SPROUT: Recent Developments in the de Novo Design of Molecules

Valerie J. Gillet, William Newell, Paulina Mata,† Glenn Myatt, Sandor Sike,‡ Zsolt Zsoldos, and A. Peter Johnson*

School of Chemistry, University of Leeds, Leeds LS2 9JT, U.K.

Received August 10, 1993®

SPROUT is a computer program for constrained structure generation. It is designed to generate molecules for a range of applications in molecular recognition. The program uses a number of approximations that enable a wide variety of diverse structures to be generated. Practical use of the program is demonstrated in two examples. The first demonstrates the ability of the program to generate candidate inhibitors for a receptor site of known 3D structure, specifically the GDP binding site of p21. In the second example, structures are generated to fit a pharmacophore hypothesis that models morphine agonists.

INTRODUCTION

An outline of the components of the SPROUT program for constrained structure generation was described recently.[1,2] The structure generation component is divided into two parts: the first is to generate skeletons or molecular graphs that satisfy steric constraints; the second is to convert the skeletons into molecules by making atom substitutions. Skeleton generation was described in some detail. In this paper we identify the components that are necessary for any program that attempts *de novo* structure design and review the various programs that have been described. SPROUT is then described in the context of these components. The derivation of the constraints and the skeleton generation phase are reviewed. Several more recent developments are described, including the implementation of a number of user defined parameters, the atom substitution phase, and methods for structure evaluation and organization. SPROUT is then applied to two new examples.

A number of computer programs are currently under development for the design of structurally novel compounds. An earlier generation of programs aimed at identifying potentially bioactive compounds is based on the use of techniques that were developed for searching databases of 3D structures. A number of programs have been developed using these techniques including ALADDIN,[3] CAVEAT,[4] and DOCK.[5] However, these programs are limited in their ability to design novel structures. Recently, advances in computer technology have made the *de novo* design of molecules feasible, and a number of such programs have been described. These programs are based on the concepts of molecular recognition where molecules interact because they are complementary to one another. Thus the properties of one molecule can be used as constraints for the design of other molecules with which it can interact. Any program that attempts to design structures from first principles must consider a number of aspects:

(1) Definition of the Constraints. The existence of three-dimensional atomic coordinates of proteins obtained from X-ray crystallography provides one starting point for *de novo* design, e.g., for the design of ligands that will bind to an active site. However, even though the number of such structures is increasing rapidly and the quality of the structures is improving, they still represent a small fraction of known proteins. In the majority of cases crystal structures are not available. Therefore it is also important to be able to design structures to fit less well defined constraints such as a pharmacophore hypothesis.

(2) Structure Generation. From the constraints it is possible to identify some interaction sites, i.e., regions where it is desirable to place ligand atoms. In some cases it is also possible to define a volume that the structures must be contained within. A method is required for generating structures that satisfy these constraints, i.e., structures that have functional groups at the interaction sites and that fit inside the specified volume.

(3) Structure Evaluation. It is important to ensure that the structures generated meet a number of other criteria, e.g., a potential enzyme inhibitor must be able to bind to the active site, must be synthetically accessible, and must meet a number of other objectives such as having the required transport properties. Some of the systems described do attempt to model some of these criteria during structure generation, but, arguably these criteria can be satisfied more easily after the generation of a set of structures, either by modifying the structures or by eliminating those with undesirable properties.

(4) Organization of the Results. Programs for structure generation can typically produce a large number of answers, and the user must be provided with tools for navigating through large answer sets. These tools could include clustering and ranking techniques and a variety of different criteria could be used, e.g., ranking the structures according to their binding energy and/or their ease of synthesis.

CHARACTERIZING RECEPTOR SITES

Several programs have been developed that can be used to characterize an enzyme receptor site in terms of potential interaction sites. These include GRID, HSITE, LUDI, and MCSS. GRID[6-8] uses an energy function and a variety of different molecular probes to determine favorable ligand binding sites for known structures. HSITE[9,10] was developed as a component of an automated ligand design program[11] and uses statistics to estimate the probability of hydrogen bond formation at different points in an enzyme active site. The LUDI program[12,13] uses a rule-based approach that is based on a statistical analysis of the nonbonded contacts found in the Cambridge Structural Database.[14] It is able to identify hydrogen bonding sites and hydrophobic pockets. MCSS[15] positions and orientates functional groups within an active site using a combination of random placement and energy

† Present address: Departamento de Quimica, Fac Ciencias e Tecnologia, UNL, Monte da Caparica, Portugal.

‡ Present address: Eötvos Loránd University, Általános Számítástudományi Tanszék, Budapest, Bogdánfy u. 10/b, Hungary.

® Abstract published in *Advance ACS Abstracts,* January 15, 1994.

0095-2338/94/1634-0207$04.50/0

minimization techniques. All of these techniques can be useful in specifying the initial constraints for structure generation.

STRUCTURE GENERATION

The programs that have been described for structure generation can be categorized as follows: (1) programs that use predefined linker groups to connect fragments that have been positioned at interaction sites; (2) programs that build up structures in a stepwise manner; (3) programs that are based on more random methods.

The first category of programs includes LUDI and HOOK. In LUDI[12,13] the identification of interaction sites has been combined with a method of generating structures. This essentially involves a look-up in a database of fragments. This is often carried out interactively with the user selecting fragments to act as seed points for attaching additional groups. HOOK[16] starts with the output from an MCSS run and attempts to link the interaction sites by searching for appropriate fragments in 3D databases.

The second category includes a number of programs that build structures in a stepwise manner, e.g., LEGEND, GenStar, GroupBuild, and GROW. LEGEND[17] builds structures one atom and bond at a time. Exploring structure space exhaustively by this method is prohibited due to the enormous combinatorial explosion that would result. Therefore, methods have been devised to limit the structures that can be generated. In LEGEND this is done by using random numbers to guide the program at all stages, e.g., choosing the seed point, choosing the new atom to add and the way in which it is joined to the seed. GenStar[18] is similar to LEGEND in that atoms are added sequentially, but it differs in the ways in which new atoms can be joined and hence in the range of structures that can be generated. The methods used in GenStar have recently been extended in the program GroupBuild[19] that uses larger molecular fragments as building blocks. GROW[20] also uses molecular fragments as building blocks. It was originally developed to generate peptides, so the building blocks were restricted to the amino acids. However it is still necessary to include a large number of fragments to cover the conformational space available to these flexible units. A scoring function is used to select the most promising partial structures for expansion. Recently the method has been extended to include other non-peptide fragments, and these can be used to bridge the gap between two amino acid units.[21] All of these programs use atoms or fragments as building blocks where the element types are distinguished.

The last category includes a number of programs. Weininger et al.[22] have developed a program based on a genetic algorithm. The structures are represented in 2D by SMILES strings.[23] These map directly to the bit string representation required by genetic algorithms. Operations of mutation, crossover, and reproduction can be performed to produce new SMILES strings; the new strings are scored and either saved in the new generation or rejected. The scoring function measures how well the structures fit the active site by generating 3D conformations from the 2D representation and docking them into the receptor site. The docking is done using a distance geometry method combined with DOCK.[5] The Chemical GENESIS project[24] is also based on a genetic algorithm. In this case 3D information is encoded in the bit strings so that they represent 3D conformations directly. Similar operations are performed on the bit strings to produce new structures. The ligand perturbation space algorithm described by Miranker[25] is an interesting approach to the problem of *de novo* structure design. The method is based on filling an enzyme cavity with atoms and then allowing perturbations in their positions to take place that can result in the formation and breaking of bonds. BUILDER[26] begins by using DOCK to search a database for structures that have complementary shape to an active site. The structures retrieved are then superimposed onto an irregular lattice and novel structures are generated by tracing paths through the lattice.

It is well recognized that structure generation is a very computationally intensive task; it represents a combinatorial problem where attempts at finding solutions lead quickly to a large number of possibilities. All of the above programs use fully described atoms or fragments, and an enormous number of fragments would be required to ensure full coverage of structure space. This is computationally infeasible and so different methods are used to restrict the problem, e.g., using random numbers at decision points, allowing the user to guide the program interactively, restricting the application area to a subset of problems, and developing scoring functions to prune some of the possibilities. Any of the methods that are based on random numbers do not permit a complete search of the problem space and have the disadvantage that different answers are likely to be produced in repeated runs of the program on the same problem.

