

UCSF

UC San Francisco Electronic Theses and Dissertations

Title

Strategies for using molecular shape and electrostatic properties in ligand design / by
Cynthia Beth Corwin

Permalink

<https://escholarship.org/uc/item/7f25t92z>

Author

Corwin, Cynthia Beth

Publication Date

1995

Peer reviewed|Thesis/dissertation

STRATEGIES FOR USING MOLECULAR SHAPE AND ELECTROSTATIC PROPERTIES

IN LIGAND DESIGN

by

Cynthia Beth Corwin
A.B., Cornell University, 1985

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

Nothing as significant or as long in the making as a doctoral degree is truly achieved alone. Many people, too many to list here, have helped me in my graduate career, and I owe them all a debt of gratitude. I would like to thank individually some of those who made significant contributions.

First and foremost, thanks are due to my research advisor, Prof. I.D. Kuntz, who provided me with continued support and encouragement on what was at times a difficult journey. Profs. Peter Kollman and Fred Cohen shared their time and their wisdom with me. I owe Prof. Patricia Babbitt thanks for her advice and encouragement while I was preparing for orals, and for her continued career guidance.

Elaine Meng joined the Kuntz group a year before I did, and I was very fortunate to benefit from her knowledge and friendship. Diana Roe and I have shared the entire graduate school experience. I am also grateful for the ideas and support of Donna Hendrix, and I wish her success in her graduate career.

I owe something to every member of the Kuntz group. Dale Bodian and Brian Shoichet, whose careers began before mine, passed along scientific knowledge as well as parts of the DOCK program. Guy Bemis was of great help to me in understanding molecular similarity. Dan Gschwend, Andy Good, Todd Ewing, and Yax Sun shared advice as well as office space. I was also fortunate to work with Connie Oshiro, Barbara Chapman, Luke Hoffman, Keith Burdick, and Malin Young; I have learned from each of them.

I would like to thank the women of Women in Life Sciences. As an organization, WILS has helped me to learn about career possibilities and

allowed me to meet other women in science. Individually, many of the women in the organization have given me their friendship and support, and I am grateful to all of them.

Finally, the most special thanks are due to my husband, Jeffrey Millman, for his support through graduate school and beyond.

Strategies for Using Molecular Shape and Electrostatic Properties in Ligand Design

Cynthia B. Corwin

Dissertation Abstract

Computer modeling methods may reduce the cost of drug discovery by reducing the number of compounds which must be synthesized and tested in the laboratory. Docking, which explores the geometric fit of potential drugs into biological macromolecules, has been a successful method for locating new drug leads. The work described in this dissertation extends the docking method in two directions: selecting drug candidates based on their electrostatic complementarity to macromolecules and applying docking methods where the target macromolecule's structure is unknown but structures are available for molecules which bind to it.

Chapter 1 describes the docking of database molecules to the charged P1 binding pocket of trypsin and their scoring by electrostatic complementarity to the site. Docking using electrostatic scoring alone retrieved compounds which were far too highly charged to bind to trypsin. When a correction for the energy cost of desolvating the charges was added, top ranking compounds had appropriately fewer charges. 21 of these compounds were tested; six of them inhibited the hydrolysis of Z-Gly-Pro-Arg-aminomethyl coumarin by trypsin.

Chapter 2 describes docking to a positive image of a target macromolecule. Target images were derived from compounds which bind to the dopamine D2 receptor and from an X-ray crystal structure of bovine pancreatic trypsin inhibitor. Docked compounds were scored using their electrostatic similarity to the target, the proximity of their atoms to target atoms, and the overlap of their surfaces with the target surface. Electrostatic

similarity alone increased the fraction of active and related compounds among the top-scoring molecules up to 8-fold. The atom-based geometric scoring method alone was not useful for locating known molecules. Combining the scores increased the fraction of known molecules among the top-scoring compounds. It was necessary to select a weighting factor for the scores on a case-by-case basis.

Appendices 1, 2, and 3 include source code for a docking subroutine and for scoring. Appendices 4 and 5 document the use of the docking software, and Appendix 6 describes retrieval of database molecules.

Table of Contents

Introduction.....	1
References.....	6
 Chapter 1: Electrostatic Scoring and Correction for Solvation Energy in Docking to a Charged Molecule.....	7
Introduction.....	7
Methods.....	9
Docking to trypsin.....	9
Docking to chymotrypsin.....	14
Results and Discussion.....	15
Docking to trypsin.....	15
Docking to chymotrypsin.....	29
Conclusions.....	29
References.....	32
 Chapter 2: Docking to Positive Images of Receptor Sites.....	35
Introduction.....	35
Methods.....	39
Dopamine D2 Pharmacophore: Positive Docking.....	39
Testing DOCK as a Tool for Pharmacophore Generation.....	63
Bovine Pancreatic Trypsin Inhibitor: Positive Docking.....	68
Results.....	79
Conclusions.....	113
References.....	115
 Conclusions.....	120
 Appendix 1: Correction of an Error in DOCK Distance Handling.....	122
FORTRAN Code for the Revised Version of makbin.....	127
References.....	130
 Appendix 2: Code for Creation of a Grid to Mark Acceptable Ligand Positions in Positive Docking.....	131
FORTRAN Code for the dconst, ddist, and gauss3 Subroutines from the Revised Version of CHEMGRID.....	131
References.....	137
 Appendix 3: Source Code for Surface-Based Scoring – Surfgrid and Surfscore.....	138
Makefile for surfgrid.....	138
Header files used with surfgrid.....	139
Source code for surfgrid and subroutines.....	141
Source code for the surface scoring routines.....	158

Appendix 4: A Beginner's Guide to DOCK 3.5	162
Scope of This Document	162
What DOCK Can Do for You.....	162
Basic DOCK References.....	162
Overview of the DOCK package	164
A Caution Concerning Disk Space.....	165
Working With Macromolecular Models and Generating the Molecular Surface.....	165
Representing the Site With Spheres	167
Creating the Scoring Grids.....	169
The Contact Scoring Grid (DISTMAP)	169
The Electrostatic Potential Map (DelPhi).....	170
The Force Field Scoring Grid (CHEMGRID)	171
Preparing Ligand Molecules.....	174
Labeling Atoms and Spheres for Chemical Matching (Optional).....	176
Running DOCK.....	177
Looking at the Results	183
 Appendix 5: A Guide to Using DOCK for Beginners at UCSF	 185
 Appendix 6: Guide to Using ISIS at UCSF	 195

List of Tables

Chapter 1:

Table 1: Compounds in the Trypsin Test Set.....	13
Table 2: Compounds in the Chymotrypsin Test Set.....	16
Table 3: Top-scoring Compounds From Docking the FCD to trypsin with DelPhi-based Scoring.....	19
Table 4: Top-scoring Compounds From Docking the FCD to trypsin with DelPhi-based Scoring plus solvation correction.....	22
Table 5: Results of Testing Compounds Selected From Docking the FCD to Trypsin Using DelPhi-based Scoring with Solvation Correction.....	27

Chapter 2:

Table 1: Compounds from the literature used in the dopamine D2 test database.....	48
Table 2: Compounds found by searching the MDDR database which were included in the D2 test database.....	51
Table 3: Examples of trypsin inhibitors and related compounds.	71
Table 4: Labels assigned to spheres derived from amino acid residues in BPTI. Atom names are those used in Brookhaven Protein Data Bank (Bernstein, Koetzle et al. 1977.....	74
Table 5: Labels assigned to charged and polar functional groups in database molecules.	75
Table 6: RMS deviations between hand and DOCK alignments.....	80
Table 7: Summary of electrostatic scoring schemes investigated in the BPTI system.....	104
Table 8: Average, minimum and maximum numbers of nonhydrogen atoms among the 300 top-scoring compounds obtained by docking the ACD-derived test database to the positive image of BPTI using variations on the geometric scoring scheme.....	105

Appendix 1:

Table 1: Interatomic Distances and Bin Contents in Molecule III.....	125
----------------------------------------------------------------------	-----

List of Figures

Chapter 1:

Figure 1: Trypsin molecular surface	10
Figure 2: Sphere centers used for docking to trypsin.	11
Figure 3: Sphere centers used for docking to chymotrypsin.....	15
Figure 4: Scores from docking the chymotrypsin test set to chymotrypsin, plotted against KI values.	30
Figure 5: DelPhi electrostatic scores without solvation correction vs. KI values for the chymotrypsin test set.	30
Figure 6: DOCK 3.0 (force field) scores vs. KI values for the chymotrypsin test set.....	31

Chapter 2:

Figure 1: Molecules used in testing positive docking methods in the dopamine D2 pharmacophore.....	40
Figure 2: The four molecules aligned to build the dopamine D2 pharmacophore model.....	43
Figure 3: Alignment of the four molecules.	43
Figure 4: Sphere centers derived from dopamine D2 pharmacophore.....	44
Figure 5: Box defining the scoring region for calculations in the dopamine D2 pharmacophore.....	46
Figure 6: Query structures used to search the MDDR database.....	47
Figure 7: Molecules used to test DOCK as a tool for aligning molecules and generating pharmacophores.....	64
Figure 8: Positive docking centers with BPTI surface.	69
Figure 9: Positive docking centers with box defining scoring grid.....	69
Figure 10: A two-dimensional illustration of the representation of a molecular surface on a grid.....	77
Figure 11: Illustration of the use of grid-based approximations to a molecular surface in scoring.	78
Figure 12: Hand and DOCK alignments to the dopamine D2 pharmacophore.....	81
Figure 13: Literature alignment of 4aR,10bR-7-hydroxy-4-n- propyl-1,2,3,4,4a,5,6,10b- octahydrobenzo[f]quinoline and orientation with top electrostatic score.....	84
Figure 14: Percentage of active compounds found in the D2 database using the sum of the electrostatic score and the atom-atom score capped at -50.....	86
Figure 15: Percentage of active compounds found in the MDDR database using the sum of the electrostatic score and the atom-atom score capped at -50.....	86
Figure 16: Percentage of actives found by searching the D2 database using electrostatic scoring alone.....	88

Figure 17: Percentage of actives found by searching the MDDR database using electrostatic scoring alone.....	88
Figure 18: Atom-atom (with cap -50) versus electrostatic scores for docking of the D2 database.....	89
Figure 19: Atom-atom (with cap -400) versus electrostatic scores for docking of the D2 database.....	90
Figure 20: Atom-atom (with cap -1000) versus electrostatic scores for docking of the D2 database.....	91
Figure 21: Atom-atom (with cap -10000) versus electrostatic scores for docking of the D2 database.....	92
Figure 22: Atom-atom (with cap -25000) versus electrostatic scores for docking of the D2 database.....	93
Figure 23: Molecules with similar geometric scores in their DOCK alignments.....	94
compounds and inactive compounds into different regions.....	95
Figure 24: Atom-atom versus electrostatic score for D2 docking using a scale factor of 0.002.....	96
Figure 25: Atom-atom versus electrostatic score for D2 docking using a scale factor of 0.001.....	97
Figure 26: Examples of top-scoring compounds from D2 docking using scaled scores.....	98
Figure 27: Molecules aligned by DOCK using lablscan.....	100
Figure 28: Another view of the aligned molecules.....	100
Figure 29: Inhibitor-related compounds found in the ACD subset by docking to BPTI using "standard" electrostatics.....	103
Figure 30: Inhibitor-related compounds found in the ACD subset by docking to BPTI using the Gaussian electrostatic function.....	103
Figure 31: Inhibitor-related compounds found in the ACD subset using the Gaussian electrostatic function and a 5 Å distance cutoff.....	104
Figure 32: Inhibitor-related compounds found by docking to the ACD subset using electrostatic scoring and coloring.....	107
Figure 33: Compound with the best atom-atom score (cap -10000) in docking to BPTI.....	107
Figure 34: Inhibitor-related compounds found in docking the ACD subset to BPTI using atom-atom scoring (cap -10000).....	108
Figure 35: Compound with the best atom-atom (cap -1000) score in docking to the ACD subset.....	108
Figure 36: Top-scoring compound from docking to the ACD subset using atom-atom scoring with a Gaussian approximation to $1/r$	109
Figure 37: Top compound from atom-atom scoring normalized	

	by the number of nonhydrogen atoms.	109
Figure 38:	Inhibitor-related compounds found in docking the ACD subset to BPTI using normalized atom-atom scoring.....	110
Figure 39:	Top-scoring compound from docking the ACD subset to BPTI using atom-atom scoring normalized by the square root of the number of nonhydrogen atoms.....	110
Figure 40:	Inhibitor-related compounds found in docking the ACD subset to BPTI with atom-atom scoring normalized by the square root of the number of nonhydrogen atoms.	111
Figure 41:	Top-scoring compound from docking with surface-based scoring.	112
Figure 42:	Top-scoring orientation from single mode docking with surface-based scoring.	112
Appendix 1:		
Figure 1:	Molecule III.	123

Introduction

Drug discovery has been very important to health care and remains so today. While drugs exist for many conditions, new diseases such as AIDS are appearing, and organisms which cause known diseases are developing resistance to drugs which were formerly effective. In addition, the arsenal of antiviral drugs remains limited, and the undesirable side effects of many known drugs might be avoided if novel compounds with slightly different properties were found to replace them. At the same time, drug discovery is becoming more expensive and time-consuming. Traditionally, pharmaceutical companies have relied upon serendipity, screening of many compounds, and the knowledge of medicinal chemists to find lead compounds from which they developed new drugs. The number of compounds tested to find a commercial product in this way has been increasing. Rising health care costs are also causing pressure to deliver drugs more cheaply, leading to a search for ways to make research more efficient.

Computer-aided drug discovery methods can increase the efficiency of the search for new leads, since automated methods allow the examination of many more compounds than a chemist might be able to consider. Computational methods can be used to pre-screen compounds for testing so that the test set is enriched in molecules most likely to show activity. They can also suggest or locate novel compounds for testing in a way that is less dependent on the experience of an individual medicinal chemist. In recent years the structures of many biologically active molecules have become available, but these molecules are often unsuitable as drugs because problems with bioavailability make administration difficult; computational methods

can suggest replacements which have similar activities but whose properties make them more suitable as drugs.

The number of biomolecules for which structural information is available is rapidly increasing, making it possible in more and more cases to search for ligands targeted to a particular receptor or for mimics for an endogenous ligand. The rapidly decreasing cost of computer power makes it possible to explore more potential ligands than was practical a few years ago or to increase the level of detail used in a method in order to make it more effective at discriminating potentially active molecules from inactive ones. The Available Chemicals Directory from MDL Information Systems contains three-dimensional structures of compounds which may be purchased for testing, thus increasing the utility of methods which search structural databases for new leads, and tools such as CONCORD (Pearlman 1987) allow pharmaceutical companies to generate three-dimensional structural databases of compounds in their archives.

All these factors make database searching methods valuable tools in the lead discovery effort. In this work I have investigated and developed searching methods using structural information from receptors and from molecules known to be biologically active.

The methods I have investigated depend upon the rapid generation of many orientations of small molecules relative to a representation of a target receptor site, for which complementary ligands are sought, or to a pharmacophore or other target image for which similar compounds are required. DOCK (Kuntz, Blaney et al. 1982), a computer program developed at UCSF, is capable of such rapid orientation generation, and the thoroughness with which it samples orientations is under user control. In its earliest form (Kuntz, Blaney et al. 1982), DOCK was used to generate many

orientations for a single molecule relative to a receptor binding site; the capability to search databases of molecules and record the best scores against a particular receptor site was introduced in DOCK 1.1 (DesJarlais, Sheridan et al. 1988). Molecules were scored according to how many favorable van der Waals contacts they made with the receptor and were penalized for unfavorably close contacts. In DOCK 2.0 (Shoichet, Bodian et al. 1992), precalculation of local contributions to the score at locations on a grid in space was introduced, allowing for more rapid searching and thus permitting the use of larger databases. These early versions of DOCK did not attempt to address the chemistry of the molecules they matched to receptors.

I have investigated the extension of DOCK in two directions. First, I have applied methods developed by Brian Shoichet and Elaine Meng for including electrostatic complementarity and solvation in DOCK scoring to trypsin. Second, I have developed methods for locating molecules which have steric and electrostatic similarity to active compounds, allowing DOCK to be used to find ligands in cases where a receptor structure is not yet available.

While contact scoring can be used to locate molecules which fit well into a receptor site, it does not account for electrostatic interactions, which are essential to the specificity of ligand binding in many systems. A scoring method which took electrostatics into account would allow DOCK to locate compounds which have charge complementarity to a receptor; these compounds could be tested directly, without the need for synthesis to introduce chemical complementarity. Brian Shoichet and Elaine Meng introduced a method for scoring using the electrostatic potential produced by DelPhi in version 2.1 of DOCK (Meng, Shoichet et al. 1992) , and Brian Shoichet added a solvation term in DOCK 2.2. Chapter 1 summarizes the

application of these methods to trypsin, which has an important charge in its substrate binding pocket.

Despite rapid increases in the number of receptor three-dimensional structures solved, a majority of problems in drug design involve receptors of unknown structure. In such cases, the information available about the receptor is derived indirectly from structural information about its ligands — drugs which are known to bind to it or macromolecular ligands whose structure is known. In Chapter 2, I describe the use of this information in developing target representations for docking. Steric and electrostatic resemblance to compounds known to be active are both important characteristics for a candidate to have the same activity, so I based scoring methods on these factors. I experimented with these new methods in the dopamine D2 pharmacophore, a system in which the available information comes from small drug molecules, and in trypsin, where BPTI provides a template for locating new molecules which are likely to bind to the enzyme. In the course of this work I discovered that some unusually small sets of ligand and target internal distances were not being handled properly by DOCK; a revised version of the DOCK subroutine makbin which corrects this problem is described in Appendix 1. I have written a modified version (Appendix 2) of the program CHEMGRID which generates the "bump" grid used by DOCK to determine where ligand atoms may fall; it restricts ligands to the interior of a positive-image target instead of the exterior of a receptor. Appendix 3 includes surfgrid, a program which generates a lattice-based representation of the surface of a molecule for use in scoring orientations of database molecules using their surface overlap with a target image.

In my years at UCSF I have acquired a working knowledge of several software packages, particularly DOCK and the database-management

programs from MDL Information Systems. In order to make learning these programs an easier task for future group members, I have written documentation. The DOCK Beginners' Guide, included as Appendix 4, summarizes the steps involved in setting up a docking problem and conducting a DOCK run, while the guide to docking for UCSF beginners in Appendix 5 describes the details of using DOCK on the Kuntz group computers. A set of HTML documents, the text of which is included in Appendix 6, describes the basics of the MDL ISIS software and summarizes its use for retrieving information about compounds which scored well in DOCK runs.

References

Available Chemicals Directory. San Leandro, CA, MDL Information Systems, Inc.

DesJarlais, R. L., R. P. Sheridan, et al. (1988). "Using Shape Complementarity as an Initial Screen in Designing Ligands for a Receptor Binding Site of Known Three-Dimensional Structure." Journal of Medicinal Chemistry 31(4): 722-729.

Kuntz, I. D., J. M. Blaney, et al. (1982). "A Geometric Approach to Macromolecule-Ligand Interactions." Journal of Molecular Biology 161: 269-288.

Meng, E. C., B. K. Shoichet, et al. (1992). "Automated Docking with Grid-Based Energy Evaluation." Journal of Computational Chemistry 13(4): 505-524.

Pearlman, R. S. (1987). "Rapid Generation of High-Quality Approximate 3-D Molecular Structures." Chemical Design Automation News 2(1): 5-6.

Shoichet, B. K., D. L. Bodian, et al. (1992). "Molecular Docking Using Shape Descriptors." Journal of Computational Chemistry 13(3): 380-397.

Chapter 1: Electrostatic Scoring and Correction for Solvation Energy in Docking to a Charged Molecule

Introduction

Small molecules which fit well into a receptor binding site are likely to bind well to the receptor, which makes them good candidates to be new drugs. The number of such binding sites whose three-dimensional structure is known has increased rapidly in recent years as more and more protein structures have been solved. Computer methods of searching for molecules to fit these sites can complement the work of medicinal chemists by locating novel leads and screening out molecules which do not fit the site. DOCK (Kuntz, Blaney et al. 1982; DesJarlais, Sheridan et al. 1988; Shoichet, Bodian et al. 1992) has been a useful tool for locating molecules whose shapes match the shapes of receptor sites. Early versions of DOCK based their scoring solely on molecular geometry. Ligands were assigned scores based on the number of favorable van der Waals contacts they could make with the receptor minus a penalty for contacts which were too close; this is termed contact scoring. The authors anticipated that DOCK would be used to search for molecules whose shape fit protein active sites well. Users would then replace parts of these skeletons with chemically appropriate groups, and the resulting molecules would be synthesized and tested for binding to the protein. However, synthesis can be expensive and time-consuming, and molecules designed in this way are not always amenable to synthesis. If it were possible to purchase compounds for testing (or, in a pharmaceutical company, retrieve them from an archive), more DOCK predictions could be tested with less expense.

While contact scoring captures steric interactions by assigning the best scores to compounds which have the most surface area in contact with the

receptor, electrostatic interactions are also essential to the specificity of binding in many ligand-receptor systems. A scoring method which took electrostatics into account should be useful in locating molecules which would complement the charges of a receptor, and would therefore be more likely to find molecules which could be tested for receptor binding without modification. When used with a database of compounds which may be purchased or otherwise obtained for testing, such a method could speed the process of making and evaluating DOCK predictions.

In DOCK 2.1, an electrostatic scoring scheme (Meng, Shoichet et al. 1992) was introduced. In this scoring scheme, the electrostatic potential due to a receptor is calculated using DelPhi (Gilson, Sharp et al. 1987) at points on a lattice superimposed on the receptor. Molecules are scored by summing the interaction energy of partial charges on each of their atoms with this electrostatic potential. As a test of this method, I have docked the Fine Chemicals Directory (MDL Information Systems, now called Available Chemicals Directory), a database of commercially available compounds, to a trypsin mutant. Trypsin is a particularly appropriate test system because a negatively charged aspartate in its S1 subsite is responsible for its specificity for cleaving next to lysine and arginine (Ruhlmann, Kukla et al. 1973; Sweet, Wright et al. 1974).

When a charged ligand interacts with a receptor site, it must be partially desolvated, and the energy cost of removing the surrounding water molecules may make a significant contribution to the overall free energy of binding. Accordingly, when evaluating interactions between charged ligands and receptors it is important to take solvation into account. Brian Shoichet introduced such a solvation term into DOCK 2.2. The solvation enthalpy calculated for each ligand molecule using methods developed by Alexander

Rashin (Rashin and Namboodiri 1987) is subtracted from each ligand's score. I used this method to dock the Fine Chemicals Directory to trypsin. In addition, I applied this scoring scheme in docking chymotrypsin to a set of small molecules which had been tested for chymotrypsin inhibition.

Methods

Docking to trypsin

Trypsin coordinates were taken from the *2trm* structure (Sprang, Standing et al. 1987) from the Brookhaven Protein Data Bank (Bernstein, Koetzle et al. 1977; Abola, Bernstein et al. 1987). This is the structure of rat trypsin in which residue 102, the catalytic aspartate, has been replaced by an asparagine; the enzyme is about four orders of magnitude less active than the wild type (Sprang, Standing et al. 1987). The benzamidine ligand and all ions and water molecules present in the crystal structure were removed. The MS algorithm (Connolly 1983) was used to generate a dot surface for all residues on the surface of the protein within 15 Å of serine 195 (Figure 1). Spheres were generated from this surface with SPHGEN (Kuntz, Blaney et al. 1982). The cluster produced by SPHGEN in the active site region did not sufficiently fill the site, so spheres from surrounding regions were selected and added to the cluster manually. The final cluster (Figure 2) contained 33 spheres. DISTMAP (Shoichet, Bodian et al. 1992) was used to generate a grid for contact scoring which enclosed the spheres and extended an additional 8 Å to each side in the x, y, and z directions. The electrostatic potential due to the entire trypsin molecule was calculated using version 3.0 of DelPhi (Gilson, Sharp et al. 1987). Three-step focusing was used in order to reduce errors associated with boundary conditions; in successive steps, the protein occupied 20%, 60%, and 90% of the grid. The protein was assigned AMBER united-atom partial charges (Weiner, Kollman et al. 1984). Internal and external dielectric

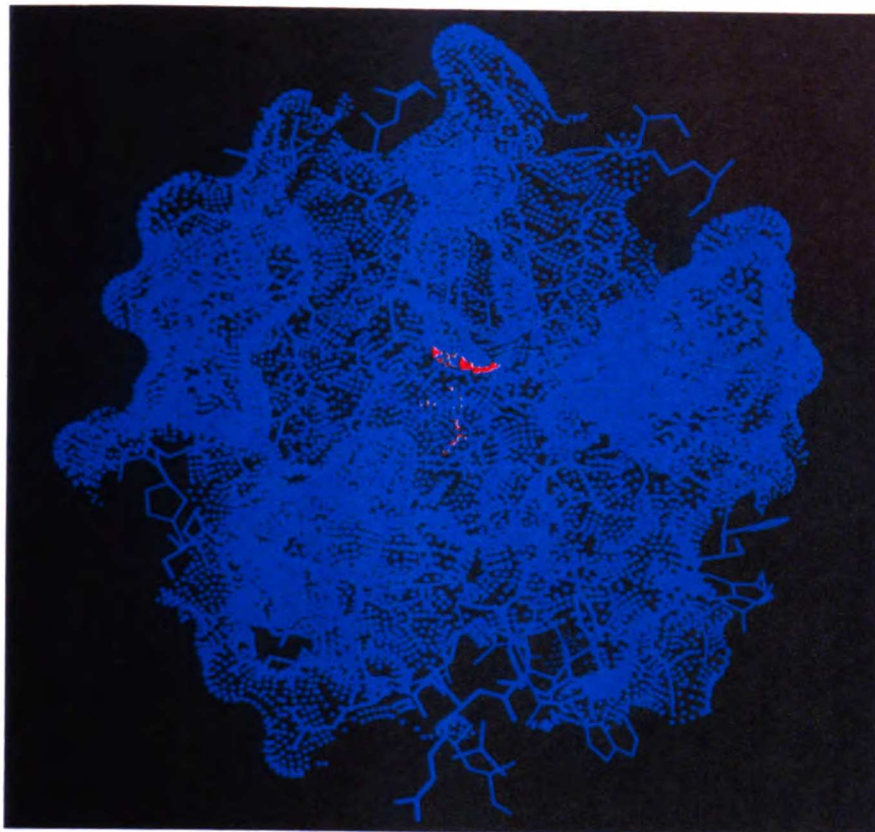


Figure 1: Trypsin molecular surface; Ser 195 is colored red.

constants were respectively 4 and 8, the ionic strength was 0.145 M, the ion exclusion radius was 2.0 Å, and the probe radius was 1.4 Å.

DOCK 2.1 and DOCK 2.2 generate orientations of each database molecule relative to the receptor by matching internal distances among spheres to those among ligand atoms. Both employ the "binning" algorithm originally used in DOCK 2.0 (Shoichet, Bodian et al. 1992) to group similar distances before matching. For this work, bin sizes and overlaps were kept small. Increasing these values causes more orientations to be generated for each ligand, with a corresponding increase in the amount of computer time required. Keeping the bins and overlaps small allowed docking of large



Figure 2: Sphere centers used for docking to trypsin.

databases in a reasonable amount of time. I used a receptor bin width of 0.20 Å and an overlap of 0.0 Å; ligand bin width and overlap were 1.0 Å and 0.0 Å.

The DOCK 2.2 electrostatic score is based on DelPhi, as with DOCK 2.1, but a correction for each ligand molecule's desolvation enthalpy and conformational flexibility is added to the score. The solvation correction is calculated by the method of Alexander Rashin (Rashin and Namboodiri 1987), which uses a continuum model for the solvent and approximates the solvation enthalpy of a molecule as the sum of its electrostatic interactions with the solvent and the solvation enthalpy of a nonpolar molecule of the same shape. The conformation correction is determined by multiplying the number of conformations generated for the ligand in COBRA (Leach and Prout 1990) by RT , where R is the gas constant and T is the absolute temperature, and adding the result to the score. Both corrections are

calculated for each ligand molecule in advance of docking and stored in a separate file which DOCK reads along with the molecule database. For DOCK 2.2, a new electrostatic potential grid was calculated with DelPhi; this time the internal dielectric used was 2. The region containing the spheres was treated as part of the protein, giving it the low dielectric value.

DOCK 3.0 (Meng, Shoichet et al. 1992) uses the same matching algorithm as DOCK 2.1 and DOCK 2.2, but it includes a force field scoring scheme. This score is a sum of a grid-based approximation to a Lennard-Jones potential and a Coulombic interaction energy. CHEMGRID (Meng, Shoichet et al. 1992), which calculates the receptor's contribution to both the van der Waals and electrostatic terms, uses a standard set of partial charges, in this case united-atom charges from AMBER (Weiner, Kollman et al. 1984). The force field grid enclosed the entire trypsin molecule at a 0.3 Å spacing; it had a 10.0 Å cutoff for inclusion of partial charges in the potential at each point and used a dielectric of $4r$.

The Fine Chemicals Directory database (FCD, now called Available Chemicals Directory), supplied by MDL Information Systems, includes 3-dimensional structures for compounds which are commercially available. For this work, the 89.2 release of the database, which included approximately 50,000 3D structures created with Concord (Pearlman 1987), was used. Partial charges were assigned using the method of Gasteiger and Marsili (Gasteiger and Marsili 1980) for iterative partial equalization of electronegativity, as implemented in the SYBYL molecular modeling package (Tripos Associates). This method has the advantage of being very fast, even for relatively large molecules.

A test database for the trypsin system (Table 1) was generated by searching the FCD for substructures, and in some cases the names, of serine

protease inhibitors described in a review of the field (Powers and Harper 1986).

Structures from databases supplied by MDL Information Systems were converted to the DOCK 2 database formats using the program mol2shp, and to the DOCK 3 database format using mol2db3. In addition, mol2db3 was modified to count the number of nitrogen atoms in each molecule which were in environments where they would normally be positively charged. These nitrogen atoms included SYBYL atom types N.4, N.2 with three connected atoms, and amidinium and guanidinium groups. Three separate databases were constructed from those molecules in the FCD which had one,

Table 1: Compounds in the Trypsin Test Set

4-(Aminomethyl)-Cyclohexanecarboxylic Acid
Isatoic Anhydride
Alpha-Toluenesulfonyl Fluoride
6-Aminocaproic Acid
Diisopropylfluorophosphate
P-Aminomethylbenzoic Acid
DAPI Dihydrochloride
M-Nitrobenzamide Hydrochloride
3-Aminobenzamide Dihydrochloride
4-Aminobenzamide Dihydrochloride
4-Amidino Benzamide Hydrochloride
Benzamine Hydrochloride
4-Amidinobenzoic Acid
4-Methoxybenzamide
Ethyl 4-Amidinobenzoate
P-Amidinophenylmethylsulfonyl Fluoride
3,4-Dichloroisocoumarin
Antipain Dihydrochloride
P-Nitrophenyl-P-Guanidinobenzoate Hydrochloride
Leupeptin Hemisulfate
Antipain
P-Toluamide Hydrochloride
Benzamide
P-Amidinobenzamide Hydrochloride
4',6-Diamidino-2-Phenylindole
4-Chlorobenzamide Hydriodide
4,4'-Diamidinodiphenylamine Dihydrochloride

two, or three or more such nitrogens.

The molecules in the Fine Chemicals Directory were docked to trypsin using DOCK 2.1 (electrostatic scoring) and again using DOCK 2.2 (electrostatic scoring with solvation and conformation corrections). The databases of compounds from the FCD with one, two, or three or more positively charged nitrogens were docked to trypsin using DOCK 3.0 with force field scoring. The trypsin test database was docked to trypsin using the electrostatic scoring scheme in DOCK 2.2 and the force field scoring scheme in DOCK 3.0.

Initial rates of trypsin hydrolysis of Z-Gly-Pro-Arg-aminomethylcoumarin and IC₅₀ values were determined by Scott Willett and David Corey.

Docking to chymotrypsin

The *4cha* structure (Tsukada and Blow 1985) from the Brookhaven Protein Data Bank was used for chymotrypsin DOCK studies. This is a structure of uncomplexed chymotrypsin. It contains two molecules per asymmetric unit; molecule A was used. Water molecules were removed and a surface was created using MS (Connolly 1983) for all residues within 15 Å of serine 195. The clustering algorithm within SPHGEN produced a large number of spheres in the active site region, so the number was reduced by decreasing the maximum sphere radius, thus using only spheres which were derived from smaller indentations in the protein's surface. A total of 47 spheres were then chosen manually based on their proximity to the active site (Figure 3). DISTMAP and DelPhi were used to calculate a contact scoring grid and the receptor electrostatic potential using the same conditions as those for DOCK 2.2 in trypsin.

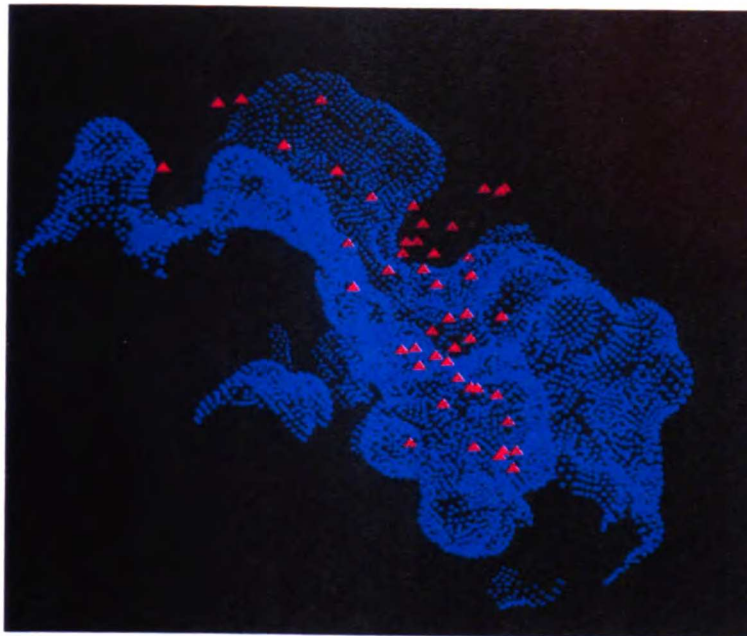


Figure 3: Sphere centers used for docking to chymotrypsin.

I was interested in comparing the chymotrypsin results to those obtained by Stewart *et al.* (Stewart, Fairley et al. 1992), who had done a docking study using contact scoring only. That study used biological data from the work of Wallace (Wallace, Kurtz et al. 1963). I therefore searched the Fine Chemicals Directory for compounds whose K_I was listed in the Wallace paper; the 80 compounds found (Table 2) were used as a test set for docking to chymotrypsin. K_I values for these compounds ranged from 0.063 μM to 200 μM . This test set was docked to chymotrypsin using DOCK 2.2 and DOCK 3.0.

Results and Discussion

Docking to trypsin

When the Fine Chemicals Directory was docked to trypsin using DelPhi-based electrostatic scoring alone (DOCK 2.1), many of the top-scoring compounds (Table 3) had multiple positive charges. For example, the best

Table 2: Compounds in the Chymotrypsin Test Set

Compound	K_I (μM)
Benzene	25
Toluene	13
Phenol	6.4
Cyclohexanol	75
Benzyl Alcohol	5.8
2-Phenylethanol	4.0
3-Phenyl-1-Propanol	4.3
Phenoxyethanol	7.7
Anisole	8.4
Aniline	6.6
N-Methylaniline	6.3
N,N-Dimethylaniline	3.4
N-Ethylaniline	6.6
Benzylamine Hydrochloride	22
Beta-Phenylethylamine	48
Formanilide	3.9
Acetanilide	13
Benzamide	10
Phenylacetamide	15
3-Phenylpropionamide	7
N-Benzylacetamide	7.5
N-Phenethylacetamide	11.4
Benzoic Acid	150
Phenylacetic Acid	200
Benzenesulfonamide	4.3
Sulfanilamide	15.4
Benzenesulfonic Acid	70
1-Naphthol	0.2
Biphenyl-4-ol	0.25
1-Naphthylamine	0.30
Beta-Naphthylamine	0.25
2-Naphthoic Acid	1.4
Diethyl-D-Tartrate	41
Pyridine	28
2,4,6-Collidine	10
2-Pyridol	110
2-Aminopyridine	9.4
3-Aminopyridine	12.3
4-Aminopyridine	2.9
Quinoline	0.6
Isoquinoline	0.32
Quinaldine	1.5

Lepidine	2.3
7-Methylquinoline	0.7
2-Quinolinol	0.87
8-Quinolinol	0.77
2-Aminoquinoline	1.3
3-Aminoquinoline	2.3
4-Quinolinecarboxylic Acid	104
8-Quinolinesulfonic Acid	177
Quinoline Ethyl Iodide	43
Quinoxaline	5
Phthalazone	2.95
4-Methyl-1-Pentene	1.88
1,3-Indandione	2.4
Ninhydrin	2.7
1-Methylindole	0.8
1-Methyl-2-Indolinone	0.87
7-Azaindole	1.33
Pyrazole	45
Benzimidazole	3
2-Benzimidazolemethanol	3.4
2-Benzimidazoleethanol	1.35
Acridine	0.22
Proflavine	0.13
Acridine	0.08
Rivanol	13.5
Phenanthridine	0.23
Benzo(F)Quinoline	0.063
7,8-Benzoquinoline	0.70
1,10-Phenanthroline	15.1
Phthalide	1.42
Coumarin	0.67
Fluorescein	10.2
Cresol Red	4.67
4-Phenylbutyric Acid	60
Naphthoresorcinol	1.4
Structure # 4418	1.84

score was assigned to *N,N*-diethyl diethylenetriamine, and 37 of the top 40 compounds were polyamines. Better scores seem to be assigned to compounds with larger numbers of positively charged groups, as might be expected from a scoring scheme based solely on charge complementarity. Approximately 20 of these compounds were tested for inhibition of bovine trypsin. There was no significant difference in the initial rates of hydrolysis of Z-Gly-Pro-Arg-AMC with any of the compounds present and the initial rates with no inhibitor present.

Running DOCK on trypsin and the Fine Chemicals Directory using DelPhi-based electrostatic scoring with solvation and conformation corrections (DOCK 2.2) produced top-scoring compounds with one or two positive charges (Table 4). Since there is a single negatively charged residue, Asp 189, in the S1 subsite of trypsin, these compounds are more likely to be good trypsin inhibitors than the top hits produced by DOCK 2.1.

Compounds of interest were selected from among the 200 top-scoring structures. 22 of these compounds were obtained and tested for inhibition of rat trypsin by David Corey and Scott Willett. 6 of these 21 showed inhibitory activity (Table 5) with IC_{50} values ranging from 10 μ M to 1100 μ M.

The structures in the Fine Chemicals Directory were grouped by the number of positive charges in each. Those with at least one positive charge were docked to the trypsin D102N structure using DOCK 3.0. The force field scoring scheme in DOCK 3.0 includes both van der Waals and electrostatic terms. In many cases, the top-scoring compounds were positioned in orientations which made many favorable van der Waals contacts with trypsin but did not have obvious charge-charge interactions. Compounds which had a positive charge oriented toward the negative charge in the trypsin binding pocket were chosen as candidates, but were not actually tested.

Table 3: Top-scoring Compounds From Docking the FCD to trypsin with DelPhi-based Scoring.

Compound	Rank	DOCK Score
N,N-Diethyl Diethylenetriamine	1	-42.132
N-(2-Aminoethylamino)Ethyl Pyrrolidine	2	-41.212
Tris(2-Aminoethyl)Amine	3	-41.092
N-(2-Aminoethylamino)Ethylmorpholine	4	-36.986
N-Methyldipropylenetriamine	5	-36.673
4-Dodecyldiethylenetriamine	6	-36.454
2-Amino-4-Azido-6-Methyl-S-Triazine	7	-36.207
1,4,7-Trimethyldiethylenetriamine	8	-34.934
2-(N-Hexamethyleneimino)Ethylamine	9	-34.227
3-Hexamethyleneimino-1-Propylamine	10	-33.157
(3-Aminopropyl)Iminodiethanol	11	-32.707
N,N-Di-N-Propyl-1,3-Propanediamine	12	-32.363
Aminopropylmorpholine	13	-32.092
N,N-Bis(2-Hydroxyethyl)Ethylenediamine Hydrochloride	14	-31.670
N,N-Bis(2-Hydroxyethyl)Ethylenediamine Dihydrochloride	15	-31.658
N,N-Bis(2-Hydroxyethyl)Ethylenediamine	16	-31.649
1-(3-Aminopropyl)-4-Methylpiperazine	17	-30.787
2-Di-N-Propylaminoethylamine	18	-30.479
N-Benzyl-N,N'-Dimethylethylenediamine	19	-30.299
N,N-Dibutyltrimethylenediamine	20	-29.491
N-Benzylethylenediamine	21	-29.432
2-(Ethyl-N-Butylamino)Ethylamine	22	-29.339
2-Di-N-Butylaminoethylamine	23	-29.124
3-Cyclohexylamino-1-Propylamine	24	-28.626
L-Ornithine 4-Methylcoumaryl-7-Amide Carbonate	25	-28.082
N,N-Diethyl-2-Butene-1,4-Diamine	26	-27.794
1-(3-Aminopropyl)-2-Pipecoline	27	-27.121
2-Diisopropylaminoethylamine	28	-26.946
N-Methyleneglycinonitrile trimer	29	-26.831
2-Diisobutylaminoethylamine	30	-26.819
4-Diisopropylaminobutylamine	31	-26.713
1-Piperidinepropylamine	32	-26.639
Decahydropyrazino(2,3-B)-Pyrazine	33	-26.468
2-Phenyl-1,4-Butanediamine Dihydrochloride	34	-26.317
4-Diethylaminobutylamine	35	-26.142
1,3,5-Triethylhexahydro-S-Triazine	36	-26.130
2-(Aminomethyl)Benzimidazole Dihydrochloride	37	-26.068
2-Aminomethylbenzimidazole Hydrochloride	38	-26.067
1,3-Cyclohexanebis(Methylamine)	39	-26.016
Dodecahydro-1,4,7,9b-Tetraazophenalene	40	-25.555

4-Azidoaniline Hydrochloride	41	-25.513
L-Lysine Methyl Ester Dihydrochloride	42	-25.292
N,N,N'-Triethylethylenediamine	43	-25.069
N,N-Diethyl-N,'n'-Dimethylethylenediamine	44	-24.721
2-(3-Chlorophenoxy)Ethylamine	45	-24.362
Alpha-Hydrazinoornithine Hydrochloride	46	-24.291
N,N-Dibutyl-1,4-Butanediamine	47	-24.288
5-Diisopropylaminoamylamine	48	-24.203
Alpha-Methylornithine Hydrochloride	49	-24.034
1,3-Cyclohexanebis(Methylamine)	50	-23.892
N,N-Diethyl-N'-Methyl-1,3-Diaminopropane	51	-23.856
2-Aminomethyl-N-Methylbenzylamine	52	-23.671
1-Amino-3-Diethylamino-2-Propanol	53	-23.642
N-1-Naphthylethylenediamine Dihydrochloride	54	-23.429
N,N'-Diethyl-3-Aminopyrrolidine	55	-23.414
N,N'-Diethyl-3-Aminopyrrolidine Dihydrochloride	56	-23.413
2-Aminoethyl Benzoate Hydrobromide	57	-23.323
L-Lysinamide Dihydrochloride	58	-23.288
N-(3-Aminopropyl)-2-Butene-1,4-Diamine	59	-23.191
Cyclohexanebutylamine	60	-23.150
N-Alpha-Methyl-L-Lysine	61	-23.055
1,4,7 Triethyl-diethylenetriamine	62	-22.957
N-6-(6-Aminohexyl)Adenosine 2',5'-Diphosphate Lithium Salt	63	-22.816
1,3-Bis-(Dimethylamino)Butane	64	-22.808
N-(2-Aminoethyl)Benzamide	65	-22.675
Azidomethyl Phenyl Sulfide/Phenylthiomethyl Azide	66	-22.590
4-(3-Aminopropyl)-2-Pyrazolin-5-One	67	-22.399
1-Benzyl-3-Aminopyrrolidine	68	-22.295
Mescaline Sulfate	69	-22.255
Mescaline Hydrochloride	70	-22.251
Mescaline Hemisulfate/3,4,5- Trimethoxyphenethylamine	71	-22.251
2-(p-Iodophenyl)Ethylamine	72	-22.130
5-Morpholinoamylamine	73	-22.118
1,4-Bis(Aminomethyl)Cyclohexane	74	-22.097
3,5-Dichlorobenzylamine	75	-22.064
3,5-Difluorobenzylamine	76	-21.988
3-Hydroxytyramine Hydrochloride	77	-21.932
3-Chloro-2-Methylbenzylamine/2-Methyl-3- Chlorobenzylamine	78	-21.928
N-(2-Hydroxy-3-(1-Naphthoxy)Propyl)Ethylendiamine Dihydrochloride/Nhnp-E	79	-21.927
3-Chlorophenethylamine	80	-21.874
2,3-Dimethyl Benzylamine	81	-21.802
M-Hydroxybenzylhydrazine Dihydrochloride	82	-21.749

Diaminobiotin Dihydrochloride/Cis-3,4-Diamino-2-Tetrahydrothiophenevaleri	83	-21.724
3,5-Dimethylbenzylamine	84	-21.681
L-Arginylglycine	85	-21.622
1-(2-Aminoethyl)-2-Methyl-5-Nitroimidazole Dihydrochloride Monohydrate	86	-21.596
3-Ethoxy-4-Hydroxyphenethylamine	87	-21.551
N-Alpha-Acetyl-L-Lysine-N-Methylamide Monohydrate	88	-21.546
4-Methoxyphenylethylamine Hydrochloride	89	-21.517
4-Bromophenethylamine Hydrochloride	90	-21.474
4-(2-Methyl Aminoethyl)Pyridine	91	-21.461
N-(Morpholino Ethyl) Ethylene Imine	92	-21.448
2-(2,6-Dichlorobenzylthio)Ethylamine	93	-21.273
Guanethidine Sulfate	94	-21.255
N-Ethyl-M-Methylbenzylamine	95	-21.216
2-Methyl-5-Hydroxytryptamine Maleate	96	-21.146
2-(2-Chloro-6-Fluorobenzylthio)Ethylamine	97	-21.068
6-Hydroxydopamine Hydrobromide	98	-20.919
2-(4-Methoxyphenoxy)Ethylamine	99	-20.899
1-Methyl-3-Phenylpropylamine Hydrochloride	100	-20.845

Table 4: Top-scoring Compounds From Docking the FCD to trypsin with DelPhi-based Scoring plus solvation correction.

