

Development and testing of a *de novo* drug-design algorithm

Eric Pellegrini & Martin J. Field*

Laboratoire de Dynamique Moléculaire, Institut de Biologie Structurale – Jean-Pierre Ebel – CEA/CNRS, 41 Rue Jules Horowitz, F - 38027 Grenoble Cedex 01, France

Received 30 January 2003; accepted in revised form 20 August 2003

Key words: *de novo* drug design, genetic algorithms, neuraminidase, sequential growth, structure-based drug design

Summary

In this article we present an implementation of a *de novo* drug-design algorithm. The algorithm starts with a molecule placed in the binding site of a protein and then modifies it using a sequential growth approach. This involves successive cycles of suppression of randomly picked groups in the molecule and their replacement by other groups chosen from databanks of linear or cyclic fragments. The algorithm has been coupled with the DYNAMO library which allows the simulation of macromolecules using molecular mechanical and quantum chemical methods. The main body of the article describes the methodologies we use to create, characterize and evaluate putative ligands. We also consider briefly an application of the algorithm to a protein of pharmacological interest, the neuraminidase of the influenza virus, and discuss the strengths and weaknesses of our approach.

Introduction

Over the last decade, the amount of information concerning the three-dimensional structures of biomolecular targets has increased dramatically thanks to improvements in experimental techniques, such as X-ray crystallography and NMR spectroscopy [1]. During the same period, there have been large increases in computer power, due principally to improvements in hardware but also in software. The combination of these two factors has helped feed the search for compounds that are active against specific biological targets using structure-based drug-design (SBDD) approaches. SBDD can be defined as the use of structural information about a given target for the design of compounds that bind strongly to it. There are a number of challenges to SBDD, including how to generate candidate inhibitors and how to estimate their binding affinity with a reasonable precision given the complexity of the protein-ligand interaction and the large number of conformational states that are often accessible to both macromolecule and ligand.

Several algorithms have been developed to take advantage of knowledge about the structure of a receptor for the search for putative ligands. In one class of methods, molecules from predefined databases are docked into the active site of the receptor and their orientational and conformational spaces are scanned until a good conformation is found. Programs in this category include DOCK [2], FLEXX [3], GOLD [4] and AUTODOCK [5]. These methods have the advantage that they are fast and allow databases of thousands of molecules to be screened rapidly. However, by definition, they cannot propose molecules with structures that are different from those present in the predefined databases.

To overcome this fundamental limitation, *de novo* drug-design algorithms have been developed. These can be divided into two broad categories. In one, possible binding sites for various chemical groups are found within the active site of the protein and then a putative ligand is constructed by connecting the fragments together. By contrast, in sequential growth approaches, a molecular precursor is placed into the active site of the target and the molecule is mutated by the addition and removal of chemical groups. Programs based on the connection of fragments include

*To whom correspondence should be addressed. E-mail: mjfield@ibs.fr

LUDI [6] and PRO-LIGAND [7] whereas the programs GROW [8] and SPROUT [9] use sequential growth. The advantage of *de novo* drug-design approaches is that, in principle, they provide a powerful way to explore the physico-chemical space offered by the active site of a given target. Thus, they can be employed to generate novel structures as starting points for the elaboration of new active compounds or they can be used to refine the structures of existing lead compounds.

The aim of the current paper is to give a methodological overview of a *de novo* drug-design algorithm that we have developed and which is based upon a sequential growth methodology. We undertook this work not only to have a functional program but also because we wanted a flexible framework within which we could investigate and test out various drug-design algorithms. Our implementation builds upon and benefits from many of the capabilities afforded by the DYNAMO molecular modeling library [10, 11] which has been developed in our laboratory. The paper is organized as follows. First, we describe in detail the main features of the algorithm in the order that they would occur when the algorithm is run. Second, we present the results of different tests of the methodologies used by the algorithm, a step that is important in understanding how well the algorithm behaves. Finally, we consider a preliminary application of the algorithm to the neuraminidase of the influenza virus. Influenza remains a major public health problem and the design of active compounds that block neuraminidase is a major field of current pharmacological research [12].

The algorithm

Overview

The algorithm developed in this study belongs to the class of *de novo* drug-design algorithms that employ a sequential growth methodology. The program implementing the algorithm was written in FORTRAN 90 and can be used *via* a graphical interface written in Python (Python programming language, <http://www.python.org>). The program makes extensive use of the DYNAMO library of molecular modeling subroutines which was developed in our laboratory for the simulation of macromolecules using molecular mechanical (MM) and quantum mechanical (QM) potentials [10, 11]. Details of the algorithm not covered in this article may be found in reference [13].

Description

Figure 1 shows the flowchart of the algorithm, the details of which will be discussed further below. From a general point of view, the algorithm works by selecting parts of a molecular precursor initially placed into the active site of the receptor and replacing them by chemical fragments randomly chosen from a database of predefined fragments. The new molecule is then scored and kept as the new precursor if it scores better than the previous one. If it scores worse, it can still be accepted if it passes a Monte Carlo test of the following form:

$$\exp(-\beta\Delta) \geq \rho \quad (1)$$

Here β is a user-defined constant (≥ 0) whose value determines the strictness of the Monte Carlo procedure, Δ is the difference between the score of the current molecule and the precursor and ρ is a random number between 0 and 1. If this condition is satisfied, the new molecule becomes the new precursor.

To start a simulation, the algorithm needs as prerequisites the coordinate files containing the structures of the receptor and the precursor with explicit hydrogens (formats may be either DYNAMO, Protein Data Bank (PDB) [14] or XYZ), the connectivity of the precursor and a file (the *topology* file) giving details of the receptor's MM energy function (in particular, the atoms' charges and Lennard-Jones parameters). For the precursor, we developed a codification for the different types of atom that the algorithm can generate. This codification specifies all the properties of the atom including, for example, whether it is cyclic or non-cyclic and what its valence state is. A full list of the atom types is given in Figure 2 from which it can be seen that the atom type depends mainly on the valence state of the atom, although sometimes it also takes into account the atom's charge or aromaticity. We think that this collection of atom types covers most species that will be of use in the design of biologically active molecules.

The Dynamo simulation library

The DYNAMO module library was developed for the simulation of molecular systems using QM, MM and hybrid QM/MM potentials. The library supports a range of different types of molecular calculation including geometry optimization, reaction-path determination and molecular dynamics and Monte Carlo simulations [10, 11]. It is written in FORTRAN 90.

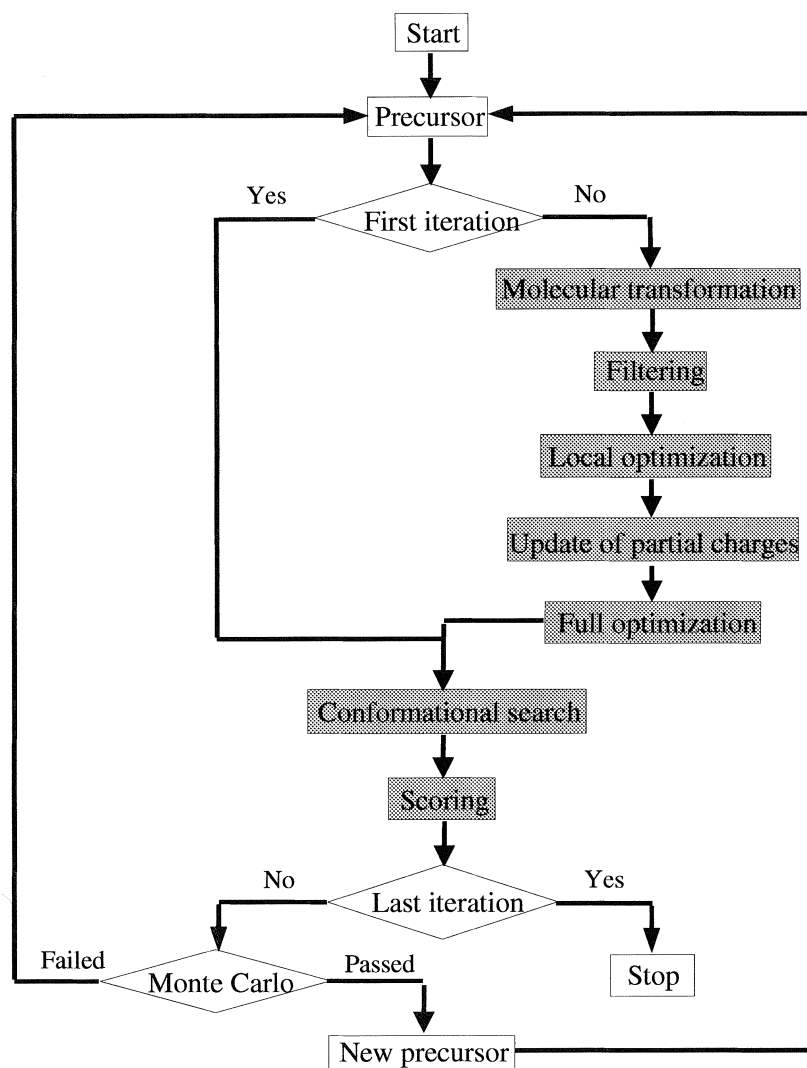


Figure 1. Flowchart of the main steps of the drug-design algorithm.

The main parts of the drug-design algorithm

Choice of a group to be destroyed

This is the first step of each iteration of the simulation and consists of the random selection of an atom of the precursor that will be transformed to create a new molecule. At the start of a simulation users select atoms that are to be conserved and those parts of the precursor that are transformable. However, after this initial choice, the algorithm is fully automatic and there is no intervention of the user in the design process.

Creation of a new molecule

After having selected an atom of the precursor to modify, the algorithm enters one of its fundamental steps – the geometrical modification of the precursor. The algorithm distinguishes between two cases depending upon whether the chosen atom belongs to a ring or not. If the atom belongs to a ring, the algorithm either breaks the ring or it fuses the ring with another one along one of the ring bonds involving the chosen atom. If the atom does not belong to a ring, the algorithm either replaces one of the groups connected to the atom with a new chemical fragment or it grafts a ring onto one of the chosen atom's bonds. For each transformation in which chemical groups are

being added, the groups are selected from two fragment databases that were developed specifically for our study and which contain many of the chemical functions that are important in pharmacology. The databases are shown in Figures 3 and 4. The first contains groups that can be used for replacing groups connected to non-ring atoms whereas the second contains rings for ring fusion and grafting operations. It should be noted that we constructed these databases manually, although algorithms do exist that generate fragments from databases of bioactive compounds (see, for example, references 15 and 16).

