Manual for GANDI

Genetic Algorithm-based de Novo Design of Inhibitors

Version 2.0

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

June 29, 2014

See also: Dey, F. and Caflisch, A.,
Fragment-Based de Novo Ligand Design by Multiobjective Evolutionary Optimization,
Journal of Chemical Information and Modeling, 48, 3, 679 - 690, **2008**

Contents

1 Getting started

1.1 Concept

GANDI assembles molecules by joining fragments ("linking" approach to de novo design), which have been previously docked into a protein binding site (also 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 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 docked fragments (see sections 2 and 3). For examples of substituents and scaffolds commonly occurring in virtual libraries and drug molecules consult e.g. [7; 8; 9; 10].

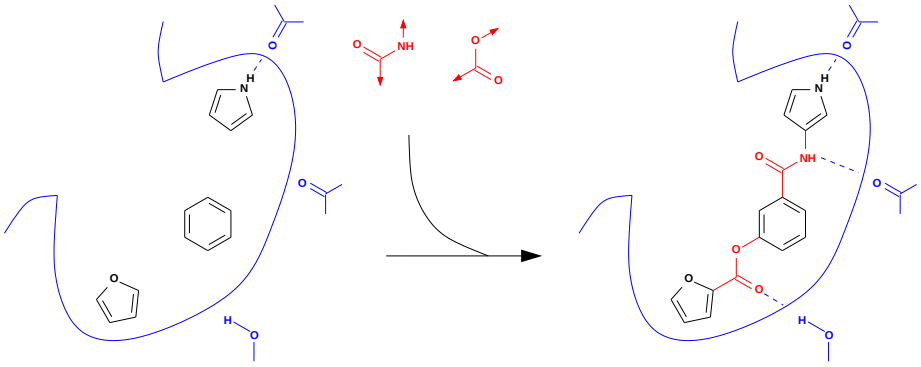


Figure 1: Concept

1.2 Files required for running GANDI

- *GANDI-executable*
  GANDI is statically compiled in 32-bit and 64-bit mode. It should thus run on any up-to-date Linux operating system. The distributed executables adhere to the following naming convention:

  `gandi_[compiler]_[release_version]_[architecture]_[tag]`

  Files tagged with "openmp" are able to distribute calculations onto different cores of a shared memory architecture (4.6.1).

- *Input file*
  Contains all project related data specific for a run (for examples see section 4).

- *Parameter file*
  Contains force-field related information, lists of forbidden connections and substructures as well as the pharmacophore definitions (section 5).

- *Receptor file*
  The receptor file has to be in Sybyl MOL2 format containing all atoms (including

both polar and non-polar hydrogens) with Accelrys-CHARMm atom types and partial charges assigned.

- *Docked fragments*
  The preparation of the fragments is similar to the receptor. Depending on the setup of the optimization procedure, additional information defined inside the MOL2-files might be needed (see sections 3.4 and 4.6.6.1)

- *Linker library*
  Preparation is identical to the receptor including additional information depending on the setup of GANDI (see sections 3.4 and 4.6.6.3).

1.3 Running GANDI

GANDI is run in a Linux shell by typing:

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

GANDI prints information about the run to standard output and standard error. In order to save this output, run GANDI as described. The verbosity of the output can be controlled through a parameter in the input file (4.6.3.5). See section 3.9 for details on how information is stored within the generated molecules. Consult the Troubleshooting and Help guide section 6 if problems arise while trying to run GANDI.

1.4 Outlook

The following two sections (2 and 3) give an overview of the implemented approach, which is followed by a discussion of the available input parameters. The latter are also mentioned in the respective sections throughout the manual and follow in most cases the format `[INPUTTAG] [INPUT TYPE]`. Examples of input files are given in section 4.