Compound	Rank	DOCK Score
2-Phenyl-2-Imidazoline	1	-37.1261
Tolazoline	2	-29.9266
1,8-Diazabicyclo[5.4.0]Undec-7-Ene	3	-26.1531
1,2,3,4-Tetrahydro-9H-Pyrido[3,4-B]Indole	4	-24.8172
N-(1-Indanyl)Propargylamine	5	-24.7186
2-Methyl-4-Chlorobenzylamine	6	-24.5661
(-)-Cyclohexylisopropylmethylamine	7	-24.4091
N-Methylcyclodecylamine	8	-24.2787
2-Phenylglycinonitrile Hydrochloride	9	-23.8825
DL-Alanyl-Beta-Naphthylamide Hydrochloride	10	-23.7720
(R)-(-)-Amphetamine Sulfate	11	-23.7587
N-Methylcyclooctylamine	12	-23.6854
1-Phenyl-1-Cyclopropanemethylamine Hydrochloride	13	-23.4648
4-Amino-1,2-Diethylpyrazolidine	14	-23.3203
(+/-)Deoxyephedrine-D5 Hydrochloride	15	-23.1931
Quinoline Ethyl Sulfate	16	-23.1514
2-Phenylglycinonitrile Hydrochloride	17	-23.0765
1-(<i>m</i> -Tolyl)Piperazine Dihydrochloride	18	-22.9876
(+)-Methamphetamine Hydrochloride	19	-22.7589
1-Amino-2-Hydroxyindane	20	-22.7145
THIP Hydrochloride	21	-22.6641
1-(2-Morpholinoethyl)-2-Thiourea	22	-22.6611
DL-1-Phenylpropylamine	23	-22.6340
2-Amino-3-Aminomethyl-6-Methylpyridine Dihydrochloride	24	-22.6141
1,2,3,4-Tetrahydro-6,7-Isoquinolinediol Hydrobromide	25	-22.2637
1-(3-Methoxyphenyl)Piperazine Dihydrochloride	26	-22.0062
4-Amino-5-Aminomethyl-2-Methylpyrimidine Dihydrochloride	27	-21.7357
4-Amino-5-Aminomethyl-2,6-Dimethylpyrimidine Dihydrochloride	28	-21.6053
Trans-Decahydroquinoline Hydrobromide	29	-21.5697
1-Naphthalenemethylamine	30	-21.5562
1-Ethylquinolinium Iodide	31	-21.5546
L-Amphetamine Free Base	32	-21.5477
1,2,3,4-Tetrahydro-1-Naphthylamine Hydrochloride	33	-21.3837
1-(3-Toluidine)Piperazine	34	-21.3502
Alpha-Amino- <i>p</i> -Tolunitrile	35	-21.3451
1,2,3,4-Tetrahydro-2-Naphthylamine	36	-21.3174
2-Guanylbenzimidazole	37	-21.2924
Allylcyclohexylamine	38	-21.2862

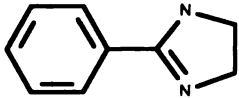
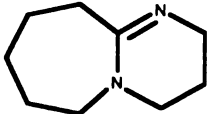
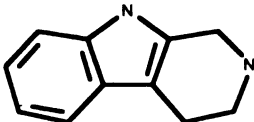
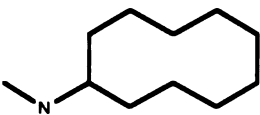
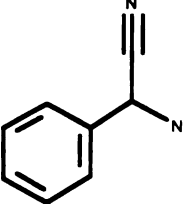
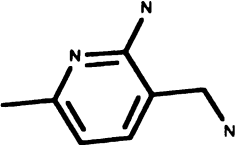
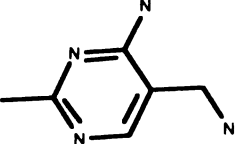
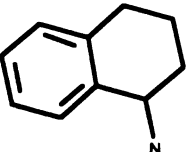
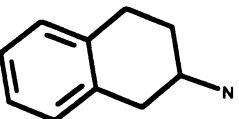
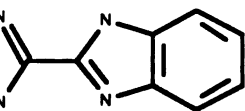
N-Cyclohexylethanolamine	39	-21.1084
2,5-Dichlorobenzylamine	40	-20.9995
Tetrahydropterine Sulfate	41	-20.9972
1,2,3,4-Tetrahydro-1-Naphthylamine	42	-20.9107
1,2,3,4-Tetrahydroisoquinoline Hydrochloride	43	-20.8213
Decahydroquinoline	44	-20.8047
1-(3,4-Xylyl)Piperazine	45	-20.7985
3,5-Dimethyl-1-Ethylpyridinium Iodide	46	-20.7474
Glycylglycine Ethyl Ester Hydrochloride	47	-20.6525
DL-Beta-Hydroxyphenethylamine Hydrochloride	48	-20.5344
4-Chlorobenzamidine Hydroiodide	49	-20.4341
2-Amino-4'-Phenylacetophenone Hydrobromide	50	-20.4125
3-Ethylbenzothiazolium Bromide	51	-20.3765
3,5-Dimethyl-1-Propylpyridinium Iodide	52	-20.3665
2-Ethylisoquinolinium Iodide	53	-20.3572
3,3,5-Trimethylcyclohexylamine	54	-20.2892
N-Omega-Methyltryptamine Oxalate	55	-20.2753
6,7-Dimethoxy-1,2,3,4-Tetrahydroisoquinoline Hydrochloride	56	-20.2645
<i>p</i> -Methoxyamphetamine Hydrochloride	57	-20.2073
1-(3-Methoxyphenyl)Piperazine	58	-20.1933
N-Alpha-Methylhistamine Dihydrochloride	59	-20.1904
1,2,3,4-Tetrahydroisoquinoline	60	-20.1726
<i>p</i> -Chloro-N-Methylbenzylamine	61	-20.1425
L-Alanyl-L-Alanyl-L-Alanine Methyl Ester Acetate	62	-20.0507
2-(Aminomethyl)Benzimidazole Dihydrochloride Hydrate	63	-19.9751
DL-Alpha-Methylamino-Epsilon-Caprolactam	64	-19.8713
Alpha-Methyl-Beta-(3-Methoxy-4-Hydroxyphenyl)Ethylamine Hydrochloride	65	-19.8107
5-Benzyloxytryptamine Hydrochloride	66	-19.7973
Decahydroisoquinoline	67	-19.7956
Alpha-Methyltryptamine Methanesulfonate	68	-19.7365
4-Aminobenzamidine Dihydrochloride	69	-19.7099
N-(3-Hydroxyphenyl)Piperazine	70	-19.6651
1-(3,4-Methylenedioxyphenyl)Piperazine Monohydrochloride	71	-19.6551
2-Ethyl-5-Hydroxyisoquinolinium Bromide	72	-19.5817
Phenelzine Sulfate Salt	73	-19.5297
2,4-Dimethyl-1-Ethylpyridinium Iodide	74	-19.5209
2-Amino-1-Phenylethanol	75	-19.4524
Quipazine Maleate	76	-19.3819
3,5-Dimethylbenzylamine	77	-19.3819
2,5-Dimethylbenzylamine	78	-19.3578
Eleagnine	79	-19.3059
L-Norephedrine Sulfate	80	-19.2617

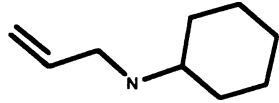
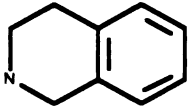
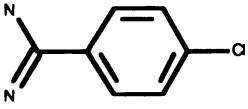
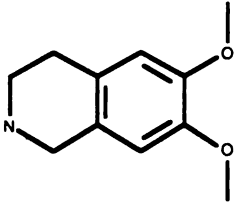
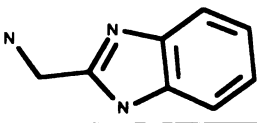
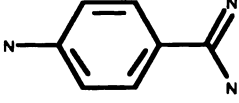
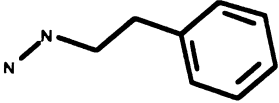
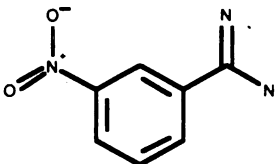
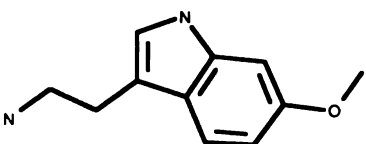
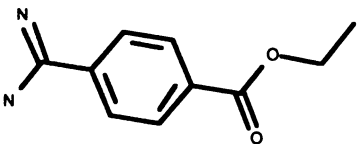
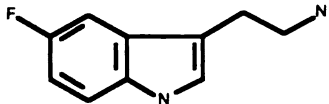
Norephedrine Hydrogen Phosphate	81	-19.2617
1-(3,5-Dimethoxyphenyl)Piperazine	82	-19.2555
5-Fluoro-Alpha-Methyltryptamine Hydrochloride	83	-19.2180
5-Benzyloxy-N,N-Dimethyltryptamine Oxalate	84	-19.1900
2-Aminomethylbenzimidazole Hydrochloride	85	-19.1719
1-Methyl-3-Phenylpropylamine Hydrochloride	86	-19.1705
3-Nitrobenzamidine Hydrochloride	87	-19.1606
Alpha-Methyl-5-Hydroxytryptamine Maleate	88	-19.0941
<i>p</i> -Aminobenzylmethylamine	89	-19.0494
4-Amino-2,6-Dimethylheptane	90	-19.0172
Serotonin Creatinine Sulfate Monohydrate	91	-18.9917
5-Benzyloxytryptamine	92	-18.9782
1-(4-Bromophenyl)Ethylamine	93	-18.9738
5-Hydroxytryptamine Bimaleate	94	-18.9698
3,5-Difluorobenzylamine	95	-18.9450
Thiochroman-4-Amine Hydrochloride	96	-18.9408
Serotonin Hydrogen Oxalate	97	-18.8870
1-(5-Isoquinolinylsulfonyl)-2-Methylpiperazine	98	-18.8852
N-Phenyltrimethylenediamine	99	-18.8201
D-(-)-Phenylglycinol	100	-18.7934
N-Methylphenethylamine	101	-18.7928
D-(-)-Alpha-Phenylglycinol	102	-18.7885
4-Chloro-Alpha-Methylbenzylamine	103	-18.7822
Serotonin Maleate	104	-18.7535
7-Acetoxy-N-Methylquinolinium Iodide	105	-18.7414
2-Phenoxyethylamine	106	-18.7374
3-Amino Nonane	107	-18.7039
2-(4-Chlorophenoxy) Ethylamine	108	-18.6475
Hordenine Hemisulfate Salt	109	-18.6470
2-(Dimethylamino)Isopropyl Acetate	110	-18.6127
P-Toluamidine Hydrochloride, Hydrate	111	-18.6080
3-Methoxytyramine Hydrochloride	112	-18.5335
2-Nonylamine	113	-18.4977
(S)-(-)-N,N-Dimethyl-1-Phenethylamine	114	-18.4506
2,3-Dimethyl Benzylamine	115	-18.4384
N-(2-Aminoethyl)Benzamide	116	-18.4352
2-(<i>P</i> -Diethylaminostyryl)-Pyridylmethyl Iodide	117	-18.4327
2,4-Dimethylbenzylamine	118	-18.4219
N,N-Diethylbenzylamine	119	-18.4147
1-(4-Tolyl)Piperazine Dihydrochloride	120	-18.4025
4-(3-Aminopropyl)-2-Pyrazolin-5-one	121	-18.3848
Omega-Aminoacetophenone Hydrochloride	122	-18.3738
3-Amino-1-Phenylbutane	123	-18.3653
2-Aminoindan Hydrochloride	124	-18.3457
2,4-Difluorobenzylamine	125	-18.3448
1-Aminoindane Hydrochloride	126	-18.3329

Ethyl Isonipecotate	127	-18.3238
2-Chloro-4-Fluorobenzylamine	128	-18.3024
3-Methoxycarbonyl-1-Methylpyridinium Iodide	129	-18.2655
1-Methyl-2-(4-Methylstyryl)Pyridinium Iodide	130	-18.2465
1-Allyl-2-(2-[3-Indolyl]-Vinyl)-Pyridinium Iodide	131	-18.2392
Norharman Methiodide	132	-18.2230
6-Methoxytryptamine	133	-18.1890
2-(4-Methoxyphenoxy) Ethylamine	134	-18.1874
Hordenine Hydrochloride	135	-18.1728
Trans-2-Phenylcyclopropylamine Hydrochloride	136	-18.1595
2-(2-Methoxyphenyl)Ethylamine	137	-18.1305
1-Methyl-3-Styrylpyridinium Iodide	138	-18.1246
5,6-Dihydroxytryptamine Creatinine Sulfate Salt	139	-18.0600
(S)-(-)-1-Amino-2-(Tert-Butyldimethylsiloxymethyl)-Pyrrolidine	140	-18.0414
Norephedrine Hydrochloride	141	-18.0213
Hexamethyleneiminoacetonitrile	142	-17.9801
N,N-Dimethyl-2-(4-Hydroxyphenyl)Ethylamine	143	-17.9755
N-(2-Aminoethyl)- <i>p</i> -Hydroxybenzamide Hydrochloride	144	-17.9698
N-N-Propylisoindoline Hydrochloride	145	-17.9494
3-Dimethylamino-2-Methylpropiophenone Hydrochloride	146	-17.9417
4-Isopropylbenzylamine	147	-17.8997
1-(<i>p</i> -Methoxyphenyl)-3-Butylamine	148	-17.8937
3-Amino-1-Phenylpropanol	149	-17.8916
3-Amino-5-Tert-Butylisoxazole	150	-17.8772
Ethyl 4-Amidinobenzoate	151	-17.8651
N-(6-Aminohexyl)-1-Naphthalenesulfonamide Hydrochloride	152	-17.8373
Primaquine Diphosphate	153	-17.8195
4-Methoxybenzylamine	154	-17.8135
5,7-Dihydroxytryptamine Creatinine Sulfate	155	-17.8117
2-(4-Ethoxystyryl)-1-Methylpyridinium Iodide	156	-17.8112
5-Chlorotryptamine Hydrochloride	157	-17.8031
7-Hydroxy-N-Methylquinolinium Iodide	158	-17.7954
Beta-Methylphenethylamine	159	-17.7619
2-(4-Benzyloxystyryl)-1-Ethylpyridinium Iodide	160	-17.7231
4-Hydroxy-3-Methoxyphenethylamine	161	-17.7004
Benzo(B)Furan-2-Methylamine	162	-17.6676
1-Aminoindan	163	-17.6619
Methoxyphenamine Hydrochloride	164	-17.6617
(+)-Alpha-(2-Naphthyl)Ethylamine	165	-17.6524
4-Amino-Alpha-Diethylamino-Ortho-Cresol Dihydrochloride	166	-17.6524
3,4-Dimethyl-1-Propylpyridinium Iodide	167	-17.6253
6-Methyl-2-Picolylmethylamine	168	-17.6062

Ethyl-4-Methyl-1-Piperazinecarboxylate	169	-17.6056
1-Methylquinolinium Iodide	170	-17.5917
Quinoline Methyl Sulfate	171	-17.5907
(+/-)-2-Amino-5,6-Dihydroxy-1,2,3,4-Tetrahydronaphthalene Hydrobromide	172	-17.5841
5-Methyltryptamine Hydrochloride	173	-17.5808
2-Amino-5-Methyloctane	174	-17.5619
Tryptamine Hydrochloride	175	-17.5589
2-(4-Isopropylstyryl)Pyridine Methiodide	176	-17.5476
DL-Alpha-Methyltryptamine	177	-17.4877
1-(4-Benzoyloxyphenyl)-2-Propylamine Hydrobromide	178	-17.4449
N-Acrylyl-1,6-Diaminohexane Hydrochloride	179	-17.4144
4-Amino-Alpha-Diethylamino-2-Cresol Dihydrochloride	180	-17.4013
P-Chloro-(2-Dimethylaminoethyl)Benzylpyridine Maleate	181	-17.3962
4-(3-Phenylpropyl)Piperidine	182	-17.3676
2,2-Dimethyl-5-Dimethylamino-3-Pentanone Hydrochloride	183	-17.3671
2,4-Dichloro-6-Methylbenzylamine	184	-17.3617
Uramil	185	-17.3373
2-Aminoethyl Benzoate Hydrobromide	186	-17.3339
2-(2-Dimethylaminoethyl)Pyridine	187	-17.3197
1-Methylpyridinium 3-Sulfonate	188	-17.3131
5-Fluorotryptamine Hydrochloride	189	-17.3034
1-(4-Methoxyphenyl)Piperazine Dihydrochloride	190	-17.2826
Beta-Alanine Benzyl Ester <i>p</i> -Tosylate	191	-17.2658
N-(3-Piperidyl)Pyrrole	192	-17.2593
Methyl Beta-Keto-Alpha-Amino adipate	193	-17.2452
2,4-Dichloro-N-Methylbenzylamine	194	-17.2265
Glycine Benzyl Ester <i>p</i> -Toluenesulfonate	195	-17.2190
N-(<i>p</i> -Methoxyphenyl)Piperazine Succinate	196	-17.2141
4-Methoxycarbonyl-1-Methylpyridinium Iodide	197	-17.1882
4-Fluoro-Alpha-Methylbenzylamine	198	-17.1673
4-Amino-Alpha-Diethylamino- <i>o</i> -Cresol	199	-17.1574
(1R,2S)-(-)-Norephedrine	200	-17.1559

Table 5: Results of Testing Compounds Selected From Docking the FCD to Trypsin Using DelPhi-based Scoring with Solvation Correction.

Compound	DOCK Score	DOCK Rank	IC ₅₀ * (μM)
	-37.1	1	NI
	-26.2	3	NI
	-24.8	4	NI
	-24.3	8	NI
	-23.9	9	NI
	-22.6	24	609
	-21.7	27	NI
	-21.4	33	NI
	-21.3	36	NI
	-21.3	37	NI

	-21.3	38	NI
	-20.8	43	NI
	-20.4	49	NI
	-20.3	56	NI
	-20.0	63	NI
	-19.7	68	15
	-19.5	73	NI
	-19.2	87	10
	-18.2	133	1100
	-17.9	151	NI
	-17.3	189	597

* NI — no inhibition

The trypsin test database (Table 1), composed of compounds which were trypsin inhibitors or shared common substructures with trypsin inhibitors, was docked to trypsin using the electrostatic scoring scheme with solvation and conformation corrections of DOCK 2.2. The top 10 compounds were all benzamidines, with their positively charged amidinium groups oriented toward the arginine in the binding pocket. The eleventh through eighteenth compounds either were uncharged or had positive charges oriented away from the arginine in their best-scoring orientations. The remaining compounds, ranked nineteenth through twenty-first, had more than one positive charge and may have scored poorly because of the large solvation correction for multiply charged compounds. The same test set was docked using DOCK 3.0. In that case, the top 6 compounds had amidinium groups oriented toward the negative charge in the binding pocket, but three of these six were the large multiply-charged compounds which scored worst in the DOCK 2.2 run.

Docking to chymotrypsin

When the compounds in the chymotrypsin test set were docked to chymotrypsin using DOCK 2.2 (Figure 4), there was no apparent correlation between DOCK score and K_I . Subtracting the solvation correction to give scores equal to those produced by DOCK 2.1 gave no correlation either (Figure 5). DOCK 3.0 produced a general trend toward better scores for compounds with lower K_I values (Figure 6).

Conclusions

Use of DOCK with DelPhi-based scoring in trypsin has demonstrated that a scoring schemes based only on electrostatic complementarity, when used in this charged system, retrieves the most highly charged molecules in the test database. These may not be realistic candidates as ligands for a charged

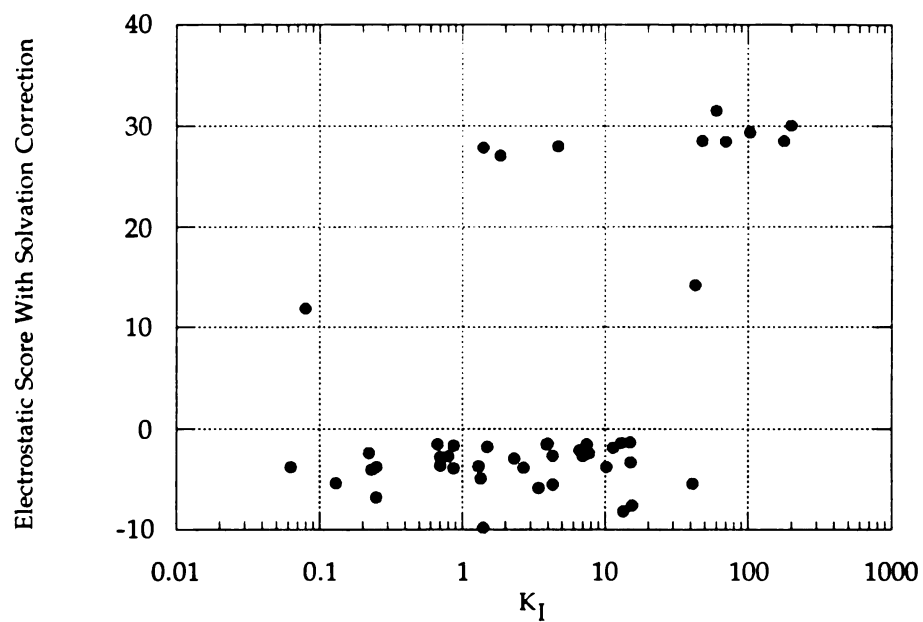


Figure 4: Scores from docking the chymotrypsin test set to chymotrypsin, plotted against K_I values.

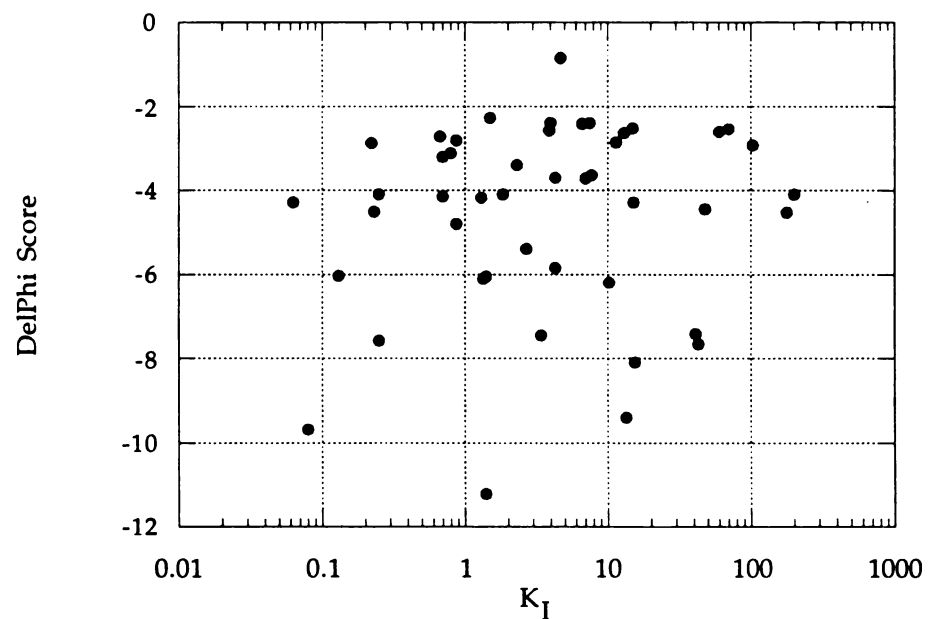


Figure 5: DelPhi electrostatic scores without solvation correction vs. K_I values for the chymotrypsin test set.

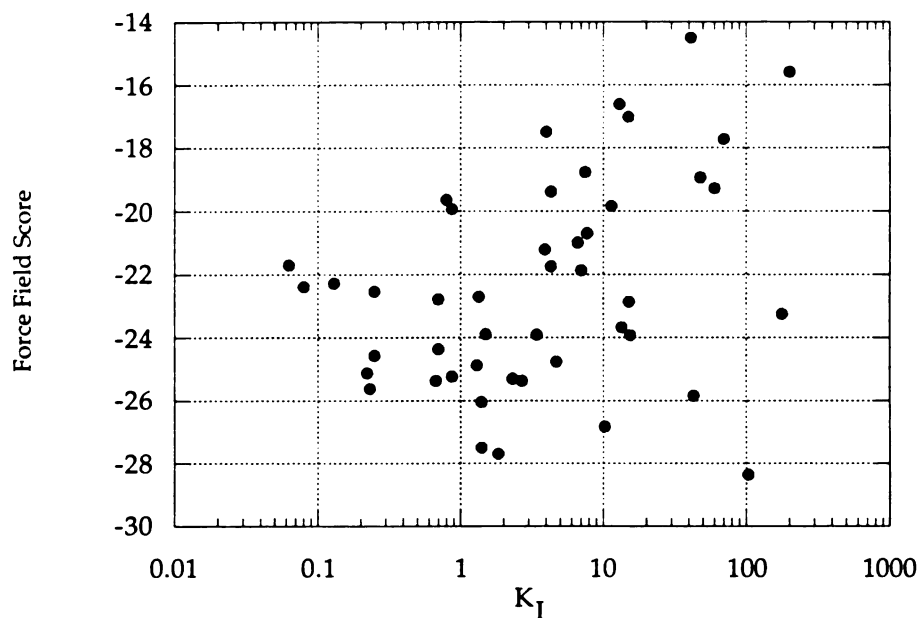


Figure 6: DOCK 3.0 (force field) scores vs. K_I values for the chymotrypsin test set.

receptor, since the energy cost of desolvating multiple charges often cannot be offset by the interaction of ligand and receptor charges. Therefore, when electrostatic scoring schemes are used alone to evaluate docked orientations it is necessary to correct for solvation.

Solvation correction, in conjunction with an electrostatic scoring scheme, has been shown to be useful in trypsin, a charged system. However, in chymotrypsin, with its uncharged binding pocket, electrostatic scoring performed poorly and, as might be expected, a solvation correction did not make a difference. The DOCK 3.0 scoring method was best for this hydrophobic system.

References

Available Chemicals Directory. San Leandro, CA, MDL Information Systems, Inc.

SYBYL Molecular Modeling Software. St. Louis, MO, Tripos, Inc.

Abola, E. E., F. C. Bernstein, et al. (1987). Protein Data Bank.

Crystallographic Databases — Information Content, Software Systems,

Scientific Applications. F. H. Allen, G. Bergerhoff and R. Sievers.

Bonn/Cambridge/Chester, Data Commission of the International Union of Crystallography: 107-132.

Bernstein, F. C., T. F. Koetzle, et al. (1977). "The Protein Data Bank: A Computer-Based Archival File for Macromolecular Structures." Journal of Molecular Biology **112**: 535-542.

Connolly, M. L. (1983). "Solvent-Accessible Surfaces of Proteins and Nucleic Acids." Science **221**(4612): 709-713.

DesJarlais, R. L., R. P. Sheridan, et al. (1988). "Using Shape Complementarity as an Initial Screen in Designing Ligands for a Receptor Binding Site of Known Three-Dimensional Structure." Journal of Medicinal Chemistry **31**(4): 722-729.

Gasteiger, J. and M. Marsili (1980). "Iterative Partial Equalization of Orbital Electronegativity — A Rapid Access to Atomic Charges." Tetrahedron **36**: 3219-3288.

Gilson, M. L., K. A. Sharp, et al. (1987). "Calculating the Electrostatic Potential of Molecules in Solution: Method and Error Assessment." Journal of Computational Chemistry **9**(4): 327-335.

Kuntz, I. D., J. M. Blaney, et al. (1982). "A Geometric Approach to Macromolecule-Ligand Interactions." Journal of Molecular Biology **161**: 269-288.

Leach, A. R. and K. Prout (1990). "Automated Conformational Analysis: Directed Conformational Search Using the A* Algorithm." Journal of Computational Chemistry **11**(10): 1193-1205.

Meng, E. C., B. K. Shoichet, et al. (1992). "Automated Docking with Grid-Based Energy Evaluation." Journal of Computational Chemistry **13**(4): 505-524.

Pearlman, R. S. (1987). "Rapid Generation of High-Quality Approximate 3-D Molecular Structures." Chemical Design Automation News **2**(1): 5-6.

Powers, J. C. and J. W. Harper (1986). Inhibitors of Serine Proteinases. Inhibitors of Serine Proteinases. A. J. Barrett and G. Salvesen. New York, Elsevier: 55-152.

Rashin, A. A. and K. Namboodiri (1987). "A Simple Method for the Calculation of Hydration Enthalpies of Polar Molecules with Arbitrary Shapes." Journal of Physical Chemistry **91**: 6003-6012.

Ruhlmann, A., D. Kukla, et al. (1973). "Structure of the Complex Formed by Bovine Trypsin and Bovine Pancreatic Trypsin Inhibitor. Crystal Structure Determination and Stereochemistry of the Contact Region." Journal of Molecular Biology **77**(3): 417-436.

Shoichet, B. K., D. L. Bodian, et al. (1992). "Molecular Docking Using Shape Descriptors." Journal of Computational Chemistry **13**(3): 380-397.

Sprang, S., T. Standing, et al. (1987). "The Three-Dimensional Structure of Asn¹⁰² Mutant of Trypsin: Role of Asp¹⁰² in Serine Protease Catalysis." Science **237**: 905-909.

Stewart, K. D., T. A. Fairley, et al. (1992). "Automated 3D Docking: Inhibitors of α -Chymotrypsin." Medicinal Chemistry Research **1**: 439.

Sweet, R. M., H. T. Wright, et al. (1974). "Crystal Structure of the

Complex of Porcine Trypsin with Soybean Trypsin Inhibitor (Kunitz) at 2.6-Å Resolution." Biochemistry **13**(20): 4212-28.

Tsukada, H. and D. M. Blow (1985). "Structure of Alpha-Chymotrypsin Refined at 1.68 Å Resolution." Journal of Molecular Biology **184**(4): 703-11.

Wallace, R. A., A. N. Kurtz, et al. (1963). "Interaction of Aromatic Compounds With α -Chymotrypsin." Biochemistry **2**(4): 824-836.

Weiner, S. J., P. A. Kollman, et al. (1984). "A New Force Field for Molecular Mechanical Simulation of Nucleic Acids and Proteins." Journal of the American Chemical Society **106**: 765-784.

Chapter 2: Docking to Positive Images of Receptor Sites

Introduction

Because the number of available protein structures has increased rapidly in recent years, programs which use these structures to facilitate the design of drugs have become popular. While such programs are useful in cases where the target for drug design has a known 3-dimensional structure, many more systems of interest involve targets whose structure has not yet been solved. When no receptor structure is available, researchers must obtain information about the target from substances whose activity is known. These substances may be known drugs and pharmacophore models derived from them, small proteins and peptides whose structures are known or predicted, or regions of larger proteins whose receptor-binding domains are somewhat understood. Their structures provide a positive image which includes some of the properties and the geometry of the drug being sought.

DOCK (Kuntz, Blaney et al. 1982; DesJarlais, Sheridan et al. 1988) has been useful in finding novel drug leads when a receptor structure is available to serve as a negative image of the drug sought. In this work I set out to investigate whether it could also serve as a tool for finding new leads based on positive images. DOCK can rapidly generate large numbers of orientations of a small molecule relative to a target image. Important features — for example, steric and electrostatic properties — of the small molecule can be compared to those of the target, and scores may be assigned based on their similarity. These scores allow orientations and molecules to be compared with each other. When large databases of molecular structures such as the Available Chemicals Directory are docked in this way, the top-scoring molecules may be promising candidates for testing as new drugs.

The use of positive-image information in drug design has historically included the construction of pharmacophores from known drugs and the use of 3D database searching to look for molecules containing them. MOLPAT (Gund, Wipke et al. 1974), the first program designed to look for pharmacophores, searched individual molecules for selected distances between atom types. More recent efforts have produced programs which are fast enough for searching large numbers of molecules, but still use the same basic approach. For example, Jakes and Willett (Jakes and Willett 1986) used a search based on distance screens, which allow molecules lacking the desired atom pair-distance combinations to be rapidly eliminated. 3DSEARCH (Sheridan, Nilakantan et al. 1989; Sheridan, Rusinko et al. 1989) and MACCS-3D (Christie, Henry et al. 1990) also use screens to eliminate uninteresting molecules; these programs can search for more complex geometric relationships and for user-specified substructures. ALADDIN (Van Drie, Weininger et al. 1989) includes a more detailed definition of query substructures. Commercially available 3D search programs include CHEMDBS-3D (produced by Chemical Design), UNITY (Tripos Associates) and CATALYST (Biocad).

Three-dimensional searching can be very rapid, but it is most appropriate for problems where the relative importance of different functional groups is well understood and the sets of distances among them are small and well-defined. With FOUNDATION, Ho and Marshall (Ho and Marshall 1993) addressed the all-or-nothing nature of 3D search queries by requiring that only a user-specified minimum number of elements of the query must be matched. This approach should be useful in cases where a pharmacophore is not well defined, but still depends on the relative locations of functional groups. CAVEAT (Lauri and Bartlett 1994) searches for bond

vectors arranged in specified orientations, with the idea that the bonds found can be replaced with selected functional groups. The search hits will then serve as scaffolds to deliver the desired functionality. This method retrieves molecules which are starting points for synthesis, rather than candidates for direct testing like those found by 3D searching and DOCK.

Instead of examining selected groups and internal distances, the search method SPERM (van Geerestein, Perry et al. 1990; Perry and van Geerestein 1992) compares the overall shapes of database molecules to a target. The centers of mass of the molecules are overlaid, and the difference in distance to the surface of each molecule along a set of approximately equally spaced directions is compared. The database molecule is rotated to optimize surface overlap. The method has been made relatively fast, but it does not address translational degrees of freedom and is most suited to small molecules of similar size. DOCK, by contrast, generates orientations which vary both in translation and rotation and is appropriate for comparing small molecules to regions of much larger models.

Three-dimensional searching methods are useful for rapidly locating small sets of distances among well-defined groups, while SPERM is appropriate for comparing the shapes of small targets to databases of similarly-sized molecules. Unlike three-dimensional searching, DOCK does not impose strict requirements on relative functional group positions. DOCK also evaluates many orientations of a given molecule instead of giving a single answer based on one set of distances. Within DOCK, several aspects of a candidate molecule may be evaluated for each orientation. A method based on DOCK could offer advantages, especially in cases where the target is complex or poorly defined or where mimics are sought for a portion of a larger molecule.

In this work, I investigated the utility of three types of scoring methods for identifying compounds similar to a positive image. Since electrostatic interactions are important to the specificity of ligand-receptor binding, one of the methods chosen was based on electrostatics. The other two methods measured different aspects of the geometric resemblance between the target image and the candidate molecules, which is of interest because steric fit of a ligand to its receptor is important in binding. One of them, an atom-based method, measured geometric similarity using the positions of atoms in the candidate ligand relative to those in the target positive image, while the other, a shape-based method, used the overlap of ligand surface with target surface as a test of their shape similarity. Since there was no obvious way to combine these scores into a single measure of fit, they were primarily evaluated individually before combining them was considered. Initial testing of the electrostatic scoring method and the atom-based geometric scoring method was conducted using the dopamine D2 receptor pharmacophore proposed by Manallack and Beart (Manallack and Beart 1988). This system was chosen because it is well-characterized and both electrostatic and steric interactions appear to be important in receptor binding. The use of DOCK as a method for aligning related molecules and deriving a pharmacophore was also investigated in the dopamine D2 pharmacophore. Since X-ray crystal structures are available for several protein inhibitors of trypsin and electrostatic and steric interactions are important to their binding, a trypsin inhibitor was a good test case for docking to positive images derived from proteins. Accordingly, further studies of the electrostatic and atom-based geometric scoring schemes were carried out using a positive image based on bovine pancreatic trypsin inhibitor in a structure determined as a complex with trypsin.

Methods

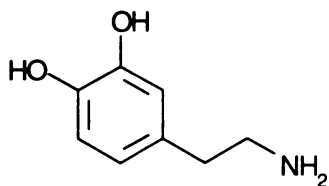
Dopamine D2 Pharmacophore: Positive Docking

Representation of Pharmacophore Atoms

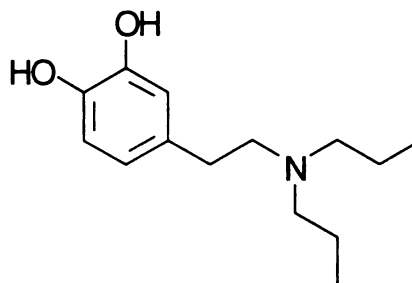
Computer models of seventeen of the compounds described in Manallack and Beart (Manallack and Beart 1988) were built following the descriptions in that paper (Figure 1). Skeletons for similar compounds were located in the Cambridge Structural Database (Allen, Kennard et al. 1973), and additional groups were added using the standard bond lengths and angles in SYBYL. Charges were calculated using the method of Gasteiger and Marsili (Gasteiger and Marsili 1980) as implemented in the SYBYL molecular modeling package. Three of the four molecules used by Manallack and Beart to define the pharmacophore (Figure 2) were then aligned to the fourth, pergolide, by matching the five points used in the original paper: the basic nitrogen atoms, points 2.8 Å from the basic nitrogen atoms in the direction of their lone pairs, the aromatic ring centroids, and the points at the end of a line perpendicular to the plane of the aromatic ring and extending 3.5 Å from the ring in both directions. After an initial alignment by hand, least-squares fitting was used to improve the alignment (Figure 3).

"Spheres", the centers which would be matched to ligand atoms in DOCK, were derived from the basic nitrogen of each molecule and its associated propyl group along with the aromatic ring, its centroid, and (where present) the attached hydroxyl group. The positions of these atoms in the aligned molecules were averaged to give the positions of the spheres (Figure 4). So that the pharmacophore would match five- or six-membered rings, the ring was represented by seven spheres, with the extra sphere located between two of the spheres in the six-membered ring.

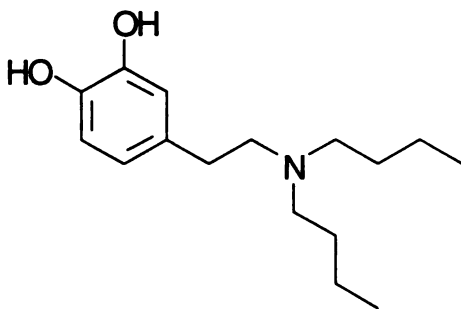
Figure 1: Molecules used in testing positive docking methods in the dopamine D2 pharmacophore.



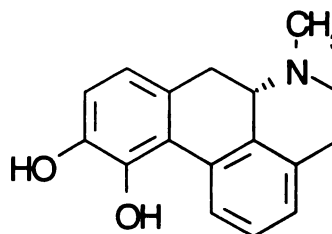
Dopamine



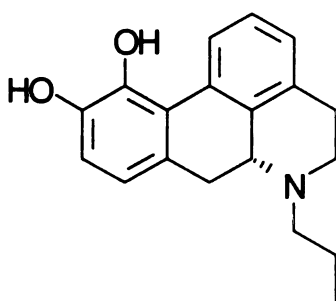
N,N-Dipropyldopamine



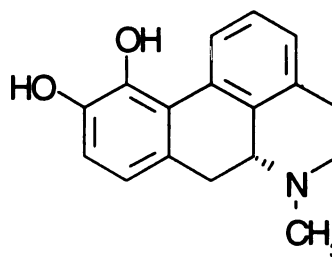
N,N-Dibutyldopamine



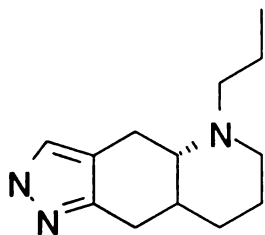
S(+)-Apomorphine



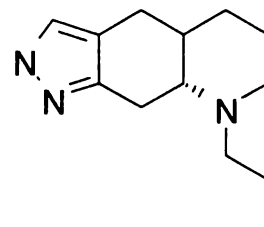
N-*n*-Propylapomorphine



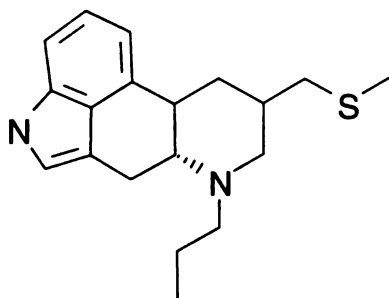
Isoapomorphine



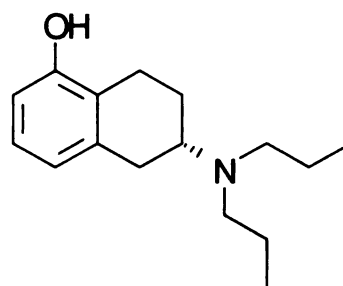
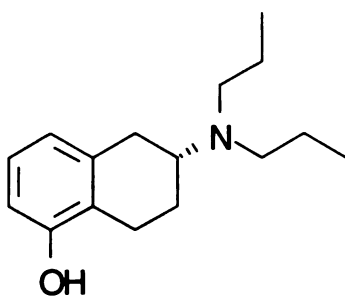
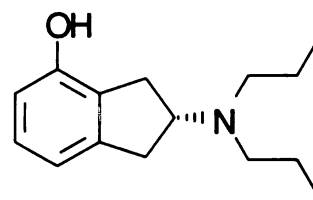
LY156525

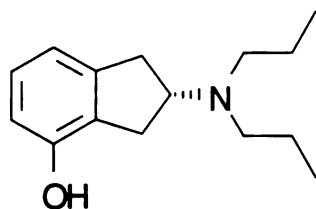


LY171555

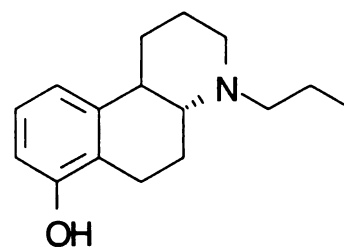


Pergolide

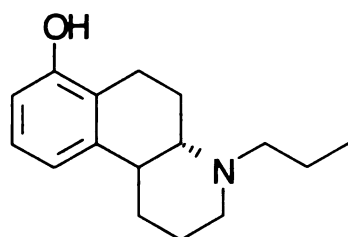
2S-5-Hydroxy-2-
(di-*n*-propylamino)tetralin2R-5-Hydroxy-2-
(di-*n*-propylamino)tetralin2S-4-Hydroxy-2-
(di-*n*-propylamino)indan



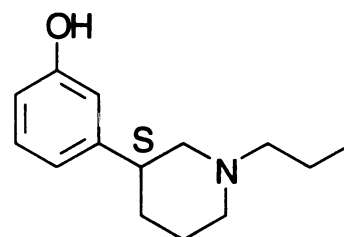
2S-4-Hydroxy-2-
(di-*n*-propylamino)indan



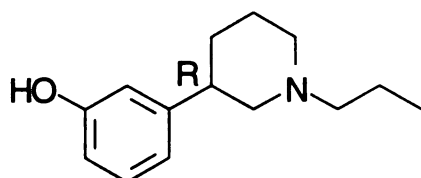
4aR,10bR-7-Hydroxy-4-*n*-propyl-
1,2,3,4,4a,5,6,10b-
octahydrobenzo[f]quinoline



4aS,10bS-7-Hydroxy-4-*n*-propyl-
1,2,3,4,4a,5,6,10b-
octahydrobenzo[f]quinoline



S(-)-3-(3-Hydroxyphenyl)-
N-*n*-propylpiperidine



R(+)-3-(3-Hydroxyphenyl)
N-*n*-propylpiperidine

Figure 2: The four molecules aligned to build the dopamine D2 pharmacophore model.

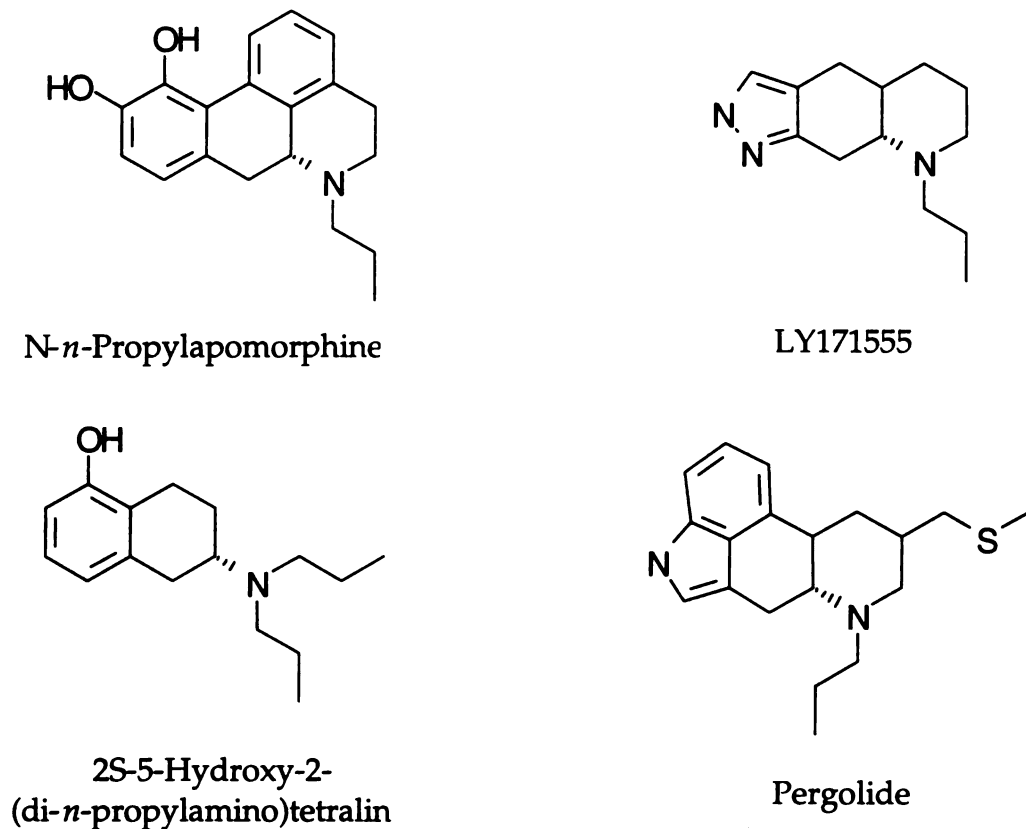
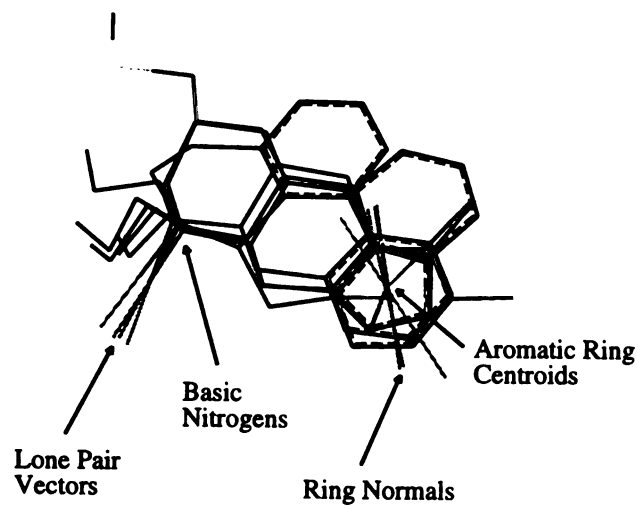


Figure 3: Alignment of the four molecules.



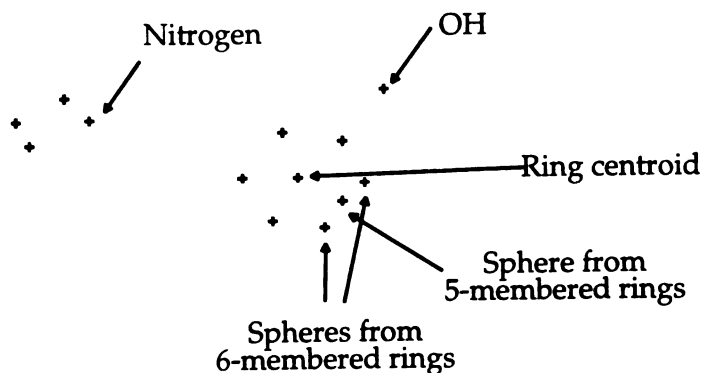


Figure 4: Sphere centers derived from dopamine D2 pharmacophore.

Electrostatic and Geometric Scoring Schemes

Two scoring functions were initially used in docking to the dopamine D2 pharmacophore, one based on electrostatics and one based on how close the atoms of the database molecule fall to the spheres in the target pharmacophore. Both scoring schemes are modified versions of those used by Meng et al. (Meng, Shoichet et al. 1992).

For electrostatic scoring, the electrostatic potential due to partial charges on the atoms of the target positive image is precalculated at regularly spaced grid points. The score of each ligand orientation is calculated by multiplying the partial charge on each atom of the ligand by the potential at its location as determined by trilinear interpolation between the nearest grid points. The results are added subject to the constraint that each atom cannot contribute more than a user-defined maximum value to the total, so that the score is not dominated by a single close interaction. The sign of the score is changed so that more negative scores represent more similar compounds. The score thus

approximates $E = \sum_i q_i \sum_j \frac{q_j}{Dr_{ij}}$ with the sign reversed. Here, q_i and q_j are the partial charges on ligand and positive image atoms, respectively, D is a dielectric constant, and r_{ij} is the distance between ligand atom and positive-image atom. A constant or function value for D may be chosen by the user, but it does not have a physical meaning in measuring similarity; it was set to 1 for this work unless otherwise noted.

The atom-based geometric score measures the fit of the candidate molecule to the positive image based on how close the candidate atoms fall to the target centers. The score is derived from the attractive term of the van der Waals potential and uses the same geometric approximation (Pattabiraman, Levitt et al. 1985) as DOCK 3.0 (Meng, Shoichet et al. 1992). A single ligand atom i and sphere j would contribute $-B_{ij}/r_{ij}^6$ if the van der Waals term were used, where B is the van der Waals B factor and r_{ij} is the distance between atom and sphere. This is approximated by $-\sqrt{B_{ii}}\sqrt{B_{jj}}/r_{ij}^6$. At each point on the grid, the sum of the values of $-\sqrt{B_{ii}}/r_{ij}^6$ due to the spheres closer than a user-defined cutoff is calculated; in this case r is the distance between sphere and grid point. During docking, orientations are scored by multiplying $\sqrt{B_{jj}}$ for each atom by the value obtained by trilinear interpolation between grid points and summing over all ligand atoms. As with electrostatic scoring, there is a user-defined maximum for the amount a single atom can contribute to the score.

As an additional measure of how closely the atoms of docked molecules fit the pharmacophore, DOCK was modified to record the distances between all pairs of spheres and atoms. The minimum distance associated with each sphere was selected and the rms of these distances was used as a proximity measure.

Representation of Pharmacophore Electrostatic and Geometric Properties

The region of space in which scoring grids for docking to the pharmacophore would be calculated was defined by creating a box enclosing the spheres and extending an additional 8 Å in each direction (Figure 5). Using CHEMGRID, electrostatic and atom-based geometric scoring grids were calculated based on the "pharmacophore portion" of each of the four aligned molecules; that is, those atoms listed above as contributing to the definition of the set of spheres representing the pharmacophore. The values at each grid location were averaged to give a single electrostatic grid and a single atom-based geometric scoring grid.

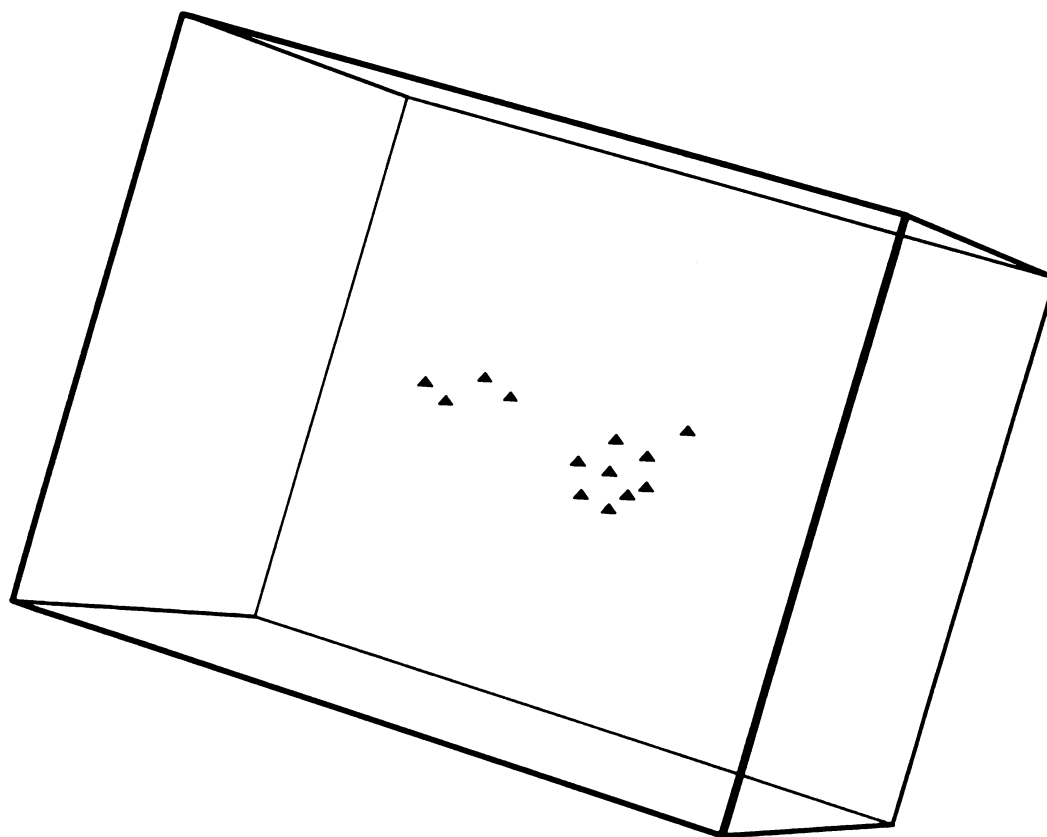


Figure 5: Box defining the scoring region for calculations in the dopamine D2 pharmacophore.