In addition to the above transformations, the algorithm can also transmutate the elemental type of an atom. Thus, for example, it is possible to transmute the cyclic oxygen of furane into a nitrogen to give pyrrole.

An important aspect of the algorithm is to ensure that the valencies of any atoms involved in a transformation maintain their correct values. In cases where there is a reduction in the available valence (for example, in which a C–Me group is replaced by a carbonyl), the algorithm will suppress the necessary number of additional sidechains. When the available valence increases, the algorithm will add an appropriate number of hydrogens to the atoms involved in the new bonds.

Filtering of the new molecule

Once the new molecule has been created, a set of physico-chemical filters is automatically applied to see whether the new molecule should be considered as a viable replacement for the precursor and has the potential to be a reasonable ligand. This step is essential to prevent the generation of highly complex or chemically intractable molecules which is one of the main weak points of *de novo* drug-design approaches. The filters we have implemented are listed in Table 1 and may be activated individually by the user. At this early stage in the evaluation of a new molecule, only filters that can be derived from the molecule's connectivity table are applied. These are the Lipinski rule of five [17], the creation of a forbidden bond, the size of the molecule, the number of fluorine, chlorine, bromine, sulfur and phosphorus atoms, the molecular charge, the number of rings and the number of flexible and triple bonds. The other filters are used later in the simulation during the local and global optimization and conformational search steps. In addition to applying the filters, the algorithm also verifies that the newly generated molecule has not been generated previously in the run.

Local optimization of the new molecule

After creation of the new molecule, it is geometry-optimized locally in order to rearrange the added fragments and atoms around the transformation site. For this, the algorithm optimizes the following simplified potential energy function, \mathcal{V} :

$$\mathcal{V} = \sum_{\text{bonds}} (r - r_0)^2 + \sum_{\text{angles}} (\theta - \theta_0)^2 + \sum_{\substack{\text{planar} \\ \text{dihedrals}}} \sin^2(\varphi_d) + \sum_{\substack{\text{planar} \\ \text{impropers}}} \sin^2(\varphi_i) \quad (2)$$

In this equation, r refers to a bond distance, θ to a bond angle, the subscript 0 to an equilibrium (or reference) value and φ_d and φ_i are the proper and improper dihedrals of planar groups (such as peptide bonds and aromatic rings). The use of a simplified mathematical expression for a potential like ours has already been applied with success by Gasteiger et al. [18]. For the reference bond length and angle parameters, r_0 and θ_0 , we use the recipe prescribed by the UFF force field [19] which, thanks to its very general formulation, allows the algorithm to generate the parameters automatically. The geometry optimization itself is carried out using the conjugate gradient algorithm implemented in the DYNAMO library.

Update of the partial charges of the new molecule

To proceed further in the refinement of the new molecule, it is necessary to complete the specification of its potential energy function by defining the electrostatic and Lennard-Jones parameters required for the calculation of the non-bonding energy. The Lennard-Jones parameters can be determined automatically as they depend uniquely on the type of atom. In contrast, parametrization of the charges on the precursor atoms causes more problems as an atom's charge depends upon its environment and, most importantly, upon the other atoms to which it is covalently bound. To have a method that automatically generates the charges, we chose to use a quantum mechanical approach which, by its nature, requires only the geometry of the molecule and the elemental type of its atoms.

To ensure that the calculation of the partial charges is not too onerous, the algorithm selects a portion of the new molecule around the transformation site. This portion consists of all the new added atoms as well as parts of the original precursor close to the transformation site. Any bonds broken when creating the simplified molecule in this way are completed by adding appropriate atoms, such as hydrogen. Once

the simplified molecule has been defined, including its geometry, the charges are calculated for the molecule in vacuum. Currently, users have the choice between Mulliken and Class IV [20] charges using one of the three semi-empirical methods, MNDO, PM3 or AM1 [21–23], which are all implemented in the DYNAMO library. Finally the charges for the simplified model are transposed back onto the full molecule using an interpolation scheme that will be described below.

Full optimization of the new molecule

The next stage in the algorithm requires the placement of the new molecule in the active site and its geometry optimization so as to ensure that it has a reasonable conformation for the subsequent search procedure. For this optimization, we use the potential \mathcal{V} defined as:

$$\mathcal{V}_{\text{Full}} = \mathcal{V} + \sum_i^{\text{ligand}} \sum_{j>i}^{\text{ligand}} \frac{q_i q_j}{r_{ij}} + \sum_i^{\text{ligand}} \sum_{j>i}^{\text{ligand}} 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] + \sum_i^{\text{ligand}} \sum_j^{\text{receptor}} \frac{q_i q_j}{r_{ij}} + \sum_i^{\text{ligand}} \sum_j^{\text{receptor}} 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right], \quad (3)$$

where q_i is the partial charge of atom i of the ligand or the protein, r_{ij} is the distance between atom i and j and \mathcal{V} is the pseudo-potential defined in Equation 2. For the Lennard-Jones energies, geometrical combination rules are used such that $\epsilon_{ij} = \sqrt{\epsilon_i \epsilon_j}$ and $\sigma_{ij} = \sqrt{\sigma_i \sigma_j}$ and for which ϵ_i and σ_i are the Lennard-Jones well-depth and radius parameters of the ligand and protein atoms. The charges of the ligand atoms are those determined by the scheme of the previous, whereas for the charges of the protein atoms and the Lennard-Jones parameters for both ligand and protein atoms we use values from the OPLS-AA force field [24]. The Lennard-Jones parameters for ligand atoms are listed in Table 2. It should be noted that the sums over atoms pairs in the internal non-bonding energies for the ligand exclude all 1–2, 1–3 and 1–4 interactions or, in other words, all atoms which are separated by three or fewer bonds.

Conformational search

To tackle the thorny problem of ligand flexibility, we decided to perform a conformational search of the ligand's degrees of freedom whilst keeping the protein fixed. Most existing drug-design software also makes this assumption although there have been some attempts at investigating receptor flexibility [25, 26].

In our conformational search, we consider both the orientational and the conformational degrees of freedom of the ligand. In other words, we optimize the molecule by rotating and translating the molecule in the active site and by changing the angles about its various rotatable bonds. Any conformational search is faced with the difficulty of exploring a vast range of structure space and so global optimization methods, such as simulated annealing, tabu search or genetic algorithms, are the most appropriate. We decided to perform our conformational search with a genetic algorithm that we developed in the laboratory and which we tested extensively both for our drug design algorithm and on standard problems, such as the optimization of Lennard-Jones clusters and the traveling salesman problem [27]. A genetic algorithm has already given good results for the same task in the docking programs GOLD [4] and DOCK [28]. To employ the genetic algorithm, we chose a function to optimize (the so-called adaptation or fitness function) of the following form:

$$\mathcal{A} = E_{\text{el}} + E_{\text{LJ}} + \mathcal{U} \quad (4)$$

with

$$E_{\text{el}} = \sum_i^{\text{ligand}} \frac{q_i \tilde{\phi}_e(\mathbf{r}_i)}{\epsilon} \quad (5)$$

and

$$E_{\text{LJ}} = \sum_i^{\text{ligand}} 2\sqrt{\epsilon_i} (\sqrt{\sigma_i})^{12} \tilde{\phi}_r(\mathbf{r}_i) + \sum_i^{\text{ligand}} 2\sqrt{\epsilon_i} (\sqrt{\sigma_i})^6 \tilde{\phi}_d(\mathbf{r}_i), \quad (6)$$

where q_i , ϵ_i and σ_i are defined in Equation 3. $\tilde{\phi}_e$, $\tilde{\phi}_r$ and $\tilde{\phi}_d$ are the electrostatic, repulsive and dispersive fields at the atom centres created by the protein atoms and \mathcal{U} is the internal energy of the molecule defined by the following equation:

$$\mathcal{U} = \mathcal{V} + \sum_{i,j} \frac{q_i q_j}{r_{ij}} + \sum_{i,j} 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right], \quad (7)$$

where all the terms are defined in Equation 3.

For the calculation of the fields due to the protein, the algorithm proceeds in the following manner. First, the electrostatic, repulsive and dispersive fields (ϕ_e , ϕ_r and ϕ_d) are computed on a regular grid surrounding the initial precursor. This grid is defined by the parameters $N_{\text{Axis}}^{\text{Sense}}$ and G_S . $N_{\text{Axis}}^{\text{Sense}}$ describes the number of

Table 1. Various physico-chemical filters, and their associated default values, used by the algorithm to detect inappropriate structures.

Physico-chemical filter	Default value
Lipinski Rule of five	Ref. 17
Number of sulfur and phosphorus atoms	≤ 2
Number of fluorine atoms	≤ 3
Number of chlorine atoms	≤ 1
Number of bromine atoms	≤ 1
Molecular charge	$> -2, < 2$
Total number of atoms	≤ 70
Number of chiral atoms in aliphatic chains	≤ 2
Number of triple bonds	≤ 1
Number of flexible bonds	≤ 12
Number of rings	≤ 3
Forbidden bond ((N,O,P,S)-(F,Cl,Br), P-P, P-S, S-S, O-O, N-N)	–
Number of failure of local optimization	5
Number of failure of global optimization	1
Redundant transformation (e.g. H replaced by H)	–
Redundant molecule	–
Number of times the molecule gets out of the grid during the conformational search	≥ 5000
Failure of the conformational search	–
Failure of partial charges calculation	–
Maximum deviation from planarity for aromatic rings	10°

points between the center of the grid and its extremity along each cartesian axis (Axis = X, Y or Z) in the negative and positive senses (Sense = – or +). G_S is the spacing between two consecutive points of the grid. Each point of the grid, \mathbf{M}_{ijk} , is defined as:

$$\mathbf{M}_{ijk} = \mathbf{C}_M + iG_S \mathbf{i} + jG_S \mathbf{j} + kG_S \mathbf{k}, \quad (8)$$

$$(i, j, k) \in \left([-N_X^-, N_X^+], [-N_Y^-, N_Y^+], [-N_Z^-, N_Z^+] \right),$$

where \mathbf{C}_M is the center of gravity of the initial precursor and (i,j,k) are the coordinates of \mathbf{M}_{ijk} in the orthonormal coordinate system ($\mathbf{i}, \mathbf{j}, \mathbf{k}$). The fields, ϕ_e , ϕ_r and ϕ_d , at the grid points are calculated using:

$$\phi_e(\mathbf{M}_{ijk}) = \sum_i^{\text{protein}} \frac{q_i}{|\mathbf{r}_i - \mathbf{M}_{ijk}|} \quad (9)$$

$$\phi_r(\mathbf{M}_{ijk}) = \sum_i^{\text{protein}} 2\sqrt{\epsilon_i} \left(\frac{\sqrt{\sigma_i}}{|\mathbf{r}_i - \mathbf{M}_{ijk}|} \right)^{12} \quad (10)$$

$$\phi_d(\mathbf{M}_{ijk}) = - \sum_i^{\text{protein}} 2\sqrt{\epsilon_i} \left(\frac{\sqrt{\sigma_i}}{|\mathbf{r}_i - \mathbf{M}_{ijk}|} \right)^6 \quad (11)$$

where q_i , ϵ_i and σ_i are the OPLS non-bonding energy parameters for the protein atoms and \mathbf{r}_i is the position of the protein atom.

