          write (6, *) 'Too many links at bond ',i
          write (6, *) 'Header of skipped structure:'
          write (6, *) header
          write (6, *) ' '
          go to 10
        endif
        links(a1,nlinks(a1))=a2
        links(a2,nlinks(a2))=a1
        if (bo .gt. maxbo(a1) .and. bo .lt. maxlks) maxbo(a1)=bo
        if (bo .gt. maxbo(a2) .and. bo .lt. maxlks) maxbo(a2)=bo
 160  continue
 180  read (1, '(A80)', end=900) line
      if (molnam(1:3) .eq. '   ') then
        if (line(7:13) .eq. 'GENERIC' .or. line(8:14) .eq.
     &  'GENERIC' .or. line(9:15) .eq. 'GENERIC' .or. line(10:16)
     &  .eq. 'GENERIC' .or. line(11:17) .eq. 'GENERIC' .or.
     &  line(12:18) .eq. 'GENERIC') then
          read (1, '(A79)') molnam
        endif
      endif
      if (line(1:4) .eq. '$$$$') then
        do 200 i=1, nats
          if (name(i) .eq. ' C ') then
            if (nlinks(i) .eq. 0) then
              type(i) ='C.3  '
            else if (maxbo(i) .eq. 1) then
              type(i)='C.3  '
            else if (maxbo(i) .eq. 2) then
              type(i)='C.2  '
            else if (maxbo(i) .eq. 3) then
              type(i)='C.1  '
            endif
          else if (name(i) .eq. ' N ') then
            if (nlinks(i) .eq. 0) then
              type(i)='N.4  '
            else if (maxbo(i) .eq. 1) then
              type(i)='N.3  '
              if (ccode(i) .eq. 3) type(i)='N.4  '
            else if (maxbo(i) .eq. 2) then
              type(i)='N.2  '
            else if (maxbo(i) .eq. 3) then
              type(i)='N.1  '
              if (unitno .eq. 2) unitno=3
            endif
          else if (name(i) .eq. ' O ') then
            if (nlinks(i) .eq. 0) then
              type(i)='O.3  '
            else if (maxbo(i) .eq. 1) then
              type(i)='O.3  '
              if (ccode(i) .eq. 5) type(i)='LP   '
            else if (maxbo(i) .eq. 2) then
              type(i)='O.2  '
              if (nlinks(i) .gt. 1) unitno=4
```

```
            else if (maxbo(i) .eq. 3) then
              unitno=4
            endif
          else if (name(i) .eq. ' S ') then
            if (nlinks(i) .eq. 0) then
              type(i)='S.3  '
            else if (maxbo(i) .eq. 1) then
              type(i)='S.3  '
            else if (maxbo(i) .eq. 2) then
              type(i)='S.2  '
            endif
            ocnt=0
            do 190 j=1, nlinks(i)
              k=links(i,j)
              if (name(k)(2:2) .eq. 'O') ocnt=ocnt + 1
  190       continue
            if (ocnt .ge. 2) then
              type(i)='S.O2 '
ccc--SYBYL version 5.4, at least, considers S.O2=S.o2 and S.O=S.o
ccc           type(i)='S.o2 '
            else if (ocnt .eq. 1) then
              type(i)='S.O  '
ccc           type(i)='S.o  '
            endif
          else if (name(i) .eq. ' P ') then
            type(i)='P.3  '
          else if (name(i) .eq. ' Br') then
            do 195 j=1, nlinks(i)
              k=links(i,j)
              if (name(k)(2:2) .eq. 'O') unitno=4
  195       continue
          else if (name(i) .eq. ' Se' .or. name(i) .eq. ' Pt'
     &     .or. name(i) .eq. ' Fe' .or. name(i) .eq. ' Hg' .or.
     &     name(i) .eq. ' Au' .or. name(i) .eq. ' Pb' .or. name(i)
     &     .eq. ' Cu' .or. name(i) .eq. ' Zn' .or. name(i) .eq.
     &     ' Mg' .or. name(i) .eq. ' Mn' .or. name(i) .eq. ' Co'
     &     .or. name(i) .eq. ' As' .or. name (i) .eq. ' Sb' .or.
     &     name(i) .eq. ' Ba' .or. name(i) .eq. ' Be' .or. name(i)
     &     .eq. ' Cs' .or. name(i) .eq. ' Bi' .or. name(i) .eq.
     &     ' Mo' .or. name(i) .eq. ' Ni' .or. name(i) .eq. ' Os'
     &     .or. name(i) .eq. ' Pd' .or. name(i) .eq. ' Rb' .or.
     &     name(i) .eq. ' Ag' .or. name(i) .eq. ' Sn' .or. name(i)
     &     .eq. ' Ti' .or. name(i) .eq. ' U ' .or. name(i) .eq.
     &     ' V ' .or. name(i) .eq. ' W ' .or. name(i) .eq. ' Yb'
     &     .or. name(i) .eq. ' Y ' .or. name(i) .eq. ' Zr' .or.
     &     name(i) .eq. ' B ' .or. name(i) .eq. ' Cd' .or. name(i)
     &     .eq. ' Xe' .or. name(i) .eq. ' Hf' .or. name(i) .eq.
     &     ' Ru' .or. name(i) .eq. ' Cr' .or. name(i) .eq. ' Ar'
     &     .or. name(i) .eq. ' Tb' .or. name(i) .eq. ' Th') then
            unitno=4
          else if (name(i)
     &     .eq. ' Tl' .or. name(i) .eq. ' Rh' .or. name(i)
     &     .eq. ' Nb' .or. name(i) .eq. ' Ce' .or. name(i) .eq.
     &     ' Dy' .or. name(i) .eq. ' Er' .or. name(i) .eq. ' Eu'
     &     .or. name(i) .eq. ' Ga' .or. name(i) .eq. ' Gd' .or.
     &     name(i) .eq. ' Ge' .or. name(i) .eq. ' He' .or. name(i)
     &     .eq. ' Ho' .or. name(i) .eq. ' In' .or. name(i) .eq.
     &     ' Ir' .or. name(i) .eq. ' La' .or. name(i) .eq. ' Lu'
     &     .or. name(i) .eq. ' Nd' .or. name(i) .eq. ' Te' .or.
     &     name(i) .eq. ' Tm' .or. name(i) .eq. ' Pr' .or. name(i)
     &     .eq. ' Ne' .or. name(i) .eq. ' Re' .or. name(i) .eq.
     &     ' Sc' .or. name(i) .eq. ' Sm' .or. name(i) .eq. ' Sr'
     &     .or. name(i) .eq. ' Ta' .or. name(i) .eq. ' Kr' .or.
     &     name(i) .eq. ' Tc')
```

```
     &     then
             unitno=4
           endif
           if (type(i) .eq. '     ') type(i)=name(i)(2:3)//'   '
  200    continue
         do 300 i=1, nats
           if (type(i) .eq. 'N.3  ') then
             charge=1
             do 290 j=1, nlinks(i)
               k=links(i,j)
               if (maxbo(k) .ge. 2) type(i)='N.pl3'
               if ((type(k) .ne. 'C.3  ') .and. .not. (type(k) .eq.
     &         'N.3  ' .and. nlinks(i) .eq. 1)) charge=0
               do 280 l=1, nlinks(k)
                 m=links(k,l)
                  if (name(m)(2:2) .eq. 'O') type(i)='N.pl3'
  280          continue
  290        continue
             if (charge .eq. 1 .and. type(i) .eq. 'N.3  ') then
               type(i)='N.4  '
             endif
           endif
  300    continue
         do 395 i=1, nats
           if (type(i) .eq. 'O.3  ' .and. nlinks(i) .eq. 1) then
             k=links(i,1)
             if (type(k) .eq. 'P.3  ' .or. type(k) .eq. 'S.O2 '
     &       .or. type(k) .eq. 'S.o2 '.or. type(k) .eq. 'N.2  '
     &       .or. type(k) .eq. 'C.2  ') then
               do 380 l=1, nlinks(k)
                 m=links(k,l)
                  if (type(m) .eq. 'O.2  ') type(i)='LP   '
  380          continue
             endif
           endif
  395    continue
         do 396 i=2, 78
           if (molnam(i:i+1) .eq. '  ' .or. molnam(i:i+1)
     &     .eq. ' [') then
             molend=i-1
             go to 397
           endif
  396    continue
  397    continue
         do 398 i=1, molend
           if (molnam(i:i) .eq. ' ') molnam(i:i)='_'
  398    continue
         do 399 i=molend+1, 79
           molnam(i:i)=' '
  399    continue
c
c  --write molecule to output
c
         write (unitno, '(A17)') '@<TRIPOS>MOLECULE'
         write (unitno, '(A50, A4, A6)') molnam(1:50), ' CMC',
     &  header(47:52)
         write (unitno, '(2I6)') nats, nbds
         write (unitno, '(A6)') 'SMALL'
         write (unitno, '(A10)') 'NO_CHARGES'
         write (unitno, '(A13)') '@<TRIPOS>ATOM'
         do 400 i=1, nats
           write (unitno, 1001) i, name(i), coords(i), type(i)
 1001      format (I7, 1x, A3, 3x, A30, 1x, A5)
  400    continue
```

```
        write (unitno, '(A13)') '@<TRIPOS>BOND'
        do 500 i=1, nbds
          write (unitno, 1002) i, bdlin(i)(1:3), bdlin(i)(4:9)
 1002     format (I5, 3x, A3, 1x, A6)
  500   continue
        if (molcnt .ge. 2000) then
          totcnt=totcnt + 1
          close (2)
          mol2a='cmc'//numlet(totcnt)
          open (unit=2, file=mol2a, status='new')
          go to 5
        endif
        go to 10
      endif
      go to 180
c
  900 continue
      close (1)
      close (2)
      close (3)
      end
c
c
c
      character*1 function numlet(num)
      integer num
c
      if (num .eq. 1) numlet='a'
      if (num .eq. 2) numlet='b'
      if (num .eq. 3) numlet='c'
      if (num .eq. 4) numlet='d'
      if (num .eq. 5) numlet='e'
      if (num .eq. 6) numlet='f'
      if (num .eq. 7) numlet='g'
      if (num .eq. 8) numlet='h'
      if (num .eq. 9) numlet='i'
      if (num .eq. 10) numlet='j'
      if (num .eq. 11) numlet='k'
      if (num .eq. 12) numlet='l'
      if (num .eq. 13) numlet='m'
      if (num .eq. 14) numlet='n'
      if (num .eq. 15) numlet='o'
      if (num .eq. 16) numlet='p'
      if (num .eq. 17) numlet='q'
      if (num .eq. 18) numlet='r'
      if (num .eq. 19) numlet='s'
      if (num .eq. 20) numlet='t'
      if (num .eq. 21) numlet='u'
      if (num .eq. 22) numlet='v'
      if (num .eq. 23) numlet='w'
      if (num .eq. 24) numlet='x'
      if (num .eq. 25) numlet='y'
      if (num .eq. 26) numlet='z'
      return
      end
```

MACCS SDfile "maccs," sample input to SMTM

```
GTMACCS-II08219019553D 1    1.00000     0.00000     4     GST

 17 18  0  0  0  0              0
    0.0537   -0.5662    0.1797 C   0  0  1  0  0
   -1.0791    0.2515   -0.3933 C   0  0  0  0  0
    1.1631   -0.6659   -0.8398 C   0  0  0  0  0
    0.4656    0.0095    1.5135 C   0  0  0  0  0
   -0.5647   -1.9475    0.3903 C   0  0  0  0  0
   -0.8758    1.5981   -0.7181 C   0  0  0  0  0
   -2.3325   -0.3368   -0.5997 C   0  0  0  0  0
    2.3757   -0.2495   -0.4861 N   0  0  0  0  0
    0.9517   -1.1186   -1.9637 O   0  0  0  0  0
    1.7347    0.3717    1.6783 N   0  0  0  0  0
   -0.3512    0.1400    2.4238 O   0  0  0  0  0
    0.4835   -2.8905    0.9738 C   0  0  0  0  0
   -1.9261    2.3563   -1.2493 C   0  0  0  0  0
   -3.3828    0.4213   -1.1309 C   0  0  0  0  0
    2.6463    0.2460    0.7181 C   0  0  0  0  0
   -3.1795    1.7679   -1.4557 C   0  0  0  0  0
    3.8168    0.6122    0.9597 O   0  0  0  0  0
  1  2  1  0  0  0
  1  3  1  0  0  0
  1  4  1  0  0  0
  1  5  1  0  0  0
  2  6  2  0  0  0
  2  7  1  0  0  0
  3  8  1  0  0  0
  3  9  2  0  0  0
  4 10  1  0  0  0
  4 11  2  0  0  0
  5 12  1  0  0  0
  6 13  1  0  0  0
  7 14  2  0  0  0
  8 15  1  0  0  0
 13 16  2  0  0  0
 15 17  2  0  0  0
 10 15  1  0  0  0
 14 16  1  0  0  0
> 4  <GENERIC.NAME>
PHENOBARBITAL [U;INN]

$$$$
```

SYBYL MOL2 file "cmca," sample output from SMTM and input to SYBYL

```
@<TRIPOS>MOLECULE
PHENOBARBITAL                                                    CMC      4
    17     18
 SMALL
NO_CHARGES
@<TRIPOS>ATOM
      1  C          0.0537   -0.5662    0.1797 C.3
      2  C         -1.0791    0.2515   -0.3933 C.2
      3  C          1.1631   -0.6659   -0.8398 C.2
      4  C          0.4656    0.0095    1.5135 C.2
      5  C         -0.5647   -1.9475    0.3903 C.3
      6  C         -0.8758    1.5981   -0.7181 C.2
      7  C         -2.3325   -0.3368   -0.5997 C.2
      8  N          2.3757   -0.2495   -0.4861 N.p13
      9  O          0.9517   -1.1186   -1.9637 O.2
     10  N          1.7347    0.3717    1.6783 N.p13
     11  O         -0.3512    0.1400    2.4238 O.2
     12  C          0.4835   -2.8905    0.9738 C.3
     13  C         -1.9261    2.3563   -1.2493 C.2
     14  C         -3.3828    0.4213   -1.1309 C.2
     15  C          2.6463    0.2460    0.7181 C.2
     16  C         -3.1795    1.7679   -1.4557 C.2
     17  O          3.8168    0.6122    0.9597 O.2
@<TRIPOS>BOND
     1     1    2   1
     2     1    3   1
     3     1    4   1
     4     1    5   1
     5     2    6   2
     6     2    7   1
     7     3    8   1
     8     3    9   2
     9     4   10   1
    10     4   11   2
    11     5   12   1
    12     6   13   1
    13     7   14   2
    14     8   15   1
    15    13   16   2
    16    15   17   2
    17    10   15   1
    18    14   16   1
```

SYBYL shell script "sybylout.com"

```
for chunk in cmc*
do
  ln -s $chunk mol2file
  time sybyl<sybin>>text
  rm mol2file
  mv mol2out.mol2 $chunk.mol2
done
```

"sybin," commands directed into SYBYL by "sybylout.com"

```
uims load doitout.macro
doitout
quit
```

"doitout.macro," SPL macro loaded and used according to "sybin"

```
@MACRO
doitout sybylbasic y
mol mult_in ml mol2file
for area in %mols(m*)
  default $area
  fillvalence * H
  for lopr in %atoms("<LP>")
    modify atom only_type $lopr O.3
  endfor
  for dum in %atoms("<Du>")
    remove atom $dum
  endfor
  modify atom name * sequential_auto
  charge $area compute gasteiger |
endfor
mol mult_out m* mol2out
.
```

"cmca.mol2," sample output from "sybylout.com" and input to MOL2DB3

```
@<TRIPOS>MOLECULE
PHENOBARBITAL                                                    CMC      4
   29     30      1      0      0
SMALL
GASTEIGER


@<TRIPOS>ATOM
      1 C1          0.0537   -0.5662    0.1797 C.3       1 <1>         0.1479
      2 C2         -1.0791    0.2515   -0.3933 C.2       1 <1>        -0.0237
      3 C3          1.1631   -0.6659   -0.8398 C.2       1 <1>         0.2398
      4 C4          0.4656    0.0095    1.5135 C.2       1 <1>         0.2398
      5 C5         -0.5647   -1.9475    0.3903 C.3       1 <1>        -0.0293
      6 C6         -0.8758    1.5981   -0.7181 C.2       1 <1>        -0.0571
      7 C7         -2.3325   -0.3368   -0.5997 C.2       1 <1>        -0.0571
      8 N8          2.3757   -0.2495   -0.4861 N.p13     1 <1>        -0.2387
      9 O9          0.9517   -1.1186   -1.9637 O.2       1 <1>        -0.2739
     10 N10         1.7347    0.3717    1.6783 N.p13     1 <1>        -0.2387
     11 O11        -0.3512    0.1400    2.4238 O.2       1 <1>        -0.2739
     12 C12         0.4835   -2.8905    0.9738 C.3       1 <1>        -0.0636
     13 C13        -1.9261    2.3563   -1.2493 C.2       1 <1>        -0.0615
     14 C14        -3.3828    0.4213   -1.1309 C.2       1 <1>        -0.0615
     15 C15         2.6463    0.2460    0.7181 C.2       1 <1>         0.3122
     16 C16        -3.1795    1.7679   -1.4557 C.2       1 <1>        -0.0617
     17 O17         3.8168    0.6122    0.9597 O.2       1 <1>        -0.2520
     18 H18        -0.9169   -2.3425   -0.5740 H         1 <1>         0.0280
     19 H19        -1.4132   -1.8672    1.0856 H         1 <1>         0.0280
     20 H20         0.0992    2.0558   -0.5576 H         1 <1>         0.0621
     21 H21        -2.4906   -1.3842   -0.3470 H         1 <1>         0.0621
     22 H22         3.1387   -0.3136   -1.1750 H         1 <1>         0.1591
     23 H23         2.0229    0.7651    2.5855 H         1 <1>         0.1591
     24 H24         0.0383   -3.8849    1.1254 H         1 <1>         0.0230
     25 H25         0.8357   -2.4956    1.9382 H         1 <1>         0.0230
     26 H26         1.3321   -2.9709    0.2785 H         1 <1>         0.0230
     27 H27        -1.7680    3.4037   -1.5019 H         1 <1>         0.0618
     28 H28        -4.3578   -0.0364   -1.2914 H         1 <1>         0.0618
     29 H29        -3.9964    2.3576   -1.8689 H         1 <1>         0.0618
@<TRIPOS>BOND
     1    1    2 1
     2    1    3 1
     3    1    4 1
     4    1    5 1
     5    2    6 2
     6    2    7 1
     7    3    8 1
     8    3    9 2
     9    4   10 1
    10    4   11 2
    11    5   12 1
    12    6   13 1
    13    7   14 2
    14    8   15 1
    15   13   16 2
    16   15   17 2
    17   10   15 1
    18   14   16 1
    19    5   18 1
    20    5   19 1
    21    6   20 1
    22    7   21 1
    23    8   22 1
    24   10   23 1
    25   12   24 1
    26   12   25 1
    27   12   26 1
    28   13   27 1
```

"cmca.mol2," sample output from "sybylout.com" and input to MOL2DB3

```
@<TRIPOS>MOLECULE
PHENOBARBITAL                                          CMC      4
   29     30      1      0      0
SMALL
GASTEIGER


@<TRIPOS>ATOM
      1 C1          0.0537   -0.5662    0.1797 C.3       1 <1>         0.1479
      2 C2         -1.0791    0.2515   -0.3933 C.2       1 <1>        -0.0237
      3 C3          1.1631   -0.6659   -0.8398 C.2       1 <1>         0.2398
      4 C4          0.4656    0.0095    1.5135 C.2       1 <1>         0.2398
      5 C5         -0.5647   -1.9475    0.3903 C.3       1 <1>        -0.0293
      6 C6         -0.8758    1.5981   -0.7181 C.2       1 <1>        -0.0571
      7 C7         -2.3325   -0.3368   -0.5997 C.2       1 <1>        -0.0571
      8 N8          2.3757   -0.2495   -0.4861 N.p13     1 <1>        -0.2387
      9 O9          0.9517   -1.1186   -1.9637 O.2       1 <1>        -0.2739
     10 N10         1.7347    0.3717    1.6783 N.p13     1 <1>        -0.2387
     11 O11        -0.3512    0.1400    2.4238 O.2       1 <1>        -0.2739
     12 C12         0.4835   -2.8905    0.9738 C.3       1 <1>        -0.0636
     13 C13        -1.9261    2.3563   -1.2493 C.2       1 <1>        -0.0615
     14 C14        -3.3828    0.4213   -1.1309 C.2       1 <1>        -0.0615
     15 C15         2.6463    0.2460    0.7181 C.2       1 <1>         0.3122
     16 C16        -3.1795    1.7679   -1.4557 C.2       1 <1>        -0.0617
     17 O17         3.8168    0.6122    0.9597 O.2       1 <1>        -0.2520
     18 H18        -0.9169   -2.3425   -0.5740 H         1 <1>         0.0280
     19 H19        -1.4132   -1.8672    1.0856 H         1 <1>         0.0280
     20 H20         0.0992    2.0558   -0.5576 H         1 <1>         0.0621
     21 H21        -2.4906   -1.3842   -0.3470 H         1 <1>         0.0621
     22 H22         3.1387   -0.3136   -1.1750 H         1 <1>         0.1591
     23 H23         2.0229    0.7651    2.5855 H         1 <1>         0.1591
     24 H24         0.0383   -3.8849    1.1254 H         1 <1>         0.0230
     25 H25         0.8357   -2.4956    1.9382 H         1 <1>         0.0230
     26 H26         1.3321   -2.9709    0.2785 H         1 <1>         0.0230
     27 H27        -1.7680    3.4037   -1.5019 H         1 <1>         0.0618
     28 H28        -4.3578   -0.0364   -1.2914 H         1 <1>         0.0618
     29 H29        -3.9964    2.3576   -1.8689 H         1 <1>         0.0618
@<TRIPOS>BOND
     1    1    2 1
     2    1    3 1
     3    1    4 1
     4    1    5 1
     5    2    6 2
     6    2    7 1
     7    3    8 1
     8    3    9 2
     9    4   10 1
    10    4   11 2
    11    5   12 1
    12    6   13 1
    13    7   14 2
    14    8   15 1
    15   13   16 2
    16   15   17 2
    17   10   15 1
    18   14   16 1
    19    5   18 1
    20    5   19 1
    21    6   20 1
    22    7   21 1
    23    8   22 1
    24   10   23 1
    25   12   24 1
    26   12   25 1
    27   12   26 1
    28   13   27 1
```

```
    29    14    28 1
    30    16    29 1
@<TRIPOS>SUBSTRUCTURE
     1 ****          1 TEMP              0 ****  ****     0 ROOT
```

"cmca.db3," sample output from MOL2DB3

```
N PHENOBARBITAL
CMC      4 17 12 29  0
 5 1 1 1 5 1 1 811 811 5 1 1 1 111 7 7 7 7 6 6 7 7 7 7 7 7
  148   -24   240   240   -29   -57   -57 -239 -274 -239 -274   -64   -62   -62   312   -62
 -252    28    28    62    62   159   159    23    23    23    62    62    62
 4412 3319 2144 3279 4137 1571 5521 3219 1124 4824 3895 3478 3793 1937 2354 3482
 5483 1246 2025 3548 1364 6734 3635 1478 5310 2766    0 6093 4257 3642 4007 4025
 4388 4842  994 2938 2432 6241  715  975 4306  833 7004 4131 2682 1178 5653  508
 8175 4497 2924 3441 1542 1390 2945 2018 3050 4457 5941 1406 1867 2501 1617 7497
 3571  789 6381 4650 4550 4396    0 3089 5194 1389 3902 5690  914 2243 2590 7289
  462    0 3849  673  362 6243   95
```

The conversion of sp-nitrogen-containing compounds is completely analogous, except that a different macro is used:

"spnout.macro," SPL macro for compounds containing sp-hybridized nitrogens