Test Database for the D2 Pharmacophore

A test database for use in docking to the dopamine D2 pharmacophore was created using compounds from the literature and from the MACCS Drug Data Report database (MDDR). 112 compounds (Table 1) whose activities at the dopamine D2 receptor were listed in at least one of four papers (Cannon 1983; Seeman, Watanabe et al. 1985; Katerinopoulos and Schuster 1987; Manallack and Beart 1988) were built using the version of CONCORD (Pearlman 1987) included in the SYBYL molecular modeling package. Three sets of compounds, one set with known activity and two which were of interest because of their molecular geometries, were generated by searching the MDDR. The set of active compounds was found by searching for the letters "dopam" in the description of each compound's activity, then removing a large group of antihypertensive dopamine β -hydroxylase inhibitors. The compounds remaining were active at the dopamine D1 or D2 receptor. A set of 60 compounds which included functional groups similar to those of the pharmacophore but matched the pharmacophore geometry only approximately was produced by searching the MDDR using a query (Figure 6)

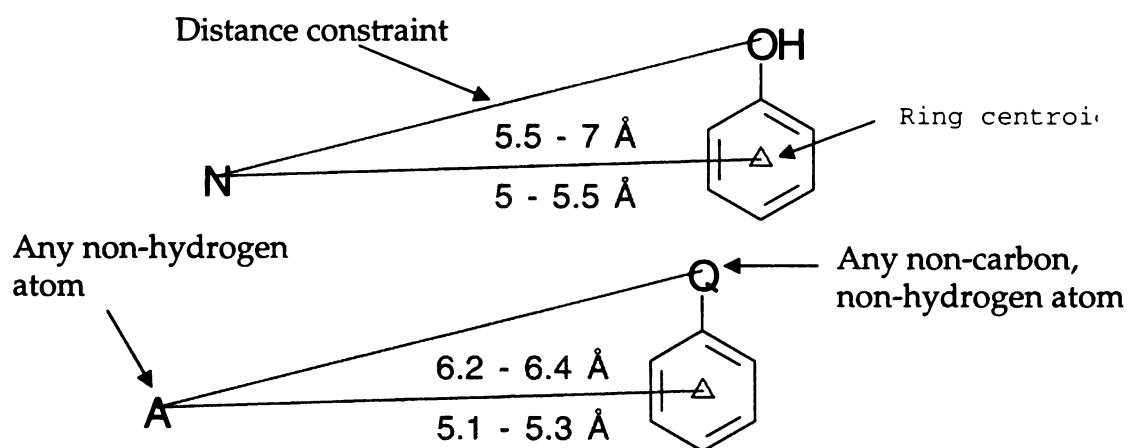


Figure 6: Query structures used to search the MDDR database. Searches found compounds which approximately matched the dopamine D2 pharmacophore (top) and compounds whose geometry approximated that of the pharmacophore (bottom).

Table 1: Compounds from the literature used in the dopamine D2 test database.

Activity codes: a — has dopamine D2 activity; i — known to be inactive at the dopamine D2 receptor. Published sources: KS (Katerinopoulos and Schuster 1987) , MB (Manallack and Beart 1988) , S (Seeman, Watanabe et al. 1985) , and C (Cannon 1983) .

Compound	Activity	Published Source
Dopamine	a	KS
Norepinephrine		KS
Epinephrine	i	KS
N,N-di- <i>n</i> -propyldopamine	a	KS
N,N-di- <i>n</i> -butyldopamine	i	
<i>M</i> -tyramine	a	KS
<i>P</i> -tyramine	i	KS
N,N-di- <i>n</i> -propyl- <i>m</i> -tyramine	a	KS
N-propyl-N-phenethyl- <i>m</i> -tyramine	a	KS
N,N-di- <i>n</i> -propylethylamine	i	KS
2-Fluorodopamine	a	KS
6-Fluorodopamine	a	KS
5-Fluorodopamine	a	KS
(+)-3-(3-Hydroxyphenyl)-N- <i>n</i> -propylpiperidine	a	KS
(-)-3-(3-Hydroxyphenyl)-N- <i>n</i> -propylpiperidine		KS
3,4-Dihydroxyphenyl-N- <i>n</i> -propylpiperidine	a	KS
3-(3-Hydroxyphenyl)-N- <i>n</i> -propylpyrrolidine	i	KS
3-(3-Hydroxyphenyl)-N- <i>n</i> -propyl-5-methylmercaptomethylpiperidine	a	KS
2-Amino-6,7-dihydroxytetralin	a	KS
N,N-dimethyl-2-amino-6,7-dihydroxytetralin	a	KS
N,N-dimethyl-2-amino-5,6-dihydroxytetralin	a	KS
N,N-diethyl-2-amino-5,6-dihydroxytetralin	a	KS
N-methyl,N-isopropyl-2-amino-5,6-dihydroxytetralin		KS
N-methyl-2-amino-5,6-dihydroxytetralin		KS
N-ethyl-2-amino-5,7-dihydroxytetralin	a	KS
N-propyl-2-amino-5,7-dihydroxytetralin	a	KS
N,N-diethyl-2-amino-5-hydroxytetralin	a	KS
S-N,N-diethyl-2-amino-5,6-dihydroxytetralin	a	KS
N,N-diethyl-2-amino-7-hydroxytetralin	a	KS
N,N-diethyl-2-amino-6-hydroxytetralin	a	KS
8-Hydroxy-2-aminotetralin		KS
2S-8-Hydroxy-2-(di-N-ethylamino)tetralin	i	MB
N-ethyl,N-phenethyl-2-amino-5-hydroxytetralin	a	KS
2-Aminotetralin	a	KS
Aminotetralin diethyl	a	S
5,8-Dimethoxytetralin		KS

5-Methylthio-7methoxyaminotetralin		KS
5-Hydroxy-6-methylaminotetralin	a	KS
2-amino-8-chloro-6,7-dihydroxytetralin		KS
2-amino-8-fluoro-6,7-dihydroxytetralin		KS
8-Chloro-N-(4-hydroxyphenylethyl)-N-ethyl-2-amino-6,7-dihydroxytetralin	a	KS
5-Chloro-2-amino-6,7-dihydroxytetralin		KS
<i>exo</i> -2-aminobenzonorbornene	i	KS
<i>endo</i> -2-aminobenzonorbornene	i	KS
6,7-Dihydroxy- <i>exo</i> -2-aminobenzonorbornene	i	KS
6,7-Dihydroxy- <i>endo</i> -2-aminobenzonorbornene	i	KS
<i>exo</i> -2-amino-6,7-dihydroxybenzobicyclo[2.2.2]octene	i	KS
6,7-Dihydroxy-3-chromanamine	a	KS
2-Amino-4,5-dihydroxyindan	a	KS
2-Amino-4,5-dihydroxy-N-methylindan	a	KS
2-Amino-4,5-dihydroxy-N,N-dimethylindan	a	KS
2-Amino-4,5-dihydroxy-N,N-diethylindan		KS
R-4-hydroxy-N,N-di- <i>n</i> -propylaminoindan	a	KS
S-4-hydroxy-N,N-di- <i>n</i> -propylaminoindan	i	KS
R-4-methoxy-N,N-di- <i>n</i> -propylaminoindan	a	S
S-4-methoxy-N,N-di- <i>n</i> -propylaminoindan	a	S
5-Hydroxy-N,N-di- <i>n</i> -propylaminoindan	a	KS
N,N-di- <i>n</i> -propylaminoindan	a	KS
2-(Dipropylaminoethyl)-4-hydroxyindan	a	KS
1,2-Dihydroxy-6-aminobenzocycloheptene	i	KS
N-ethyl-6,7-dihydroxyoctahydrobenzo[g]quinoline	a	KS
N-propyl-6,7-dihydroxyoctahydrobenzo[g]quinoline	a	KS
7,8-Dihydroxyoctahydrobenzo[g]quinoline	i	KS
3-(N,N-diethyl-N-sulfanidyl-N-propyl-6-hydroxyoctahydrobenzo[g]quinoline	a	KS
3-methylthiomethyl-N-propyl-6-hydroxyoctahydrobenzo[g]quinoline	a	KS
<i>trans</i> -N-methyl-octahydrobenzo[f]quinoline-8,9-diol	a	KS
<i>trans</i> -N-ethyl-octahydrobenzo[f]quinoline-8,9-diol	a	KS
<i>trans</i> -N-propyl-octahydrobenzo[f]quinoline-8,9-diol	a	KS
<i>cis</i> -N-propyl-benzo[f]quinoline-8,9-diol	i	S
<i>trans</i> -N-propyl-octahydrobenzo[f]quinoline-7,8-diol	a	KS
<i>trans</i> -Benzo[f]quinoline-7,8-diol	a	S
<i>cis</i> -Benzo[f]quinoline-7,8-diol	a	S
<i>cis</i> -Benzo[f]quinoline-N-methyl-7-ol	i	S
<i>trans</i> -N-propyl-octahydrobenzo[f]quinoline-7,9-diol	a	S

7-Hydroxy-N-propyloctahydrobenzo[f]quinoline	a	KS
4aR,10bR-7-hydroxy-4- <i>n</i> -propyl-1,2,3,4,4a,5,6,10b-octahydrobenzo[f]quinoline	i	MB
8-Hydroxy-N-propyloctahydrobenzo[f]quinoline	a	KS
9-Hydroxy-N-propyloctahydrobenzo[f]quinoline	a	KS
<i>cis</i> -9-hydroxy-N-propyloctahydrobenzo[f]quinoline	i	KS
<i>cis</i> -10-hydroxy-N-propyloctahydrobenzo[f]quinoline		KS
<i>cis</i> -N-propyloctahydrobenzo[f]quinoline		KS
8,9-Dihydroxy-octahydrobenzo[h]isoquinoline	i	KS
R(-)-Apomorphine	a	KS
S(+)-Apomorphine	i	S
Isoapomorphine	i	KS
1,2-Dihydroxyapomorphine	i	KS
2,10,11-Trihydroxy- <i>n</i> -propylnoraporphine	a	KS
N- <i>n</i> -propylnorapomorphine	i	KS
N- <i>n</i> -butylnorapomorphine	i	MB
1,2,9,10-Tetrahydroxyaporphine	a	KS
2,10-Dihydroxyaporphine	a	KS
2,11-Dihydroxyaporphine	a	KS
11-Hydroxy-N-propylnoraporphine	a	KS
11-Methoxy-N-propylnoraporphine	i	S
8-Hydroxy-N-propylnoraporphine	a	KS
9-Hydroxyaporphine	i	KS
7-Hydroxyaporphine	i	KS
N-ethylnorapomorphine	a	KS
Phenanthro[10,1- <i>b,c</i>]azepine	i	KS
Abeorphine	a	KS
2-(3,4-Dihydroxybenzyl)piperidine	i	KS
2-(3',4'-Dihydroxybenzyl)-N-methyl-1,2,3,4-tetrahydroisoquinoline		KS
N-Ethyl- α -phenylphenethylamine		KS
Dibutyldopamine	i	MB
α -Methyldopamine		C
<i>trans</i> -1-methyl-2-(di- <i>n</i> -propylamino)-5-hydroxy-tetralin	a	C
Pergolide	a	MB
Lisuride	a	C
(-)- <i>trans</i> -6-Ethyl-9-oxaergoline	a	C
LY 156525	a	C
LY 171555	a	MB

Table 2: Compounds found by searching the MDDR database which were included in the D2 test database.

Generic Name	Compound Name
FOSQUIDONE	(R,S)-9-[Benzyloxy(hydroxy)phosphoryloxy]-5,14-dihydro-14-methylbenz[5,6]isoindolo[2,1-b]isoquinoline-8,13-dione
	2-[2-(2-Aminoethoxy)ethoxymethyl]-6-methyl-4-(pentafluorophenyl)-1,4-dihydropyridine-3,5-dicarboxylic acid 3-ethyl 5-methyl ester
	5-(1-Ethyl-2-pyrrolidinyl)-2-[5-(ethylsulfonyl)-2-methoxyphenyl]pyrrole
	2-(3,5-Dibromo-2-methoxyphenyl)-5-(1-ethyl-2-pyrrolidinyl)pyrrole
	Benzoylmethyl (14-methyl-8,13-dioxo-5,8,13,14-tetrahydrobenz[5,6]isoindolo[2,1-b]isoquinoline-9-yl) phosphate
	10-Bromo-6beta-aporphin-11-ol
	(R)-6-Methyl-5,6,6a,7-tetrahydro-4H-dibenzo[de,g]quinolin-11-ol
CISCONAZOLE	rac-cis-2-(1H-Imidazol-1-ylmethyl)-3-(2,6-difluorobenzyloxy)-2,3-dihydro-5-fluorobenzo[b]thiophene
	trans-rac-6-[2-[2-(4-Fluorophenyl)-5-isopropyl-3-(4-pyridyl)-1H-pyrrol-1-yl]ethyl]-4-hydroxytetrahydropyran-2-one
	1-Ethyl-3-[3-(dimethylamino)propyl]-3-(1-allyl-6-methoxy-1,2,3,4,4aalpha,5,10,10beta-octahydrobenzo[g]quinolin-3beta-ylcarbonyl)urea
U-72717E	3a(S)-trans-5alpha-(Dipropylamino)-2,3,3a,4,5,6-hexahydro-1H-benzo[de]quinolin-2-one4-methylbenzenesulfonate
	9alpha,11alpha,15alpha-Trihydroxy-16-phenoxy-17,18,19,20-tetranorprosta-4,5,13(E)-trienthioicacid benzyl ester
	3-[3-(4-Phenyl-1-piperazinyl)propoxy]benzeneamine
	1-[3-(Dimethylamino)propyl]-3-ethyl-1-(6-methoxy-1-propyl-1,2,3,4,4aalpha,5,10,10beta-octahydrobenzo[g]quinolin-3beta-ylcarbonyl)urea
	9alpha,11alpha,15alpha-Trihydroxy-16-phenoxy-17,18,19,20-tetranorprosta-4,5,13(E)-trienthioicacid methyl ester

	9alpha,11alpha,15alpha-Trihydroxy-16-phenoxy-17,18,19,20-tetranorprosta-4,5,13(E)-trienthioic acid ethyl ester
	trans-rac-6-[2-[2-(4-Fluorophenyl)-5-isopropyl-3-(2-pyridyl)-1H-pyrrol-1-yl]ethyl]-4-hydroxytetrahydropyran-2-one
	trans-rac-6-[2-[2-(4-Fluorophenyl)-5-isopropyl-3-(3-pyridyl)-1H-pyrrol-1-yl]ethyl]-4-hydroxytetrahydropyran-2-one
ROXINDOLE MESYLATE	3-[4-(3,6-Dihydro-4-phenyl-1(2H)-pyridyl)butyl]-indol-5-ol mesylate
	5-(Dipropylamino)-1-propyl-2,3,3a,4,5,6-hexahydro-1H-benzo[de]quinolin-2-one
	5-(Dimethylamino)-2,3,3a,4,5,6-hexahydro-1H-benzo[de]quinolin-2-one
	5-(Dipropylamino)-5,6-dihydro-4-H-benzo[de]quinoline
	5-(Dimethylamino)-5,6-dihydro-4-H-benzo[de]quinoline
	8-Amino-2,3,7,8,9,9a-hexahydro-1H-benzo[de]quinoline-1-carboxylic acid ethyl ester
	8-(Dimethylamino)-1-methyl-2,3,7,8,9,9a-hexahydro-1H-benzo[de]quinoline
ZY-16681	3-(4-Hydroxyphenyl)-3,4-dihydro-2H-[1]-benzopyran-7,8-diol
	2-(4-Hydroxy-3-methoxyphenyl)thiazolidine
	(2S,4S)-(+)-4-O-alpha-L-Daunosaminy-2-nonanoyl-2,5,12-trihydroxy-1,2,3,4-tetrahydronaphthacen-6,11-dione hydrochloride
	(2S,4S)-(+)-4-O-(alpha-L-Daunosaminy)-2,5,12-trihydroxy-2-pentanoyl-1,2,3,4-tetrahydronaphthacene-6,11-dione hydrochloride
	4-[5-(Benzyloxy)-2-[4-[2(S)-hydroxy-3-phenoxypropylamino]butoxy]phenyl]-2,6-dimethyl-1,4-dihydropyridine-3,5-dicarboxylate dimethyl ester
	4-[5-Amino-2-[4-[2(S)-hydroxy-3-phenoxypropylamino]butoxy]phenyl]-2,6-dimethyl-1,4-dihydropyridine-3,5-dicarboxylate dimethyl ester hydrochloride
	4-[5-Hydroxy-2-[4-[2(S)-hydroxy-3-phenoxypropylamino]butoxy]phenyl]-2,6-dimethyl-1,4-dihydropyridine-3,5-dicarboxylate dimethyl ester hydrochloride

	4-[5-Allyloxy-2-[4-[2(S)-hydroxy-3-phenoxypropylamino]-2(E)-butenyloxy]phenyl]-2,6-dimethyl-1,4-dihydropyridine-3,5-dicarboxylate dimethyl ester hydrochloride
	4-[2-[3-(2-Hydroxy-3-phenoxypropylamino)-2,2-dimethylpropoxy]-5-nitrophenyl]-2,6-dimethyl-1,4-dihydropyridine-3,5-dicarboxylate dimethyl ester hydrochloride
	4-[2-[2-(2-Hydroxy-3-phenoxypropylamino)-2-methylpropoxy]-5-nitrophenyl]-2,6-dimethyl-1,4-dihydropyridine-3,5-dicarboxylate dimethyl ester
	4-[2-[3-(2-Hydroxy-3-phenoxypropylamino)-3-methylbutoxy]-5-nitrophenyl]-2,6-dimethyl-1,4-dihydropyridine-3,5-dicarboxylate dimethyl ester hydrochloride
	4-[5-(Difluoromethoxy)-2-[4-[2(S)-hydroxy-3-phenoxypropylamino]butoxy]phenyl]-2,6-dimethyl-1,4-dihydropyridine-3,5-dicarboxylate dimethyl ester hydrochloride
FAD-104	7-O-(2,6-Dideoxy-2-fluoro-alpha-L-talopyranosyl)pimelyladryamycinone
	2-Isopropyl-5-[2-(4-nitrophenoxyl)ethylamino]-2-(3,4,5-trimethoxyphenyl)pentanenitrile fumarate
	rac-2-Amino-6-propyl-5,5a,6,7,8,9,9a,10-octahydropyrido[2,3-g]quinazolin-9-oldihydrochloride
	7-(Hydroxymethyl)-5-propyl-4,4a,5,6,8a,9-hexahydro-2H-pyrazolo[3,4-g]quinoline
	2-Amino-8-(hydroxymethyl)-6-propyl-5,5a,6,7,9a,10-hexahydropyrido[2,3-g]quinazoline
	2-Amino-8-(methylthiomethyl)-6-propyl-5,5a,6,7,9a,10-hexahydropyrido[2,3-g]quinazoline
	8-(Hydroxymethyl)-6-propyl-5,5a,6,7,9a,10-hexahydropyrido[2,3-g]quinazoline
	8-(Methoxymethyl)-6-propyl-5,5a,6,7,9a,10-hexahydropyrido[2,3-g]quinazoline
	2-Amino-7-(hydroxymethyl)-5-propyl-4,4a,5,6,8a,9-hexahydrothiazolo[4,5-g]quinoline
	5-(Methoxymethyl)-3-propyl-1,2,3,4,5,6,7,7a-octahydro-4a,7-ethanobenzofuro[3,2-e]isoquinolin-9-ol
	3-(Cyclopropylmethyl)-5-(methoxymethyl)-1,2,3,4,5,6,7,7a-octahydro-4a,7-ethanobenzofuro[3,2-e]isoquinolin-9-ol
	4-[4-[3-[4-Acetyl-3-hydroxy-2-propylphenoxy]propylthio]-2-bromophenyl]-3-methyl-4-oxobutanoic acid methyl ester

	3-[1-(3,4-Dichlorophenyl)ethyl]-2-oxoindoline-1-acetic acid
GYKI-52895	1-(4-Aminophenyl)-4-methyl-7,8-methylenedioxy-3H-4,5-dihydro-2,3-benzodiazepine
CY-208-243	(-)-trans-4,6,6a,7,8,12b-Hexahydro-7-methylindolo[4,3-ab]phenanthridine
	9-[2-[N-Ethyl-N-(2-hydroxyethyl)amino]ethylamino]-14-methyl-5,8,13,14-tetrahydrobenz[5,6]isoindolo[2,1-b]isoquinoline-8,13-dione
	9-[N'-[3-(Diethylamino)-1-methylpropylidene]hydrazino]-14-methyl-5,8,13,14-tetrahydrobenz[5,6]isoindolo[2,1-b]isoquinoline-8,13-dione
	2,6-Dimethyl-4-[5-nitro-2-[4-(3-phenoxypropylamino)butoxy]phenyl]-1,4-dihydropyridine-3,5-dicarboxylic acid dimethyl ester
	2-[N-[2-(2-Benzothiényl)ethyl]-N-propylamino]-5-hydroxytetralin
	5-Hydroxy-2-[N-[2-(1H-indol-4-yl)ethyl]-N-propylamino]tetraline
	5-Hydroxy-2-[N-[2-(2-naphthyl)ethyl]-N-propylamino]tetraline
	2-[N-[2-(Benzo[b]thien-3-yl)ethyl]-N-propylamino]-5-hydroxytetraline
	2-(Methoxyimino)-1alpha,3beta-bis(3,4,5-trimethoxyphenyl)cyclopentane
YM-16151-4	4-[2-[4-[2(S)-Hydroxy-3-phenoxypropylamino]butoxy]-5-nitrophenyl]-2,6-dimethyl-1,4-dihydropyridine-3,5-dicarboxylic acid dimethyl ester hydrochloride
PRAMIPEXOLE HYDROCHLORIDE	(S)(-)-2-Amino-6-(propylamino)-4,5,6,7-tetrahydrobenzothiazole dihydrochloride
SCH-39166	(-)-trans-3-Chloro-2-hydroxy-7-methyl-6,7,7a,8,9,13b-hexahydro-5H-benzo[d]naphtho[2,1-d]azepine
	6-Methyl-8beta-(1-methylpyrazol-3-yl)ergoline
PYRINDAMYCIN A	8alpha-(Chloromethyl)-5-hydroxy-2alpha-methyl-1-oxo-6-(5,6,7-trimethoxy-1H-indol-2-ylcarbonyl)-1,2,3,6,7,8-hexahydrobenzo[1,2-b:4,3-b']dipyrrole-2beta-carboxylic acid methyl ester
LU-23-130	3-(N,N-Dipropylaminomethyl)-5-hydroxy-2,3-dihydrobenzofuran
	3-(Dipropylaminomethyl)-5-(isopropylcarbamoxyloxy)-2,3-dihydrobenzofuran
	1-(Diisopropylamino)-6-hydroxyindan

	3-(Dipropylaminomethyl)-5-(4-methylbenzoyloxy)-2,3-dihydrobenzofuran
	4-(3-Chlorophenyl)-2-[2-(isopropylamino)ethylthiomethyl]-6-methylpyridine-3,5-dicarboxylic acid diethyl ester
	4-[2-(Methoxycarbonyl)-3,4-methylenedioxyphenyl]-2,6-dimethyl-1,4-dihydropyridine-3,5-dicarboxylic acid dimethyl ester
	4-[5-Methoxy-2-(methoxycarbonyl)-3,4-methylenedioxyphenyl]-2,6-dimethyl-1,4-dihydropyridine-3,5-dicarboxylic acid dimethyl ester
ELOPIPRAZOLE	2-[[4-(7-Benzofuranyl)-1-piperazinyl]methyl]-5-(4-fluorophenyl)pyrrole
	8-Chloro-5-ethoxy-3-methyl-2,3,4,5-tetrahydro-1H-3-benzazepine-7-ol
	7-Chloro-1-methoxy-3-methyl-2,3,4,5-tetrahydro-1H-3-benzazepin-8-ol
	7-Chloro-3-methyl-1-(2,2,2-trifluoroethoxy)-2,3,4,5-tetrahydro-1H-3-benzazepin-8-ol
	7-Chloro-3-methyl-1-(2-phenylethoxy)-2,3,4,5-tetrahydro-1H-3-benzazepin-8-ol
	7-Chloro-3-methyl-1-pyrrolo-2,3,4,5-tetrahydro-1H-3-benzazepin-8-ol
	1-Allyl-7-chloro-3-methyl-8-(2-methylpropionyloxy)-2,3,4,5-tetrahydro-1H-3-benzazepine
	1-Allyl-7-chloro-8-(2-methoxyacetoxy)-3-methyl-2,3,4,5-tetrahydro-1H-3-benzazepine
	1-Allyl-8-(tert-butoxymethoxy)-7-chloro-3-methyl-2,3,4,5-tetrahydro-1H-3-benzazepine
	4-(1-Propyl-3-piperidinyl)-1,3-dihydro-2H-benzimidazol-2-one
	N,N-Dipropyl-5,6-dihydro-4H-thieno[2,3-b]thiopyran-5-amine
	N-[(5R,8S,10R)-2,6-Dimethylergolin-8-yl]-2-ethyl-2-methylbutyramide
	4-[2(E)-(4'-Fluoro-3,3',5-trimethylbiphenyl-2-yl)ethenylhydroxyphosphinyl]-3(S)-hydroxybutanoic acid
	1-(2,2-Dimethyl-6-nitrochroman-4-yl)pyridin-2(1H)-one
	trans-3-Hydroxy-2,2-dimethyl-4-(2-oxo-1,2-dihydropyridin-1-yl)chromane-6-phosphonic acid diethyl ester

(+)-N-0437	(+)-(R)-2-[N-Propyl-N-[2-(2-thienyl)ethyl]amino]-5-hydroxytetraline
LY-245196	N-(1-Ethyl-2-pyrrolidinylmethyl)-3-methoxy-4,5-dimethylthiophene-2-carboxamide
LY-170913	N-[(1-Ethyl-2-pyrrolidinyl)methyl]-3-methoxy-5-methylthiophene-2-carboxamide
	N-(1-Ethyl-2-pyrrolidinylmethyl)-5-ethylthio-3-methoxythiophene-2-carboxamide
	N-(1-Ethyl-2-pyrrolidinylmethyl)-5-isopropylthio-3-methoxythiophene-2-carboxamide
	N-(1-Ethyl-2-pyrrolidinylmethyl)-3-methoxy-5-(methylsulfonyl)thiophene-2-carboxamide
	4,5-Dichloro-N-(1-ethyl-2-pyrrolidinylmethyl)-3-methoxythiophene-2-carboxamide
	N-(1-Ethyl-2-pyrrolidinylmethyl)-3,4-dimethoxythiophene-2-carboxamide
	7-[2-(2-Aminothiazol-4-yl)-2-[(Z)-3,4-diacetoxybenzoyloxyimino]acetamido]-3-(1-methyl-4-pyridinioaminomethyl)-3-cephem-4-carboxylate
	7-[2-(2-Aminothiazol-4-yl)-2-[(Z)-3,4-diacetoxybenzoyloxyimino]acetamido]-3-[N-ethyl-N-(1-methyl-4-pyridinio)aminomethyl]-3-cephem-4-carboxylate
CQA-206-291	[5R-(5beta,8alpha,10alpha)]-N,N-Diethyl-N'-(1-ethyl-6-methylergolin-8-yl)sulfamide hydrochloride
ISOMOLPAN HYDROCHLORIDE	rac-trans-9-Hydroxy-4-propyl-1,2,3,4a,5,10b-hexahydro-4H-[1]benzopyrano[3,4-b]pyridine hydrochloride
U-66444B	rac-2-(N,N-Dipropylamino)-2,3-dihydro-1H-phenalen-5-ol hydrobromide
U-65556A	2,3-Dihydro-N,N-dimethyl-1H-phenalen-2-amine monohydrochloride
	7,8-Dihydroxy-3-(3,4-dimethoxyphenyl)-3,4-dihydro-2H-1-benzopyran
	4-[2-(3-Chlorobenzylamino)phenyl]-2-methyl-5-oxo-1,4,5,7-tetrahydrofuro[3,4-b]pyridine-3-carboxylic ethyl ester
	14-O-(3,4-Dihexyloxybenzoyl)adriamycin
	14-O-[3-Methoxy-4-(methoxymethyl)benzoyl]adriamycin
	14-O-[3-Hexyloxy-4-(hexyloxymethyl)benzoyl]adriamycin
	10-O-[4-(Hydroxymethyl)-3-undecyloxybenzoyl]adriamycin

14-O-[3-Hydroxy-4-(hydroxymethyl)benzoyl]adriamycin
14-O-(3,4-Isopropylidenedioxy)benzoyl]adriamycin
14-O-(3,4-Dimethoxy)benzoyl]adriamycin
14-O-[(3,4-Dipentanoyloxy)benzoyl]adriamycin
2-[4-(Cyclopropylcarbonyloxy)-3-hydroxyphenyl]-L-alanine
2-[3-Hydroxy-4-(1-methylcyclopropylcarbonyloxy)phenyl]-L-alanine
1-[1-Methyl-5-[4-[2-(1-methylpropoxy)phenyl]-1-piperazinylmethyl]-1H-pyrrol-2-ylmethyl]piperidin-2-one
6-[6-(4-Bromobenzenesulfonamido)bicyclo[2.2.2]octan-2-yl]-5(Z)-hexenoic acid
6,7-Dichloro-4-(N,N-dimethylsulfamoyl)benzofuran-2-carboxylic acid
6,7-Dichloro-4-(N,N-diethylsulfamoyl)benzofuran-2-carboxylic acid
4-(N-Benzyl-N-methylsulfamoyl)-6,7-dichlorobenzofuran-2-carboxylic acid
4-[2-(Dipropylamino)ethyl]-2,3-dihydro-1H-benzimidazole-2-thione
4-[2-(N-Butyl-N-methylamino)ethyl]-2,3-dihydro-1H-benzimidazol-2-one
4-[2-[N-(2-Phenylethyl)-N-propylamino]ethyl]-2,3-dihydro-1H-benzimidazol-2-one
4-[2-[N-Propyl-N-[2-(2-thienyl)ethyl]amino]ethyl]-2,3-dihydro-1H-benzimidazole-2-thione
4-[2-(N-Butyl-N-methylamino)ethyl]-2,3-dihydro-1H-benzimidazole-2-thione
4-[2-[N-(2-Phenylethyl)-N-propylamino]ethyl]-2,3-dihydro-1H-benzimidazole-2-thione
4-[2-[N-Propyl-N-[2-(2-thienyl)ethyl]amino]ethyl]-2,3-dihydro-1H-benzimidazol-2-one
4-[2-[N-Propyl-N-[2-(3-thienyl)ethyl]amino]ethyl]-2,3-dihydro-1H-benzimidazol-2-one
4-[2-[N-Propyl-N-[2-(3-thienyl)ethyl]amino]ethyl]-2,3-dihydro-1H-benzimidazole-2-thione
1-Benzyl-5-[3,3-bis(4-methoxyphenyl)-2-propenyl]-1H-imidazole
3-[trans-5-(2-Methyl-1-propenyl)-1,3,4,5-tetrahydrobenz[cd]indol-4-ylamino]propanoic acid methyl ester

trans-4-(2-Hydroxypropylamino)-5-(2-methyl-1-propenyl)-1,3,4,5-tetrahydrobenz[cd]indole
N-[trans-5-(2-methyl-1-propenyl)-1,3,4,5-tetrahydrobenz[cd]indol-4-yl]-N'-phenylurea
trans-5-(2-Methyl-1-propenyl)-4-(2-phenylethylamino)-1,3,4,5-tetrahydrobenz[cd]indole
N-[trans-5-(2-Methyl-1-propenyl)-1,3,4,5-tetrahydrobenz[cd]indol-4-yl]glycine methyl ester
N-(Methoxycarbonylmethyl)-N-[trans-5-(2-methyl-1-propenyl)-1,3,4,5-tetrahydrobenz[cd]indol-4-yl]glycine methyl ester
(+)-4-Demethoxy-14-(2,2,2-trifluoroethyl)daunorubicin hydrochloride
(+)-4-Demethoxy-3'-N-(trifluoroacetyl)-14-(2,2,2-trifluoroethoxy)daunorubicin
N-[4-[4-(6-Fluoro-1,2-benzisoxazol-3-yl)-1-piperidinyl]butyl]cyclohexane-1,2-dicarboximide
N-[4-[4-(6-Fluoro-1,2-benzisoxazol-3-yl)-1-piperidinyl]butyl]-4-cyclohexene-1,2-dicarboxamide
N-[4-[4-(6-Fluoro-1,2-benzisoxazol-3-yl)-1-piperidinyl]butyl]bicyclo[2.2.1]heptane-2,3-di-exo-carboxamide
N-[4-[4-(6-Fluoro-1,2-benzisoxazol-3-yl)-1-piperidinyl]butyl]bicyclo[2.2.1]hept-5-en-2,3-di-exo-carboximide
N-[4-[4-(6-Fluoro-1,2-benzisoxazol-3-yl)-1-piperidinyl]butyl]bicyclo[2.2.2]octane-2,3-dicarboximide
N-[2-(Cyclopentylthio)-4-nitrophenyl]methanesulfonamide
(3R,5R,8R,9S,10R)-6-Cyano-8,9-dihydroxy-8-methyl-1-(4-methylphenylsulfonyl)-2,3-dihydroergoline
(3R,5R,8R,9S,10R)-8,9-Dihydroxy-6,8-dimethyl-1-(4-methylphenylsulfonyl)-2,3-dihydroergoline
(5R,8R,9S,10R)-8,9-Dihydroxy-6,8-dimethylergoline
(3S,5R,8R,9S,10R)-6-Cyano-8-(3,3-diethylureido)-9,10-dihydroxy-1-(4-methylphenylsulfonyl)-2,3-dihydroergoline
(3S,5R,8R,9R,10S)-6-Cyano-8-(3,3-diethylureido)-9,10-dihydroxy-1-(4-methylphenylsulfonyl)-2,3-dihydroergoline
(3S,5R,8R,9S,10R)-8-(3,3-Diethylureido)-9,10-dihydroxy-6-methyl-1-(4-methylphenylsulfonyl)-2,3-dihydroergoline

	(5R,8R,9S,10R)-8-(3,3-Diethylureido)-9,10-dihydroxy-6-methylergoline
DUOCARMYCIN B2	8-(Bromomethyl)-5-hydroxy-2-methyl-1-oxo-6-(5,6,7-trimethoxyindol-2-ylcarbonyl)-1,2,3,6,7,8-hexahydrobenzo[1,2-b:4,3-b']dipyrrole-2-carboxylic acid methyl ester
SILYCHRISTIN	3beta,5,7-Trihydroxy-2alpha-[7-hydroxy-2beta-(4-hydroxy-3-methoxyphenyl)-3alpha-(hydroxymethyl)-2,3-dihydrobenzofuran-5-yl]-2,3-dihydro-4H-benzo[b]pyran-4-one
	8-Fluoro-2-[3-(3-pyridyl)propyl]-2,3,4,5-tetrahydro-1H-pyrido[4,3-b]indole N-oxide
	7-Butyl-3-(4-chloro-3-nitrophenylsulfonyl)-9,9-dimethyl-3,7-diazabicyclo[3.3.1]nonane
	trans-1,1-Dimethyl-5-nitro-3-(2-oxo-1,2-dihydropyridin-1-yl)indan-2-ol
	trans-5-Amino-1,1-dimethyl-3-(2-oxo-1,2-dihydropyridin-1-yl)indan-2-ol
	trans-1-(4-Fluorobenzamido)-6-niroindan-2-ol
	1-[[4,6-Dichloro-1-(2,4-dichlorophenyl)-1,3-dihydro-1-isobenzofuranyl]methyl]-1H-imidazole
	2-(2,2-Dimethyl-6-nitro-3,4-dihydro-2H-1-benzopyran-4-yl)pyridine-N-oxide
	7-Fluoro-2-[5-fluoro-2-[4-[4-[2-(3,4,5-trimethoxyphenyl)ethyl]-1-piperazinyl]-2-butenyloxy]phenyl]-2-isopropyl-4-methyl-2,3-dihydro-1,4-benzothiazin-3-one 1,1-dioxide
	9,12-Epoxy-3-(ethylthio)-10-hydroxy-10-(hydroxymethyl)-9-methyl-2,3,9,10,11,12-hexahydro-1H-diindolo[1,2,3-fg:3',2',1'-kl]pyrrolo[3,4-i][1,6]benzodiazocin-1-one
	9,12-Epoxy-3-(ethylthio)-10-hydroxy-9-methyl-1-oxo-2,3,9,10,11,12-hexahydro-1H-indolo[1,2,3-fg:3',2',1'-kl]pyrrolo[3,4-i][1,6]benzodiazocine-10-carboxylic acid methyl ester

which contained a nitrogen and an aromatic ring with attached hydroxyl but which allowed a wide range of distances among them. The third set of MDDR compounds, which included 83 structures which had geometries approximating that of the pharmacophore but which did not necessarily contain its functional groups, was found with a query (Figure 6) which required distances between an aromatic ring, an atom attached to it, and another atom to be quite close to those of the pharmacophore but allowed a wide range of atom types. After removing duplicates, a total of 176 compounds (Table 2) were found in the MDDR; the entire D2 test database included 293 compounds. Partial charges were assigned to all molecules in SYBYL using the method of Gasteiger and Marsili (Gasteiger and Marsili 1980). Centroids were added to the aromatic rings. Molecules in the D2 test database were designated as active, inactive, or of other activity (not active at the dopamine D2 receptor) according to the literature or the "activ.class" field in the MDDR database. Centroids and charges were also added to the MDDR database, which at that time contained about 11,000 3D structures, so that it could be used as a test database for docking. It should be noted that some compounds were included in both the MDDR database and the D2 test database.

DOCK Experiments in the D2 Pharmacophore

Single Mode Docking of Molecules from the Literature

Ten of the remaining molecules taken from the Manallack and Beart paper (Manallack and Beart 1988) were individually aligned to pergolide using SYBYL. The points used for matching were defined as for the four molecules used to construct the pharmacophore positive image. Conformations of the molecules were adjusted as necessary to better match pergolide. The resulting models were docked individually to the

pharmacophore to see if DOCK could locate matching orientations for the molecules. Both ligand and receptor bins had widths of 0.6 Å and overlaps of 0.0 Å. Electrostatic scores, atom-based geometric scores, and the rms proximity measure were calculated and saved for all orientations generated. The rms deviation in atom positions between each docked orientation and the orientation generated by hand alignment of the molecule to the pharmacophore was also recorded. Electrostatic scores were calculated using a cap of -1000; atom-based geometric scores were subject to a cap of -50.

Search Mode Docking of D2 Database to the Pharmacophore

The D2 test database was docked to the pharmacophore in SEARCH mode, meaning that the best-scoring orientation of each molecule is saved and molecules are ranked according to the score of this orientation. In this case, the score used to determine the ranking was the sum of the electrostatic score and the atom-based geometric score. A cap of -1000 was used for the electrostatic score, -50 for the atom-based geometric score. The ligand bin width was 0.4 Å and the overlap was 0.1 Å; receptor bin width and overlap were 0.8 Å and 0.2 Å. The D2 test database was also docked to the pharmacophore using the same conditions but with only the electrostatic score.

Search Mode Docking of MDDR to the Pharmacophore

The same conditions were used to dock the MDDR database to the pharmacophore in SEARCH mode. As for the D2 database, docking was done using the sum of the electrostatic and atom-based geometric scores and using the electrostatic score alone.

Studying the Effect of Varying the Geometric Score Cap

The program SCOREOPT, which calculates DOCK scores for molecules in single orientations, was used to calculate atom-based geometric scores for

the top orientations generated by docking the D2 test database to the pharmacophore using only the electrostatic score. The cap used was varied to study the effect of different maxima; values used were -50, -400, -1000, -10000, and -25000. The value of the geometric score cap determines the maximum distance at which an atom overlapping a sphere receives the maximum score. The maximum distance is also a function of the van der Waals B values for the atoms involved. Since the score at a given point is approximately equal to $-\frac{\sqrt{B_{ii}}\sqrt{B_{jj}}}{r_{ij}^6}$, cap values were selected so that if $\sqrt{B_{ii}} = \sqrt{B_{jj}} = 20$, chosen to approximate the most common values for carbon, oxygen, and nitrogen, the value of r where that score is achieved was in a range which included 1 Å (cap = -400) and 0.5 Å (cap = -25000).

Search Mode Docking of the D2 Database Using a Weighted Combination of Scores

In an attempt to choose orientations using approximately equal contributions from the electrostatic and atom-based geometric scores, the D2 test database was docked to the pharmacophore using the sum of the electrostatic score and the product of the atom-based geometric score with a value based on the magnitude of the two scores in this system. Since a typical value for the electrostatic score (with a cap of -1000) was about -300, and a typical value for the atom-based geometric score (with a cap of -10000) was about -100000, DOCK runs were carried out using scale factors of 0.001 and 0.002 for the atom-based geometric score. Electrostatic and atom-based geometric caps were -1000 and -10000 respectively.

Testing DOCK as a Tool for Pharmacophore Generation

Models for Testing Pharmacophore Alignment

Models of the dopaminergic compounds (Figure 7) used as a test case for DISCO (Martin, Bures et al. 1993) were constructed following the methods used by the authors of DISCO as closely as possible. Using SYBYL, models were generated with CONCORD (Pearlman 1987), ring centroids and site points were created, and hydrogens and charges were added. A set of spheres with centers at the coordinates of the nonhydrogen atoms of molecule I was constructed. A box to define the region of space used for scoring grids was created by enclosing the largest molecule, XX, and adding an additional 8 Å in each direction. Grids for electrostatic and atom-based geometric scoring were generated for molecule I using a dielectric of 1 and a cutoff of 10 Å for electrostatic and atom-atom interactions.

Docking to Generate Possible Alignments

Molecules II through XX were docked to molecule I in SINGLE mode, so that all orientations generated were saved. Because large numbers of orientations require large amounts of computer storage space, small bin sizes were used to keep the number of orientations reasonable. Both ligand and receptor bins had a width of 0.3 Å and an overlap of 0.1 Å; the resulting number of orientations generated per molecule ranged from 367 to 3458. Electrostatic and atom-based geometric scores were calculated (with caps of -1000 and -10000) but were not used to rank orientations. For molecules II, XIII, and XVIII, docking was repeated with bin widths of 0.6 Å and bin overlaps of 0.15 Å, yielding 1788 to 9496 orientations.

Lablsan: a Program to Identify Overlaid Functional Groups

In order to identify pharmacophore points by finding orientations of the pharmacophore molecules in which similar functional groups lie in the

Figure 7: Molecules used to test DOCK as a tool for aligning molecules and generating pharmacophores

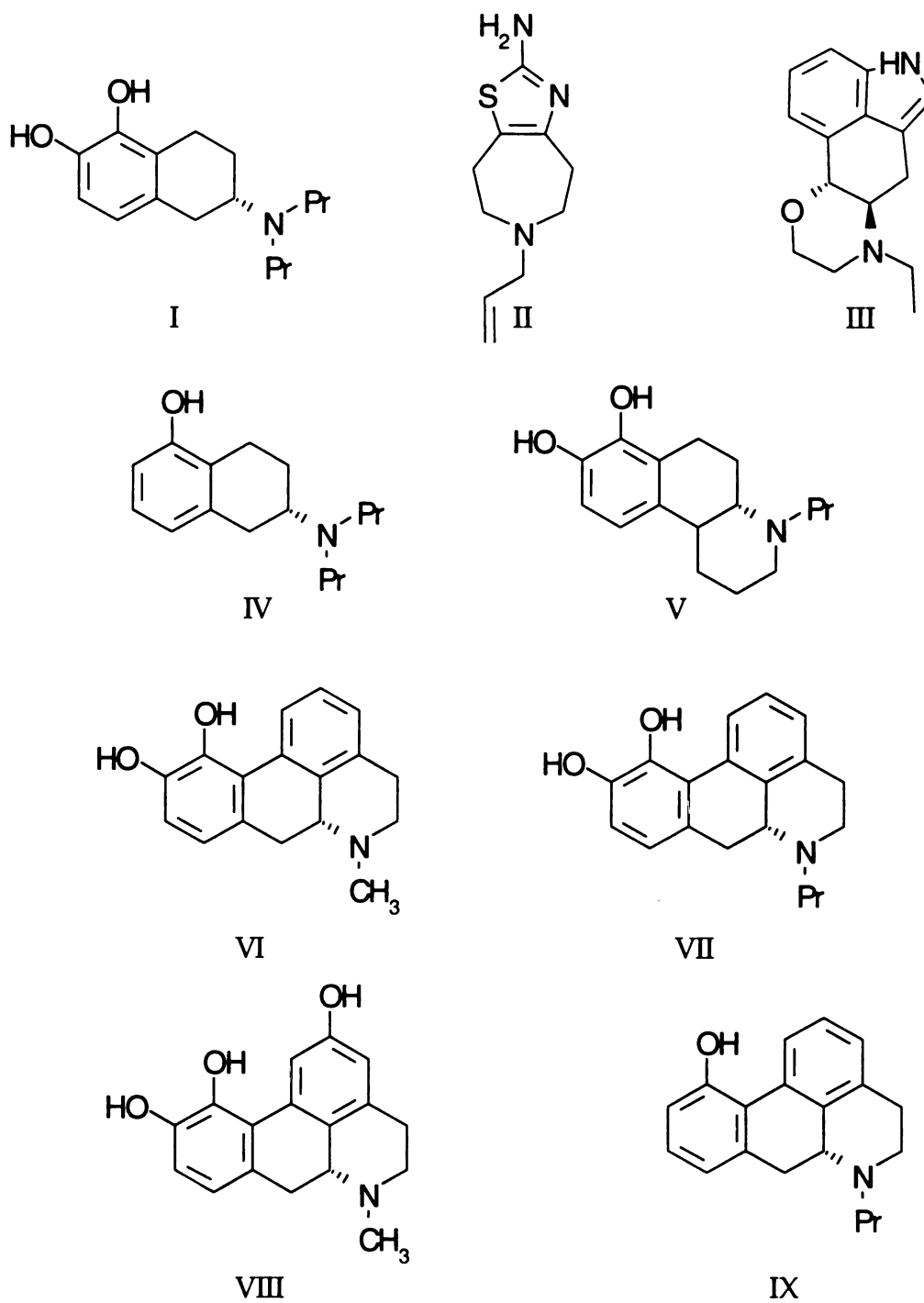


Figure 7 (continued)

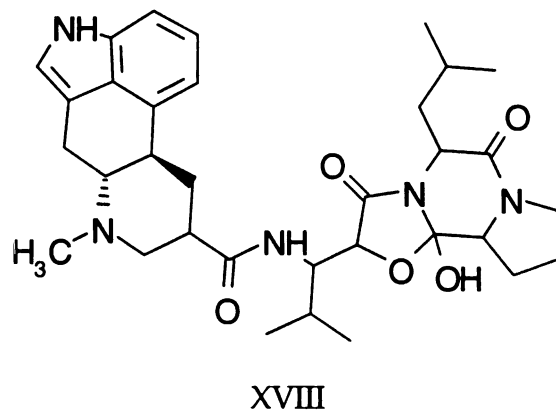
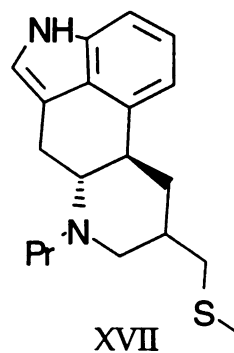
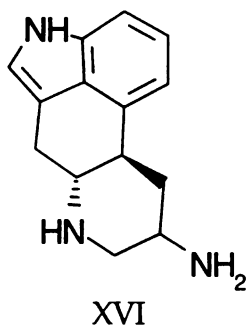
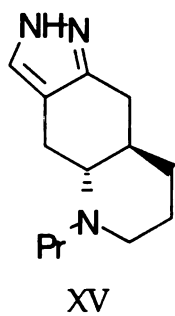
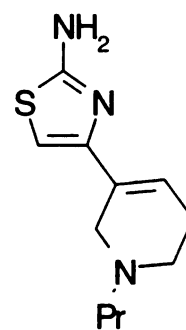
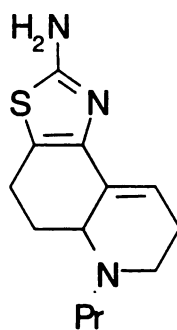
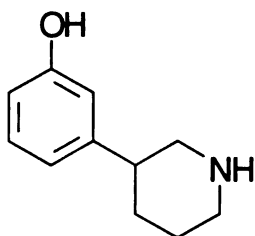
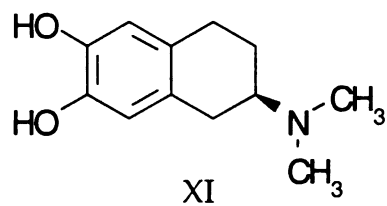
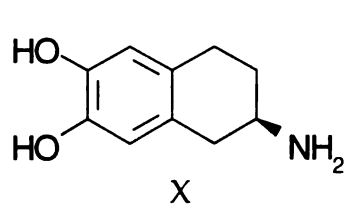
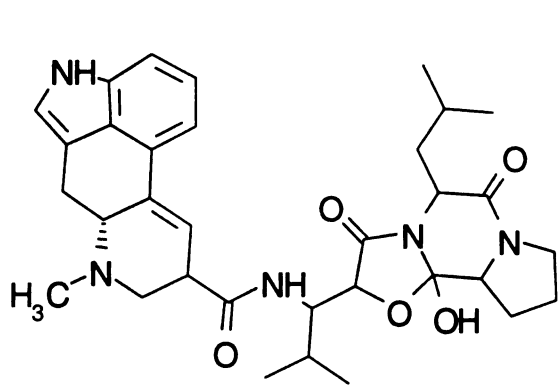
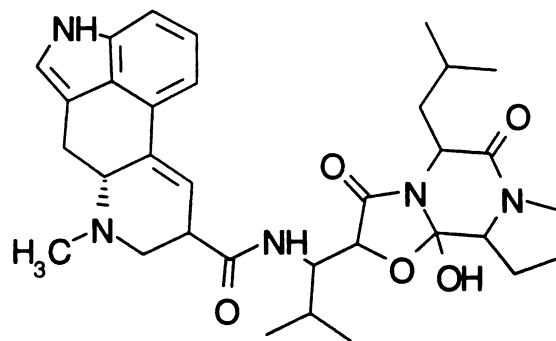


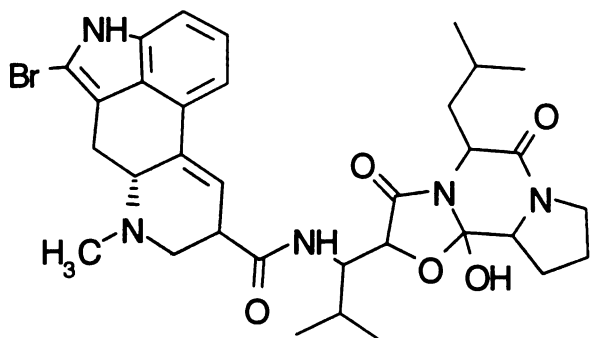
Figure 7 (Continued)



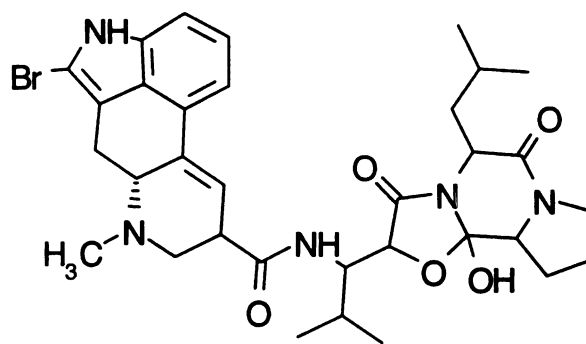
XIX



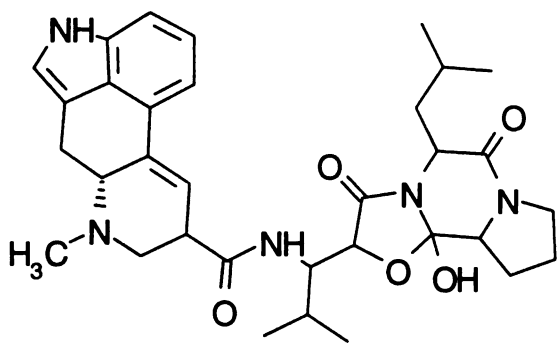
XIX



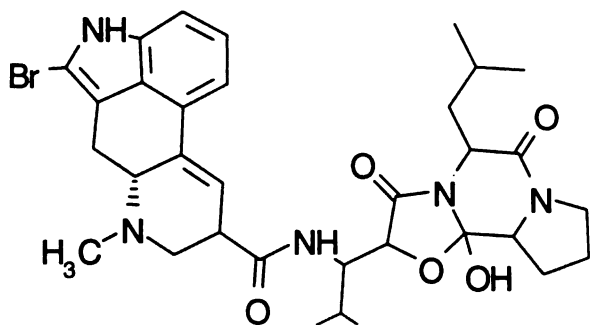
XX



XX



XIX



same region of space, I wrote the program lablscan. This program reads orientations from DOCK output and a set of labels for atoms and spheres, then identifies orientations in which spheres and atoms of the same type are within a user-specified distance of each other. Although DOCK may generate many orientations in which a particular combination of spheres and atoms are close to each other, lablscan saves only one orientation for each such combination. Lablscan matches only one atom to a sphere, since associating an atom with the closest sphere may result in the same atom being matched to multiple spheres. To reduce the number of orientations considered, the user may specify a minimum number of matched pairs with a given label which must be present for an orientation to be considered.