SPROUT

SPROUT[1,2] falls into the second category of programs in that it builds structures in a stepwise manner; however, there are important differences from the programs described above. SPROUT uses a number of approximations that are described in the next section, to attempt to reduce the processing time to a manageable size. These approximations enable structure space to be searched exhaustively for a given set of constraints and user defined parameters and so enable a large number of diverse structures to be generated. Additionally, since it is not a random process, the same structures are generated in repeated runs. This provides a level of understanding of how the structures were generated that is not available when random methods are used.

The basic methodology used in SPROUT has been described previously.[1] It is summarized here together with a description of some new developments. The main components are as identified in the Introduction with one significant difference. In SPROUT structure generation has been divided into two phases: (i) the generation of skeletons that satisfy the steric and geometric constraints (a skeleton is defined as a molecular graph whose vertices are labeled by hybridization state and whose edges are labeled by bond type), and (ii) the substitution of atoms in the skeletons to produce molecules that have the required properties, e.g., electrostatic and hydrophobic properties. An outline of the overall structure of SPROUT is given in Figure 1.

METHODS

1. Deriving the Constraints. Molecules are able to recognize one another because they have complementary steric and electrostatic properties. In ligand design mode SPROUT uses the properties of one molecule to generate potential molecules with which it can interact. Some of the electrostatic interactions are well defined with respect to the relative positions of the atoms, e.g., hydrogen bonding interactions. These localized interaction sites are used to define *target sites* for structure generation. Target sites represent small regions of space where it is desirable to place an atom of the potential ligand so that the required interaction is possible. The

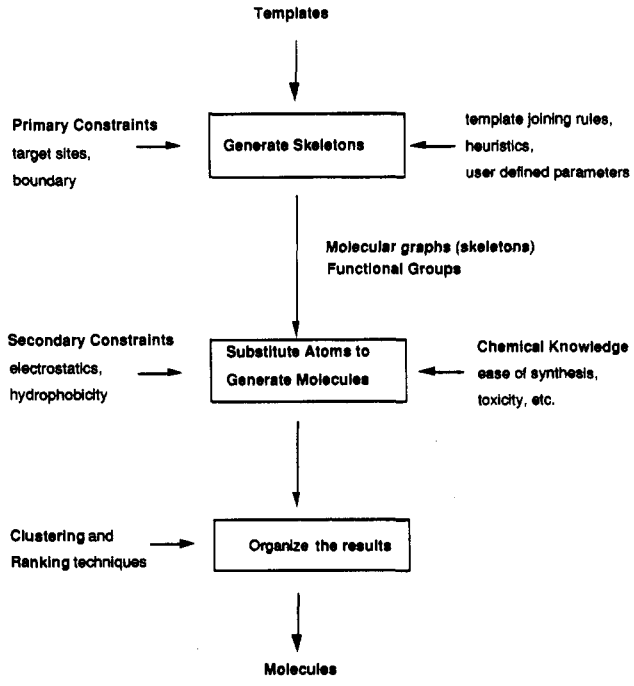


Figure 1. Overview of the components of SPROUT.

constraints for structure generation can also include a volume that restricts the shape of the structures. For example, the 3D shape of the enzyme receptor site can define the volume the inhibitor, or ligand, must lie within.

Target sites can be derived in a number of ways. When generating structures as potential inhibitors of a known active site, target sites can be derived by (1) using the atom coordinates of known ligands when an X-ray structure of a enzyme–ligand complex is available, or alternatively using the coordinates of water molecules in the active site; (2) manually positioning small fragments in the active site and using energy minimization techniques to optimize their position; (3) using a program to automatically analyze an active site for points of interaction, e.g., GRID, MCSS, LUDI, or HSITE. When no 3D coordinates of the receptor are available, targets sites can be derived from the atom positions of a known ligand or by a pharmacophore hypothesis generator such as CATALYST.[27]

When the 3D structure of an active site is known, the volume for structure generation is derived from the solvent accessible surface of the active site. The active site can be a closed binding site, e.g., the acetylcholine binding site of acetylcholinesterase, or it can be a partially enclosed site with one surface open to the solvent, e.g., the guanine diphosphate binding site of p21. In the latter case the volume is defined by "capping" the active site. A volume derived from a receptor site is not essential for structure generation, but, as shown later, it can increase the efficiency of SPROUT by providing a mechanism for pruning the search graph. When no 3D coordinates are available, a volume is defined by placing a parallelepiped around the target sites. The size of the parallelepiped is configurable by the user. It is also possible to define a volume by superimposing sets of compounds to produce a molecular surface.

The volume is represented internally on a 3D grid with a resolution of 0.2 Å. Elements of the grid are switched on if they fall within the active site and off if they represent forbidden regions for structure generation. The boundary is "softened" by including a margin around it. This allows for some flexibility in the positioning of the growing structures. The target sites are represented by spheres with a radius of 0.5 Å, and a vertex is said to cover or satisfy a target site if it falls anywhere within the sphere.

2. Structure Generation. 2.1. Skeleton Generation. Skeletons are built in a stepwise fashion using small three-dimensional fragments, called *templates*. Each template represents a single conformation of a substructure as a molecular graph. The vertices of a template are distinguished by hybridization state only and not by element type; thus each template may represent several substructures. The program currently includes acyclic templates up to four vertices in size, a range of single 3- to 7-membered rings in different conformations, and a small number of templates that represent bridged ring systems. Templates can be joined by fusion, by spiro joining, or by forming a new bond between two vertices belonging to different templates. In the latter case a discrete set of conformations is chosen about each rotatable bond that is formed. The construction of skeletons is governed by a set of template joining rules. These are designed to increase the efficiency of skeleton generation, by preventing the building of identical skeletons using different templates, and also by preventing the joining of unlikely combinations of templates. The template library is designed in a flexible way so that new templates can be added into the library and also so that the set of templates available for a run can be restricted to a subset of the whole library.

Skeleton generation begins by selecting a template from the library and positioning it at one of the target sites thus satisfying one of the requirements. One vertex of the template is anchored at the center of the target site, but the other vertices can be rotated about the anchoring vertex. A representative set of orientations is chosen by the program, and each orientation gives rise to a partial skeleton. New templates are added to build skeletons of increasing size. A solution is found when all the steric and geometric requirements are satisfied and no boundary violations have occurred. This process is illustrated in Figure 2.

The problem space for structure generation is represented by a search graph, Figure 3. The root of the graph represents the initial state of the problem. This corresponds to the primary or steric constraints. The goal nodes of the graph represent solutions, i.e., skeletons that satisfy the steric constraints. The intermediate nodes represent partial solutions or partial skeletons. The expansion of a node in the graph is achieved by adding each template to the skeleton in every possible way. The search graph can be very large, and hence an efficient control strategy is required for searching the graph. The search is directed by associating a cost or score with each of the nodes in the graph; i.e., it is an A algorithm.[28] The A algorithm is a best first method that uses knowledge about the problem domain in the form of heuristics to decide which node in the graph to expand next. If the aim of a search is to generate the full range of structures that satisfy the constraints, then the graph should be fully expanded. In this case the A algorithm is not used to prune the graph but to order the nodes for expansion, and hence it has an effect on the order in which the solutions appear.

2.2. User-Definable Parameters. Structure generation can be guided by a number of user-definable parameters. These are entered to the program via a graphical user interface. Some of the parameters that are available are described below.