```
@MACRO
spnout sybylbasic y
mol mult_in ml mol2file
for area in %mols(m*)
  default $area
  for atm in %atoms(*)
    if %not(%streql("%atom_info("$atm" type)" "N.1"))
      fillvalence $atm H
    endif
  endfor
  for lopr in %atoms("<LP>")
    modify atom only_type $lopr O.3
  endfor
  for dum in %atoms("<Du>")
    remove atom $dum
  endfor
  modify atom name * sequential_auto
  charge $area compute gasteiger |
endfor
mol mult_out m* mol2out
.
```

References

1. I. D. Kuntz, J. M. Blaney, S. J. Oatley, R. Langridge, and T. E. Ferrin, *J. Mol. Biol.*, **161,** 269 (1982).

2. R. L. DesJarlais, R. P. Sheridan, G. L. Seibel, J. S. Dixon, I. D. Kuntz, and R. Venkataraghavan, *J. Med. Chem.*, **31,** 722 (1988).

3. B. K. Shoichet, D. L. Bodian, and I. D. Kuntz, *J. Comp. Chem.*, **13,** 380 (1992).

4. E. C. Meng, B. K. Shoichet, and I. D. Kuntz, *J. Comp. Chem.*, **13,** 505 (1992).

5. I. Klapper, R. Hagstrom, R. Fine, K. Sharp, and B. Honig, *Proteins,* **1,** 47 (1986).

6. M. K. Gilson, K. A. Sharp, and B. H. Honig, *J. Comp. Chem.*, **9,** 327 (1987).

7. Molecular Design Limited, San Leandro, CA 94577.

8. C. Hansch, P. G. Sammes, and J. B. Taylor, Eds., *Comprehensive Medicinal Chemistry,* Oxford, Pergamon, 1990.

9. B. D. Christie, D. R. Henry, O. F. Güner, and T. Moock, *Online Information 90,* 137 (1990).

10. R. S. Pearlman, *Chem. Design Automation News,* **2,** 1 (1987).

11. R. S. Pearlman, A. Rusinko, J. M. Skell, and R. Balducci, 'CONCORD User's Manual,' Tripos Associates, St. Louis, 1988.

12. SYBYL, Tripos Associates, St. Louis, MO 63117.

13. J. Gasteiger and M. Marsili, *Tetrahedron,* **36,** 3219 (1980).

14. M. Marsili and J. Gasteiger, *Croat. Chem. Acta,* **53,** 601 (1980).

15. J. Gasteiger and M. Marsili, *Organ. Magn. Reson.*, **15,** 353 (1981).

APPENDIX 3: DOCK 3.0 DATABASES

FROM CAMBRIDGE STRUCTURAL DATABASE FILES

BACKGROUND

There are several reasons for using approximate interaction energies in addition to shape complementarity for evaluating docked complexes, as discussed in Chapter 2[1] and Appendix 2. To allow calculation of these energies, atom types and partial charges need to be stored for the structures in the DOCK database. Appendix 2 outlines how DOCK 3.0 databases can be created from CONCORD-generated[2,3] structures in MACCS[4] SDfile format. Here the analogous process for experimental coordinates from the Cambridge Structural Database[5] (CSD) is described.

METHODS

The CSD FDAT files provide coordinates and elemental identities, but unlike SDfiles, they do not contain hybridization information. It is therefore more difficult to assign atom types, which are required for hydrogen addition and charge calculation. Some structures in the CSD include hydrogens, but often not all of them. In addition, the hydrogens may be poorly placed. I adapted the IDATM algorithm[6] (Chapter 1 and Appendix 4), which infers connectivity and hybridization states from coordinates, to write out SYBYL[7] MOL2 format. The adapted program is called IDTOSYB. The subsequent steps, processing with SYBYL and MOL2DB3, are as described in Appendix 2.

The program CSDTOPDB is needed to convert CSD FDAT format into Protein Data Bank[8] (PDB) format for input to IDTOSYB. CSDTOPDB is based on XTLCHEM

by Arthur Lewis. Renée DesJarlais altered the program to write out only the largest bonded group per CSD refcode; I corrected the handling of two-letter element symbols and fixed the section which writes information to the screen.

IDTOSYB differs from IDATM in more than just output format; additional atom types are recognized, and protons are removed (whenever present) from groups that are negatively charged at physiological pH. Isolated atoms are also detected and deleted. These tend to be hydrogens in poorly determined structures. A message is sent to the screen whenever an atom is removed from a structure. Although the output format includes connectivity and bond orders, IDTOSYB only discerns connectivity and hybridization states. It can be difficult to derive bond orders from this information, particularly in highly conjugated systems. Fortunately, the Gasteiger-Marsili charge calculation[9-11] employed via SYBYL depends on connectivity and hybridization states but not bond orders. All bonds are written out in MOL2 format as if they were single, resulting in structures that are technically incorrect but that give, ultimately, the same DOCK database entries as those with the correct bond orders.

Sample input and output files are given below.

CSD JNL file (record of commands and search results)

```
T1 *REFCode pimozd
SAVE 0 FDAT
SAVE 0 FBIB FDAT
QUES  T1
--------+---------+---------+---------+---------+---------+---------+---------+
*REFC=PIMOZD // Pimozide perchlorate // 1-(4,4-di(4-fluorophenyl)butyl)-4-(2-oxo
-1-benzamidazolinyl)piperidine // *FORM=C28 H30 F2 N3 O1 1+,Cl1 O4 1- // *AUTH=J
= 81 // *YEAR=1979 //
```

CSD FDAT file, sample input to CSDTOPDB

```
#PIMOZD  23790525        30  9  0   0  0  4  6 69   0   0 781321000000200000000000079
 17462 10090  3166     90  9632     90332020 6 2 1 0 3 0134134 15C2/c        840
R=0.1090
211 0121 0112 0211 6121 6112 0011 0121 0110 6011 6121 6110 6
C  68H  23O  68N  68CL 99F  64
C1      48860 -60780  38650 C2      46290 -48360  39750 C3      50420 -41930  43100
C4      56790 -47970  45370 C5      59360 -60240  44270 C6      55370 -66690  40890
N7      59460 -39010  48530 C8      55120 -27810  48340 O9      55770 -18140  50670
N10     49540 -29600  44940 C11     43690 -19710  43450 C12     47030  -7250  41800
C13     40730   1850  40000 N14     35400   5110  43220 C15     32130  -7020  44940
C16     38490 -16160  46810 C17     29260  15030  41640 C18     23370  10080  38110
C19     18250  21670  36590 C20     11480  18150  33370 C21     13690  11000  29450
C22     18590  16110  26900 C23     20380   9380  23260 C24     16850  -2270  22300
F25     18150  -9090  18730 C26     11920  -7670  24800 C27     10430   -850  28380
C28      6470  30330  32070 C29     -1130  28380  30700 C30     -6100  38970  29650
C31     -3170  51330  30010 F32     -8130  61680  29040 C33      4290  53910  31290
C34      9220  43110  32330 CL1     26760   4340   6670 O1      34110    150   7310
O2      26630  15650   4160 O3      22220  -3900   4640 O4      24190   5600  10530
H1      45600 -65600  36100 H2      42300 -43800  38400 H5      63700 -63500  45700
H6      56500 -76200  40100 H7      63300 -39200  50900 H11     40300 -24000  41100
H121    50500  -9800  39400 H122    50200  -2800  44200 H131    37300  -1900  37900
H132    42500  10900  39000 H14     39000  10300  45100 H151    29500  -2700  46900
H152    29200 -11400  42800 H161    41600 -12300  49200 H162    36100 -24200  47500
H171    26600  16500  44100 H172    31700  21900  40600 H181    25900   6900  35900
H182    20100   4600  39400 H191    16600  26400  39200 H192    21600  26600  35400
H20      8300  12600  35100 H22     20800  24000  27600 H23     23600  13200  21400
H26      9100 -14700  24100 H27      6100  -3500  30400 H29     -3400  20200  30700
H30    -11200  37300  28600 H33      6600  62700  31600 H34     14500  45400  33200
 2 3 4 5 6 1 4 7 8 3101112131411141718192021222324242120282930313128360353535 1
 2 5 6 7111212131314151516161717181819192022232627293033348101516262733340
```

PDB file "pdbfil," sample output from CSDTOPDB and input to IDTOSYB

```
REMARK  STRUCTURE        1   PIMOZD
ATOM        1   C1   PIM      1        7.185   -6.133   12.162
ATOM        2   C2   PIM      1        6.698   -4.880   12.508
ATOM        3   C3   PIM      1        7.302   -4.231   13.563
ATOM        4   C4   PIM      1        8.335   -4.840   14.277
ATOM        5   C5   PIM      1        8.823   -6.078   13.931
ATOM        6   C6   PIM      1        8.244   -6.729   12.867
ATOM        7   N7   PIM      1        8.692   -3.936   15.271
ATOM        8   C8   PIM      1        7.940   -2.806   15.211
ATOM        9   O9   PIM      1        7.973   -1.830   15.945
ATOM       10   N10  PIM      1        7.084   -2.987   14.142
ATOM       11   C11  PIM      1        6.115   -1.989   13.673
ATOM       12   C12  PIM      1        6.756   -0.732   13.153
ATOM       13   C13  PIM      1        5.718    0.187   12.587
ATOM       14   N14  PIM      1        4.675    0.516   13.600
ATOM       15   C15  PIM      1        4.044   -0.708   14.142
ATOM       16   C16  PIM      1        5.090   -1.631   14.730
ATOM       17   C17  PIM      1        3.658    1.517   13.103
ATOM       18   C18  PIM      1        2.753    1.017   11.992
ATOM       19   C19  PIM      1        1.912    2.187   11.514
ATOM       20   C20  PIM      1        0.842    1.831   10.501
ATOM       21   C21  PIM      1        1.364    1.110    9.267
ATOM       22   C22  PIM      1        2.309    1.625    8.465
ATOM       23   C23  PIM      1        2.748    0.946    7.319
ATOM       24   C24  PIM      1        2.165   -0.229    7.017
ATOM       25   F25  PIM      1        2.517   -0.917    5.894
```

```
ATOM      26  C26 PIM     1       1.217   -0.774    7.804
ATOM      27  C27 PIM     1       0.832   -0.086    8.931
ATOM      28  C28 PIM     1       0.012    3.060   10.092
ATOM      29  C29 PIM     1      -1.267    2.864    9.661
ATOM      30  C30 PIM     1      -2.099    3.932    9.330
ATOM      31  C31 PIM     1      -1.599    5.179    9.443
ATOM      32  F32 PIM     1      -2.432    6.224    9.138
ATOM      33  C33 PIM     1      -0.341    5.440    9.846
ATOM      34  C34 PIM     1       0.483    4.350   10.173
ATOM      35  H1  PIM     1       6.705   -6.619   11.360
ATOM      36  H2  PIM     1       6.048   -4.419   12.084
ATOM      37  H5  PIM     1       9.531   -6.407   14.381
ATOM      38  H6  PIM     1       8.468   -7.689   12.619
ATOM      39  H7  PIM     1       9.279   -3.955   16.017
ATOM      40  H11 PIM     1       5.605   -2.422   12.933
ATOM      41  H12 PIM     1       7.445   -0.989   12.398
ATOM      42  H12 PIM     1       7.225   -0.283   13.909
ATOM      43  H13 PIM     1       5.192   -0.192   11.926
ATOM      44  H13 PIM     1       6.062    1.100   12.272
ATOM      45  H14 PIM     1       5.238    1.039   14.192
ATOM      46  H15 PIM     1       3.517   -0.272   14.758
ATOM      47  H15 PIM     1       3.607   -1.150   13.468
ATOM      48  H16 PIM     1       5.549   -1.241   15.482
ATOM      49  H16 PIM     1       4.648   -2.442   14.947
ATOM      50  H17 PIM     1       3.108    1.665   13.877
ATOM      51  H17 PIM     1       4.120    2.210   12.776
ATOM      52  H18 PIM     1       3.271    0.696   11.297
ATOM      53  H18 PIM     1       2.137    0.464   12.398
ATOM      54  H19 PIM     1       1.533    2.664   12.335
ATOM      55  H19 PIM     1       2.538    2.684   11.140
ATOM      56  H20 PIM     1       0.226    1.271   11.045
ATOM      57  H22 PIM     1       2.670    2.422    8.685
ATOM      58  H23 PIM     1       3.375    1.332    6.734
ATOM      59  H26 PIM     1       0.749   -1.483    7.584
ATOM      60  H27 PIM     1       0.006   -0.353    9.566
ATOM      61  H29 PIM     1      -1.664    2.038    9.661
ATOM      62  H30 PIM     1      -2.953    3.764    9.000
ATOM      63  H33 PIM     1       0.051    6.326    9.944
ATOM      64  H34 PIM     1       1.375    4.581   10.447
TER
```

SYBYL MOL2 file "sybfil," sample output from IDTOSYB and input to SYBYL

```
@<TRIPOS>MOLECULE
-                                                          PIMOZD
    64     68
  SMALL
NO_CHARGES
@<TRIPOS>ATOM
      1  C1        7.185   -6.133   12.162 C.2
      2  C2        6.698   -4.880   12.508 C.2
      3  C3        7.302   -4.231   13.563 C.2
      4  C4        8.335   -4.840   14.277 C.2
      5  C5        8.823   -6.078   13.931 C.2
      6  C6        8.244   -6.729   12.867 C.2
      7  N7        8.692   -3.936   15.271 N.p13
      8  C8        7.940   -2.806   15.211 C.2
      9  O9        7.973   -1.830   15.945 O.2
     10  N10       7.084   -2.987   14.142 N.p13
     11  C11       6.115   -1.989   13.673 C.3
     12  C12       6.756   -0.732   13.153 C.3
```

```
      13  C13       5.718    0.187   12.587 C.3
      14  N14       4.675    0.516   13.600 N.4
      15  C15       4.044   -0.708   14.142 C.3
      16  C16       5.090   -1.631   14.730 C.3
      17  C17       3.658    1.517   13.103 C.3
      18  C18       2.753    1.017   11.992 C.3
      19  C19       1.912    2.187   11.514 C.3
      20  C20       0.842    1.831   10.501 C.3
      21  C21       1.364    1.110    9.267 C.2
      22  C22       2.309    1.625    8.465 C.2
      23  C23       2.748    0.946    7.319 C.2
      24  C24       2.165   -0.229    7.017 C.2
      25  F25       2.517   -0.917    5.894 F
      26  C26       1.217   -0.774    7.804 C.2
      27  C27       0.832   -0.086    8.931 C.2
      28  C28       0.012    3.060   10.092 C.2
      29  C29      -1.267    2.864    9.661 C.2
      30  C30      -2.099    3.932    9.330 C.2
      31  C31      -1.599    5.179    9.443 C.2
      32  F32      -2.432    6.224    9.138 F
      33  C33      -0.341    5.440    9.846 C.2
      34  C34       0.483    4.350   10.173 C.2
      35  H1        6.705   -6.619   11.360 H
      36  H2        6.048   -4.419   12.084 H
      37  H5        9.531   -6.407   14.381 H
      38  H6        8.468   -7.689   12.619 H
      39  H7        9.279   -3.955   16.017 H
      40  H11       5.605   -2.422   12.933 H
      41  H12       7.445   -0.989   12.398 H
      42  H12       7.225   -0.283   13.909 H
      43  H13       5.192   -0.192   11.926 H
      44  H13       6.062    1.100   12.272 H
      45  H14       5.238    1.039   14.192 H
      46  H15       3.517   -0.272   14.758 H
      47  H15       3.607   -1.150   13.468 H
      48  H16       5.549   -1.241   15.482 H
      49  H16       4.648   -2.442   14.947 H
      50  H17       3.108    1.665   13.877 H
      51  H17       4.120    2.210   12.776 H
      52  H18       3.271    0.696   11.297 H
      53  H18       2.137    0.464   12.398 H
      54  H19       1.533    2.664   12.335 H
      55  H19       2.538    2.684   11.140 H
      56  H20       0.226    1.271   11.045 H
      57  H22       2.670    2.422    8.685 H
      58  H23       3.375    1.332    6.734 H
      59  H26       0.749   -1.483    7.584 H
      60  H27       0.006   -0.353    9.566 H
      61  H29      -1.664    2.038    9.661 H
      62  H30      -2.953    3.764    9.000 H
      63  H33       0.051    6.326    9.944 H
      64  H34       1.375    4.581   10.447 H
@<TRIPOS>BOND
     1     1    2  1
     2     1    6  1
     3     1   35  1
     4     2    3  1
     5     2   36  1
     6     3    4  1
     7     3   10  1
     8     4    5  1
     9     4    7  1
    10     5    6  1
    11     5   37  1
```

12	6	38	1
13	7	8	1
14	7	39	1
15	8	9	1
16	8	10	1
17	10	11	1
18	11	12	1
19	11	16	1
20	11	40	1
21	12	13	1
22	12	41	1
23	12	42	1
24	13	14	1
25	13	43	1
26	13	44	1
27	14	15	1
28	14	17	1
29	14	45	1
30	15	16	1
31	15	46	1
32	15	47	1
33	16	48	1
34	16	49	1
35	17	18	1
36	17	50	1
37	17	51	1
38	18	19	1
39	18	52	1
40	18	53	1
41	19	20	1
42	19	54	1
43	19	55	1
44	20	21	1
45	20	28	1
46	20	56	1
47	21	22	1
48	21	27	1
49	22	23	1
50	22	57	1
51	23	24	1
52	23	58	1
53	24	25	1
54	24	26	1
55	26	27	1
56	26	59	1
57	27	60	1
58	28	29	1
59	28	34	1
60	29	30	1
61	29	61	1
62	30	31	1
63	30	62	1
64	31	32	1
65	31	33	1
66	33	34	1
67	33	63	1
68	34	64	1

References

1. E. C. Meng, B. K. Shoichet, and I. D. Kuntz, *J. Comp. Chem.*, **13,** 505 (1992).

2. R. S. Pearlman, *Chem. Design Automation News,* **2,** 1 (1987).

3. R. S. Pearlman, A. Rusinko, J. M. Skell, and R. Balducci, 'CONCORD User's Manual,' Tripos Associates, St. Louis, 1988.

4. Molecular Design Limited, San Leandro, CA 94577.

5. F. H. Allen, O. Kennard, W. D. S. Motherwell, W. G. Town, and D. G. Watson, *J. Chem. Doc.*, **13,** 119 (1973).

6. E. C. Meng and R. A. Lewis, *J. Comp. Chem.*, **12,** 891 (1991).

7. SYBYL, Tripos Associates, St. Louis, MO 63117.

8. F. C. Bernstein, T. F. Koetzle, G. J. B. Williams, E. F. Meyer, Jr., M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi, *J. Mol. Biol.*, **112,** 535 (1977).

9. J. Gasteiger and M. Marsili, *Tetrahedron,* **36,** 3219 (1980).

10. M. Marsili and J. Gasteiger, *Croat. Chem. Acta,* **53,** 601 (1980).

11. J. Gasteiger and M. Marsili, *Organ. Magn. Reson.*, **15,** 353 (1981).

APPENDIX 4: IDATM SOURCE CODE

```
c                  PROGRAM IDATM                    Elaine Meng
c
c        Copyright (C) 1990 Regents of the University of California
c                           All Rights Reserved.
c
c     This program determines the type of each atom in the input
c     pdb file (types are defined according to hybridization,
c     oxidation state, and/or neighbor atom type).  Hydrogens need
c     not be present.  More than one pdb file can be handled as long
c     as the files are separated by blank lines or TER cards.  The
c     presence of a bond is determined as in the Cambridge Structural
c     Database (CSD) GEOM78 program:  there is a bond between the ith
c     atom and the jth atom if the distance between them is less than
c     or equal to the sum of their covalent bond radii plus some
c     tolerance.  The output is the same as the pdb input file except
c     that the atom identifier at (13:16) is replaced by the atom type
c     as determined in this program at (13:15) and a space at (16:16).
c     All distance criteria used for typing were derived from careful
c     inspection of the bond length data in:  Allen, Kennard, and Watson,
c     J. Chem. Soc. Perkin Trans. II (1987), S1-S19.  This paper is
c     essentially a tabulation and statistical analysis of bond lengths;
c     the 10,324 structures used were present in the September 1985
c     version of the CSD and met the following stipulations:
c       1)structure is "organic" (CSD classes 1-65, 70)
c       2)atomic coordinates for the structure have been published
c       3)structure was determined from diffractometer data
c       4)structure does not contain unresolved numeric data errors
c         from the original publication
c       5)structure was not reported to be disordered
c       6)R factor .le. 0.07 and estimated standard deviation (esd) for
c         C-C bond lengths .le. 0.01 angstroms, OR, if the esd for C-C
c         bond lengths was not reported, R factor .le. 0.05
c       7)structure corresponds to the most precise determination for
c         that compound
c
c All of the code for IDATM is in this file; two parameter files are
c necessary (see below).  In this version, the input file must be named
c "pdbfil" and the output file will be named "pdbtyp."
c
c The 7 lines between the dashed lines below must be placed in a file
c called "attyps".  "c " must be removed from the beginning of each line.
c-----------------------------------------------------------------------
c C3 C2 C1 Cac
c N3+N3 Npl Nl NoxNtrNg+
c O3 O2 O-
c S3+S3 S2 SacSoxS
c BacBoxB
c PacPoxP3+P
c HC H  DC D
c-----------------------------------------------------------------------
c
c The 8 lines between the dashed lines below must be placed in a file
c called "params".  "c " must be removed from the beginning of each line.
c-----------------------------------------------------------------------
c 115.0  122.0  160.0
c 1.22  1.41  1.37
c 1.20  1.38  1.43  1.41
c 1.30  1.685
c 1.76  2.11
c 1.53  1.46  1.44
c 1.38  1.32
c 1.42  1.41  1.45
c-----------------------------------------------------------------------
c
c     maxatm--maximum number of atoms allowed per molecule or fragment
```