Once the fields at the grid points are known, the fields $\tilde{\phi}(\mathbf{r})$ at any other point lying within the grid, such as the positions of the ligand atoms, are obtained by linear interpolation:

$$\begin{aligned} \tilde{\phi}(\mathbf{r}) = & \sum_{l,m,n=0}^1 [(1-l) + \text{sign}(l-0.5)\delta_l] [(1-m) + \\ & \text{sign}(m-0.5)\delta_m] [(1-n) + \text{sign}(n-0.5)\delta_n] \\ & \phi(\mathbf{M}_{u+l,v+m,w+n}) \end{aligned} \quad (12)$$

where (u, v, w) are the indices of the grid point at the lower corner of the cube surrounding the point, \mathbf{r} , and δ_l , δ_m and δ_n are the fractional distances of the same point along the grid axes in the cube. Several programs, including the solver for the Poisson-Boltzmann equation implemented in the program UHBD, use a similar methodology of computation and storage of fields on a grid [29].

To encode the geometrical information in a chromosomal representation suitable for the genetic al-

Table 2. Atom types for ligand atoms and their Lennard-Jones parameters taken from the OPLS-AA force field [24].

Atom type	Description	σ (Å)	ϵ (kcal mol ⁻¹)
C1	C sp1	3.750	0.070
C2	C sp2	3.550	0.070
C3	C sp3	3.500	0.066
CR	C aromatic	3.550	0.070
N1	N nitrile	3.250	0.170
N21	N protonated imine	3.250	0.170
N22	N imine	3.250	0.170
N23	N peptide bond	3.250	0.170
N31	N amine	3.250	0.170
N32	N protonated amine	3.250	0.170
NR	N aromatic	3.250	0.170
O	O carbonyl	2.960	0.210
OC	O carboxylate	2.960	0.210
OA	O ether	3.120	0.170
OR	O aromatic	3.120	0.170
S2	S sulphenyl	3.550	0.250
S31	S sulfonium	3.550	0.250
S41	S sulfone	3.550	0.250
S42	S sulfide	3.550	0.250
SR	S aromatic	3.550	0.250
P31	P phosphate	3.740	0.200
P4	P phosphonium	3.740	0.200
F	Fluorine	2.733	0.720
CL	Chlorine	4.417	0.118
BR	Bromine	4.624	0.090
HV1	H non polar linked to C1	2.420	0.300
HV2	H non polar linked to C2	2.420	0.030
HV3	H non polar linked to C3	2.500	0.030
H	H polar	0.000	0.000

gorithm, $6 + n$ genes were used corresponding to the 6 rotational and translational degrees of freedom and the n rotatable bonds of the ligand. Each gene was formed by 10 binary numbers, thus ensuring a minimal increment of 0.002 Å for the translational degrees of freedom and 0.35° for the rotational and torsional ones. We implemented a number of genetic operators in our algorithm (see reference 13) but for the optimization we use only four with the following percentages – reproduction (50%), cross-over (90%), mutation (10%) and rejection (10%). Each optimization is normally performed with 20 chromosomes for 1000 generations.

Scoring the new molecule

After the conformational search of the ligand, the algorithm keeps the best conformer and evaluates it with a scoring function that we built up following a QSAR approach. Table 3 gives a full list of the descriptors that we have currently implemented and can be used for scoring a molecule. Full details about the building and testing of the scoring function are described below.

Results

In this section, we first present the results of tests of some of the protocols that were described in the previous section. These checks are fundamental to a

Table 3. Scoring function descriptors, and their identifying indices, determined by the algorithm.

Descriptor Index	Descriptor
1	Conformational search averaged electrostatic receptor-ligand interaction energy
2	Conformational search averaged Lennard-Jones receptor-ligand interaction energy
3	Conformational search averaged ligand internal energy
4	Ligand solvation energy using the Generalized-Born/surface area approach developed by Pellegrini and Field [42]
5	Number of flexible bonds
6	Ligand's charge
7	Ligand's number of atoms
8	Ligand's mass
9	Protein's charge
10	Protein's number of atoms
11	Solvent accessible surface (SAS) area of polar atoms in the receptor-ligand complex
12	SAS of non-polar atoms in the receptor-ligand complex
13	SAS of polar atoms in the receptor
14	SAS of non-polar atoms in the protein
15	SAS of polar atoms in the ligand
16	SAS of non-polar atoms in the ligand
17	Δ SAS of polar atoms between the receptor-ligand complex and the separated ligand and receptor
18	Δ SAS of non-polar atoms between the receptor-ligand complex and the separated ligand and receptor
19	Ligand's number of peptide bonds
20	Ligand's number of rings
21	Hydrogen-bond score – receptor-ligand charged atoms/charged atoms using the approach of Eldridge et al. [37]
22	Hydrogen-bond score – receptor-ligand charged atoms/non-charged atoms using the approach of Eldridge et al. [37]
23	Hydrogen-bond score – receptor-ligand non-charged atoms/non-charged atoms using the approach of Eldridge et al. [37]
24	Ligand's Log P using the approach of Ghose et al. [43]
25	Sum of the charges of the residues in a sphere of 5 Å surrounding the initial precursor
26	Ligand's number of chiral atoms

Table 4. Summary of the different partial charges calculations performed on the 568 molecules generated by the algorithm and their associated simplified molecules.

Method type	Methodology	Partial charges calculations
<i>Ab initio</i>	Hartree-Fock 6-31G** basis set	Mulliken
<i>Ab initio</i>	Hartree-Fock 6-31G** basis set	ESP [32]
Semi-empirical	AM1	Function 1 Mulliken
Semi-empirical	AM1	Function 1 Class IV
Semi-empirical	AM1	Function 2 Mulliken
Semi-empirical	AM1	Function 2 Class IV
Semi-empirical	AM1	Function 3 Mulliken
Semi-empirical	AM1	Function 3 Class IV

full assessment of the functioning of the algorithm. The section terminates with an initial application of the algorithm to a protein, the neuraminidase of the influenza virus.

Tests of the main parts of the algorithm

Validation of the local optimization procedure

Local geometry optimization of the new precursor molecule is necessary so as to ensure that the new fragment and the extra added atoms have a chemically reasonable geometry. The conjugate gradient optimization method that we use is able to find a minimum energy structure with a small number of iterations and it is less time-consuming than methods that require second derivatives as the size of the ligand molecule increases. To validate the simplified potential of Equation 2, we selected a set of 92 small organic molecules of various types and designated this the reference set. The structures for these molecules were taken from the Molecular Models website at Okanagan University [30]. Starting from these structures, we optimized them using an *ab initio* DFT method with the B3LYP functional and the 6-31G** basis set. These optimizations were performed with the JAGUAR program [31] and gave a set of structures that we refer to as the *ab initio* set. A third set of structures was obtained by starting from the reference structures and optimizing using the simplified potential. These we denote the local set.

To compare the structures, we calculated the RMS coordinate deviation (RMSD) between identical molecules of the different sets. These results are shown in Figure 5. For most of the molecules we obtain a good correlation between the different sets. The exceptions are molecules 12, 17, 29, 63, 68, 69 and 81 which correspond to bicyclo[1.1.0]pentane, cyclamate, penta-1,4-diene, sarin, parathion, nicotinic acid and thioanisole, respectively. Apart from molecule 12, the differences are due to differences in the conformations of various sidechains. We were not too concerned with these differences as sidechain flexibility is explored fully by the conformational search. It should also be noted that our simplified potential does not have an explicit dihedral potential as we did not consider this crucial in this initial work, although it may be added in later versions of the program. In the case of molecule 12, bicyclo[1.1.0]pentane, the structural differences come from the angle formed by the two fused rings of the molecule. The *ab initio* calculations predict a smaller angle than the simplified

Table 5. Descriptor weights and statistical parameters for the best model obtained for the neuraminidase scoring function.

Descriptor	Weight
σ_0	5.61118 kcal mol ⁻¹
1	0.00842
2	0.02461
4	-0.00233
18	0.52927
22	-0.03349
23	-0.08627
Statistical parameter	Value
R^2	0.602
Q^2	0.449
R_p^2	0.816
Predictive R^2	0.517
$ b $	0.988
P	0.001
\mathcal{F}	3.052

potential (99° as opposed to 110°). This is probably due to the difficulty of the UFF force field in distinguishing between a tetravalent carbon in an aliphatic chain and in a double ring system and to the lack of non-bonding interactions between atoms in the simplified potential, although these are present in the full optimization performed later.

Validation of the method for updating of the partial charges

After the local optimization, the algorithm determines the partial charges for the new molecule. As discussed above, the charges are calculated for a truncated molecule that contains the mutated atoms and then transposed back to the full precursor using an interpolation scheme. It should be emphasized that interpolation is only applied to a small set of atoms around the transformation site, which means that atoms of the old precursor far from the transformation site keep their original partial charges and that atoms of the new molecule far from the transformation site are assigned the charges from the simplified molecule. Which atom charges are to be interpolated in this way is defined separately for each different type of transformation but usually includes the transformed atom itself and those atoms separated from it by one or two bonds (1–2 and 1–3 interactions).

We tried three interpolation schemes:

- Function 1: the partial charge of an atom in the new molecule is related to the charge of its analogue in the simplified molecule by a scaling factor.
- Function 2: the partial charge of an atom in the new molecule is related to the charge of its analogue in the simplified molecule by an additive factor.
- Function 3: the partial charge of an atom in the new molecule is related to the charges of its analogue in both the precursor and the simplified molecule by a scaling factor. In the case that there is no equivalent in the precursor, the old charge is taken to be zero.