2 Flow chart

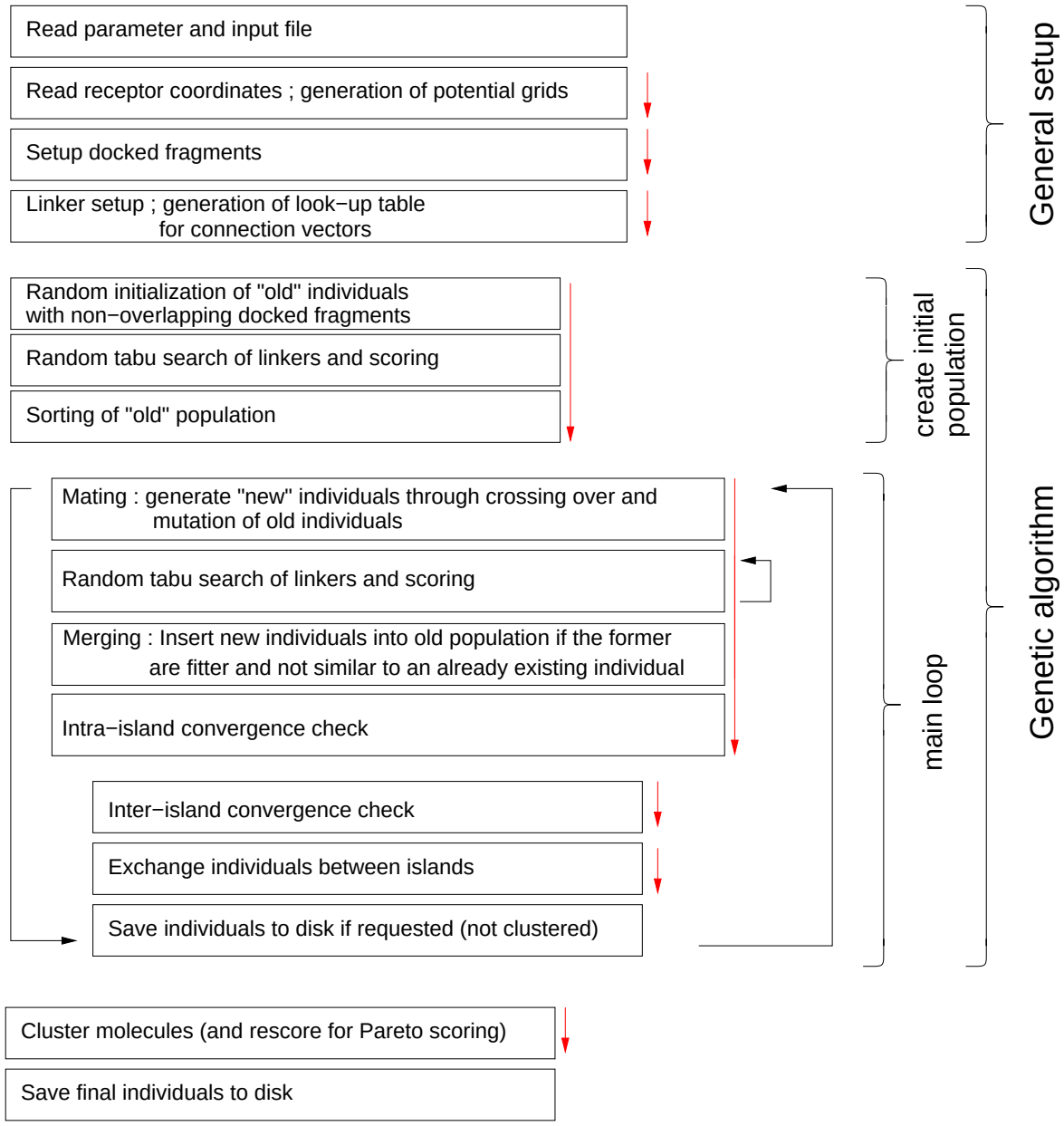


Figure 2: Process flow is top to bottom including two iterative procedures, which are the main loop of the genetic algorithm and the random tabu search embedded within the former. Red arrows denote parts of the program which contain parallel code (see section 4.6.1).