```
c     atms--number of atoms in the molecule or fragment
c     i, j, k, l, m, n--integer variables used as do loop variables,
c       "links" numbers (a "link" number is the sequential number of an
c       atom bonded to the atom currently being considered), and counters
c     dist(i,j)--distance between the ith atom and the jth atom
c     coninf(i)--covalent bond radius for atom i (depends on element
c       only); the values are those used in the CSD GEOM78 program
c     toler--tolerance in defining the presence of a bond, described
c       above
c     pdblin--current line in the pdb file being read
c     kplin(i)--line in the pdb file corresponding to the ith atom
c     maxval--maximum valence allowed
c     links(i,n)--the index (such as j) of the nth atom bonded to the
c       ith atom
c     valnce(i)--the number of atoms bonded to the ith atom
c     hvys(i)--the number of nonhydrogen atoms bonded to the ith atom
c     atmtyp(i)--the type of the ith atom, as determined by this program
c     hyds--logical variable indicating whether hydrogens are present
c       in the pdb file
c     finish--logical variable indicating whether the end of the input
c       has been reached
c     noplus--logical variable used to distinguish guanidine and related
c       structures (noplus=true) from a guanidinium group (noplus=false)
c     freeos--the number of oxygens that are bonded to the current atom
c       but that are not bonded to any other heavy atoms; used to dis-
c       tinguish acids from esters and to identify other oxidation states
c     redo(i)--integer variable indicating whether more analysis is
c       needed to determine the type of the ith atom; 0 means no further
c       consideration is deemed necessary; 1 means tentatively c3 but
c       further analysis is needed; 2 means tentatively n3 but further
c       analysis is needed; 3 means tentatively c2 but further analysis
c       is needed
c     ang--bond angle of a valence 2 atom, or average bond angle of a
c       valence 3 atom
c     ang1, ang2, ang3--bond angles of a valence 3 atom
c     ang23a--sp2 versus sp3 angle cutoff
c     ang23b--angle below which the valence atom assignment is marked
c       as a redo (initial assignment sp3 but not certain)
c     ang12--sp versus sp2 angle cutoff
c     v1*--bond length cutoffs for valence 1 atoms; the third and fourth
c       characters refer to the atom of interest; the fifth (and sixth,
c       if any) characters refer to the bond partner of this atom; if '3'
c       is included in the name, the value is a lower cutoff; otherwise,
c       it is an upper cutoff
c     v2*--bond length cutoffs for valence 2 atoms; the third and fourth
c       characters refer to the atom of interest; the fifth (and sixth,
c       if any) characters refer to a neighbor atom; if '3' is included
c       in the name, the value is a lower cutoff; otherwise, it is an
c       upper cutoff
c     c3ccnd--weak (conditional) lower bond length cutoff defining sp3
c       carbon bonded to carbon; can be overridden by another cutoff
c     bndrad--real function that associates each atom with a covalent
c       bond radius (depends on element only, i.e., not on hybridization
c       or oxidation state)
c     bndlen--real function that calculates the distance between two
c       atoms
c     angle--real function that calculates the angle between three atoms
c
      integer maxatm
      parameter (maxatm=150)
      integer atms, i, j, k, l, m, n
      real dist(maxatm,maxatm), coninf(maxatm)
      real toler
      parameter (toler=0.40)
      character*80 pdblin, kplin(maxatm)
      integer maxval
      parameter (maxval=4)
      integer links(maxatm,maxval), valnce(maxatm), hvys(maxatm)
      character*3 atmtyp(maxatm)
      logical hyds, finish, noplus, log1, log2
      integer freeos, redo(maxatm)
      real ang, ang1, ang2, ang3
```

```
      real ang23a, ang23b, ang12
      real v1c1c1, v1c2c, v1c2n
      real v1n1c1, v1n3c, v1n3n3, v1n3n2
      real v1o2c2, v1o2as
      real v1s2c2, v1s2as
      real v2c3c, v2c3n, v2c3o
      real v2n2c, v2n2n
      real v2c2c, v2c2n, c3ccnd
      character*3 c3, c2, c1, cac
      character*3 n3p, n3, npl, n1, nox, ntr, ngp
      character*3 o3, o2, om
      character*3 s3p, s3, s2, sac, sox, s
      character*3 bac, box, b
      character*3 pac, pox, p3p, p
      character*3 hc, h, dc, d
      real bndrad
      real bndlen
      real angle
c
c --open the input and output files; initialize variables.
c
      open (unit=1, file='pdbfil', status='old')
      open (unit=3, file='pdbtyp', status='new')
      finish=.false.
      open (unit=2, file='params', status='old')
      read (2, *) ang23a, ang23b, ang12
      read (2, *) v1c1c1, v1c2c, v1c2n
      read (2, *) v1n1c1, v1n3c, v1n3n3, v1n3n2
      read (2, *) v1o2c2, v1o2as
      read (2, *) v1s2c2, v1s2as
      read (2, *) v2c3c, v2c3n, v2c3o
      read (2, *) v2n2c, v2n2n
      read (2, *) v2c2c, v2c2n, c3ccnd
      close (2)
      open (unit=2, file='attyps', status='old')
      read (2, '(4A3)') c3, c2, c1, cac
      read (2, '(7A3)') n3p, n3, npl, n1, nox, ntr, ngp
      read (2, '(3A3)') o3, o2, om
      read (2, '(6A3)') s3p,s3, s2, sac, sox, s
      read (2, '(3A3)') bac, box, b
      read (2, '(4A3)') pac, pox, p3p, p
      read (2, '(4A3)') hc, h, dc, d
      close (2)
    5 do 10 i=1, maxatm
        coninf(i)=0.0
        valnce(i)=0
        atmtyp(i)='xxx'
        redo(i)=0
        do 8 j=1, 4
          links(i,j)=0
    8   continue
        do 9 j=1, maxatm
          dist(i,j)=0.0
    9   continue
   10 continue
      atms=0
      hyds=.false.
c
c --read lines from the input file, writing remarks to the output;
c   ignore lines that are not ATOM or HETATM records or that do not
c   signify the end of a molecule or fragment.  Blank lines and TER
c   cards signify the end of a molecule or fragment; go to distance
c   calculations when these are encountered.  When the line is an ATOM
c   or HETATM record, store the line in array 'kplin', increment the
c   atom count, and check for too many atoms.
c
   20 continue
      read (1, 1000, end=900) pdblin
 1000 format (A80)
      if (pdblin(1:4) .eq. 'REMA') then
        write (3, 1000) pdblin
        go to 20
```

```
      endif
      if (pdblin(1:4) .ne. 'ATOM' .and. pdblin(1:4) .ne. 'HETA'
     +.and. pdblin(1:4) .ne. '    ' .and. pdblin(1:4) .ne. 'TER ')
     +go to 20
      if (pdblin(1:4) .eq. '    ' .or. pdblin(1:4) .eq. 'TER ')
     +go to 50
      atms=atms + 1
      kplin(atms)=pdblin
      if (atms .gt. maxatm) then
        write (6, 1001) 'Error--too many atoms in fragment'
 1001   format (A33)
        stop
      endif
c
c --if the line corresponds to a hydrogen or deuterium atom, the
c   pdb file contains at least some of the hydrogens of the molecule.
c
      if (pdblin(14:14) .eq. 'H' .or. pdblin(14:14) .eq. 'D')
     +then
        hyds=.true.
      endif
c
c --associate the atom that corresponds to the current line with the
c   appropriate covalent bond radius.  If this cannot be done, write
c   an error message to the terminal and stop.
c
      coninf(atms)=bndrad(pdblin)
      if (coninf(atms) .eq. 0.0) then
        write (6, 1003) 'Error--no bond length information for atom ',
     +  pdblin(13:14)
 1003   format (A43, A2)
        stop
      endif
      go to 20
c
c --when all the lines for the molecule or fragment have been stored,
c   calculate all pairwise distances and determine which distances
c   are consistent with covalent bonding.  A distance is considered
c   to be consistent with a covalent bond if it is less than or equal
c   to the sum of the covalent bond radii of the two atoms plus some
c   tolerance.  Increment the valence of each atom appropriately, and
c   store information on what is bonded to what in the array 'links'.
c   Caution the user if the valence of an atom exceeds the maximum
c   valence allowed (which is a parameter, and thus can be changed
c   relatively easily by the user).  Nonbonded distances are given a
c   negative sign; this might be useful if more features are added to
c   the program.
c
   50 continue
      do 80 i=1, atms
        do 70 j=1, atms
          dist(i,j)=bndlen(kplin(i), kplin(j))
          if (i .ne. j) then
            if (dist(i,j) .le. (coninf(i) +
     &      coninf(j) + toler)) then
              valnce(i)=valnce(i) + 1
              if (valnce(i) .le. maxval) then
                links(i,valnce(i))=j
              else
                write (6, 1004) 'Caution--valence>maxval for atom ',
     +          kplin(i)(7:11), ' in ', kplin(i)(18:20)
 1004           format (A33, A5, A4, A3)
              endif
            else
              dist(i,j)=0.0 - dist(i,j)
            endif
          endif
   70   continue
   80 continue
c---------------------------------------------------------------------
c --FIRST PASS--type all hydrogens and deuteriums by whether they are
c   attached to carbon or not; calculate the number of heavy atoms bonded
```

```
c   to each atom by subtracting the number of hydrogens attached from the
c   valence.
c
      do 102 i=1, atms
        hvys(i)=valnce(i)
  102 continue
      do 105 i=1, atms
        if (kplin(i)(14:14) .eq. 'H') then
          k=links(i,1)
          hvys(k)=hvys(k) - 1
          if (kplin(k)(14:14) .eq. 'C') then
            atmtyp(i)=hc
          else
            atmtyp(i)=h
          endif
        else if (kplin(i)(14:14) .eq. 'D') then
          k=links(i,1)
          hvys(k)=hvys(k) - 1
          if (kplin(k)(14:14) .eq. 'C') then
            atmtyp(i)=dc
          else
            atmtyp(i)=d
          endif
        endif
  105 continue
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
c
c --SECOND PASS--type all atoms whose type depends only on the element;
c   handle other atoms after grouping them by valence.
c
      do 500 i=1,atms
        log1=(kplin(i)(13:14) .eq. 'AC' .or. kplin(i)(13:14) .eq.
     +  'AG' .or. kplin(i)(13:14) .eq. 'AL' .or. kplin(i)(13:14)
     +  .eq. 'AM' .or. kplin(i)(13:14) .eq. 'AS' .or. kplin(i)
     +  (13:14) .eq. 'AU' .or. kplin(i)(13:14) .eq. 'BA' .or.
     +  kplin(i)(13:14) .eq. 'BE' .or. kplin(i)(13:14) .eq. 'BI'
     +  .or. kplin(i)(13:14) .eq. 'BR' .or. kplin(i)(13:14) .eq.
     +  'CA' .or. kplin(i)(13:14) .eq. 'CD' .or. kplin(i)(13:14)
     +  .eq. 'CE' .or. kplin(i)(13:14) .eq. 'CL' .or. kplin(i)
     +  (13:14) .eq. 'CO' .or. kplin(i)(13:14) .eq. 'CR' .or.
     +  kplin(i)(13:14) .eq. 'CS' .or. kplin(i)(13:14) .eq. 'CU'
     +  .or. kplin(i)(13:14) .eq. 'DY' .or. kplin(i)(13:14) .eq.
     +  'ER' .or. kplin(i)(13:14) .eq. 'EU' .or. kplin(i)(13:14)
     +  .eq. 'FE' .or. kplin(i)(13:14) .eq. 'GA' .or. kplin(i)
     +  (13:14) .eq. 'GD' .or. kplin(i)(13:14) .eq. 'GE' .or.
     +  kplin(i)(13:14) .eq. 'HF' .or. kplin(i)(13:14) .eq. 'HG'
     +  .or. kplin(i)(13:14) .eq. 'HO' .or. kplin(i)(13:14) .eq.
     +  'IN' .or. kplin(i)(13:14) .eq. 'IR' .or. kplin(i)(13:14)
     +  .eq. 'LA' .or. kplin(i)(13:14) .eq. 'LI' .or. kplin(i)
     +  (13:14) .eq. 'LU' .or. kplin(i)(13:14) .eq. 'MG')
        log2=(kplin(i)(13:14) .eq. 'MN' .or. kplin(i)(13:14) .eq.
     +  'MO' .or. kplin(i)(13:14) .eq. 'NA' .or. kplin(i)(13:14) .eq.
     +  'NB' .or. kplin(i)(13:14) .eq. 'ND' .or. kplin(i)(13:14)
     +  .eq. 'NI' .or. kplin(i)(13:14) .eq. 'NP' .or. kplin(i)
     +  (13:14) .eq. 'OS' .or. kplin(i)(13:14) .eq. 'PA' .or.
     +  kplin(i)(13:14) .eq. 'PB' .or. kplin(i)(13:14) .eq. 'PD'
     +  .or. kplin(i)(13:14) .eq. 'PM' .or. kplin(i)(13:14) .eq.
     +  'PO' .or. kplin(i)(13:14) .eq. 'PR' .or. kplin(i)(13:14)
     +  .eq. 'PT' .or. kplin(i)(13:14) .eq. 'PU' .or. kplin(i)
     +  (13:14) .eq. 'RA' .or. kplin(i)(13:14) .eq. 'RB' .or.
     +  kplin(i)(13:14) .eq. 'RE' .or. kplin(i)(13:14) .eq. 'RH'
     +  .or. kplin(i)(13:14) .eq. 'RU' .or. kplin(i)(13:14) .eq.
     +  'SB' .or. kplin(i)(13:14) .eq. 'SC' .or. kplin(i)(13:14)
     +  .eq. 'SE' .or. kplin(i)(13:14) .eq. 'SI' .or. kplin(i)
     +  (13:14) .eq. 'SM' .or. kplin(i)(13:14) .eq. 'SN')
        if (kplin(i)(13:14) .eq. 'SR' .or. kplin(i)(13:14) .eq. 'TA'
     +  .or. kplin(i)(13:14) .eq. 'TB' .or. kplin(i)(13:14) .eq.
     +  'TC' .or. kplin(i)(13:14) .eq. 'TE' .or. kplin(i)(13:14)
     +  .eq. 'TH' .or. kplin(i)(13:14) .eq. 'TI' .or. kplin(i)
     +  (13:14) .eq. 'TL' .or. kplin(i)(13:14) .eq. 'TM' .or.
     +  kplin(i)(13:14) .eq. 'YB' .or. kplin(i)(13:14) .eq. 'ZN'
```

```
     +  .or. kplin(i)(13:14) .eq. 'ZR' .or. log1 .or. log2) then
          atmtyp(i)=kplin(i)(13:14)//' '
        else if (kplin(i)(13:14) .eq. ' U' .or. kplin(i)(13:14)
     +  .eq. ' V' .or. kplin(i)(13:14) .eq. ' W' .or. kplin(i)
     +  (13:14) .eq. ' Y') then
          atmtyp(i)=kplin(i)(14:14)//'  '
c-----------------------------------------------------------------------
c --valence 4: C must be sp3 (c3); N must be part of an N-oxide (nox)
c   or a quaternary amine (n3p); P must be part of a phosphate (pac),
c   a P-oxide (pox), or a quaternary phosphine (p3p); S must be part
c   of a sulfate (sac), a sulfone (sox), or a sulfoxide (also sox);
c   B may be part of a borate (bac), or may be another oxidized form
c   (box), or may not be oxidized (b).
c
        else if (valnce(i) .eq. 4) then
          if (kplin(i)(14:14) .eq. 'C') then
            atmtyp(i)=c3
          else if (kplin(i)(14:14) .eq. 'N') then
            freeos=0
            do 110 j=1, 4
              if (kplin(links(i,j))(14:14) .eq. 'O' .and. hvys
     +        (links(i,j)) .eq. 1) then
                freeos=freeos + 1
              endif
  110       continue
            if (freeos .ge. 1) then
              atmtyp(i)=nox
            else
              atmtyp(i)=n3p
            endif
          else if (kplin(i)(14:14) .eq. 'P') then
            freeos=0
            do 111 j=1, 4
              if (kplin(links(i,j))(14:14) .eq. 'O' .and. hvys
     +        (links(i,j)) .eq. 1) then
                freeos=freeos + 1
              endif
  111       continue
            if (freeos .ge. 2) then
              atmtyp(i)=pac
            else if (freeos .eq. 1) then
              atmtyp(i)=pox
            else
              atmtyp(i)=p3p
            endif
          else if (kplin(i)(14:14) .eq. 'B' .or. kplin(i)(14:14)
     +    .eq. 'S') then
            freeos=0
            do 112 j=1, 4
              if (kplin(links(i,j))(14:14) .eq. 'O' .and. hvys
     +        (links(i,j)) .eq. 1) then
                freeos=freeos + 1
              endif
  112       continue
            if (freeos .ge. 3) then
              if (kplin(i)(14:14) .eq. 'B') then
                atmtyp(i)=bac
              else
                atmtyp(i)=sac
              endif
            else if (freeos .ge. 1) then
              if (kplin(i)(14:14) .eq. 'B') then
                atmtyp(i)=box
              else
                atmtyp(i)=sox
              endif
            else
              if (kplin(i)(14:14) .eq. 'B') then
                atmtyp(i)=b
              else
                atmtyp(i)=s
              endif
```

```
          endif
        endif
c-----------------------------------------------------------------------
c --valence 3: calculate the three bond angles and average them.  Since
c   the input may not have hydrogens or may be missing a few hydrogens,
c   hybridization cannot be determined on the basis of valence alone; the
c   average bond angle is used to help discriminate between the possible
c   types for each atom.  C may be sp3 (c3), sp2 (c2), or part of a
c   carboxylate (cac); N may be sp3 (n3), sp2 or planar (as in amides
c   and aniline derivatives; npl), or part of a nitro group (ntr); S
c   may be, depending on oxidation state, sox or s3p; B may be, depending
c   on oxidation state, box or b.
c
        else if (valnce(i) .eq. 3) then
          k=links(i,1)
          l=links(i,2)
          m=links(i,3)
          ang1=angle(kplin(k), kplin(i), kplin(l))
          ang2=angle(kplin(k), kplin(i), kplin(m))
          ang3=angle(kplin(l), kplin(i), kplin(m))
          ang=(ang1 + ang2 + ang3)/3.0
          if (kplin(i)(14:14) .eq. 'C') then
            if (ang .lt. ang23a) then
              atmtyp(i)=c3
            else
              freeos=0
              do 116 j=1, 3
                if (kplin(links(i,j))(14:14) .eq. 'O' .and. hvys
     +          (links(i,j)) .eq. 1) then
                  freeos=freeos + 1
                endif
  116           continue
              if (freeos .ge. 2) then
                atmtyp(i)=cac
              else
                atmtyp(i)=c2
              endif
            endif
          else if (kplin(i)(14:14) .eq. 'N') then
            if (ang .lt. ang23a) then
              atmtyp(i)=n3
            else
              freeos=0
              do 117 j=1, 3
                if (kplin(links(i,j))(14:14) .eq. 'O' .and. hvys
     +          (links(i,j)) .eq. 1) then
                  freeos=freeos + 1
                endif
  117           continue
              if (freeos .ge. 2) then
                atmtyp(i)=ntr
              else
                atmtyp(i)=npl
              endif
            endif
          else if (kplin(i)(14:14) .eq. 'B' .or. kplin(i)(14:14)
     +    .eq. 'S') then
            do 120 j=1, 3
              k=links(i,j)
              if (kplin(k)(14:14) .eq. 'O') then
                if (kplin(i)(14:14) .eq. 'B') then
                  atmtyp(i)=box
                else if (kplin(i)(14:14) .eq. 'S') then
                  atmtyp(i)=sox
                endif
                go to 125
              endif
  120         continue
            if (kplin(i)(14:14) .eq. 'B') atmtyp(i)=b
            if (kplin(i)(14:14) .eq. 'S') atmtyp(i)=s3p
  125       continue
          endif
```

```
c-----------------------------------------------------------------------
c --valence 2: calculate the bond angle and assign a tentative atom type
c   accordingly (a single angle is often not a good indicator of type).
c   Mark these atoms for further consideration by placing nonzero values
c   in the array 'redo'.  C may be sp3 (c3), sp2 (c2), or sp (c1);
c   N may be sp3 (n3), sp2 or planar (npl), or sp (n1); O and S are
c   sp3 (o3 and s3, respectively).  If any atom has not been typed
c   yet, type it by element.  The lines that are commented out are useful
c   in determining the causes of typing errors.
c
        else if (valnce(i) .eq. 2) then
          k=links(i,1)
          l=links(i,2)
          ang=angle(kplin(k), kplin(i), kplin(l))
          if (kplin(i)(14:14) .eq. 'C') then
            if (ang .lt. ang23a) then
              atmtyp(i)=c3
              redo(i)=1
            else if (ang .lt. ang12) then
              atmtyp(i)=c2
              if (ang .lt. ang23b) then
                redo(i)=3
              endif
            else
              atmtyp(i)=c1
            endif
          else if (kplin(i)(14:14) .eq. 'N') then
            if (ang .lt. ang23a) then
              atmtyp(i)=n3
              redo(i)=2
            else if (ang .lt. ang12) then
              atmtyp(i)=npl
            else
              atmtyp(i)=n1
            endif
          else if (kplin(i)(14:14) .eq. 'O') then
            atmtyp(i)=o3
          else if (kplin(i)(14:14) .eq. 'S') then
            atmtyp(i)=s3
          endif
        endif
        if (atmtyp(i) .eq. 'xxx') then
          atmtyp(i)=kplin(i)(14:14)//'  '
        endif
        ang=0.0
c
c --test writing to list geometrical data
c
c --     write (3, '(I3, x, A3, x, F6.2, 4F8.3)') i, atmtyp(i), ang,
c -- +  (dist(i,links(i,j)), j=1, valnce(i))
  500 continue
c --  write (3, '(A3)') '   '
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
c --THIRD PASS--determine the types of valence 1 atoms; these were typed
c   by element only in the previous pass, but can be typed more accurately
c   now that the atoms they are bonded to have been typed.  Bond lengths are
c   used in this pass.  The names of the types are as explained above.
c
      do 600 i=1, atms
        if (valnce(i) .eq. 1 .and. atmtyp(i) .eq. 'C  ') then
          j=links(i,1)
          if (dist(i,j) .le. v1c1c1 .and. atmtyp(j) .eq. c1) then
            atmtyp(i)=c1
          else if (dist(i,j) .le. v1c2c .and. kplin(j)(14:14) .eq. 'C')
     +    then
            atmtyp(i)=c2
          else if (dist(i,j) .le. v1c2n .and. kplin(j)(14:14) .eq. 'N')
     +    then
            atmtyp(i)=c2
          else
            atmtyp(i)=c3
```

```
          endif
        else if (valnce(i) .eq. 1 .and. atmtyp(i) .eq. 'N  ') then
          j=links(i,1)
          if (dist(i,j) .le. v1n1c1 .and. atmtyp(j) .eq. c1) then
            atmtyp(i)=n1
          else if (dist(i,j) .gt. v1n3c .and. (atmtyp(j) .eq. c2 .or.
     +    atmtyp(j) .eq. c3)) then
            atmtyp(i)=n3
          else if ((dist(i,j) .gt. v1n3n3 .and. atmtyp(j) .eq. n3)
     +    .or. (dist(i,j) .gt. v1n3n2 .and. atmtyp(j) .eq. np1)) then
            atmtyp(i)=n3
          else
            atmtyp(i)=np1
          endif
        else if (valnce(i) .eq. 1 .and. atmtyp(i) .eq. 'O  ') then
          j=links(i,1)
          if (atmtyp(j) .eq. cac .or. atmtyp(j) .eq. pac .or.
     +    atmtyp(j) .eq. sac .or. atmtyp(j) .eq. ntr) then
            atmtyp(i)=om
          else if (atmtyp(j) .eq. nox .or. atmtyp(j) .eq. pox
     +    .or. atmtyp(j) .eq. sox) then
            atmtyp(i)=o2
          else if (dist(i,j) .le. v1o2c2 .and. kplin(j)(14:14) .eq. 'C')
     +    then
            atmtyp(i)=o2
            atmtyp(j)=c2
            redo(j)=0
          else if (dist(i,j) .le. v1o2as .and. atmtyp(j) .eq. 'AS ')
     +    then
            atmtyp(i)=o2
          else
            atmtyp(i)=o3
          endif
        else if (valnce(i) .eq. 1 .and. atmtyp(i) .eq. 'S  ') then
          j=links(i,1)
          if (kplin(j)(14:14) .eq. 'P') then
            atmtyp(i)=s2
          else if (dist(i,j) .le. v1s2c2 .and. kplin(j)(14:14) .eq. 'C')
     +    then
            atmtyp(i)=s2
            atmtyp(j)=c2
            redo(j)=0
          else if (dist(i,j) .le. v1s2as .and. atmtyp(j) .eq. 'AS ')
     +    then
            atmtyp(i)=s2
          else
            atmtyp(i)=s3
          endif
        endif
  600 continue
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
c --FOURTH PASS--reexamine all atoms associated with nonzero 'redo'values,
c   and retype them if necessary.
c
      do 700 i=1, atms
        if (redo(i) .eq. 1) then
          do 610 j=1, valnce(i)
            k=links(i,j)
            if ((dist(i,k) .le. v2c2c .and. kplin(k)(14:14) .eq.
     +      'C') .or. (dist(i,k) .le. v2c2n .and. kplin(k)(14:14)
     +      .eq. 'N')) then
              atmtyp(i)=c2
            endif
  610     continue
          do 611 j=1, valnce(i)
            k=links(i,j)
            if ((dist(i,k) .gt. v2c3c .and. kplin(k)(14:14) .eq. 'C')
     +      .or. (dist(i,k) .gt. v2c3n .and. kplin(k)(14:14) .eq. 'N')
     +      .or. (dist(i,k) .gt. v2c3o .and. kplin(k)(14:14) .eq. 'O'))
     +      then
              atmtyp(i)=c3
```