For each of the functions, the expressions for the additive or scaling factors were devised in such a way as to ensure that the total charge on the new molecule was correct.

To test the three functions, we generated 568 molecules with our algorithm. Most of the physico-chemical filters were switched off during this step so as to have a reasonably diverse set of molecules that could serve as a good test of the robustness of our method. Molecules deemed to have structures that were too complex were rejected as were molecules that were too similar to existing ones in the set. For each molecule, and its associated simplified molecule, we performed the various partial charge calculations that are summarized in Table 4. The *ab initio* calculations were done with JAGUAR and consisted of the determination of both Mulliken charges and charges determined using electrostatic potential (ESP) fitting [32]. The semi-empirical calculations were done with DYNAMO.

To evaluate the charges, we compared the charges calculated with semi-empirical and *ab initio* methods. For the simplified molecules the agreement is reasonable with RMS deviations in the range 0.1 to 0.3 *e*. For the full molecules, functions 1 and 3 gave poor results especially when there were heavy elements, such as phosphorus or sulphur. In comparison, function 2 gave charges with RMS deviations comparable to those observed for the simplified models. We thus chose this function for our algorithm. To be explicit, the charge of an atom that is to be interpolated is equal to the charge in the simplified model plus a correction factor, δ , which has the form:

$$\delta = \frac{Q_T - Q_F - Q_M}{N_M}. \quad (13)$$

Here N_M is the number of atoms whose charge is being changed, Q_T is the total charge of the new molecule, Q_F is the total charge of the atoms in the molecule whose charge is not being changed and Q_M is the total

charge (from the simplified molecule) of the atoms whose charge is being changed.

Optimization of the electrostatic and Lennard-Jones potential grids

After generation of a new molecule and its parametrization, the algorithm starts the conformational search procedure. To make this as fast as possible, so as to explore as many configurations as possible, the interaction of the protein and ligand is calculated using electrostatic, repulsive and dispersive fields that are precalculated for the protein atoms. To check the validity of using a grid in this way, we compared protein-ligand interaction energies calculated using the grid method against energies calculated with the exact pairwise formula (see, for example, Equation 3). The grid energy, \mathcal{E}_I , is:

$$\mathcal{E}_I = E_{\text{el}} + E_{\text{LJ}}, \quad (14)$$

where E_{el} and E_{LJ} are defined in Equation 4. Different grid energies were determined using different values of the grid spacing G_S and different translations and rotations of the grid. We considered translations along the three cartesian axes with increments of $G_S/10$ from 0 to $9G_S/10$ and rotations about the three cartesian axes from 0° to 85° with increments of 5° .

The system chosen for the study was the structure of neuraminidase 1NNC complexed with the 4-guanidino-Neu5Ac2En inhibitor [33]. The results for rotation and translation about the different coordinate axes exhibit the same trends and so we show, in Figure 6, the variation in \mathcal{E}_I for rotation about the y-axis only. As can be seen, the sensitivity of \mathcal{E}_I to rotation strongly depends upon the value of G_S . In general, the larger the value of G_S , the larger the deviation between the grid-calculated \mathcal{E}_I and the asymptotical, reference pairwise potential. This is due to the fact that linear interpolation is less accurate at large grid spacings particularly for the repulsive and, to a lesser extent, the dispersive potentials which can vary rapidly with distance. Starting at values of G_S of 0.4 Å, however, \mathcal{E}_I approaches the pairwise value. In all our applications, we usually work with a grid spacing of 0.15 Å which is a good compromise between accuracy and computational and memory cost.

Development of a scoring function for the test protein

To apply our algorithm to the protein neuraminidase, we developed a scoring function using a multiple linear regression analysis and a leave-one-out (LOO) procedure. As reference data, we chose 59 inhibitors

of the protein for which experimental free energies of binding are known and which are illustrated in Figure 7. The inhibitors can be divided into two groups – ones that contain a hydrophobic group at the C6 position and that trigger a conformational change in two key residues, R224 and E276, and others that contain a hydrophilic group at the C6 position and that do not trigger such a transformation [34]. For each group, we chose a reference complex. These were the PDB structures 1BJI for the hydrophobic group [35] (complexes 25–47 and 59 in Figure 7) and 1NNC for the hydrophilic group (1–24, 48–58 in Figure 7). The two reference complexes were fully geometry optimized using a conjugate gradient algorithm treating the protein with the OPLS-AA MM potential and the inhibitor with the semi-empirical AM1 method. Then, for each group, the structures of the remaining inhibitors were constructed from the reference structures using the program Builder implemented in the INSIGHTII suite of programs [36]. Finally, the structure of each complex was minimized using the same conditions as before but keeping the coordinates of the protein fixed.

After several attempts, we finally fixed upon a method based upon global optimization to obtain a good model for a scoring function. The approach was as follows. We employed a genetic algorithm to partition the set of 59 complexes into a set of 45 complexes to form a training set, 12 to form an external set and two that were considered as outliers. This meant that the chromosomes used by the genetic algorithm consisted of 45 numbers identifying the indices of the training set complexes followed by 12 numbers identifying the complexes of the external set. The choice of a fitness function posed some problems but, again after several trials, we chose the following form which permitted the optimization of the various statistical parameters in a uniform way:

$$\mathcal{F} = e^{-10R^2} + e^{-Q^2} + e^{-R_p^2} + e^{-Q_p^2} + e^{-P} + e^{|b|} \quad (15)$$

Here R^2 and R_p^2 are the Pearson coefficient for the training and external sets, Q^2 and Q_p^2 are the LOO cross-validated r^2 and predictive r^2 for the training and external sets, and P and b are the regression coefficient and constant of linear regression between the experimental free energies of binding for the inhibitors of the external set and the free energy of binding predicted by our scoring function built on the training set of 45 inhibitors. This latter function has the following form:

$$\sigma = \sigma_0 + \sum_i^{\text{Descriptors}} w_i \mathcal{D}_i \quad (16)$$

where \mathcal{D}_i are the descriptors 1, 2, 4, 18, 22, 23 from Table 3 chosen for this study and w_i are their associated (optimized) weights. The formulation of Equation 15 comes from the fact that we found that it was a good way to optimize simultaneously the scoring function in the interpolative range with the training set and in the extrapolative range with the external set. To optimize \mathcal{F} , we used the genetic algorithm we developed with the following operators and parameters: reproduction (40%), mutation (20%), cross-over (70%), inversion (10%), rejection (10%), death rate (2000), death limit 2, warp rate (40) with full warp, 40000 generations with 100 chromosomes per generation. For further details about the death and warp operators see [13].

The performance of our best model for the training and the external sets, derived using the above approach, is presented in Table 5 and in Figure 8 and a comparison with existing scoring functions in Table 6. The latter shows that our scoring function is less accurate than the others although the values of R^2 and Q^2 remain in the limits usually deemed acceptable for a linear regression model. There are a number of explanations for this. For example, it is possible that the choice and variety of descriptors in our algorithm were not sufficient to build an accurate function. However, we have also noticed that other scoring functions [37, 38] were less effective when used to predict the affinity of inhibitors of only a single protein. This is perhaps because it may be more difficult for a scoring function to capture the small differences between similar molecules that produce large changes in activity.

Application to neuraminidase

Neuraminidase and hemagglutinin are the two major surface glycoproteins of the influenza virus. Neuraminidase catalyses the hydrolysis of terminal sialic acid residues present on the cell surface and is critical for viral release from the infected cell and transport through the mucus in the respiratory tract. As such, it represents an important target for the search for active compounds against influenza, which is still a major cause of death around the world.

For our tests, we chose the PDB structure of neuraminidase 1NNC [33] and, for the precursor, the neuraminidase inhibitor Bana 105 (inhibitor 48 in Figure 7) [39]. We did not start from the original ligand

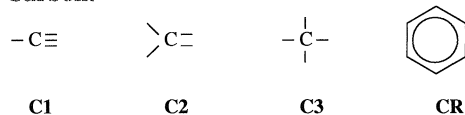
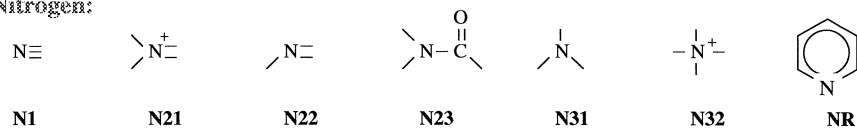
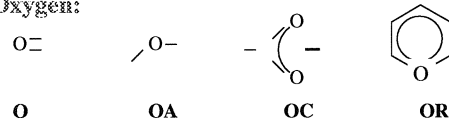
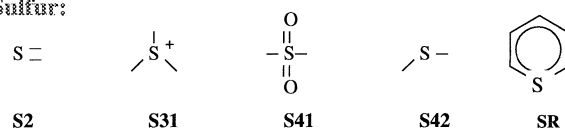
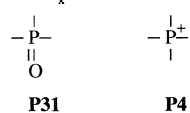
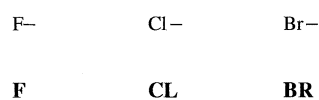
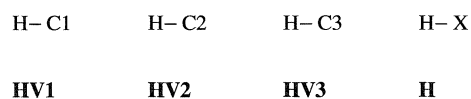
Carbon:**Nitrogen:****Oxygen:****Sulfur:****Phosphorus:****Fluorine, chlorine, bromine:****Hydrogen:**

Figure 2. The atom types used for the description of the ligand by the drug-design algorithm.

Table 6. Comparison between the scoring function developed for our study and existing ones. References: VALIDATE [41]; Böhm's score, values taken from Ref. 44; Eldridge's score [37]; BLEEP [38].