3 Algorithm

Genetic algorithms [2; 3] are stochastic optimization procedures which mimic natural selection based on Darwin's theory of evolution [11]. Genetic algorithms use a simplified approach of mutation, crossover 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 crossover 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 (referred to as $(\mu + \lambda)$ selection), 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 [12]. In contrast to classic genetic algorithms, 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 3.8), which introduces new genetic material to an island and thus helps a population to escape a local minima as well as avoiding premature convergence. An user-defined number of iterations of the genetic algorithm or two convergence checks can serve as the termination criterion. For an in-depth treatment of genetic algorithms consult [2].

3.1 Encoding

Genes in a genetic algorithm often encode their information as a binary string of numbers. The genetic algorithm implemented in GANDI stores the encoded information as integers, where a specific gene value corresponds to a single docked fragment position read in and numbered at startup. The number of genes per individual, which equals the number of docked fragments to be linked, is fixed during the entire run. 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 genetic algorithm would result in exploration of vast unfeasible regions of the search space. This can be circumvented by uncoupling the process of adding linkers by means of a tabu search on only feasible linker connections.

3.2 Reproduction

The reproduction (mating) process is responsible for generating new offspring from the "old" population. Individuals are selected from the latter and their genetic information undergoes modification to produce a set of new individuals. New individuals with clashing docked fragments are immediately removed from the population (explained in more detail in 3.2.4 and 4.6.5.1). This mating process is repeated for a user-defined number of trials for a single individual (`NUMINDITERATIONS [integer]` see 4.6.2.2) until a feasible, i.e. non-overlapping set of docked fragments is obtained.

3.2.1 Individual selection

For the first step in the reproduction process, the user can choose between two procedures of how individuals should be selected for reproductionwhere the rank-based roulette wheel approach is the default:

- **Tournament selection**
  Two individuals are randomly picked from the population and the fitter is used for reproduction (activated with `SELECTIONMODE TOURNAMENT`)

- **Rank-based roulette wheel selection**
  This procedure mimics the spinning of a roulette-wheel where the size of the pies is proportional to the rank of the individuals (activated with `SELECTIONMODE RANKROULETTE`). The probability p_i of selecting a specific individual i for reproduction is:

$$\text{with} \quad R_{norm,i} = R_{total,i} - R_{total,worst} \tag{1}$$

$$\text{and} \quad p_i = \frac{R_{norm,i}}{\sum_{k=1}^{N} R_{norm,k}} \tag{2}$$

$$\sum_{i=1}^{N} p_i = 1 \tag{3}$$

  Where R_{norm} is the rank normalized with the worst rank of the population $R_{norm,worst}$ and N is the number of individuals. The normalization procedure ensures that the sum of all probabilities equals to one.

3.2.2 Crossover

If a picked individual is chosen to undergo crossover, a second individual is drawn from the old population with one of the methods described before. A random crossover point between two genes is determined and all subsequent genes beyond this point are exchanged between individuals (i.e. sets of docked fragments are swapped). The probability of a crossing over event to happen is controlled with `CROSSOVERP [probability]` ($\in [0, 1]$) in the input file.

3.2.3 Mutation

Each gene of a new individual is mutated with the mutation probability specified in the input file (`MUTATIONP [probability]` see 4.6.2.4). 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 $\mu = \textit{current gene value}$ and an user-defined σ-ratio (`MUTATIONSIGMA [real]`).

$$\sigma - ratio = \frac{\sigma}{n_{fp}} \tag{4}$$

$$\text{gene value}_{new} = \text{gene value}_{old} + \text{random number} \tag{5}$$

Where n_{fp} corresponds to the total number of fragment positions and the random number is drawn from a normal distribution with standard deviation $\sigma = \sigma_{ratio} * n_{fp}$. Using

a small σ-ratio leads to only small changes in the encoding value upon mutation (i.e. subsequently read fragment poses), whereas with a σ-ratio > 1 all fragment positions are almost equally likely to be reached with a single mutation (see figure 3).