```
              go to 635
            endif
  611       continue
          else if (redo(i) .eq. 2) then
            do 620 j=1, valnce(i)
              k=links(i,j)
              if ((dist(i,k) .le. v2n2c .and. kplin(k)(14:14) .eq.
     +        'C') .or. (dist(i,k) .le. v2n2n .and. kplin(k)(14:14)
     +        .eq. 'N')) then
                atmtyp(i)=npl
              endif
  620       continue
          else if (redo(i) .eq. 3) then
            do 630 j=1, valnce(i)
              k=links(i,j)
              if ((dist(i,k) .le. v2c2c .and. kplin(k)(14:14) .eq.
     +        'C') .or. (dist(i,k) .le. v2c2n .and. kplin(k)(14:14)
     +        .eq. 'N')) then
                atmtyp(i)=c2
                go to 635
              endif
  630       continue
            do 631 j=1, valnce(i)
              k=links(i,j)
              if ((dist(i,k) .gt. v2c3c .and. kplin(k)(14:14) .eq. 'C')
     +        .or. (dist(i,k) .gt. v2c3n .and. kplin(k)(14:14) .eq. 'N')
     +        .or. (dist(i,k) .gt. v2c3o .and. kplin(k)(14:14) .eq. 'O'))
     +        then
                atmtyp(i)=c3
                go to 635
              endif
              if (dist(i,k) .gt. c3ccnd .and. kplin(k)(14:14) .eq. 'C')
     +        then
                atmtyp(i)=c3
              endif
  631       continue
  635     continue
          endif
  700   continue
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
c --FIFTH PASS--change isolated sp2 carbons to sp3 carbons, because it is
c   impossible for an atom to be sp2 hybridized if all the heavy atoms it
c   is bonded to are sp3 hybridized.  In addition, a carbon atom cannot
c   be doubly bonded to a carboxylate carbon, phosphate phosphorus, sulfate
c   sulfur, sulfone sulfur, sulfoxide sulfur, or sp1 carbon.
        do 703 i=1, atms
          if (atmtyp(i) .eq. c2) then
            m=0
            do 701 j=1, valnce(i)
              k=links(i,j)
              if (atmtyp(k) .ne. c3 .and. atmtyp(k) .ne. dc
     +        .and. atmtyp(k) .ne. hc .and. atmtyp(k) .ne. n3
     +        .and. atmtyp(k) .ne. n3p .and. atmtyp(k) .ne. o3
     +        .and. atmtyp(k) .ne. cac .and. atmtyp(k) .ne. pac
     +        .and. atmtyp(k) .ne. sac .and. atmtyp(k) .ne. sox
     +        .and. atmtyp(k) .ne. cl
     +        .and. atmtyp(k) .ne. s3) go to 702
  701       continue
            atmtyp(i)=c3
  702       continue
          endif
  703   continue
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
c --SIXTH PASS--1) make decisions about the charge states of nitrogens.
c   If an sp3 nitrogen is bonded to sp3 carbons and/or hydrogens and/or
c   deuteriums only, assume that it is positively charged (the pKa of its
c   conjugate acid is probably high enough that the protonated form pre-
c   dominates at physiological pH).  If an sp2 carbon is bonded to three
c   planar nitrogens, it may be part of a guanidinium group.  Make the
c   nitrogens positively charged (ngp) if guanidine or similar
```

```
c   structures can be ruled out (if noplus=false).
c               2) make carboxyl oxygens negatively charged even if the
c   proton is present in the input (the pKa of the carboxyl group is
c   probably low enough that the unprotonated form predominates at phys-
c   iological pH).
c
      do 800 i=1, atms
        if (atmtyp(i) .eq. n3) then
          do 710 j=1, valnce(i)
            k=links(i,j)
            if (atmtyp(k) .ne. c3 .and. atmtyp(k) .ne. h
     +      .and. atmtyp(k) .ne. d) go to 715
  710     continue
          atmtyp(i)=n3p
  715     continue
        else if (atmtyp(i) .eq. c2) then
          m=0
          do 720 j=1, valnce(i)
            k=links(i,j)
            if (atmtyp(k) .eq. npl) then
              m=m + 1
            endif
  720     continue
          if (m .eq. 3) then
            noplus=.false.
            do 730 j=1, valnce(i)
              k=links(i,j)
              if (atmtyp(k) .eq. npl) then
                atmtyp(k)=ngp
                do 725 l=1, valnce(k)
                  n=links(k,l)
                  if ((atmtyp(n) .eq. c2 .or. atmtyp(n) .eq.
     +            npl) .and. n .ne. i) then
                    atmtyp(k)=npl
                    noplus=.true.
                  endif
  725           continue
              endif
  730       continue
          endif
          if (noplus) then
            do 735 j=1, valnce(i)
              k=links(i,j)
              if (atmtyp(k) .eq. ngp) then
                atmtyp(k)=npl
              endif
  735       continue
          endif
        else if (atmtyp(i) .eq. cac) then
          do 750 j=1, valnce(i)
            k=links(i,j)
            if (kplin(k)(14:14) .eq. 'O' .and. hvys(k) .eq. 1) then
              atmtyp(k)=om
            endif
  750     continue
        endif
  800 continue
c
c --write to output; the output is the same as the pdb input file
c   except that the atom identifier at (13:16) is replaced by the atom
c   type as determined in this program at (13:15) and a space at (16:16).
c   Separate outputs for different molecules or fragments using blank
c   lines.
c
      do 810 i=1, atms
        write (3, '(A12, A3, A1, A64)') kplin(i)(1:12), atmtyp(i), ' ',
     +  kplin(i)(17:80)
  810 continue
      if (atms .gt. 0) then
        write (3, '(A3)') 'TER'
      endif
c-----------------------------------------------------------------------
```

```
c-----------------------------------------------------------------------
c --if the current molecule or fragment is not the last, go on to the
c   next; if it is the last, close the input and output files.
c
      if (finish .eq. .false.) go to 5
      close (3)
      go to 910
  900 continue
      close (1)
      finish=.true.
      go to 50
  910 continue
      end
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
      real function bndlen(line1, line2)
      character*80 line1, line2
      real x1, y1, z1, x2, y2, z2
c
      read (line1, 2000) x1, y1, z1
      read (line2, 2000) x2, y2, z2
 2000 format (30x, 3F8.3)
      bndlen=sqrt((x2-x1)**2 + (y2-y1)**2 + (z2-z1)**2)
      return
      end
c
c-----------------------------------------------------------------------
c
      real function angle(line1, line2, line3)
      character*80 line1, line2, line3
      real x1, y1, z1, x2, y2, z2, x3, y3, z3
      real radtodeg
      parameter (radtodeg=180.0/3.1415926)
c
      read (line1, 3000) x1, y1, z1
      read (line2, 3000) x2, y2, z2
      read (line3, 3000) x3, y3, z3
 3000 format (30x, 3F8.3)
      costh=((x1-x2)*(x3-x2) + (y1-y2)*(y3-y2) + (z1-z2)*(z3-z2))/
     &(bndlen(line1, line2)*bndlen(line2, line3))
      angle=(acos(costh))*radtodeg
      return
      end
c
c-----------------------------------------------------------------------
c
      real function bndrad(line)
      character*80 line
c
      if (line(13:14) .eq. 'AC') then
        bndrad=1.88
      else if (line(13:14) .eq. 'AG') then
        bndrad=1.59
      else if (line(13:14) .eq. 'AL') then
        bndrad=1.35
      else if (line(13:14) .eq. 'AM') then
        bndrad=1.51
      else if (line(13:14) .eq. 'AS') then
        bndrad=1.21
      else if (line(13:14) .eq. 'AU') then
        bndrad=1.50
      else if (line(13:14) .eq. ' B') then
        bndrad=0.83
      else if (line(13:14) .eq. 'BA') then
        bndrad=1.34
      else if (line(13:14) .eq. 'BE') then
        bndrad=0.35
      else if (line(13:14) .eq. 'BI') then
        bndrad=1.54
      else if (line(13:14) .eq. 'BR') then
        bndrad=1.21
```

```
else if (line(13:14) .eq. ' C') then
  bndrad=0.68
else if (line(13:14) .eq. 'CA') then
  bndrad=0.99
else if (line(13:14) .eq. 'CD') then
  bndrad=1.69
else if (line(13:14) .eq. 'CE') then
  bndrad=1.83
else if (line(13:14) .eq. 'CL') then
  bndrad=0.99
else if (line(13:14) .eq. 'CO') then
  bndrad=1.33
else if (line(13:14) .eq. 'CR') then
  bndrad=1.35
else if (line(13:14) .eq. 'CS') then
  bndrad=1.67
else if (line(13:14) .eq. 'CU') then
  bndrad=1.52
else if (line(13:14) .eq. ' D') then
  bndrad=0.23
else if (line(13:14) .eq. 'DY') then
  bndrad=1.75
else if (line(13:14) .eq. 'ER') then
  bndrad=1.73
else if (line(13:14) .eq. 'EU') then
  bndrad=1.99
else if (line(13:14) .eq. ' F') then
  bndrad=0.64
else if (line(13:14) .eq. 'FE') then
  bndrad=1.34
else if (line(13:14) .eq. 'GA') then
  bndrad=1.22
else if (line(13:14) .eq. 'GD') then
  bndrad=1.79
else if (line(13:14) .eq. 'GE') then
  bndrad=1.17
else if (line(13:14) .eq. ' H') then
  bndrad=0.23
else if (line(13:14) .eq. 'HF') then
  bndrad=1.57
else if (line(13:14) .eq. 'HG') then
  bndrad=1.70
else if (line(13:14) .eq. 'HO') then
  bndrad=1.74
else if (line(13:14) .eq. ' I') then
  bndrad=1.40
else if (line(13:14) .eq. 'IN') then
  bndrad=1.63
else if (line(13:14) .eq. 'IR') then
  bndrad=1.32
else if (line(13:14) .eq. ' K') then
  bndrad=1.33
else if (line(13:14) .eq. 'LA') then
  bndrad=1.87
else if (line(13:14) .eq. 'LI') then
  bndrad=0.68
else if (line(13:14) .eq. 'LU') then
  bndrad=1.72
else if (line(13:14) .eq. 'MG') then
  bndrad=1.10
else if (line(13:14) .eq. 'MN') then
  bndrad=1.35
else if (line(13:14) .eq. 'MO') then
  bndrad=1.47
else if (line(13:14) .eq. ' N') then
  bndrad=0.68
else if (line(13:14) .eq. 'NA') then
  bndrad=0.97
else if (line(13:14) .eq. 'NB') then
  bndrad=1.48
else if (line(13:14) .eq. 'ND') then
  bndrad=1.81
```

```
else if (line(13:14) .eq. 'NI') then
  bndrad=1.50
else if (line(13:14) .eq. 'NP') then
  bndrad=1.55
else if (line(13:14) .eq. ' O') then
  bndrad=0.68
else if (line(13:14) .eq. 'OS') then
  bndrad=1.37
else if (line(13:14) .eq. ' P') then
  bndrad=1.05
else if (line(13:14) .eq. 'PA') then
  bndrad=1.61
else if (line(13:14) .eq. 'PB') then
  bndrad=1.54
else if (line(13:14) .eq. 'PD') then
  bndrad=1.50
else if (line(13:14) .eq. 'PM') then
  bndrad=1.80
else if (line(13:14) .eq. 'PO') then
  bndrad=1.68
else if (line(13:14) .eq. 'PR') then
  bndrad=1.82
else if (line(13:14) .eq. 'PT') then
  bndrad=1.50
else if (line(13:14) .eq. 'PU') then
  bndrad=1.53
else if (line(13:14) .eq. 'RA') then
  bndrad=1.90
else if (line(13:14) .eq. 'RB') then
  bndrad=1.47
else if (line(13:14) .eq. 'RE') then
  bndrad=1.35
else if (line(13:14) .eq. 'RH') then
  bndrad=1.45
else if (line(13:14) .eq. 'RU') then
  bndrad=1.40
else if (line(13:14) .eq. ' S') then
  bndrad=1.02
else if (line(13:14) .eq. 'SB') then
  bndrad=1.46
else if (line(13:14) .eq. 'SC') then
  bndrad=1.44
else if (line(13:14) .eq. 'SE') then
  bndrad=1.22
else if (line(13:14) .eq. 'SI') then
  bndrad=1.20
else if (line(13:14) .eq. 'SM') then
  bndrad=1.80
else if (line(13:14) .eq. 'SN') then
  bndrad=1.46
else if (line(13:14) .eq. 'SR') then
  bndrad=1.12
else if (line(13:14) .eq. 'TA') then
  bndrad=1.43
else if (line(13:14) .eq. 'TB') then
  bndrad=1.76
else if (line(13:14) .eq. 'TC') then
  bndrad=1.35
else if (line(13:14) .eq. 'TE') then
  bndrad=1.47
else if (line(13:14) .eq. 'TH') then
  bndrad=1.79
else if (line(13:14) .eq. 'TI') then
  bndrad=1.47
else if (line(13:14) .eq. 'TL') then
  bndrad=1.55
else if (line(13:14) .eq. 'TM') then
  bndrad=1.72
else if (line(13:14) .eq. ' U') then
  bndrad=1.58
else if (line(13:14) .eq. ' V') then
  bndrad=1.33
```

```
      else if (line(13:14) .eq. ' W') then
        bndrad=1.37
      else if (line(13:14) .eq. ' Y') then
        bndrad=1.78
      else if (line(13:14) .eq. 'YB') then
        bndrad=1.94
      else if (line(13:14) .eq. 'ZN') then
        bndrad=1.45
      else if (line(13:14) .eq. 'ZR') then
        bndrad=1.56
      else if (line(14:14) .eq. 'C') then
        bndrad=0.68
      else if (line(14:14) .eq. 'D') then
        bndrad=0.23
      else if (line(14:14) .eq. 'H') then
        bndrad=0.23
      else if (line(14:14) .eq. 'N') then
        bndrad=0.68
      else if (line(14:14) .eq. 'O') then
        bndrad=0.68
      else if (line(14:14) .eq. 'P') then
        bndrad=1.05
      else if (line(14:14) .eq. 'S') then
        bndrad=1.02
      else
        bndrad=0.0
      endif
      return
      end
```

APPENDIX 5: CHEMGRID SOURCE CODE

```
#
# compilation command file for program CHEMGRID
#
#
f77 -O -u -c chemgrid.f
f77 -O -u -c dconst.f
f77 -O -u -c ddist.f
f77 -O -u -c dist.f
f77 -O -u -c grdout.f
f77 -O -u -c mkbox.f
f77 -O -u -c parmrec.f
f77 -O -u chemgrid.o mkbox.o dist.o ddist.o dconst.o grdout.o parmrec.o -o CHEMGRID
```

chemgrid.f

```
c                          PROGRAM CHEMGRID
c
c       Copyright (C) 1991 Regents of the University of California
c                          All Rights Reserved.
c
c  --calculates the grids used in DOCK 3.0 for force field type scoring,
c  given a receptor pdb file with hydrogens on the polar atoms, and the
c  appropriate parameter tables.  4 grids are created and stored in 3
c  files:  a bump grid/file, an electrostatic potential grid/file and
c  van der Waals repulsion and attraction grids stored in the same
c  file.
c  --informative output files:
c    BOX         pdb-format box allowing visualization of grid location
c                and size
c    OUTCHEM     restatement of input parameters; messages pertaining
c                to calculation of the grids
c    OUTPARM     messages pertaining to parameterization of receptor
c                atoms; net charge on the receptor molecule (including
c                any ions or waters in the receptor pdb file); created
c                in subroutine parmrec
c    PDBPARM     shows which parameters have been associated with each
c                atom in the receptor pdb file; created in subroutine
c                parmrec, and used in subroutines dconst (constant
c                dielectric) and ddist (distance-dependent dielectric)
c                                                        ECMeng   4/91
c-----------------------------------------------------------------------
      include 'chemgrid.h'
c
      character*80 vdwfil
      real scrd(3), sumcrd(3), com(3)
c  scrd()--coordinates of a sphere center
c  sumcrd()--sum of sphere center coordinates so far
c  com()--grid center coordinates; user input or sphere cluster center
c   of mass
      character*80 recfil, sphfil, boxfil, grdfil
c  recfil--pdb-format receptor file (input file)
c  sphfil--file containing one or more sphere clusters, read when the
c   user wishes to center the grids on a sphere cluster center of mass
c   (input file)
c  boxfil--pdb-format file for displaying the grid boundaries (output
c   file)
c  grdfil--prefix name for grid files (output)
      character*80 table, dumlin
c  table--table containing receptor atom parameters
      character*1 ctrtyp
c  ctrtyp--character flag for grid center option; 'u' or 'U' for user
```

```
c   input, otherwise a sphere cluster center of mass
      integer i, j, n
c  variables for reading sphere coordinates:
      integer cntemp, nstemp, clnum, nsph
      logical done
c  done--whether or not sphere cluster center of mass has been calculated
c
      open (unit=1, file='INCHEM', status='old')
      open (unit=2, file='OUTCHEM', status='new')
c
      read (1, 1000) recfil
 1000 format (A80)
      write (2, *) 'receptor pdb file:'
      write (2, 1000) recfil
      read (1, 1000) table
      write (2, *) 'receptor parameters will be read from:'
      write (2, 1000) table
      read (1, 1000) vdwfil
      write (2, *) 'van der Waals parameter file:'
      write (2, 1000) vdwfil
c
      call parmrec(recfil, table, vdwfil, 2)
c
      read (1, '(A1)') ctrtyp
      if (ctrtyp .eq. 'U' .or. ctrtyp .eq. 'u') then
        write (2, *) 'box center will be user-defined'
        read (1, *) (com(i), i=1,3)
      else
        read (1, 1000) sphfil
        write (2, 1001) 'box center will be sphere cluster ',
     &  'center of mass'
 1001   format (A34, A14)
        write (2, *) 'sphere file:'
        write (2, 1000) sphfil
        read (1, *) clnum
        write (2, *) 'cluster number:', clnum
c
c  --initialize coordinate sums for calculating center of mass
c
        do 10 i=1,3
          sumcrd(i)=0
   10   continue
        done=.false.
c
c  --open the sphere file; read cluster and calculate center of mass
c
        open (unit=3, file=sphfil, status='old')
   15   read (3, 1002, end=60) cntemp, nstemp
 1002   format (8x, I5, 32x, I5)
        if (cntemp .eq. clnum) then
          nsph=nstemp
          do 30 i=1,nsph
            read (3,1003) (scrd(j), j=1,3)
 1003       format (5x, 3F10.5)
            do 20 j=1,3
              sumcrd(j)=sumcrd(j) + scrd(j)
   20       continue
   30     continue
          do 40 i=1,3
            com(i)=sumcrd(i)/real(nsph)
   40     continue
          done=.true.
        else
          do 50 i=1,nstemp
            read (3, 1000) dumlin
   50     continue
          go to 15
        endif
   60   continue
        close (3)
        if (done) then
          write (2, *) 'done calculating sphere cluster center of mass'
```

```
        else
          write (2, *) 'error--sphere cluster not found'
          stop
        endif
      endif
      write (2, *) 'box center coordinates [x y z]:'
      write (2, *) (com(i), i=1,3)
      read (1, *) (boxdim(i), i=1,3)
      write (2, *) 'box x-dimension = ', boxdim(1)
      write (2, *) 'box y-dimension = ', boxdim(2)
      write (2, *) 'box z-dimension = ', boxdim(3)
c
c  --set offset to xmin, ymin, zmin of box
c
      do 65 i=1,3
        offset(i)=com(i) - boxdim(i)/2.0
   65 continue
c
      read (1, 1000) boxfil
      write (2, *) 'filename for pdb format box:'
      write (2, 1000) boxfil
c
      call mkbox(boxfil, 3, com, boxdim)
c
      read (1, *) grddiv
      write (2, *) 'grid spacing in angstroms'
      write (2, *) grddiv
      npts=1
c
c  --convert box dimensions to grid units, rounding upwards
c  --note that points per side .ne. side length in grid units,
c    because lowest indices are (1,1,1) and not (0,0,0)
c
      do 70 i=1,3
        grddim(i)=int(boxdim(i)/grddiv + 1.0)
        grdpts(i)=grddim(i) + 1
        npts=npts*grdpts(i)
   70 continue
      if (npts .gt. maxpts) then
        write (2, *) 'maximum number of grid points exceeded--'
        write (2, *) 'decrease box size, increase grid spacing, or'
        write (2, *) 'increase parameter maxpts'
        write (2, *) 'program stops'
        stop
      endif
      write (2, *) 'grid points per side [x y z]:'
      write (2, *) (grdpts(i), i=1,3)
      write (2, *) 'total number of grid points = ', npts
      read (1, *) estype
      if (estype .ne. 0) then
        estype=1
        write (2, *) 'a distance-dependent dielectric will be used'
      else
        write (2, *) 'a constant dielectric will be used'
      endif
      read (1, *) esfact
      write (2, '(A31, A15, I3)') 'the dielectric function will be',
     &' multiplied by ', esfact
   75 continue
      read (1, *) cutoff
      write (2, *) 'cutoff distance for energy calculations:'
      write (2, *) cutoff
      cutsq=cutoff*cutoff
c
c  --convert cutoff to grid units, rounding up (only add 1 rather
c    than 2, because differences in indices rather than the
c    absolute indices are required)
c
      grdcut=int(cutoff/grddiv + 1.0)
      read (1, *) pcon, ccon
      write (2, *) 'distances defining bumps with receptor atoms:'
      write (2, '(A21, F5.2)') 'receptor polar atoms ', pcon
```

```
      write (2, '(A22, F5.2)') 'receptor carbon atoms ', ccon
      pconsq=pcon*pcon
      cconsq=ccon*ccon
      read (1, 1000) grdfil
      write (2, *) 'output grid prefix name:'
      write (2, 1000) grdfil
      close (1)
c
c  --initialize grid
c
      do 80 n=1, maxpts
        aval(n)=0.0
        bval(n)=0.0
        esval(n)=0.0
        bump(n)='F'
   80 continue
c
      if (estype .eq. 0) then
        call dconst(3, grdcut, grddiv, grdpts, esfact, offset)
      else
        call ddist(3, grdcut, grddiv, grdpts, esfact, offset)
      endif
c
      call grdout(grdfil, 3, npts, grddiv, grdpts, offset)
c
      close (2)
      end
```

---

chemgrid.h