Resolution. When the first template is positioned at the first target site, one of its vertices is anchored at the center of the target site. A discrete set of orientations is then generated. This is done by aligning a bond from the anchoring

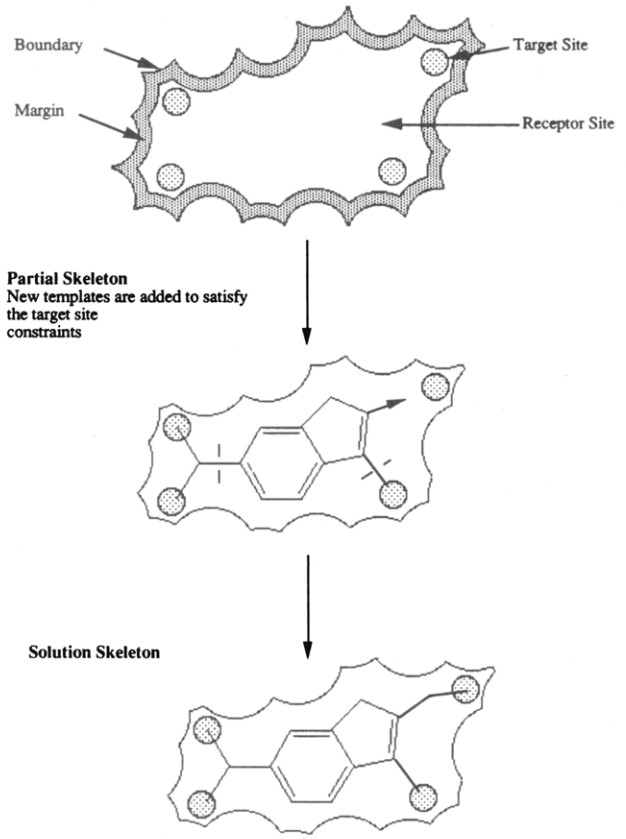


Figure 2. Structures generated from small molecular fragments called templates. A solution has been found when all the target sites are satisfied and no boundary violations have occurred.

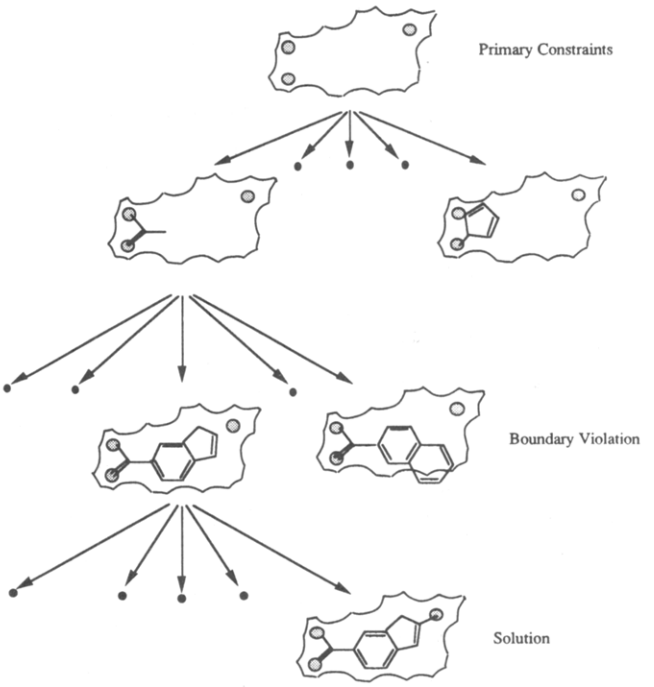


Figure 3. Search graph for skeleton generation. The root of the graph represents the initial constraints; in this case these are target sites together with a volume derived from a fully enclosed receptor site. The intermediate nodes represent partial skeletons, and the goal nodes represent solutions, i.e., skeletons that satisfy the target site constraints but do not violate the boundary.

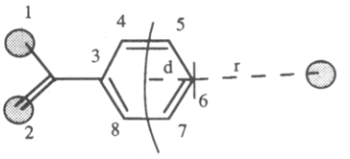


Figure 4. Best vertex defined as the vertex of a skeleton that is closest to an unsatisfied target site, vertex 6. The distance between the target site and the best vertex is given as r. The search can be directed toward the target sites by ensuring that new templates are always added to the best vertex. A wider range of structures can be generated by specifying a distance tolerance d. In this case all vertices at a distance of $(d + r)$ from the target site (vertices 5–7) will be used as seed points for the addition of new templates.

vertex along each of a set of unit vectors centered on the target site that give rise to an even distribution of points on the surface of a sphere. The template is then rotated about this bond to produce a discrete set of orientations (at 60-deg intervals). The resolution determines the number of vectors that are available and hence the number of different orientations of the template. This parameter has a direct effect on the size of the graph since each orientation gives rise to a different node at the first level of the graph.

Skeleton Size. The maximum number of vertices allowed in a skeleton can be specified by the user.

Rings. The maximum number of 3-, 4-, 5-, and 6-membered rings allowed in a skeleton can be specified.

Skeleton Rigidity. A value can be set for the minimum ratio of ring vertices to chain vertices. This provides a way of forcing some rigidity into the structures.

Directed Growth. In the general case, new templates can be added to a partial skeleton at any of its vertices that have the appropriate free valencies. However, in some cases it may be desirable to limit the addition of new templates so that they are always added in the direction of unsatisfied target sites. This is achieved by determining the *best vertex* of a skeleton. The best vertex is defined as the vertex that is closest to an unsatisfied target site. This vertex must then be included in any joining operation; e.g., a new bond can be formed from it, a ring can be spiro joined to it, or it is involved in fusion to a new ring. It is also possible to select an intemediate level where, as shown in Figure 4, the number of vertices used are increased to all those within a user-specified distance tolerance, d, away from the best vertex.

Node Expansions. An exhaustive search of the problem space is achieved by including all of the successors of a node in the search graph. In this case the scoring function of the A algorithm merely orders the nodes for expansion. When the program is applied to a loosely constrained problem that can result in a very large number of structures, it may be desirable to restrict the number of successors of a node that can be included in the graph. This has the effect of sampling the structure space. In this case, the successors are ordered according to their score and the best scoring nodes included in the graph. The number of nodes to be included can be specified by the user.

Monte Carlo Method. As described so far, the templates and skeletons are treated as rigid bodies. This means that whenever a rotatable bond is formed a predefined discrete set of torsional angles are chosen. Conformational flexibility is handled by (i) trying to ensure sufficient coverage of conformational space by sampling and (ii) relaxing the conditions for satisfying the binding constraints, i.e, by using large target sites. However, this method can result in the loss of potential solutions, and so the user is provided with the option of exploring different conformational space using a Monte Carlo method. This has a necessary overhead in terms of processing time, so a number of parameters are available to manage its effect; e.g., the temperature and the number of iterations to be performed can be specified.

Pharmacophore Mode. SPROUT can be applied to the generation of structures to fit a pharmacophore hypothesis. As already described, when the program operates in this mode, a volume is defined by a parallelepiped placed around the target sites. The user is able to specify the size of this box.

2.3. Atom Substitution. The skeletons resulting from primary structure generation do not contain information about element type: the vertices are described by hybridization state alone, and the connections between them are described by bond type. These hybridized vertices are substituted for atoms in order to (1) to confer the appropriate character at binding sites, e.g., hydrogen bond donor, hydrogen bond acceptor, etc.; (2) to stabilize certain bonding situations or conformations, e.g., an enol is normally unstable but substitution of an additional O gives a stable carboxylic acid; (3) to confer certain physical properties, e.g., solubility; and (4) to facilitate ease of synthesis, e.g., from a synthetic viewpoint two rings connected by a hydrocarbon chain can be simplified by substituting heteroatoms into the chains to act as cleavage sites in a retrosynthetic plan.

An outline of atom substitution to satisfy the first of these criteria is described here. A fuller description will be presented in a future publication. The method uses a knowledge base that contains all the information required to perform atom substitutions. Flexibility is built in since the library can be readily updated. In this knowledge base, heteroatoms are stored within functional groups and properties are attributed to specific atoms of the functional groups. The functional groups and their properties are described using a linear notation similar to both SMILES[23] and the PATRAN notation used in the LHASA program.[29] Each functional group entry has an associated rule that describes the necessary atom and bond substitutions to be performed.

Properties are associated with the target sites during the derivation of the constraints for structure generation. Target sites are currently labeled as (1) hydrogen bond donor, (2) hydrogen bond acceptor, or (3) site not specified. The property required at a target site is transferred to any vertex that satisfies that target site. Functional groups having the required property are then extracted from the library and matched against the skeleton. There is a correspondence between an atom of a functional group and a vertex of a skeleton if the hybridization state of the atom matches the hybridization of the vertex, they have the same connectivity, and the properties match. Following atom substitution, any vertices that remain undefined by element type default to carbon.

