Manual for GANDI

Genetic Algorithm-based de Novo Design of Inhibitors

Version 1.1

Fabian Dey and Amedeo Caflisch

To improve this documentation, please send comments and feedback to:
Amedeo Caflisch
E-mail: caflisch@bioc.uzh.ch
FAX: (++41 44) 635 68 62

May 28, 2008

See also: Dey, F. and Caflisch, A.,
Fragment-Based de Novo Ligand Design by Multiobjective Evolutionary Optimization,
## Contents

1 **Getting started**  
   1.1 Concept .................................................. 1  
   1.2 Files required for running GANDI .......................... 1  
   1.3 Running GANDI ............................................. 2  
   1.4 Output of GANDI ............................................ 2  

2 **Flow chart**  

3 **Algorithm**  
   3.1 Encoding ................................................... 4  
   3.2 Reproduction ............................................... 5  
     3.2.1 Individual selection ................................... 5  
     3.2.2 Cross over ............................................. 5  
     3.2.3 Mutation ............................................... 5  
   3.3 Linker Placement .......................................... 6  
   3.4 Scoring .................................................... 7  
     3.4.1 Scoring functions ..................................... 7  
     3.4.2 Multi-objective optimization .......................... 9  
   3.5 Merging of populations ................................... 9  
   3.6 Migration .................................................. 9  
   3.7 Storing of molecules ...................................... 10  

4 **Input file**  
   4.1 Structure .................................................. 11  
   4.2 Description ............................................... 14  
     4.2.1 Running GANDI in parallel ............................ 14  
     4.2.2 Optimizer ............................................. 15  
     4.2.3 Varia .................................................. 16  
     4.2.4 Receptor .............................................. 16  
     4.2.5 Cutoff values ......................................... 17  
     4.2.6 Docked Fragments and Linkers ........................ 17  

5 **Parameter File**  
   5.1 Structure .................................................. 20  
   5.2 Description ............................................... 21  
     5.2.1 Atom Section .......................................... 21  
     5.2.2 Bond Section ......................................... 21  
     5.2.3 Forbidden Connections ................................. 21  

6 **Troubleshooting**  
   6.1 Why ....? .................................................. 22  
   6.2 Error Messages ............................................ 23  

7 **Acknowledgments**  


1 Getting started

1.1 Concept

GANDI assembles molecules by joining fragments, which have been previously docked into a protein binding site (henceforth referred to as receptor), with user-defined linker fragments [1]. Heavy atom – hydrogen atom vectors constitute the possible attachment points on both the docked fragments and the linkers. The build-up method implemented in GANDI uses a combination of a genetic algorithm [2; 3] and a random tabu search [4; 5; 6], where the former is used to select the set of docked fragments and the latter explores possible linker attachments to join the docked fragments.

For lists of substituents and scaffolds commonly occurring in virtual libraries and drug molecules consult [7] and [8].

![Figure 1: Concept](image)

1.2 Files required for running GANDI

- **GANDI-executable**
  GANDI is statically compiled in 32-bit and 64-bit mode using the GNU-compiler suite. It should thus run on any up-to-date Linux operating system. The distributed executables adhere to the following naming:
  gandi. [release_version]_[architecture]_[tag]

- **Input file**
  Contains all project related data, specified for a certain run. For a detailed description see section 4.

- **Parameter file**
  Contains force-field related information as well as a list of forbidden connections. For a detailed description see section 5.

- **Receptor file**
  The receptor file needs to be in Sybyl MOL2 format containing all atoms (including both polar and non-polar hydrogens) with MSI CHARMM atom types and partial charges assigned.
• **Docked fragments**
  The preparation of the structures is identical to the one of the receptor file. Depending on the setup of the optimization procedure, additional information defined in the MOL2-files might be needed (see sections 3.4 and 4.2.6.1).

• **Linker library**
  Preparation is identical to the receptor including additional information depending on the setup of GANDI (see sections 3.4 and 4.2.6.2).

### 1.3 Running GANDI

GANDI is run on a Linux operating system in a Linux shell by typing:

```
gandi [input file] >& [output file]
```

### 1.4 Output of GANDI

GANDI prints information about the run to standard output and standard error. In order to save this output, run GANDI as described in 1.3. The verbosity of the output can be controlled through a parameter in the input file (4.2.3.5). See section 3.7 for details on the how information is stored within the generated molecules. Consult the troubleshooting section (6) if problems arise while trying to run GANDI.
Figure 2: Process flow is top to bottom including two iterative procedures, which are the main loop of the genetic algorithm and the random tabu search embedded within the former. Red arrows denote parts of the program which contain parallel code (see section 4.2.1.1).
3 Algorithm

Genetic algorithms are stochastic optimization procedures which mimic natural selection based on Darwin’s theory of evolution ([9] and [2] [3]). Derived from living beings, genetic algorithms use a simplified approach of mutation, cross over and selection to evolve a population. The latter consists of multiple individuals whose genetic material, the chromosomes, are strings of numbers encoding the traits to be optimized. Individuals are selected from the population and mutation or cross over of their chromosomes creates new individuals with novel chromosomes. The score of each new individual is calculated and the fittest of both the old and the new population survive, while maintaining diversity of the population to avoid premature convergence.