```
c      header file for CHEMGRID                                    ECMeng  4/91
c-------------------------------------------------------------------------------
      integer maxpts
      parameter (maxpts=1000000)
c  maxpts--maximum number of grid points
      integer npts
c  npts--number of grid points
      real aval(maxpts), bval(maxpts), esval(maxpts)
      character*1 bump(maxpts)
c  aval(), bval(), esval(), bump()--values stored "at" grid points
      real rsra, rsrb, rcrg, rcrd(3)
      integer natm, vdwn
c  rsra, rsrb, rcrg, rcrd(), natm, vdwn--values for current receptor atom
      integer nearpt(3)
c  nearpt()--3D indices of grid point closest to current receptor atom
      real gcrd(3)
c  gcrd()--coordinates in angstroms of current grid point
      real grddiv
c  grddiv--spacing of grid points in angstroms
      real boxdim(3)
c  boxdim()--box dimensions in angstroms (x,y,z)
      real offset(3)
c  offset()--box xmin, ymin, zmin in angstroms
      integer grddim(3)
c  grddim()--box dimensions in grid units (x,y,z)
      integer grdpts(3)
c  grdpts()--number of grid points along box dimensions (x,y,z)
c    NOTE: grdpts(i)=griddim(i) + 1   (lowest indices are (1,1,1))
      integer estype, esfact
c  estype--type of electrostatic calculation desired:
c    0 = use constant dielectric function
c    1 = use distance-dependent dielectric function
c    2 = use previously generated (DelPhi) electrostatic potential map
c  esfact--factor to multiply dielectric by when estype = 0 or estype = 1;
c    not read or used when estype = 2
c    examples:  D = 1    estype = 0, esfact = 1
c               D = 4    estype = 0, esfact = 4
c               D = r    estype = 1, esfact = 1
c               D = 4r   estype = 1, esfact = 4
```

```
      real cutoff, cutsq, pcon, ccon, pconsq, cconsq
c  cutoff--cutoff distance for energy calculations
c  cutsq--cutoff distance squared
c  pcon (ccon)--distance defining a bump with a polar atom (a carbon)
c    of the receptor
c  pconsq, cconsq--the squares of pcon and ccon, respectively
      integer grdcut
c  grdcut--cutoff, in grid units
      real dist2
c  dist2--function to calculate distance squared
      integer indx1
c  indx1--function to convert the 3-dimensional (virtual) indices of a
c    grid point to the actual index in a 1-dimensional array

c
      common
     &/rmaps/ aval, bval, esval
     &/cmap/ bump
     &/vals/ cutsq, pconsq, cconsq
```

---

dconst.f

```
c
c       Copyright (C) 1991 Regents of the University of California
c                          All Rights Reserved.
c
      subroutine dconst(unitno, grdcut, grddiv, grdpts, esfact, offset)
c
c  --called from CHEMGRID
c  --increments vdw and electrostatics values at grid points, using
c    a constant dielectric function                    ECMeng    4/91
c-----------------------------------------------------------------------
c
      include 'chemgrid.h'
c
      real mincon, minsq
      parameter (mincon=0.0001)
      integer unitno, i, j, k, n
      real r2, r6
c
      minsq=mincon*mincon
c
c  --open parameterized receptor file (from subroutine parmrec)
c
      open (unit=unitno, file='PDBPARM', status='old')
c
  100 read (unitno, 1006, end=500) natm, vdwn, rsra, rsrb, rcrg,
     &(rcrd(i), i=1,3)
 1006 format (2I5, 2(1x, F8.2), 1x, F8.3, 1x, 3F8.3)
      if (vdwn .le. 0) go to 100
c
c  --subtract offset from receptor atom coordinates, find the 3D indices
c    of the nearest grid point (adding 1 because the lowest indices
c    are (1,1,1) rather than (0,0,0)); ignore receptor atoms farther
c    from the grid than the cutoff distance
c
      do 110 i=1,3
        rcrd(i)=rcrd(i) - offset(i)
        nearpt(i)=nint(rcrd(i)/grddiv) + 1
        if (nearpt(i) .gt. (grdpts(i) + grdcut)) go to 100
        if (nearpt(i) .lt. (1 - grdcut)) go to 100
  110 continue
c
c --loop through grid points within the cutoff cube (not sphere) of
c   the current receptor atom, but only increment values if the grid
c   point is within the cutoff sphere for the atom
c
      do 400 i=max(1,(nearpt(1)-grdcut)),
     &min(grdpts(1),(nearpt(1)+grdcut))
```

```
        gcrd(1)=float(i-1)*grddiv
        do 300 j=max(1,(nearpt(2)-grdcut)),
     &  min(grdpts(2),(nearpt(2)+grdcut))
          gcrd(2)=float(j-1)*grddiv
          do 200 k=max(1,(nearpt(3)-grdcut)),
     &    min(grdpts(3),(nearpt(3)+grdcut))
            gcrd(3)=float(k-1)*grddiv
            n = indx1(i,j,k,grdpts)
            r2 = dist2(rcrd,gcrd)
            if (r2 .gt. cutsq) go to 120
            if (r2 .lt. minsq) then
              bump(n)='X'
              r2 = minsq
            else if(((r2 .lt. cconsq .and. vdwn .le. 5) .or. (r2 .lt.
     &      pconsq .and. vdwn .ge. 8)) .and. bump(n) .eq. 'F') then
              bump(n)='T'
            endif
            r6 = r2*r2*r2
            aval(n)=aval(n) + rsra/(r6*r6)
            bval(n)=bval(n) + rsrb/r6
            esval(n)=esval(n) + 332.0*rcrg/(esfact*sqrt(r2))
  120       continue
  200     continue
  300   continue
  400 continue
      go to 100
  500 continue
      close (unitno)
      return
      end
```

---

ddist.f

```
c
c       Copyright (C) 1991 Regents of the University of California
c                          All Rights Reserved.
c
      subroutine ddist(unitno, grdcut, grddiv, grdpts, esfact, offset)
c
c  --called from CHEMGRID
c  --increments vdw and electrostatics values at grid points, using
c    a distance-dependent dielectric function          ECMeng    4/91
c-----------------------------------------------------------------------
c
      include 'chemgrid.h'
c
      real mincon, minsq
      parameter (mincon=0.0001)
      integer unitno, i, j, k, n
      real r2, r6
c
      minsq=mincon*mincon
c
c  --open parameterized receptor file (from subroutine parmrec)
c
      open (unit=unitno, file='PDBPARM', status='old')
c
  100 read (unitno, 1006, end=500) natm, vdwn, rsra, rsrb, rcrg,
     &(rcrd(i), i=1,3)
 1006 format (2I5, 2(1x, F8.2), 1x, F8.3, 1x, 3F8.3)
      if (vdwn .le. 0) go to 100
c
c  --subtract offset from receptor atom coordinates, find the 3D indices
c    of the nearest grid point (adding 1 because the lowest indices
c    are (1,1,1) rather than (0,0,0)); ignore receptor atoms farther
c    from the grid than the cutoff distance
c
      do 110 i=1,3
        rcrd(i)=rcrd(i) - offset(i)
```

```
        nearpt(i)=nint(rcrd(i)/grddiv) + 1
        if (nearpt(i) .gt. (grdpts(i) + grdcut)) go to 100
        if (nearpt(i) .lt. (1 - grdcut)) go to 100
  110 continue
c
c --loop through grid points within the cutoff cube (not sphere) of
c   the current receptor atom, but only increment values if the grid
c   point is within the cutoff sphere for the atom
c
      do 400 i=max(1,(nearpt(1)-grdcut)),
     &min(grdpts(1),(nearpt(1)+grdcut))
        gcrd(1)=float(i-1)*grddiv
        do 300 j=max(1,(nearpt(2)-grdcut)),
     &  min(grdpts(2),(nearpt(2)+grdcut))
          gcrd(2)=float(j-1)*grddiv
          do 200 k=max(1,(nearpt(3)-grdcut)),
     &    min(grdpts(3),(nearpt(3)+grdcut))
            gcrd(3)=float(k-1)*grddiv
            n = indx1(i,j,k,grdpts)
            r2 = dist2(rcrd,gcrd)
            if (r2 .gt. cutsq) go to 120
            if (r2 .lt. minsq) then
              bump(n)='X'
              r2 = minsq
            else if(((r2 .lt. cconsq .and. vdwn .le. 5) .or. (r2 .lt.
     &      pconsq .and. vdwn .ge. 8)) .and. bump(n) .eq. 'F') then
              bump(n)='T'
            endif
            r6 = r2*r2*r2
            aval(n)=aval(n) + rsra/(r6*r6)
            bval(n)=bval(n) + rsrb/r6
            esval(n)=esval(n) + 332.0*rcrg/(esfact*r2)
  120       continue
  200     continue
  300   continue
  400 continue
      go to 100
  500 continue
      close (unitno)
      return
      end
```

---

dist.f

```
      real function dist2(c1, c2)
c
c  --calculates the square of the distance between two points
c
      real c1(3), c2(3)
      dist2 = (c1(1)-c2(1))**2 + (c1(2)-c2(2))**2 +
     &      (c1(3)-c2(3))**2
      return
      end
c-----------------------------------------------------------------------
      integer function indx1(i,j,k,grdpts)
c
c  --converts the 3-dimensional (virtual) indices of a grid point to the
c    actual index in a 1-dimensional array
c
      integer i, j, k
      integer grdpts(3)
c
      indx1 = grdpts(1)*grdpts(2)*(k-1) + grdpts(1)*(j-1) + i
      return
      end
```

---

grdout.f

```
      subroutine grdout(grdfil, unitno, npts, grddiv, grdpts, offset)
c
c  --called from CHEMGRID
c  --writes out grids; makes a formatted "bump" file and unformatted
c    van der Waals and electrostatics files
c                                                    EQMeng    4/91
c----------------------------------------------------------------------
      include 'chemgrid.h'
c
      character*80 grdfil
      integer i, namend, unitno
c
      namend=80
      do 100 i=2,80
       if (grdfil(i:i) .eq. ' ') then
         namend=i-1
         go to 105
        endif
  100 continue
  105 continue
c
    1 format (A17)
    2 format (4F8.3, 3I4)
    3 format (80A1)
      open (unit=unitno, file=grdfil(1:namend)//'.bmp', status='new')
      write (unitno, 1) 'bump map          '
      write (unitno, 2) grddiv, (offset(i), i=1,3), (grdpts(i), i=1,3)
      write (unitno, 3) (bump(i), i=1, npts)
      close (unitno)
      open (unit=unitno, file=grdfil(1:namend)//'.vdw', status='new',
     &form='unformatted')
      write (unitno) (aval(i), i=1, npts)
      write (unitno) (bval(i), i=1, npts)
      close (unitno)
      open (unit=unitno, file=grdfil(1:namend)//'.esp', status='new',
     &form='unformatted')
      write (unitno) (esval(i), i=1, npts)
      close (unitno)
c
      return
      end
```

mkbox.f

```
      subroutine mkbox(boxfil, unitno, com, boxdim)
c
c  --makes a PDB format box which shows the size and location of the grids
c                                                      EQMeng   4/91
c----------------------------------------------------------------------
      character*80 boxfil
      real boxdim(3), com(3)
      integer unitno, i
c
      open (unit=unitno, file=boxfil, status='new')
      write (unitno, '(A24)') 'HEADER    CORNERS OF BOX'
      write (unitno, 1) 'REMARK    CENTER (X Y Z) ', (com(i), i=1,3)
    1 format (A25, 3F8.3)
      write (unitno, 2) 'REMARK    DIMENSIONS (X Y Z) ',
     &(boxdim(i), i=1,3)
    2 format (A29, 3F8.3)
      write (unitno, 8) 'REMARK    Due to upwards rounding, the grids ',
     &'may be slightly larger'
```

```
    8 format (A45, A22)
      write (unitno, 9) 'REMARK     than this box.'
    9 format (A24)
      write (unitno, 3) 'ATOM', 1, 'DUA', 'BOX', 1,
     &(com(1) - boxdim(1)/2.0), (com(2) - boxdim(2)/2.0),
     &(com(3) - boxdim(3)/2.0)
      write (unitno, 3) 'ATOM', 2, 'DUB', 'BOX', 1,
     &(com(1) + boxdim(1)/2.0), (com(2) - boxdim(2)/2.0),
     &(com(3) - boxdim(3)/2.0)
      write (unitno, 3) 'ATOM', 3, 'DUC', 'BOX', 1,
     &(com(1) + boxdim(1)/2.0), (com(2) - boxdim(2)/2.0),
     &(com(3) + boxdim(3)/2.0)
      write (unitno, 3) 'ATOM', 4, 'DUD', 'BOX', 1,
     &(com(1) - boxdim(1)/2.0), (com(2) - boxdim(2)/2.0),
     &(com(3) + boxdim(3)/2.0)
      write (unitno, 3) 'ATOM', 5, 'DUE', 'BOX', 1,
     &(com(1) - boxdim(1)/2.0), (com(2) + boxdim(2)/2.0),
     &(com(3) - boxdim(3)/2.0)
      write (unitno, 3) 'ATOM', 6, 'DUF', 'BOX', 1,
     &(com(1) + boxdim(1)/2.0), (com(2) + boxdim(2)/2.0),
     &(com(3) - boxdim(3)/2.0)
      write (unitno, 3) 'ATOM', 7, 'DUG', 'BOX', 1,
     &(com(1) + boxdim(1)/2.0), (com(2) + boxdim(2)/2.0),
     &(com(3) + boxdim(3)/2.0)
      write (unitno, 3) 'ATOM', 8, 'DUH', 'BOX', 1,
     &(com(1) - boxdim(1)/2.0), (com(2) + boxdim(2)/2.0),
     &(com(3) + boxdim(3)/2.0)
    3 format (A4, 6x, I1, 2x, A3, 1x, A3, 5x, I1, 4x, 3F8.3)
      write (unitno, 4) 'CONECT', 1, 2, 4, 5
      write (unitno, 4) 'CONECT', 2, 1, 3, 6
      write (unitno, 4) 'CONECT', 3, 2, 4, 7
      write (unitno, 4) 'CONECT', 4, 1, 3, 8
      write (unitno, 4) 'CONECT', 5, 1, 6, 8
      write (unitno, 4) 'CONECT', 6, 2, 5, 7
      write (unitno, 4) 'CONECT', 7, 3, 6, 8
      write (unitno, 4) 'CONECT', 8, 4, 5, 7
    4 format (A6, 4I5)
      close (unitno)
      return
      end
```

parmrec.f

```
c
c       Copyright (C) 1991 Regents of the University of California
c                          All Rights Reserved.
c
      subroutine parmrec(recfil, table, vdwfil, unitno)
c
c     --called from CHEMGRID
c          Parmrec reads charges and VDW parameters for receptor
c     atom types from the appropriate files, indexes them via a hash
c     table, and then associates them with the atoms in a given
c     pdb-format receptor file.
c     Much of this code, namely the hashing and lookup routines,
c     has been adapted from the DelPhi code (program qdiffx and
c     subroutines) of Honig et al., version 3.0.
c
c     ECMeng      January 1991
c
c  recfil--name of receptor pdb file
c  table--name of the table to be referenced for receptor atom
c    parameters
c  vdwfil--name of file containing van der Waals parameters
c  unitno--logical unit number to write parameterization information
c    and warnings to
c-----------------------------------------------------------------------
      include 'parmrec.h'
c
```

```
      character*80 recfil, table, vdwfil
      integer unitno
      integer i, natm, n
c
      nptyp=0
      do 10 i=1,maxtyp
        inum(i)=0
        ilink(i)=0
   10 continue
c
c  --read receptor atom parameter file, index entries via a hash table
c
      open (unit=11, file=table, status='old')
c
  100 read (11, 1000, end=190) line
 1000 format (A80)
      if (line(1:1) .eq. '!') go to 100
      nptyp=nptyp + 1
      if (nptyp .gt. maxtyp) then
        write (6, *)
     &  'maximum number of atom types exceeded'
        write (6, *) 'increase parameter maxtyp'
        stop
      endif
      read (line, 1001) atm(nptyp), res(nptyp), resnum(nptyp),
     &chain(nptyp), crg(nptyp), vdwtyp(nptyp)
 1001 format (A4, 3x, A3, A4, A1, F8.3, 1x, I2)
c
      call enter(atm(nptyp), res(nptyp), resnum(nptyp),
     &chain(nptyp), nptyp)
c
      go to 100
  190 continue
      close (11)
c
c  --read vdw parameter file
c
      open (unit=11, file=vdwfil, status='old')
c
      nvtyp=0
  200 read (11, 1000, end=290) line
      if (line(1:1) .eq. '!') go to 200
      nvtyp=nvtyp + 1
      if (nvtyp .gt. maxtyv) then
        write (6, *) 'maximum number of vdw types exceeded'
        write (6, *) 'increase parameter maxtyv'
        stop
      endif
      read (line, 1002) sra(nvtyp), srb(nvtyp)
 1002 format (10x, F8.2, 5x, F8.2)
      go to 200
  290 continue
      close (11)
c
c  --read receptor pdb file, associate atoms with parameters, write
c    parameters and coordinates out to another file (PDBPARM)
c
      natm=0
      crgtot=0.0
c
      open (unit=11, file=recfil, status='old')
      open (unit=12, file='PDBPARM', status='new')
      open (unit=13, file='OUTPARM', status='new')
c
   20 read (11, '(A80)', end=990) line
      if (line(1:4) .ne. 'ATOM' .and. line(1:4) .ne. 'HETA') go to 20
      natm=natm + 1
      if (natm .gt. maxatm) then
        write (6, *) 'maximum number of receptor atoms exceeded'
        write (6, *) 'increase parameter maxatm'
        stop
      endif
```

```
      atom=line(13:16)
      resid=line(18:20)
      chn=line(22:22)
      resno=line(23:26)
c
      call find(atom, resid, resno, chn, found, n)
      if (.not. found) then
        schn=chn
        chn=' '
        call find(atom, resid, resno, chn, found, n)
        if (.not. found) then
          chn=schn
          sresno=resno
          resno='    '
          call find(atom, resid, resno, chn, found, n)
          if (.not. found) then
            schn=chn
            chn=' '
            call find(atom, resid, resno, chn, found, n)
            if (.not. found) then
              chn=schn
              resno=sresno
              sresid=resid
              resid='   '
              call find(atom, resid, resno, chn, found, n)
              if (.not. found) then
                schn=chn
                chn=' '
                call find(atom, resid, resno, chn, found, n)
                if (.not. found) then
                  chn=schn
                  sresno=resno
                  resno='    '
                  call find(atom, resid, resno, chn, found, n)
                  if (.not. found) then
                    schn=chn
                    chn=' '
                    call find(atom, resid, resno, chn, found, n)
                    if (.not. found) then
                      write (13, *) 'WARNING--parameters not found for'
                      write (13, *) line(1:27)
                      write (13, '(A18, A21)') 'sqrt(A), sqrt(B), ',
     &                 'and charge set to 0.0'
                      write (12, 2000) natm, 0, 0.0, 0.0, 0.0,
     &                 line(31:54)
                      go to 20
                    endif
                  endif
                endif
              endif
            endif
          endif
        endif
      endif
      write (12, 2000) natm, vdwtyp(n), sra(vdwtyp(n)), srb(vdwtyp(n)),
     &crg(n), line(31:54)
 2000 format (2I5, 2(1x, F8.2), 1x, F8.3, 1x, A24)
      crgtot=crgtot + crg(n)
      go to 20
  990 continue
      close (11)
      close (12)
      write (13, *) ' '
      write (13, '(A15, F8.3)') 'Total charge = ', crgtot
      close (13)
      return
      end
c
c-----------------------------------------------------------------------
c
      subroutine enter(atom, resid, resno, chn, nent)
c
```

```
      include 'parmrec.h'
c
c  --enter receptor atom type entries into hash table according to
c    entry number (sequential number of occurrence within the parameter
c    table)
c
      integer n, new, nent
c
      integer ihash
c
c
c  --get hash number using function ihash
c
      n=ihash(atom, resid, resno, chn)
      if (inum(n) .ne. 0) then
c
c  --slot filled; keep going along linked numbers until zero found
c
  100   continue
        if (ilink(n) .eq. 0) go to 200
        n=ilink(n)
        go to 100
  200   continue
c
c  --find an empty slot and fill it, leaving a trail in ilink()
c
        do 300 new=1,maxtyp
        if (inum(new) .eq. 0) go to 400
  300   continue
  400   continue
        ilink(n)=new
        n=new
      endif
      inum(n)=nent
      ilink(n)=0
      return
      end
c
c-----------------------------------------------------------------------
c
      integer function ihash(atxt,rtxt,ntxt,ctxt)
c
c  --produce a hash number for an atom, using atom name, residue name,
c    residue number, and chain indicator
c
      include 'parmrec.h'

      character*4 atxt
      character*3 rtxt
      character*4 ntxt
      character*1 ctxt
      character*38 string
      integer n, i, j
      data string /'* 0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ'/
      n = 1
      do 100 i = 1,3
        j = index(string,rtxt(i:i))
        n = 5*n + j
 100    continue
      do 101 i = 1,4
        j = index(string,atxt(i:i))
        n = 5*n + j
 101    continue
      do 102 i = 1,4
        j = index(string,ntxt(i:i))
        n = 5*n + j
 102    continue
      do 103 i = 1,1
        j = index(string,ctxt(i:i))
        n = 5*n + j
 103    continue
        n = iabs(n)
```

```
      ihash = mod(n,maxtyp) + 1
      return
      end
c
c-----------------------------------------------------------------------
c
      subroutine find(atom, resid, resno, chn, found, n)
c
c  --use the hash number of a receptor atom to find the appropriate
c    parameters, following links when necessary; check explicitly for
c    a match
c
      include 'parmrec.h'
c
      integer n
      integer ihash
c
      n=ihash(atom, resid, resno, chn)
      found=.false.
  100 continue
      if (inum(n) .eq. 0) then
        found=.false.
        return
      endif
      if ((resid .eq. res(inum(n))) .and. (atom .eq. atm(inum(n)))
     &.and. (resno .eq. resnum(inum(n))) .and. (chn .eq.
     &chain(inum(n)))) then
        n=inum(n)
        found=.true.
        return
      else
        if (ilink(n) .ne. 0) then
          n=ilink(n)
        else
          found=.false.
          return
        endif
      endif
      go to 100
      end
```

parmrec.h

```
c     header file for subroutine parmrec          ECMeng   4/91
c-----------------------------------------------------------------------
      integer maxtyp, nptyp
      parameter (maxtyp=1000)
c  maxtyp--maximum number of entries in 'prot.table' or 'na.table'
c  nptyp--number of entries in 'prot.table' or 'na.table' so far
      integer inum(maxtyp), ilink(maxtyp)
c  inum()--id numbers in hash table
c  ilink()--links for hash table
      character*1 chain(maxtyp), chn, schn
      character*3 res(maxtyp), resid, sresid
      character*4 atm(maxtyp), resnum(maxtyp), atom, resno, sresno
      real crg(maxtyp)
      integer vdwtyp(maxtyp)
c  vdwtyp()--integer vdw type indicators
      integer maxtyv
      parameter (maxtyv=50)
c  maxtyv--maximum number of entries in 'vdw.parms'
      integer nvtyp
c  nvtyp--number of entries in 'vdw.parms' so far
      real sra(maxtyv), srb(maxtyv)
c  sra(), srb()--vdw parameters, sqrt of A and sqrt of B
      integer maxatm
      parameter (maxatm=10000)
c  maxatm--maximum number of receptor atoms
      logical found
```