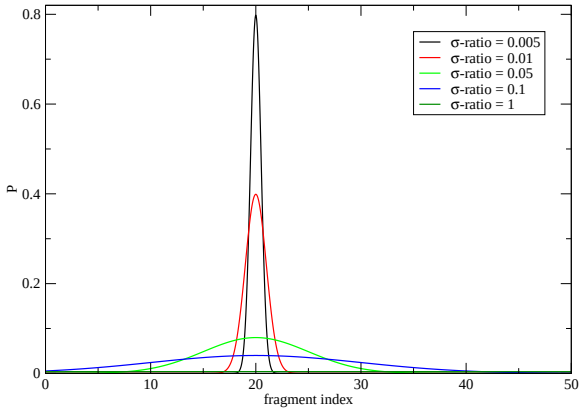


Figure 3: Mutation with $n_{fp} = 100$ and $\mu = 20$

3.2.4 Feasibility assessment

A check of the feasibility of a molecule is performed during reproduction by evaluating the fragments for heavy atom - heavy atom clashes. Two types of clashes ("severe" and "small") are distinguished and the maximal tolerated number of both can be set in the input file (`MAXNUMSEVCLASH [integer]` and `MAXNUMCLASH [integer]`). Clashes are docked fragment atoms that are closer than their combined van der Waals radii scaled by a factor (`SEVCLASHSCALFACTOR [real]` and `CLASHSCALFACTOR [real]`, see section 4.6.5.1). Hydrogens are ignored in the clash check as direct linking of two docked fragments would be difficult to achieve.

Molecules that do not fulfill either of these conditions are discarded and a new set of docked fragment poses is obtained. Attempts to create a specific individual are repeated for an user-defined number of times (`NUMINDITERATIONS [integer]`, section 4.6.2.2), if only overlapping fragment poses are found. A molecule is tagged as "dead" if it cannot be created after this specified number of times, added to the population, but is ignored in further calculations.

3.3 Linker placement with tabu search

3.3.1 Linker look-up

As outlined before, once the linking process starts, the "surviving" individuals consist of non-overlapping fragment poses (or with the allowed amount of clashes) and contain the mandatory fragments (explained below) if the user decided to run GANDI with the respective options. Linking is done separately for all individuals in a random tabu search.

To do so, all pairs of connection vectors of docked fragment poses of an individual are investigated for the existence of potential linkers. For efficiency reasons a look-up table of all linker vector angles and distances is generated at the start of GANDI (before optimization). This includes a "zero-atom" linker used to directly link two fragments, which can be switched off in the input file (`ZEROATOMLINKER [y/n]`). Self-connections of a docked fragment pose with itself are ignored. Linkers are retrieved from the bin (and neighboring bins) of the look-up table that match the connection vectors of pairs of docked fragment poses. Where the extended hydrogen positions are used to derive distances and angles (vector extensions are explained in 4.6.5.3). The look-up table is built with user-defined bin sizes (see 4.6.5.3).

A computationally inexpensive test is performed if two fragments should be connected together. This check is only based on the connecting atom types (see 5.2.5) and does not include a substructure search, which follows at a later stage (as the molecule is not yet fully formed). Thus, the resulting list of possible connections is made up of all linkers with roughly appropriate connection geometry between all pairs of docked fragment poses, as well as a pre-screening of unwanted connections based on atom types. No tabu search is performed for a given set of docked fragments if one or more docked fragments cannot be linked to any other. The individual encoding the latter is subsequently ignored.

3.3.2 Linking docked fragments

Using this list of potential connections, a docked fragment and a possible linking solution are selected randomly (figure 4 step 1) and the linker fragment is added to the individual. Using a breadth-first search strategy, the 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 (figure 4 step 3) or a maximal number of trials has been reached, thus yielding no solution.

The new coordinates of the selected linkers are calculated by superpositioning 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 [13].