Note that each skeleton may give rise to many heterosubstituted derivatives because of the many possible combinations of functional groups.

3. Structure Evaluation. Having generated a set of structures for a particular application, methods are required to rank the structures. Different criteria can be used to rank the compounds produced, depending on the particular application. In the case of potential drug leads, transport properties and toxicity are important factors. Irrespective of the application one of the most important features for ranking compounds must be the estimated ease of synthesis. The program CAESA is being developed as an expert system, the aim of which is to make the same judgement on the ease of synthesis of a compound as would be made by an expert synthetic chemist. CAESA includes a number of components including a rule base to detect synthetic complexity and a method of rapidly searching a database of starting materials. The judgment on the ease of synthesis is represented by a number and is accompanied by a description of how the estimate was reached. This description also provides some information on the starting materials from which the compound could be synthesized. A fuller description of this work is in preparation.

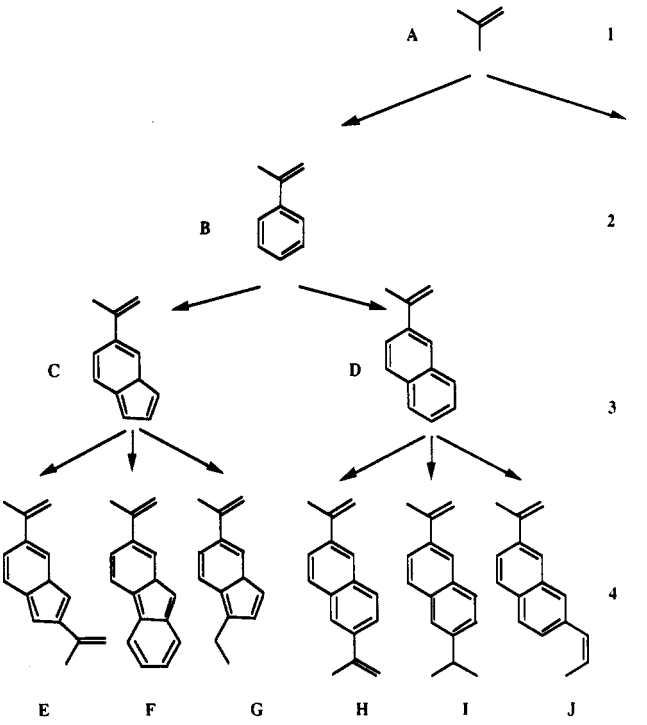


Figure 5. Extract from a search graph showing the history of some goal nodes. The relative positions of nodes in the graph can be used to cluster the solutions.

4. Organizing the Results. The output from SPROUT can typically produce several hundred structures. SPROUT includes a structure display program SKELSHO that allows the user to browse through a large answer set. However, it is also useful to be able to cluster the structures into similar groups. In SPROUT the structures are implicitly clustered by their relative positions in the search graph. Each node in the search graph represents a partial skeleton that is composed of templates. If two nodes have a common ancestor at level n in the graph, then they have n common templates (where the root of the graph, i.e., the initial constraints, is defined as level 0). The solution structures can be clustered simply by recording the history of the nodes in the search graph.

Figure 5 illustrates an extract from a search graph. This section of the graph gives rise to six solutions labeled E–J. The root of the graph represents the initial constraints and has been omitted (as level 0). Level 1 of the graph represents the first template. Subsequent levels of the graph are produced by adding new templates to the growing structures; any nodes that do not lead to solutions have been omitted. This section of the graph then records the history of the solution structures in the search graph. Clustering can be applied at different levels. Structures E–G can be grouped together since they share the common ancestor C. Similarly, structures H–J can be clustered together since they share the common ancestor D. At a higher level of clustering, E–J can be clustered into a single group since they share the common ancestor B. A hierarchy of clusters can be generated in this way.

EXAMPLES

SPROUT has been applied to a number of problems in molecular recognition. Two examples are presented here: (i)

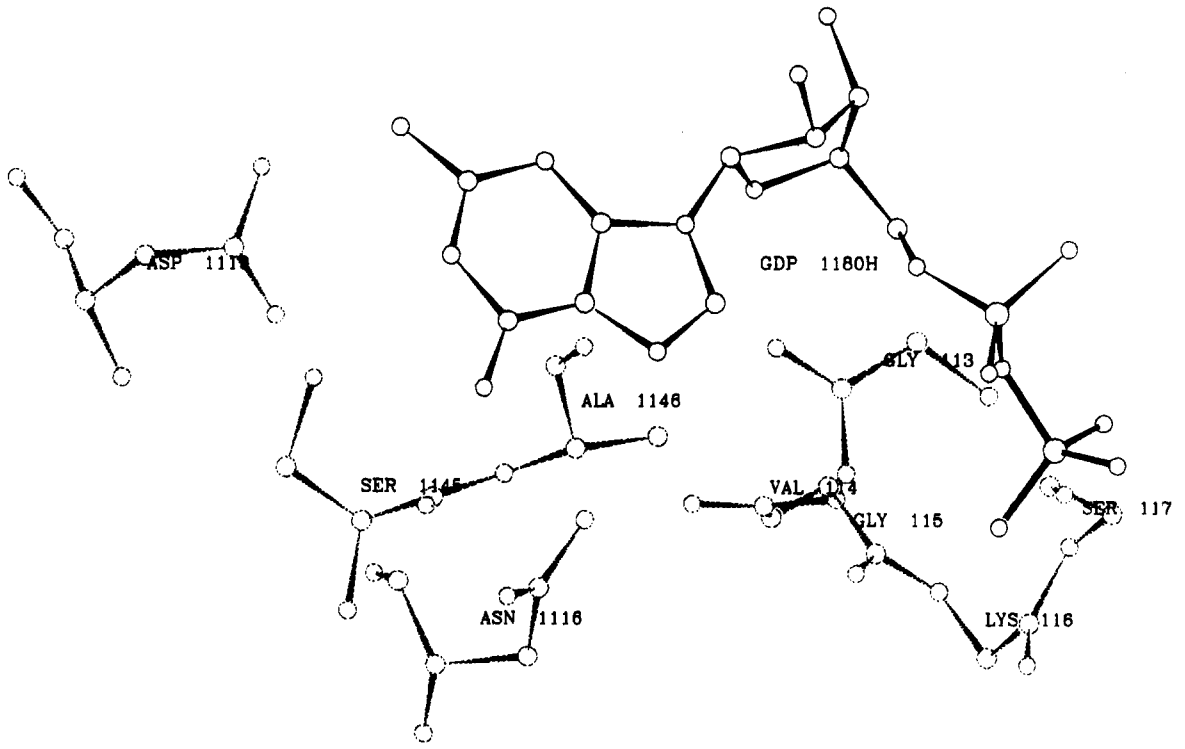


Figure 6. GDP bound in the active site of p21. The guanine base form hydrogen bonds to the side chains of Asn[116] and Asp[119]. The β-phosphate group interacts with the main chain amide groups of residues 13–17.

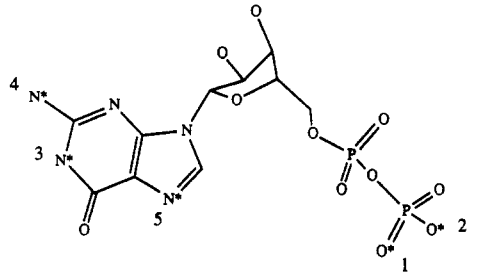


Figure 7. Target sites used for structure generation. Target sites 1–4 were used for the first run. The second run was further constrained by the inclusion of target site 5.

structures are generated to fit the active site of an enzyme of known 3D structure, and (ii) structures are generated to fit a pharmacophore hypothesis.