Scoring function	Method	Procedure	Training set	Components	R^2	Q^2	Test set	Predictive R^2
VALIDATE	PLS	LOO	51	6	0.85	0.78	(14,13,6)	(0.81,0.57,0.72)
VALIDATE	NN	LOO	51	6	0.81	0.77	(14,13,6)	(0.69,0.53,0.70)
Böhm's score	MLRA	—	39	4	0.69	—	—	—
Eldridge's score	MLRA	LOO	82	4	0.71	0.66	(20,10)	(0.63,-3.0)
BLEEP	—	—	90	—	0.74	—	—	—
σ	MLRA	LOO	45	6	0.60	0.45	12	0.52

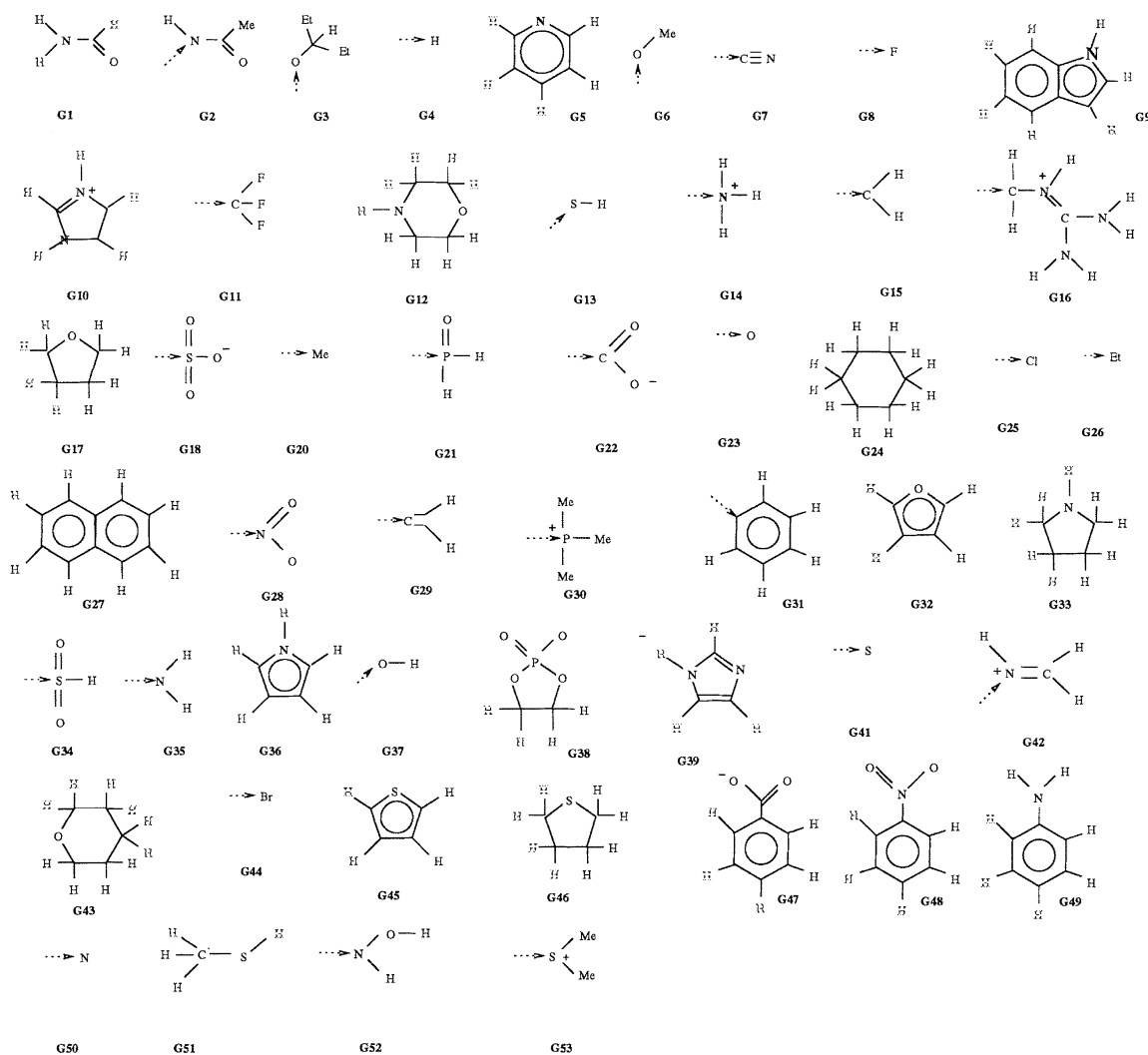


Figure 3. List of the fragments available in the database of groups to be added to non-cyclic groups.

of the 1NNC complex (inhibitor 3 in Figure 7) because it is already a good inhibitor and we thought that this could bias the evolution of the algorithm. We did two simulations, in each of which 30 000 different molecules were generated and screened. In the first simulation, the aromatic ring atoms of the precursor were not allowed to be changed and the conformational search was performed by optimizing the rotatable torsional degrees of freedom of the ligand only. In the second simulation, all atoms were defined as mutable and the conformational search was performed with both rotatable torsional and orientational degrees of freedom. The calculations were carried out in this way so as to illustrate different aspects of the algorithm's performance. The first simulation aimed to

refine the structure of an inhibitor that is already well-positioned in the active site of neuraminidase as the precursor molecule, Bana 105, was superimposed onto the original ligand of the 1NNC complex. The second simulation is more flexible and requires searching both the chemical and the geometrical landscapes of the ligand in the active site. Each search consisted of 10 runs of the genetic algorithm so as to avoid the problem of premature convergence. This method has also been used by other workers [4]. The initial charges of the precursor were ESP charges calculated with the program JAGUAR using a Hartree-Fock method and a 6-31 G** basis set [31]. All calculations were done on a Pentium III 1GHz computer running Linux and each simulation took approximately 3 days. The great

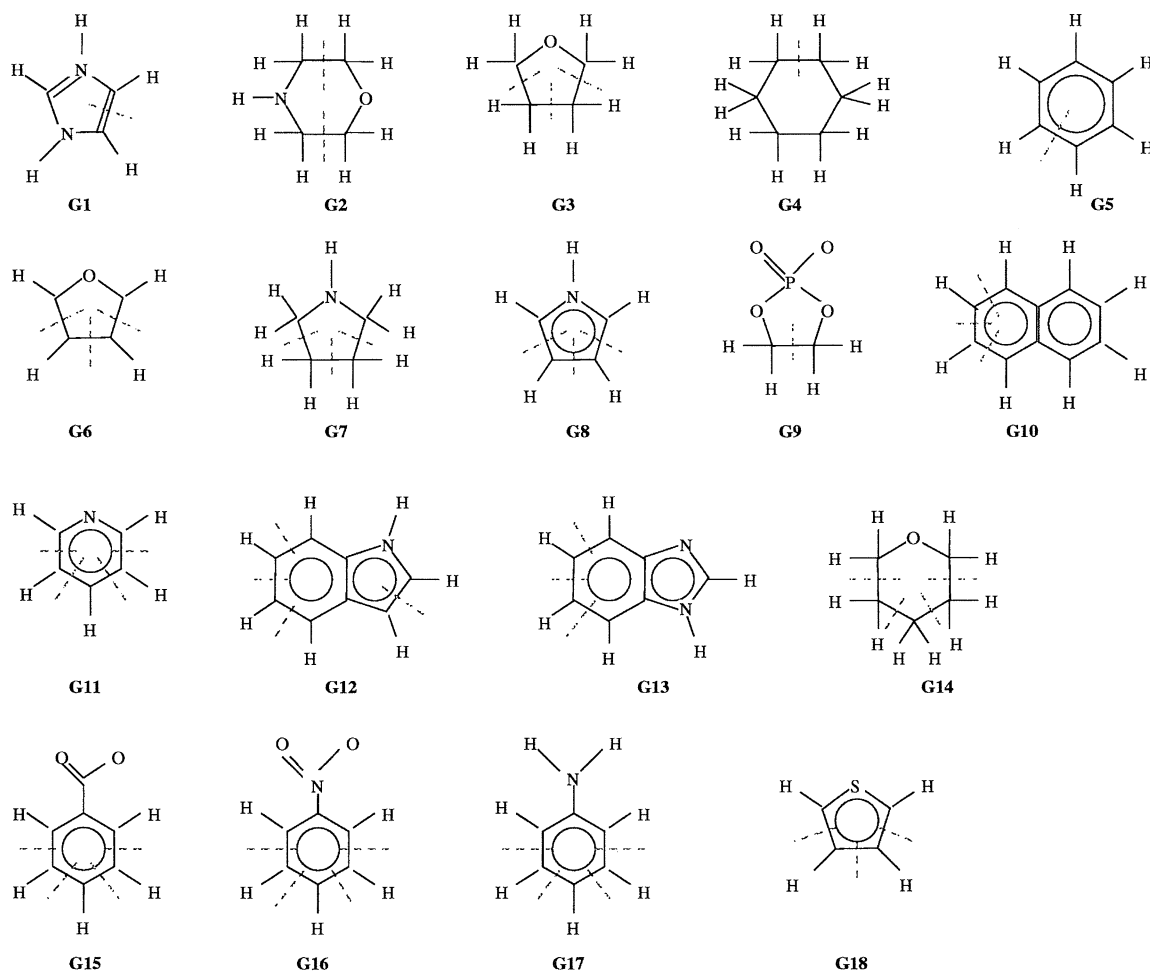


Figure 4. List of the fragments available in the database of cyclic groups.

majority of the computational time was taken by the conformational searches.

The evolution of the value of the scoring function for the 'best' proposed ligand versus the number of steps of the algorithm for the two simulations is shown in Figure 9. The algorithm optimizes satisfactorily the scoring function and generates molecules with values of the scoring function close to those of the best proposed molecules, such as zanamivir and oseltamivir, from Figure 7. A detailed study of the different molecules generated during each simulation showed the algorithm's ability to propose a wealth of molecules of various types and to give chemically reasonable structures, thus validating our methodologies of molecule generation and structure optimization.