Identifying Labeled Matches

Spheres and atoms of the molecules from the paper describing DISCO were given labels similar to those used in that paper. Each molecule's basic nitrogen was labeled NH, the site point at the location of the probable hydrogen-bond acceptor on the receptor was labeled HA, the H-bond donor on the aromatic ring was labeled RD, and the ring centroid was labeled CR.

In order to see if DOCK and lablscan could retrieve the same combination of orientations found by DISCO, lablscan was run on the orientations generated by docking all other molecules to I. Orientations were saved if they met one of three criteria: two HA pairs, one NH pair, and one RD pair; two HA pairs; or one NH pair, one RD pair, and one CR pair.

Docking Using Only Labeled Atoms

Compounds II to XX were also docked to I considering only the labeled atoms. Since using only labeled atoms produced much smaller sets of spheres and atoms, it was necessary to increase bin widths to 0.9 Å and bin overlaps to 0.4 Å before at least some orientations were produced for all compounds. The

number of orientations ranged from 2 to 162. (In the course of docking, I discovered and fixed a bug in DOCK which caused some centers not to be checked against the bin overlap. This was primarily a problem for small sets of distances. A version of the makbin subroutine which corrects the problem is included in Appendix 1.) Compounds were examined for orientations matching the pharmacophore; when DOCK failed to find some compounds which should have matched the pharmacophore, individual bins were examined for matching atom-sphere pairs.

Bovine Pancreatic Trypsin Inhibitor: Positive Docking

Representing BPTI for Docking

As a test system for positive docking to a region of a protein, I used the trypsin-binding region of bovine pancreatic trypsin inhibitor (BPTI) taken from the *2ptc* structure of the BPTI/bovine trypsin complex (Marquart, Walter et al. 1983) from the Brookhaven Protein Data Bank (Bernstein, Koetzle et al. 1977; Abola, Bernstein et al. 1987). Spheres for positive docking, 60 in all (Figure 8), were derived from the nonhydrogen atoms of those residues of BPTI which were in contact with trypsin. The boundaries of the scoring grid were defined by creating a box enclosing the spheres plus an additional 5 Å in each direction (Figure 9). Electrostatic and atom-based geometric scoring grids were created with CHEMGRID using a spacing of 0.30 Å between points and a cutoff of 10 Å for atoms interacting with a given point.

Confining Docked Ligands to the Region of the Inhibitor

To keep orientations of database molecules from extending very far outside the region occupied by the inhibitor, I modified CHEMGRID to create a grid in which the region outside the inhibitor was marked. The version of CHEMGRID which accompanied DOCK 3.0 (Meng, Shoichet et al. 1992)



Figure 8: Positive docking centers with BPTI surface.

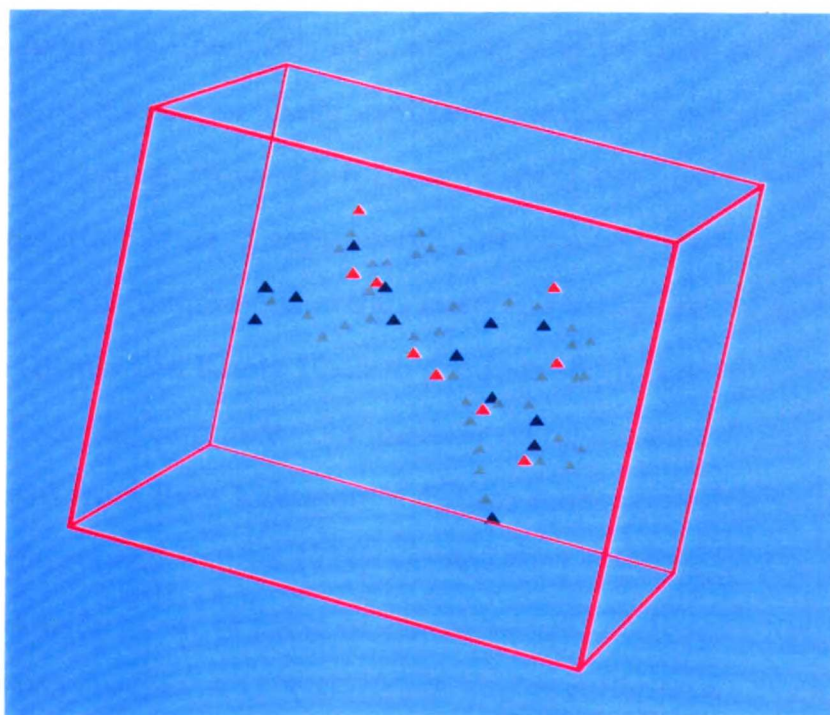


Figure 9: Positive docking centers with box defining scoring grid.

produced a "bump grid" in which the space occupied by the receptor was marked by setting all points within user-specified distances of polar and nonpolar receptor atoms to "F", meaning false, and other points to "T" for true. DOCK rejected all orientations in which a user-defined number of atoms fell within the region of "F" points, thus preventing ligands from penetrating the receptor. I modified CHEMGRID to set all points to "F" *except* those within user-defined limits of the inhibitor atoms, so that DOCK would discard orientations in which more than a user-defined number of atoms extended outside the inhibitor. Even when the allowed distance from a ligand atom for a grid point to be set to "T" was increased to 2.5 Å, a few points within the inhibitor were still set to "F"; I therefore added code to CHEMGRID to search for isolated "F" points in "T" regions and change their value.

A Test Database for Positive Docking

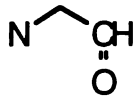
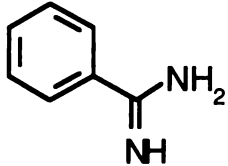
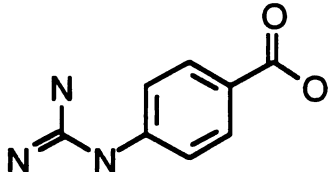
The Available Chemicals Directory is a useful database for docking because it contains commercially available compounds which may be purchased for testing. However, it contains more than 100,000 3D structures, which makes it too large to conveniently use as a test database for developing new methods. The database was thus clustered using the method of Bemis and Kuntz (Bemis and Kuntz 1992) to produce a smaller set of compounds for use in methods development. Compounds were clustered based on hash codes derived from their interatomic distances using the Jarvis-Patrick algorithm (Jarvis and Patrick 1973); ten nearest neighbors were examined and compounds with seven of these in common were clustered together. This produced 5065 clusters. A DOCK-format database was created using one compound from each cluster.

Compounds Related to Trypsin Inhibitors for Testing Positive Docking

As a supplement to the test database derived from the ACD, in which compounds were selected with diversity in mind, a small list of compounds were selected because they were known to inhibit trypsin or were related structurally to known trypsin inhibitors. These compounds were located by

Table 3: Examples of trypsin inhibitors and related compounds. Molecules were retrieved from the Available Chemicals Directory and taken from trypsin complexes in the Protein Data Bank (Bernstein, Koetzle et al. 1977; Abola, Bernstein et al. 1987) .

(a) Substructure queries used to locate molecules in the ACD and the resulting hits.

Query	Hits
 (for peptide aldehydes)	leupeptin antipain
	3-Nitrobenzamidine Hydrochloride 4-Aminobenzamidine Dihydrochloride 4-Amidinobenzamidine Hydrochloride Ethyl-4-amidinobenzoate <i>p</i> -APMSF Benzamidine 4',6-Diamidino-2-phenylindole 4-Chlorobenzamidine Hydroiodide Diminazene Aceturate <i>p</i> -Amidinophenyl- <i>p</i> -(6-amidino-2-indolyl) phenyl ether Maybridge SPB 06264 4-Hydroxybenzamidine Hydrochloride 1,4-(Diamidino)benzene Dihydrochloride
ϵ -Aminocaproic Acid (name search)	6-Aminocaproic Acid
4-Aminomethyl Cyclohexanecarboxylic Acid (name search)	<i>trans</i> -4-Aminomethyl Cyclohexanecarboxylic Acid
	4-Nitrophenyl-4-guanidinobenzoate

searching the entire Available Chemicals Directory and by examining the structures of trypsin complexes in the Brookhaven Protein Data Bank (Bernstein, Koetzle et al. 1977; Abola, Bernstein et al. 1987). The queries used to search the ACD were derived from the classes of reversible serine protease inhibitors listed in a review by Powers and Harper (Powers and Harper 1986) and included peptide aldehydes, benzamidines and guanidinobenzoates; the queries and the resulting hits are listed in Table 3a. The small molecule inhibitors and segments of protein inhibitors derived from trypsin complexes in the PDB are described in Table 3b. After eliminating duplicates, twenty-five molecules were retrieved; they were added to the test database as a way to examine the ability of the docking methods to pick compounds which should bind from among a variety of molecules.

DOCK Experiments in BPTI

Varying Electrostatic Scoring Methods

The test database was docked to the positive image of BPTI using

Table 3 (b) Ligands from trypsin complexes in the pdb. Residue numbers indicate that residue coordinates taken from the structure of a protein inhibitor were used.

PDB File Name	Residues	Ligand Name
1gbt		guanidinobenzoyl group from guanidinobenzoylated serine
1ppc		NAPAP — N-alpha-(2-naphthylsulphonylglycyl)-DL- <i>p</i> -amidinophenylalanyl piperidine
1pph		TAPAP — N-alpha-(2-tosylsulphonylglycyl)-DL- <i>p</i> -amidinophenylalanyl piperidine
1tpp		2- <i>p</i> -amidino-phenylpyruvate
3ptb		benzamidine
1mct	4-6 Pro-Arg-Ile	bitter gourd (<i>Momordica charantia</i>) seed inhibitor
1ppe	4-6 Pro-Arg-Ile	squash (<i>Cucurbita maxima</i>) seed inhibitor

variations on the electrostatic scoring scheme first tested in the dopamine D2 pharmacophore. The electrostatic function described above was used alone for scoring, along with the bump grid to keep ligands inside the inhibitor. The electrostatic function was also modified by replacing the factor $1/r$ with the three-Gaussian approximation used by Good *et al.* (Good, Hodgkin *et al.* 1991). In both of these runs the maximum distance from an atom to a grid point which was used in calculating the potential was 10 Å; a run was also carried out using the Gaussian approximation to $1/r$ and a maximum distance of 5 Å. Bin widths of 0.5 Å and bin overlaps of 0.1 Å were used in all three cases, and the contribution of a single atom to the score was capped at 1000. A "normalized" electrostatic score was constructed by dividing the electrostatic score of each molecule by the square root of the sum of the squares of the molecule's partial charges. The test database was docked using this score, with bin widths of 0.6 Å and bin overlaps of 0.1 Å and a cap of 1000.

Positive Docking Using Labeled Matching

The test database was also docked to the positive image by labeling some spheres according to their chemistry and only allowing atoms with corresponding labels to match them. Labeled matching is based on the work of Brian Shoichet (Shoichet and Kuntz 1993) but was implemented slightly differently in DOCK 3.5 by Mike Connolly. The labels assigned to spheres derived from amino acid residues in BPTI are listed in Table 4; those assigned to charged and polar functional groups in database molecules are listed in Table 5. The test database was docked to the positive image of BPTI using the same electrostatic function as in the dopamine D2 pharmacophore; the other conditions were the same as those used for the initial dock runs in the BPTI system.

Table 4: Labels assigned to spheres derived from amino acid residues in BPTI. Atom names are those used in Brookhaven Protein Data Bank (Bernstein, Koetzle et al. 1977; Abola, Bernstein et al. 1987) format.

Label	Residue Name	Atom
Hydrogen-bond Acceptor	Asparagine	OD1
	Glutamine	OE1
Hydrogen-bond Donor	Asparagine	ND2
	Glutamine	NE2
	Histidine (HIS or HIP)	ND1
		NE2
	Tryptophan	NE1
Negatively Charged	Aspartate	OD1
		OD2
		CG
	Glutamate	OE1
		OE2
		CD
Positively Charged	Arginine	NE
		CZ
		NH1
		NH2
	Lysine	NZ
Hydroxyl	Serine	OG
	Threonine	OG1
	Tyrosine	OH

Table 5: Labels assigned to charged and polar functional groups in database molecules.

The labels are associated with the atoms shown in **bold** within each functional group.

Label	Functional Groups
Hydrogen-bond acceptor	P=O S=O C=O N aromatic N sp ² (except nitro) O sp ³
Hydrogen-bond donor	NH NH₂ NH₃
Negatively charged	O⁻ P—O⁻ S—O⁻ C—O⁻
Positively charged	N sp ³ , positively charged
Hydroxyl	OH

Varying Geometric Scoring Methods

Variations on the atom-based geometric scoring scheme were investigated by docking the test database to the positive image of BPTI using the atom-based score with the bump grid. In all cases, bin widths were 0.6 Å and bin overlaps were 0.1 Å. Two separate runs, one with a cap of -10000 and one with a cap of -1000, were made with the scoring function which had been used in the dopamine D2 pharmacophore. The Gaussian approximation to $1/r$ which had been used as a factor in the electrostatic score was used to replace $1/r^6$ in the atom-based geometric score. Two forms of "normalization" were used to compensate for the fact that large molecules can match more atoms. In the first, the score of each molecule was divided by the number of

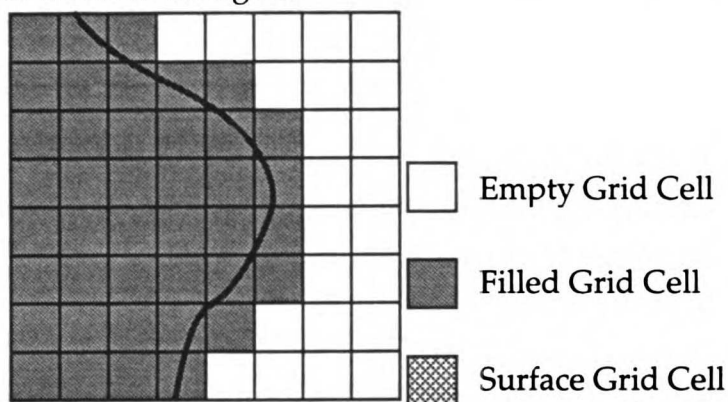
nonhydrogen atoms; in the second, the score was divided by the square root of this number.

The Shape-Based Scoring Scheme

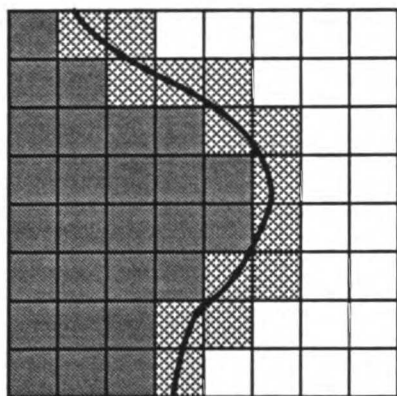
The method used to represent the inhibitor surface on a grid for scoring was based on some of the techniques developed by Karfunkel and Eyraud (Karfunkel and Eyraud 1989) for creating union surfaces for sets of overlaid small molecules. A grid is defined containing the molecule or region of interest, and grid cells are labeled as being filled by the molecule if they lie within the van der Waals radius of any atom (Figure 10a). Filled cells which have empty neighbors are designated as surface cells (Figure 10b). In order to represent surfaces at grid resolutions finer than those typically used for DOCK scoring grids, a box was created to define a smaller region; it enclosed the spheres with an additional 2.25 Å on each side. Surfaces created in this manner at low resolution (grid spacing of about 1 Å) occupied grid cells in the region of the protein surface; however, surfaces created at higher resolution (grid spacing of 0.2 Å) included surface points in regions of low atom density within the protein. To keep the detailed surface confined to the region of the protein surface, I modified the program used for surface generation to produce an initial surface at a very coarse resolution, typically 1.6 Å. A final, higher-resolution surface grid (Figure 10c) is produced from this by examining the region of the initial surface and of the points inside the protein which border the initial surface; points in this region are marked as filled or not filled by the protein, and once again those points which are filled but have unfilled neighbors are marked as part of the surface.

The surface was used to develop a scoring scheme that approximated the overlap of a surface of the positive image with some thickness with a ligand surface of the same thickness. This overlap volume had been

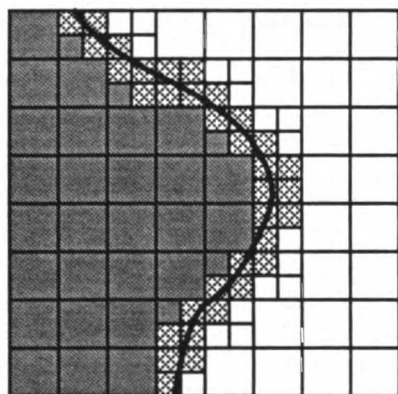
Figure 10: A two-dimensional illustration of the representation of a molecular surface on a grid.



(a) An example region of space showing a surface (outlined in black) and corresponding regions of filled and empty grid cells.



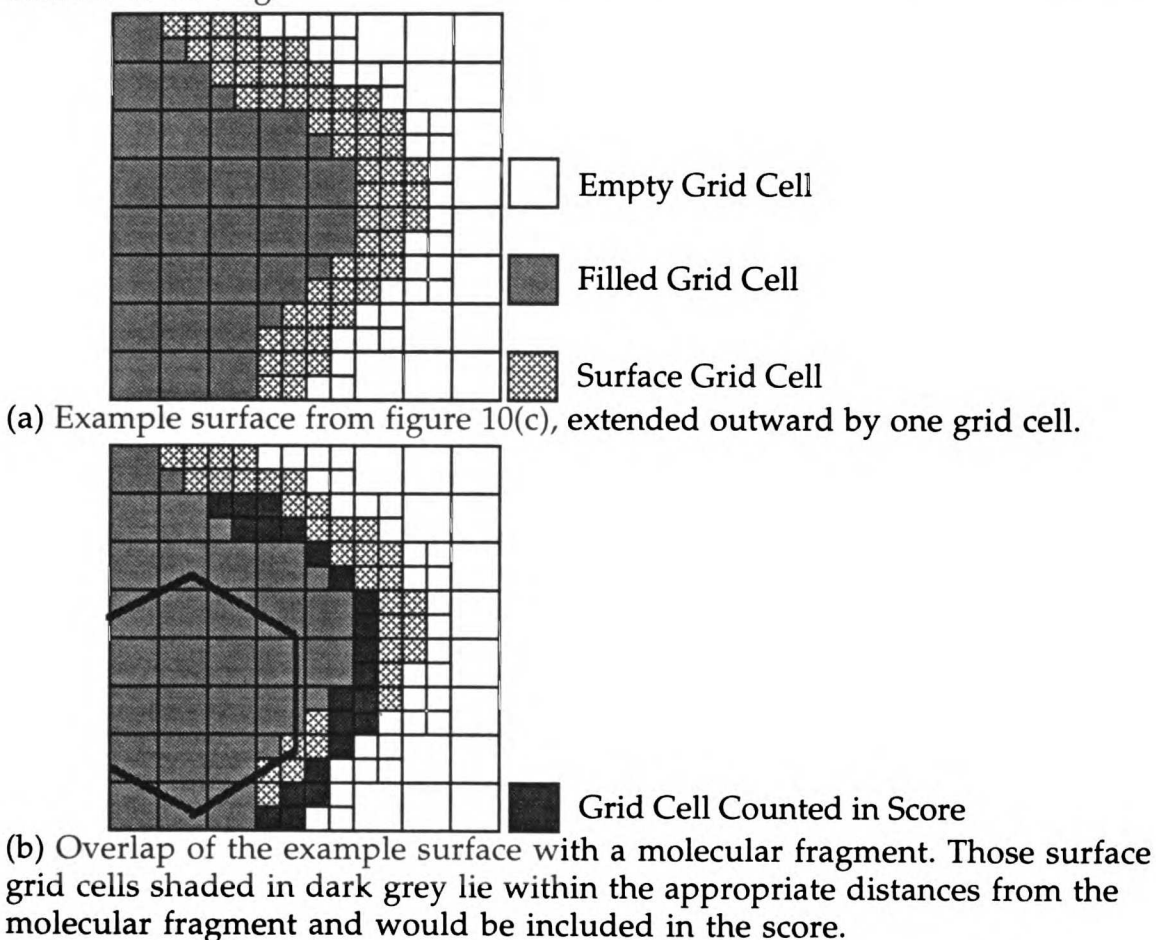
(b) The same region with surface cells indicated by a different pattern.



(c) The same region again, here with the surface region divided into smaller grid cells which have been marked empty, filled, or surface as appropriate.

calculated analytically and used to measure shape similarity by Masek *et al.* (Masek, Merchant et al. 1993). To add thickness to the grid representation of the surface, I modified the surface creation program to add additional layers of surface cells by converting unfilled cells adjacent to surface cells to surface (Figure 11a); this process may be repeated to vary the surface thickness. A new scoring routine was added to DOCK to count, for each atom, the number of surface points which fall between the van der Waals radius and the van der Waals radius plus the surface thickness away (Figure 11b). Points which overlap the surface of more than one atom are counted only once, and the total number of points in the overlap region is used as the score for the ligand orientation.

Figure 11: Illustration of the use of grid-based approximations to a molecular surface in scoring.



To test whether the surface scoring method indeed located orientations with good overlap with the target surface, the first 100 compounds of the ACD subset used for testing the other scoring schemes were docked to the BPTI spheres using only surface-based scoring. N-(3-aminopropyl)cyclohexylamine was also docked to BPTI in single mode with surface scoring.

Results

DOCK Experiments in the Dopamine D2 Pharmacophore

Single Mode Docking of Molecules from the Literature

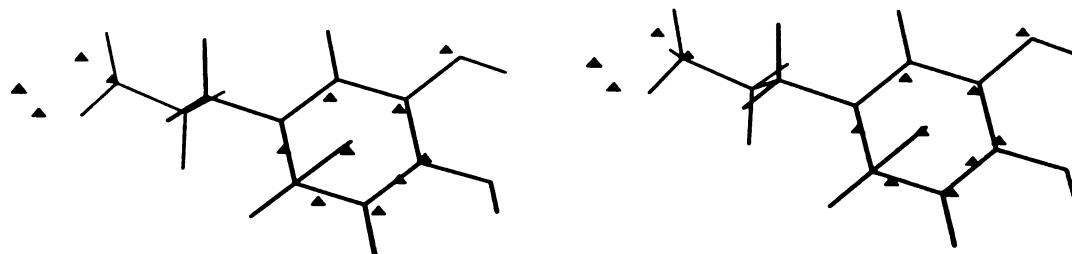
When molecules from the literature were individually docked to the pharmacophore, the rms deviations (Table 6) between the molecules as aligned to the pharmacophore by hand and the closest orientations produced by DOCK ranged from 0.129 Å to 1.790 Å. Only three of the ten molecules had rms deviations greater than 1 Å in their best docked orientations. These three molecules, isoapomorphine, 2R-4-hydroxy-2-(di-*n*-propylamino)indan, and 2S-4-hydroxy-2-(di-*n*-propylamino)indan, matched the pharmacophore poorly in hand alignment as well (Figure 12). The remaining seven molecules had orientations which were visually similar to those produced by hand alignment. For six of the ten molecules, the orientation with the best electrostatic score had a lower rmsd from the hand-aligned orientation than did the orientation with the best score by the atom-sphere proximity measure (Table 6). Isoapomorphine and S(-)-3-(3-hydroxyphenyl-N-*n*-propyl)piperidine both achieved orientations in which many of their atoms were very close to spheres; these orientations had both good proximity scores and low rmsd values. The orientation of 4aR,10bR-7-hydroxy-4-*n*-propyl-1,2,3,4,4a,5,6,10b-octahydrobenzo[f]quinoline with the best electrostatic score matched the pharmacophore, but in a different way than that described by Manallack and Beart (Figure 13), so its best electrostatic orientation had a

Table 6: RMS deviations between hand and DOCK alignments.

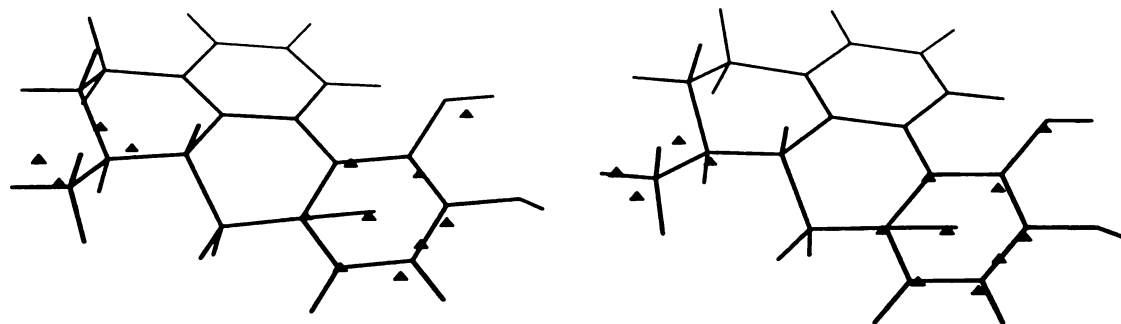
Root-mean-square deviations between atom positions of molecules aligned to the pharmacophore by hand and orientations of the same molecule produced by DOCK. For each molecule, rms values are given for the orientation with the best electrostatic score, the orientation with the best value of the proximity measure, and the orientation with the best rms of all orientations generated in the DOCK run.

Compound	RMS of orientation with best electrostatic score (Å)	RMS of orientation with best proximity score (Å)	Best RMS (Å)
Dopamine	0.355	2.454	0.258
S(+) apomorphine	0.800	3.014	0.768
Isoapomorphine	4.259	3.833	1.552
LY 156525	0.434	0.830	0.363
2R-4-hydroxy-2-(di- <i>n</i> -propylamino)indan	2.163	2.255	1.790
4aR,10bR-7-hydroxy-4- <i>n</i> -propyl-1,2,3,4,4a,5,6,10b-octahydrobenzo[f]quinoline	3.018	2.919	0.805
2R-5-hydroxy-2-(di- <i>n</i> -propylamino)tetralin	3.525	1.678	0.833
S(-)-3-(3-hydroxyphenyl-N- <i>n</i> -propyl)piperidine	2.949	1.518	0.796
2S-4-hydroxy-2-(di- <i>n</i> -propylamino)indan	1.487	5.487	1.487
4aS,10bS-7-hydroxy-4- <i>n</i> -propyl-1,2,3,4,4a,5,6,10b-octahydrobenzo[f]quinoline	0.257	3.866	0.129

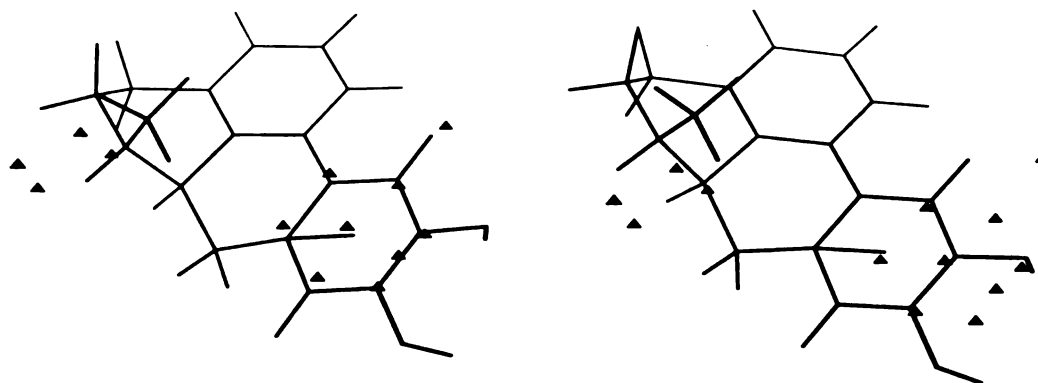
Figure 12: Hand and DOCK alignments to the dopamine D2 pharmacophore.



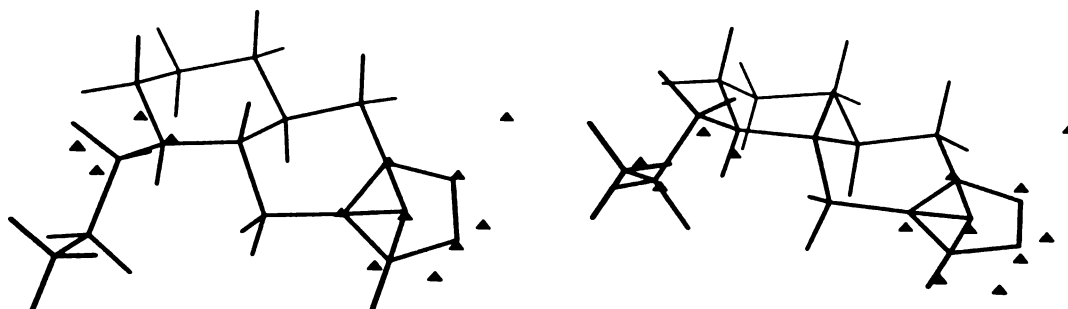
(a) Dopamine. Left, hand alignment; right, DOCK alignment.



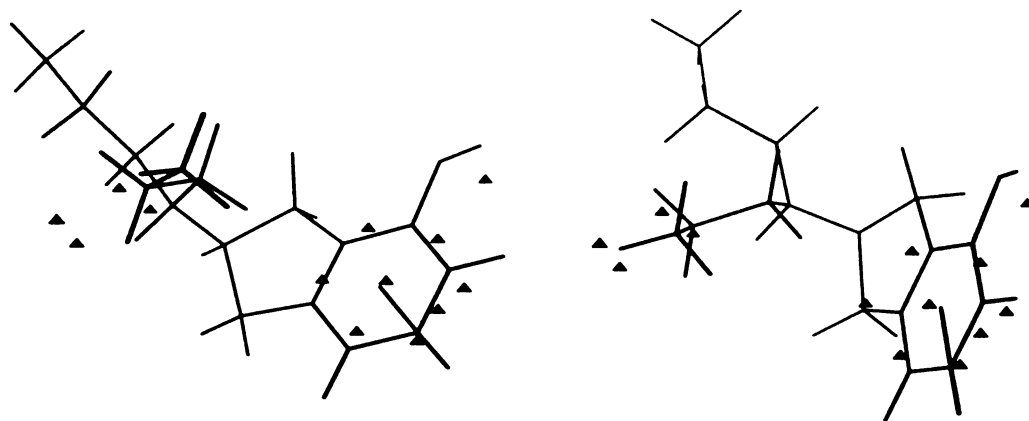
(b) S(+)-apomorphine.



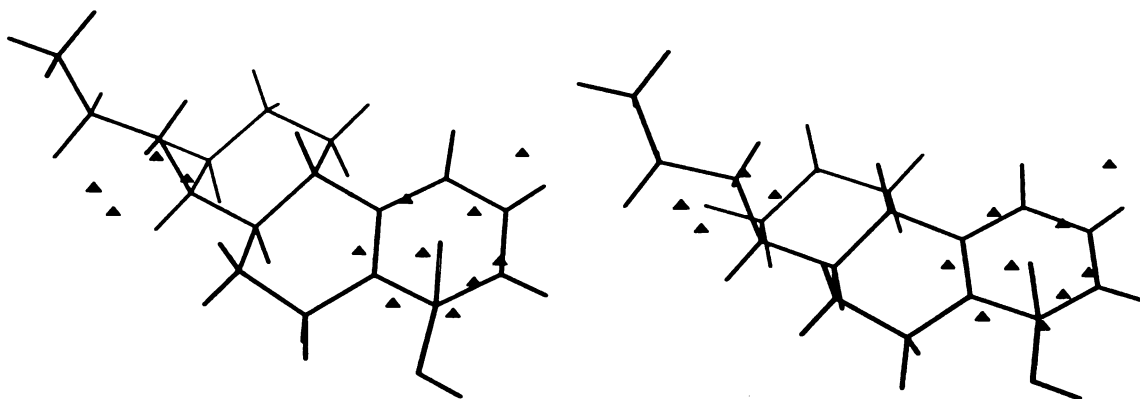
(c) Isoapomorphine.



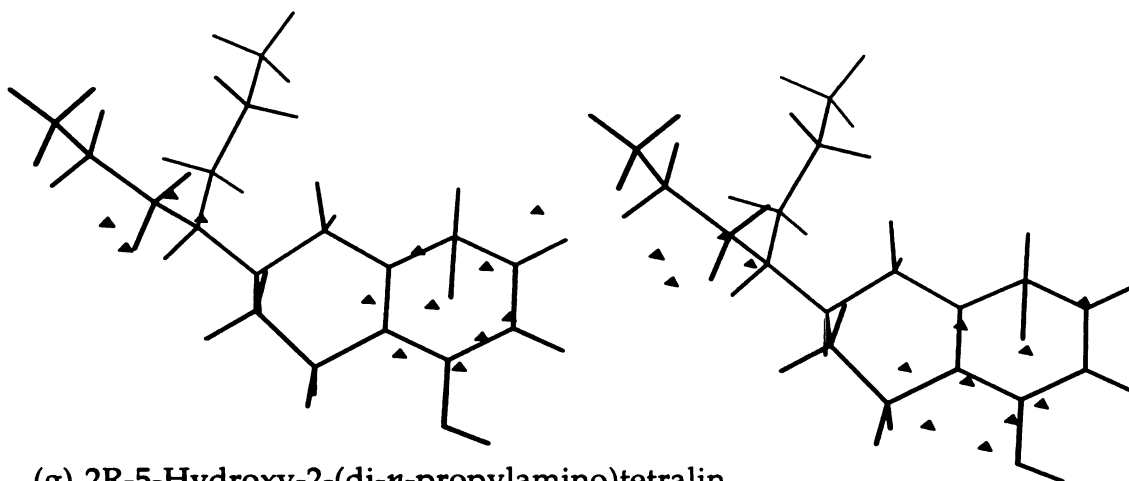
(d) LY156525.



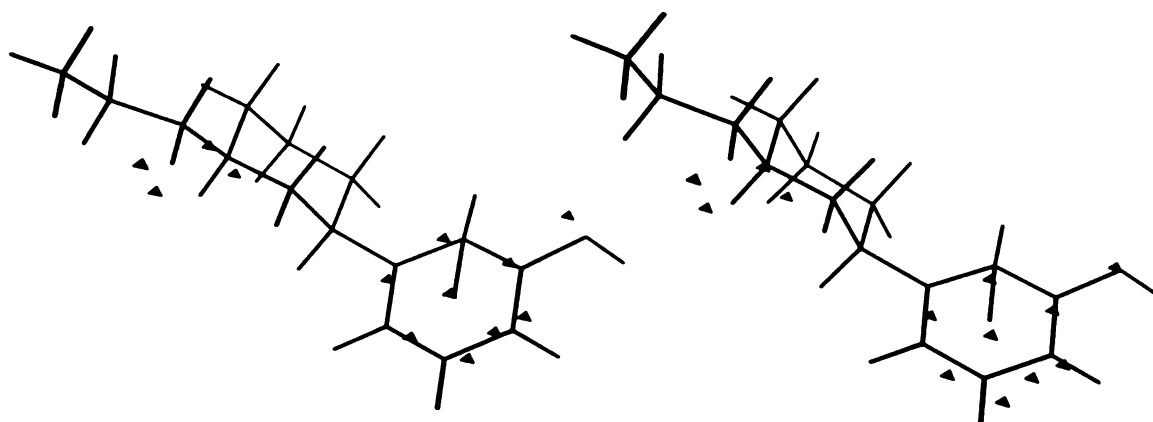
(e) R-4-hydroxy-N,N,-di-*n*-propylaminoindan.



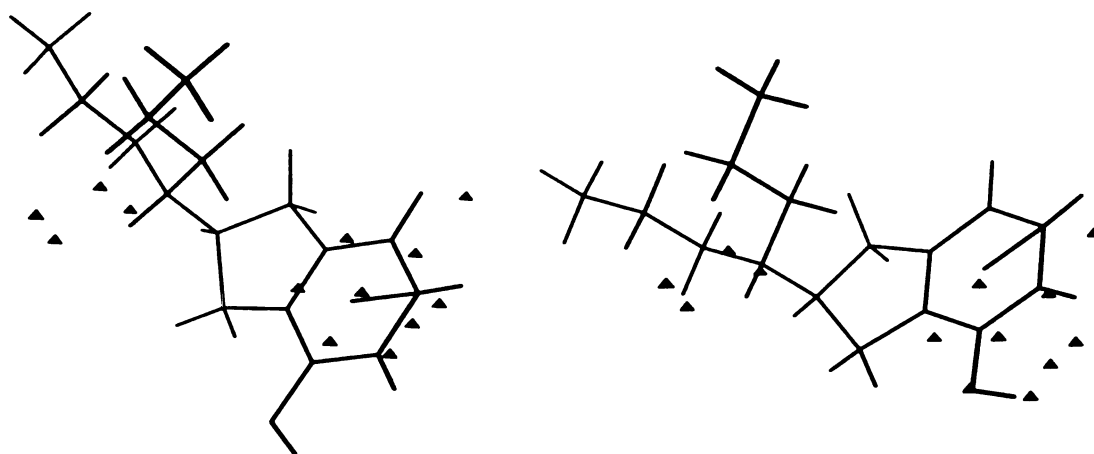
(f) 4aR,10bR-7-hydroxy-4-*n*-propyl-1,2,3,4,4a,5,6,10b-octahydrobenzo[f]quinoline.



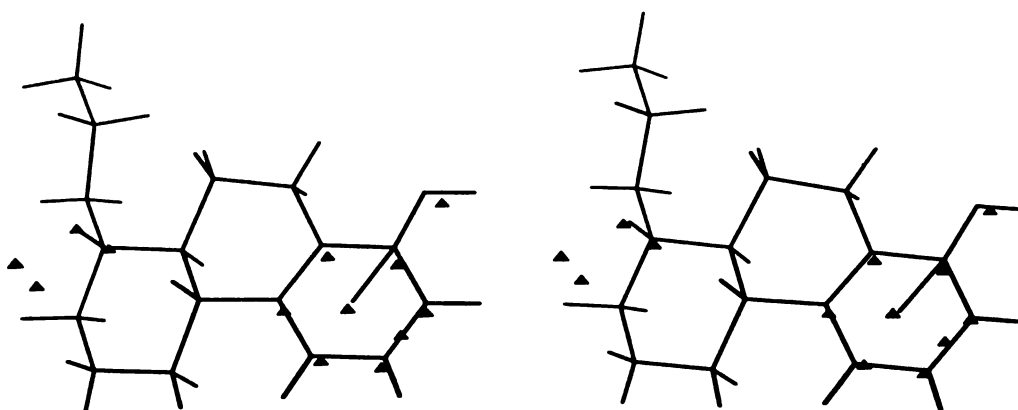
(g) 2R-5-Hydroxy-2-(di-*n*-propylamino)tetalin.



(h) S-(-)-3-(3-Hydroxyphenyl)-N-*n*-propylpiperidine.

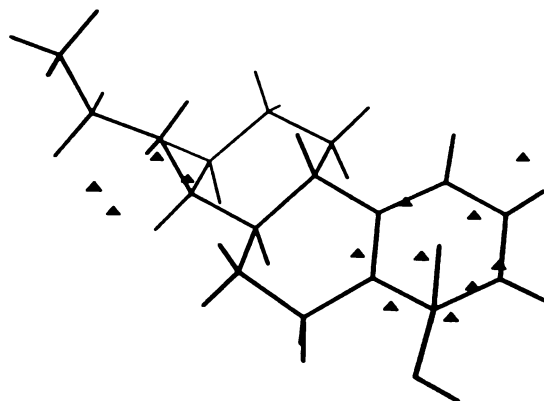


(i) 2S-4-Hydroxy-2-(di-*n*-propylamino)indan.

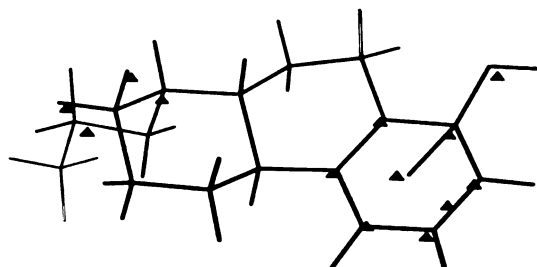


(j) 4aS,10bS-7-Hydroxy-4-*n*-propyl-1,2,3,4,4a,5,6,10b-octahydrobenzo[f]quinoline.

Figure 13: Literature alignment of 4aR,10bR-7-hydroxy-4-*n*-propyl-1,2,3,4,4a,5,6,10b-octahydrobenzo[*f*]quinoline and orientation with top electrostatic score.



(a) Alignment used by Manallack and Beart(1988).



(b) Orientation with the best electrostatic score.

higher rmsd than its best proximity orientation. DOCK generated an orientation similar to the hand-aligned orientation in nearly all cases. In this system, the electrostatic scoring scheme was slightly better than the proximity-based scoring scheme at choosing orientations which resembled the hand alignment.

Search Mode Docking to the Pharmacophore: Electrostatic + Geometric Score

The atom-based geometric score dominated the total score when the D2 test database was docked to the pharmacophore using the sum of electrostatic

and geometric scores, despite the fact that the per-atom cap used for the electrostatic score was -1000 while that used for the atom-based geometric score was -50. The average total score for the 293 compounds in the database was -1189; the average contribution from the geometric score was -1098 while the average electrostatic contribution was only -91. For the 11274 compounds from the MDDR database docked using the same conditions, the average total score was -1184, with -1143 contributed by the atom-based geometric score and only -41 coming from the electrostatic score. Under these conditions, weighting the two scores equally did not produce equal contributions to the total score. Since DOCK saves only the orientation of each compound with the best total score, this particular set of conditions for combining the two types of scores favors molecules which do well in the atom-based geometric scheme regardless of whether they have any electrostatic similarity to the pharmacophore.

A graph (Figure 14) of the percentage of active compounds found vs. the percentage of the D2 test database which scored as well or better is close to a line with slope equal to 1, indicating that active compounds did not score better than their counterparts in the database which did not have D2 activity. In fact, only 6% of the active compounds scored among the top 10% of the database, less than the 10% which would be expected if compounds were chosen randomly. The equivalent graph (Figure 15) for the MDDR database shows that 20 of 43 active compounds scored in the top 10% of the database, a four-fold increase over random selection. While the difference in performance between the two databases appears large, it should be noted that many of the inactive compounds in the D2 test database match some aspects of the pharmacophore and may have scored better as a result than the unrelated compounds which make up the bulk of the MDDR.

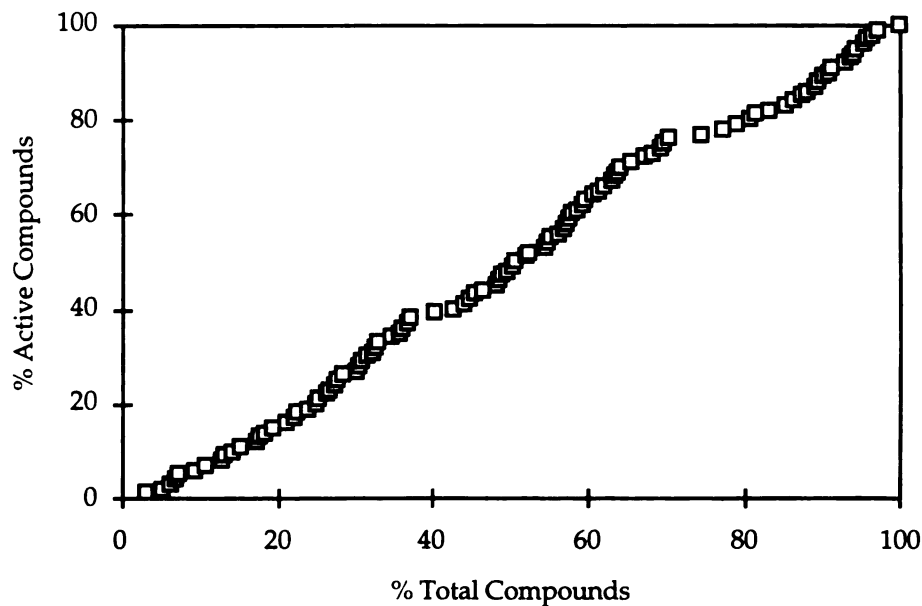


Figure 14: Percentage of active compounds found in the D2 database using the sum of the electrostatic score and the atom-atom score capped at -50.

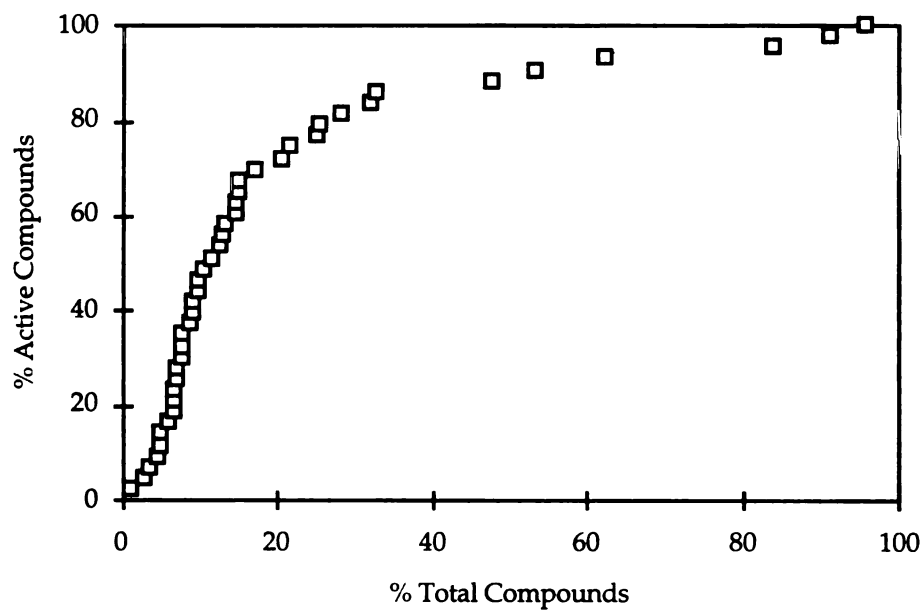


Figure 15: Percentage of active compounds found in the MDDR database using the sum of the electrostatic score and the atom-atom score capped at -50.

Search Mode Docking to the Pharmacophore: Electrostatic Score

When the D2 test database was docked to the pharmacophore using only the electrostatic score, the average score for a molecule in the database was -200; it had been only -91 when the sum of scores was used. Since larger absolute values mean "better" scores, the electrostatic scores for these molecules improved significantly when electrostatics alone determined the best-scoring orientations. The average score of a molecule in the MDDR database when docked using only the electrostatic score was -139, an increase in absolute value from -41. The electrostatic score alone was thus able to find orientations which were more favorable electrostatically than those found by the sum of scores.

Compared to the graph (Figure 14) of actives found using the sum of electrostatic and geometric terms, the graph for electrostatic-only scoring (Figure 16) shows a slight increase in the rate at which active compounds are found relative to other molecules in the database. 15% of the active compounds ranked in the top 10%, 1.5 times what would be expected with random selection. The graph of the scores of active molecules vs. the percentage of database molecules found in the MDDR (Figure 17) shows an increase as well; 35 of 43 actives scored in the top 10%, an 8-fold enrichment over random selection.