The genetic algorithm implemented in GANDI corresponds to an “island”-model (also referred to as parallel genetic algorithm introduced by Grosso [10]). In contrast to the “classic” genetic algorithm, the island model employs multiple populations which evolve independently of one another. After a user-defined number of iterations is reached, the islands exchange individuals (see 3.6), which introduces new genetic material to an island and thus helps a population in escaping a local minima as well as avoiding premature convergence. The user-defined number of iterations of the genetic algorithm serves as the termination criterion. For a in-depth treatment of genetic algorithm consult [2].

3.1 Encoding

Genes in a genetic algorithm usually encode their information as a binary string of numbers. The genetic algorithm implemented in GANDI however, stores the encoded information as integers, where a specific gene value corresponds to a single docked fragment position read in and numbered at startup of GANDI. Linker fragments are not encoded in the chromosome, but are evaluated separately for each individual in a random tabu search (see 3.3). The main reason not to encode the linker fragments with the genetic algorithm arises from the fact, that only few of the available linkers are able to connect a given set of docked fragments. Thus optimizing the linkers directly with the help of the genetic algorithm would result in exploration of unfeasible regions of the search space, which can be circumvented by un-coupling the process of adding the linkers by means of a tabu search on all reasonable linker connections.

Heavy atom clashes between docked fragments of a single individual are evaluated, whenever a new set of genes of an individual is created through random initialization at start-up or through reproduction. The individual is killed and removed from the population if the number of clashes overshoots the cutoff specified in the input file (4.2.5.1). This procedure tries to generate offspring with the tolerated amount of clashes for a defined amount of times, after which the dead individual is added to the new population and is ignored during the subsequent steps (3.5).

The number of genes per individual, which equals the number of docked fragments to be linked, is fixed during the entire run.
3.2 Reproduction

The reproduction process is responsible for generating new offspring from the current, “old” population. Individuals are selected from the latter and their genetic information undergoes modification to produce a set of new individuals. Freshly created individuals with clashing docked fragments are immediately removed from the population (see 3.1 and 4.2.5.1).

3.2.1 Individual selection

For this first step in the reproduction process, the user can choose between three procedures of how individuals should be selected for reproduction:

- **Tournament selection**
  Two individuals are randomly picked from the population and the fitter is kept for reproduction.

- **Fitness-based roulette wheel selection**
  This procedure mimics the spinning of a roulette-wheel where the size of the pies, which correspond to the individuals, is proportional to their score. The probability \( p_i \) of selecting a specific individual \( i \) for reproduction is proportional to its score or fitness:

  \[
  \text{with } S_{\text{norm},i} = S_{\text{total},i} - S_{\text{total,worst}} \\
  \text{and } p_i = \frac{S_{\text{norm},i}}{\sum_{k=1}^{N} S_{\text{norm},k}} \\
  \sum_{i=1}^{N} p_i = 1
  \]  

  Where \( S_{\text{norm}} \) is the score normalized with the worst score of the population \( S_{\text{norm,worst}} \) and \( N \) is the number of individuals. The normalization procedure ensures that the sum of all probabilities equals to one.

- **Rank-based roulette wheel selection**
  Rank-based roulette wheel selection is identical to its fitness-based counterpart with the only difference being that the rank is used instead of the score.

3.2.2 Cross over

If a picked individual is chosen to undergo cross over, a second individual is drawn from the old population with one of the methods described before. A random cross over point between any two genes is determined and the genes subsequent this point are exchanged between individuals.

3.2.3 Mutation

Each gene of a new individual is mutated with the mutation probability specified in the input file (see 4.2.2.4). Mutation of a selected gene is performed by modifying the
encoding integer value by a random amount, where the latter is drawn from a normal distribution with $\mu = \text{gene value}$ and a user-defined $\sigma$-ratio, where the $\sigma$-ratio is defined as:

$$\sigma - \text{ratio} = \frac{\sigma}{n_{fp}} \quad (4)$$

$n_{fp}$ corresponds the number of fragment positions. Using small $\sigma$-ratio leads to only small changes in the encoding value upon mutation, whereas with $\sigma$-ratio $> 1$ all fragment positions are almost equally likely to be reached with a single mutation (see figure 3).

Figure 3: Mutation with $n_{fp} = 100$ and $\mu = 20$

3.3 Linker Placement

Once a set of docked fragments has been assigned to an individual the docked fragments have to be linked, which is done separately for all individuals in a random tabu search. To do so, all docked fragment-vector – docked fragment-vector combinations of an individual are investigated for possible linker attachments fulfilling the tolerance criterions (4.2.5.3). For efficiency reasons a lookup table of all linker vector angles and distances is generated at the start of GANDI including a “zero-atom” linker used to directly merge two fragments. One docked fragment and a possible linkage solution (figure 4 step 1) are selected randomly and the linker fragment is added to the individual. Using a breadth-first search strategy, the recently connected docked fragments (and docked fragments joined previously) are partitioned into a “connected” and a “not-connected” group of docked fragments (figure 4 step 2). In the next step a “not connected” docked fragment and a linking solution are randomly picked again. The assembly continues until all docked fragments are connected or a maximal number of trials has been reached, thus yielding no
solution (figure 4 step 3). The new coordinates of the selected linkers are then calculated by superimposing the hydrogen atoms of the docked fragment vectors with the heavy atoms of the linker vectors and vice versa with the algorithm described by Kabsch [11]. No tabu search is performed for a given set of docked fragments if one or more docked fragments cannot be linked to any other and the individual encoding the latter is consequently killed. The scores (see 3.4) are calculated for the assembled molecule, the linkers are removed and a new attempt in linking the docked fragments is performed, where the maximal number of linking trials is a user-defined parameter specified in the input file (4.2.2.2). The build-up trials are stored in a tabu list to avoid unnecessary computation of the scores. At the end of the assembly trials of a given set of docked fragments only the linker combination with the lowest score is restored and kept.