```
      character*80 line
      real crgtot
c
      common
     &/link/ inum, ilink
     &/name/ atm, res, resnum, chain
     &/value/ crg, vdwtyp, sra, srb
```

APPENDIX 6: HALOPERIDOL AND HIV-1 PROTEASE: HYPOTHETICAL BINDING MODES

BACKGROUND

One of the largest collaborations involving the Kuntz group is the quest for nonpeptidic inhibitors of the human immunodeficiency virus 1 (HIV-1) protease. This enzyme plays a crucial role in viral maturation[1] and thus infectivity,[2] and nonpeptidic inhibitors are desired to circumvent the bioavailability problems characteristic of peptides.[3] Renée DesJarlais performed a DOCK 1.1 search[4,5] using the structure of the unliganded HIV-1 protease,[6] 3hvp in the Brookhaven Protein Data Bank[7,8] (PDB). The DOCK database, which was evaluated according to shape complementarity only, consisted of approximately 10,000 molecules from the Cambridge Structural Database[9] (CSD). The 200 top-scoring molecules were examined in their docked orientations using the graphics package MidasPlus.[10] One of the compounds chosen for testing based on the search, haloperidol (Figure 1), was subsequently shown to inhibit the HIV-1 protease with a K_i of approximately 100 μM.[5] By this time, a structure of the HIV-1 protease complexed with a peptide-based inhibitor had become available (4hvp in the PDB).[11] Along with George Seibel and Randall Radmer in Peter Kollman's group, I set out to model how haloperidol might be binding to the protease.

METHODS

Different low-energy conformers of haloperidol and the closely related compound bromperidol were investigated. The CSD reference code of the bromperidol structure in the original docking[5] is BIBSEK; bibliographic searching yielded structures for

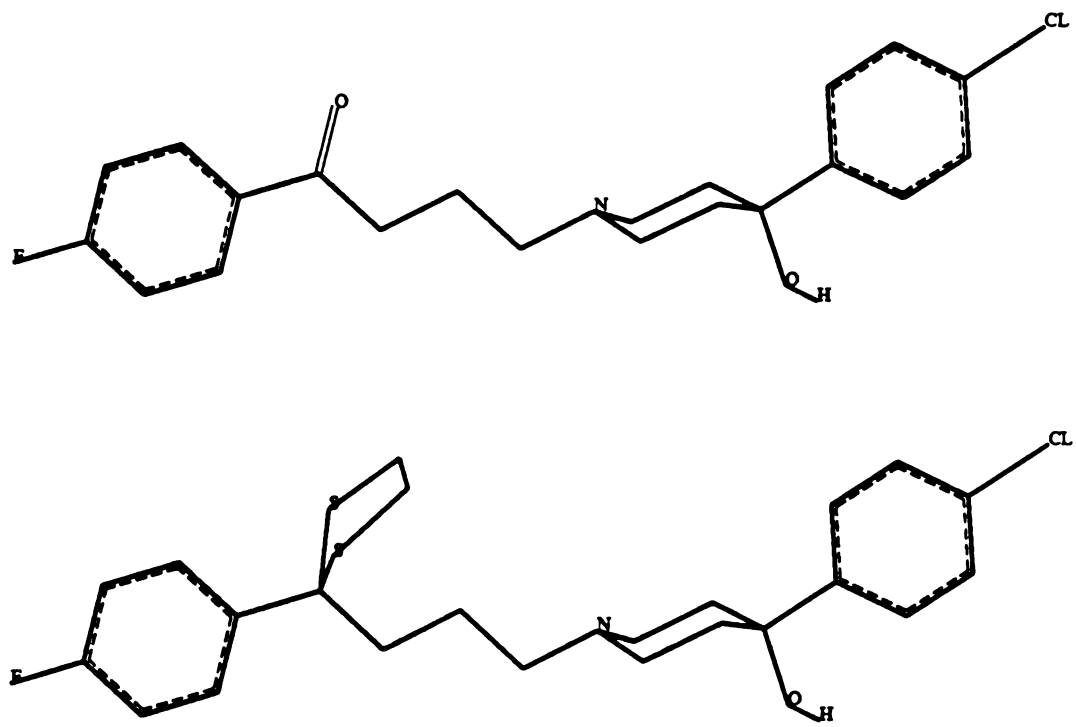
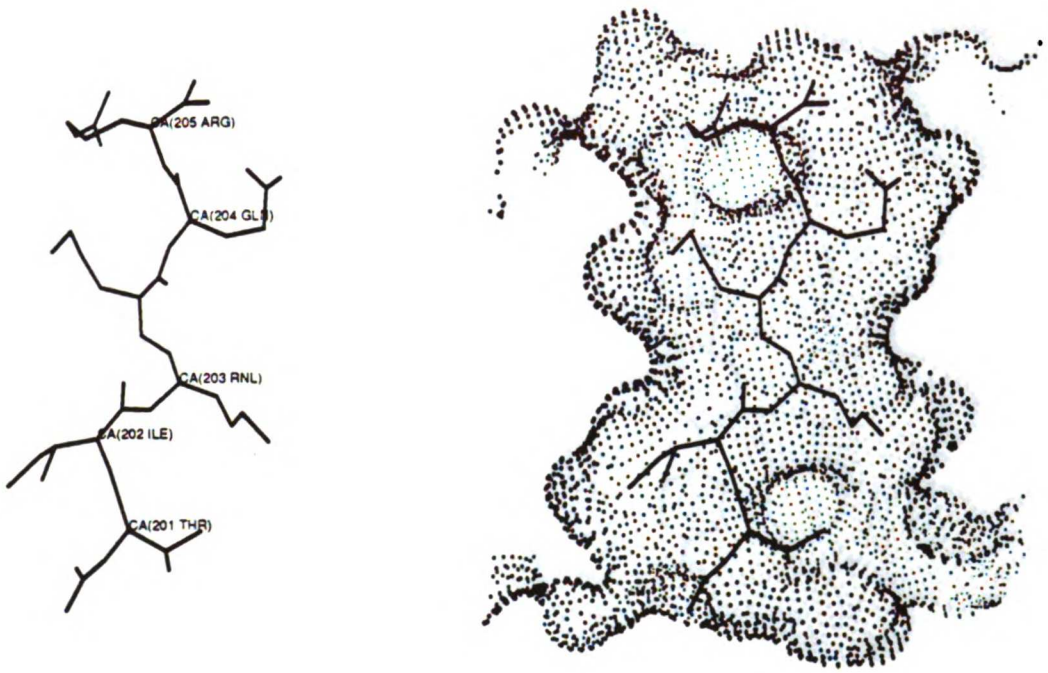


Figure 1. Haloperidol (top) and thioketal derivative UCSF8 (bottom).

haloperidol and haloperidol hydrobromide, HALDOL and HALOPB, respectively. Together, these structures represented just two out of the three major families of conformations available to butyrophenones, which include haloperidol and similar compounds;[12] therefore, a representative of the third class was generated from BIBSEK by manual bond rotations within MidasPlus.[10]

With the goal of determining the most likely mode(s) of binding to the HIV-1 protease, I docked the four structures and selected orientations for further study with molecular mechanics.

DOCK 1.1. The algorithm of Kuntz *et al.*[13] as implemented in DOCK 1.1[4] was used to position each of the four butyrophenone structures relative to the complexed conformation of the HIV-1 protease[11] (4hvp). For ease of comparison, 4hvp was moved into the same frame of reference as the uncomplexed protease structure. After removal of the crystallographic waters and the peptide-based inhibitor MVT-101 (Figure 2), a cluster of spheres was generated in the active site. For DOCK to produce an orientation, the internal distances among eight ligand atoms were required to match the internal distances among eight spheres, within a tolerance of 2.0 angstroms. Contact-scoring parameters were as described previously[4] and no bad contacts were allowed. Hundreds of orientations per ligand structure were obtained, and several criteria were applied in order to select a reasonable number of orientations for further study. Those chosen had either a high contact score, or a moderate contact score and one or more polar atoms near the active site aspartyl groups. In addition, each had to be different from the others. With these guidelines, 20 orientations including the original BIBSEK docking were selected for energy minimization in the active site of the protease. I grouped the 20 orientations into six families (Figure 3). The contact scores are listed in Table I.



MVT-101 UCSF MidasPlus MVT-101 in 4hvp site

Figure 2. The peptide-based inhibitor MVT-101, alone (left) and with the molecular surface of the HIV-1 protease active site (right). The inhibitor and protease coordinates are from 4hvp.[11]

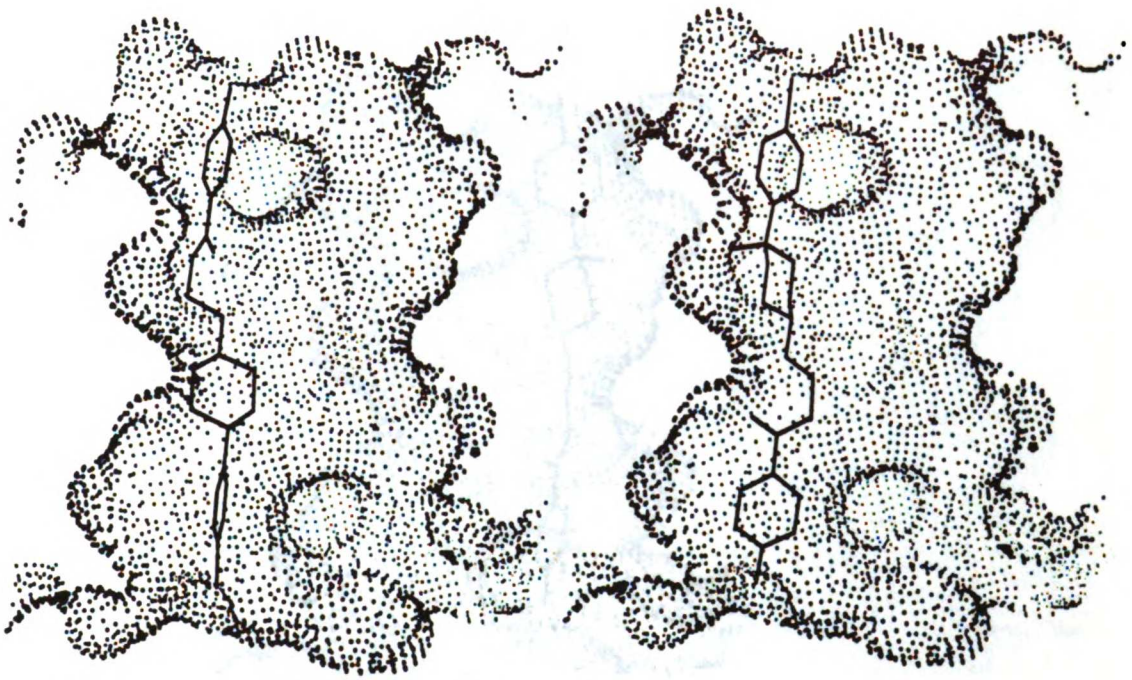
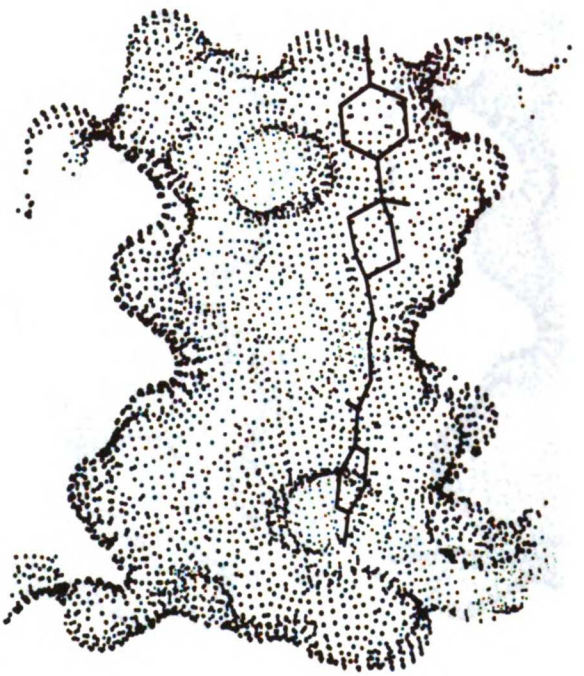


Figure 3A. The "axis" orientations.



296 UCSF MidasPlus

Figure 3A. (continued)

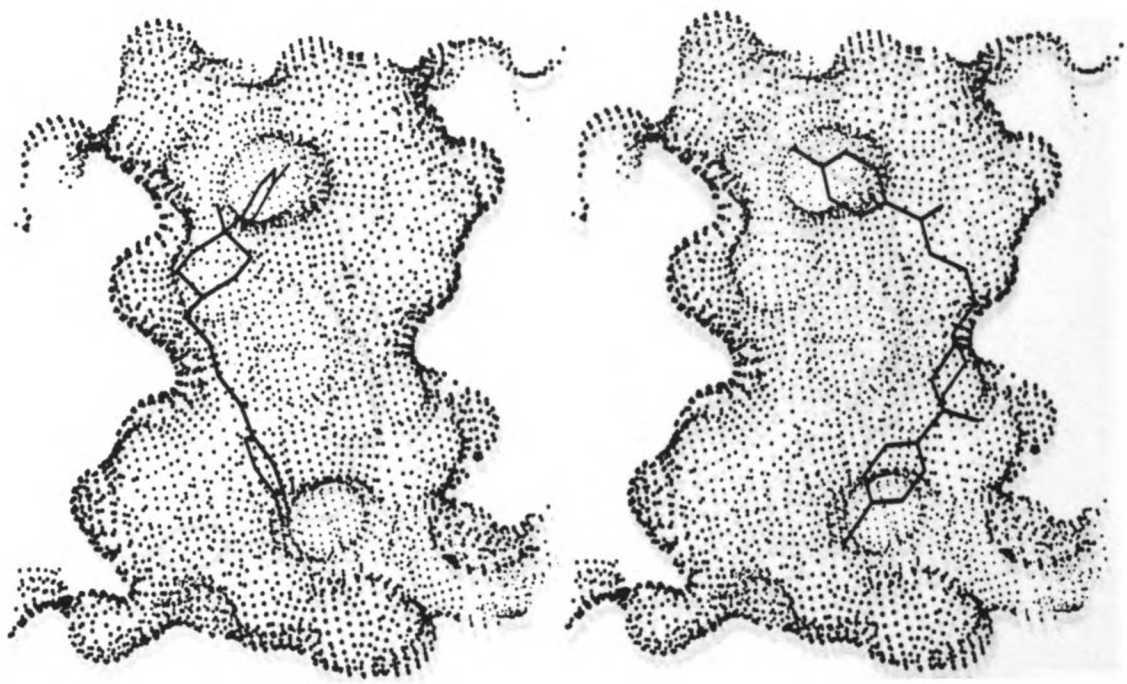


Figure 3B. The "bent" orientations.

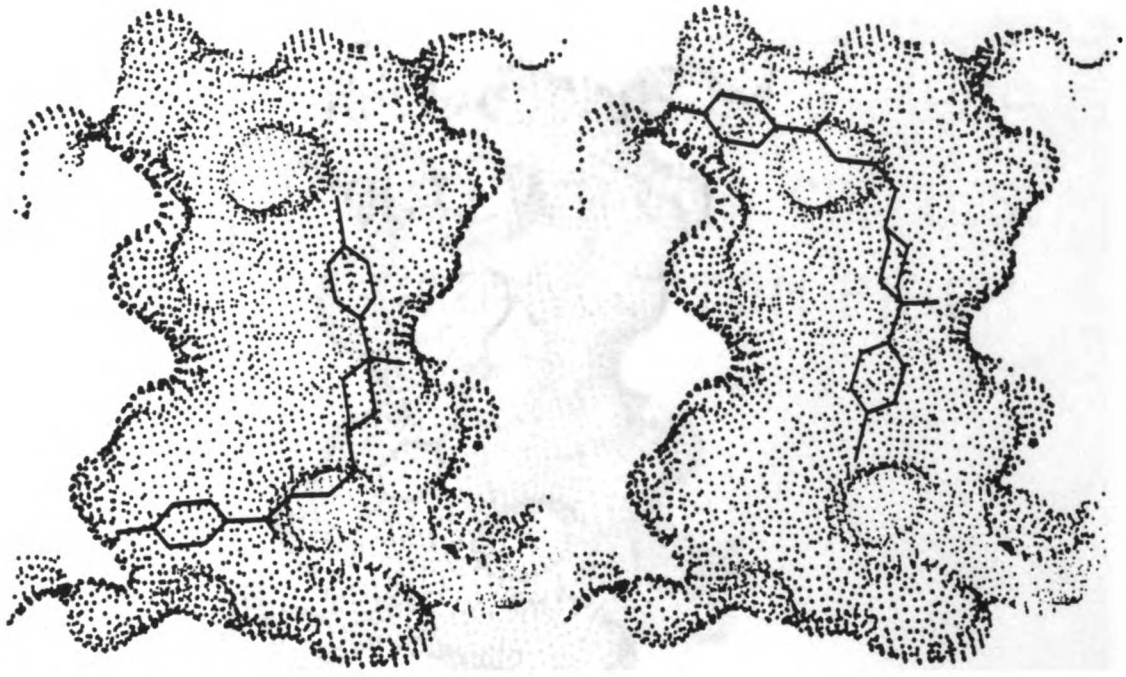
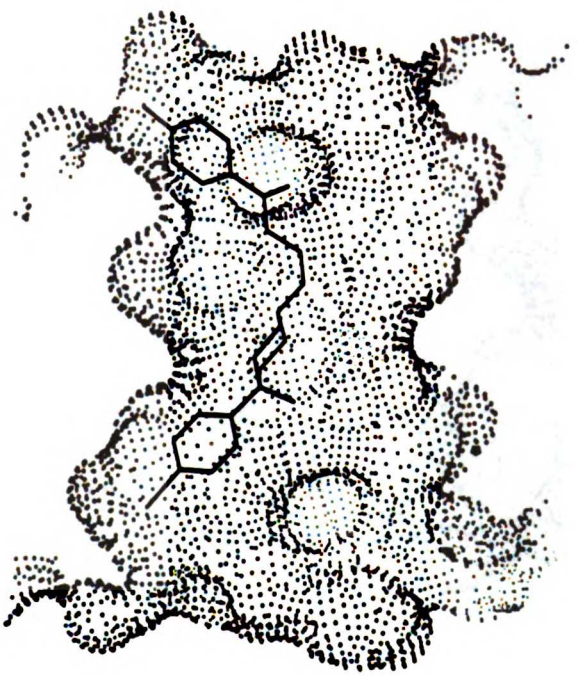


Figure 3B. (continued)



204 UCSF MidasPlus

Figure 3B. (continued)

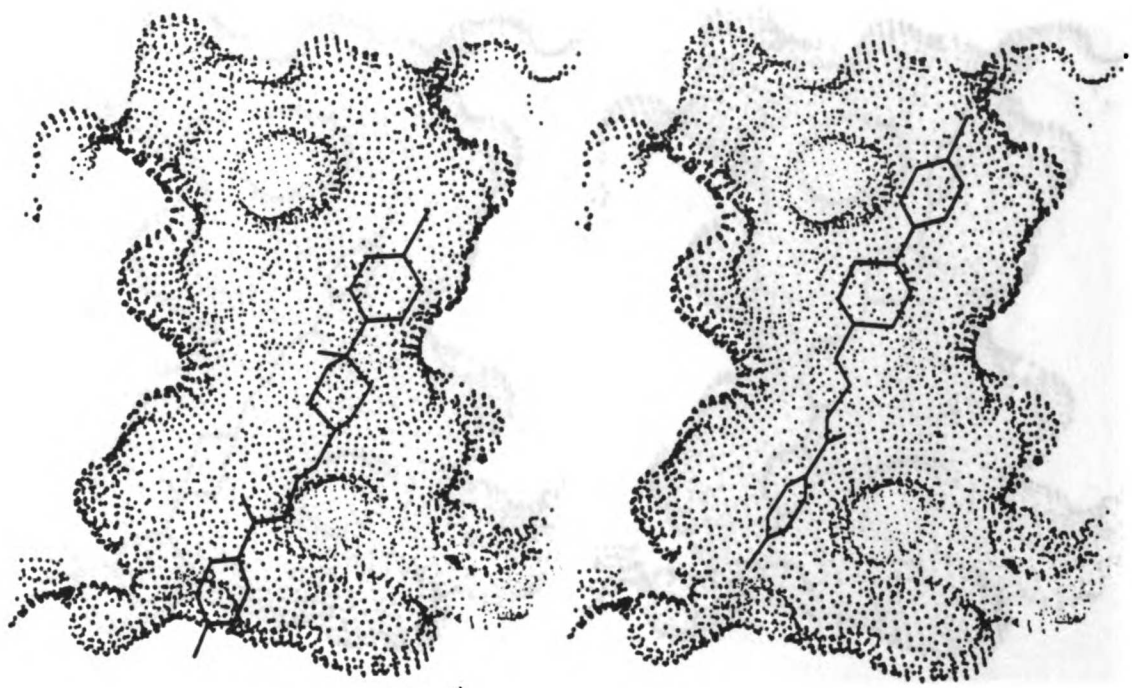


Figure 3C. The "cross" orientations.

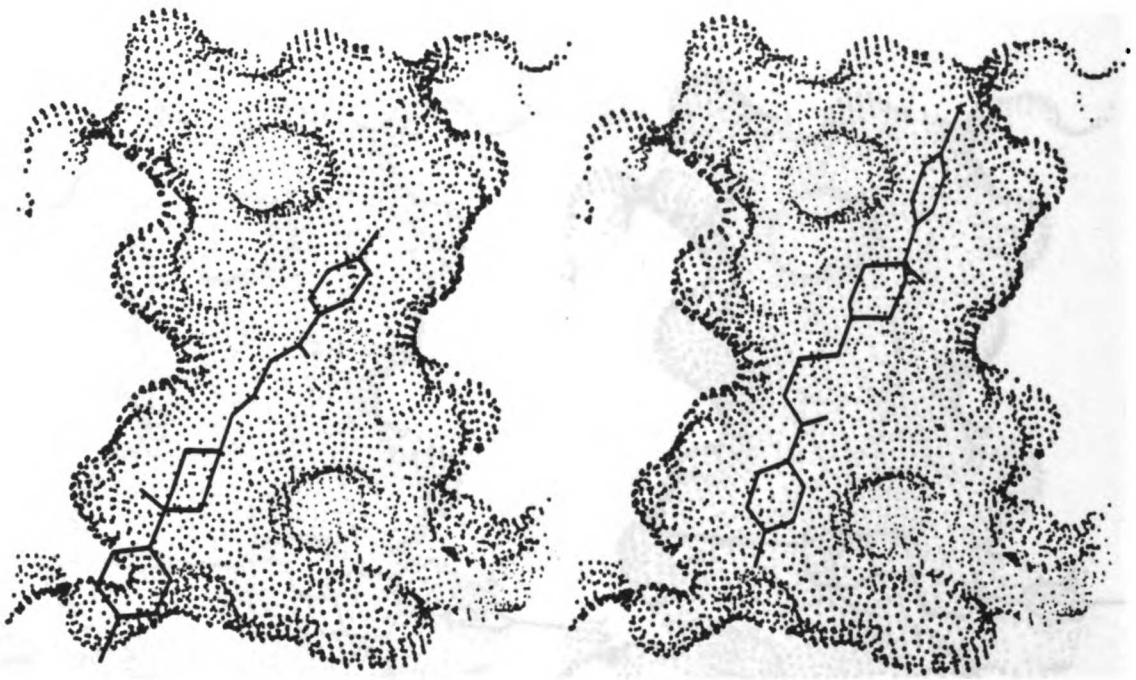


Figure 3C. (continued)

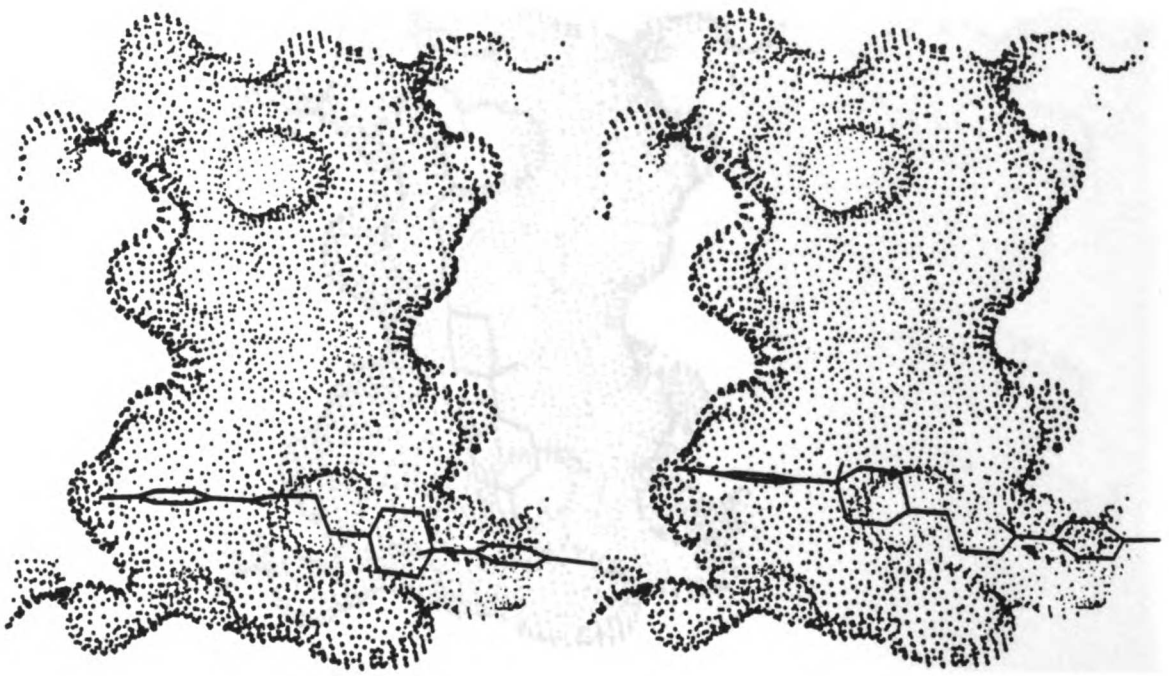
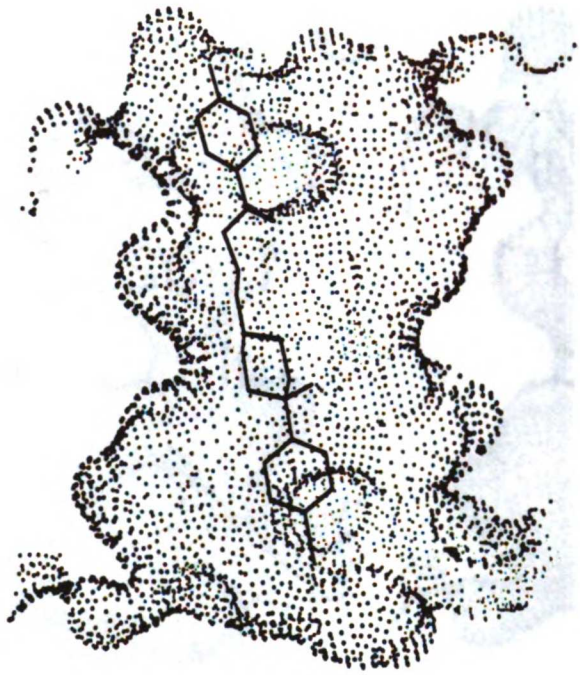


Figure 3D. The "entry" orientations.



85 UCSF MidasPlus

Figure 3E. The "flip" orientation.

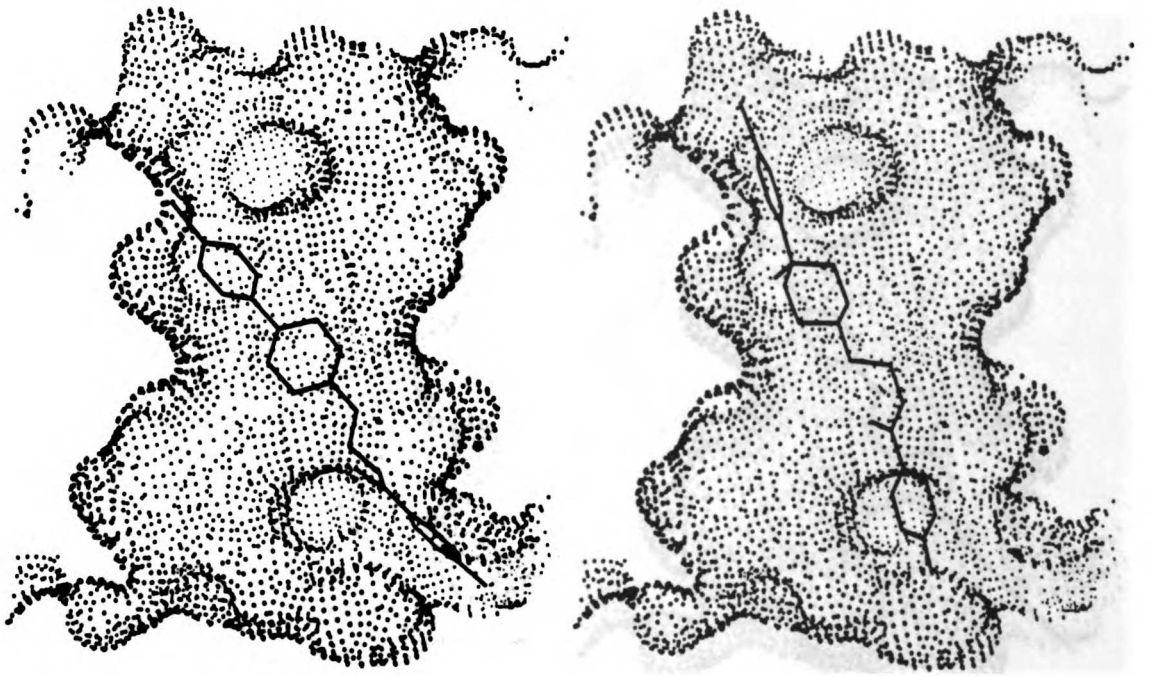


Figure 3F. The "orig" orientations.

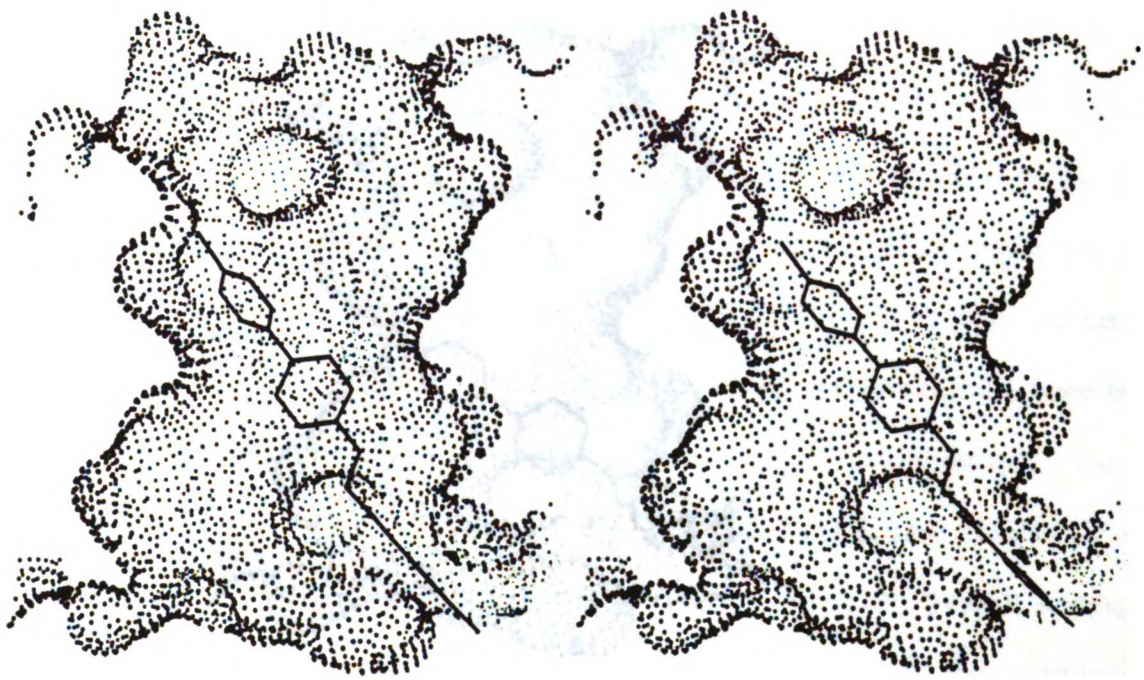
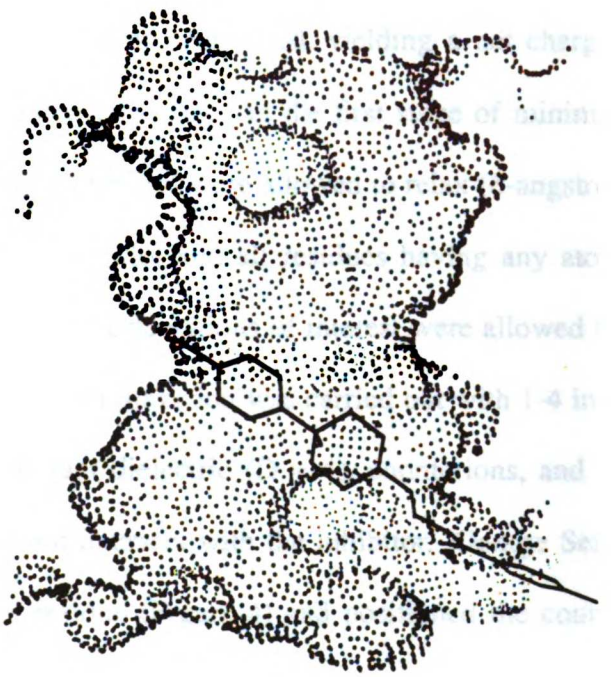


Figure 3F. (continued)



588 UCSF MidasPlus

Figure 3F. (continued)

AMBER. Minimizations utilized the 4hvp protease structure and were carried out in two stages with the AMBER united-atom force field.[14,15] Prior to minimization, bromperidol was converted to haloperidol by changing the bromine atom into a chlorine atom. The piperidine nitrogen was protonated, yielding a net charge of +1, and all inhibitor hydrogens were treated explicitly. In the first stage of minimization, only halogen and hydrogen atoms on haloperidol were allowed to relax (6-angstrom nonbonded cutoff). In the second stage, haloperidol and all residues having any atom within 6 angstroms of haloperidol in any of the orientations of interest were allowed to move (8-angstrom nonbonded cutoff). Each minimization was carried out with 1-4 interactions scaled down by a factor of 2, a constant dielectric (D = 4), counterions, and all crystallographic water molecules that did not intersect with the inhibitor. George Seibel created the PREP file for the haloperidol residue (Figure 4) and positioned the counterions. Randall Radmer and I ran the minimizations. Analyses of the different contributions to the interaction energies (Table I) were performed with the AMBER module ANAL.

DelPhi. The 20 orientations were also examined for electrostatic complementarity using potential maps calculated with DelPhi[16,17] (Table II). DelPhi uses a finite-difference algorithm to solve the Poisson-Boltzmann equation for a two-dielectric system with point charges embedded in the region of low dielectric. Three-step focusing,[17] in which the protein occupied 20, 60, and then 90% of the electrostatic potential grid, was used in order to reduce any errors associated with boundary conditions. Internal and external dielectric constants were 4 and 80, respectively, the ionic strength was 0.145 M, the ion exclusion radius was 2.0 angstroms, and the probe radius was 1.4 angstroms. DelPhi runs included the entire protease from 4hvp with AMBER united-atom partial charges,[14] except that one of the active-site aspartic acid residues was protonated. A