Studying the Effect of Varying the Geometric Score Cap

The electrostatic scores of the compounds in the D2 test database when docked using electrostatics alone were plotted against the geometric scores calculated for the same orientations using a varying geometric score cap; active and known inactive compounds were indicated. When the geometric score cap was -50 (Figure 18) or -400 (Figure 19), the compounds were scattered. At a cap value of -1000 (Figure 20), a group of compounds with

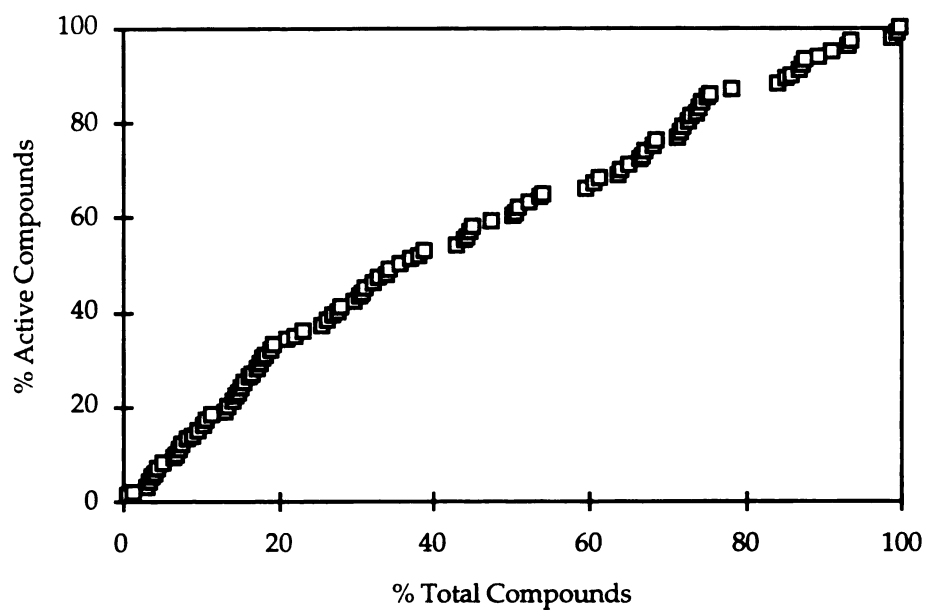


Figure 16: Percentage of actives found by searching the D2 database using electrostatic scoring alone.

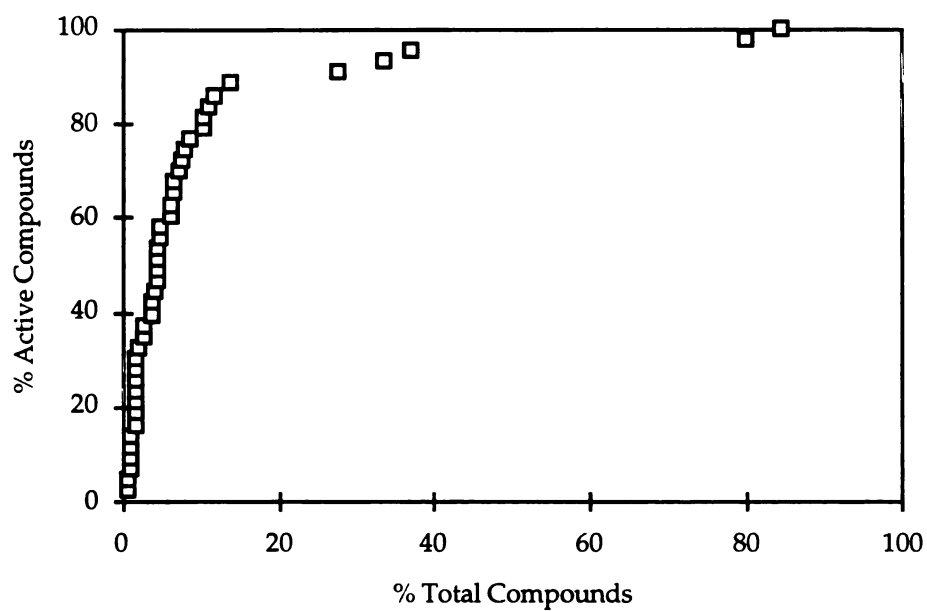


Figure 17: Percentage of actives found by searching the MDDR database using electrostatic scoring alone.

similar geometric scores begins to emerge. There is a very distinct band of compounds with geometric scores of about -90,000 in the graph (Figure 21) corresponding to a cap of -10,000; with a cap of -25,000 (Figure 22), the same group of compounds all score between -225,000 and -227,000. Examination of these compounds (Figure 23) revealed that all of them aligned to the pharmacophore by placing a nitrogen near the sphere derived from the N of the pharmacophore molecules, an aromatic ring over the pharmacophore ring, and often an oxygen over the pharmacophore O. Since these results seem to indicate that a larger score cap is required for accuracy in scoring similar overlays, a cap of -10,000 was used for subsequent experiments.

None of the graphs (Figures 18-22) produced by plotting the atom-based geometric score against the electrostatic score showed a division of active

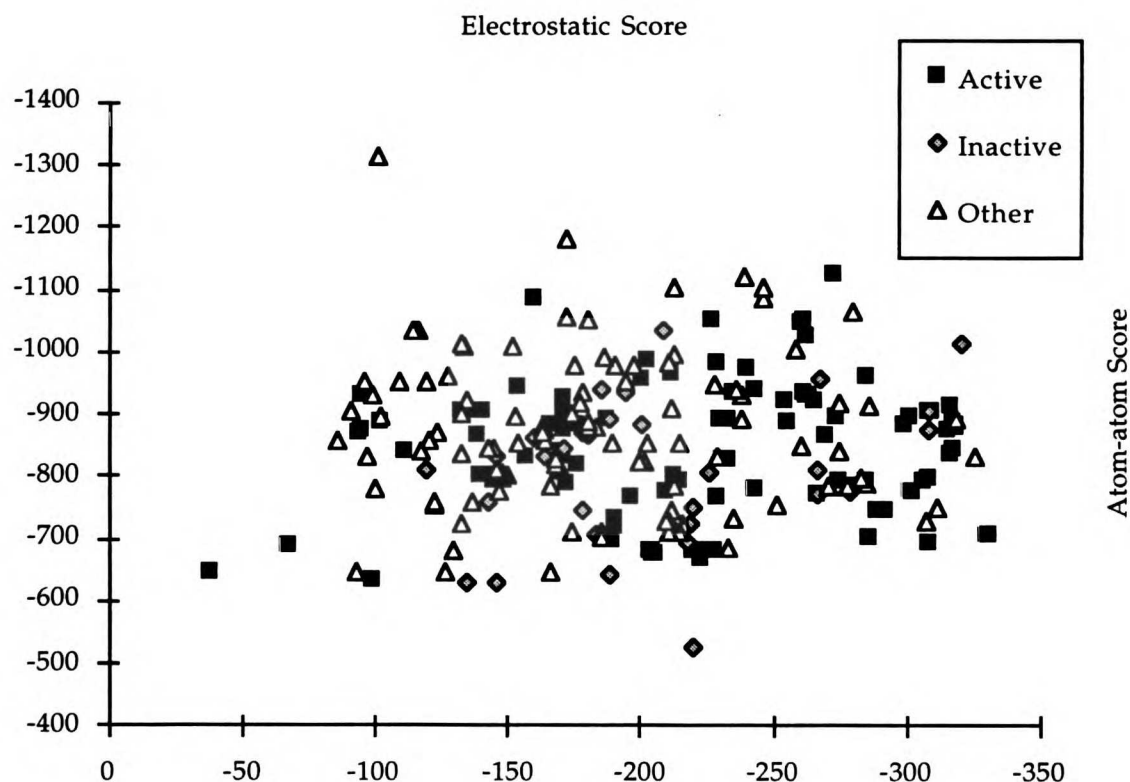


Figure 18: Atom-atom (with cap -50) versus electrostatic scores for docking of the D2 database.

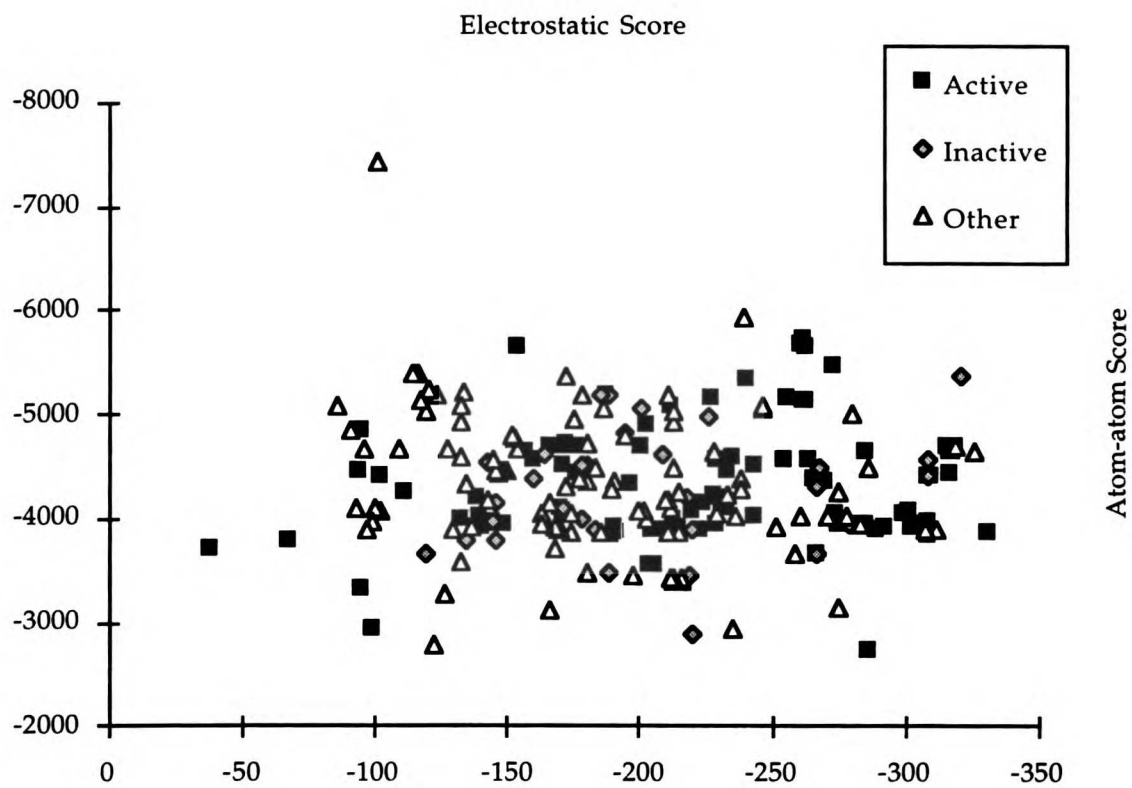


Figure 19: Atom-atom (with cap -400) versus electrostatic scores for docking of the D2 database.

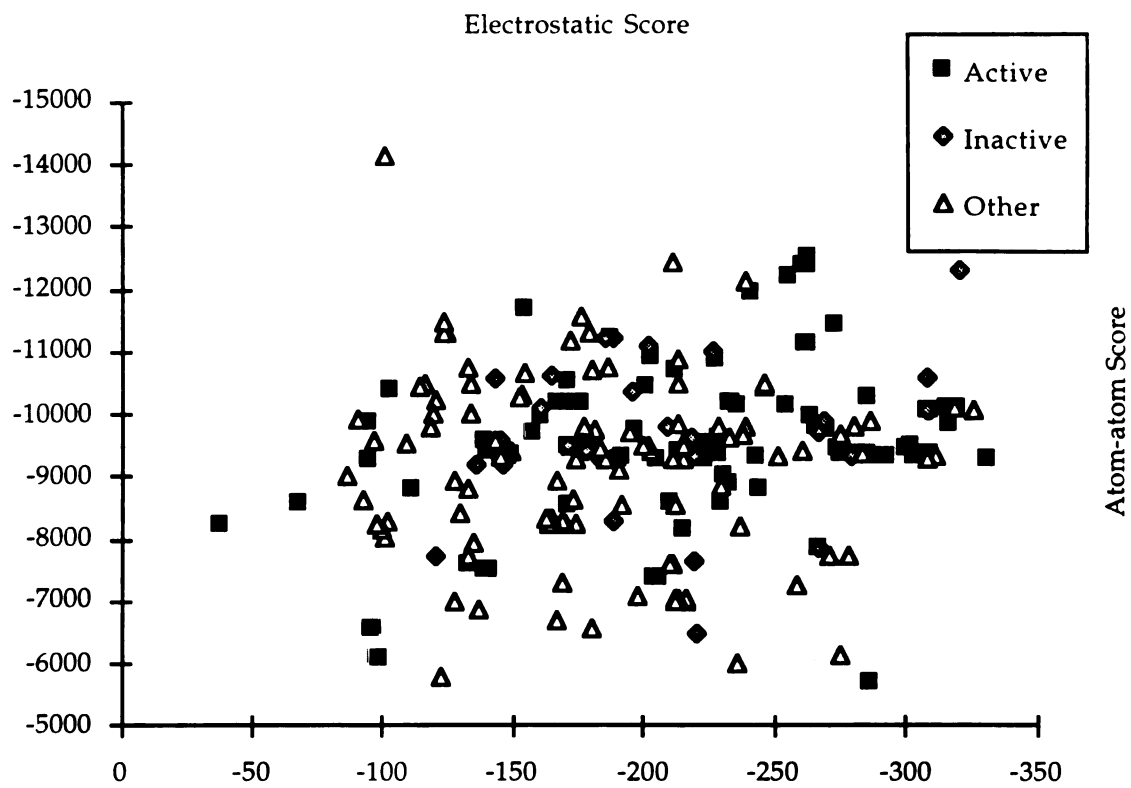


Figure 20: Atom-atom (with cap -1000) versus electrostatic scores for docking of the D2 database.

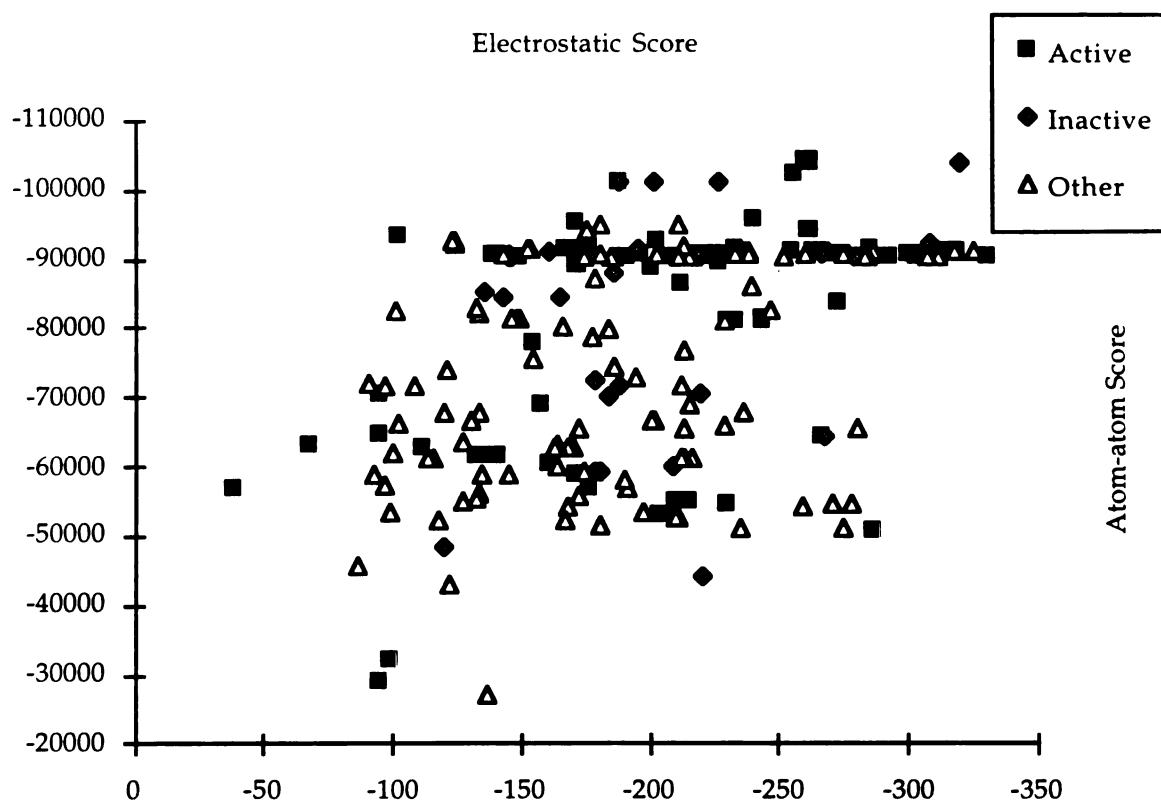


Figure 21: Atom-atom (with cap -10000) versus electrostatic scores for docking of the D2 database.

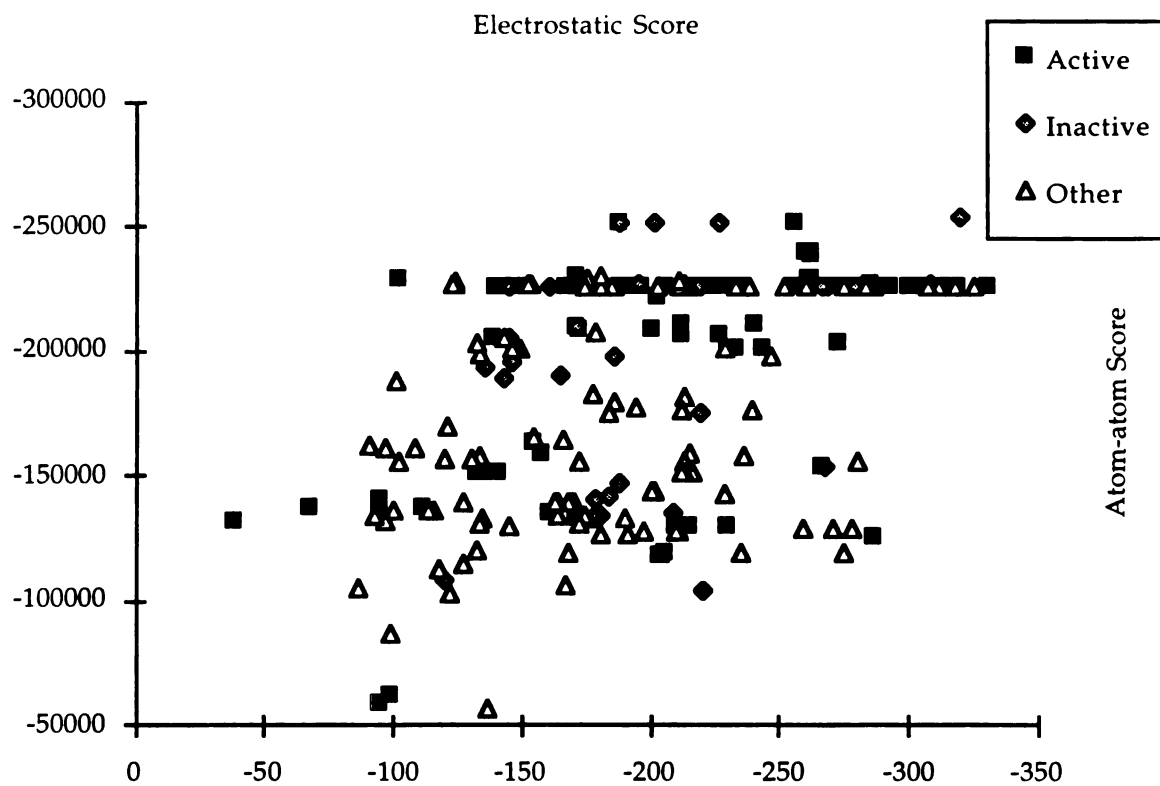
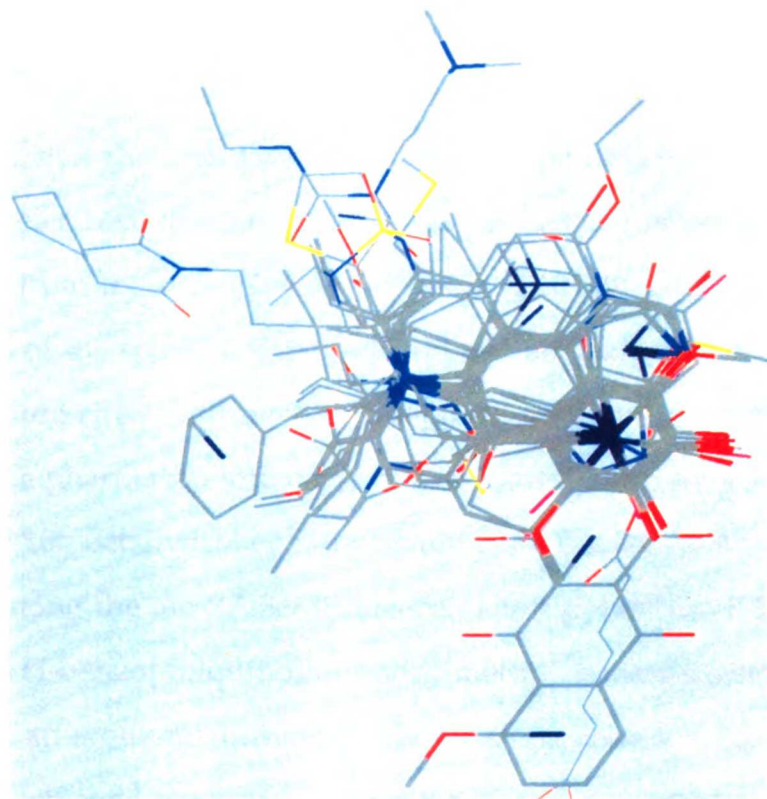


Figure 22: Atom-atom (with cap -25000) versus electrostatic scores for docking of the D2 database.

Figure 23: Molecules with similar geometric scores in their DOCK alignments.



compounds and inactive compounds into different regions.

Search Mode Docking of the D2 Database Using a Weighted Combination of Scores

Plots of the scaled geometric component of the combined score against the electrostatic component (Figures 24 and 25) once again showed no clustering of active and inactive components into particular regions of the graph. (Note that the orientations of these compounds were those with the best sum of electrostatic and scaled geometric scores.) When the top-scoring compounds from the run using a scale factor of 0.002 were examined visually, however, 26 of the top 27 compounds aligned an aromatic ring to the pharmacophore ring, a nitrogen to the pharmacophore nitrogen and an oxygen to the pharmacophore oxygen. 4aR,10bR-7-hydroxy-4-*n*-propyl-1,2,3,4,4a,5,6,10b-octahydrobenzo[f]quinoline, Figure 26a, is an example. The exception among the top 27, 3a(S)trans-5 α -(Dipropylamino)-2,3,3a,4,5,6-hexahydro-1H-benzo[de]quinolin-2-one-4-methylbenzenesulfonate (Figure 26b) aligned an aromatic nitrogen to the pharmacophore oxygen. The same 27 compounds achieved top scores in the DOCK run using a scale factor of 0.001, but in a slightly different order. Although the active compounds did not cluster among molecules which scored well by both measures, the total score was useful in locating compounds which overlaid the pharmacophore well.

DOCK as a Tool for Pharmacophore Generation

Identifying Labeled Matches

When the program lablscan was used to search the orientations generated by docking the remaining 19 molecules used as test cases for DISCO (Martin, Bures et al. 1993) to molecule I, a set of matched spheres was found which was common to all but three of the compounds. In this common match, the basic nitrogen atoms, the hydrogen-bond acceptor groups, the

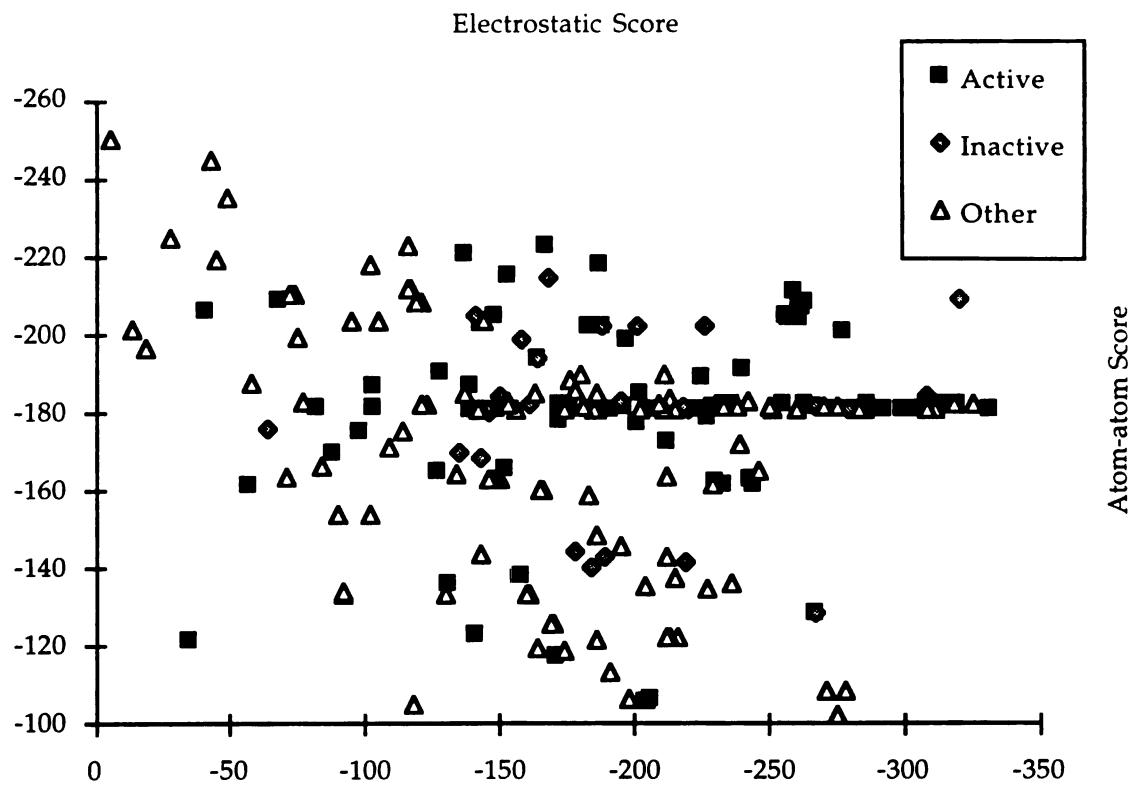


Figure 24: Atom-atom versus electrostatic score for D2 docking using a scale factor of 0.002.

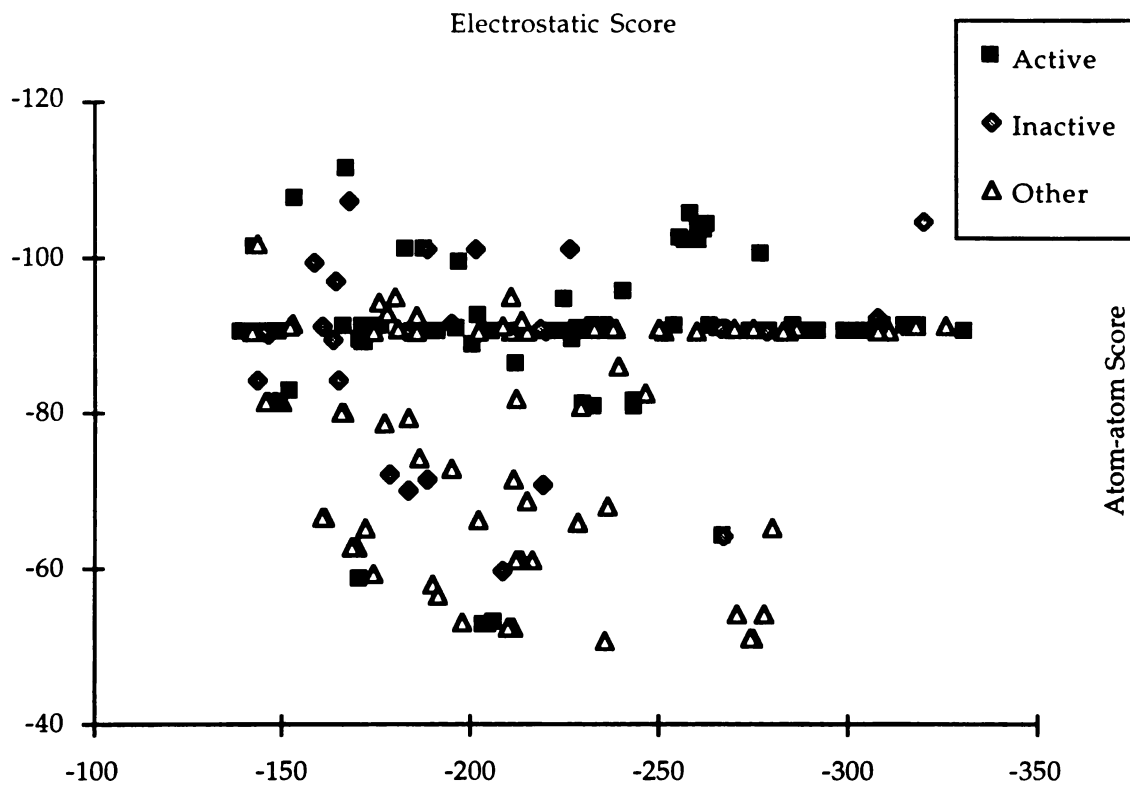
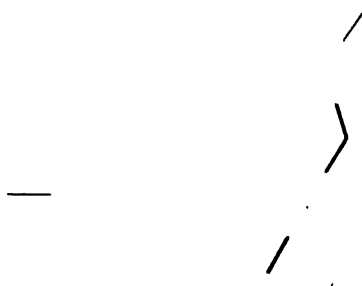


Figure 25: Atom-atom versus electrostatic score for D2 docking using a scale factor of 0.001.

Figure 26: Examples of top-scoring compounds from D2 docking using scaled scores.



(a) Top-scoring compound overall.



(b) The only compound among the top 27 which did not share a common alignment to the pharmacophore.

hydrogen-bond donors on the aromatic ring, and the points representing an H-bond acceptor associated with that donor were aligned (Figures 27 and 28). Molecules II, XIII and XVIII did not share this match under the original sampling conditions (bin widths of 0.3 Å and bin overlaps of 0.1 Å) but had orientations including the match when the bin widths used were increased to 0.6 Å and the bin overlaps to 0.15 Å. The docked orientations were compared visually to the orientations pictured in the DISCO paper. Most of the molecules were in similar relative orientations although the initial matched orientation of XIV was quite different from that shown. Docking XIV using the larger bin sizes produced an orientation which was more like the one shown in the paper but still not very close. Overall, DOCK reproduced the pharmacophore found by DISCO, if not its exact details.

Docking Using Only Labeled Atoms

Using only the labeled atoms reduced the number of internal distances involved in docking and therefore the amount of time required. The distance sets were so much smaller that larger bins and overlaps were required for each compound to have at least some matches. As with the full sets of spheres and atoms, matching was carried out without considering the labels; when the matches were screened using labscan, only ten of the nineteen compounds (IV, V, VI, VII, VIII, IX, X, XI, XII, and XVI) had a common orientation relative to I which matched a minimal set of points from DISCO. Examination of the molecules which did not share this orientation revealed that they included the labeled spheres and that the inter-sphere distances should have allowed them to match. The failure to find the matches in this case is a result of the way the matching algorithm employed in DOCK 3.0 and DOCK 3.5 uses the distance sets in generating the match. DOCK constructs matches by examining the distances from a seed sphere to the remaining

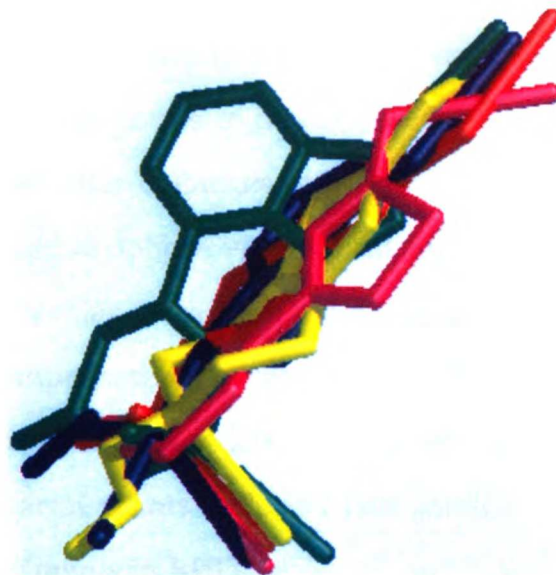


Figure 27: Molecules aligned by DOCK using lablscan.

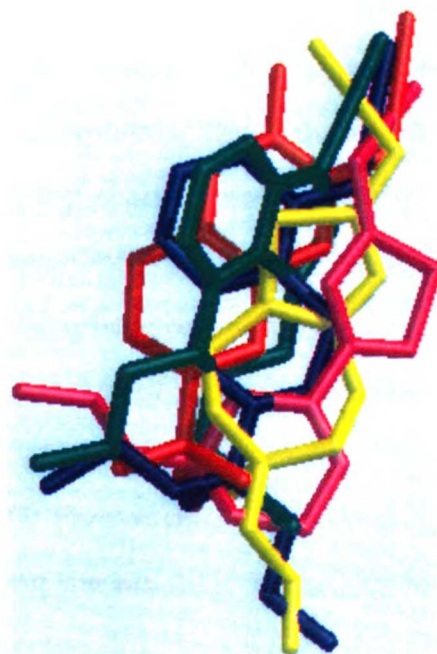


Figure 28: Another view of the aligned molecules.

spheres and from a seed atom to the remaining atoms, trying each atom and sphere as the seeds. Matching begins with the longest distance from the seed to another point and proceeds by adding shorter distances. In no case will the total number of distances examined exceed one less than the user-specified maximum number of points to be used for matching; matches which require examination of smaller distances will not be found. Since orientations which matched the pharmacophore were found for all compounds when all the atoms in the molecules were used for docking, it appears that using larger sets of points can compensate for ignoring shorter distances. In this case, the current DOCK algorithm was less effective at finding the desired matches when the number of points included was small.

DOCK Experiments in BPTI

Varying Electrostatic Scoring Methods

Docking the test database constructed by clustering the ACD to the positive image of BPTI and scoring with the Coulombic electrostatic scoring scheme favored molecules which had moderate partial charges placed very close in space to charged spheres. Small charges at short distances appeared more often among the top-scoring molecules than larger partial charges at slightly longer distances. Only one molecule which scored in the top 200 had a positive charge near the sphere derived from the charged nitrogen of Lys 15, which binds in the P1 pocket of trypsin. Five molecules had positive charges near the guanidinium group of Arg 17, which is located at the edge of the binding region of BPTI. Five of the test compounds related to trypsin inhibitors scored among the top 10% of the test database (Figure 29), about twice the number expected if 10% of the database were chosen at random. When the Gaussian approximation to $\frac{1}{r}$, which levels off at about 1.4 instead of approaching infinity as r approaches zero, was used to replace $\frac{1}{r}$,

more highly charged and multiply charged molecules appeared among the top scorers. In addition, 31 of the top 200 molecules matched positive charges to Lys 15 and 12 to Arg 17. All 24 of the inhibitor-related test compounds scored among the top 50% of the test database, and 22 of them fell in the top 10% (Figure 30), a ninefold enrichment over random selection. Reducing the maximum distance at which pairs of partial charges were included in the calculation from 10 Å to 5 Å while still using the Gaussian approximation to $\frac{1}{r}$ increased the number of multiply-charged compounds among the top molecules. It also increased the number of compounds among the top 200 matching Lys 15 to 54 and Arg 17 to 19. However, only 14 of the inhibitor-related test compounds scored in the top 10% of the database (Figure 31), a five-fold enrichment over random selection, relatively poorer than scoring with a larger distance cutoff. The Gaussian approximation was generally better than the Coulombic scheme at locating compounds with partial charges in the vicinity of charges in the positive image; while using the short-range cutoff with the Gaussian function gave good scores to more compounds with charges, the fact that it did not do as well at separating inhibitors from the rest of the database indicates that the longer-range cutoff is more appropriate in this system. The electrostatic methods examined are summarized in Table 7.

Positive Docking Using Labeled Matching

Docking the test database to the positive image of BPTI using the Coulombic scoring scheme along with labeled matching produced results which were very similar to those obtained without labeled matching. Two of the 200 top-scoring compounds matched a positive charge to Lys 15, while three of them matched a positive charge to Arg 17. 7 of the compounds related to trypsin inhibitors scored among the top 10% of the test database

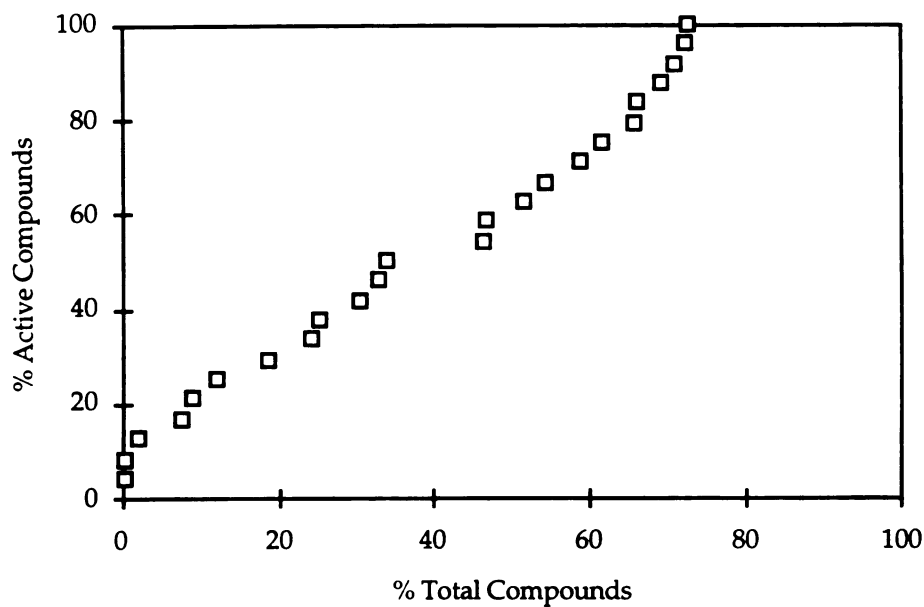


Figure 29: Inhibitor-related compounds found in the ACD subset by docking to BPTI using "standard" electrostatics.

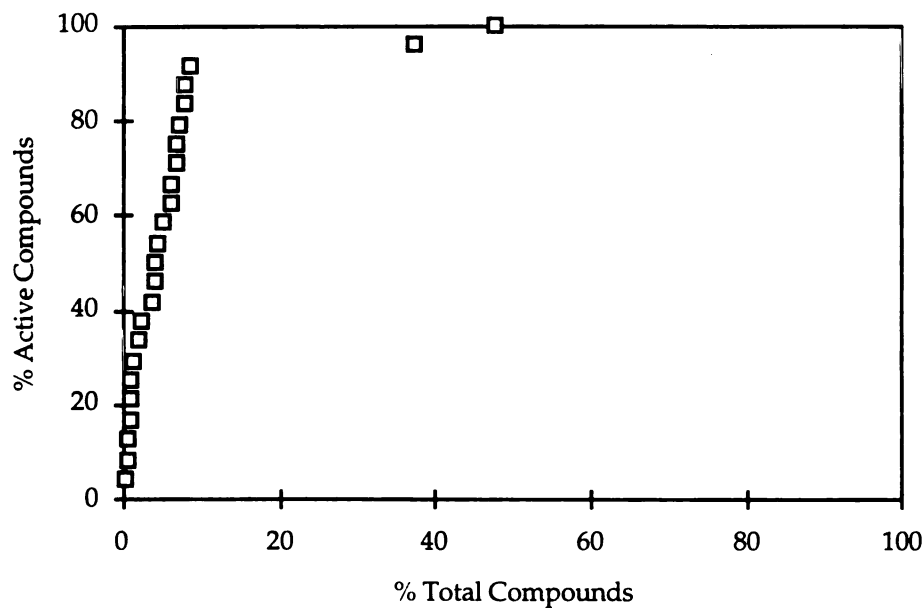


Figure 30: Inhibitor-related compounds found in the ACD subset by docking to BPTI using the Gaussian electrostatic function.

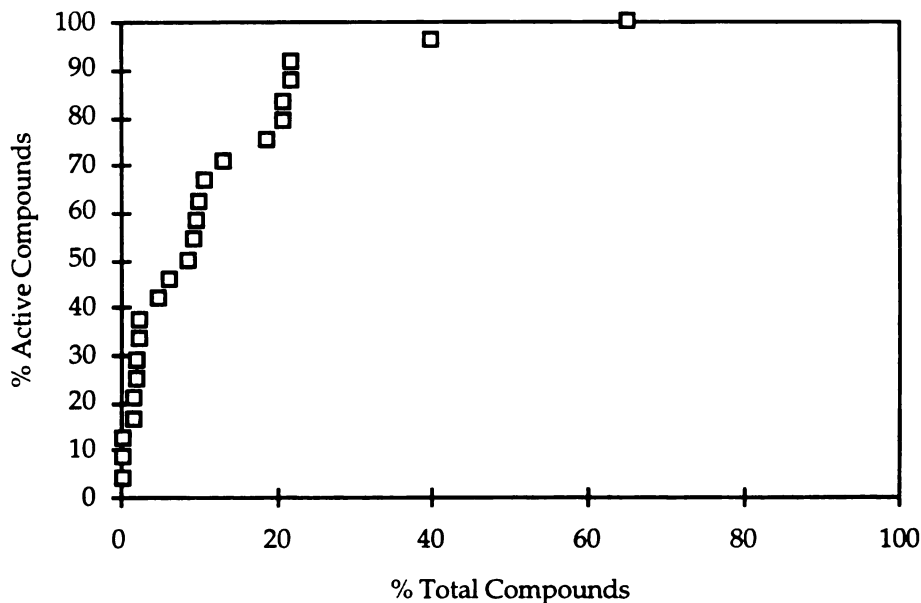


Figure 31: Inhibitor-related compounds found in the ACD subset using the Gaussian electrostatic function and a 5 Å distance cutoff.

Table 7: Summary of electrostatic scoring schemes investigated in the BPTI system.

The final two columns refer to the number of compounds scoring in the top 200 which had a positive charge located at Lys 15 or Arg 17.

Electrostatic Scoring Scheme	Scoring Function	Maximum Score Per Atom	Maximum Interatomic Distance	Matches to Lys	Matches to Arg
Coulombic	$\frac{q_i q_j}{r_{ij}}$	1000	10 Å	1	5
Gaussian	$q_i q_j \times 3$ -Gaussian approximation to $\frac{1}{r}$	1000	10 Å	31	12
Short Range Gaussian	$q_i q_j \times 3$ -Gaussian approximation to $\frac{1}{r}$	1000	5 Å	54	19

(Figure 32), similar to the performance of the Coulombic electrostatic scoring scheme alone.

Varying Geometric Scoring Methods

When the ACD subset used as a test database was docked to the positive image of BPTI and compounds were ranked by geometric score using a per-atom cap of -10,000, the structures which scored highest overlaid the target image in many places (Figure 33). However, most of them did not reflect the overall shape of any part of the inhibitor. The top 300 molecules were relatively large, averaging 30.4 nonhydrogen atoms each (Table 8). Six of 24 inhibitors and related compounds used as a test set scored within the top 10% of the test database, an enrichment of 2.5-fold over random, but the remaining compounds did not score well (Figure 34); in fact, half of them scored in the bottom 20% of the database. Docking the same database using geometric scoring with a cap of -1000 gave top-scoring compounds which appeared less closely superimposed to the target (Figure 35) and also did not reflect the shape of the inhibitor. Substituting the Gaussian approximation to $\frac{1}{r}$ for $\frac{1}{r^6}$ in the scoring function gave top-scoring compounds (Figure 36) which were similar to those obtained using the original function with a cap of -10,000. Regardless of which score cap was used, the geometric scoring scheme

Table 8: Average, minimum and maximum numbers of nonhydrogen atoms among the 300 top-scoring compounds obtained by docking the ACD-derived test database to the positive image of BPTI using variations on the geometric scoring scheme.

Scoring Scheme	Number of Nonhydrogen Atoms		
	Minimum	Maximum	Average
Standard Geometric Score	18	64	30.38
Geometric Score Divided by Number of Heavy Atoms	8	14	9.92
Geometric Score Divided by Square Root of Number of Heavy Atoms	8	30	15.14

gave the best ranking to large molecules whose many atoms matched many of the spheres in the positive image, but which did not reflect the shape of the image.

Docking the ACD subset to the BPTI positive image using the geometric scoring function (cap value -10,000) but dividing by the number of nonhydrogen atoms in each molecule dramatically reduced the size of the top-scoring compounds (Figure 37); the top 300 compounds had an average of 9.9 nonhydrogen atoms each (Table 8). Atoms in the top compounds matched the target spheres very closely. Only one the inhibitor-related compounds scored in the top 10% of the database (Figure 38), but 18 of the 24 did score in the top 50%. When the procedure was repeated but the score was divided by the square root of the number of heavy atoms, the top-scoring compounds (Figure 39) were slightly larger and their atoms did not match the target spheres as exactly. The average size of a molecule in the top 300 compounds was 15.1 heavy atoms (Table 8). Using this scheme, four of 24 test compounds scored in the top 10% (Figure 40), about a two-fold enrichment over random selection, but only 7 of the test compounds scored in the top 50%. The top scoring compounds using the normalized scheme generally reflected the shape of some part of the target image better than the compounds which scored well without normalization. The inhibitor-related test molecules scored better when normalized by the number of heavy atoms than by its square root, but it should be noted that many of the compounds in the test set are fairly small molecules. Normalizing by the square root of the number of heavy atoms represents a compromise between matching many atoms, but not necessarily the shape of the target, and tightly matching a few atoms.

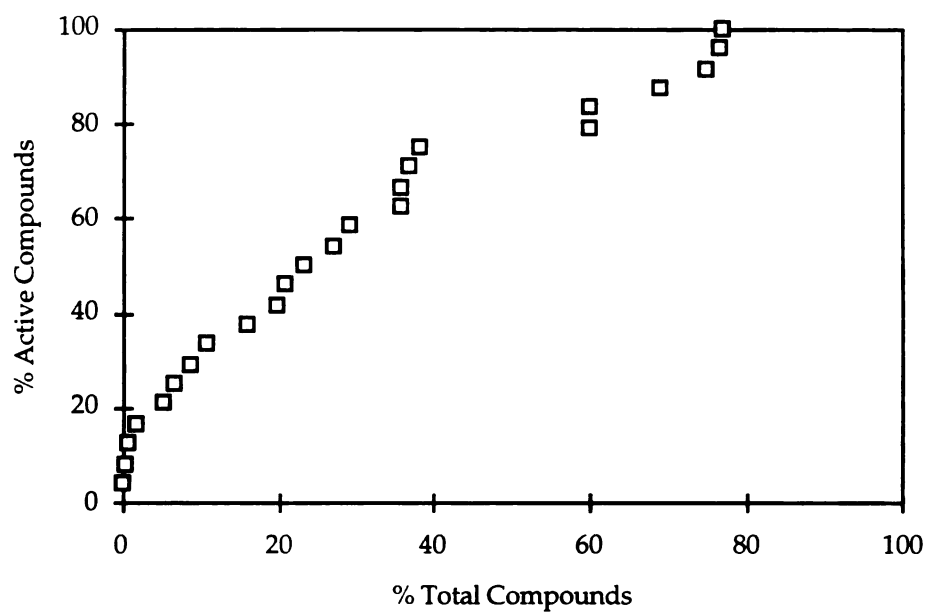


Figure 32: Inhibitor-related compounds found by docking to the ACD subset using electrostatic scoring and coloring.

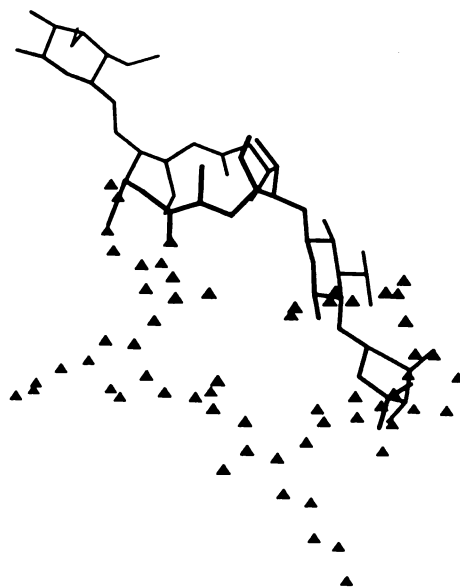


Figure 33: Compound with the best atom-atom score (cap -10000) in docking to BPTI.

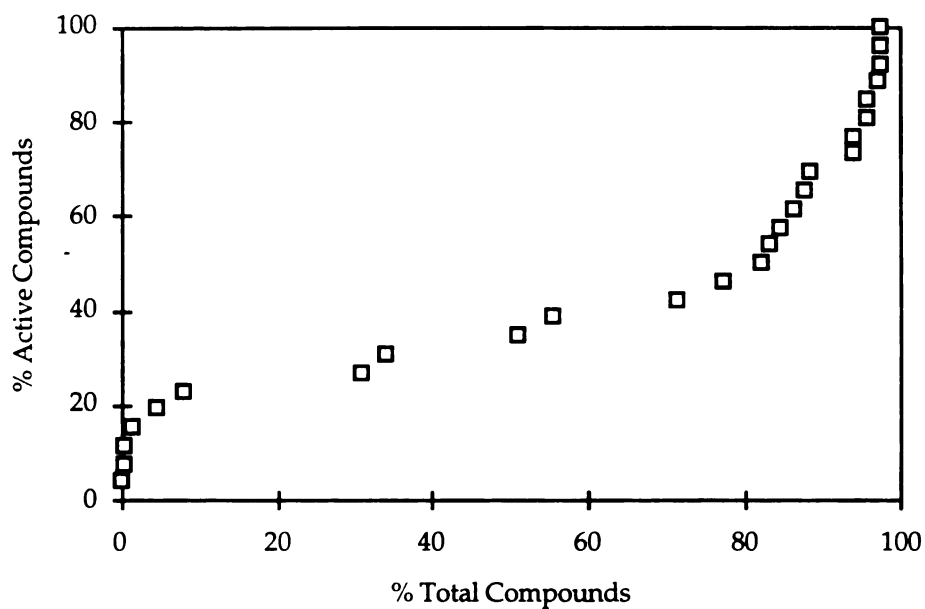


Figure 34: Inhibitor-related compounds found in docking the ACD subset to BPTI using atom-atom scoring (cap -10000).