3.4 Scoring

3.4.1 Scoring functions

There are three scoring function terms implemented in GANDI, which account for different properties of the ligand. A 3D structural (3.4.1.2) and a 2D fingerprint-based (3.4.1.3) scoring function term provide the user with the possibility to steer the optimization process towards an user-supplied target structure. Furthermore a force field-based scoring function term serves to calculate the interaction and internal energies of the generated molecule.

3.4.1.1 Force-field based scoring

The first scoring function term is force field based using MSI-CHARMMm parameters for the evaluation of the energy. This scoring function term is a sum of inter and intra van der Waals and electrostatic terms:

\[ E_{ff} = E_{vdW}^{inter} + E_{elec}^{inter} + E_{vdW}^{intra} + E_{elec}^{intra} \]  

(5)

The terms labeled \textit{inter} are the van der Waals and coulombic energies, using a distance-dependent dielectric, between all ligand atoms and the receptor.
\[ E_{vdW} = \sum_{i<j} \sqrt{\varepsilon_i \varepsilon_j} \left\{ \left( \frac{R_{vdW}^i + R_{vdW}^j}{r_{ij}} \right)^{12} - 2 \left( \frac{R_{vdW}^i + R_{vdW}^j}{r_{ij}} \right)^6 \right\} \] (6)
\[ E_{elect} = 332 \sum_{i<j} \frac{q_i q_j}{\epsilon_{int} r_{ij}^2} \] (7)

Where \( R_{vdW}^i, R_{vdW}^j \) are the van der Waals radii of atom i and j, \( \varepsilon_i \) is the minimum of the van der Waals potential between two atoms of type i at optimal distance of 2 \( \cdot R_{vdW}^i \), and \( r_{ij} \) is the distance between atoms i and j in Å. \( q_i \) and \( q_j \) are the partial charges in electronic units of atoms i and j, respectively, \( r_{ij} \) is the interatomic distance in Å.

The \textit{intra} terms are the internal interaction energies between groups (linkers and docked fragments) of the ligand. Interactions between atoms separated by one (1–2 interactions) or two bonds (1–3 interactions) as well as interactions between intra-group atoms are not calculated.

The potential of the receptor is calculated and stored on a grid to save computational time. Docked fragment pose energies are read in from the substructure section of the MOL2-file, where the second last number on the line represents the energy of the corresponding docked fragment:

\texttt{TRIPOS<SUBSTRUCTURE>}

1 MOL1 1 GROUP **** **** -10.87 0

3.4.1.2 3D similarity based scoring

The second scoring function term measures the similarity between the newly assembled ligand and a user supplied template molecule.

\[ Sim_{3D}(A, B) = \frac{S_{AB}}{\max(S_{AA}, S_{BB})} \] (8)
\[ S_{XY} = \sum_{i \in X} \sum_{j \in Y} w_{i,j} e^{-\gamma r_{ij}^2} \] (9)

where \( r_{ij} \) is the distance between two atoms \( (i \in \text{molecule } X, j \in \text{molecule } Y) \), \( w_{i,j} \) is a matrix whose coefficients reflect the similarity between element types (an unit matrix is currently used), and \( \gamma \) is a coefficient which acts on the broadness of the distribution of the positions. This scoring function term is identical to the one used for evaluating the similarity between two individuals of a population (see 3.5).

3.4.1.3 2D similarity based scoring

The last scoring function term is a fingerprint-based 2D-measure of similarity between the template ligand and the compound assembled by GANDI. The fingerprint similarity between the two molecules is calculated with the Tanimoto-coefficient:

\[ Sim_{2D} = \frac{\sum_{i=1}^{n} x_{iA} x_{iB}}{\sum_{i=1}^{n} x_{iA}^2 + \sum_{i=1}^{n} x_{iB}^2 - \sum_{i=1}^{n} x_{iA} x_{iB}} \] (10)

where \( x_{iA} \) denotes the \( i \)th fingerprint entry of molecule A.
To avoid computational cost and allow for flexibility, the fingerprints of the template, the linkers and the docked fragments are read in from the input MOL2 files. The fingerprint definition must appear before the “TRIPOS<MOLECULE>” section in the MOL2-file and has the general form:

```
#FINGERPRINT [number of entries] [F1] [F2] ... [Fn]
```

The fingerprint of the assembled molecule is then calculated by summing up the individual fingerprint entries of the linkers and docked fragments.