1. p21. The p21 *ras* proteins are oncogene proteins that bind guanosine triphosphate (GTP) and guanosine diphosphate (GDP).[30] They are involved in the hydrolysis of GTP to GDP. Figure 6 illustrates GDP bound in the active site of p21 (PDB entry 1q21). GDP binds strongly with the polar hydrophilic region of the guanine base interacting with the polar side chains of residues Asn116 and Asp119. The β-phosphate group interacts extensively via hydrogen bonds to the main chain amide groups of residues 13–17. A sphere of radius 10 Å enclosing the active site of p21 and centered on GDP was saved. GDP itself was removed from the coordinates. The remaining coordinates are sufficient to describe the volume of the active site that is available for structure generation. Two runs were performed as detailed below. In both cases target sites were defined based on the atom coordinates of GDP bound to the protein.

Run 1. Four target sites were chosen to model the hydrogen bonding interactions between GDP and the active site. The target sites were specified by the atomic coordinates of two nitrogens of the guanine group that hydrogen bond to Asp119 and two oxygens of the β-phosphate group that interact with the main chain amide groups. They are labeled 1–4 in Figure 7. The target sites at the guanine end of the active site were labeled as hydrogen bond donors, and the target sites at the phosphate end were labeled as hydrogen bond acceptors. These properties are used in the atom substitution phase. Skeleton generation was initiated after defining a library of available templates and setting a number of parameters as described below. The templates available for the run included a full set of acyclic templates and a number of 5- and 6-membered rings in a range of conformations. These are shown in Figure 8.

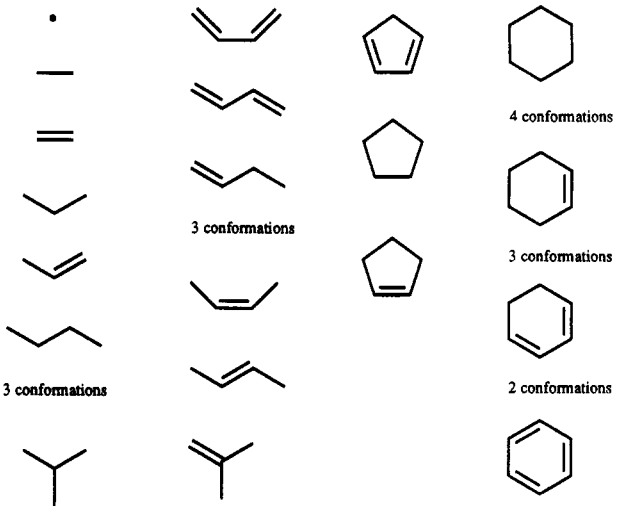


Figure 8. Templates used to generate potential inhibitors of p21.

Parameters were set as follows:

Target site 3 was chosen as the starting point for structure generation, and the template representing benzene was chosen as the first template

The resolution was set to 15°. This gives rise to an even distribution of 176 points on the surface of a sphere centered on the start target site. One bond of the benzene

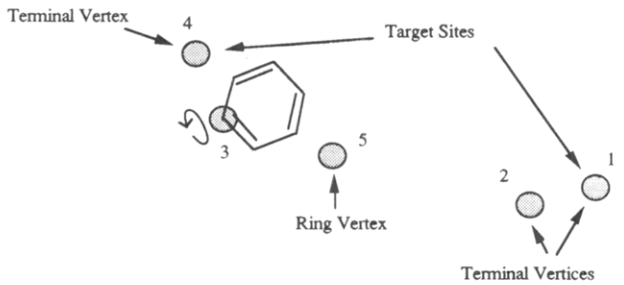


Figure 9. Schematic representation of the initial constraints for the design of potential inhibitors of p21. The first run used target site 1–4. Target site 5 was included in a second run.

template was aligned along each of these vectors in turn with one vertex anchored at the center of the target site. The template was rotated about this bond to give six orientations per vector. Each orientation was then checked against the boundary—any orientation that violated the boundary was discarded. The result was a set of 162 orientations, or 162 nodes at the first level of the search graph.

The maximum number of vertices allowed in a skeleton was restricted to 30, and an upper limit of five 5-membered rings and five 6-membered rings per skeleton was set. The minimum ratio of ring vertices to chain vertices was set at 1:1. This means that at least half of the vertices must be ring vertices. The vertices allowed to cover the target sites were restricted to terminal vertices, except for target site 3, covered by the benzene ring.

The graph was searched exhaustively and no Monte Carlo procedure was used so that the skeletons and templates were treated as rigid structures. However, a number of conformations were produced whenever a rotatable bond was made.

The nonboundary constraints on skeleton generation are shown schematically in Figure 9.

A total of 360 skeletons were generated in approximately 6 h of cpu time (using one R4000 processor of a Silicon Graphics Challenge). The graph was searched exhaustively, i.e., all the nodes were expanded, and produced a total of 43 378 nodes. SPROUT writes the solutions to disk as they are generated so that it is possible to browse through the answers as the search is continuing. The first solutions were found after 5 min of cpu time. These solutions represent hydrocarbon skeletons that satisfy the steric constraints; i.e., they are able to fit into the active site and have vertices at the appropriate positions to be able to form interactions to the protein. A variety of different skeletons were produced, and a sample is shown in Figure 10. One of these skeletons (B) is shown in Figure 11 superimposed on GDP in the active site of p21. It can be seen that it follows the backbone of GDP to occupy a similar volume in the active site.

The results of atom substitution applied to one of these skeletons (skeleton A) is shown in Figure 12. The hydrogen bonding properties required at the target sites were specified as shown: hydrogen bond acceptors are required at target sites 1 and 2, and hydrogen bond donors are required at target sites 3 and 4. A small library of functional groups was compiled, and their hydrogen bonding properties were specified. Two hydrogen bond accepting functional groups were matched to target sites 1 and 2, a carboxy group and an amidino group. Hydrogen bond donors were matched to target sites 3 and 4 in two different ways. This resulted in the four molecules shown in Figure 12.

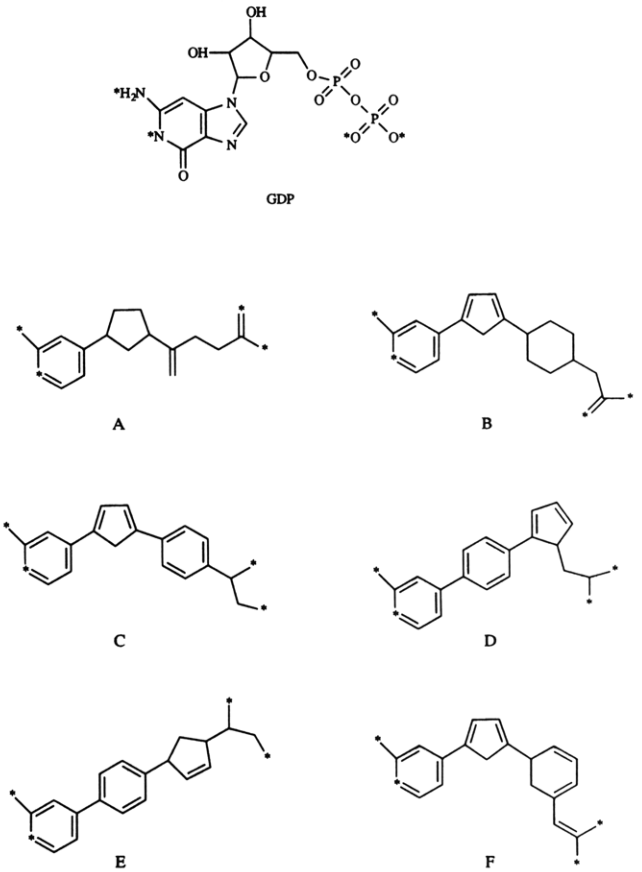


Figure 10. Sample of the solution skeletons generated to fit four target sites. The asterisks indicate the vertices that correspond to the target sites.

Run 2. The search was further constrained in the second run by the addition of an extra target site. This target site is labeled 5 in Figure 9 and unlike sites 2, 3 and 4 was constrained to be a ring vertex. All the other parameters for the run remained the same. A set of 122 solutions was generated in approximately 87 min of cpu time. Some of the results are shown in Figure 13, and skeleton G is shown superimposed on GDP in Figure 14. The total size of the search graph was 11 791 nodes. The additional target site had the effect of reducing the size of the search graph and also the number of solutions with a corresponding reduction in processing time.