Figure 10 shows the top five scored molecules, which we denote T1 and T2, resulting from each of the simulations. The molecules within each group

show a strong degree of structural similarity and both sets are reasonably complex, indicating that the algorithm was able to explore efficiently the physico-chemical landscape of the neuraminidase active site (see Figure 11). Both sets of molecules have retained carboxylate groups that interact with the arginine pocket made up of the residues R118, R292 and R371. These residues are clearly fundamental for the stabilization of molecular species in the active site as all the known inhibitors have a negatively charged group, either carboxylate or phosphate, in the C1 position. The stabilization is mainly due to electrostatic and hydrogen-bond interactions, two descriptors that appear explicitly in our scoring function. In addition, the T1 molecules have a hydrophobic region near the C5 lateral chain, which is also a required feature for a neuraminidase inhibitor as there is a hydrophobic pocket made up of the residues W178, I222 and part

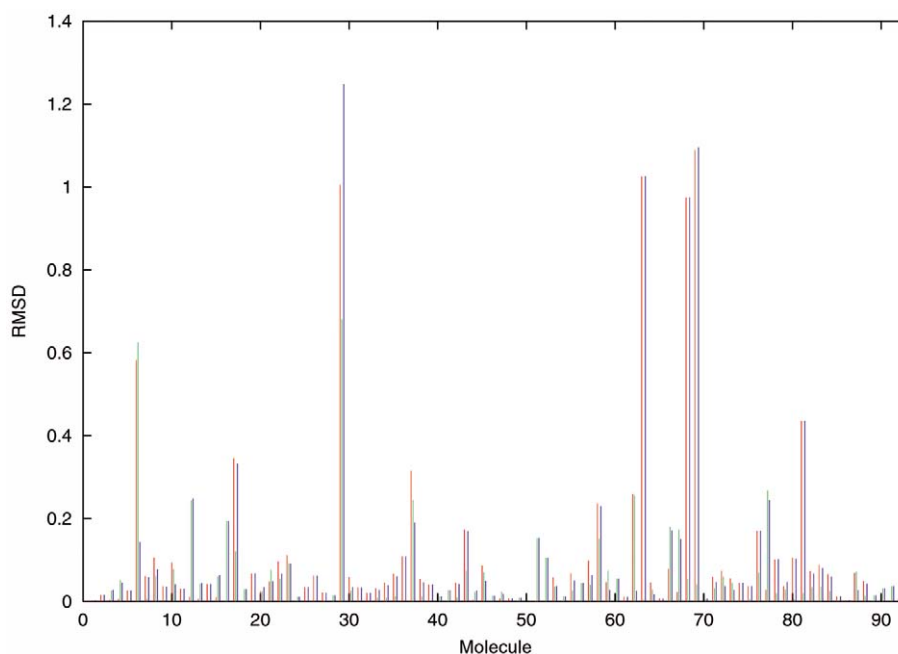


Figure 5. Barplot of the RMS coordinate deviation (Å) for each of the 92 molecules selected for the test of the local optimization method. Each color represents the variation of one set of structures with respect to another. Red, *ab initio*/reference set; green, reference/local set; blue, *ab initio*/local set.

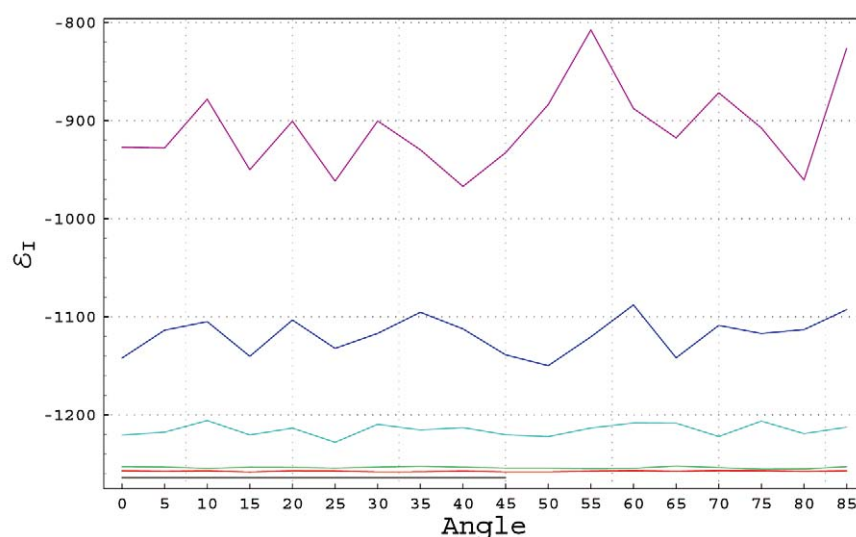
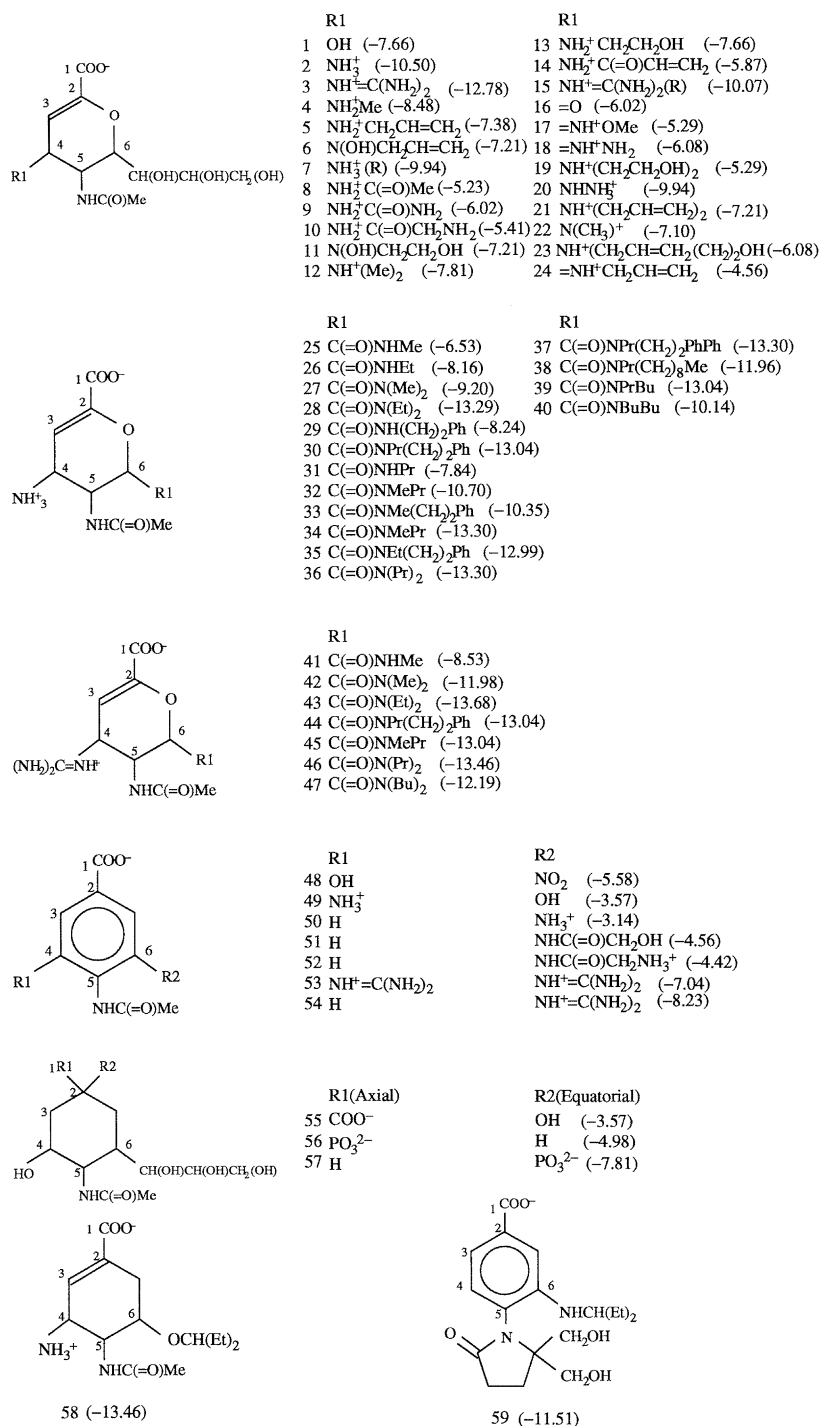


Figure 6. Plot showing the variation of ϵ_I (kJ mol^{-1}) versus the angle of rotation ($^\circ$) of the grid about the y-axis. Black, reference pairwise value; purple, $G_S = 0.8$ Å; dark blue, $G_S = 0.6$ Å; light blue, $G_S = 0.4$ Å; green, $G_S = 0.2$ Å; red, $G_S = 0.15$ Å.

of the sidechain of R224. The T1 molecules also retain a hydroxyl group in the C4 position which hydrogen bonds with the residue D151. As well as these conserved interactions for the T1 set, the algorithm shows its ability to establish new interactions by creating two new hydrogen bonds with residues R152 and E276.

The T2 molecules had less specific interactions with the protein than those of the T1 set except for a strong hydrogen bond with residue R152. The lower score of the molecules compared to the starting precursor, however, was due to a more stabilizing protein-ligand non-bonding interaction energy resulting from the op-



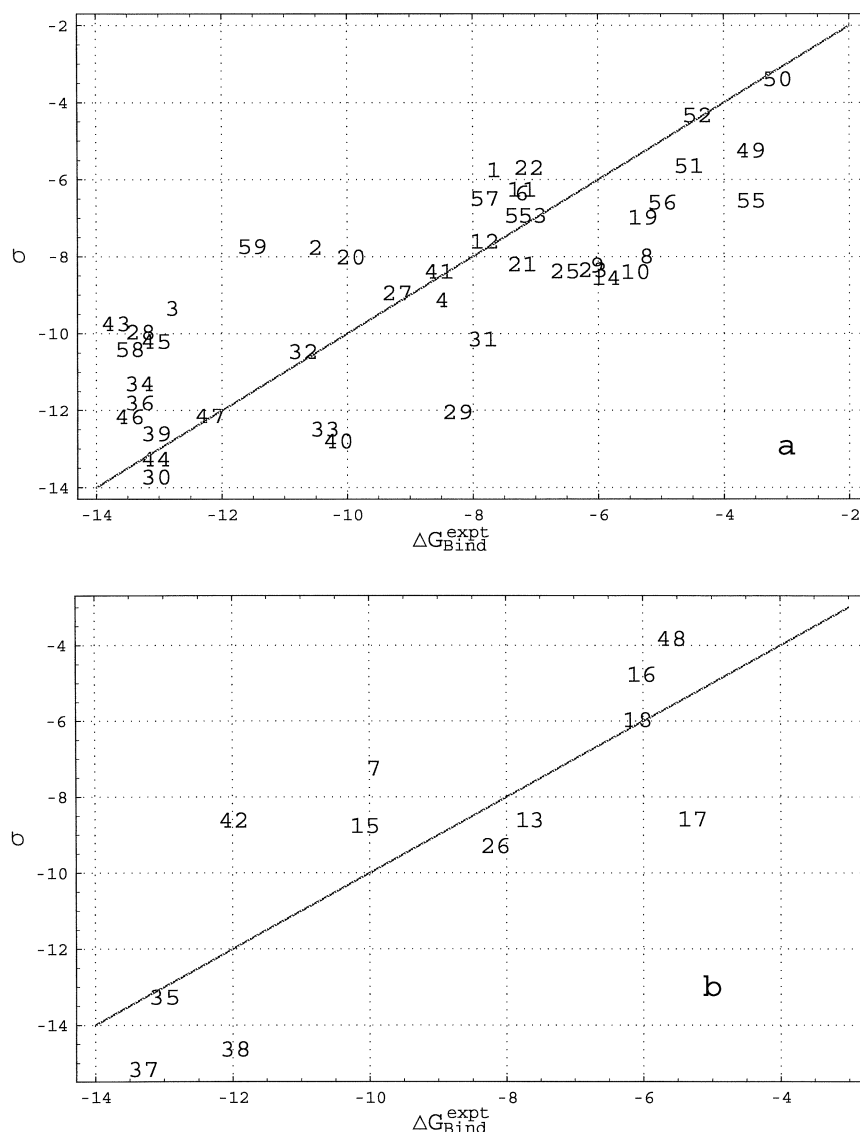


Figure 8. Plot of free energy of binding (kcal mol^{-1}) predicted with our scoring function σ versus the experimental values. (a) The training set; (b) the external set. Numbers correspond to the inhibitor numbers of Figure 9.

timization of the position of the ligand in the active site and the fact that the T2 molecules have a negative charge. Finally, we note that although the aromatic ring was mutable in the second simulation, the final putative ligands retained this central cyclic group, suggesting that this is an important feature for potential neuraminidase inhibitors.