### 3.4.2 Multi-objective optimization

There exist numerous approaches of how scores measuring different traits of the same object can be used to assign an overall score or rank to an object under investigation ([12]). In the weighted sum approach the contributions of each individual scoring function term are multiplied by a user defined coefficient specified for each scoring function separately in the input file and summed up to yield the overall score:

$$S_{total} = w_{ff}E_{ff} - w_{3D}Sim_{3D} - w_{2D}Sim_{2D}$$

(11)

Where $w_{ff}$ is the force field coefficient, $E_{ff}$ the force field energy, $w_{3D}$ and $w_{2D}$ are the structural and fingerprint coefficients, and $Sim_{3D}$ and $Sim_{2D}$ are the corresponding similarities. The minus signs for the similarity scores stem from the fact that the similarities, contrary to the force field energy, increase with rising fitness. In GANDI, the user can choose to optimize the ligand building process according to any combination of the three scoring function terms listed in the previous section. If the weight of a specific scoring function term is set to zero in the input file (4.2.3.4), the corresponding scoring function term will not be computed and hence not used during optimization.

### 3.5 Merging of populations

Once the tabu search and the associated scoring have been performed for all individuals of the new population, the two populations are merged into a single population with the specified number of individuals. The merging procedure starts with the lowest scoring individuals of both populations. Insertion of members of the new into the old population occurs if the old population does not contain any too similar individual with a lower score. The similarity between two individuals is calculated with Equation 8. The similarity cutoff is specified in the input file (4.2.3.3). If an insertion of a new individual occurs, the remaining individuals of the old population with a less favorable score are checked for their similarity to the added member. All individuals of the old population are given an arbitrary high score and moved to the end of the score-sorted population if they are less fit and too similar compared with the inserted individual. The procedure stops once the old population has reached the requested amount of individuals at the current step of the merging.

### 3.6 Migration

The main difference between a classic and an island genetic algorithm is, that in the latter multiple populations are evolved simultaneously in separated niches. After a user-defined
number of iterations (4.2.2.5), the isolated islands exchange a fixed amount of individuals with one another according to one of the following models

- All with all
  Every island exchanges individuals with every other island, which can lead to early convergence of the genetic material of the islands if the number of individuals that are exchanged is set too high.

- Neighbor
  Only neighboring islands exchange individuals.

- Random
  Every island chooses one island randomly with which individuals are exchanged.

Figure 5: Exchange models: all with all, neighbor and random

3.7 Storing of molecules

GANDI writes out MOL2-files of all surviving individuals (see section 6) at the end of the optimization and after a user-defined number of steps. The individual scores of each scoring function term as well as the overall score are written to the header of the MOL2-file. All docked fragments and linkers of each GANDI molecule are stored in separate groups labeled MP_# or LK_#, respectively.
4 Input file

4.1 Structure

---

```plaintext
# GANDI input file -- Version 1.1
#
# 4.2.1.1
# number of threads
1
#
# 4.2.2.1
# number of islands | number of individuals / island | number of kept individuals
4 100 0
#
# 4.2.2.2
# number of cycles | number of individual trials | number of linker trials
1000 20 20
#
# 4.2.2.3
# ligand size (number of docked fragment positions / ligand)
2
#
# 4.2.2.4
# selection mode
3
#
# mutation probability | mutation sigma-ratio | crossover probability
0.2 0.1 0.8
#
# 4.2.2.5
# exchange mode | exchange steps | exchange ratio
3 20 0.05
#
# 4.2.3.1
# saving step
20
#
# 4.2.3.2
# seed
3294
#
# 4.2.3.3
# similarity cutoff | similarity exponential factor | squared distance cutoff
0.8 0.9 16
#
# 4.2.3.4
# Scoring function term coefficients: \( w_{ff} \) | \( w_{3D} \) | \( w_{2D} \)
# template name
0.02 1 0
/1ke5_nativeligand.mol2
#
# 4.2.3.5
# Print level in the output file: 0 (lean) - 4 (verbose)
2
#
# 4.2.3.6
# Parameter filename
/parameter.file
#
# 4.2.4.1
# Receptor coordinates (in mol2 format) filename
protein/1ke5.mol2
#
```

Table 1: Input file part 1: This input file was used in [1]
# Binding site residue list
# First line: number of residues
20
10
11
13
18
31
33
57
73
74
75
76
77
78
79
82
124
125
127
137
138
#

4.2.4.2
# van der Waals energy:
# grid margin | grid spacing
# write (w) or read (r) van der Waals grid | grid filename
10.0 0.3
w ./Grids/vdwaals.grid.dat
#

4.2.4.3
# Coulombic energy:
# dielectric constant | distance dependent dielectric (1=yes,0=no)
# grid margin | grid spacing
# write (w) or read (r) Coulombic grid | grid filename
4.0 1
10.0 0.5
w ./Grids/coulombic.grid.dat
#

4.2.5.1
# Clash checking for fragment-fragment and fragment-linker-fragment:
# scaling factor for interatomic distance
# severe clash factor
# maximal number of tolerated clashes
# maximal number of tolerated severe clashes
0.89 0.5 10 0
#