2. Morphine. The second example demonstrates the ability of SPROUT to operate in "pharmacophore mode". A hypothesis for the mode of binding of morphine agonists has been proposed by Taylor and Kennewell.[31] Morphine is shown in Figure 15. The structural requirements include: a quaternary carbon atom, an aromatic nucleus linked to this carbon, and a tertiary amino group two saturated carbon atoms away from the quaternary carbon. These requirements were modeled by specifying target sites for structure generation using atomic coordinates of some of the atoms of morphine, as shown in Figure 16. (Morphine was extracted from the Cambridge Structural Database[14]). A volume for structure generation was derived by placing a rectangular box around the target site coordinates. Initially the box was sized by using the extremes of the target site coordinates in each dimension. It was then enlarged in each direction by 1.0 Å. The templates consisted of 5- and 6-membered rings, and the saturated acyclic templates are shown in Figure 17.

The first template was benzene, and this was anchored at target sites 1 and 2, as shown in Figure 18. The maximum

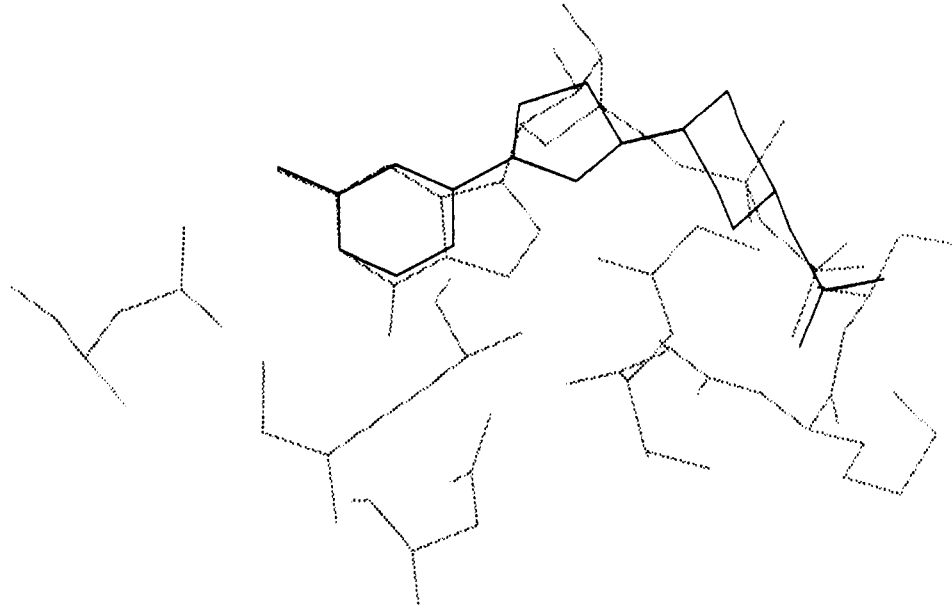


Figure 11. Skeleton B (bold) shown superimposed on GDP (light) in the active site of p21.

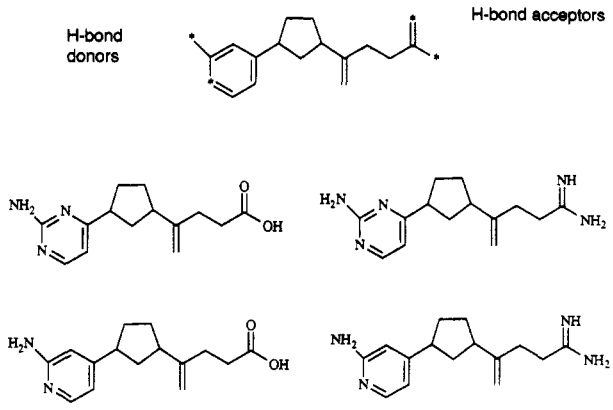
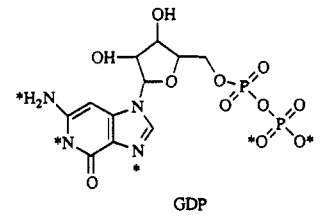


Figure 12. Atom substitution performed on skeleton A. Each skeleton can give rise to a number of molecules following heteroatom substitution. A small library of functional groups was used to guide the heteroatom substitution according to the hydrogen bonding constraints shown.



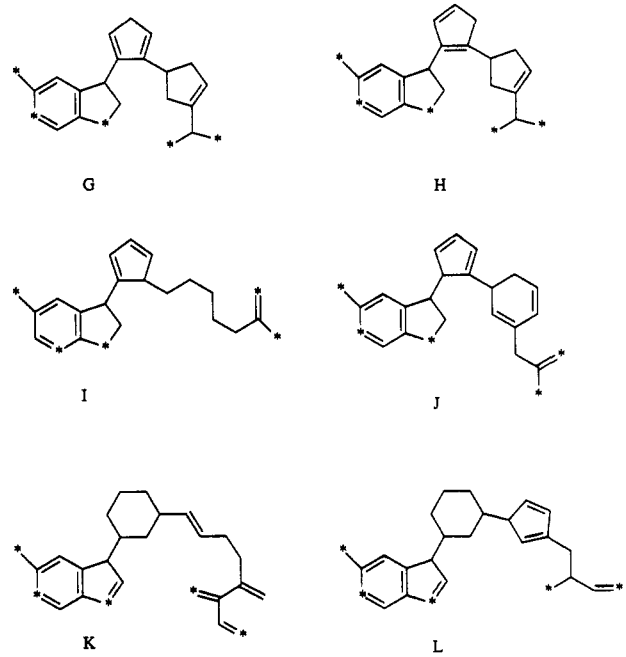


Figure 13. Sample of the solution skeletons generated to fit five target sites. The asterisks indicate the vertices that correspond to the target sites.

number of vertices allowed in a skeleton was limited to 25, and at least 60% of the vertices had to be ring vertices. Restrictions at the target sites were as follows: target site 4 had to be covered by a ring vertex; target site 6 had to be covered by a terminal vertex; and no restriction was placed on the vertices at target sites 3 and 5. The maximum number of 5-membered rings permitted in a skeleton was restricted to three, and the maximum number of 6-membered rings was restricted to five. The search graph was expanded exhaustively, and the skeletons and templates were treated as rigid structures.

A total of 168 skeletons were generated in approximately 3 h of cpu time (using one R4000 processor of a Silicon Graphics Challenge). The first structures were generated after 4 min. The total size of the search graph was 53 371 nodes, and the search ran to completion. Some of these structures are shown in Figure 19. Example structures are shown superimposed on morphine in Figures 20 and 21.

Morphine itself was not generated. This can be because the search is directed by the target sites; once all of the target sites have been covered by vertices, the skeleton is considered to be a solution and forms a goal node in the search graph. Skeleton M is a substructure of morphine. It is possible that changing the termination condition used in SPROUT so that new templates would continue to be added until the volume is filled would allow an additional ring to be fused to this skeleton to coincide with the B ring of morphine. The second example shows a skeleton that superimposes well with the A–D rings of morphine.