```
    0    0    2

    Haloperidol with MNDO 6G Connolly charges
coh.res
COH    INT     0
CHANGE   OMIT DU   BEG
  0.0
   1    DUMM  DU    M       0.000       0.000       0.000       0.000
   2    DUMM  DU    M       1.000       0.000       0.000       0.000
   3    DUMM  DU    M       1.000       1.000       0.000       0.000
   4    N1    N3    M       2.747       9.749       1.034       0.0633
   6    C2    CT    3       3.030       9.875      -0.391      -0.1170
   9    C3    CT    3       1.898      10.570      -1.127      -0.2507
  12    C4    CT    3       1.574      11.964      -0.538       0.2821
  26    C5    CT    3       1.351      11.798       0.959      -0.2507
  29    C6    CT    B       2.543      11.104       1.593      -0.1170
  32    C7    CT    M       3.875       9.132       1.752      -0.3629
  35    C8    CT    M       4.256       7.737       1.250      -0.0645
  38    C9    CT    M       5.239       7.019       2.155      -0.1335
  41    C10   C     M       6.634       7.619       2.169       0.5016
  42    O11   O     E       6.892       8.632       1.529      -0.4008
  43    C12   C     S       7.694       6.953       2.961      -0.3274
  44    C13   CA    B       8.970       7.524       2.977       0.0628
  46    C14   CA    B       9.981       6.971       3.730      -0.3118
  48    C15   C     B       9.746       5.845       4.427       0.4803
  50    C16   CA    B       8.527       5.229       4.435      -0.3119
  52    C17   CA    S       7.516       5.790       3.669       0.0628
  49    F18   F     E      10.730       5.344       5.213      -0.2479
  13    O19   OH    S       2.701      12.766      -0.784      -0.4502
  15    C20   CA    S       0.355      12.549      -1.280       0.0608
  16    C21   CA    B      -0.947      12.317      -0.853      -0.2847
  18    C22   CA    B      -2.036      12.905      -1.488       0.2448
  20    C23   C     B      -1.827      13.688      -2.573      -0.4929
  22    C24   CA    B      -0.543      13.910      -3.023       0.2447
  24    C25   CA    S       0.516      13.344      -2.372      -0.2847
  21    CL6   CL    E      -3.275      14.563      -3.417       0.1456
   7    H27   HC    E       3.835      10.277      -0.482       0.1682
   8    H28   HC    E       3.167       8.998      -0.838       0.1194
  10    H29   HC    E       2.035      10.626      -2.052       0.1364
  11    H30   HC    E       1.152       9.918      -1.061       0.0979
  27    H31   HC    E       0.540      11.221       1.132       0.0979
  28    H32   HC    E       1.309      12.861       1.478       0.1364
  30    H33   HC    E       3.500      11.729       1.551       0.1681
  31    H34   HC    E       2.410      11.035       2.599       0.1195
  33    H35   HC    E       4.667       9.765       1.687       0.1889
  34    H36   HC    E       3.624       9.105       2.671       0.1889
  36    H37   HC    E       4.625       7.848       0.300       0.0658
  37    H38   HC    E       3.494       7.230       1.199       0.0658
  39    H39   HC    E       5.334       6.068       2.002       0.0754
  40    H40   HC    E       4.835       6.904       3.069       0.0754
  45    H41   HC    E       9.024       8.251       2.405       0.0993
  47    H42   HC    E      10.724       7.368       3.713       0.1767
  51    H43   HC    E       8.436       4.454       4.962       0.1767
  53    H44   HC    E       6.653       5.451       3.670       0.0993
  14    H45   HO    E       2.666      13.373      -0.528       0.3031
  17    H46   HC    E      -1.154      11.841      -0.028       0.1335
  19    H47   HC    E      -2.919      12.646      -1.218       0.0569
  23    H48   HC    E      -4.480      14.557      -3.693       0.0569
  25    H49   HC    E       1.368      13.686      -2.676       0.1335
   5    H50   H3    E       2.334       9.468       1.084       0.3199