4.2.5.2
# Maximal squared distance between two hydrogen atoms of two docked fragments
# connection vectors for which linker connection vectors with the
# same origin are considered
2.0
#

4.2.5.3
# Tolerances:
# squared distance (Å²), planar angle (deg), dihedral angle (deg)
1.0 30.0 50.0
#

Table 2: Input file part 2: This input file was used in [1]
4.2.6.1

# Docked Fragments:
# number of docked fragments
# type of vector definition
# find equivalent vectors
# connection vector bump check / cutoff
# docked fragment filename
# number of connection vectors
# list of connection vectors (heavy atom - hydrogen atom numbers)
100027
list

fragments/fragment_10_pos1.mol2
  1
  2
fragments/fragment_100_pos11.mol2
  1
  2
fragments/fragment_108clus1.mol2
  1
  2
.
.
.
.

4.2.6.2

# Linkers:
# number of linkers
# type of vector definition
# linker filename
# number of connection vectors
# list of connection vectors (heavy atom - hydrogen atom numbers)
17372
list

linkers/linkers_777.mol2
  2
  2 10 7 13
linkers/linkers_521.mol2
  2
  2 11 8 9
linkers/linkers_590.mol2
  2
  2 1 8 9
.
.
.
.

Table 3: Input file part 3: This input file was used in [1]
4.2 Description

Some helpful hints on the build-up of the input file:

- Lines starting with # are comment lines and are ignored.
- The order of the information containing sections is fixed and must not be altered.
- Regions encoding information must not contain any unnecessary information like additional comments. No template file should be defined in the input file (4.2.3.4), if both the coefficient of the structural and the compositional scoring function are set to zero (see 3.4).

4.2.1 Running GANDI in parallel

4.2.1.1 GANDI Setup

GANDI supports the use of multiple processors of a shared memory architecture (i.e. a PC with multiple cores) by use of the application programming interface OpenMP. The executable capable of running in parallel is named:

```
gandi.release_version_[architecture].openmp
```

Some parts of the setup of GANDI before the actual optimization have also been parallelized (see Figure 2). The islands do not interact between exchanges of individuals during optimization and can be evolved on separate processors. Synchronization is enforced before the exchange of individuals between islands, i.e. all islands have to undergo the same amount of iterations before the exchange (Figure 6). The number of threads (which often equals the number of CPU cores available) used by GANDI should ideally be a divisor of the number of islands to achieve optimal performance gain. Otherwise idle threads can encounter long waiting periods. The gained speedup thus depends critically on the setup of the optimization parameters as well as the number of threads used (for an example see Table 4).

Figure 6: Running GANDI with 2 threads, 10 islands and 100 iterations in total with exchange of individuals every 20 steps. Synchronization of the optimization occurs before the exchange of individuals.
<table>
<thead>
<tr>
<th>Number of threads</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>1.64</td>
</tr>
<tr>
<td>3</td>
<td>1.79</td>
</tr>
<tr>
<td>4</td>
<td>2.19</td>
</tr>
<tr>
<td>5</td>
<td>2.41</td>
</tr>
</tbody>
</table>

Table 4: Speedup by using multiple processors with a setup consisting of 10 islands with 100 individuals optimizing for 200 iterations and exchanging individuals every 50 iteration steps.

4.2.2 Optimizer

4.2.2.1 Population

The first value defines how many islands, corresponding to simultaneous genetic algorithms, should be run at the same time. The latter two specify how many individuals a single island should be made up of and how many of the fittest should not be exchanged with other islands respectively (see 3.6).

4.2.2.2 Iterations

This section defines the number of iterations for the genetic algorithm, the tabu search and how often GANDI should try generating a single molecule with non overlapping docked fragments.

4.2.2.3 Size

The ligand size, which remains constant throughout the entire run, corresponds to the number of docked fragments that should be connected together.

4.2.2.4 Reproduction

Possible values for the selection mode are tournament selection (1), rank-based roulette wheel selection (2) and fitness-based roulette wheel selection (3), responsible for selecting individuals for reproduction. The following three values define the likelihood and broadness with which the reproduction operators are carried out (see 3.2.2 and 3.2.3).

4.2.2.5 Migration

The exchange of individuals between islands is defined by which islands exchange individuals with one another (1: all with all, 2: only with neighboring and 3: a randomly selected island), after how many iterations of the genetic algorithm exchanges occur and how many individuals should migrate (3.6).
4.2.3 Varia

4.2.3.1 Saving step

Additionally to writing out all alive individuals at the end of the optimization run, GANDI provides the possibilities to store these to disk repeatedly after a certain number of steps.

4.2.3.2 Random seed

The random seed is used to initialize the random number generators responsible for selecting islands in the random exchange mode (3.6), individuals during the reproduction procedure (3.2) and linkers during the tabu search (3.3).

4.2.3.3 Similarity

The similarity cutoff determines the maximal similarity of two molecules residing in the same island (see 3.5) determined with formula 8 with the exponential factor corresponding to $\gamma$. The last value determines the maximal squared distance between any two atoms for which the similarity should be calculated. This distance cutoff does not apply for the structural similarity based scoring function, but only when assessing the similarity of two individuals of a single island (see 3.5), which decreases the computational cost without losing too much accuracy.