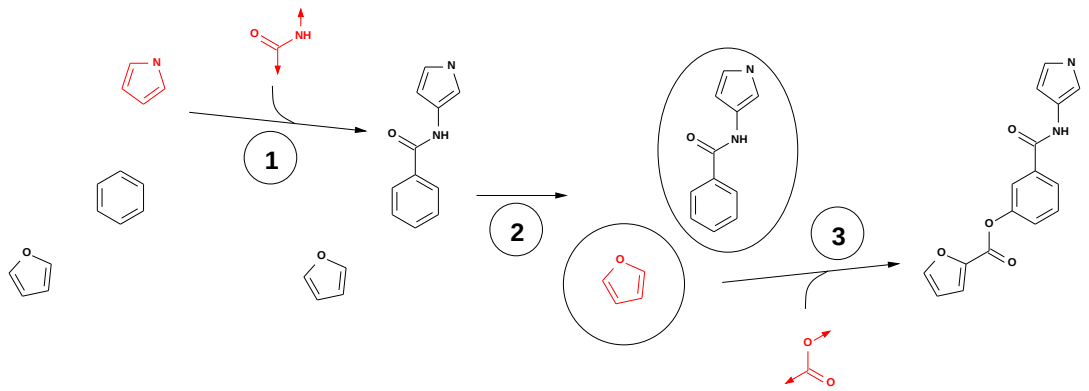


Figure 4: Linker placement

Before the scores are calculated the molecule is evaluated if it fulfills the user-defined constraints (3.5). If the latter test is passed successfully, the scores are calculated for the assembled molecule (see 3.4), the linkers are removed and a new attempt of linking the docked fragments is performed. The maximal number of linking trials is an user-defined parameter specified in the input file (`NUMLINKERITERATIONS [integer]`, see 4.6.2.2).

3.3.3 Scoring

During the energy or score calculation the current scores are compared to the most favorable score found while trying to assemble a molecule from a particular set of docked fragments (discussed in more detail below) If the weighted-sum approach is used and the current score is less favorable that the most favorable found previously (i.e. cannot be improved even when the remaining scores are at a minimum or maximum depending on the scoring function), the calculation is stopped, linkers are removed and a new trial is started to connect the fragments. This ensures that for a particular molecule only the necessary calculations are performed. When using Pareto-based scoring all individual scoring functions are calculated. Once the scores are calculated and the resulting total score is more favorable than the most favorable found previously (i.e. Pareto-dominant if Pareto based scoring is used), the molecule is checked for forbidden substructures (5.2.6).

The build-up trials are stored in a tabu list for a given set of docked fragments to avoid unnecessary computation of scores. At the end of the linking trials of this set of docked fragments only the linker combination with the lowest score is restored and kept. The tabu list of solutions is stored transiently for a given set of docked fragments and is removed after these trials (not stored globally).

3.4 Scoring

3.4.1 Scoring functions

Different scoring functions are implemented in GANDI accounting for various properties of the generated ligands and can be switched on and off at the user's request. A 3D structural (3.4.1.2) and a 2D fingerprint-based (3.4.1.3) scoring function provide the user with the possibility to steer the optimization process towards an user-supplied target structure. An additional fingerprint-based scoring function that optimizes the similarity to a target molecule is based on the so called *molecular quantum numbers*. Furthermore a force field-based scoring function (3.4.1.1) serves to calculate the interaction and internal energies of the generated molecule. The latter should always be active as it removes molecules with unlikely binding mode and geometry. Another available scoring function is pharmacophore-based (3.4.1.5), and similar to the 3D or 2D scoring function mentioned before, allow the user to specifiy features that should occur in a molecule. In addition, GANDI has the possibility to produce molecules that are optimized for their burial in the binding site (3.4.1.6).

3.4.1.1 Force-field based scoring This scoring function is a sum of inter and intra van der Waals and electrostatic terms as well as angle and dihedral penalty for newly formed and distorted bonds. The latter are calculated for atoms involved in newly formed

bonds only.

$$E_{ff} = E_{vdW}^{inter} + E_{elec}^{inter} + E_{vdW}^{intra} + E_{elec}^{intra} + E_{penalty}^{intra} \tag{6}$$

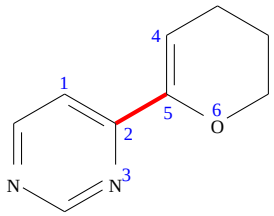
The terms labeled *inter* are the van der Waals and coulombic energies, using a distance-dependent dielectric, between all ligand atoms and the receptor.