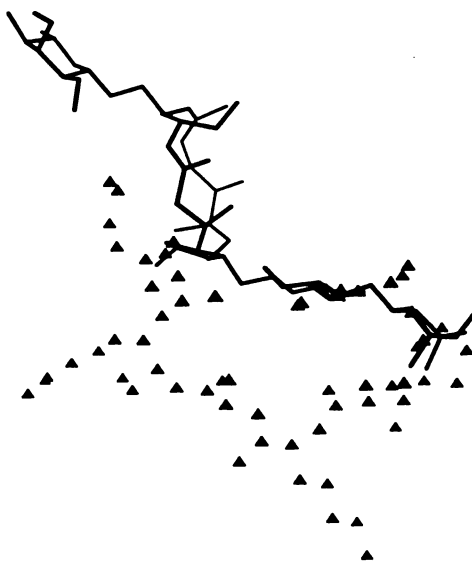


Figure 35: Compound with the best atom-atom (cap -1000) score in docking to the ACD subset.

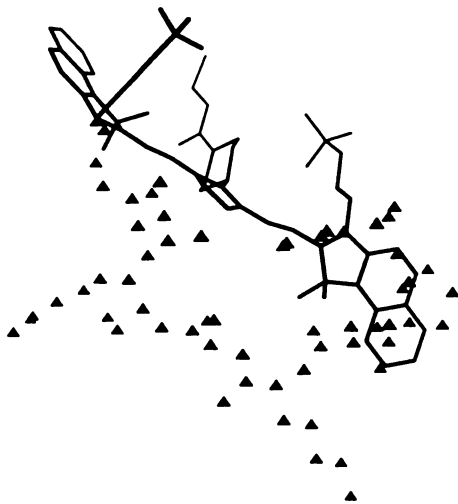


Figure 36: Top-scoring compound from docking to the ACD subset using atom-atom scoring with a Gaussian approximation to $1/r$.

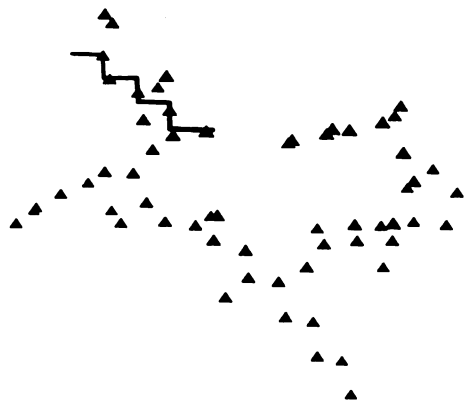


Figure 37: Top compound from atom-atom scoring normalized by the number of nonhydrogen atoms.

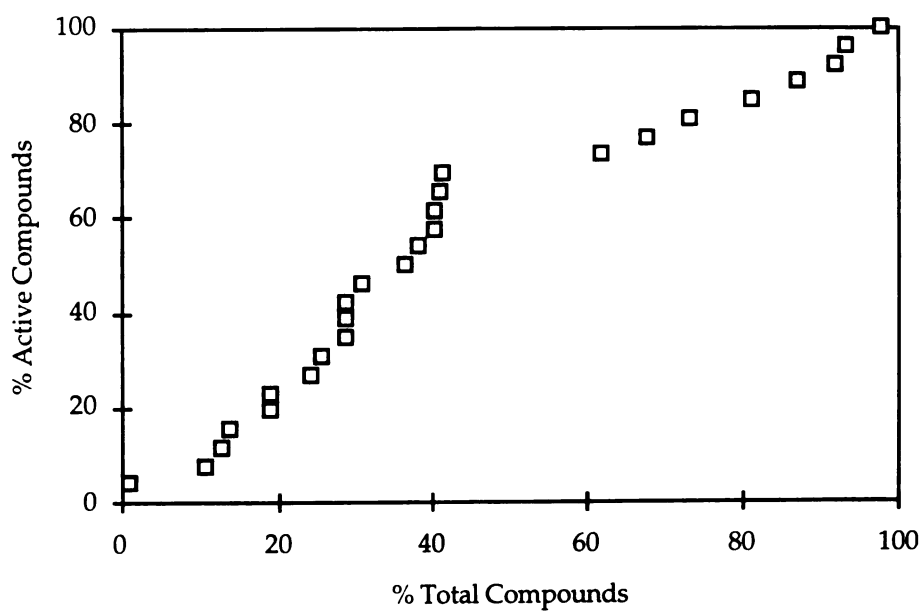


Figure 38: Inhibitor-related compounds found in docking the ACD subset to BPTI using normalized atom-atom scoring.

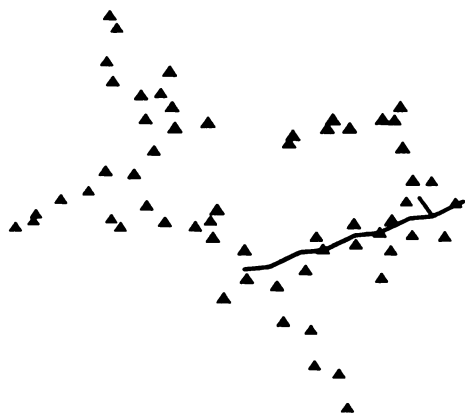


Figure 39: Top-scoring compound from docking the ACD subset to BPTI using atom-atom scoring normalized by the square root of the number of nonhydrogen atoms.

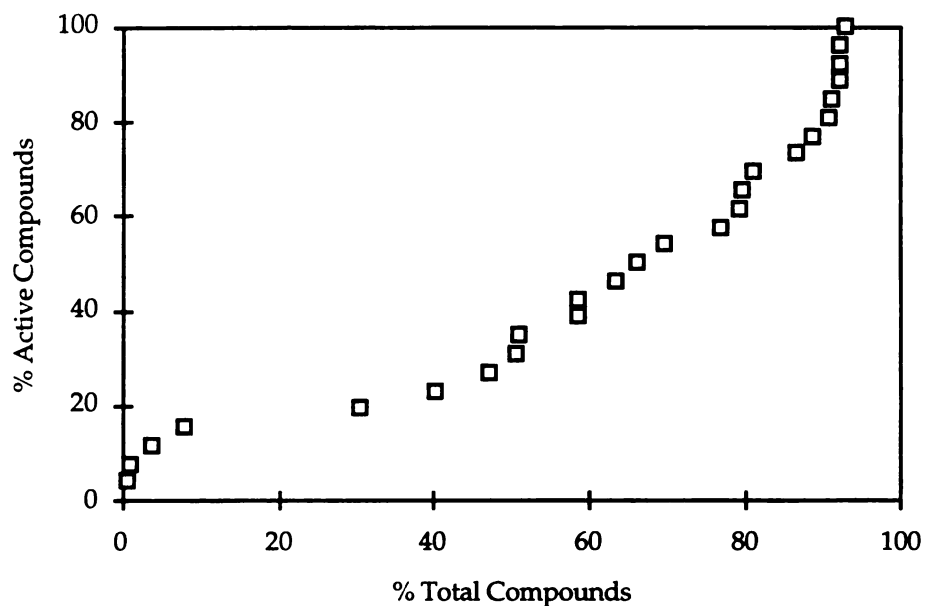


Figure 40: Inhibitor-related compounds found in docking the ACD subset to BPTI with atom-atom scoring normalized by the square root of the number of nonhydrogen atoms.

The Shape-Based Scoring Scheme

The search-mode docking of the first 100 compounds of the test database to BPTI using surface scoring proceeded very slowly, taking an average of 34 minutes per compound on a Silicon Graphics Personal Iris 4D/35. The top-scoring compounds appeared to overlap the surface well (see Figure 41). When the top 500 orientations of N-(3-aminopropyl)cyclohexylamine relative to the BPTI surface were examined visually, those with the lowest scores fell outside the box containing the surface, while those with the highest scores (Figure 42) fit nicely into the surface defined by the P1 lysine.



Figure 41: Top-scoring compound from docking with surface-based scoring.

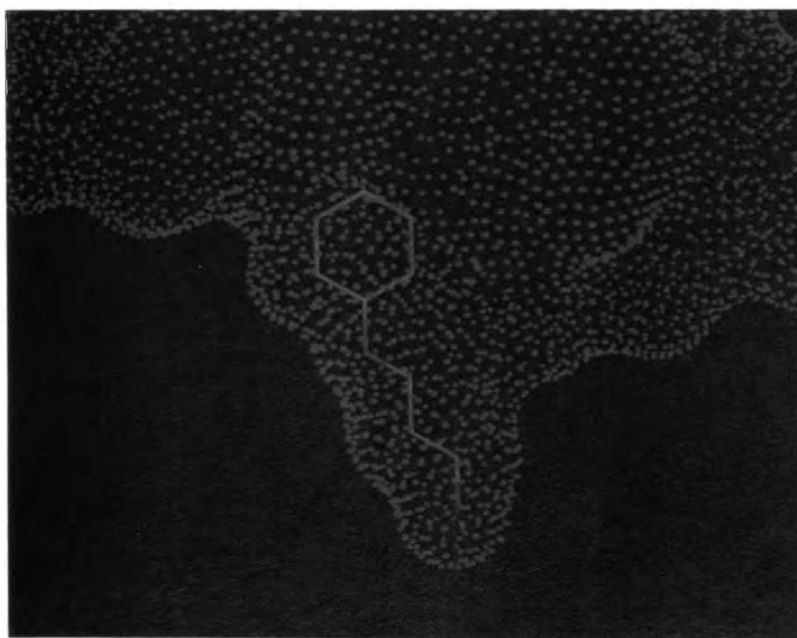


Figure 42: Top-scoring orientation from single mode docking with surface-based scoring.

Conclusions

DOCK is capable of generating orientations which appropriately overlay a pharmacophore or positive image, as is apparent from the docking of compounds of known activity to the D2 pharmacophore in SINGLE mode. Comparison of DOCK to DISCO indicates that when large distance sets are used, DOCK can reproduce a pharmacophore from a set of known molecules. However, using DOCK with reduced sets of distances may not be appropriate for this application. DOCK is potentially more useful in generating orientations relative to a positive image for large numbers of molecules so that those orientations may be examined for similarity to the target.

Electrostatic scoring using the Gaussian approximation to $1/r$ proved better than the Coulombic potential at favoring charge similarity between the candidate and target over charge proximity. In the charged trypsin inhibitor system it was also fairly successful at selecting compounds related to inhibitors from a database. In geometric scoring, a large cap value, which implies a small distance at which an atom and a target point receive the maximum score for closeness, is important for reproducibility among similar compounds. In addition, some form of normalization is necessary with this type of scoring scheme to avoid favoring large molecules excessively. The geometric scoring schemes alone did not perform particularly well at selecting known compounds from databases; their utility may lie in discriminating among molecules which are electrostatically appropriate.

Since the electrostatic scoring schemes cannot discriminate between molecules which overlay a pharmacophore and are shaped appropriately for receptor binding with those that have good electrostatics but are sterically inappropriate, a geometric or shape-based score is an important supplement to them. Since the geometric scoring schemes are not effective at finding

known compounds, both types of scores are necessary to a positive docking application. Attempts to combine the scores show that it is highly unlikely that the two terms can contribute equally to a final score unless some sort of weighting is performed; in addition, it may be desirable in some cases to weight the scores unequally if they are believed to have unequal influence on binding. If scores are added it is therefore necessary to choose a scaling factor after the relative magnitudes of the score components have been separately determined.

References

Available Chemicals Directory. San Leandro, CA, MDL Information Systems, Inc.

MDL Drug Data Report. San Leandro, CA, MDL Information Systems, Inc.

SYBYL Molecular Modeling Software. St. Louis, MO, Tripos, Inc.

Abola, E. E., F. C. Bernstein, et al. (1987). Protein Data Bank.

Crystallographic Databases — Information Content, Software Systems,

Scientific Applications. F. H. Allen, G. Bergerhoff and R. Sievers.

Bonn/Cambridge/Chester, Data Commission of the International Union of Crystallography: 107-132.

Allen, F. H., O. Kennard, et al. (1973). Journal of Chemical Documentation 13: 119.

Bemis, G. W. and I. D. Kuntz (1992). "A fast and efficient method for 2D and 3D molecular shape description." Journal of Computer-Aided Molecular Design 6: 607-628.

Bernstein, F. C., T. F. Koetzle, et al. (1977). "The Protein Data Bank: A Computer-Based Archival File for Macromolecular Structures." Journal of Molecular Biology 112: 535-542.

Cannon, J. G. (1983). "Structure-Activity Relationships of Dopamine Agonists." Annual Reviews of Pharmacology and Toxicology 23: 103-130.

Christie, B. D., D. R. Henry, et al. (1990). MACCS-3D: A Tool for Three-dimensional Drug Design. Online Information 90. D. I. Raitt. Oxford, Learned Information: 137-161.

DesJarlais, R. L., R. P. Sheridan, et al. (1988). "Using Shape Complementarity as an Initial Screen in Designing Ligands for a Receptor

Binding Site of Known Three-Dimensional Structure." Journal of Medicinal Chemistry **31**(4): 722-729.

Gasteiger, J. and M. Marsili (1980). "Iterative Partial Equalization of Orbital Electronegativity — A Rapid Access to Atomic Charges." Tetrahedron **36**: 3219-3288.

Good, A. C., E. E. Hodgkin, et al. (1991). "Utilization of Gaussian Functions for the Rapid Evaluation of Molecular Similarity." Journal of Chemical Information and Computer Sciences **32**: 188-191.

Gund, P., W. T. Wipke, et al. (1974). "Computer Searching of a Molecular Structure File for Pharmacophoric Patterns." Computing in Chemical Research, Education, and Technology **3**: 5.

Ho, C. M. W. and G. R. Marshall (1993). "FOUNDATION: A Program to Retrieve All Possible Structures Containing a User-defined Minimum Number of Matching Query Elements from Three-dimensional Databases." Journal of Computer-Aided Molecular Design **7**: 3-22.

Jakes, S. E. and P. Willett (1986). "Pharmacophoric Pattern Matching in Files of 3-D Chemical Structures: Selection of Interatomic Distance Screens." Journal of Molecular Graphics **4**: 12-20.

Jarvis, R. A. and E. A. Patrick (1973). IEEE Transactions on Computing **C-22**: 1025-1034.

Karfunkel, H. H. and V. Eyraud (1989). "An Algorithm for the Representation and Computation of Supermolecular Surfaces and Volumes." Journal of Computational Chemistry **10**(5): 628-634.

Katerinopoulos, H. E. and D. I. Schuster (1987). "Structure-Activity Relationships for Dopamine Analogs: A Review." Drugs of the Future **12**(3): 223-253.

Kuntz, I. D., J. M. Blaney, et al. (1982). "A Geometric Approach to Macromolecule-Ligand Interactions." Journal of Molecular Biology **161**: 269-288.

Lauri, G. and P. A. Bartlett (1994). "CAVEAT: A Program to Facilitate the Design of Organic Molecules." Journal of Computer-Aided Molecular Design **8**: 51-66.

Manallack, D. T. and P. M. Beart (1988). "A Three Dimensional Receptor Model of the Dopamine D2 Receptor from Computer Graphic Analyses of D2 Agonists." Journal of Pharmacy and Pharmacology **40**: 422-428.

Marquart, M., J. Walter, et al. (1983). "The Geometry of the Reactive Site and of the Peptide Groups in Trypsin, Trypsinogen and its Complexes with Inhibitors." Acta Crystallographica, Section B **39**: 480.

Martin, Y. C., M. G. Bures, et al. (1993). "A fast new approach to pharmacophore mapping and its application to dopaminergic and benzodiazepine agonists." Journal of Computer-Aided Molecular Design **7**: 83-102.

Masek, B. B., A. Merchant, et al. (1993). "Molecular Skins: A New Concept for Quantitative Shape Matching of a Protein With Its Small Molecule Mimics." PROTEINS: Structure, Function, and Genetics **17**: 193-202.

Meng, E. C., B. K. Shoichet, et al. (1992). "Automated Docking with Grid-Based Energy Evaluation." Journal of Computational Chemistry **13**(4): 505-524.

Pattabiraman, N., M. Levitt, et al. (1985). "Computer Graphics in Real-time Docking with Energy Calculation and Minimization." Journal of Computational Chemistry **6**(5): 432-436.

Pearlman, R. S. (1987). "Rapid Generation of High-Quality Approximate 3-D Molecular Structures." Chemical Design Automation News 2(1): 5-6.

Perry, N. C. and V. J. van Geerestein (1992). "Database Searching on the Basis of Three-Dimensional Molecular Similarity Using the SPERM Program." Journal of Chemical Information and Computer Sciences 32: 607-616.

Powers, J. C. and J. W. Harper (1986). Inhibitors of Serine Proteinases. Inhibitors of Serine Proteinases. A. J. Barrett and G. Salvesen. New York, Elsevier: 55-152.

Seeman, P., M. Watanabe, et al. (1985). "Dopamine D₂ Receptor Binding Sites for Agonists." Molecular Pharmacology 28: 391-399.

Sheridan, R. P., R. Nilakantan, et al. (1989). "3DSEARCH: A System for Three-Dimensional Substructure Searching." Journal of Chemical Information and Computer Sciences 29: 255-260.

Sheridan, R. P., A. Rusinko, III, et al. (1989). "Searching for Pharmacophores in Large Coordinate Databases and Its Use in Drug Design." Proceedings of the National Academy of Sciences of the U.S.A. 86: 8165-8169.

Shoichet, B. K. and I. D. Kuntz (1993). "Matching Chemistry and Shape in Molecular Docking." Protein Engineering 6(7): 723-732.

Van Drie, J. H., D. Weininger, et al. (1989). "ALADDIN: An Integrated Tool for Computer-assisted Molecular Design and Pharmacophore Recognition from Geometric, Steric, and Substructure Searching of Three-dimensional Molecular Structures." Journal of Computer-Aided Molecular Design 3: 225-251.

van Geerestein, V. J., N. C. Perry, et al. (1990). "3D Database Searching on the Basis of Ligand Shape Using the SPERM Prototype Method."

Tetrahedron Computer Methodology 3(6C): 595-613.

Conclusions

DOCK is a useful tool for generating orientations of molecules relative to representations of receptor sites. Coupled with appropriate scoring schemes, DOCK can search databases in order to locate new drug leads. The different scoring methods which have been developed for use within DOCK allow it to be adapted for use in a variety of different drug design problems. The methods developed in this work facilitate docking to charged receptor sites using electrostatic scoring with correction for solvation and allow docking to positive images of receptor sites by scoring the similarity of small molecules to the target image.

The electrostatic scoring scheme used in this work retrieved database molecules with complementary to the charged trypsin binding pocket, but most of these molecules were far too highly charged; the use of a correction for the cost of desolvating the charges fixed this problem. Such a correction should probably be applied to any score based solely on charge interaction, since such scores will always favor larger charges. Since the cost of desolvating relatively uncharged molecules is small, the inclusion of a solvation term in the score should not affect the results in uncharged molecules, as was evident in the experiments with docking to chymotrypsin.

In positive docking, the electrostatic scoring scheme proved useful in retrieving molecules related to active compounds from databases in two charged systems. However, the electrostatic scoring scheme by itself cannot be expected to identify molecules of an appropriate shape and size for binding to the receptor associated with the positive image. The geometric scoring scheme, which might be useful in this regard, did not discriminate between inhibitor-like molecules and the remainder of the databases. Therefore, a

minimum of two scoring schemes, one based on electrostatics and one related to shape, should be used together to identify molecules similar to a positive image. Since these scores cannot readily be made to fall into the same numeric range, and since there may be *a priori* reasons for giving them different weights, any method for docking to positive images needs a scoring scheme which can be readily adjusted for different problems.

Basing the geometric scoring method on atom positions allows for simple and rapid calculation. However, a method based on the overall shape of the target and the candidate molecules might better capture the fit between ligand and receptor. If the surface-based scoring method proves practical for database searching, it may be a useful adjunct to or replacement for the atom-based method. A method for scoring likely hydrophobic interactions could add a level of detail by accounting for the types of surfaces involved in binding. Applying the methods to a drug design problem which allows evaluation of molecules would be a practical test of positive docking.

Appendix 1: Correction of an Error in DOCK Distance Handling

Before it matches distances between atoms to distances between spheres, DOCK sorts these distances into bins, or groups of similar distances. This allows the matching algorithm to save time by only attempting to pair those distances which are close enough to each other that they fall into the same bin. In the course of investigating whether DOCK could be a useful tool for aligning molecules to each other once their functional groups had been labeled, I discovered that the subroutine makbin, which assigns distances to bins, did not work properly for sets of only a few distances. The code for a corrected version of makbin is included here.

I discovered the problem while investigating the potential of DOCK for generating pharmacophore alignments. I attempted to DOCK each of 19 test molecules to a reference molecule, which served as a template. I reduced the template molecule to a set of nine spheres labeled as the locations of functional groups known to occur in the dopamine D2 pharmacophore. The test molecules were reduced to sets of six to thirteen correspondingly labeled atoms. All of the molecules had a particular set of four distances in common, but DOCK located orientations which matched these distances for only 10 of the 19 molecules. This led me to investigate the bin assignment and matching routines in DOCK.

Code introduced into DOCK in version 2.0 (Shoichet, Bodian et al. 1992) manages distances by choosing a seed sphere or atom and dividing the remaining spheres or atoms into bins based on their distance from the seed. DOCK generates matches based on this division of the atoms and spheres, then repeats the process until all atoms and spheres have been used as seeds. The user specifies the width of the bins — that is, the size of the distance

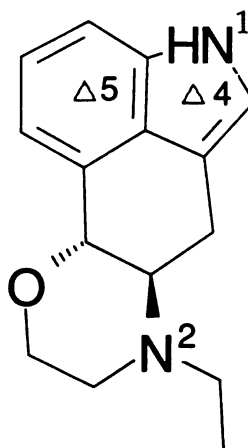


Figure 1: Molecule III.

Points used for labeled docking included atoms 1, 2, 4, and 5 as indicated; point 3 is the position of a lone pair on atom 2 and point 6 is the position of a lone pair on atom 5.

range which each bin contains — and an additional interval by which each bin overlaps its neighbors. A distance x is included in the bin marked i if

$$\text{int}(x/\text{binwidth}) = i$$

where $\text{int}(a)$ is the integer portion of a , i.e. a is truncated. The overlap range of each bin includes all distances x such that

$$i * \text{binwidth} \geq x \geq i * \text{binwidth} - \text{overlap}.$$

A comparison of the distances among labeled atoms in molecule III (Figure 1) with the contents of one set of bins generated by DOCK (Table 1) showed that many bins did not contain all the spheres they should; some bins were missing from the list entirely. Examination of the code showed that, for small distance sets, some centers were not being tested to see whether they fell into the overlap region of some bins. The makbin routine worked as follows:

1. Sort the centers in decreasing order by their distance from the seed center.

2. Place the most distant center in the bin which corresponds to its distance from the seed.
3. Add centers to this bin until one is found which does not fall within the range of the bin label ($i * binwidth > x$).
4. Beginning with this center, check centers to see whether they fall within the overlap range of the previous bin ($x \geq i * binwidth - overlap$). Continue until a center is found which does not meet this condition.
5. Go to step 2, starting with the first center not within the previous bin.

If no centers fell within the range corresponding to a given bin, no centers **were** checked to see whether they fell into its overlap range, resulting in **missing** centers.

The new version of makbin starts with the first center with a distance **less** than the upper boundary for the current bin and continues checking until **a center** is found with a distance less than the lower boundary of the overlap **range** for the current bin. When this version was used, the bin contents for **molecule III** were those expected given the interatomic distances. When the **examples** used by Meng (Meng, Shoichet et al. 1992) were docked with the old **and** new versions of makbin, the bin contents differed only in the presence of **a few** additional centers with the new version. Results using a bin overlap of **zero** were identical.

Table 1: Interatomic Distances and Bin Contents in Molecule III

(a) Distances in Ångstroms between the six labeled atoms of III:

Atom	1	2	3	4	5	6
1	0	6.019	7.109	1.151	2.725	3.000
2	6.019	0	2.900	4.893	5.142	8.986
3	7.109	2.900	0	6.063	6.440	9.947
4	1.151	4.893	6.063	0	2.156	4.151
5	2.725	5.142	6.440	2.156	0	5.209
6	3.000	8.986	9.947	4.151	5.209	0

(b) Bin contents found using each of the atoms in III as the seed for distance bin generation, with missing centers noted. Bin width was 0.9 Å; bin overlap was 0.4 Å.

Center	Bin Distance Label	Centers in Bin	Centers Missing
1	1	4	
	3	6,5	
	6	2	
	7	2,3	
	8		3
2	3	3	
	5	4,5	
	6	5,1	
	7		1
	9	6	
	10		6
3	3	2	
	6	4	
	7	4,5,1	
	8		1
	11	6	
4	1	1	
	2	5	
	4	6	
	5	2,6	
	6	3	
	7		3
	5	2	4
3		1	
5		2,6	
6			2,6
7		3	

6	3	1	
	4	4	
	5	4,5	
	6		5
	9	2	
	10		2
	11	3	

FORTRAN Code for the Revised Version of makbin

```

*****
*-----makbin-----*
c   this subroutine creates and fills the distance bin array ctrnum
c   using the lists of centers in ctrtmp and their corresponding
c   distances in dist
c   Original code: BKS, 1992.
c   Rewritten to fix bug in bin overlap by C.Corwin, August 1993
*****
      subroutine makbin(dist,maxnum,ctrtmp,nodlim,ctrnum,
& dum1,dum2,dum3,binSZ,ovlap,totbn,center)

      include 'max.h'

      real dist(maxpts,maxpts)
c     dist -- row i is list of distances from center i to centers
listed
c           in ctrtmp(i)
      integer maxnum
c     maxnum -- total number of centers for which bins will be made
      integer ctrtmp(maxpts,maxpts)
c     ctrtmp -- row i is list of centers sorted in decreasing order
c             of distance from center i
      integer ctrnum(maxpts,0:maxnod,maxwid)
c     ctrnum -- final bin array; ctrnum(i) contains bins for center i
c             ctrnum(i,1) -- bin distance label
c             ctrnum(i,2) -- number of centers in bin
c             ctrnum(i,3,..ctrnum(i,2)+2) -- centers
      real binSZ,ovlap
c     binSZ -- size of the distance range which determines bin
placement
c     ovlap -- distance range by which bin i overlaps bin i-1;
c             add to binSZ to get the net size of the bin
      integer totbn(maxpts)
c     totbn(i) -- number of bins for center i
      integer dum1,dum2,dum3,center,nodlim
c     dum1,dum2,dum3,center,nodlim -- not used;
c             retained to preserve compatibility with
c             older calling routines

      integer i,j,k,n
      integer dispos,binpos
c     dispos,binpos: positions where centers and bins are inserted
      integer dislbl
c     dislbl -- label corresponding to current distance
      integer totnod
c     totnod -- maximum number of bins for current center
      integer bintmp(0:maxnod,maxwid)
c     bintmp -- temporary array of potential bins for a single center
c             bintmp(i) is the bin with distance label i
      real binbnd(0:maxnod,2)
c     binbnd(i,1) -- lower distance bound of bin i
c     binbnd(i,2) -- upper distance bound of bin i
c             add to binSZ to get the net size of the bin

```

```

c      * initialize ctrnum, the final bin array *
do 25 k = 1,maxpts
    do 20 i=1,maxnod
        do 10 j=1,2
            ctrnum(k,i,j)=0
10            continue
20        continue
25    continue
c      * initialize bin distance labels in bintmp *
do 50 i=0,maxnod
    bintmp(i,1)=i
50    continue
c      * initialize binbnd to bin boundaries *
c      * the bin with distance label i contains distances x such that *
c      * binsz*(i+1) >= x >= binsz*(i)-ovlap *
do 100 i=0,maxnod
    binbnd(i,1)=binsz*i-ovlap
    binbnd(i,2)=binsz*(i+1)
100    continue

c      * for each center i *
do 500 i=1,maxnum
c      * initialize temporary bins *
do 150 j=0,maxnod
    do 140 k=2,maxwid
        bintmp(j,k)=0
140        continue
150    continue
    totnod=nint(dist(i,1)/binsz)
    if (totnod.gt.maxnod) totnod=maxnod

c      * for each center j!=i *
do 300 j=1,maxnum-1
c      * Starting with bin corresponding to distance label, *
c      * place center in bins with increasing labels until *
c      * distance no longer exceeds the lower bin boundary *
dislbl=int(dist(i,j)/binsz)
200    continue
    if (dislbl.gt.totnod) goto 250
    if (dist(i,j).lt.binbnd(dislbl,1)) goto 250
    bintmp(dislbl,2)=bintmp(dislbl,2)+1
    dispos=bintmp(dislbl,2)+2
    if (dispos.gt.maxwid) then
        write(6,*) 'Array bound exceeded!'
        write(6,*) 'Recompile with larger maxwid parameter'
        stop
    endif
    bintmp(dislbl,dispos)=ctrtmp(i,j)
    dislbl=dislbl+1
    goto 200
250    continue
300    continue

c      * for each bin k *
binpos=1
do 400 k=totnod,0,-1
c      * if not empty, copy the bin into ctrnum(i,binpos,...) *

```

```
        if (bintmp(k,2).ne.0) then
            do 350 n=1,bintmp(k,2)+2
                ctrnum(i,binpos,n)=bintmp(k,n)
350         continue
            binpos=binpos+1
        endif
400     continue
        totbn(i)=binpos-1
500 continue

return
end
```

References

Meng, E. C., B. K. Shoichet, et al. (1992). "Automated Docking with Grid-Based Energy Evaluation." Journal of Computational Chemistry **13**(4): 505-524.

Shoichet, B. K., D. L. Bodian, et al. (1992). "Molecular Docking Using Shape Descriptors." Journal of Computational Chemistry **13**(3): 380-397.

Appendix 2: Code for Creation of a Grid to Mark Acceptable Ligand Positions in Positive Docking

In docking small molecules to a positive image it is desirable to restrict the atoms of the candidate molecules to a region of space lying within or near the target image. DOCK 3.0 (Meng, Shoichet et al. 1992) determines whether atoms fall in acceptable regions by using a grid which indicates acceptable and unacceptable locations. CHEMGRID, the program used to generate this grid, marks positions inside a macromolecule as acceptable ("F") and positions outside the molecule as unacceptable ("T"). For positive docking, I modified this program so that it marked locations within or close to the positive image as acceptable and those outside as unacceptable.

To change how locations were marked, the routines *ddist*, *dconst*, and *gauss3* were modified so that grid points are initialized to "T" (atoms not allowed) and are set to "F" (atoms allowed) only if they lie within the user-specified cutoff values for an atom in the positive image. It is necessary to use much larger cutoff values than would be appropriate for docking to a negative image; I typically set both *pcon*, the maximum distance from a polar atom, and *ccon*, the maximum distance from a nonpolar atom, to 2.4 Å. This scheme alone leaves small forbidden regions within the positive image, so the new subroutines examine the nearest neighbors of all cells marked "T" and sets each cell with five or six "F" neighbors to "F". Using these conditions produces an allowed region which extends beyond the positive image slightly.

FORTRAN Code for the *dconst*, *ddist*, and *gauss3* Subroutines from the Revised Version of CHEMGRID

```
c*****
c
c      Copyright (C) 1991 Regents of the University of California
```

```

C                               All Rights Reserved.
C
C      subroutine dconst(unitno, grdcut, grddiv, grdpts, esfact, offset,
C      &                               invgrd)
C
C      --called from CHEMGRID
C      --increments vdw and electrostatics values at grid points, using
C      a constant dielectric function                               ECMeng 4/91
C Modified to 'invert' the bump grid so that bumps are outside the
molecule
C used for grid generation by C. Corwin
C*****
C      include 'chemgrid.h'
C
C      real mincon, minsq
C      parameter (mincon=0.0001)
C      integer unitno, i, j, k, n
C      real r2, r6
C
C      minsq=mincon*mincon
C
C      --open parameterized receptor file (from subroutine parmrec)
C
C      open (unit=unitno, file='PDBPARM', status='old')
C
C      100 read (unitno, 1006, end=500) natm, vdown, rsra, rsrb, rcrd,
C      &(rcrd(i), i=1,3)
C      1006 format (2I5, 2(1x, F8.2), 1x, F8.3, 1x, 3F8.3)
C      if (vdown .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
C      do 110 i=1,3
C          rcrd(i)=rcrd(i) - offset(i)
C          nearpt(i)=nint(rcrd(i)/grddiv) + 1
C          if (nearpt(i) .gt. (grdpts(i) + grdcut)) go to 100
C          if (nearpt(i) .lt. (1 - grdcut)) go to 100
C      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
C      do 400 i=max(1,(nearpt(1)-grdcut)),
C      &min(grdpts(1),(nearpt(1)+grdcut))
C          gcrd(1)=float(i-1)*grddiv
C          do 300 j=max(1,(nearpt(2)-grdcut)),
C      & min(grdpts(2),(nearpt(2)+grdcut))
C              gcrd(2)=float(j-1)*grddiv
C              do 200 k=max(1,(nearpt(3)-grdcut)),
C      & min(grdpts(3),(nearpt(3)+grdcut))
C                  gcrd(3)=float(k-1)*grddiv
C                  n = indx1(i,j,k,grdpts)
C                  r2 = dist2(rcrd,gcrd)

```



```

        if (r2 .gt. cutsq) go to 120
c --set points within cutoff of atoms to F for inverted bump grid;
c   T or X for normal bump grid (CBC)
        if (invgrd) then
            if (r2 .lt. minsq) then
                bump(n)='F'
                r2 = minsq
            else if(((r2 .lt. cconsq .and. vdown .le. 5) .or. (r2 .lt.
& pconsq .and. vdown .ge. 8)) .and. bump(n) .eq. 'T') then
                bump(n)='F'
            endif
        else
            if (r2 .lt. minsq) then
                bump(n)='X'
                r2 = minsq
            else if(((r2 .lt. cconsq .and. vdown .le. 5) .or. (r2 .lt.
& pconsq .and. vdown .ge. 8)) .and. bump(n) .eq. 'T') then
                bump(n)='T'
            endif
        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

```

C-----

```

c
c   Copyright (C) 1991 Regents of the University of California
c   All Rights Reserved.
c

```

```

        subroutine ddist(unitno, grdcut, grddiv, grdpts, esfact, offset,
&                          invgrd)

```

```

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 Modified to 'invert' the bump grid so that bumps are outside the
molecule

```

```

c used for grid generation by C. Corwin

```

```

c
c   include 'chemgrid.h'
c
c   real mincon, minsq
c   parameter (mincon=0.0001)
c   integer unitno, i, j, k, n
c   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
c  100 read (unitno, 1006, end=500) natm, vdown, rsra, rsrb, rcrd,
        &(rcrd(i), i=1,3)
c  1006 format (2I5, 2(1x, F8.2), 1x, F8.3, 1x, 3F8.3)
        if (vdown .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
c  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
c  --set points within cutoff of atoms to F for inverted bump grid;
c  T or X for normal bump grid (CBC)
                    if (invgrd) then
                        if (r2 .lt. minsq) then
                            bump(n)='F'
                            r2 = minsq
                        else if(((r2 .lt. cconsq .and. vdown .le. 5) .or. (r2 .lt.
& pconsq .and. vdown .ge. 8)) .and. bump(n) .eq. 'T') then
                            bump(n)='F'
                        endif
                    else
                        if (r2 .lt. minsq) then
                            bump(n)='X'
                            r2 = minsq
                        else if(((r2 .lt. cconsq .and. vdown .le. 5) .or. (r2 .lt.
& pconsq .and. vdown .ge. 8)) .and. bump(n) .eq. 'T') then
                            bump(n)='T'
                        endif
                    endif
                endif
            endif
        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
C-----
*****
C
C      Copyright (C) 1991 Regents of the University of California
C              All Rights Reserved.
C
C      subroutine gauss3(unitno, grdcut, grddiv, grdpts, esfact, offset,
C      &                  invgrd)
C
C      --called from CHEMGRID
C      --increments vdw and electrostatics values at grid points, using
C      the three-Gaussian approximation to 1/r described in
C      J. Chem. Inf. Comp. Sci. 32, pp. 188-190 (1992)
C      --"inverts" the bump grid so that all grid points are true except
C      those within pcon or ccon of an atom, which are made false
C      C. Corwin, February 1994, based on code by EC Meng
*****
C
C      include 'chemgrid.h'
C
C      real mincon, minsq
C      parameter (mincon=0.0001)
C      integer unitno, i, j, k, n
C      real r2, r6
C      real g3rinv
C g3rinv -- 3-gaussian approximation to 1/r
C
C      minsq=mincon*mincon
C
C      --open parameterized receptor file (from subroutine parmrec)
C
C      open (unit=unitno, file='PDBPARM', status='old')
C
C      100 read (unitno, 1006, end=500) natm, vdwn, rsra, rsrb, rcrg,
C          &(rcrd(i), i=1,3)
C      1006 format (2I5, 2(1x, F8.2), 1x, F8.3, 1x, 3F8.3)
C          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
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
c --set points within cutoff of atoms to F for inverted bump grid;
c T or X for normal bump grid (CBC)
                    if (invgrd) then
                        if (r2 .lt. minsq) then
                            bump(n)='F'
                            r2 = minsq
                        else if(((r2 .lt. cconsq .and. vdown .le. 5) .or. (r2 .lt.
& pconsq .and. vdown .ge. 8)) .and. bump(n) .eq. 'T') then
                            bump(n)='F'
                        endif
                    else
                        if (r2 .lt. minsq) then
                            bump(n)='X'
                            r2 = minsq
                        else if(((r2 .lt. cconsq .and. vdown .le. 5) .or. (r2 .lt.
& pconsq .and. vdown .ge. 8)) .and. bump(n) .eq. 'T') then
                            bump(n)='T'
                        endif
                    endif
                    r6 = r2*r2*r2
                    aval(n)=aval(n) + rsra/(r6*r6)
                    bval(n)=bval(n) + rsrb/r6
                    g3rinv=(0.3001*exp(-0.0499*r2))
& + (0.9716*exp(-0.5026*r2))
& + (0.1268*exp(-0.0026*r2))
                    esval(n)=esval(n) + (332.0*rcrg*g3rinv)/esfact
120 continue
200 continue
300 continue
400 continue
go to 100
500 continue
close (unitno)
return
end
c-----

```

References

Meng, E. C., B. K. Shoichet, et al. (1992). "Automated Docking with Grid-Based Energy Evaluation." Journal of Computational Chemistry **13**(4): 505-524.

Appendix 3: Source Code for Surface-Based Scoring – Surfgrid and Surfscore

The program surfgrid creates a representation of a molecular surface on a grid. Surfgrid includes four routines, dist2, indx1, grdout, and parmrec, which were originally part of CHEMGRID, written by Elaine Meng, and are used with minor modifications. The main surfgrid program and the routine chekat include some code which was adapted from CHEMGRID. The remaining routines, cheksf, layer2, newgen, subdiv, stilsf, finsf, and growsf, are unique to surfgrid.

Surfscore calculates the surface-based score for an orientation of a ligand. Mkshel precomputes, in grid units, the offsets between an atom of a given type and the grid cells whose distances from it lie between its van der Waals radius and its van der Waals radius plus a surface thickness; this makes surface scoring faster. Both of these routines are called from within DOCK.

Makefile for surfgrid

```
FFLAGS= -u -g -check_bounds -trapeuv

SRC= surfgrid.f \
     dist.f \
     grdout.f \
     chekat.f \
     cheksf.f \
     layer2.f \
     newgen.f \
     subdiv.f \
     stilsf.f \
     finsf.f \
     growsf.f \
     parmrec.f

all : $(SRC:f=o)
      f77 -o surfgrid $(SRC:f=o)

# Additional dependencies
surfgrid.o: chemgrid.h
chekat.o: chemgrid.h
grdout.o : chemgrid.h
```

```

cheksf.o : chemgrid.h
layer2.o : chemgrid.h
newgen.o : chemgrid.h
subdiv.o : chemgrid.h
stilsf.o : chemgrid.h
finsf.o : chemgrid.h
growsf.o : chemgrid.h
parmrec.o : chemgrid.h parmrec.h

```

Header files used with surfgrid

```

c-----
c      header for SURFGRID
c      adapted from Elaine Meng's CHEMGRID header
c      C. Corwin, May 1994
c-----
      integer maxpts
      parameter (maxpts=10000)
c maxpts--maximum number of points in intial grid
      integer maxfin
      parameter (maxfin = 500000)
c maxfin -- maximum number of points in final grid
      integer maxatm
      parameter (maxatm = 10000)
c maxatm -- maximum number of protein atoms
      integer maxasc
      parameter (maxasc = 20)
c maxasc -- maximum number of atoms associated with a point in starting
grid
      integer npts
c npts--number of grid points
c      real aval(maxpts), bval(maxpts), esval(maxpts)
c      character*1 bump(maxpts)
c aval(), bval(), esval(), bump()--values stored "at" grid points
      real rsra, rsrb, rcrd, rcrd(3)
c rsra, rsrb, rcrd, rcrd()--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))
      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 * variables added - CBC *

```

```

c * atom and van der Waals radius info *
  integer natm
c natm -- number of atoms
  character*4 atname(maxatm)
c atname -- atom names
  real vdwrads(maxatm)
c vdwrads -- vdW radii of target atoms
  real atcrd(maxatm,3)
c atcrd -- atom coordinates

c * initial (coarse) grid *
  character*1 celtyp(maxpts)
c celtyp -- status of each grid cell, e.g. '*' for filled, ' ' for
empty
  integer numasc(maxpts)
c numasc -- number of atoms associated with each cell
  integer*2 celatm(maxpts,maxasc)
c celatm -- list of atoms associated with each grid cell
  integer ascmax,srfmax
  real ascavg,srfavg
c ascmax,srfmax,ascavg,srfavg -- maximum and average numbers of atoms
c associated with a cell and with a surface cell
  integer nsrfpt
c nsrfpt -- number of surface points

c * final grid *
  character*1 fincel(maxfin)
c fincel -- status of each cell in final grid
  integer finpts
c finpts -- number of points in final grid
  real finoff(3)
c finoff -- offset for final grid
  integer fingrd(3)
c fingrd()--number of grid points along box dimensions (x,y,z)
c NOTE: fingrd(i)=findimm(i) + 1 (lowest indices are (1,1,1))
  real findiv
c findiv -- desired final grid division
  integer divrat
c divrat -- ratio of initial grid division to final grid division
  integer ndesc
c ndesc -- number of descendants in final grid for each cell of initial
grid
c
  common
  &/model/ natm, atname, vdwrads, atcrd
  &/initgr/ celtyp, numasc, celatm,
  & ascmax, ascavg, srfmax, srfavg, nsrfpt
  &/fgrid/ findiv, fincel
c-----
c-----
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, sch
    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
    logical found
    character*80 line
    real crgtot
c
    common
    &/link/ inum, ilink
    &/name/ atm, res, resnum, chain
    &/value/ crg, vdwtyp, sra, srb
c-----

```

Source code for surfgrid and subroutines

```

c-----
c      program SURFGRID
c
c      Create a grid representation of a SKINNY-type thick surface
c      given a pdb structure and a box within which the grid may lie
c
c      Much of this code was adapted from the program CHEMGRID,
c      which was written by Elaine Meng
c
c      C. Corwin, May 1994
c-----
    include 'chemgrid.h'
c
    character*80 vdwfil
    character*80 recfil, boxfil, grdfil
c recfil--pdb-format receptor file (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
    real com(3)
c com -- center of mass of input box
    integer sfthik
c sfthik -- thickness of final surface in number of grid cells
    integer i, j, n
c
    open (unit=1, file='INSURF', status='old')
    open (unit=2, file='OUTSURF', status='new')

```

```

c
c   * get receptor and parameter file info *
    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
c   * read and hash receptor parameters *
    call parmrec(recfil, table, vdwfil, 2)
c
c   * get grid box location and dimensions *
    read (1, 1000) boxfil
    write (2, *) 'input box file defining grid location:'
    write (2, 1000) boxfil
c
    open (unit=3, file=boxfil, status='old')
    read (3, 1000) dumlin
    read (3, 1001) (com(i), i=1,3)
1001 format (25x, 3f8.3)
    read (3, 1002) (boxdim(i), i=1,3)
1002 format (29x, 3f8.3)
    close (3)
c
    write (2, *) 'box center coordinates [x y z]:'
    write (2, *) (com(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, *) grddiv
    write (2, *) 'initial 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

c    * get output grid name *
    read(1,1000) grdfil
    write (2,*) 'output grid prefix name:'
    write (2,1000) grdfil

c    * Read and process additional input parameters *
c    * including final grid spacing *
    read (1, *) findiv
    write (2, *) 'final grid spacing in angstroms'
    write (2, *) findiv
    read(1,*) sfthik
    write (2, *) 'Layers of grid cells in surface ',sfthik

    close (1)

c
c --initialize coarse grid
c
    do 90 n=1, maxpts
c    * make all grid elements unfilled *
        celtyp(n) = ' '
        numasc(n) = 0
        do 85 i=1,maxasc
            celatm(n,i) = 0
85    continue
90    continue

c    * for each atom, mark grid elements within vdw radius *
c    * filled and add atom to associated list *
    call chekat(grddiv,grdpts,offset)
    call cheksf(npts,grdpts)
    call layer2(celtyp,npts,grdpts)

c    * write initial grid statistics *
    write (2,*) 'Maximum number of associated atoms:', ascmax
    write (2,*) 'Average for cells with associated atoms:', ascavg
    write (2,*) 'Maximum number of atoms associated with a surface ',
&    'cell:', srfmax
    write (2,*) 'Average for surface cells with associated atoms:',
&    srfavg
    write (2,*) 'Number of initial surface cells:',nsrfpt
    call grdout(celtyp,'init ', 3, npts, grddiv, grdpts, offset,
&    1)

c    * calculate final grid parameters *
    divrat = nint(grddiv/findiv)
    ndesc = divrat**3
    do 100 i=1,3
        fingrd(i) = grdpts(i)*divrat
        finoff(i) = offset(i) - (grddiv-findiv)/2
100    continue

```

```

      finpts = npts*ndesc
c   * initialize final grid *
      do 110 n=1,maxfin
          fincel(n) = ' '
110 continue

c   * Mark descendants of filled initial grid cells filled *
c   * Check descendants of filled surface cells: *
c   * are atoms still associated with them? *
      call newgen(npts,divrat,ndesc,grdpts,fingrd)
      call stilsf(npts,divrat,ndesc,grdpts,fingrd,finoff)
c   * Mark surface cells in current generation *
      call finsf(fincel,finpts,fingrd)
      write (2,*) 'Number of final surface cells:',nsrfpt

c   * Expand grid to desired thickness *
      call growsf(finpts,fingrd,sfthik)
      write (2,*) 'Number of surface cells at thickness = ',sfthik,':'
      write (2,*) nsrfpt

c   * write grid to output file *
      call grdout(fincel,grdfil, 3, finpts, findiv, fingrd, finoff,
&                sfthik)
c
      close (2)
      end
c-----
c-----
c
c   Copyright (C) 1991 Regents of the University of California
c   All Rights Reserved.
c
c   subroutine parmrec(recfil, table, vdwwil, unitno)
c
c   --called from surfgrid
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   4/93 ECM altered to report residues with nonzero charge to OUTCHEM
c
c   recfil--name of receptor pdb file
c   table--name of the table to be referenced for receptor atom
c   parameters
c   vdwwil--name of file containing van der Waals parameters
c   unitno--logical unit number to write parameterization information
c   and warnings to
c-----
c   include 'chemgrid.h'
c   include 'parmrec.h'
c
c   character*80 recfil, table, vdwwil

```

```

integer unitno
integer i, n, j
c
character*3 presid
integer resn, prevn
real rescrg

real typrad(maxtyv)
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) typrad(nvtyp)
1002 format (10x, f8.4)
  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
C     natm=0
C     crgtot=0.0
C     rescrgr=0.0
C
C     open (unit=11, file=recfil, status='old')
C     open (unit=12, file='PDBPARM', status='new')
C     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
C
   if (resid.ne.' ') then
     presid=resid
   else
     presid=sresid
   endif
C
   atom=line(13:16)
   resid=line(18:20)
   chn=line(22:22)
   resno=line(23:26)
C
   read (resno, *) resn
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)') 'vdW radius set to ',
                        & '1.375 and charge set to 0.0'
                    * Assign vdW radius of H on carbon if *
                    * no parameters are found *
                    write (12, 2000) natm, 0, 1.375, 0.0,
                        & line(31:54)
                    atname(natm) = atom
                    vdwrad(natm) = 1.375
                    read (line, 1003) (atcrd(natm,j), j=1,3)
                    1003 format (30x,3f8.3)
                    go to 20
                endif
            endif
        endif
    endif
endif
endif
endif
endif
endif
endif
write (12, 2000) natm, vdwtyp(n), typrad(vdwtyp(n)),
&crg(n), line(31:54)
2000 format (2I5, 1x, F8.2, 1x, F8.3, 1x, A24)
c
c      * Save parameters in arrays; code added by CBC *
atname(natm) = atom
vdwrad(natm) = typrad(vdwtyp(n))
read(line,1003) (atcrd(natm,j), j=1,3)
c
    if (natm.eq.1) prevn=resn
    if (resn.ne.prevn) then
        if (abs(rescrg).gt.0.0001) then
            write (13, 2001) ' CHARGED RESIDUE ', presid, prevn, rescrg
            2001 format (A17, A3, I5, F8.3)
        endif
        rescrg = crg(n)
        prevn = resn
    else
        rescrg = rescrg + crg(n)
    endif
c
    crgtot=crgtot + crg(n)
    go to 20
990 continue
    if (abs(rescrg).gt.0.0001) then
        if (resid.eq.' ') resid = sresid
        write (13, 2001) ' CHARGED RESIDUE ', resid, resn, rescrg