4.2.3.4 Scoring

The three floats correspond to $w_{ff}$, $w_{3D}$ and $w_{2D}$ of formula 11, which are the weights of the individual scoring function terms. A specific scoring function term is omitted and not even evaluated whenever its coefficient is set to zero. The template MOL2-file should not be defined when both $w_{3D} = 0$ and $w_{2D} = 0$.

4.2.3.5 Print level

The print level determines the output verbosity and increases with increasing number (from 0-4).

4.2.3.6 Parameter file

Specifies the location and name of the parameter file (see 5).

4.2.4 Receptor

4.2.4.1 Binding site

Defines the location and name of the receptor MOL2-file as well as the number of binding site residues and their index. The binding site residues are needed to determine the coordinate maxima of the binding site and by that determine the size of the potential energy grids.
4.2.4.2 Van der Waals energy grid

The first value determines the margin around the binding site residues for which the potential energy grid will be calculated and the second its resolution. The grid has to be only written once ("w") for every protein, after which it can be simply read ("r") in, which greatly reduces the computational cost.

4.2.4.3 Coulombic energy grid

Similar to 4.2.4.2 with the exception that the value of the dielectric constant has to be defined and whether or not GANDI should use a distance-dependent dielectric model to account for electrostatic interactions.

4.2.5 Cutoff values

4.2.5.1 Heavy atom clash

The number of small and severe clashes between the heavy atoms of two docked fragments is calculated with formula 12 and 13 whenever a new individual is created (see 3.1). Where $r_{ij}$ is the interatomic distance, $c_{small}$ and $c_{severe}$ are the scaling factors for the distances and $R_i$ and $R_j$ are the van der Waals radii of atoms $i$ and $j$. The maximal tolerated number of small and severe clashes between two docked fragments can be set by the user.

\[
\text{small if: } r_{ij}^2 < r_{small}^2 = (c_{small} \times (R_{vdW_i} + R_{vdW_j}))^2 \\
\text{severe if: } r_{ij}^2 < (c_{severe} \times r_{small})^2
\]  

(12)

(13)

4.2.5.2 Maximal hydrogen distance

This values determines the maximal squared distance between two hydrogen atoms of two docked fragments for which linker connections with the same origin are considered. This cutoff serves as a fast measure for feasible linker connections, by avoiding highly constrained bond angles.

4.2.5.3 Vector tolerances

GANDI builds a look-up table with all angles and distances of all linker vectors on start-up. The distances and angles of all pairs of connection vectors for all sets of docked fragments are measured separately during the tabu search and linkers with vectors fulfilling the constraints within the tolerances specified in this section are deemed feasible and used during linker placement (see 3.3).

4.2.6 Docked Fragments and Linkers

4.2.6.1 Docked Fragment section

The first number specifies how many docked fragments should be used during the optimization procedure. The next key word specifies if the connections are defined by the user ("list" keyword) or should be generated by GANDI (keywords "all" or "daim"). In the latter case the user only has to specify the location of the docked fragments while
omitting any connection vector information. Possible vectors are all heavy → hydrogen (origin → extension) atom bonds, where the indexes of the origin and the extension of the vectors refer to the first column of the @<TRIPOS> ATOM section in the Sybyl MOL2 file (1.1).

The three approaches of defining connection vectors are:

- **list**: The user is responsible for defining the connection vectors in the input file (see Table 5 above).

- **all**: GANDI automatically generates the list of all possible connection vectors (all heavy–hydrogen atom vectors).

- **daim**: With this approach GANDI relies on fragments obtained by the decomposition of molecules with DAIM. When DAIM decomposes molecules into fragments (with the command line options “exhaustive-sets –connection-info”), individual fragments are written to disk and the names of heavy atoms, which were the previous linking points inside the entire molecule are marked with a small “x”. GANDI checks heavy atom names for the occurrence of the letter “x” and evaluates all connections vectors originating from the latter.

The following letter (“n” or “y”) specifies if GANDI should look for equivalent connection vectors to the ones already defined, e.g. when a docked benzene fragment is defined with only one connection vector this procedure will also include the other five. Two vectors are not equivalent if the number and atom types of atom neighbors of the origins of the two vectors are not equal. If the afore mentioned criterion is met, GANDI makes a copy of the fragment and superimposes the hydrogen, heavy and a neighboring atom of the latter of the first vector of the original and the second vector of the copied fragment with one another using the method described by Kabsch [11] (Figure 7). If the similarity according to Equation 8 is $\geq 0.99$ the two vectors are considered equivalent and the new connection vector is added to the fragment definition.

The last letter (“y [energy cutoff]” or “n”) before the docked fragment list specifies if GANDI should investigate the connection vectors for their feasibility, i.e. when a vector
is pointing towards the protein surface and is thus inaccessible. This is done by placing a probe atom at 1.5 Å from the origin of every connection vector (heavy atom) in direction of the connection vector (heavy – hydrogen atom vector). The van der Waals interaction energy is calculated between the probe atom and the receptor and connection vectors are discarded if the energy is higher than the user defined cutoff. The probe atom is defined in the parameter file (atom type “BUMP”).