$$E_{\text{vdW}} = \sum_{i<j} \sqrt{\varepsilon_i \varepsilon_j} \left\{ \left(\frac{R_i^{\text{vdW}} + R_j^{\text{vdW}}}{r_{ij}} \right)^{12} - 2 \left(\frac{R_i^{\text{vdW}} + R_j^{\text{vdW}}}{r_{ij}} \right)^{6} \right\} \tag{7}$$

$$E_{\text{elect}} = 332 \sum_{i<j} \frac{q_i q_j}{\epsilon_{\text{int}} r_{ij}^2} \tag{8}$$

Where R_i^{vdW},R_j^{vdW} are the van der Waals radii of atom i and j, ε_i is the minimum of the van der Waals potential between two atoms of type i at optimal distance of $2 \cdot R_i^{\text{vdW}}$, and r_{ij} is the distance between atoms i and j in Å. q_i and q_i are the partial charges in electronic units of atoms i and j, respectively, r_{ij} is the interatomic distance in Å. Van der Waals and electrostatic 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 *intra* terms are the internal interaction energies between groups (linkers and docked fragments) of the ligand, and angle and dihedral penalty for newly formed bonds. The sum of these intra terms can be scaled by a constant factor to lessen the effect of minor distortions of the added linkers in the input file (`INTENERWEIGHT [real]`). Geometries are checked for angles where two of three atoms are part of a newly formed bond. Dihedrals are calculated with the central two atoms stemming from the newly formed bond (Figure 1). For every angle and dihedral value which is outside of the allowed region as defined in the parameter file, a bond penalty is added to the force field based score (`BONDVIOLATIONPENALTY [real]` with the default being 5 kcal/mol). Both the dihedral and angle check can be switched off separately in the parameter file (see 5.2.3 and 5.2.4).



checked angles and dihedrals for atom pairs:	
1 – 2 – 5	1 – 2 – 5 – 4
3 – 2 – 5	1 – 2 – 5 – 6
2 – 5 – 4	3 – 2 – 5 – 6
2 – 5 – 6	3 – 2 – 5 – 4

Table 1: The newly formed bond is in red

A further option of the force field-based scoring function is to divide the calculated force field energy by the number of heavy atoms (`LIGEFF heavyatom`) or the molecular weight (`LIGEFF weight`). This allows the user to optimize molecules with GANDI according to *ligand efficiency*, i.e. a measure of how much atoms contribute to the energy on average. This option counteracts at least in part the tendency of pairwise potentials such as the van der Waals term to favor large molecules.

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 (as output e.g. by SEED):

```
TRIPOS<SUBSTRUCTURE>
   1 MOL1 1 GROUP **** **** -10.87 0
```

3.4.1.2 3D similarity based scoring The second scoring function measures the similarity between the newly assembled ligand and a user supplied template molecule (`TEMPLATE [MOL2FILE]`).

$$Sim_{3D}(A,B) = \frac{S_{AB}}{max(S_{AA}, S_{BB})} \quad (9)$$

$$S_{XY} = \sum_{i \in X} \sum_{j \in Y} w_{t_i t_j} e^{-\gamma r_{ij}^2} \quad (10)$$

where r_{ij} is the distance between two atoms ($i \in$ molecule X, $j \in$ molecule Y), $w_{t_i t_j}$ is a matrix whose coefficients reflect the similarity between element types (an unit matrix is currently used), and γ is a coefficient which acts on the broadness of the distribution of the positions (`SIMEXPFACTOR [real]`). This scoring function is identical to the one used for evaluating the similarity between two individuals of a population (see 3.6).

3.4.1.3 2D similarity based scoring This scoring function 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:

$$Sim_{2D} = \frac{\sum_{i=1}^{n} x_{iA} x_{iB}}{\sum_{i=1}^{n} x_{iA}^2 + \sum_{i=1}^{n} x_{iB}^2 - \sum_{i=1}^{n} x_{iA} x_{iB}} \quad (11)$$

where x_{iA} denotes the ith fingerprint entry of molecule A.

To 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 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, i.e. additive descriptor terms are assumed.