Drawbacks of the method

Although we consider the algorithm to have been promising in its initial application to neuraminidase,

our algorithm suffers from several drawbacks which we hope to tackle in the future. First, the computation time is quite long compared to other programs in its class. This is primarily due to the conformational search using the genetic algorithm, but the partial-charge calculation also contributes. The former calculation could be speeded up trivially by implementing a parallel version of the program, whereas the latter could be alleviated by switching to an empirical method for generating partial charges.

Second, the preliminary studies on neuraminidase show that after exploring several putative directions

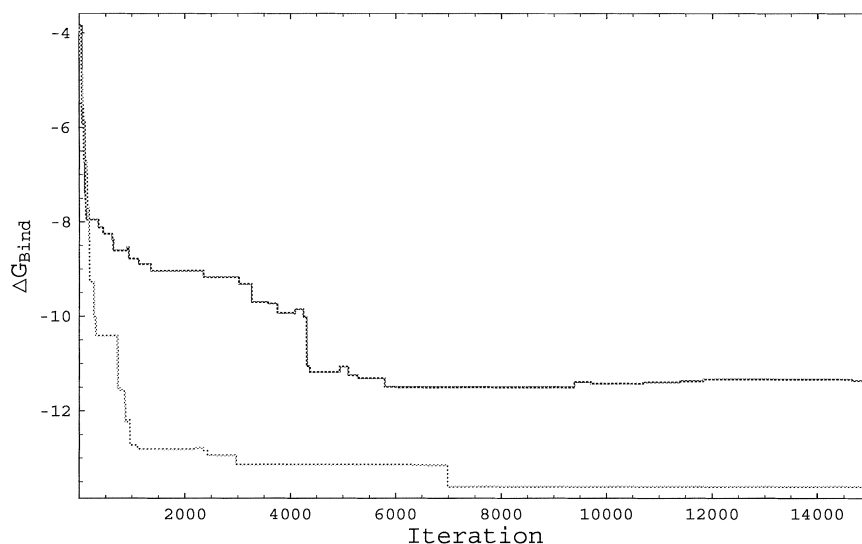


Figure 9. Plot showing the evolution of the value of the scoring function (kcal mol^{-1}) of the best molecule as a function of iteration number of the algorithm. Black, simulation 1; gray, simulation 2.

in structure space, the algorithm tends to focus on a particular class of structures after several hundred iterations (such as the sets T1 and T2). In the case of neuraminidase, this could be explained by supposing that the presence of strong pharmacophoric sites, such as the arginine triad, tends to limit the structural diversity of the ligands that are proposed by the algorithm. Alternatively, it could be a problem of premature convergence that occurs frequently for global optimization methods [40]. As a result, it will be important to investigate ways of circumventing this problem, such as by modifying the Monte Carlo procedure or by introducing a more diverse set of fragments (including larger groups and heterocycles) into the fragment databases.

Third, the scoring function that we built to test the ability of our algorithm to generate putative ligands for neuraminidase will need to be improved for applications of the algorithm to other protein targets. This will be done either with an enhanced version of the scoring function described above or by the introduction of existing scoring functions [6, 37, 41]. This should be straightforward given the modular architecture of our program. Indeed, it is likely that a range of different scoring functions and other scoring approaches will be necessary if a wide variety of problems are to be treated in a robust fashion.

Conclusions

The principal aim of this paper has been to give a methodological overview of a *de novo* drug-design algorithm that we have developed. The algorithm is based on a sequential growth methodology and works by replacing fragments of a precursor by fragments randomly chosen from predefined databases of chemical groups. Although our program adopts features of existing drug-design software, we consider the following aspects to be among those that are either novel or noteworthy: (i) the program is coupled with the DYNAMO molecular modeling library and makes use of many of its capabilities including geometry optimization and QM partial-charge calculations. It also means that future developments of the algorithm that require, for example, molecular dynamics to treat protein or ligand flexibility can be straightforwardly incorporated; (ii) the use of QM calculations on simplified molecules to update the charges on ligand atoms; (iii) the local and global optimization of a ligand's geometry before the conformational search; (iv) the procedure employed for building new ligands including the modification of an atom's valence when chemical groups are added or removed; and (v) the introduction of a generalized-Born/surface area term for the determination of solvation-related descriptors in the scoring function.

Our discussion focused upon the main tools used by the algorithm to build, optimize, characterize and

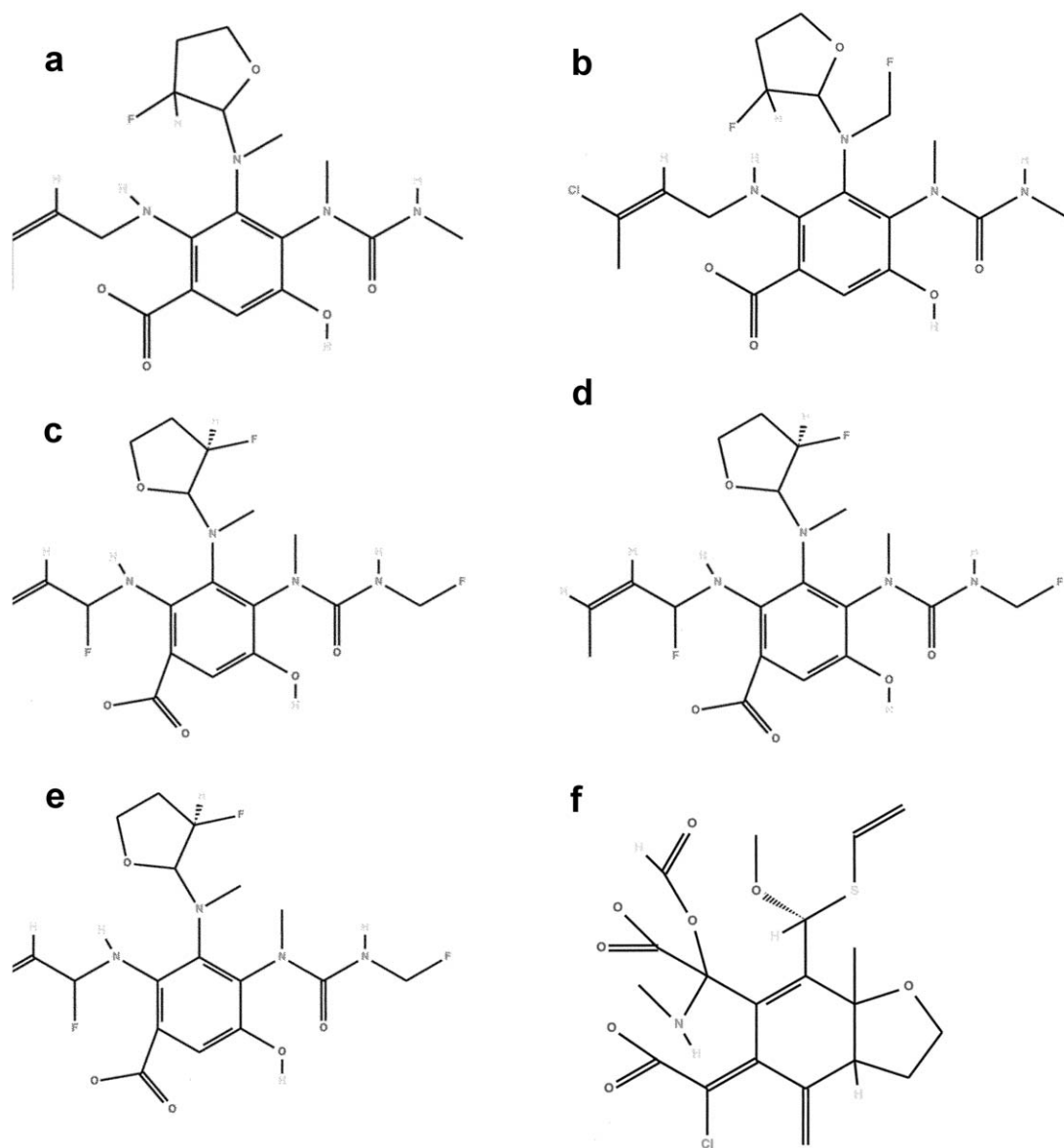


Figure 10. Structures of the top 10 scored molecules found by simulations 1 (from a to e) and 2 (from f to j).

evaluate molecules as potential ligands. We described, in particular, the local optimization procedure, the partial-charge calculation, the procedure for the fast calculation of protein-ligand non-bonding interaction energies and the scoring function for the protein that we chose as a first application of our algorithm, the neuraminidase of the influenza virus. The application to neuraminidase illustrated the ability of our algorithm to rapidly propose and evaluate a wide

range of ligand molecules and an examination of these compounds showed that they had some of the main features required for a molecule to be a good neuraminidase inhibitor.