MOL2-file of docked fragments may contain more than one substructure, corresponding to distinct binding modes of the same fragment. The energy has to be specified separately in the substructure section (see 3.4), whereas the fingerprint is defined only once in the header of the MOL2-file. GANDI finds equivalent and assigns connection vectors based solely on the first binding mode of a docked fragment in case multiple poses are stored in an individual MOL2-file. It is thus important to only store fragments with the same conformation in a single MOL2-file.

4.2.6.2 Linker section

The linker section is identical to the docked fragment section (4.2.6.1) described above. The main difference in the MOL2-files of the docked fragments and the linkers is the fact that the former may contain multiple conformations whereas the latter can not. Furthermore, no search for equivalent connection vectors or bump checking of the same is performed for linkers.

Figure 7: Superpositioning example adamantane: vectors under investigation are red, neighboring atoms are green.
5 Parameter File

5.1 Structure

5.2.1 Atom element van der Waals

<table>
<thead>
<tr>
<th>Type</th>
<th>Number</th>
<th>Radius</th>
<th>Energy</th>
<th>Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>5</td>
<td>1.17</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>6</td>
<td>1.870</td>
<td>0.1410</td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td>6</td>
<td>1.870</td>
<td>0.1410</td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td>6</td>
<td>1.870</td>
<td>0.1410</td>
<td></td>
</tr>
<tr>
<td>CSR</td>
<td>6</td>
<td>2.040</td>
<td>0.0500</td>
<td></td>
</tr>
<tr>
<td>CSRE</td>
<td>6</td>
<td>2.100</td>
<td>0.0903</td>
<td></td>
</tr>
<tr>
<td>CSRP</td>
<td>6</td>
<td>2.040</td>
<td>0.0500</td>
<td></td>
</tr>
</tbody>
</table>

5.2.2 Atom atom bond

<table>
<thead>
<tr>
<th>Type</th>
<th>Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>CT</td>
<td>1.58</td>
</tr>
<tr>
<td>B</td>
<td>N</td>
<td>1.42</td>
</tr>
<tr>
<td>B</td>
<td>OE</td>
<td>1.36</td>
</tr>
<tr>
<td>B</td>
<td>OT</td>
<td>1.325</td>
</tr>
</tbody>
</table>

5.2.3 Number of forbidden connections

<table>
<thead>
<tr>
<th>List of forbidden connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
</tr>
<tr>
<td>DC</td>
</tr>
<tr>
<td>DC</td>
</tr>
<tr>
<td>DC</td>
</tr>
<tr>
<td>DC</td>
</tr>
<tr>
<td>DC</td>
</tr>
<tr>
<td>OE</td>
</tr>
<tr>
<td>OE</td>
</tr>
<tr>
<td>OE</td>
</tr>
<tr>
<td>OE</td>
</tr>
</tbody>
</table>

Table 6: Excerpt of parameter file
5.2 Description

5.2.1 Atom Section
Contains a list of atom properties extracted from the MSI CHARMm force field needed for calculating the similarity and the van der Waals energy between two molecules.

5.2.2 Bond Section
Maintains a list of bond parameter extracted from the MSI CHARMm force field as well.

5.2.3 Forbidden Connections
This section comprises a list of atom type pairs that should never be connected to one another.
6 Troubleshooting

6.1 Why .... ?

- Why is the number of individuals defined in the input file larger than the number of MOL2-files actually written to disk?
  GANDI only writes MOL2-files of individuals to disk that are “alive” (see 3.7).

- Why does a MOL2-file sometimes not contain any or fewer linker (LK) substructures?
  GANDI also checks if two fragments can be merged directly with one another without any intermediate linker fragment (see 3.3), which reduces the number of linkers.

- Why are molecules with an arbitrarily high score generated during the optimization procedure?
  There are multiple scenarios during optimization where an individual is not considered to be valuable anymore and is thus given a high score:
  - Linker placement (3.3): One or more docked fragments cannot be connected to any other docked fragments due to missing feasible linkers.
  - Merging (3.5): There is at least one similar individual with a lower score in the population.
  - No non-overlapping docked fragments were found for the individual.

- Why does GANDI finish without finding any possible solutions?
  Two common causes are:
  - There are only overlapping docked fragments
  - No suitable linkers were found to join the docked fragments

Different approaches can be used to circumvent the problem:
  - Increasing the number of tolerated clashes (4.2.5.1).
  - Increasing the connection vector angle and distance tolerances (4.2.5.3).
  - Increase the number of individual trials and tabu search iterations (4.2.2.2).

- Why does GANDI quit prematurely complaining about missing files or improper input values which appear to be properly defined in the input file?
  An incomplete or incorrect section, preceding the actual section where the error message was issued, might be responsible for the early termination of GANDI.

- Why does GANDI create molecules which show little overlap with the template structure although $w_{3D} > 0$?
  The force field based scoring function term (or the fingerprint-based scoring function term) seem to be overemphasized compared to $Sim_{3D}$. Increase $w_{3D}$ so far as that both the force field and the fingerprint-based term have on average as little as e.g. 5% influence on the total score of a single molecule.
6.2 Error Messages

- **WARNING**, Number of threads <= 0 : ... , resetting to 1!
  