3.4.1.4 Molecular quantum numbers Molecular quantum numbers (MQN) are descriptors derived from the graph structure of the respective molecules as developed by Nguyen et al. [14]. A fingerprint of 42 of these integer values descriptors is calculated by GANDI and the distance to the user-defined template molecule is calculated with the Manhatten distance (sum of absolute descriptor differences):

$$Dist_{MQN} = \sum_{i=1}^{42} |f_i^{template} - f_i^{GANDImolecule}| \quad (12)$$

where f_i is the ith descriptor. Contrary to the similarities described before, the Manhattan distance decreases with increasing similarity to the template molecule.

3.4.1.5 Pharmacophore scoring The pharmacophore scoring function implemented in GANDI covers the commonly used pharmacophore definitions which are listed in Table 2. All pharmacophore point definitions follow the same scheme except for the hydrogen

Name	Tag (input file)	required format in input file
Hydrophobic group†	HYDROPHOBIC	[Tag] [Radius] [Coordinates]
Hydrogen bond acceptor †	HBACCEPTOR	[Distance] [Angle] [Atom ID]
Hydrogen bond donor †	HBDONOR	[Distance] [Angle] [Atom ID]
Weak hydrogen bond donor †	WEAKHBDONOR	[Distance] [Angle] [Atom ID]
Weak hydrogen bond acceptor †	WEAKHBACCEPTOR	[Distance] [Angle] [Atom ID]
Positive (formal) charged atom	POSCHARGE	[Tag] [Radius] [Coordinates]
Negative (formal) charged atom	NEGCHARGE	[Tag] [Radius] [Coordinates]
Centroid of ring	RING	[Tag] [Radius] [Coordinates]
Any atom	ANY	[Tag] [Radius] [Coordinates]
Custom definition †	CUSTOM	[Tag] [Radius] [Coordinates]
Exclusion volume	EXCLUSION	[Tag] [Radius] [Coordinates]

Table 2: Pharmacophore definitions (coordinates have to be whitespace separated). The pharmacophore points marked with † are defined in the parameter file (5.2.7) the others are computed by GANDI.

bond donors and acceptors (weak and strong). For the latter the geometry of the hydrogen bond is defined with the hydrogen – acceptor distance, the hydrogen bond angle (donor heavy atom – hydrogen – acceptor heavy atom) and the sequence ID of the receptor atom (either the donor hydrogen or the acceptor atom). If an acceptor is requested in the generated molecules the `[Atom ID]` would be the donor hydrogen of the receptor to which the distance would be calculated. To obtain the sequence ID of the acceptor atom examine the MOL2 file of the receptor where the sequence ID is the first value in the line of the "TRIPOS<ATOM>" section, e.g. `186`:

```
186 O -3.15200 22.54300 6.39000 O 11 GLY_11 -0.555374
```

Pharmacophore features can be combined with the "`AND`" tag (all features have to match), or alternative features can be defined where at least one has to match with the "`OR`" tag (for examples see section 4). The pharmacophore similarity is calculated with

$$Sim_{PH4} = \frac{\#\text{ matched points}}{\text{total }\#\text{ of points defined}} \tag{13}$$

The possible values of Sim_{PH4} are thus $\in [0, 1]$ and "OR"ed points count as one match even if multiple alternative points are satisifed.

In addition to using the pharmacophore module as a scoring function (`PH4WEIGHT [real]`), the latter can also be used to filter molecules (`PH4WEIGHT` FILTER). If the latter is selected the pharmacophore module is not used as a scoring function, but works in such a way that molecules are discarded if they do not conform to the defined pharmacophore (see section 4.3).

3.4.1.6 Burial of molecules Another scoring function implemented in GANDI evaluates the burial of the generated molecule with a geometric function developped to detect binding sites. The detection of binding sites as indentations of the protein surface originates from the observation that in most cases the binding site of a small molecule is the deepest pocket. The re-implemented LIGSITE [15] approach works in such a way that a grid is generated encompassing the protein and grid points close to protein atoms (within the van der Waals radius of that atom) are marked as belonging to the protein. For every non-protein grid point the x-, y- and z-axes as well as the four diagonals are scanned for "protein-solvent-protein events" (PSP). A PSP event occurs if in both directions for one scanning orientation (e.g. x-axis) a grid point is found which belongs to the protein (Figure 5). Each grid point is then assigned the number of PSP events that occur while scanning the seven direction leading to a maximal buriedness value of 7. In order to determine the buriedness of GANDI molecules, the buriedness is calculated for every atom ($Burial_{atom_i}$) from the surrounding grid points with trilinear interpolation and averaged over the number of ligand atoms (see Equation 14). However, negative burial values can occur as grid points belonging to the protein are given a burial score of -1.