The search graph in the example is larger than those produced in the p21 runs. This can be because of the poorly described volume. The boundary derived from the active site

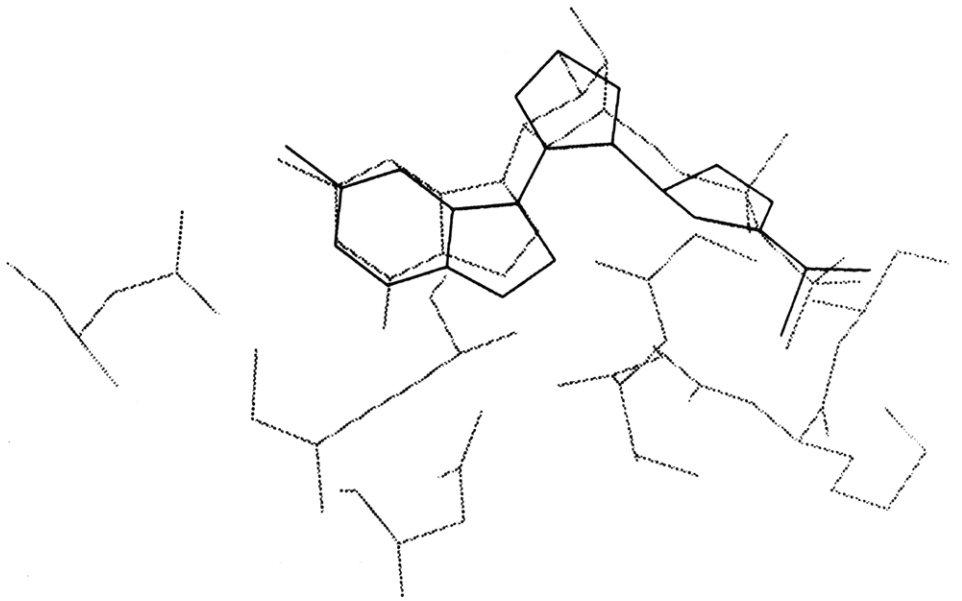


Figure 14. Skeleton G (bold) shown superimposed on GDP (light) in the active site of p21.

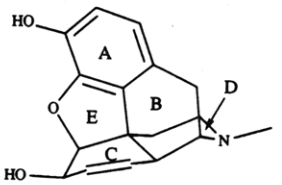


Figure 15. Morphine.

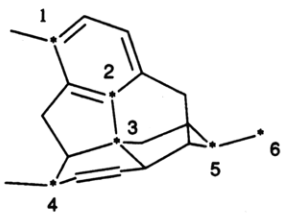


Figure 16. Atoms used to define target sites for structure generation labeled by an asterisk.

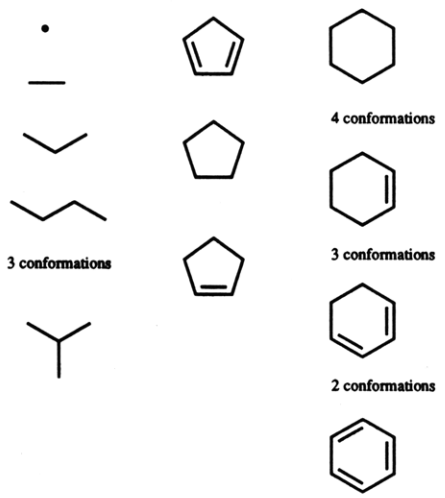


Figure 17. Templates used to generate potential morphine agonists.

atoms of p21 can provide an efficient means of pruning the search graph. In contrast the relatively large box containing the target sites in this example is less effective in pruning the graph. However, each node expansion step should be faster in this case because of the smaller number of templates used.

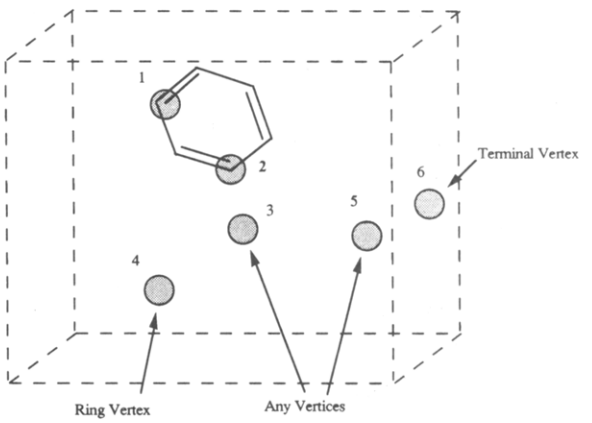


Figure 18. Schematic representation of the initial constraints for the generation of potential morphine agonists. The rectangular box indicates the volume available for structure generation.

CONCLUSIONS

Structure generation is a problem of enormous complexity. The method used by SPROUT, graph searching, is a commonly employed algorithm that has been used to solve a number of problems. In general, different methods can be used to decrease the complexity of graph searching, for example, backtracking, heuristics, and the A (or A*) algorithms. SPROUT uses the A algorithm and incorporates some knowledge about the problem domain to decrease the number of possibilities.

In SPROUT a number of factors are used to reduce processing costs involved in the combinatorial process. These include using generalized molecular fragments as building blocks, the selection of the start target site; the selection of the starting template(s); the discrete and limited resolution that governs the size of the graph at the first level; treating the templates and skeletons as rigid bodies; and restricting the coverage of torsional angles about rotatable bonds. Some of these factors are under the control of the users who are also provided with a number of parameters to further restrict the size of the search graph and to tailor the solutions for their own requirements. These factors are combined in SPROUT to provide a practical solution to the problem of structure generation.

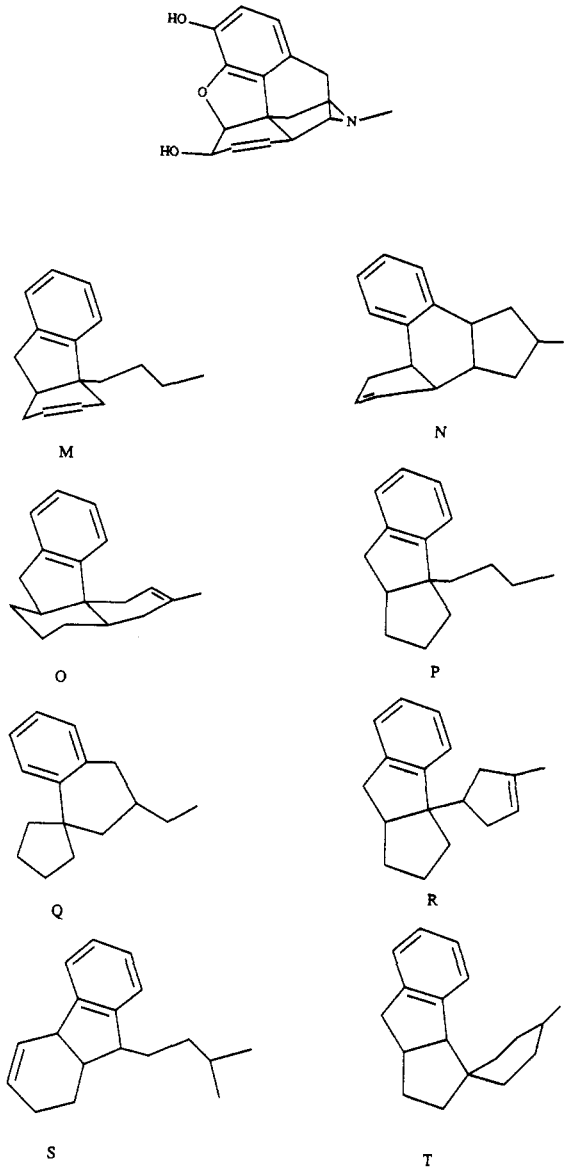


Figure 19. Sample of the solution skeletons generated to fit the morphine agonist hypothesis.

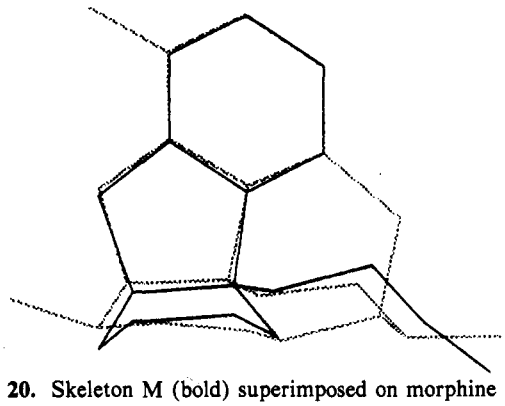


Figure 20. Skeleton M (bold) superimposed on morphine (light).

We are continuing to develop SPROUT in a number of areas. These include the development of a more flexible approach for specifying both hydrogen bonding and hydrophobic target sites; optimizing the structures in conformational space following atom substitution; providing a more flexible interface that will allow the user to interact with the search graph during processing; and developing methods for ranking the output, e.g., by estimating their synthetic accessibility and by scoring their fit to the specified receptor.