```

```

endif
close (11)
close (12)
write (13, *) ' '
write (13, '(A15, F8.3)') 'Total charge = ', crgtot
close (13)
return
end

c
c-----
c
c      subroutine enter(atom, resid, resno, chn, nent)
c
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
c      integer n, new, nent
c
c      integer ihash
c
c      --get hash number using function ihash
c
c      n=ihash(atom, resid, resno, chn)
c      if (inum(n) .ne. 0) then
c
c      --slot filled; keep going along linked numbers until zero found
c
c      100  continue
c          if (ilink(n) .eq. 0) go to 200
c          n=ilink(n)
c          go to 100
c      200  continue
c
c      --find an empty slot and fill it, leaving a trail in ilink()
c
c      do 300 new=1,maxtyp
c          if (inum(new) .eq. 0) go to 400
c      300  continue
c      400  continue
c          ilink(n)=new
c          n=new
c      endif
c      inum(n)=nent
c      ilink(n)=0
c      return
c      end

c
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
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
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
c      include 'parmrec.h'
c
c      integer n
c      integer ihash
c
c      n=ihash(atom, resid, resno, chn)
c      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

c
c -----
c -----
c
c      subroutine chekat(grddiv, grdpts, offset)
c
c called from surfgrid
c For each atom in the protein, mark grid elements within the van der
c Waals
c radius filled ('*') and increment the count of atoms associated with
c each grid element
c
c Some of this code was adapted from the CHEMGRID program written by
c Elaine Meng
c
c C. Corwin, May 1994
c -----

    include 'chemgrid.h'

    integer i,j,k,n
    integer pnt
    real cutoff, cutsq, grdcut
    real r2

    do 500 n=1,natm
c      * find location (in grid units) of grid point nearest atom *
c      * (add 1 because grid starts at (1,1,1) *
c      * ignore atoms outside grid *
        do 50 i=1,3
            rcrd(i) = atcrd(n,i) - offset(i)
            nearpt(i) = nint(rcrd(i)/grddiv) + 1
            if (nearpt(i) .gt. grdpts(i)) goto 499
            if (nearpt(i) .lt. 1) goto 499
50        continue

c      * cutoff is the van der Waals radius of the atom plus the *
c      * maximum distance from a grid point to a point within the *
c      * corresponding cube; convert cutoff to grid units *
        cutoff = vdwrad(n) + sqrt(3.0)*grddiv/2.0
        cutsq = cutoff**2
        grdcut = int(cutoff/grddiv + 1.0)

c      * loop through grid points within the cutoff cube of the *
c      * current receptor atom *
        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
    pnt = indx1(i,j,k,grdpts)
c    * if grid point is within cutoff distance of atom *
c    * mark filled and increment number of associated atoms *
    r2 = dist2(rcrd,gcrd)
    if (r2 .le. cutsq) then
      celtyp(pnt) = '*'
      numasc(pnt) = numasc(pnt)+1
      celatm(pnt,numasc(pnt)) = n
    endif
200    continue
300    continue
400    continue

499 continue
500 continue

return
end

c-----
c
c    subroutine layer2(cells,npts,grdpts)
c
c    Examine grid points; change all '*' neighbors of 'S' cells
c    to type '2' cells
c
c    called from surfgrid
c    C. Corwin, June 1994
c-----
c    include 'chemgrid.h'

c    character*1 cells(*)
c    cells -- grid cells to be updated
c    integer ptndx(3)
c    ptndx -- grid indices of current grid point
c    integer neigh(6)
c    neigh -- indices of current point's neighbors in 1D grid array
c    integer j,n
c    integer k, nrfill

do 500 n=1,npts
  if (cells(n) .eq. 'S') then
c    * set neigh() to indices of neighbors; *
c    * if an index is outside the grid substitute the closest *
c    * index in the grid *
    call indx3(n,grdpts,ptndx)
    neigh(1) = indx1(max(ptndx(1)-1,1),ptndx(2),ptndx(3),
& grdpts)
    neigh(2) = indx1(min(ptndx(1)+1,grdpts(1)),ptndx(2),
& ptndx(3),grdpts)
    neigh(3) = indx1(ptndx(1),max(ptndx(2)-1,1),ptndx(3),
& grdpts)

```

```

        neigh(4) = indx1(ptndx(1),min(ptndx(2)+1,grdpts(2)),
&          ptndx(3),grdpts)
        neigh(5) = indx1(ptndx(1),ptndx(2),max(ptndx(3)-1,1),
&          grdpts)
        neigh(6) = indx1(ptndx(1),ptndx(2),
&          min(ptndx(3)+1,grdpts(3)),grdpts)
c      * set each '*' neighbor to '2' *
        do 400 j=1,6
            if (cells(neigh(j)).eq.'*') then
                cells(neigh(j)) = '2'
            endif
400        continue
        endif
500 continue

        return
        end

c-----
        subroutine grdout(cells,grdfil, unitno, npts, grddiv, grdpts,
&offset,sfthik)
c
c  --called from CHEMGRID
c  --writes out grids; makes a formatted "bump" file and unformatted
c  van der Waals and electrostatics files
c
c                                          ECMeng      4/91
c-----
        include 'chemgrid.h'
        integer sfthik
c sfthik -- thickness of the final surface in number of grid cells
c
        character*1 cells(*)
c cells -- array of grid points to be written
        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, 4I4)
        3 format (80A1)
        open (unit=unitno, file=grdfil(1:namend)//'.srf', status='new')
        write (unitno, 1) 'coarse grid'
        write (unitno, 2) grddiv, (offset(i), i=1,3), (grdpts(i), i=1,3),
&          sfthik
        write (unitno, 3) (cells(i), i=1, npts)
        close (unitno)
c
        return
        end
c-----

```

```

c -----
c
c      subroutine newgen(npts,divrat,ndesc,grdpts,fingrd)
c
c For each interior ('*') or surface ('S') cell in the initial surface,
c mark the elements of fincel which occupy the same region in space
c as filled ('*') cells
c
c called from surfgrid
c C. Corwin, June 1994
c -----
c      include 'chemgrid.h'
c
c      integer descen(512)
c descen -- array indices of current cell's descendants
c      integer i,n
c
c      * for each occupied cell in initial grid *
c      do 100 n=1,npts
c          if ((celtyp(n).eq.('*')) .or. (celtyp(n).eq.'S')) then
c              call subdiv(n,divrat,descen,grdpts,fingrd)
c              do 50 i=1,ndesc
c                  fincel(descen(i)) = '*'
c          50      continue
c          endif
c      100 continue
c
c      return
c      end
c -----
c -----
c
c      subroutine subdiv(n,divrat,descen,grdpts,fingrd)
c
c "Subdivide" the nth cell of the initial grid (with spacing grddiv)
c into smaller cells with spacing findiv, setting descen() to their
c indices in the final grid array.
c
c Called from newgen and stilsf
c C. Corwin, June 1994
c -----
c      include 'chemgrid.h'
c
c      integer n
c n -- index of cell in initial grid to be expanded
c      integer descen(512)
c descen -- indices of current cell's descendants in final grid
c
c      integer incrd(3)
c incrd -- (virtual) 3-D coordinates of initial cell n *
c      integer curdes
c curdes -- index of current descendant cell in descen()
c      integer i,j,k

```

```

c      * Find virtual 3-D coordinates of initial cell n *
      call indx3(n,grdpts,incr)

      curdes = 0
c      * for final virtual x-coordinates within initial cell n *
      do 300 i=((incr(1)-1)*divrat)+1,incr(1)*divrat
c      * for final virtual y-coordinates within initial cell n *
      do 200 j=((incr(2)-1)*divrat)+1,incr(2)*divrat
c      * for final virtual z-coordinates within initial cell n *
      do 100 k=((incr(3)-1)*divrat)+1,incr(3)*divrat
c      * Add 1-D index of descendant to list*
      curdes = curdes+1
      descen(curdes) = indx1(i,j,k,fingrd)
100      continue
200      continue
300      continue

      return
      end

c -----
c -----
c
c      subroutine stilsf(npts,divrat,ndesc,grdpts,fingrd,finoff)
c
c For each surface ('S') cell in the initial surface,
c check all descendants in final surface against atom list.
c Remove those final surface cells which no longer correspond to
c any atoms from the filled region by marking them ' '
c
c called from surfgrid
c C. Corwin, June 1994
c -----
      include 'chemgrid.h'

      integer descen(512)
c descen -- array indices of current cell's descendants
      real cutoff, cutsq
c cutoff -- maximum atom-grid point distance for them to be associated
c cutsq -- cutoff squared
      integer grcoor(3)
      real atcoor(3), ptcoor(3)
c grcoor -- coordinates of current point in grid units
c atcoor -- atom coordinates minus offset
c ptcoor -- current grid point coordinates minus offset
      real r2
c r2 -- square of point-atom distance
      integer i,j,k,n

c      * for each surface cell or '2' cell in initial grid *
      do 100 n=1,npts
      if ((celtyp(n).eq.'S').or.(celtyp(n).eq.'2')) then
      call subdiv(n,divrat,descen,grdpts,fingrd)
c      * for each of its descendants in final grid *
      do 50 i=1,ndesc
      fincel(descen(i)) = ' '

```

```

c          * check against atoms associated with initial cell *
c          do 40 j=1,numasc(n)
c              * find cutoff for this atom *
c              cutoff = vdwrad(celatm(n,j)) + sqrt(3.0)*findiv/2.0
c              cutsq = cutoff**2
c              * find coordinates of grid point and atom *
c              call indx3(descen(i),fingrd,grcoor)
c              do 30 k=1,3
c                  atcoor(k) = atcrd(celatm(n,j),k) - finoff(k)
c                  ptcoor(k) = float(grcoor(k)-1)*findiv
c              30 continue
c              * make grid cell '*' if atom is associated *
c              r2 = dist2(atcoor,ptcoor)
c              if (r2 .le. cutsq) then
c                  fincel(descen(i)) = '*'
c              endif
c              40 continue
c              50 continue
c          endif
c      100 continue

c          return
c          end

c -----
c -----
c
c      subroutine finsf(cells,npts,grdpts)
c
c      Examine grid points; change each '*' grid point with at least
c      one ' ' neighbor to 'S'
c      called from surfgrid
c
c      C. Corwin, June 1994
c -----
c
c      include 'chemgrid.h'
c
c      character*1 cells(*)
c      cells -- grid cells to be updated
c      integer ptndx(3)
c      ptndx -- grid indices of current grid point
c      integer neigh(6)
c      neigh -- indices of current point's neighbors in 1D grid array
c      integer j,n
c      real ascsum,nstar
c      ascsum -- sum of number of associated atoms
c      nstar -- number of * points (before conversion to S)
c      real srfsum
c      integer k, nrfill

c      ascmax=0
c      srfmax=0
c      ascsum=0.0
c      srfsum=0.0
c      nstar=0.0
c      nsrfpt=0
c      do 500 n=1,npts

```

```

c      if (cells(n) .eq. '*') then
c          * update statistics *
c          nstar = nstar +1.0
c          ascsum = ascsum + numasc(n)
c          if (numasc(n) .gt. ascmax) then
c              ascmax = numasc(n)
c          endif
c          * set neigh() to indices of neighbors; *
c          * if an index is outside the grid substitute the closest *
c          * index in the grid *
c          call indx3(n,grdpts,ptndx)
c          neigh(1) = indx1(max(ptndx(1)-1,1),ptndx(2),ptndx(3),
&              grdpts)
c          &      neigh(2) = indx1(min(ptndx(1)+1,grdpts(1)),ptndx(2),
c          &              ptndx(3),grdpts)
c          &      neigh(3) = indx1(ptndx(1),max(ptndx(2)-1,1),ptndx(3),
c          &              grdpts)
c          &      neigh(4) = indx1(ptndx(1),min(ptndx(2)+1,grdpts(2)),
c          &              ptndx(3),grdpts)
c          &      neigh(5) = indx1(ptndx(1),ptndx(2),max(ptndx(3)-1,1),
c          &              grdpts)
c          &      neigh(6) = indx1(ptndx(1),ptndx(2),
c          &              min(ptndx(3)+1,grdpts(3)),grdpts)
c          * set cells(n) to 'S' if a neighbor is blank *
c          do 400 j=1,6
c              if (cells(neigh(j)).eq. ' ') then
c                  cells(n) = 'S'
c              endif
400      continue
c          if (cells(n) .eq. 'S') then
c              * update surface statistics *
c              nsrfpt = nsrfpt + 1
c              srfsum = srfsum + numasc(n)
c              if (numasc(n) .gt. srfmax) then
c                  srfmax = numasc(n)
c              endif
c          endif
c      endif

c      * Count filled neighbors of blank cells *
c      * Mark as 'H' (for hole) any blank cell with 4 filled neighbors
c      *
c      if (cells(n) .eq. ' ') then
c          call indx3(n,grdpts,ptndx)
c          neigh(1) = indx1(max(ptndx(1)-1,1),ptndx(2),ptndx(3),
c          &              grdpts)
c          &      neigh(2) = indx1(min(ptndx(1)+1,grdpts(1)),ptndx(2),
c          &              ptndx(3),grdpts)
c          &      neigh(3) = indx1(ptndx(1),max(ptndx(2)-1,1),ptndx(3),
c          &              grdpts)
c          &      neigh(4) = indx1(ptndx(1),min(ptndx(2)+1,grdpts(2)),
c          &              ptndx(3),grdpts)
c          &      neigh(5) = indx1(ptndx(1),ptndx(2),max(ptndx(3)-1,1),
c          &              grdpts)
c          &      neigh(6) = indx1(ptndx(1),ptndx(2),
c          &              min(ptndx(3)+1,grdpts(3)),grdpts)
c          nrfill = 0
c          do 450 k=1,6

```



```

c          if ((cells(neigh(k)).eq.'*').or.(cells(neigh(k)).eq.
c      &      'S')) then
c          nrfill = nrfill+1
c          endif
c 450      continue
c          if (nrfill .ge. 4) then
c              cells(n) = 'H'
c          endif
c      endif
c      * End counting filled neighbors *

500 continue

c      ascavg = ascsum/nstar
c      srfavg = srfsum/nsrftpt

      return
      end

-----
c      subroutine indx3(i,grdpts,ind)
c
c      --converts the index of a grid point in a 1-dimensional array into
c      the 3-dimensional (virtual) indices
c
c      integer i, grdpts(3), ind(3)
c
c      --first (x) index equals remainder of (total # of points)/(# of grid
c      points along x)
c      ind(1) = mod(i, grdpts(1))
c      if (ind(1) .eq. 0) ind(1) = grdpts(1)
c      --second (y) index equals remainder of (total # of lines)/(# of grid
c      points along y)
c      ind(2) = mod((i - ind(1))/grdpts(1) + 1, grdpts(2))
c      if (ind(2) .eq. 0) ind(2) = grdpts(2)
c      --third (z) index equals total # of planes
c      ind(3) = (i - ind(1) - (ind(2)-1)*grdpts(1))/(grdpts(1)*grdpts(2))
c      & + 1
c      return
c      end
-----

-----
c
c      subroutine growsf(finpts,fingrd,sfthik)
c
c      Examine grid points;
c      Change each ' ' neighbor of an 'S' cell to 'S' and repeat until
c      sfthik, the desired thickness of the 'S' layer, is reached
-----
c      include 'chemgrid.h'

      integer sfthik
c  sfthik -- desired thickness of the layer of 'S' cells, in grid units
      integer ptndx(3)
c  ptndx -- grid indices of current grid point
      integer neigh(6)
c  neigh -- indices of current point's neighbors in 1D grid array
      integer i,j,n

```

```

do 600 i=1, sfthik-1
  do 500 n=1,finpts
    if (fincel(n) .eq. 'S') then
c      * set neigh() to indices of neighbors; *
c      * if an index is outside the grid substitute the closest
c      *
c      * index in the grid *
      call indx3(n, fingrd, ptndx)
      neigh(1) = indx1(max(ptndx(1)-1,1),ptndx(2),ptndx(3),
&                    fingrd)
&      neigh(2) = indx1(min(ptndx(1)+1,fingrd(1)),ptndx(2),
&                    ptndx(3),fingrd)
&      neigh(3) = indx1(ptndx(1),max(ptndx(2)-1,1),ptndx(3),
&                    fingrd)
&      neigh(4) = indx1(ptndx(1),min(ptndx(2)+1,fingrd(2)),
&                    ptndx(3),fingrd)
&      neigh(5) = indx1(ptndx(1),ptndx(2),max(ptndx(3)-1,1),
&                    fingrd)
&      neigh(6) = indx1(ptndx(1),ptndx(2),
&                    min(ptndx(3)+1,fingrd(3)),fingrd)
c      * set fincel(neigh(j)) to 'N' for each blank neighbor *
c      * (use N to differentiate surface points already present
c      *
c      * from those created this iteration *
      do 400 j=1,6
        if (fincel(neigh(j)).eq.' ') then
          fincel(neigh(j)) = 'N'
          nsrfpt = nsrfpt + 1
        endif
400      continue
      endif
500    continue
c    * Mark surface points created this iteration as 'S' *
      do 550 j=1,finpts
        if (fincel(j) .eq. 'N') then
          fincel(j) = 'S'
        endif
550    continue
600  continue

      return
      end

```

c -----

Source code for the surface scoring routines

```

c -----
c
c      subroutine surfscore(ligats,xatm,score)
c
c      Some of this code was adapted from the CHEMGRID program written by
c      Elaine Meng
c

```

```

c C. Corwin, May 1994
c -----
    include 'max.h'
    include 'chemscore.h'
    include 'surf.h'

    integer ligats
c ligats -- number of heavy atoms in ligand
    real xatm(3,maxlig)
c xatm -- rotated and translated ligand coordinates
    real score
c score -- surface-overlap score of current ligand

    real dist2

    integer i,j,k,n
    integer pnt
c pnt -- index of current grid cell
    integer nearpt(3)
c nearpt -- grid indices of grid point nearest atom
    real rcrd(3)
c rcrd -- coordinates of current atom relative to grid origin
    integer grx, gry, grz
c grx,gry,grz -- grid indices of current grid point
    real r2
c r2 -- square of distance between current atom and grid point
    integer currx
c currx -- number of grid points currently marked 'X'
    integer Xlist(50000)
c Xlist -- list of grid locations marked as 'X'

    currx = 0
    score = 0.0
    do 500 n=1,ligats
c      * find location (in grid units) of grid point nearest atom *
c      * (add 1 because grid starts at (1,1,1) *
c      * ignore atoms outside grid *
        do 50 i=1,3
            rcrd(i) = xatm(i,n) - srfoff(i)
            nearpt(i) = nint(rcrd(i)/srfdiv) + 1
            if (nearpt(i) .gt. srfgrd(i)) goto 499
            if (nearpt(i) .lt. 1) goto 499
50        continue

c      * loop through grid points on the list of offsets for the *
c      * current receptor atom *
        do 400 i=1,nshell(vdwtyp(n))
            grx=nearpt(1) + xshell(i,vdwtyp(n))
            gry=nearpt(2) + yshell(i,vdwtyp(n))
            grz=nearpt(3) + zshell(i,vdwtyp(n))
            pnt = indxl(grx,gry,grz,srfgrd)
            if (srfval(pnt).eq.'S') then
                score = score + 1
c            * mark point 'X' once it has been counted in score *
c            * to avoid counting it more than once *
                srfval(pnt) = 'X'
                currx = currx+1
                Xlist(currx) = pnt

```

```

        endif
400    continue

499 continue
500 continue

c    * Change grid points temporarily marked 'X' back to 'S' *
      do 600 i=1,currx
          srfval(Xlist(i)) = 'S'
600    continue

      return
      end
c -----
c -----
c
c    subroutine mkshel
c
c    MKSHEL - for each vdW type, calculate the offsets, in grid units, i
c             from an atom to each grid cell in the spherical shell around
c             it which will be checked in surface scoring
c    Some of this code was adapted from the CHEMGRID program written by
c    Elaine Meng
c
c    C. Corwin, May 1994
c -----
      include 'max.h'
      include 'chemscore.h'
      include 'surf.h'

      real dist2

      integer i,j,k,n
      real cutoff, cutsq, grdcut
c    cutoff -- maximum distance from an atom to a grid point for scoring
c    cutsq -- cutoff squared
c    grdcut -- cutoff in grid units
      real rcrd(3)
c    rcrd -- coordinates of current atom relative to grid origin
      real gcrd(3)
c    gcrd -- real coordinates of current grid point, relative to grid
origin
      real vdwrsg
c    vdwrsg -- square of the vdW radius of current atom
      real r2
c    r2 -- square of distance between current atom and grid point

      do 500 n=1,nvtyp
c          * set atom coords to the origin for calculating offsets *
          do 50 i=1,3
              rcrd(i) = 0.0
          50    continue
              nshell(n) = 0

c          * cutoff is the van der Waals radius of the atom plus *
c          * the user-determined thickness of the surface *

```

```

c      * convert cutoff to grid units *
      cutoff = vdwrad(n) + (sfthik-1)*srfdiv +
&          sqrt(3.0)*srfdiv/2
      cutsq = cutoff**2
      grdcut = int(cutoff/srfdiv + 1.0)
      vdwrsg = vdwrad(vdwtyp(n))**2

c      * loop through grid points within the cutoff cube of the *
c      * current vdw type *
      do 400 i=(-1*grdcut),grdcut
        gcrd(1)=float(i-1)*srfdiv
        do 300 j=(-1*grdcut),grdcut
          gcrd(2)=float(j-1)*srfdiv
          do 200 k=(-1*grdcut),grdcut
            gcrd(3)=float(k-1)*srfdiv
c            * if grid point is within cutoff distances of atom *
c            * add its indices to list *
            r2 = dist2(rcrd,gcrd)
            if ((r2 .le. cutsq).and.(r2.ge.vdwrsg)) then
              nshell(n) = nshell(n) + 1
              if (nshell(n).gt.maxshl) then
                write(6,*) 'Shell size bigger than maxshl'
                write(6,*) 'cube size = ',8*(grdcut**3)
                stop
              endif
              xshell(nshell(n),n)=i
              yshell(nshell(n),n)=j
              zshell(nshell(n),n)=k
            endif
          200 continue
        300 continue
      400 continue

      499 continue
      500 continue

      return
      end

```

c -----

Appendix 4: A Beginner's Guide to DOCK 3.5

Scope of This Document

This document is intended as a supplement to the DOCK manual. It describes the steps a new user would typically take to apply the programs to a macromolecule and potential ligands of interest. While the DOCK manual describes in detail the various input and output files, this guide is meant to convey in informal terms the process as a whole. Some of the difficulties we have encountered as well as approaches we have found useful are discussed.

What DOCK Can Do for You

DOCK is a program for locating feasible binding orientations, given the structures of a "ligand" molecule and a "receptor" molecule. What is considered feasible depends on how the orientations are evaluated. Current options are a contact (shape-fitting) score, a force field interaction energy, and an electrostatic energy calculated by using a DelPhi potential map (the program DelPhi is not distributed with DOCK). Atoms may be labeled so that they may fall only in chemically appropriate regions (as labeled by the user), and orientations may be varied to optimize their force field scores. In SINGLE mode, DOCK generates many orientations of one ligand. In SEARCH mode, orientations are generated for each of the molecules in a database in turn; the best-scoring orientation of each molecule is saved, and the best-scoring molecules are written out. Some of the molecules in the list of best-scoring compounds, perhaps with modifications, may be interesting as potential new ligands for the receptor.

Basic DOCK References

New users should become familiar with the algorithms used by DOCK. Reading these papers is strongly recommended:

Kuntz, I.D., Blaney, J.M., Oatley, S.J., Langridge, R., Ferrin, T.E. (1982) *J. Mol. Biol.* **161**: 269. The original DOCK publication. This paper outlines the docking process and describes the way spheres are generated. Corresponds to DOCK version 1.0.

Shoichet, B.K., Bodian, D., and Kuntz, I.D. (1992) *J. Comp. Chem.*, **13**: 380. The current methods for generating matches between ligand atoms and receptor spheres, introduced in DOCK 2.0, are described here.

Meng, E.C., Shoichet, B.K., and Kuntz, I.D. (1992) *J. Comp. Chem.*, **13**: 505. Describes the use of DelPhi electrostatic potentials and force-field-like electrostatic and van der Waals potentials for scoring. DOCK 3.0.

These papers are also worthwhile reading, especially if you plan to use the features and techniques they describe:

DesJarlais, R.L., Sheridan, R.P., Seibel, G.L., Dixon, J.S., Kuntz, I.D., and Venkataraghavan, R. (1988) *J. Med. Chem.* **31**: 722. The first use of DOCK to search molecule databases. DOCK version 1.1.

Shoichet, B.K., and Kuntz, I.D. (1991) *J. Mol. Biol.*, **221**: 327. Discusses protein-protein docking.

Meng, E.C., Gschwend, D.A., Blaney, J.M., and Kuntz, I.D. (1993) *Proteins: Structure, Function, and Genetics*, **17**: 266. Introduces the use of minimization to improve the scores of promising matches from DOCK. Minimization is a new option in DOCK 3.5.

Shoichet, B.K., and Kuntz, I.D. (1993) *Protein Engineering*, **6**: 223. Introduces the concept of labeled or colored matching (the implementation in DOCK 3.5 is slightly different from what is described in this paper).

A complete list of Kuntz group papers referring to DOCK is in the DOCK3.5 manual.

Overview of the DOCK package

The basic requirement for docking is a structure of the macromolecule of interest. The docking procedure can be divided into four general stages: site characterization, calculation of grids for scoring, preparation of databases, and DOCK itself.

For DOCK, sites on the receptor are characterized by sphere clusters. These are simple geometric descriptions of the volume available to ligands. The program SPHGEN calculates these sphere clusters using the molecular surface of the receptor produced by Connolly's MS program. (MS is not distributed with DOCK, but may be obtained from the Quantum Chemistry Program Exchange.)

While site characterization prepares the information needed to generate ligand orientations, grid calculations are necessary so that the orientations can be evaluated, or scored. The kind of scoring desired dictates which programs need to be run. For contact scoring, DISTMAP is used to generate the grid. This grid is also necessary for DelPhi electrostatic scoring, since it determines which orientations must be thrown out due to bad contacts with receptor atoms. Electrostatic scoring requires a potential map from DelPhi (Honig *et al.*, Columbia University). The grid used for force field scoring, which includes both steric and electrostatic terms, is produced by CHEMGRID. The CHEMGRID result may be used alone or combined with the DISTMAP grid to use both contact and force field scoring.

If DOCK will be used to search a database of potential ligands, their structures must be converted to DOCK 3 database format; MOL2DB can create

the database from an input list of ligands in SYBYL ASCII (MOL2) format. MOL2DB can also be used to label atoms by type and chemical environment so that they can be restricted to matching only chemically appropriate spheres. The spheres are given labels using COLSPH.

The final stage of the process is running DOCK and viewing the results. DOCK attempts to match ligand centers to receptor sphere centers, then scores each orientation using the information in the precalculated grids. At the user's option, DOCK can attempt to adjust ligand orientations to optimize their scores. The best-scoring molecules or orientations may be viewed using a molecular graphics program.

A necessary side effect of having many options that the user can control is requiring the user to enter many numbers, which can result in confusion. We hope to make the learning curve an easier place to be with this guide.

A Caution Concerning Disk Space

The output from some of the programs associated with DOCK, particularly MS, SPHGEN, and DOCK itself, may require substantial amounts of disk storage. It is a good idea to be cautious at first; check before starting your job to make sure there is space available. While DOCK jobs are running, check to be sure they are not creating overly large files, especially if you have increased the bin parameters.

Working With Macromolecular Models and Generating the Molecular Surface

Removing Ligands and Crystallographic Waters

The macromolecular structure you are working with may include a ligand, and crystal structures usually contain water molecules and sometimes

ions which were found on the surface of the protein. These molecules are usually not included in the structure used to generate the molecular surface. To prepare for molecular surface generation, make a copy of the protein coordinate file. If there is a ligand present, remove it by deleting all of its records (they often start with HETATM in Brookhaven Protein Data Bank format files) from your copy of the file. (Note - sometimes, as in the case of a cofactor or catalytic metal ion, it may make chemical sense to keep a ligand in the PDB file.) Whether or not crystallographic waters and ions should be preserved when generating surfaces for use by SPHGEN is a matter of some debate. In structures of complexes, water molecules and ions are often found in the protein binding pocket along with the ligand(s). However, ligands can displace waters and ions, and the volume of a receptor site will be explored more completely if the waters and ions are removed, so if you don't have particular reasons for preserving any of the water molecules or ions in the crystal, it is probably best to remove all of them. Waters are usually located near the end of the PDB file and are often HETATM records with HOH or WAT residue types. Ions are often near the waters in the PDB file.

Please note that the PDB file used for generating the molecular surface should not include hydrogen atoms. NMR structures will include hydrogens; delete the hydrogens from a copy of each structure and use that copy in MS.

Creating the Molecular Surface

The dot surface which will be used to produce spheres is generated by the program MS (available from QCPE). When setting up for docking, it is acceptable just to generate surface for the site of interest and adjacent regions; this will also reduce the computer time used by SPHGEN. The surface points must have associated normals.

If you use the QCPE version of MS, you must run REFORMATMS to convert the surface to the format used by SPHGEN (both formats are described in the manual section on REFORMATMS). REFORMATMS is interactive and requires the surface and the PDB file used to generate the surface.

Users of the UCSF MidasPlus package may use the output from the dms program directly as input for SPHGEN.

Representing the Site With Spheres

SPHGEN uses the points of the molecular surface and their associated normals to determine spheres to fill the site. It then reduces the number of spheres to one per atom and groups them into clusters. You can inspect these clusters and regroup the spheres if necessary.

Creating INSPH

The parameters which tell SPHGEN exactly how to create the surface are placed in a file called INSPH, which must be present when SPHGEN is run. The contents of this file are described in the DOCK manual. To create it, make a file with each variable on a separate line. Most of the parameter values given in the DOCK manual should work fine. You will need to replace *msfil* with the name of your surface file and *outfil* with the desired name of your output file.

Running SPHGEN

SPHGEN must use the directory containing INSPH as its working directory; this means that it should be started while you are in that directory.

The SPHGEN output file contains clusters of spheres which have been selected and grouped by SPHGEN; the clusters are listed in order of decreasing size. The last cluster, numbered 0, contains all the spheres produced. It may be used with the program CLUSTER to make new sphere clusters if the original clustered output doesn't describe the site well.

Looking at the Output - SHOWSPHERE

Once you've generated spheres, you should look at the sphere clusters using a molecule display program. SHOWSPHERE may be used to generate a PDB-like file of sphere centers for display. It can also generate a surface for the sphere cluster (in the MS format used by SPHGEN). SHOWSPHERE is interactive. You will be prompted for the name of the cluster file (that is, the SPHGEN output), the number of the cluster, and names for the desired output file. In the pdb-like file of sphere center coordinates, each sphere is a separate residue and the spheres are separated by TER cards.

Getting a Good Sphere Cluster

Displaying the protein and sphere centers together should tell you how each sphere cluster is related to the site you are trying to represent. Examine sphere clusters until you find one that occupies the region (or regions) into which you want to dock ligands. Clusters of 50 or fewer spheres are best; larger numbers of spheres will cause DOCK to use more computer time. It is generally unwise to try docking with more than 100 spheres, although you may be able to use more if your database is small or you are using chemical matching. Initial sphere clusters are sometimes spider-like structures which include the area of interest but also branch into other regions. If your cluster has too many spheres, branches out, or is unsatisfactory for some other reason, you can correct the problem.

The easiest way to fix a sphere cluster is to use graphics to identify spheres that you don't really need, then remove them. When you've found the unnecessary ones, go back to the *original* sphere cluster file (i.e. the one from SPHGEN) and delete the corresponding lines — the residue number in the PDB-like file of centers is the first number in the line in the sphere file.

Remember to change the number of spheres listed on the line with the cluster number to reflect the deletions.

If your cluster is large — more than about 100 spheres — and deleting spheres by hand looks too tedious, you can use CLUSTER to break it into smaller clusters. CLUSTER is described in the DOCK manual; read the documentation completely before you try it. Start with the parameters given and experiment with the values; small changes can make a big difference in the result. Be aware that if the best cluster found is the same as the original input cluster, the program will appear not to have done anything.

The two methods just described may be combined if the best CLUSTER output is not quite right. More spheres can be deleted from the new cluster, or, if the new cluster is too small, additional spheres may be chosen graphically. A cluster containing all the desired spheres may then be created by editing the SPHGEN output.

If nothing else works, it is possible to run CLUSTER on all possible spheres rather than a preselected group. Use the analytical clustering algorithm in CLUSTER on cluster 0, and experiment until you get what you want. Flagging spheres in important regions of the site may help.

Creating the Scoring Grids

Before running DOCK, you must choose which scoring option you will use and generate the scoring grid or grids required. Current scoring options are contact only, contact and DelPhi, contact and force field, and force field only. All the options involving contact scoring require the grid generated by DISTMAP. DelPhi scoring requires a potential map from DelPhi, and force field scoring uses the result from CHEMGRID.

The Contact Scoring Grid (DISTMAP)

DISTMAP produces a score for each point in a cubic lattice based on how many receptor atoms are positioned to make favorable or unfavorable contacts with that point. The resulting grid will be used by DOCK to score each atom in a ligand orientation; the score of the orientation is the sum of the atom scores.

Positioning the Grid

DISTMAP uses a PDB-format file as input; by default it will make a lattice which encloses all the atoms in that file. However, it is really only necessary to enclose the region of space where the ligand atoms will lie, plus the space containing those receptor atoms which might come close to ligand atoms. In other words, the grid should contain the site of interest and a generous amount of its surroundings but need not take in the whole protein.

The interactive program SHOWBOX is recommended for defining the size and shape of the grid. One way to do this is to make a box which encloses your sphere cluster and add an extra margin which encloses all the receptor atoms which might contact the ligands. The box generated is in pdb format; it should be viewed along with the receptor and possibly regenerated until it seems appropriate.

Creating INDIST and Running DISTMAP

Input to DISTMAP is placed in a file called INDIST, which is described in the DOCK manual. You will need to change *pdbnam* and *scoren*, the input and output file names, but the rest of the parameters given in the example are reasonable starting values. Be sure to include the name of the box file from SHOWBOX on the last line. DISTMAP, like SPHGEN and all the programs that use similar input files, must be started while in the directory where INDIST is located.

The Electrostatic Potential Map (DelPhi)

DelPhi may be used to calculate the electrostatic potential due to the receptor. DOCK can use this potential grid to calculate electrostatic scores for ligands with partial charges.

DelPhi is not supplied with DOCK; it is available from Barry Honig, Columbia University, or from Biosym. If you have the program and wish to use it for electrostatic scoring, follow the directions supplied with it to generate a potential map for your protein. DOCK currently reads the potential map (phi file) format used by the export_v3 version of DelPhi.

The Force Field Scoring Grid (CHEMGRID)

CHEMGRID saves information about the steric and electrostatic environment at each point on a grid, so that ligand orientations can be scored rapidly during a DOCK run.

Positioning the Grid

You determine the location and dimensions of the region to be gridded by using the program SHOWBOX to create a box which contains the desired region. For CHEMGRID, the box should enclose the volume that the ligand orientations are likely to occupy (note that this is slightly smaller than the volume required for DISTMAP). An easy way to accomplish this is to generate a box which encloses the spheres to be used for docking along with an extra margin. The box generated should be viewed along with the receptor and possibly regenerated until it looks good. Unlike the limiting box that can be used as input to DISTMAP, the grid region does not determine which receptor atoms are used in the calculation; all receptor atoms within the specified cutoff distance of any grid point will be included.

Preparing the Receptor File

In order for the steric and electrostatic environment within the site to be evaluated, the program will need to associate each of the receptor atoms

with the proper charge and steric qualities. This is accomplished by matching their names to names in a parameter file. Three parameter files for macromolecules are supplied with this release:

na.table.ambcrg	nucleic acid parameters
prot.table.ambcrg.ambH	protein parameters; AMBER hydrogen names
prot.table.ambcrg.pdbH	protein parameters; PDB hydrogen names

All three contain AMBER united-atom parameters. The only explicit hydrogens included are those bonded to polar atoms (anything except carbon). The protein files differ only in the hydrogen-naming conventions used; the heavy atom names are PDB standard in both. In your receptor file, the atom names should match the names in the parameter file you will be using, and hydrogens bonded to polar atoms should be present. It is all right to have all the hydrogens and even lone pairs present since any atom not found in the parameter file receives zero charge and volume. Special attention should be given to the names of atoms at the termini and the residue names for histidine and cysteine. The different protonation states of histidines correspond to different residue names: HIP for positively charged (hydrogens on both nitrogens), HID for neutral with the delta nitrogen protonated, and HIE for neutral with the epsilon nitrogen protonated. CYS refers to a cysteine with a free sulfhydryl group; CYX refers to a cysteine involved in a disulfide bond (a half-cystine). Note that some structures in the PDB use CYS in disulfides; these should be edited to CYX.

Creating INCHEM and Running CHEMGRID

Input to CHEMGRID is placed in a file called INCHEM, which is described in the DOCK manual. The input values should be placed in the file as shown in the example. You will need to replace *recfil* with the name of your receptor file. Replace *table* and *vdwfil* with the locations of your chosen

parameter table and *vdw.parms.amb* on your system. *Inbox* should be replaced with the name of the SHOWBOX output file.

Grddiv values between 0.2 and 0.5 are recommended; fine grids are preferred. Any combination of grid point spacing and x, y, and z dimensions can be used as long as the number of points does not exceed the maximum array size specified in the program. If this happens, you will get a message when you run CHEMGRID. To resolve the problem, the grid spacing may be increased or the box dimensions may be decreased. DOCK may also be recompiled with larger array sizes if your computer's memory allows; edit the file 'chemgrid.h' appropriately.

A dielectric function of 4.0r or 4.5r and a cutoff of 10.0 or more Ångstroms are appropriate in most cases. (This dielectric corresponds to an *estype* of 1 and *esfact* of 4.0 or 4.5.) If a constant dielectric is selected, an "infinite" cutoff (one large enough to include the whole receptor) should be used. We tend to use close contact limits of 2.0-2.5 and 2.5-3.0 Ångstroms for receptor polar and nonpolar atoms, respectively. The close contact limits do not affect the force field scores that orientations receive, but they determine which orientations are thrown out when "force field scoring only" is performed in DOCK. The resolution of the protein structure should be kept in mind when setting these limits; if the receptor atom positions are not very well-defined, it is best not to constrain the results too strongly based on these positions.

Grdfil will be the beginning of the output file names.

Although we have recommended certain values, we would like you to try whatever you feel is appropriate, based on your knowledge of the parameters and their significance.

Parameterization takes place in the first few seconds of a CHEMGRID run, and produces two files: PDBPARM, which lists the coordinates of each receptor atom together with the associated parameters, and OUTPARM, which reports each atom not found in the parameter file and the apparent net charge of the receptor. A long list of hydrogen atoms is normal, since there are no parameters for hydrogens not attached to polar atoms. It is important to check the net charge since a strange value (for example, a noninteger or a very large number) can alert you to parameterization problems. Such values usually mean that something is wrong with the hydrogens or the residue names.

Three output files, named *grdfil.bmp*, *grdfil.esp*, and *grdfil.vdw*, make up the actual grid.

Preparing Ligand Molecules

Before you can run DOCK, you must make sure that the ligands you intend to use are in a format which DOCK can read. Consult the table below for a list of acceptable formats. SINGLE mode reads files in all the listed formats directly. Formats other than DOCK databases must be converted to DOCK databases for SEARCH mode, using the indicated programs. Version numbers for DOCK databases identify the database format only; all are acceptable input for DOCK 3.5 SEARCH mode, but SINGLE mode can use only the DOCK 3.0 and DOCK3.5 formats. Please note that ligands with charges should also have hydrogens.

SINGLE Mode File Formats

Format	Charges?	Compatible Scoring Options
standard PDB	no	contact
extended PDB	yes	all
SYBYL ASCII (MOL2)	yes	all

SYBYL ASCII (MOL2)	no	contact
DOCK database (version 3.0)	yes	all
DOCK database (version 3.5)	yes	all

SEARCH Mode File Formats

Format	Charges?	Conversion Program	Compatible Scoring Options
SYBYL ASCII(MOL2)	no	MOL2DB	contact
SYBYL ASCII(MOL2)	yes	MOL2DB	all
DOCK database (version 1.1 or 2.0)	no		contact
DOCK database (version 3.0)	yes		all
DOCK database (version 3.5)	yes		all
CSD	no	MKDB	contact

MKDB is an interactive program which reads molecules in CSD format and writes them in DOCK 2.0 format. DOCK does not consider the hydrogens during its calculations, but they should be included in the MKDB output if you want them on the oriented ligands written out by DOCK.

MOL2DB is also interactive; it reads SYBYL ASCII (MOL2) format. If you wish to use contact scoring only and your ligands have no charges, choose version 2.0 output. Otherwise, your ligands should have hydrogens and charges and you should create a version 3.5 database. MOL2DB may be used to label ligand atoms for chemical matching, described below.

DOCK databases may include any number of molecules. They consist of lists of molecule records; you may edit them after they are created to delete molecules or to separate some molecules into smaller databases. If you create smaller files from a DOCK 3.5 database, be sure to include the header information in each file. (The first line should read DOCK3.5 ligand_atoms,

and the chemical matching information should follow it starting on the second line. You can copy the header from the beginning of the original database.)

Labeling Atoms and Spheres for Chemical Matching (Optional)

In some cases, you may wish to label receptor spheres and ligand atoms for chemical matching. When chemical matching is used, labeled (sometimes called colored) spheres are permitted to match labeled atoms only if their labels correspond as specified in the INDOCK file (for example, positively charged spheres might be allowed to match only negatively charged ligand atoms). Labeled spheres may still match unlabeled atoms and unlabeled spheres may match labeled atoms. Labeling reduces the number of orientations which must be scored somewhat, but the overall DOCK run will not be significantly shorter unless many of the spheres and atoms are labeled. Chemical matching using labels assigned by the programs COLSPH and MOL2DB is most useful in conjunction with contact scoring or for highly charged or polar sites.

COLSPH may be used to label spheres according to the electrostatic potential at their location in space. You will need a sphere cluster file and either a potential map from DelPhi or the force field grid files created by CHEMGRID. Create a file specifying the labels and the range of receptor potential or of the electrostatic potential component of the force field grid which corresponds to each label. On each line, list one label and the upper and lower bounds of its electrostatic potential range. The potential will be evaluated at each sphere and the sphere will be assigned a label if the potential is in the appropriate range. Run COLSPH interactively. The program will prompt you for the map type and name, the name of the range

file you just created, and the input and output sphere file names. (COLSPH will color spheres in all clusters in the sphere file.)

To color ligand atoms, begin with a list of molecules in SYBYL ASCII(MOL2) format. Run MOL2DB interactively (see the DOCK 3.5 manual). At the coloring prompt, enter one label at a time with its corresponding SYBYL atom type. To indicate that a given atom type should be (or not be) in a particular functional group, you may include on the same line a second atom type and the number of bonds which separate it from the first atom. Atoms of the first type will then be labeled only if they are the specified number of bonds from an atom of the second type. If the number of bonds is negative, atoms of the first type which are the specified number of bonds from the second type will not be labeled (this is useful for excluding some atoms which meet a previous labeling criterion). Enter a blank line to end label entry.

In your INDOCK file, you will need to specify which of the labels you have assigned to spheres should match which of the ligand atom labels.

Running DOCK

Setting Up Directories

Before starting DOCK, it is a good idea to confirm that there is disk space where you plan to put the output.

Each DOCK run requires a file called INDOCK, which contains the input parameters. Since some of the parameters will be different for each run, a directory should be created to contain each INDOCK and probably the corresponding output. Using a separate directory is a good idea even for just one run.

Creating INDOCK

Parameters in the INDOCK file are specified by keywords, which are listed one to a line. The desired value of each parameter follows the keyword

on the same line. You may create the INDOCK file with a text editor or copy one of the examples supplied with DOCK and modify it to suit your needs. The DOCK 3.5 manual includes several sample INDOCK files. You only need to include in your INDOCK file those parameters relevant to your calculation or variables whose values you want to change from their defaults. Any line beginning with '#' will be considered a comment and ignored; comments can be quite useful in making the file more understandable.

Keywords available for INDOCK are listed here. The values suggested are reasonable initial guesses; they may not be the best values for your particular system. We suggest that you experiment with them to see what works best for you. Pay particular attention to the variables listed in *italics* since these are most likely to need changing.

The input and output parameters are relevant to all DOCK runs:

Input and Output Parameters for INDOCK

Keyword	Default	Suggested Value
scoring_option	contact	contact <i>OR</i> contact+delphi <i>OR</i> contact+forcefield <i>OR</i> forcefield (choose desired scoring option)
mode	single	single <i>OR</i> search
receptor_sphere_file		path name of file containing spheres
cluster_numbers		number(s) of cluster(s) you are DOCKING to
ligand_atom_file		name of ligand atom file
ligand_type	C	C to use ligand atom coordinates for docking (most runs) S to use ligand spheres (mostly for protein-protein docking)
restart	n	y if restart run n (or omit keyword) if not

ligand_sphere_file		name of ligand sphere file (omit if not using ligand spheres)
output_file_prefix		string to be used as the beginning of output file names
output_hydrogens	y	y to include hydrogens in ligand output n to leave them out

The matching parameters determine how many different ligand orientations DOCK will examine. Increasing or decreasing the number of orientations increases or decreases the amount of computer time used by DOCK and the disk space used by single-mode runs. It may take some experimentation with these parameters to discover what works best for a particular system. Be cautious when increasing bin sizes and bin overlaps; small changes can produce large increases in the number of orientations generated.

These parameters are relevant to all DOCK runs:

Matching Parameters for INDOCK

Keyword	Default	Suggested Value
distance_tolerance	1.5	1-1.5 (larger values for poorer-quality structural data)
nodes_maximum	4	4 - 6
nodes_minimum	4	4
size_ratio	0.0	0.0
ligand_binsize	1.0	1.0 or smaller
ligand_overlap	0.0	0.2 or smaller
receptor_binsize	1.0	1.0 or smaller
receptor_overlap	0.0	0.2 or smaller

These parameters are used for single mode runs, where many orientations are generated for one molecule:

Single Mode Parameters for INDOCK

Keyword	Default	Suggested Value
contact_minimum	0.0	0.0

energy_maximum	100.0	100.0
rmsd_override	0.0	0.0

These parameters are used for search mode runs, where the best orientation is saved for a list of best-scoring compounds:

Search Mode Parameters for INDOCK

Keyword	Default	Suggested Value
atom_minimum	1	6 - 10
atom_maximum	80	80
number_save	100	100-200
normalize_save	0	(100-200 if normalized output is desired; omit otherwise)
molecules_maximum	10000	larger than the database used
restart_interval	100	100
initial_skip	0	0 (or omit)

Which scoring parameters to use depends on the scoring option chosen.

Distmap_file should be included for any option involving contact scoring (contact, contact+delphi, contact+forcefield). Delphi_file is used only for contact+delphi. The remaining parameters in this table pertain to force field scoring and are used with the contact+forcefield or forcefield options.

Scoring Parameters for INDOCK

Keyword	Default	Suggested Value
distmap_file		name of distmap output (for contact scoring)
delphi_file		name of DelPhi receptor potential file (for DelPhi electrostatic scoring)
vdw_parameter_file		location of 'vdw.parms.amb' on your system(for force field scoring)
chemgrid_file_prefix		prefix of chemgrid output file names

bump_maximum	0	0 - 2 (larger values may be better if your structural data are of poor quality)
interpolate	y	y
vdw_maximum	1.0E+10	1.0E+10 (or omit)
electrostatic_scale	1.0	1.0 (or omit)
vdw_scale	1.0	1.0 (or omit)

These parameters pertain to focusing (informally called zooming):

Focusing Parameters for INDOCK

Keyword	Default	Suggested Value
focusing_cycles	0	0 (no focusing) or omit keyword
focus_bump	bump_maximum	omit

The chemical matching parameters are used to specify how labeled spheres and atoms (if any are used) are to be matched. Leave them out if you do not use chemical matching.

Chemical Matching Parameters for INDOCK

Keyword	Default	Suggested Value
chemical_matching	n	y if chemical matching is used; omit keyword otherwise
case_sensitive	y	n
match		pairs of labels to be matched; repeat once for each sphere label - atom label pair

Parameters for force field score optimization (minimization) of orientations are not listed here. Minimization varies the position of ligands in order to find orientations with improved force field scores. It is a useful tool, but its options are somewhat involved and you will probably want to familiarize yourself with the basic matching and force field scoring parameters before you try it out. Consult the DOCK 3.5 manual for a description of how to use

minimization. It lists the relevant INDOCK keywords and gives guidelines for choosing parameter values.

Running DOCK

Before running DOCK it is a good idea to check whether there are other jobs running on the same machine. DOCK runs use substantial amounts of cpu time; consider any other users sharing your computers when deciding whether to start more than one run at a time. Be aware of any policies your site has regarding cpu time used.

Start DOCK from the directory you created for INDOCK. Check a few minutes after you start the run to be sure that it is still going; if it has stopped, look for mistakes in the input. Beginners should check disk usage occasionally while the job is running, just in case the program is creating incredibly large files which might overflow the available space.

During a search run (which can take a few hours or several days to finish), you can follow DOCK's progress through the database by looking at the last few lines of OUTDOCK. The number preceding the last "nathvy" tells approximately how many ligands have been examined.

Restarting a Search Run

In search mode, DOCK periodically saves in the output file the information necessary to restart the search from its current location in the database. If there is a power failure or the system crashes, you can set up a new run to start where the last one was stopped. First, you must delete or rename OUTDOCK, since DOCK will try to create a new OUTDOCK file, and it cannot do so if one already exists. Then set the restart parameter in INDOCK to y and start the job again. (Do not change the remaining files, since DOCK needs them to restart successfully.) When the restarted run finishes, the sorted list of ligands in the output file will include the top scorers from the

entire database. However, some of the statistics in OUTDOCK will refer to just those ligands examined in the restarted run - see the DOCK manual for details.