Clearly our algorithm can be improved in several respects but our initial implementation should provide a solid foundation for future work. We intend, in particular, to explore in more detail the use of the algorithm for refining existing inhibitors, the use of

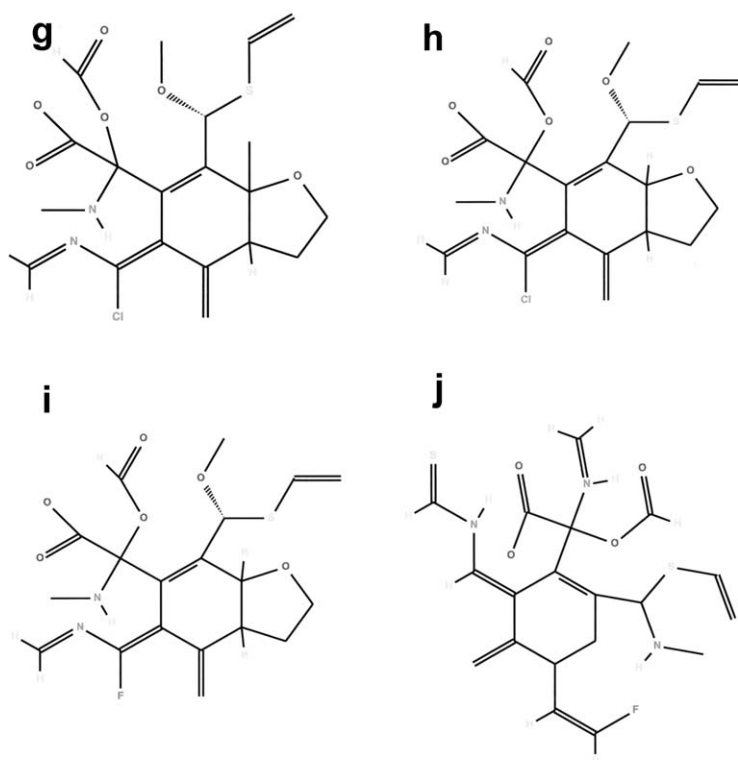


Figure 10. Continued.

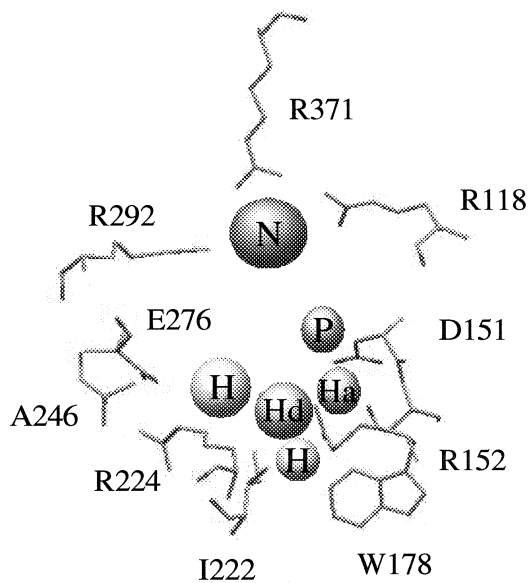


Figure 11. Main residues of the neuraminidase with their associated pharmacophoric sites. N, negative charge; P, positive charge; Ha, hydrogen-bond acceptor; Hd, hydrogen-bond donor; H: hydrophobic group.

external scoring functions, either individually or in the context of a consensus scoring function, the addition of new physico-chemical filters and the inclusion of receptor flexibility and the handling of active-site waters. We are also continuing to apply the current algorithm to neuraminidase and a number of other proteins and will present a critical examination of the results of these applications in due course.

Acknowledgements

The authors would like to thank the Institut de Biologie Structurale – Jean-Pierre Ebel, the Commissariat à l’Energie Atomique (CEA) and the Centre National de Recherche Scientifique (CNRS) for support of this work.

References

- Whittle, P.J. and Blundell, T.L., *Annu. Rev. Biophys. Biomol. Struct.*, 23 (1994) 349.
- Kuntz, I.D., Blaney, J.M., Oatley, S.J., Langridge, R. and Ferrin, T.E., *J. Mol. Biol.*, 161 (1982) 269.
- Rarey, M., Kramer, B., Lengauer, T. and Klebe, G., *J. Mol. Biol.*, 261 (1996) 470.
- Jones, G., Willett, P., Glen, R.C., Leach, A.R. and Taylor, R., *J. Mol. Biol.*, 267 (1997) 727.
- Morris, G.M., Goodsell, D.S., Halliday, R.S., Huey, R., Hart, W.E., Belew, R.K. and Olson, A.J., *J. Comput. Chem.*, 19 (1990) 1639.
- Böhm, H.J., *J. Comput.-Aided Mol. Des.*, 6 (1992) 593.
- Westhead, D.R., Clark, D.E., Frenkel, D., Li, J., Murray, C.W., Robson, B. and Waszkowycz, B., *J. Comput.-Aided Mol. Des.*, 9 (1995) 139.
- Moon, J.B. and Howe, W.J., *Proteins*, 11 (1991) 314.
- Gillet, V., Newell, W., Mata, P., Myatt, G., Sike, S., Zsoldos, Z. and Johnson, A.P., *J. Chem. Inf. Comput. Sci.*, 34 (1994) 207.
- Field, M.J., *A Practical Introduction to the Simulation of Molecular Systems*. Cambridge University Press, Cambridge, UK, 1999.
- Field, M.J., Albe, M., Bret, C., Proust-De Martin, F. and Thomas, A., *J. Comput. Chem.*, 21 (2000) 1088.
- Laver, W.G., Bischofberger, N. and Webster, R.G., *Sci. Am.*, 280 (1999) 78.
- Pellegrini, E., *Elaboration d’un Algorithme de Drug Design: Aspects Fondamentaux et Application à une Protéine*. Ph.D. Thesis, Université Joseph Fourier, Grenoble, France, 2002.
- Berman, H.M., Battistuz, T., Bhat, T.N., Bluhm, W.F., Bourne, P.E., Burkhardt, K., Feng, Z., Gilliland, G.L., Iype, L., Jain, S., Fagan, P., Marvin, J., Padilla, D., Ravichandran, V., Schneider, B., Thanki, N., Weissig, H., Westbrook, J.D. and Zardecki, C., *Acta Crystallogr. D. Biol. Crystallogr.*, 58 (2002) 899.
- Lewell X.-Q., Judd D.B., Watson S.P., and Hann M.M., *J. Chem. Inf. Comput. Sci.*, 38 (1998) 511.
- Schneider G., Lee M.-L., Stahl M., and Schneider, P., *J. Comput.-Aided Mol. Des.*, 14 (2000) 487.
- Lipinski, C.A., Lombardo, F., Dominy, B.W. and Feeney, P.J., *Adv. Drug Deliv. Rev.*, 118 (1997) 3.
- Gasteiger, J., Rudolph, C. and Sadowski, J., *Tetrahedron Comp. Method.*, 3 (1990) 537.
- Rappé, A.K., Casewit, C.J., Colwell, K.S., Goddard III, W.A. and Skiff, W.M., *J. Am. Chem. Soc.*, 114 (1992) 10024.
- Storer, J.W., Giesen, D.J., Cramer, C.J. and Truhlar, D.G., *J. Comput.-Aided Mol. Des.*, 9 (1995) 87.
- Dewar, M.J.S. and Thiel, W., *J. Am. Chem. Soc.*, 99 (1977) 4899.
- Stewart, J.J.P., *J. Comput. Chem.*, 10 (1989) 209; *ibid* 10 (1989) 221.
- Dewar, M.J.S., Zoebisch, E.G., Healy, E.F. and Stewart, J.J.P., *J. Am. Chem. Soc.*, 107 (1985) 3902.
- Jorgensen, W.L., Maxwell, D.S. and Tirado-Rives, J., *J. Am. Chem. Soc.*, 118 (1996) 11225.
- Leach, A.R., *J. Mol. Biol.*, 235 (1994) 345.
- Carlson, H.A., *Curr. Opin. Chem. Biol.*, 6 (2002) 447.
- Holland, J.H., *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge, MA, 1992.
- Oshiro, C.M., Kuntz, I.D. and Dixon, J.S., *J. Comput.-Aided Mol. Des.*, 9 (1995) 113.
- Davis, M.E., Madura, J.D., Luty, B.A. and McCammon, J.A., *Comp. Phys. Comm.*, 62 (1991) 187.
- Woodcock, D. *Molecule Models Web Site* at <http://people.ouc.bc.ca/woodcock/molecule/molecule.html>, Okanagan University, British Columbia: 2002.
- Schrödinger Inc., Jaguar 3.5. Schrödinger Inc., Portland, OR, 1998.
- Williams, D.E., *Rev. Comp. Chem.*, 2 (1991) 219.
- Varghese, J.N., Epa, V.C. and Colman, P.M., *Protein Sci.*, 4 (1995) 1081.
- Smith P.W., Sollis S.L., Howes P.D., Cherry P.C., Cobley K.N., Taylor H., Whittington A.R., Scicinski J., Bethell R.C., Taylor N., Skarzynski T., Cleasby A., Singh O., Wonacott A., Varghese J. and Colman P., *Bioorg. Med. Chem. Lett.*, 6 (1996) 2931.
- Taylor, N.R., Cleasby, A., Singh, O., Skarzynski, T., Wonacott, A.J., Smith, P.W., Sollis, S.L., Howes, P.D., Cherry, P.C., Bethell, R., Colman, P. and Varghese, J., *J. Med. Chem.*, 41 (1998) 798.
- Molecular Simulations Inc., *Insight II. Molecular Simulations Inc.*, San Diego, CA, 1993.
- Eldridge, M.D., Murray, C.W., Auton, T.R., Paolini, G.V., and Mee, R.P., *J. Comput.-Aided Mol. Des.*, 11 (1997) 425.
- Mitchell, J.B.O., Laskowski, R.A., Alex, A., Forster, M.J. and Thornton, J.M., *J. Comput. Chem.*, 20 (1999) 1177.
- Jedrzejewski, M.J., Singh, S., Brouillette, W.J., Laver, W.G., Air, G.M. and Luo, M., *Biochemistry*, 34 (1995) 3144.
- Collard, P. and Clergue, M., *Complex Systems*, 12 (2000) 1.
- Head, R.D., Smythe, M.L., Oprea, T.I., Waller, C.L., Green, S.M. and Marshall, G.R., *J. Am. Chem. Soc.*, 118 (1996) 3959.
- Pellegrini, E. and Field, M.J., *J. Phys. Chem. A*, 106 (2002) 1316.
- Ghose, A.K., Viswanadhan, V.N. and Wendoloski, J.J., *J. Phys. Chem. A*, 102 (1998) 3762.
- Muegge, I. and Martin, Y.C., *J. Med. Chem.*, 42 (1999) 791.
- Luo, Z., Wang, R. and Lai, L., *J. Chem. Inf. Comput. Sci.*, 36 (1996) 1187.