LOOP
 N1   C6
 C12  C17
 C20  C25

DONE
STOP
```

Figure 4. The AMBER PREP file for the haloperidol residue.

Table I. Haloperidol orientations: DOCK scores[a] and AMBER interaction energies.[b]

Orientation		DOCK	AMBER total	electrostatic	6-12	10-12
axis:	93	120	−67.4	−31.9	−34.7	−0.852
	96	110	−60.7	−24.3	−36.5	−0.015
	296	127	−63.1	−27.3	−35.1	−0.669
bent:	47	191	−63.5	−25.7	−37.8	−0.028
	100	111	−55.8	−26.2	−29.5	−0.084
	116	139	−61.7	−24.0	−36.8	−0.865
	173	109	−54.7	−24.1	−30.0	−0.496
	204	119	−65.7	−31.4	−34.3	−0.034
cross:	363	131	−61.3	−23.7	−36.7	−0.813
	452	150	−69.8	−30.0	−39.8	−0.017
	517	151	−62.3	−24.4	−37.0	−0.920
	590	152	−65.5	−26.6	−38.9	−0.033
entry:	16	170	−56.6	−20.6	−35.4	−0.526
	131	172	−57.5	−17.8	−39.7	−0.046
flip:	85	86	−56.3	−23.9	−32.2	−0.152
orig:	51	119	−57.8	−22.6	−35.1	−0.116
	102	92	−61.8	−29.9	−31.9	−0.026
	125	128	−56.6	−22.4	−33.6	−0.539
	130	134	−56.1	−23.2	−32.4	−0.477
	588	120	−53.5	−23.1	−30.3	−0.092

[a]DOCK 1.1 contact scores before minimization, in the context of the 4hvp active site. Hydrogens do not contribute to the score.

[a]Kcal/mol after minimization.

Table II. Haloperidol orientations: DelPhi interaction energies.[a,b]

Orientation		high-dielectric site	low-dielectric site
axis:	93	−2.862	−12.550
	96	−3.325	−15.880
	296	−5.609	−22.632
bent:	47	−2.893	−12.243
	100	−6.004	−32.495
	116	−6.785	−12.022
	173	−3.777	−22.571
	204	−3.400	−23.032
cross:	363	−7.032	−10.227
	452	−2.650	−24.050
	517	−1.868	−9.781
	590	−1.103	−4.141
entry:	16	−0.276	−1.492
	131	−1.543	−6.595
flip:	85	−2.137	−3.376
orig:	51	−3.093	−12.668
	102	−3.854	−22.368
	125	−5.600	−17.363
	130	−7.241	−19.459
	588	−3.525	−17.233

[a]Before minimimization, in units of RT = 0.5924 kcal/mol at 298K.

[b]See text for run parameters and descriptions of the two site models.

hydrogen was placed on the oxygen calculated to have the most negative potential when neither residue was protonated. Two models of the system were used for the DelPhi runs. In one, the site was considered wholly high-dielectric (filled with "continuum solvent"); in the other, the atoms of MVT-101 were treated as chargeless regions of low dielectric. The potential at each atom of a ligand orientation was obtained by trilinear interpolation of the values at the eight surrounding grid points and multiplied by the atomic charge to give the interaction energy. The total electrostatic interaction energy for an orientation was obtained by summing over its atoms.

DOCK 3.0. A second phase of docking was initiated after a structure of the HIV-1 protease complexed with a thioketal derivative of haloperidol, UCSF8 (Figure 1), was determined by members of the collaboration.[18] This structure was markedly different from the original BIBSEK docking. As the receptor structure was seen to resemble the unliganded protease conformation rather than the flaps-down conformation typical of complexes with peptide-based inhibitors (such as 4hvp), the uncomplexed protease[6] was used for docking. To determine the information necessary to reproduce the experimental binding mode, I utilized DOCK 3.0, in which one of the scoring options is an approximate AMBER interaction energy or "force field" score.[19] Force field grids were calculated in CHEMGRID, using the entire unliganded protease with AMBER united-atom partial charges,[14] 0.3-angstrom spacing, a 10.0-angstrom cutoff, and a dielectric function of $4r$, where r is the distance in angstroms between interacting atoms. Trilinear interpolation of the grid values was used for scoring. Four atoms were required to match four sphere centers to generate an orientation, with an internal distance tolerance of 1.5 angstroms. Bin widths and overlaps were 1.0 and 0.2 angstroms, respectively. Gasteiger-Marsili charges[20-22] were used for the ligand.

RESULTS AND DISCUSSION

The rankings of the 20 orientations according to different measures are given in Table III.

DOCK 1.1. In Figure 3, the orientations of interest are shown in the same frame of reference as the inhibitor MVT-101 in Figure 2. Nearly all positions imaginable within the site volume were observed in the DOCK output. Notably, the second and third highest contact scores from DOCK (Table I and Table III) correspond to haloperidol spanning the mouth of the active site tunnel ("entry" family), rather than binding inside. While the original docking ("orig" family) forms an angle with the long axis of MVT-101, other orientations are nearly parallel to this structure ("axis" family). Members of the "cross" family also form an angle with MVT-101 and roughly span the region between the S2 and S2′ subsites. The "bent" orientations resulted from docking the least extended of the four butyrophenone conformations. The "flip" structure is very similar to "axis" orientations and probably should have been classified with them. Although the name refers to which end of the site the fluorophenyl group is nearest, members of the "axis," "bent," and "cross" families are also flipped according to this definition. All conformations used for docking contain the chair form of the 4-hydroxypiperidyl group, so that the NH and OH groups point in opposite directions. The 20 selected orientations include "OH-up" (toward the flaps), "OH-down" (toward the active site aspartic acid residues) and "OH-sideways" structures.

AMBER. An important issue is how to interpret the results of energy-minimizing different orientations or ligands within a receptor site. Taking the energy of interaction as the most relevant value (Table I), there does not appear to be a strong differentiation among families. In general, the "entry" and "flip" positions are found to be

Table III. Haloperidol orientations: rankings according to different measures.

ranking	DOCK[a]	AMBER[b]	ES[c]	6-12[d]	DelPhi-high[e]	DelPhi-low[f]
1	47	452	93	452	130	100
2	131	93	204	131	363	452
3	16	204	452	590	116	204
4	590	590	102	47	100	296
5	517	47	296	517	296	173
6	452	296	590	116	125	102
7	116	517	100	363	102	130
8	130	102	47	96	173	125
9	363	116	517	16	588	588
10	125	363	96	296	204	96
11	296	96	173	51	96	51
12	93	51	116	93	51	93
13	588[g]	131	85	204	47	47
14	204	16	363	125	93	116
15	51[g]	125[g]	130	130	452	363
16	100	85	588	85	85	517
17	96	130	51	102	517	131
18	173	100	125	588	131	590
19	102	173	16	173	590	85
20	85	588	131	100	16	16

[a]Ranking by DOCK 1.1 contact scores.

[b]Ranking by AMBER total interaction energies.

[c]Ranking by AMBER electrostatic interaction energies.

[d]Ranking by AMBER van der Waals interaction energies.

[e]Ranking by DelPhi electrostatic interaction energies, high-dielectric site model.

[f]Ranking by DelPhi electrostatic interaction energies, low-dielectric site model.

[g]Tied with the preceding entry.

electrostatically unfavorable relative to the other families, which each have at least one member with an electrostatic interaction energy close to -30.0 kcal/mol or lower. As a group, the "cross" orientations have the most favorable van der Waals interaction energies; 452 has the best total interaction energy, although other orientations' energies differ by only a few kcal/mol. The total interaction energies range from -53.5 to -69.8 kcal/mol (Table I).

Orientation 452 places phenyl rings in S2 and S2′; thus, it is in keeping with the qualitative symmetry of the site. Unlike the hydroxyl group in the original docking (orientation 51), which is "down" between the active site aspartyl groups, the hydroxyl group in this orientation points "up" and is suitably placed to H-bond to a carbonyl oxygen in one of the flaps (the carbonyl oxygen of Gly-148). The proton on the positively charged piperidine nitrogen is pointing down at the catalytic residues. While this is reasonable in terms of electrostatic interactions, the importance of the hydroxyl moiety in certain of the peptide-based inhibitors has been stressed; this group H-bonds to the catalytic aspartic acid residues.[23,24] Since the best complexation energy calculated corresponds to an OH-up, NH-down mode, there was some concern about whether our model of the protease active site could be causing an overestimation of the favorable interactions between the protonated nitrogen and the aspartates. Both of the catalytic aspartic acid groups were unprotonated in the original model; theoretically, however, their proximity favors the protonation of one (or the other) at any given time.

To address this concern, Randall Radmer constructed an alternate model of the HIV-1 protease in which one of the aspartates is protonated. The residue chosen for protonation is the one that does not H-bond to the peptide-based inhibitors in the extant complex structures (inhibitor binding breaks the symmetry of the dimer so that the mono-

mers can be distinguished from one another). It is somewhat surprising that minimizations using the second protease model yielded largely the same results as those using the first. In each case, the S2-S2′-spanning orientation 452 is found to have the best energy of interaction with the protease, approximately 10 kcal/mol better than that of orientation 51.

DelPhi. It should be stressed that the method used to calculate the DelPhi interaction energies in Table II is an approximation, even within the continuum dielectric model. The assumption is made that a suitable potential map can be calculated using the receptor alone; the potential is not recalculated in the presence of each orientation. A more rigorous application of DelPhi to calculate electrostatic interaction energies in solution has been described,[25] and involves evaluation of a full thermodynamic cycle including the bound and unbound states of the molecules.

The energies obtained with the high-dielectric site model (Table II) are relatively small and yield completely different rankings (Table III) than the AMBER total or electrostatic interaction energies (Table I). This model undoubtedly overestimates charge screening due to solvent. The low-dielectric site model yields rankings somewhat consistent with the AMBER electrostatic interaction energies; both measures place orientations 204, 452, and 296 among the top five (Table III).

Inhibition data. Another source of information is experimental data on haloperidol and its derivatives. Several analogs have been tested for inhibition of the HIV-1 protease *in vitro*; some generalizations can be made. First, inhibitory activity tends to increase when the carbonyl of haloperidol is replaced with a large hydrophobic group, although certain shapes are apparently not well accomodated by the site. Many compounds with this kind of substitution are thioketals, such as UCSF8 (Figure 1). Second, activity

decreases by only a factor of two when the hydroxyl group and the chlorine are replaced with hydrogens. The relative contributions of the two modifications to the change in activity are not known. Third, the effects of quaternizing the nitrogen vary; the N-phenethyl derivative is approximately as active as the parent compound, haloperidol, while the N-methyl and N-oxide derivatives have much lower activities.

Could these observations be used to determine how haloperidol binds? The data suggested that the hydrophobic groups replacing the carbonyl bind in a fairly well-defined hydrophobic pocket. Unfortunately for the purposes of analysis, the subsites all fit this description to some extent. George Seibel's work in docking several dynamically generated conformations of UCSF8 and performing calculations on the complexes, however, showed that S2-S2′-spanning orientations similar to haloperidol orientation 452 are qualitatively (visually apparent fit) and quantitatively (magnitude of interaction energy) favorable. Notably, the "best" orientation he found is OH-down and NH-up, while orientation 452 is OH-up and NH-down. In George's structure, the hydroxyl is much closer to one of the active site aspartates than to the other. The nitrogen could be interacting in some way with one or both of the flaps, possibly through a water molecule. The data on the OH- and N-modified compounds suggested that both regions of haloperidol and its derivatives contribute to binding, but an unambiguous distinction between OH-up and OH-down models could not be made without further information. In addition, inconsistencies in the structure-activity relationships were seen that could best be resolved by the postulation of multiple binding modes.[26]

Crystal structure. The most direct information, of course, is structural. Months after the studies described above, coordinates for a complex of the HIV-1 protease with UCSF8 were determined crystallographically.[18] In the initial version, the piperidinyl por-

tion of the molecule was in a twist-boat conformation; in the final version (circa May 1992), it is in a chair conformation. The overall protease conformation is about the same in these two structures and is more similar to the open, unliganded structure[6] than to the conformation observed to bind peptide-based inhibitors.[11] The newer, chair-containing structure is shown in Figure 5. The inhibitor is closer to the "roof" of the cavity than to the "floor;" a chloride ion sits between the piperidine nitrogen and the protease flaps. In contrast, the original docking of BIBSEK to the uncomplexed protease (orientation 51) placed the inhibitor close to the floor, with the OH down between the active site aspartic acid residues. Orientation 452, favored by energy minimizations in the context of the complexed conformation of the protease, is also close to the floor but with OH up and NH down. Each predicted mode is angled relative to the observed mode. The crystallographic orientation is shown with orientations 51 and 452 in Figure 6 (complexes were placed in the same frame of reference by matching the nonflap regions of the protease structures).

The spheres used to generate orientation 51 were based on the molecular surface of the floor only. Renée DesJarlais had also created a cluster of spheres completely filling the site, based on the surface of the entire cavity. With the larger cluster and DOCK 3.0,[19] the experimental binding mode could be identified *given the ligand conformation from the new crystal structure*. The best-scoring (force field) docking of UCSF8 with the uncomplexed conformation of the protease is compared to the experimental position in Figure 7. In addition, the contact score identifies orientations similar to the symmetry-related partner of the ligand shown in Figure 5. Since the protease structure used for docking is completely symmetrical, this is an equally successful result. Therefore, the observed binding mode could be generated and identified without prior knowledge of the

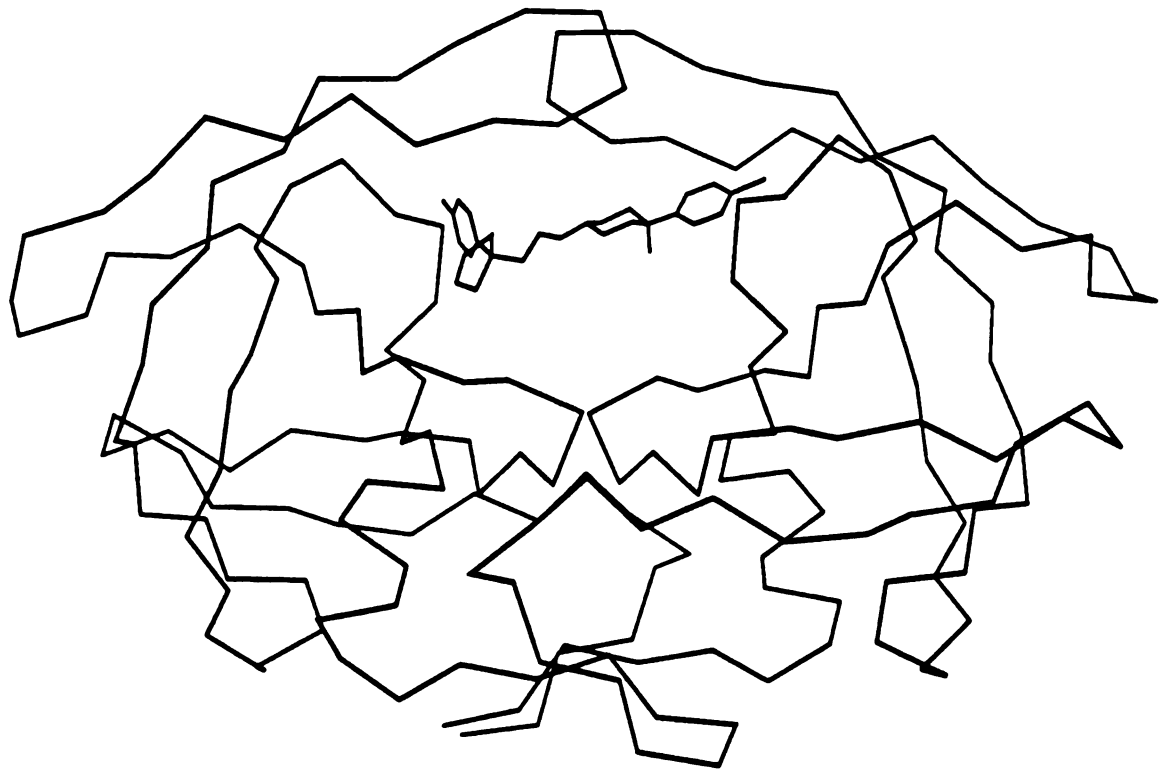


Figure 5. The HIV-1 protease/UCSF8 complex.[18] The Cα trace of the protein is shown.

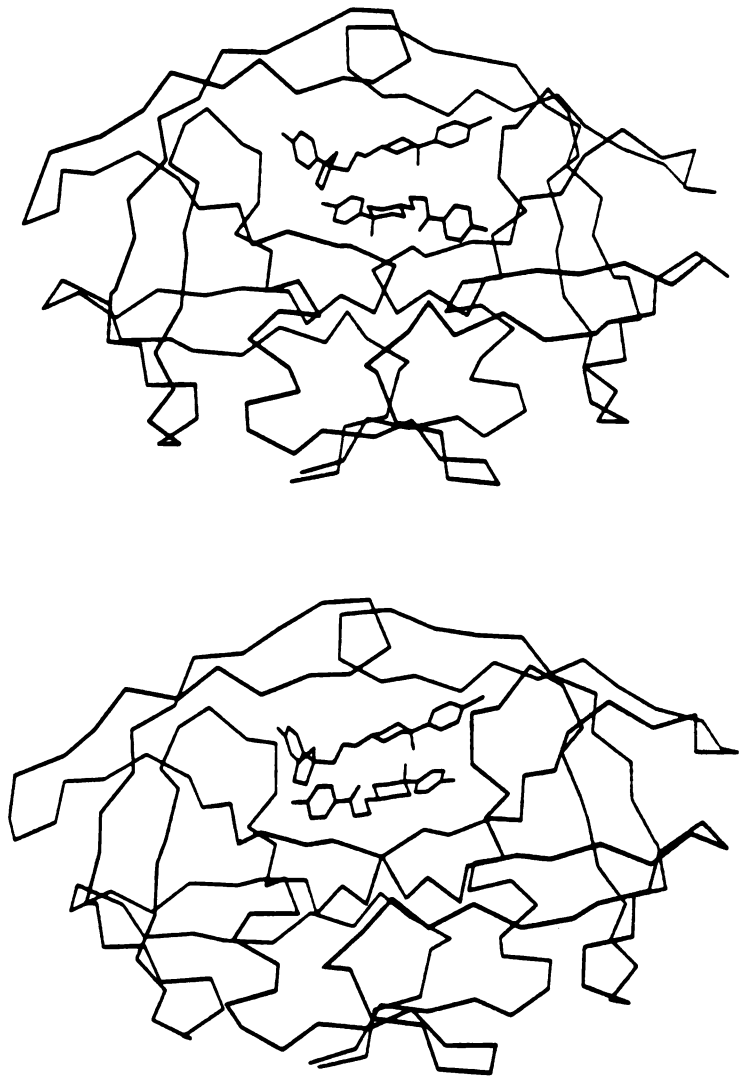


Figure 6. Comparison of the experimental HIV-1 protease/UCSF8 complex structure with orientations 51 (top) and 452 (bottom).

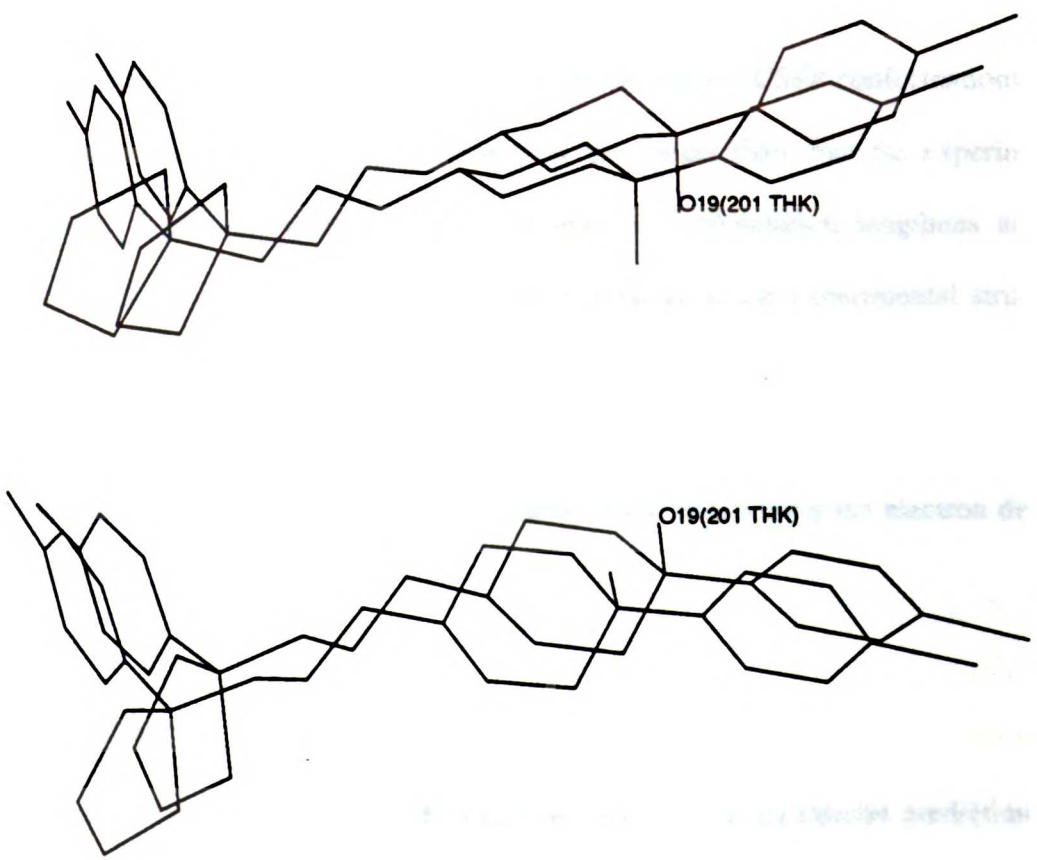


Figure 7. Two views of the best orientation of UCSF8 according to the force field score superimposed on its experimental position (labeled).

structure except the ligand conformation. The conformation of the uncomplexed protease was apparently similar enough to the conformation complexed with UCSF8 to be used successfully for docking. It was not necessary to include the chloride ion in either the receptor or the ligand representation.

The observed binding mode was not reproduced using UCSF8 conformations generated by other means, apparently because they are longer than than the experimental conformation. Upon minimization, the experimental conformation lengthens and no longer docks in the observed mode. There is uncertainty in the experimental structure, however, as revealed by the difficulty in discerning the boat and chair conformations of the piperidine ring. The presence of overlapping, symmetry-related binding modes and the flexibility of the alkyl portion of the inhibitor made interpreting the electron density difficult.[18]

CONCLUSIONS

It is instructive to consider which factors can prevent the successful prediction of a ligand binding mode. As stated by van Gunsteren and Berendsen,[27] there are two fundamental problems in molecular modeling: limitations in sampling the configurations of a system and limitations in the accuracy of the potential energy function (and thus the representation of the system) used to evaluate the configurations. In docking, sampling difficulties occur in ligand conformational space, receptor conformational space, and orientation space (the relative positions of the two molecules). The evaluation of docked configurations is generally limited to estimates of the interaction enthalpy, as hundreds to thousands of configurations per ligand need to be examined in a reasonable amount of time. Free energy determinations require extensive simulations with explicit solvent

(which adds many degrees of freedom that must be sampled statistically) and careful parameterization of each molecule; they are not affordable in this context. Furthermore, approximations inherent in the potential function can limit accuracy even in the calculation of interaction enthalpies.

As explicit waters and ions are usually not considered during the evaluation of a docked complex, it can be difficult to predict a configuration in which one or more water molecules or ions is intimately involved. In addition, there may be multiple binding modes that differ only slightly in free energy. While it would be reasonable to consider the prediction of any of these a success, it is unlikely for all modes to be known. It is possible for a prediction to be correct but considered incorrect because the predicted mode has not been observed.[26]

Turning to the specific case of the UCSF8/HIV-1 protease complex, the initial docking study used a slightly different ligand (haloperidol) in a somewhat different conformation than observed for UCSF8 (more extended) and a slightly different conformation of the protease (the unliganded structure). In the final analysis, this choice of receptor conformation was correct. Orientational sampling, however, was biased by the preconception that the ligand must be interacting with the aspartic acid residues in the floor of the site. Evaluation consisted of simple contact scoring and manual inspection; orientation 51 was favored. Subsequent docking and AMBER energy minimizations used the 4hvp conformation of the protease, which differs substantially from the conformation observed in the complex with UCSF8. Since the site becomes much more constricted upon binding a peptide-based ligand such as MVT-101, this is again an assumption that the ligand interacts directly with the catalytic aspartates. It is not possible to reproduce the experimental binding mode using the 4hvp protease conformation since the flaps occlude the

region where UCSF8 binds. As above, haloperidol rather than UCSF8 was docked. Orientation 452 was favored by this work.

After-the-fact docking used the uncomplexed conformation of the protease, spheres filling the entire active site, and the experimental complexed conformation of UCSF8. Both the contact score and the DOCK 3.0 force field score identified the experimental binding mode; these simple scoring schemes were apparently not limiting. It was unnecessary to consider explicit water molecules or ions during evaluation, even though a chloride ion is an integral part of the complex. Evidently, the crucial factor is knowledge of the detailed shape of the ligand. All other necessary information was available beforehand; the assumption that the ligand would directly contact the catalytic residues was the only other barrier to success. These two obstacles exemplify conformational and orientational aspects, respectively, of the fundamental problem of limited sampling.

References

1. W. C. Farmerie, D. D. Loeb, N. C. Casavant, C. A. Hutchison, III, M. H. Edgell, and R. Swanstrom, *Science,* **236,** 305 (1987).

2. T. D. Meek, D. M. Lambert, G. B. Dreyer, T. J. Carr, T. A. Tomaszek, Jr., M. L. Moore, J. E. Strickler, C. Debouck, L. J. Hyland, T. J. Matthews, B. W. Metcalf, and S. R. Petteway, *Nature,* **343,** 90 (1990).

3. H. P. Schnebli and N. J. Braun in *Proteinase Inhibitors,* A. J. Barrett and G. Salvensen, Eds., Elsevier Science, New York, 1986, vol. 12, pp. 613-627.

4. R. L. DesJarlais, R. P. Sheridan, G. L. Seibel, J. S. Dixon, I. D. Kuntz, and R. Venkataraghavan, *J. Med. Chem.,* **31,** 722 (1988).

5. R. L. DesJarlais, G. L. Seibel, I. D. Kuntz, P. S. Furth, J. C. Alvarez, P. R. Ortiz de Montellano, D. L. DeCamp, L. M. Babé, and C. S. Craik, *Proc. Natl. Acad. Sci. USA,* **87,** 6644 (1990).

6. A. Wlodawer, M. Miller, M. Jaskólski, B. K. Sathyanarayana, E. Baldwin, I. T. Weber, L. M. Selk, L. Clawson, J. Schneider, and S. B. H. Kent, *Science,* **245,** 616 (1989).

7. F. C. Bernstein, T. F. Koetzle, G. J. B. Williams, E. F. Meyer, Jr., M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi, *J. Mol. Biol.,* **112,** 535 (1977).

8. E. E. Abola, F. C. Bernstein, S. H. Bryant, T. F. Koetzle, and J. Weng, in *Crystallographic Databases: Information Content, Software Systems, Scientific Applications,* F. H. Allen, G. Bergerhoff, and R. Seivers, Eds., Data Commission of the International Union of Crystallography, Bonn/Cambridge/Chester, 1987, pp. 107-132.

9. F. H. Allen, O. Kennard, W. D. S. Motherwell, W. G. Town, and D. G. Watson, *J. Chem. Doc.,* **13,** 119 (1973).

10. T. E. Ferrin, C. C. Huang, L. E. Jarvis, and R. Langridge, *J. Mol. Graph.*, **6,** 13 (1988).

11. M. Miller, J. Schneider, B. K. Sathyanarayana, M. V. Toth, G. R. Marshall, L. Clawson, L. Selk, S. B. Kent, and A. Wlodawer, *Science,* **246,** 1149 (1989).

12. M. H. J. Koch, *Mol. Pharmacol.,* **10,** 425 (1974).

13. I. D. Kuntz, J. M. Blaney, S. J. Oatley, R. Langridge, and T. E. Ferrin, *J. Mol. Biol.,* **161,** 269 (1982).

14. S. J. Weiner, P. A. Kollman, D. A. Case, U. C. Singh, C. Ghio, G. Alagona, S. Profeta, Jr., and P. Weiner, *J. Am. Chem. Soc.,* **106,** 765 (1984).

15. G. Seibel, U. C. Singh, P. K. Weiner, J. Caldwell, and P. A. Kollman, AMBER 3.0 Revision A, University of California, San Francisco, 1989.

16. I. Klapper, R. Hagstrom, R. Fine, K. Sharp, and B. Honig, *Proteins,* **1,** 47 (1986).

17. M. K. Gilson, K. A. Sharp, and B. H. Honig, *J. Comp. Chem.,* **9,** 327 (1987).

18. E. Rutenber, E. B. Fauman, S. Fong, P. S. Furth, P. R. Ortiz de Montellano, E. Meng, I. D. Kuntz, D. L. DeCamp, R. Salto, C. Craik, and R. M. Stroud, manuscript in preparation.

19. E. C. Meng, B. K. Shoichet, and I. D. Kuntz, *J. Comp. Chem.,* **13,** 505 (1992).

20. J. Gasteiger and M. Marsili, *Tetrahedron,* **36,** 3219 (1980).

21. M. Marsili and J. Gasteiger, *Croat. Chem. Acta,* **53,** 601 (1980).

22. J. Gasteiger and M. Marsili, *Organ. Magn. Reson.,* **15,** 353 (1981).

23. N. A. Roberts, J. A. Martin, D. Kinchington, A. V. Broadhurst, J. C. Craig, I. B. Duncan, S. A. Galpin, B. K. Handa, J. Kay, A. Kro:hn, R. W. Lambert, J. H. Merrett, J. S.

Mills, K. E. B. Parkes, S. Redshaw, A. J. Ritchie, D. L. Taylor, G. J. Thomas, and P. J. Machin, *Science,* **248,** 358 (1990).

24. J. Erickson, D. J. Neidhart, J. VanDrie, D. J. Kempf, X. C. Wang, D. W. Norbeck. J. J. Plattner, J. W. Rittenhouse, M. Turon, N. Wideburg, W. E. Kohlbrenner, R. Simmer, R. Helfrich, D. A. Paul, and M. Knigge, *Science,* **249,** 527 (1990).

25. M. K. Gilson and B. Honig, *Proteins,* **4,** 7 (1988).

26. The latest crystallographic data supports the existence of multiple binding modes, and in fact a binding mode similar in many respects to orientation 51 has been observed.

27. W. F. van Gunsteren and H. J. C. Berendsen, *Angew. Chem. Int. Ed. Engl.,* **29,** 992 (1990).