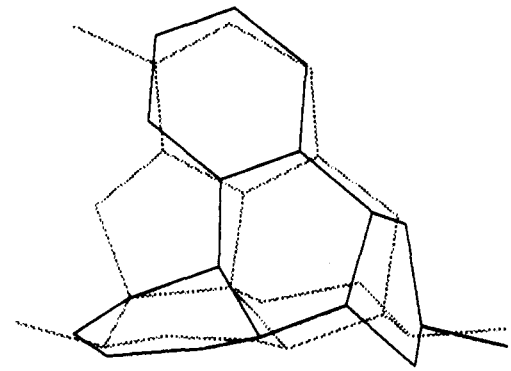


Figure 21. Skeleton N (bold) superimposed on morphine (light).

REFERENCES AND NOTES

(1) Gillet, V. J.; Johnson, A. P.; Mata, P.; Sike, S.; Williams, P. SPROUT: A Program for Structure Generation. *J. Comput.-Aided Mol. Des.* **1993**, *7*, 127–153.
(2) Gillet, V. J.; Johnson, A. P.; Mata, P.; Sike, S. Automated Structure Design in 3D. *Tetrahedron Comput. Methodol.* **1990**, *3* (6C), 681–696.
(3) Van Drie, J. H.; Weininger, D.; Martin, Y. C. ALADDIN: An Integrated Tool for Computer-Assisted Molecular Design and Pharmacophore Recognition from Geometric, Steric, and Substructure Searching of Three Dimensional Molecular Structures. *J. Comput.-Aided Mol. Des.* **1989**, *3*, 225–251.
(4) Bartlett, P. A.; Shea, G. T.; Telfer, S. J. In *Molecular Recognition: Chemical and Biological Problems*; Roberts, S. M., Ed.; The Royal Society of Chemistry: London, 1989; pp 182–196.
(5) DesJarlais, R. L.; Sheridan, R. P.; Seibel, G. L.; Dixon, J. S.; Kuntz, I. D.; Venkataraghavan, R. Using Shape Complementarity as an Initial Screen in Designing Ligands for a Receptor Binding Site of Known Three-Dimensional Structure. *J. Med. Chem.* **1988**, *31*, 722.
(6) Wade, R. C.; Goodford, P. J. Further Development of Hydrogen Bond Functions for Use in Determining Energetically Favorable Binding Sites on Molecules of Known Structure. 2. Ligand Probe Groups with the Ability To Form More Than Two Hydrogen Bonds. *J. Med. Chem.* **1993**, *36*, 148–156.
(7) Wade, R. C.; Clark, K. J.; Goodford, P. J. Further Development of Hydrogen Bond Functions for Use in Determining Energetically Favorable Binding Sites on Molecules of Known Structure. 1. Ligand Probe Groups with the Ability To Form Two Hydrogen Bonds. *J. Med. Chem.* **1993**, *36*, 140–147.
(8) Goodford, P. J. A Computational Procedure for Determining Energetically Favorable Binding Sites on Biologically Important Macromolecules. *J. Med. Chem.* **1985**, *28*, 849–857.
(9) Danziger, D. J.; Dean, P. M. Automated Site-Directed Drug Design: The Prediction and Observation of Ligand Point Positions at Hydrogen-Bonding Regions on Protein Surfaces. *Proc. R. Soc. London* **1989**, *B236*, 115–124.
(10) Danziger, D. J.; Dean, P. M. Automated Site-Directed Drug Design: A General Algorithm for Knowledge Acquisition About Hydrogen-Bonding Regions at Protein Surfaces. *Proc. R. Soc. London* **1989**, *B236*, 101–113.
(11) Lewis, R. A.; Dean, P. M. Automated Site-Directed Drug Design: the Concept of Spacer Skeletons for Primary Structure Generation. *Proc. R. Soc. London* **1989**, *B236*, 125–140.
(12) Böhm, H. J. LUDI: Rule-Based Automatic Design of New Substituents for Enzyme Inhibitor Leads. *J. Comput.-Aided Mol. Des.* **1992**, *6*, 593–606.
(13) Böhm, H. J. The Computer Program LUDI: A new method for de novo design of enzyme inhibitors. *J. Comput.-Aided Mol. Des.* **1992**, *6*, 61–78.
(14) Allen, F. H.; Bellard, S.; Brice, M. D.; Cartwright, B. A.; Doubleday, A.; Higgs, H.; Hummelink, T.; Hummelink-Peters, B. G.; Kennard, O.; Motherwell, W. D. S.; Rodgers, J. R.; Watson, D. G. The Cambridge Crystallographic Data Centre: Computer-Based Search, Retrieval, Analysis and Display of Information. *Acta Crystallogr.* **1979**, *B35*, 2331–2339.
(15) Miranker, A.; Karplus, M. Functionality Maps of Binding Sites: A Multiple Copy Simultaneous Search Method. *Proteins: Struct. Funct. Genet.* **1991**, *11*, 29–34.
(16) Eisen, M.; Wiley, D. C.; Karplus, M.; Hubbard, R. E. HOOK: A Program for Finding Novel Molecular Architectures That Satisfy the Chemical and Steric Requirements of a Macromolecule Binding Site. Submitted for publication in *Proteins: Struct., Funct. Genet.*
(17) Nishibata, Y.; Itai, A. Automatic Creation of Drug Candidate Structures Based on Receptor Structure. Starting Point for Artificial Lead Generation. *Tetrahedron* **1991**, *47*, 8985–8990.
(18) Rotstein, S. H.; Murcko, M. A. GenStar: A Method for *De Novo* Drug Design. *J. Comput.-Aided Mol. Des.* **1993**, *7*, 23–43.
(19) Rostein, S. H.; Murcko, M. A. GroupBuild: A Fragment-Based Method for *De Novo* Drug Design. *J. Med. Chem.* **1993**, *36*, 1700–1710.

(20) Moon, J. J.; Howe, W. J. Computer Design of Bioactive Molecules: A Method for Receptor-Based de Novo Ligand Design. *Proteins: Struct., Funct. Genet.* **1991**, *11*, 314–328.
(21) Moon, J. B.; Howe, W. J. Computer Design of Ligands: Recent Developments in the "GROW" Program. Presented at The Molecular Graphics Society Meeting on Binding Sites, York, U.K., March 1993.
(22) Weininger, D.; Dixon, J. S.; Blaney, J. M. Evolution of Molecules to Fit a Binding Site of Known Structure. Presented at The Molecular Graphics Society Meeting on Binding Sites, York, U.K., March 1993.
(23) Weininger, D. Smiles. 3. Depict. Graphical Depiction of Chemical Structures. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 237–243.
(24) Payne, A. W. R.; Glen, R. C. Molecular Recognition Using a Binary Genetic Search Algorithm. *J. Mol. Graphics* **1993**, *11*, 74–91.
(25) Miranker, A. Ligand Perturbation Space: An Algorithm for De Novo Ligand Design. Presented at The Molecular Graphics Society Meeting on Binding Sites, York, U.K., March 1993.
(26) Lewis, R. A.; Roe, D. C.; Huang, C.; Ferrin, T. E.; Langridge, R.; Kuntz, I. D. Automated Site-Directed Drug Design using Molecular Lattices. *J. Mol. Graphics* **1992**, *10*, 66–78.
(27) Teig, S. L. In *Proceedings of the Montreux 1992 International Chemical Conference*; Collier, H., Ed.; Infonortics Ltd.: Calne, U.K., 1992; pp 195–208.
(28) Nilsson, N. J. *Principles of Artificial Intelligence*: Springer-Verlag: Berlin, 1982.
(29) Hopkinson, G. A. Computer-Assisted Organic Synthesis Design. Ph.D. Thesis, Leeds University, 1985.
(30) Tong, L.; de Vos, A. M.; Milburn, M. V. Crystal Structures at 2.2 Å Resolution of the Catalytic Domains of Normal *ras* Protein and an Oncogene Mutant Complexed with GDP. *J. Mol. Biol.* **1991**, *217*, 503–516.
(31) Taylor, J. B.; Kennewell, P. D. *Introductory Medicinal Chemistry*; Ellis Horwood Ltd.: Chichester, U.K., 1981.