Looking at the Results

DOCK puts its output in the directory it was started from, that is, where the INDOCK file is. For SINGLE runs, there is one file of orientations per sphere center; the names of these files are *outfil*+the cluster number. For search runs, there are files containing top-scoring ligands for each type of scoring chosen. Ligands with the highest contact scores are in a file named *outfil*+the cluster number, top electrostatic ligands are in a file named *outfil*+'eel'+the cluster number, and the file of ligands with the best force field scores is called *outfil*+'ff'+the cluster number. The ligand files are in "extended PDB" format, which differs from PDB format in the columns to the right of the coordinates in the ATOM records. Each orientation or ligand in the file has a separate residue number. The scores are given in the REMARK records at the beginning of each residue and are also listed near the end of OUTDOCK.

"Extended PDB" format allows more information to be included in the atom records. Scores for options not used originally may be quickly evaluated for ligand files with this format using SCOREOPT or SCOREOPT2. However, some molecular display programs may not accept this format. If you find that you cannot display your ligands, you can convert them to PDB format using the program *x2pdb* supplied with DOCK. For future runs, create a DOCKOPT file as described in the manual, setting *pdbopt* to 1. Ligands will then be written in standard PDB format with the partial charges in the temperature factor column.

A useful way to view ligands is to display the surface of the protein active site along with a few important residues, then examine ligands one at a time. SHOWESP may be used to visualize the electrostatic potential due to the protein, and SHOWPROBE can display the interaction energy of a probe with the force field grid. SPLITMOL can be used to separate ligand orientations into individual files if it is necessary.

Appendix 5: A Guide to Using DOCK for Beginners at UCSF

The material in this appendix is a supplement to the DOCK Beginner's Guide (Appendix 4). It is intended for the use of members of the Kuntz Group and other groups at UCSF which use DOCK. It is presented in a question-and-answer format intended to fill in details of DOCK use at UCSF which are not appropriate for inclusion in the Beginner's Guide, which is distributed along with DOCK to users at other sites.

What is this document about?

This document is intended primarily for new members of the Kuntz group. It covers the practical aspects of running DOCK and associated programs in the group. Members of other groups at UCSF may also find it useful in learning how to use DOCK, but the details about our computer systems will not apply.

The DOCK Beginner's Guide (included with the DOCK Reference Manual) explains what to do to run DOCK. Because it is intended for a range of users, both at UCSF and elsewhere, it cannot address local details of program usage. This document explains how to carry out the steps described in the Beginner's Guide and where to find the necessary resources. It is structured as a series of questions and answers.

How do I get started?

You should begin by getting two computer accounts, one on the machines maintained by the Computer Graphics Laboratory (CGL) and one on the Kuntz group machines. Apply for a CGL account first by filling out the form available from the CGL administrative assistants in S-1024. You will need to write a brief description of your project and explain how it involves the use of computer graphics. You should also ask the administrative assistants for a copy of the Midas manual. Once your CGL account has been set up, the Kuntz group system administrator will give you an account on the group machines and assign you a home directory. (Group members take turns serving as system administrator, so you will need to ask someone in the group to find out who currently has the job.)

Kuntz group Silicon Graphics workstations are located in S-926, S-955, and HSE-1119. They run IRIX, a variant of unix. These machines are on the desks of current group members, and those people have priority in access to them. You may use them when the individuals assigned to them are away. There are also Silicon Graphics workstations available at the Computer Graphics Laboratory in S-1022 (turn right when you enter, then left). Group Macintosh computers are located in S-926 and HSE-1119.

Documentation for DOCK is available on the World Wide Web at <http://www.cmpharm.ucsf.edu/kuntz/manual/howto.html>. It can be viewed using Netscape or any World Wide Web browser. Since new versions of Netscape are released fairly frequently, check with a group member about which version to use and where it is located. If you would like a hard copy of the manual, contact Paul McCloskey at 6-9031 or mcclosk@picasso.ucsf.edu. Printed documentation for most of the other programs we use is located in S-955, mostly on the high shelves to the right of the windows. Please ask before you take it out of the office and sign it out on the sheet next to the door.

What is Midas, and how do I use it?

Midas is an interactive molecular display program developed in the Computer Graphics Laboratory at UCSF. Group members use it to display proteins, surfaces, and docked molecules. It can be started by typing `midas` on any of the Silicon Graphics machines. One way to learn how to use it is to obtain (or borrow) a copy of the manual and read the first few pages to become familiar with model manipulation and the command syntax. Then display a molecule of interest and skim through the list of commands, trying the ones which look useful. CGL sometimes offers courses in various aspects of Midas as well.

What is the most appropriate way to run large background jobs?

First, make sure that there is space available on the disk where you intend to put the output using the unix `df` command. If your job attempts to write to a disk that has filled, it can cause problems for other users (as well as for you, since you won't have usable results).

Then, find a machine that doesn't already have a job running on it. Check `francisco`, `osric`, and `hamlet` first, then `polonius` and `yorick`. Don't run background jobs on `rosencrantz` or `guildenstern`. Use `top` to check for jobs using large percentages of the available cpu time, and `ps -d` to check for jobs which have large amounts of accumulated time or are running programs which use a great deal of time, such as `dock`.

Do not run background jobs on CGL machines. You may use them to rlogin to Kuntz group machines to set up and start jobs.

Finally, once you have found disk space and an unused machine, use `nice` to start your job at an appropriate priority. To start a program called `foo`, type `/usr/sbin/nice -19 foo &`. Check jobs a few minutes after starting to be sure that they are running; minor errors in setting up a program often show up when it fails in the first few minutes. Continue to check long jobs periodically.

How do I run the programs listed in this document?

To Run	Type
Midas	<code>midas</code>
Netscape	<code>/bert/Netscape-1.1N/netscape</code> or current pathname
dms	<code>/usr/sbin/nice -19 dms protein file name -a -i residue file name -n -o output file name</code>
SPHGEN	<code>/usr/sbin/nice -19 ~dock/bin/sphgen</code>
addprh	<code>~dock/bin/addprh</code>
CHEMGRID	<code>/usr/sbin/nice -19 ~dock/bin/chemgrid</code>
DOCK	<code>/usr/sbin/nice -19 ~dock/bin/dock3.5</code>

sortDOCKout	~dock/bin/sortDOCKout
x2pdb	~dock/bin/x2pdb
ISIS	see the ISIS Guide
SYBYL	trigo sybyl6.2

How can I find a protein model for docking?

If you are docking to a particular target with the intent of testing molecules for biological activity, your collaborators may give you a protein structure. If you do not have a suitable set of coordinates, you may find one in the Brookhaven Protein Data Bank (PDB). PDB structures are available on our machines in `/usr/mol/pdb`; you may also search the PDB via the World Wide Web.

The directory `/usr/mol/pdb/index` includes various lists of PDB id codes and corresponding names or data. Among the most useful are `compound.idx`, which lists id codes and compound names, and `author.idx`, which includes id codes and the names of the authors who published the structure. The structures themselves are in subdirectories of `/usr/mol/pdb`, grouped by the middle two characters of their four-character id code. The filenames are formed by adding 'pdb' before the id code and '.ent' after it. For example, the molecule with id code '2ptc' would be found in the directory `/usr/mol/pdb/pt` in a file called `pdb2ptc.ent`.

As an alternative to looking through the index files in `/usr/mol/pdb`, you may want to search the database using Netscape or another World Wide Web browser. An overview of www search options may be found at http://www.nih.gov/molecular_modeling/database_access.html. *Molecules R US*, at <http://molbio.info.nih.gov/cgi-bin/pdb>, allows you to perform a keyword search of the PDB molecule headers. Clicking on an item in its list of hits retrieves the first 100 lines of the file; if you decide that you would like

the entire file, you can download it by selecting **Text** from the **Output Requested** menu and clicking on **Submit Request**. Use **Save As...** in Netscape to save the file. You may browse a list of PDB molecules sorted by category using *PDB At a Glance*, at http://www.nih.gov/molecular_modeling/pdb_at_a_glance.html. Click on the name of a protein to retrieve it.

There is a brief description of the PDB file format in the Midas manual. A much more thorough description may be found in `/usr/mol/pdb/pub/format.desc.txt`.

How do I create a molecular surface for use with SPHGEN?

Use a version of the protein structure without hydrogens for molecular surface creation. At UCSF, we create surfaces using `dms`, a version of Mike Connolly's `ms` program which is included in the Midas program suite. SPHGEN uses the normals to the points on the surface to place spheres in the receptor site.

If you know which region of the protein you wish to dock to, you should create a surface for only that region. The program `~dock/bin/get_near_res` takes a ligand and a protein structure and writes a file containing only the atoms of the protein within a specified distance of the ligand. If the resulting file is longer than 100 lines, the upper limit for use with `dms`, run `~dock/bin/condense` to shorten it by replacing atoms with whole residues.

Generate the molecular surface by running `dms`, which is documented in Appendix 6 of the Midas manual. Use the command `/usr/sbin/nice -19 dms protein file name -a -i residue file name -n -o output file name`. The `-a` flag causes all atoms, including those not in amino acid residues, to be included. It is necessary if waters, ions, or cofactors

should have surface points. The `-i` flag causes dms to calculate surface only for residues listed in the named file. The `-n` flag triggers the calculation of surface normals, which are required by SPHGEN. The surface produced by dms may be used directly by SPHGEN; no conversion is necessary. It is also in the correct format for viewing along with the protein in Midas.

When and how should I add hydrogens to the protein?

Protein X-ray crystal structures do not include hydrogen atoms, but hydrogens bound to polar atoms must be included for chemical scoring to work properly. You must add hydrogens before running CHEMGRID, but not before running dms. Hydrogens are ignored by DISTMAP.

The simplest way to add hydrogens is to run `~dock/bin/addprh`. This program attempts to add the hydrogens in favorable geometries but does not account for hydrogen bonding. Run it interactively; it will prompt you for input and output file names and your choice of PDB or AMBER hydrogen names. Choose either set of names, but be sure to use the corresponding parameter file for CHEMGRID. If you are comfortable using SYBYL, you may want to sprout hydrogens with it instead.

Can I use DelPhi to make an electrostatic potential map for scoring?

Yes, but the group members who used and understood it have all moved on, so you will need to learn how to use it your own. Your protein must have hydrogens with AMBER names.

The most recent version of DelPhi available here is in `/bert/delphi/export_v3`, and documentation for it is in `/bert/delphi/export_v3/docs`. This version dates back to 1989, so you may want to pursue getting a newer version. Biosym currently distributes the program.

Check the DOCK FAQ at http://www.cmp Pharm.ucsf.edu/kuntz/dockfaq_contents.html for notes about the current status of using DelPhi with DOCK.

Where are the small molecule databases for docking?

The Kuntz group maintains three small molecule databases obtained from MDL Information Systems (MDL, San Leandro, CA) in DOCK 3.5 database format. The structures in the database were built by MDL using CONCORD and assigned charges in SYBYL in the process of conversion to DOCK format. Dockable versions are in /marco/db/db35.95.1 (note that the name of this directory may change slightly with new versions of the database).

The Available Chemicals Directory (ACD), which includes compounds offered for sale, is in /marco/db/db35.95.1/acd. (Many of these compounds have turned out to be unavailable or prohibitively expensive in the past, but at least some of them should be available for testing.) The MDL Drug Data Report database (MDDR), which contains compounds which have appeared in *Drug Data Report* since the late 1980s, is in /marco/db/db35.95.1/mddr. These compounds are the subjects of current or recent investigation by pharmaceutical companies. They are not generally available for testing. The Comprehensive Medicinal Chemistry (CMC) database in /marco/db/db35.95.1/cmc consists of the compounds, mostly known drugs, listed in the multi-volume *Comprehensive Medicinal Chemistry*.

The database molecules are divided into groups by the total *number* of formal charges on each molecule (not the net charge). These groups have been further divided into files containing several thousand molecules each. The number appearing in each database filename indicates the total number of formal charges, and letters are used to distinguish multiple files within

each charge group. Division of the databases into smaller files allows a database to be docked more rapidly by running smaller groups of molecules on several different machines at once.

How can I manage DOCK output? How can I conveniently view the results?

Dan Gschwend's sortDOCKout program (~dock/bin/sortDOCKout) is very useful for selecting the best-scoring ligands (or the ligands of most interest to you) from the output of a large DOCK run. Since Midas cannot display molecules in extended pdb format, convert your ligands to standard pdb format with ~dock/bin/x2pdb before attempting to view them. An entire ligand output file may be opened as a single model in Midas; each ligand is a separate residue.

How can I find the suppliers of my ACD compounds?

Suppliers for compounds in the ACD are listed in the original MDL database from which the DOCK database was produced. The MDL databases are accessed using the program ISIS. For more information, see the ISIS Guide HTML pages (view them with Netscape using **Open File**) at /marco/mdl/guide/isis.html. The page on using DOCK hit lists at /marco/mdl/guide/docklists.html should be particularly useful.

Where can I find the people and resources mentioned in this document?

People

Kuntz group offices	S-955	6-5873
	S-926	6-5326
	HSE-1119	6-3312
Tack's office	S-1025	6-1937
Paul McCloskey mcclosk@picasso.ucsf.edu	U-64B	6-9031
Computer Graphics Lab	S-1024	
Administrative Assistants: Willa Crowell Norma Belfer		

Computers

Silicon Graphics Irises - Kuntz Group	S-955 S-1022 HSE-1119
Silicon Graphics Irises - Computer Graphics Lab	S-1022
Macintoshes - Kuntz Group	S-926 HSE-1119
Greyscale Laser Printers	His S-955 Tyr HSE-1119 Trp S-1022
Color Printer	Ala S-1022

Programs and Documentation

Midas Documentation	Ask the CGL administrative assistants in S-1022; however, they often run out of copies
DOCK documentation, hard copy	Ask Paul McCloskey
DOCK documentation, HTML version for viewing with Netscape	manuals: http://www.cmp Pharm.ucsf.edu/kuntz/manual/howto.html answers to frequently asked questions: http://www.cmp Pharm.ucsf.edu/kuntz/dockfaq_contents.html utility programs organized by function: http://www.cmp Pharm.ucsf.edu/kuntz/manual/accessories/organizer.html
DOCK executables	~dock/bin
Local DOCK-related utilities	~dock/local
ISIS: Guide for UCSF Users, HTML version for viewing with Netscape	/marco/mdl/guide/isis.html
ISIS Programs	Consult the ISIS Guide
Small molecule databases for docking	/marco/db/db35.95.1 (the name may change when new database versions are installed)
SYBYL	run by typing <code>trigo sybyl6.2</code>

Appendix 6: Guide to Using ISIS at UCSF

Databases of small molecules obtained from MDL Information Systems (San Leandro, CA) are useful and convenient targets for docking. These databases are managed using MDL ISIS software. The compound structures are converted to DOCK database format for the actual docking, but users who need further information, such as lists of commercial suppliers for their compounds, must retrieve it from the original databases with ISIS. Because the ISIS software is a relatively complex set of programs, I have written a brief summary of the procedures most often used by members of the group seeking further information.

The ISIS Guide pages which follow were written originally in HTML, the Hypertext Mark-up Language. The HTML pages contain links which, when viewed with a World Wide Web browser, allow the user to jump rapidly to other pages and to related topics. Links are underlined in the ISIS Guide pages.

Guide to Using ISIS in the Kuntz Group, UCSF

ISIS is a chemical database management program produced by MDL Information Systems which we most often use to retrieve information about compounds from the Available Chemicals Directory which have scored well in a DOCK run. This set of documents explains the basics of using ISIS and describes features of the ISIS installation at UCSF which are not covered in the printed documentation. Information is available about these topics:

General information about ISIS

Descriptions of the ISIS programs and the available databases

Setting up and running ISIS

Specific directions for users of the Kuntz Group irises

Using ISIS on the Macintosh

Starting ISIS on the Mac, pasting structures, and printing

Using DOCK hit lists in ISIS

Directions for converting lists of registry numbers from DOCK to ISIS format, viewing structures and supplier data, and printing results

Searching ISIS Databases

Basic descriptions of ID number, substructure, and 3D substructure searching and notes on managing search hit lists

ISIS Exporting

Directions for exporting lists of registry numbers, structures, and data from ISIS to text files

Miscellaneous ISIS Concepts

An explanation of hviews and some information about text fields for programmers

Information for ISIS administrators

MDL Contact Information

Currently, the MDL program administrators are Cindy Corwin (corwin@cgl.ucsf.edu) and Donna Hendrix (hendrix@laertes.ucsf.edu).

Cindy Corwin (corwin@cgl.ucsf.edu)

October 1995

ISIS Program Description

ISIS is a set of programs for the management and searching of databases of chemical structures. ISIS is distributed by MDL Information Systems.

ISIS consists of three programs: ISIS/Draw, ISIS/Base, and ISIS/Host. ISIS/Base is used for retrieving molecules from databases and performing searches. ISIS/Draw is used for drawing structures and structure search queries. ISIS/Host performs database management together with ISIS/Base; users never interact with ISIS/Host directly. ISIS/Draw and ISIS/Base are collectively referred to as ISIS/Desktop.

The Kuntz group maintains ISIS on polonium. In addition, we have installed ISIS/Draw and ISIS/Base on the Macintosh in S-926. An account on polonium is required for access to the databases regardless of which version of ISIS/Base is used.

Available Databases

The Kuntz group currently has three databases of molecules which we have licensed from MDL Information Systems:

The Available Chemicals Directory (ACD3D)

The ACD contains structures and 3D models for compounds which are commercially available. Suppliers may discontinue or run out of a compound before it is removed from the database, so it is unwise to count on buying every molecule retrieved from a search of the ACD. Contact information for suppliers of the ACD compounds is included; pricing is available for some compounds. Users should note that the first 2000 records in the ACD contain information about the suppliers (or are empty); compounds start with record 2001.

MDL Drug Data Report (MDDR3D)

The MDDR includes compounds which have appeared in *Drug Data Report* since the late 1980s. Most of these compounds have been investigated or are currently being investigated as new drugs.

The Comprehensive Medicinal Chemistry Database (CMC3D)

This is an electronic version of the list of biologically interesting compounds included in *Comprehensive Medicinal Chemistry*. It includes many known drugs.

[ISIS Guide Table of Contents](#)

[ISIS setup directions for users of Kuntz group machines](#)

Cindy Corwin (corwin@cgl.ucsf.edu)

October 11, 1995

Setting up the ISIS Environment

Add these 5 lines to your `.login` or `.cshrc`:

```
setenv ISISHOME /marco/mdl/isis12
alias mdlbase $ISISHOME/base/mdlbase
alias mdldraw $ISISHOME/draw/mdldraw
alias plhelp $ISISHOME/pl/plhelp
alias isispl $ISISHOME/pl/isispl
```

You must source `.login` or source `.cshrc` for the changes to take effect in your current session.

Starting ISIS

To run ISIS/Base, enter
`mdlbase &`

To run ISIS/Draw, enter
`mdldraw &`

(Using the `&` allows you to continue using the shell you started ISIS from.)

Using the Databases

To open one of the three databases the Kuntz group has licensed from MDL:

Start ISIS/Base.

Select **Open Database** from the **File** menu.

In the space marked "Selection" at the bottom of the dialog box that appears, enter the location of one of the databases:

```
/marco/mdl/db/acd3d/acd3dfinder.db
/marco/mdl/db/mddr3d951/mddr3d.db
/marco/mdl/db/cmc3d/cmc3d.db
```

ISIS/Base will inform you that this is a read-only database and ask if you want to open it anyway; click **Yes**.

In the Network Connection dialog box which appears, enter your user name and press return. Enter your password in the next dialog box to connect to the database.

Documentation

We have several sets of most of the ISIS Manuals. They are located in S-955 above Dan's desk, to the left of Cindy's desk, and to the right of Diana's desk. If you remove them from S-955, please write your name, location, phone number, and e-mail address on the sign-out sheet to the left of the door.

ISIS Guide Table of Contents

Description of ISIS Programs and Databases

*Cindy Corwin (corwin@cgl.ucsf.edu); much of this material was adapted from a document prepared by Dan Gschwend
October 11, 1995*

Using ISIS on the Kuntz Group Macintosh

Isis/Base and ISIS/Draw are available on the Kuntz group Macintosh in S-926. (That's the Mac located closest to the door.) ISIS/Base may be used to connect to the databases on polonius. An account on polonius is required to use the databases.

Starting ISIS/Base and Connecting to the Databases

To find ISIS/Base, open warhol, then the Programs folder, then the ISIS 1.2.2 folder. Double-click on the ISIS/Base 1.2.2 icon to launch the program.

Choose **Open Database** from the File Menu. Navigate to the "Finder" for the database you want to use:

- The ACD-3D Finder is in the ACD-3D Finder folder, which is inside the ISIS 1.2.2 folder.
- The MDDR-3D Finder is in the MDDR-3D Finder folder, which is inside the ISIS 1.2.2 folder.
- The CMC-3D Finder is in the CMC-3D Finder folder, which is inside the ISIS 1.2.2 folder.

Select the Finder file and click the **Open** button in the dialog box. Type your user name and password for polonius in the dialog boxes that follow.

Note that ISIS program help is available under the apple menu.

Starting ISIS/Draw

To find ISIS/Draw, open warhol, then the Programs folder, then the ISIS 1.2.2 folder. Double-click on the ISIS 1.2.1 icon to start it.

Note that ISIS program help is available under the apple menu.

Pasting Structures from ISIS/Base into Other Applications

Locate the molecule you want to paste and click in the structure box to select it. Then either

Choose **Copy** from the **Edit** menu.

Go to the point in your document where you would like to paste the structure and choose **Paste** from the **Edit** menu. This will paste the structure in the size it appears in ISIS/Base.

OR

Choose **Transfer to ISIS/Draw** from the **Edit** menu or click the notebook-and-pencil icon in the upper left corner of the ISIS/Base Window. This will open ISIS/Draw if it is not already open and place a copy of the structure in the ISIS/Draw window, where you can scale, rotate, or otherwise alter it until it appears the way you want it to.

Select the structure in the ISIS/Draw window.

Choose **Copy** from the **Edit** menu, go to your document, and choose **Paste** from the **Edit** menu.

Printing ISIS Structures

ISIS/Draw 1.2.1 and ISIS/Base 1.2.2 require the use of LaserWriter drivers numbered 8.1 or higher for printing. This is also true for documents from other applications into which ISIS structures have been pasted. Currently, the only printer to which we can print using LaserWriter 8 drivers is tryptophan, which is located in the hallway in the Computer Graphics Lab (S-1022).

To set up for printing, open the Chooser from the apple menu and click on the LaserWriter8 icon. Select zone CGLZone and printer Tryptophan. You can then print by choosing **Print** from the **File** menu.

ISIS Guide Table of Contents

Cindy Corwin (corwin@cgl.ucsf.edu)
November 2, 1995

Using lists of hits from DOCK to find compounds and information in ISIS databases

The ACD, MDDR, and CMC databases which the Kuntz group maintains for use with DOCK were supplied to us originally as ISIS databases by MDL Information Systems. The DOCK databases include registry numbers which may be used in ISIS to look up the compounds in the original databases. The ISIS databases contain information about the compounds which is not present in the DOCK databases. The information available varies among the databases; we use ISIS most often to look up the suppliers and prices of the compounds in the Available Chemicals Directory.

Compounds may be retrieved one at a time by searching for the registry number, but it is generally most convenient to make a file containing a list of compound numbers and import the entire list into ISIS at once.

Creating an ISIS-readable list of database registry numbers

DOCK-format database files maintained for group use include an ISIS external registry number for each compound. The numbers have been modified slightly to fit into the fields available in DOCK3.0 format. To make a list of ACD numbers which is understandable to ISIS, find the Fnnnnnnnn numbers for the compounds of interest in the DOCK output. List them in a file, one to a line. Replace the 'F' at the beginning of each number with 'MFCD'; the resulting format is MFCDnnnnnnnn. For CMC numbers, replace the initial 'C' with 'MCMC'. For MDDR numbers, remove the 'R' and the two leading zeroes, so that you are left with a six-digit integer. In all cases, add *E* on a line by itself at the beginning, followed by a single comment line (which may be blank).

Examples of registry number formats.

Importing a list file into ISIS

Start ISIS/Base and open the appropriate database.

Click on the **Browse** button.

From the **File** menu, choose **Import** and then select **List...** from the cascading menu which appears.

Enter the list file name in the dialog box and click **OK**. (You may need to enter the full path name of the list file if it is not in your home directory.)

Viewing the compounds

The buttons at the left side of the window with the arrows on them may be used to move forward and backward in the list. The button between them with the # on it will display a compound given its position in the list; i.e., enter 1 to view the first item; 10 to view the tenth, and so on.

The set of boxes in which the structure and data are displayed is called a form; click on one of the labeled file tabs to display the associated form.

Viewing ordering information for the compounds

To see a list of suppliers and prices for a given compound, click on the file tab marked "Prices" at the top of the display area. Click on the "Catalog" file tab, then on the name of one of the suppliers, to see ordering information for all forms of the compound supplied by that company. To retrieve contact information for a particular supplier, click on the name of the supplier in the list on either the Prices or Catalog form, then click on the "Address" file tab.

Printing compound structures and data

The Structure and Model forms for an entire list of compounds may be printed in a single operation with multiple forms per page. The Price, Catalog, and Supplier forms may be printed for one compound at a time. ISIS also offers the option of printing a Compound Report, which lists suppliers for the current compound, or a supplier report, which lists contact information for the currently selected supplier together with that supplier's catalog information for the current compound.

Before printing, choose **Print Setup...** from the **File** menu to select the paper size and page orientation. Make sure the form you intend to print is the one showing.

To print, select **Print...** from the **File** menu.

Select **Current Form**, **Current Compound Report**, or **Current Supplier Report**.

To print a form or report for the current compound only, choose "Print Current" and "Actual Size, One per page" and Click OK.

To print forms for a list of compounds, choose "Print List" and "Multiple per page". Enter the number of forms you want to appear in each direction in the "Horizontal" and "Vertical" boxes, then enter the margins and interbox distances. Click OK.

ISIS will create a PostScript file which you can print from the operating system. Enter a name for this file in the "Selection" area of the dialog box which appears.

Once "printing" is complete, use `lpr` to print the file.

Setting Up and Using ISIS

Cindy Corwin (corwin@cgl.ucsf.edu)

October 12, 1995

Sample Lists of External Registry Numbers, as they appear in DOCK output and in ISIS format:

ACD Numbers, DOCK database format:

F00001137
F00004047
F00004229
F00004230
F00004231

ACD Numbers, ISIS list format:

E

MFCD00001137
MFCD00004047
MFCD00004229
MFCD00004230
MFCD00004231

MDDR Numbers, DOCK database format:

R00124775
R00128637
R00131226
R00135331
R00138979

MDDR Numbers, ISIS list format:

E

124775
128637
131226
135331
138979

CMC Numbers, DOCK database format:

C00000007
C00000008
C00000009
C00006511
C00000011

CMC Numbers, ISIS list format:

E

MCMC00000007
MCMC00000008
MCMC00000009
MCMC00006511
MCMC00000011

ISIS Searching

General Notes on ISIS Searching

An ISIS search is usually conducted by placing a query that describes what you are searching for in a box in one of the ISIS forms, then choosing a menu command to conduct the search. You may notice something called the Query Builder; it is used to construct complex queries to retrieve molecules meeting specific combinations of criteria. Unless your searching needs are very involved, you can safely forget about it.

This page explains a bit about finding molecules with particular registry numbers and molecules with specified structural features. If you need to conduct a different type of search, check the section titled "Which Type of Search?" in the *ISIS/Base Database Searching manual* for a reference to the appropriate documentation.

To conduct a search, open the database to be searched and display the form containing the information you are interested in. Click the **Query** button. If you want to search the entire database, make sure that Domain: All is displayed at the upper right.

If you have imported a list of molecules or retrieved a list of molecules in a search, **Domain:Subset** will be displayed at the upper right. By default, the next search will be conducted only over the molecules in the current subset of the database. If you want to search over the entire database, choose **Clear Domain** from the **List** menu. Note that ISIS automatically switches from **Query** to **Browse** when a search is completed, so you will need to click **Query** before conducting a second search.

Retrieving Molecules by Registry Number

Searching for an ID number is a convenient way to get data for a few molecules of interest. For larger numbers of molecules, such as lists of DOCK hits, importing a list of molecules may be easier.

To retrieve compounds from the ACD, you will enter your query in the unmarked box to the left of the word "Name". In the MDDR, the appropriate box is marked "Extreg"; in the CMC, it is marked "MDL Number". Click in the box to select it. For a single ACD registry number, type "=MFCDnnnnnnnnn". (For CMC registry numbers, substitute MCMCnnnnnnnnn. MDDR numbers are six-digit integers; in this case the quotation marks are optional.) Choose **By Form** from the **Search** menu.

Substructure Searching

A substructure search retrieves all molecules which contain the query, that is, all molecules of which the query is a substructure. It is carried out by drawing the query in ISIS/Draw, then transferring it to ISIS/Base before conducting the search. Before you try substructure searching it is a good idea to work through the section of the *ISIS/Draw Tutorial* on drawing molecules.

To conduct a search, start by creating a query in ISIS/Draw. You may wish to look at the section on "Retrieving Molecules by Substructure" in the *ISIS/Base Database Searching* manual, which describes the available query features. In ISIS/Draw, select the query. Choose **Copy** from the **Edit** menu. In ISIS/Base, make sure that the Structure form is displayed and **Query** is selected, then click in the structure box to select it and choose **Paste** from the **Edit** menu. Choose **SSS** from the **Search** menu.

3D Searching

Several types of 3D searches are possible in ISIS; see the section on "Which Type of 3D Search?" in the *ISIS 3D Searching* manual for a list. We most often use 3D substructure searching, which retrieves molecules that both contain specified structural components and meet given geometric constraints (distance between atoms, etc.).

A 3D substructure search is conducted much like a substructure search. The query, including the 3D constraints, is created in ISIS/Draw. The constraints are added by selecting the atoms or objects to which they will apply, then choosing **3D** from the **Chem** menu and **Create...** from the pop-up menu. In the "Create 3D Object" dialog box, select the type of constraint and enter the value(s). The available 3D objects and constraints are listed in the *ISIS 3D Searching* manual. To transfer the query from ISIS/Draw to ISIS/Base, select it and choose **Copy** from the **Edit** menu. In ISIS/Base, make sure that the Model form is displayed and **Query** is selected, then click in the model box to select it and choose **Paste** from the **Edit** menu. Choose **SSS** from the **Search** menu.

ISIS also offers conformationally flexible searching, which is documented in *ISIS 3D Searching New Features (1.2)*. If you find that conformationally flexible searching does not work, it may not have been properly configured; consult the ISIS program administrator.

Managing Search Results

Chapter 9 of *ISIS/Base Database Searching* explains how to manage lists of records retrieved by searches. Since all our databases are read only, lists and

records may be saved in the database only for the current session. However, saving lists is still useful because it allows you to combine them with the results of other searches using logical operations. To keep a list after you leave ISIS, use the **Export List...** option under the **File** menu to write it to a file. (Get it back later using **Import List...**). Lists of molecules may be printed using the method described on the [Using DOCK Lists Page](#).

[ISIS Guide Table of Contents](#)
[Setting Up and Using ISIS](#)
[Exporting Lists, Structures and Data](#)

Cindy Corwin (corwin@cgl.ucsf.edu)
October 16, 1995

Exporting Lists of Molecule IDS

To write the ID numbers of the current list of molecules to a file, choose **Export** from the **File** menu, then select **List...** from the cascading menu. Enter a name for the output file in the dialog box which appears and click OK. (Unless you specify a full pathname, the file will be placed in your home directory.)

Lists exported in this way contain external registry numbers (for the ACD, the MFCDnnnnnnnn numbers; for the CMC, MCMCnnnnnnnn numbers; for the MDDR, six-digit numbers). The first line of the file begins with *E* and includes information about how and when the file was written; the second line should be blank.

Exporting Structures and Data

Exporting SDFiles and RDFiles from the ISIS Menus

The most direct way to write a file containing ISIS data for use by another program is often to export it as an SDFile or an RDFile. These files have defined formats, so you will probably need to write a program or script to extract the information you need. However, this is usually easier than attempting to use ISIS to write out exactly what you want.

SDFiles and RDFiles have two major differences: (1) Before MDL developed ISIS, they maintained two separate products. MACCS was used for molecule databases and used SDFiles; REACCS was used for reaction databases and used RDFiles. The files have different formats. (2) Not all database fields can be exported from the ISIS menus using SDFiles. RDFiles allow export of more fields.

To export one of these files for the current list, choose **Export** from the **File** menu and select **SDFile...** or **RDFile...** from the cascading menu. The "Export Fields" dialog box will appear. Select the name of one of the fields you want to write out and click Add. (The field names are presented as a hierarchy. The bracketed, capitalized names identify levels of the hierarchy, while the names in small letters identify actual fields. Indentation indicates that some levels are below others. Select from the names in small letters; they will appear in the right-hand box along with the identifiers for their levels.) Repeat until you have selected all the fields you want, then click OK and wait. If your list is large or you are exporting structures, it can be a very long wait (up to several days to export structures from an entire database).

Exporting SDFiles of 3D Structures using the exportSDF ISIS/PL Script

To export molecule names, MDL numbers, and 3D model coordinates in the SDF file format used in creating DOCK databases, follow Dan Gschwend's directions:

If this is your first time through this process, begin at step 0. If you have been through these steps before, jump to step 2.

0. Do NOT load ISIS/Base yet.

1. Create an ISIS/PL ("Programming Language") program that will load the PL menu in ISIS/Base.

a. In the directory you will start ISIS/Base from, enter the text between the lines into a file called "autobase.pl":

```
program AutoBase;
begin
  ActivatePLMenu;
end.
```

b. Compile this program:

```
cp /marco/mdl/isis12/pl/isisbase.inc .
cp /marco/mdl/isis12/pl/isispl.int .
isispl autobase
```

This step generates "autobase.epl", which is read automatically by ISIS/Base on startup.

2. Load ISIS/Base in a directory containing the autobase.epl file. Open your favorite database containing 3D structures (e.g. cmc, mddr, acd). Select compounds you are interested in with any appropriate query (see the searching page).

3. Load the exportSDF PL program: Select the menu item **File:PL:Execute Program** and enter /marco/mdl/isis12/pl/local/exportSDF.epl when prompted. If you do not have a PL menu item under the File menu, you need to go back to step 0.

4. Run the program to export the active List of compounds. Select the menu item **File:Export:Special SDF file** and enter a filename when prompted. Note: there is a bug in ISIS such that any pathname you enter here will be ignored - the filename will be placed in the current directory, regardless of where you tell it to put it! BTW, the exporting process is VERY slow...

To proceed with the conversion to DOCK databases, consult the documentation under sdf2mol2 in the DOCK 3.5 manual. You will need to run sdf2mol2, then sybdb, then mol2db.

Exporting Other Data Using ISIS/PL

If you have looked at the code for exportSDF.pl, you may be thinking that it shouldn't be too difficult to modify it to export your choice of fields, in your choice of format. In fact, this is possible, but it is not as straightforward as it looks. Before you try it, check out the [Miscellaneous ISIS Concepts Page](#).

[ISIS Guide Table of Contents](#)

[ISIS Administration Table of Contents](#)

Cindy Corwin (corwin@cgl.ucsf.edu)

October 17, 1995

Miscellaneous Advanced ISIS Concepts

Hviews

An hview is a database description which allows ISIS to treat a database of any type (or a combination of databases) as if it had a hierarchical structure. ISIS can use reaction databases, which are hierarchical, and relational databases, but all the databases we license from MDL are molecule (or MACCS) databases. Conceptually, these are flat files. Hviews supplied by MDL define a hierarchical structure for them, and it is possible to change the apparent hierarchy by modifying the hview.

You may view the hierarchical structure of a database within ISIS by choosing **Definition** from the **Database** menu.

The public hviews which are used by default when a database is opened reside in `/marco/mdl/ih131/hviews`. Users may also have their own hviews in their home directories, which will be used in preference to the public hviews.

Cursors

ISIS/PL applications keep track of which level of the hierarchy they are currently working at by using a cursor. To access fields at a different level of the database, it is necessary to move the cursor. This is the reason for the `SetCursorContext` commands in `exportSDF.pl`.

Flattening Fields

ISIS treats flexible fields (data fields which do not have a fixed length) as sets of many fields of one line each. For example, the `generic.name` and `chem.name` fields in the MDDR are actually multiple fields located one level below the other molecule fields in the hierarchy; the chemical name appears as

```
<CHEM.NAME>  
chem.name
```

in the Database Definition dialog box. The section called "Viewing Flexible Fields" in Chapter 5 of the *ISIS/Host Hview Developer's Guide* describes this in more detail.

It is possible to set the cursor context to the level of the desired set of fields and access all the data by retrieving the lines one at a time. However, if you

do not want to search it line by line, you can cause ISIS to treat a flexible field as a single long field by adding a line like

```
tinfo fieldtype chem.name chem.name flatten text
```

at the end of the molecule tree information. Comment out any lines which rename the field. For an example, see the modified hview created by Gary Marshall of MDL Technical Support.

You may use this hview to open the databases as you normally would, but the flattened fields may not be displayed properly in the forms.

Documentation for ISIS/PL Programmers

The *ISIS/Host Hview Developer's Guide* describes the contents of hviews. Note that the contents vary according to the type of database used.

There are three manuals for ISIS/PL itself. The *ISIS Procedural Language Introduction* describes the basic constructs and how to write and compile a program. The *ISIS Procedural Language Users' Guide* and *ISIS Procedural Language Reference* list and describe the functions available for retrieving and manipulating data.

We have only one set of these manuals; they are located over Dan's desk in S-955. Please do not remove them without permission, and be sure to sign them out.

ISIS Guide Table of Contents

ISIS Administration Table of Contents

Cindy Corwin (corwin@cgl.ucsf.edu)

October 17, 1995

```

----- modified MDDRCFS.HVD:
hview mddr3d

comment This is the Hview for the MDDR-3D 95.1 database with
MODEL over MOL.
comment 3/24/95

tree mol
  device maccsdb
  database mddr3d:
  tname mol
  password

  rename mol>(molclass) to
  rename mol>(molskeys) to
comment I commented out the following two lines:
comment  rename mol>chem.name>(chem.name_text) to (chem.name)
comment  rename mol>generic.name>(generic.name_text) to
(generic.name)
  rename mol>company.code>(company.code_text) to
(company.code)
  rename mol>trademark>(trademark_text) to (trademark)
  rename mol>cas>(cas_ftext) to (cas)
  rename mol>cas>(cas_text) to
tinfo fieldtype pref.number pref.number flatten ftext
  rename mol>rel.code>(rel.code_ftext) to (rel.code)
  rename mol>rel.code>(rel.code_text) to
tinfo fieldtype phase phase flatten text
  rename mol>source>(source_text) to (source)
  rename mol>license.info>(license.info_text) to
(license.info)
  tinfo fieldtype phys.properties phys.properties flatten
text
  rename mol>comments>(comments_text) to (comments)
  tinfo fieldtype act.investigation act.investigation flatten
ftext
  tinfo fieldtype activ_index activity text 1 5
  tinfo fieldtype activ_class activity text 8 200
  tinfo fieldtype action action flatten text
  rename mol>prous.ref>(prous.ref_ftext) to (prous.ref)
  rename mol>prous.ref>(prous.ref_text) to
  rename mol>lit.title>(lit.title_text) to (lit.title)
  rename mol>lit.type>(lit.type_text) to (lit.type)
  rename mol>lit.ref>(lit.ref_text) to (lit.ref)
  rename mol>pat.source>(pat.source_text) to (pat.source)
  rename mol>pat.title>(pat.title_text) to (pat.title)
  rename mol>pat.invent>(pat.invent_text) to (pat.invent)
  rename mol>pat.priority>(pat.priority_text) to
(pat.priority)
  rename mol>pat.number>(pat.number_text) to (pat.number)
  rename mol>pat.type>(pat.type_text) to (pat.type)
  tinfo fieldtype preview preview flatten ftext

comment I added these lines to flatten these fields:

```

```
tinfo fieldtype generic.name generic.name flatten text
tinfo fieldtype chem.name chem.name flatten text

tinfo key molregno

tree model basedon mol
  tname model
  password

  rename model>(molregno) to (2d_regno)
  tinfo fieldtype source model.source flatten text
  tinfo fieldtype ccratio model.ccratio flatten num1
  tinfo fieldtype warning model.warning flatten text

  tinfo key modelregno

link model model>(molregno) over mol (molregno)
```

Available Advice on ISIS Program Installation and Administration

- SGI Installation Hints
- Macintosh Installation Hints
- MDL Contact Information
- ISIS Exporting, including exporting structures for conversion to DOCK databases
- Miscellaneous Concepts – hviews and flexible fields

Cindy Corwin (corwin@cgl.ucsf.edu)

Critical Things to Do When Installing ISIS on the Silicon Graphics Machines

Installation of ISIS/Host, ISIS/Base and ISIS/Draw are documented in the ISIS/Host and ISIS/Desktop manuals in the green binder. However, important details are listed in many different places and it can be tough to keep them all straight ... hence this document.

These configuration steps must be performed before ISIS will function properly and find all the databases. They are all documented, but they've been collected here because if only part of ISIS is being installed, it is not always obvious which of these things need to be changed:

1. These files must be in /etc and must correspond to the most recent version of ISIS/Host:

- isisd.13
- rc.isishost.131
- mdlauditsrvr.131

Version number extensions will probably change. If the install script for ISIS/Host is run as root, these will automatically be placed in /etc.

Documentation: *ISIS/Host Installation and Administration*, Chapter 4

2. A port must be set up for the ISIS/Host daemon and the internet daemon must be restarted in order for the ISIS/Host daemon to start. This operation requires root privileges. Files affected are /etc/services and /etc/inetd.conf. Note that we maintain the log file in the ISIS/Host directory.

Documentation: *ISIS/Host Installation and Administration*, Chapter 6.
ISIS/Host documentation changes for version 1.3.1.

3. Copy the database hviews from the database directories to /marco/mdl/ih131/hviews.

Documentation: Database installation instructions.

4. Define the database environment variables in /marco/mdl/ih131/bin/mdlnames and /marco/mdl/ih131/bin/mdlnames.csh to be the path names in our installation.

Documentation: Database installation instructions.

5. Rename the "finder" files for the databases (acd3dfinder.unixdb to acd3dfinder.db, etc.).

Documentation: Database installation instructions.

6. Configure the database remote-access files. Include this information:

- Service Name: isishost
- Node Name: polonius.ucsf.edu
- Network: TCP/IP
- In the TCP/IP Config... dialog box, enter
 - o Internet Address: polonius.ucsf.edu
 - o Port: isishost (may need to pick from list)
 - o Agent: Conduit

Documentation: Database installation instructions.

7. Copy the conformationally flexible search files from /marco/mdl/ih131/cfsdesktop to /marco/mdl/isis12/base/basestart. This only affects users doing conformationally flexible searching.

Documentation: ISIS/Host Installation and Administration, Chapter 4.

8. In the default hvIEWS directory (/marco/mdl/ih131/hvIEWS), make these modifications:

- In mddr3d.hvd, replace tinfo key molregno with tinfo key extreg.
- In cmc3d.hvd, replace tinfo key molregno with tinfo key mdlnumber.

These changes cause ISIS to import and export lists of external registry numbers from these databases (otherwise, **Import List...** will refuse to read lists of numbers taken from the DOCK-format databases).

Cindy Corwin (corwin@cgl.ucsf.edu)
October 17, 1995

General Directions for Installing ISIS/Draw and ISIS/Base on the Kuntz group Macintosh

Most of the instructions necessary for the installation are in "ISIS/Draw and ISIS/Base Installation & New Features," which probably came with the software. Here is an outline of the options used for installation on our machine:

- Allow the installer to place ISIS/Draw at the top level (e.g. on "warhol") and perform a complete installation.
- Choose to overwrite the existing 'tpl.cfg' file.
- Install ISIS/Base at the top level; it will automatically be placed in the same folder as ISIS/Draw.
- Once both programs are installed, move the newly created ISIS n.n folder into the "Programs" folder.
- Open the old ISIS folder(s) and move the folders containing the database "Finders" into the new folder. Move the "hosts" and "services" files to the new folder as well.
- Start ISIS/Base and configure the location of ISIS/Draw as described in the documentation.
- Verify that all the databases are reachable, help (under the apple menu) is functioning, and the ISIS/Draw templates are accessible. Check that transferring structures from ISIS/Draw to ISIS/Base works.
- Trash the old files.

Configuring ISIS/Base

Depending on the modifications made by MDL, it may not be necessary to reconfigure ISIS/Base with each upgrade. Configuration is documented in the *ISIS/Host Installation and Administration* manual, in the section of Chapter 12 entitled "Configuring TCP/IP Software on Workstations." (Hint: the workstation referred to in the documentation is the Macintosh.)

If a new ISIS/Base installation fails to connect to the databases, check these things:

- "hosts" and "services" files must exist on the Mac. Use SimpleText or another word processor to create them, using the same format as you would on the SGIs, and save them in text format in the folder with the applications.
- In addition to MacTCP, ISIS requires another communications tool to connect to the outside world. This may be either the VersaTerm telnet tool or the TCPack Connection Tool; we are currently using the VersaTerm telnet tool. If you can't establish a connection, contact MDL technical support to find out if ISIS requires a later version than the

one we have.

Database Finders

MDL supplies separate applications called "Finders" for viewing each of the databases they supply us with. On the SGI, these applications are supplied on the CDs with the databases, but versions for the Mac are sent separately (and are not necessarily sent with each database release). They are installed by creating a folder for them within the main ISIS folder and copying the relevant files from the floppy disk into this folder.

Before each finder is used, it must be configured using the **Configure Database...** option under the **File** menu, following the directions for TCP/IP connection. The "hosts" and "services" files must be present for configuration.

Cindy Corwin (corwin@cgl.ucsf.edu)
October 9, 1995

Contacting MDL Information Systems

MDL Information Systems, Inc. supplies the ISIS software and databases.

MDL Headquarters

MDL Information Systems, Inc.
14600 Catalina Street
San Leandro, CA 94577

(510) 895-1313
fax (510) 352-2870

MDL Customer Support

(800) 362-3002
(510) 895-2213
fax (510) 895-6092 or (510)895-5968

e-mail techsupp@mdli.com

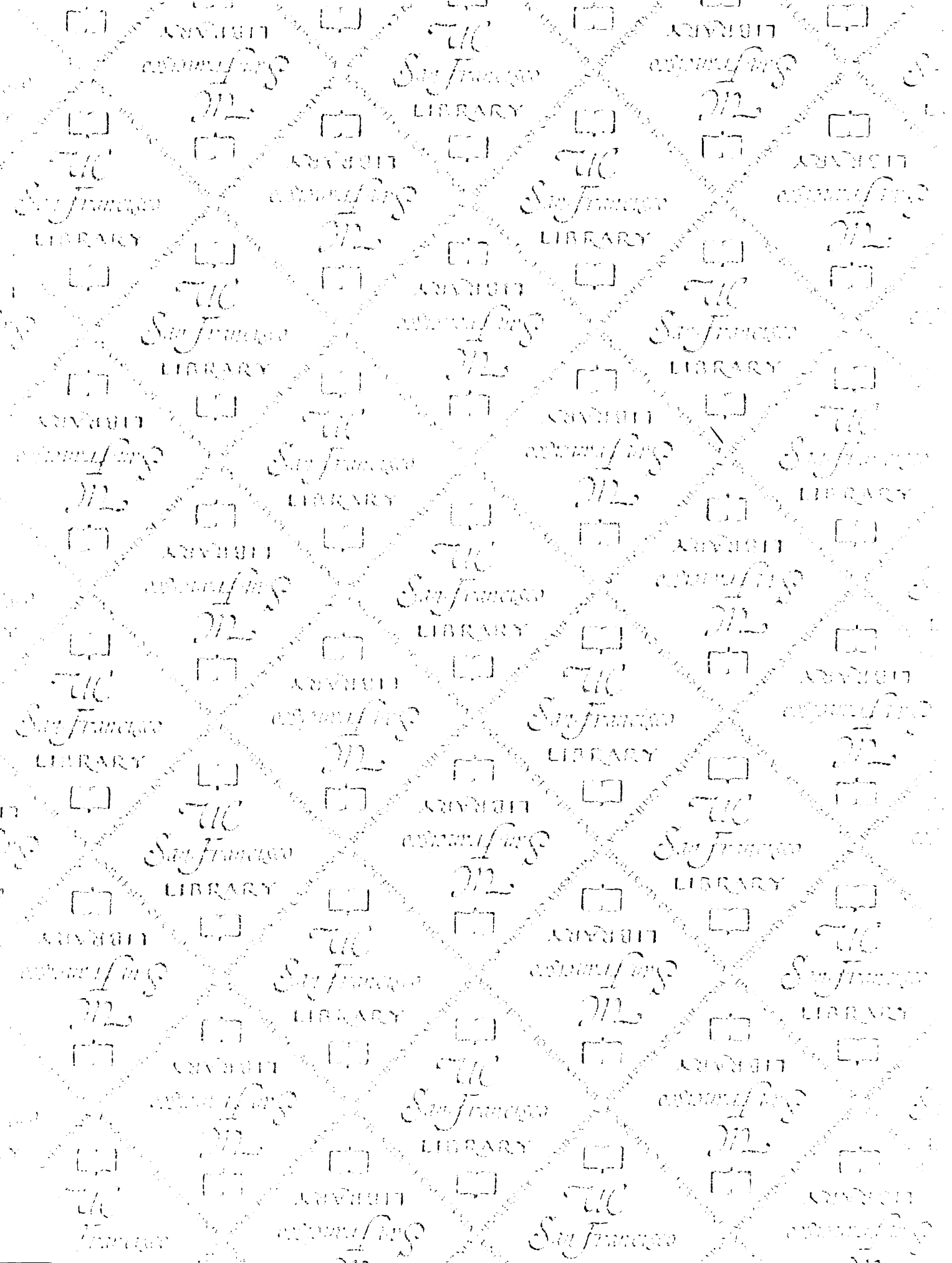
E-mail response has generally taken 3-5 days, although you should get a note within a day stating that your request has been forwarded to the appropriate person. Gary Marshall, the ISIS/Host specialist, has been quite helpful.

MDL License Issues

Osman Güner (osman@mdli.com) is our contact person for matters concerning our ISIS license. He may also be reached by mail, phone, or fax at MDL Headquarters.

Cindy Corwin (corwin@cgl.ucsf.edu)

October 12, 1995



For reference

Not to be taken from the room.

6462652



3 1378 00646 2652