  **WARNING**, Number of threads >= number of islands : ... ... , resetting number of threads==number of islands!
  
  **WARNING**, Number of threads > max. number threads : ... , resetting number_threads==max_threads!

  Reason: The maximal number of threads cannot be higher than the computer’s threshold and should not be higher than the number of islands.
  
  Solution: Adjust number of threads.

- **WARNING**, ... vector definition should be "all", "list" or "daim"!
  
  Reason: The keyword defining how the connection vectors are generated is wrong.
  
  Solution: Update input file.

- **WARNING**, The docked fragment ... does not have any connection vectors!
  
  Reason: GANDI did not find any possible connection vectors for a specific fragment when using the "all" or "daim" approach (see 4.2.6.1).
  
  Solution: Update input file.

- **WARNING**, The connection vector ... for ... is not correct.
  
  **WARNING**, The list of connection vectors for ...
  
  Reason: Definition of a vector for a specific docked fragment or linker is wrong.
  
  Solution: Check vector definition.

- **WARNING**, The ... grid you want to read has not been created with the same input file parameters.
  
  Reason: Reading of a potential energy grid which has been created with different parameters.
  
  Solution: Re-write grids with the actual input file in use.

- **WARNING**, There are no parameters for atom type ...
  
  Reason: Encountered unknown atom types while reading certain Sybyl MOL2 files.
  
  Solution: Check and update atom type definition in MOL2 file or expand GANDI parameter file.

- **WARNING**, ... should be either 'y' or 'n', Exiting!
  
  Reason: Incorrect switch setting.
  
  Solution: Check switches in input file.

- **WARNING**, The file switch for ... is neither 'w' nor 'r'.
  
  Reason: Incorrect switch setting.
  
  Solution: Check switches in input file.

- **WARNING**, caught exception while reading ...
  
  Reason: Encountered premature end of file or file read error.
  
  Solution: Check corresponding section or file.
• WARNING, Number of islands <= 0, exiting
  Reason: At least one island has to be defined.
  Solution: Update input file.

• WARNING, Number of individuals <= 0, exiting
  Reason: The genetic algorithm needs multiple individuals to work efficiently.
  Solution: Update input file.

• WARNING, Number of kept individuals < 0, exiting
  WARNING, Number of kept individuals >= number of individuals: ...
  Reason: The number of kept individuals must not be below zero or larger than the
  number of individuals itself.
  Solution: Update input file.

• WARNING, Number of iterations <= 0
  Reason: The number of iterations must be larger than zero.
  Solution: Update input file.

• WARNING, Ligand size < 2: ... , exiting
  Reason: The ligand size corresponds to the number of docked fragments that
  should be connected, which needs to be larger than one.
  Solution: Update input file.

• WARNING, Number of binding site residues should be > 0
  , exiting
  Reason: No binding site residues have been specified in the input file from which
  the binding site grids would be calculated.
  Solution: Update input file.

• WARNING, ... grid margin should be > 0! : ... , exiting
  Reason: The safety margin around the vdW or coulombic grid was not defined
  properly.
  Solution: Update input file.

• WARNING, ... grid spacing should be > 0! : ... , exiting
  Reason: The grid spacing of the vdW or coulombic grid was not defined properly.
  Solution: Update input file.

• WARNING, Number of docked fragments/linkers should be > 0! : ...
  , exiting
  Reason: No docked fragments or linkers were defined.
  Solution: Update input file.

• WARNING, There is more than one substructure in ...
  Reason: One of the specified linkers contains more than one molecule.
  Solution: Remove all but one structure from file.

• WARNING: File read error occurred in parameter file,
  atom section, Exiting!
WARNING : File read error occurred in parameter file, bond section, Exiting!
WARNING : File read error occurred in parameter file, forbidden connection section, Exiting!
Reason : Encountered premature end of file or file read error while reading parameter file.
Solution : Check corresponding section of parameter file.

• WARNING The file ... cannot be opened/created
Reason : The specified file is missing or cannot be created due to write permissions or missing folders.
Solution : check existence of files or access permission of folders.

• WARNING, "#FINGERPRINT" not properly defined for template!
WARNING, Missing fingerprint information in linkers : ...
WARNING, Missing fingerprint information in docked fragment : ...
WARNING, Error reading #FINGERPRINT in file ... ; exiting!
Reason : Missing or incorrect fingerprint definition for template, linker or docked fragment.
Solution : Update fingerprint definition.
7 Acknowledgments

We thank N. Majeux, P. Kolb, P. Schuetz and P. Alfarano for useful discussions and comments.

References


### A  Changelog

<table>
<thead>
<tr>
<th>Version</th>
<th>Release Date</th>
<th>Changes</th>
</tr>
</thead>
</table>
| 1.1     | 27.05.2008   | - Possibility of automatic assignment of connection vectors by GANDI  
- Bump check for unfeasible connection vectors of docked fragments  
- Search for equivalent connection vectors of docked fragments where only a subset of all possible connection vectors is given  
- Parallelization of source code for shared memory architectures with OpenMP  
- General function optimizations for increase in efficiency  
- Change of random number generators to ensure that results are the same - independent of how many processors are used to run GANDI. |
| 1.0     | 01.02.2008   | - Initial GANDI release - |