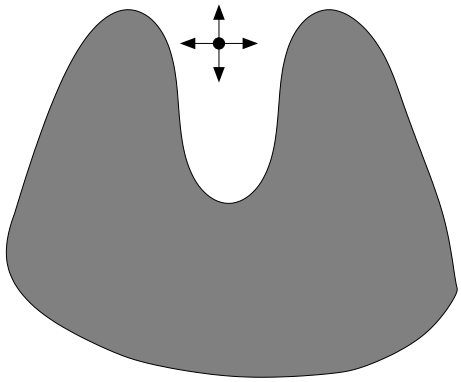


Figure 5: Burial with LIGSITE shown in x-, y-direction for a single grid point, yielding one PSP event along the x-axis

$$Burial = \frac{\sum_{i=1}^{n_{atoms}} Burial_{atom_i}}{n_{atoms} * 7} \tag{14}$$

3.4.2 Multi-objective optimization

3.4.2.1 Weighted-sum scoring There exist numerous approaches of how scores measuring different traits can be used to assign an overall score or rank to an object under investigation ([16]). In the weighted sum approach (`SCORINGMODE WEIGHTEDSUM`) 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:

$$S_{total} = w_{ff}E_{ff} - w_{3D}Sim_{3D} - w_{2D}Sim_{2D} - w_{Burial}Burial - w_{PH4}Sim_{PH4} + w_{MQN}Dist_{MQN} \tag{15}$$

Where w are the respective weights for the scoring function terms described before (see also 4.6.3.3). The minus signs for the similarity and burial scores stem from the fact that

the similarities, contrary to the force field energy and MQN-distance, increase with rising fitness. The user can choose to optimize the ligand building process according to any combination of the 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 (4.6.3.3), the corresponding scoring function term will not be computed and hence not used during optimization.

3.4.2.2 Pareto-based scoring Another approach is the use of the concept of Pareto optimality (`SCORINGMODE PARETO`). An individual A *dominates* individual B in the Pareto sense, if the scores of all scoring functions of individual A are equal or lower (at least one) than the corresponding scores of individual B (assuming a minimizing problem):

$$\forall f_i : f_i(A) \leq f_i(B) \wedge \exists f_j : f_j(A) < f_j(B) \tag{16}$$

An individual is *non-dominated* if there is no other individual in the current population that dominates the former. *Pareto-optimal* individuals are the non-dominated solutions of the entire search space. The method of Fonseca and Fleming [17] is used to calculate a rank of an individual A in a population, where the Pareto rank equals the number of individuals a given individual is dominated by resulting in lower ranks for fitter individuals. The Pareto-based approach is attractive as it does not require the tuning of any scoring function weights as is required for the weighted-sum approach. Individual scoring function are used if their weights are not equal to zero, where the value of the weight does not have any influence on the outcome (weights can be set to "1", see 3.4).

The default scoring mode is set to Pareto, which can be changed in the input file (`SCORINGMODE [PARETO or WEIGHTEDSUM]`). The optimization with this scoring scheme can yield molecules that represent compromises, i.e. a molecule with one very favorable score but poor performance with respect to the other activated functions. This is different in the weighted-sum approach where one function can be made the driving force of the optimization when setting the weights appropriately.

If GANDI is run with Pareto scoring (and more than one scoring function), an archive is kept of the non-dominated solutions found up to the current step of the optimization. The archive is examined at every iteration and new individuals are either added or removed if they represent non-dominated solutions. Molecules from the archive are also saved to disk at the end of the optimization procedure (labeled with the tag "`arch`").

3.5 Constraints

In addition to the discussed scoring functions, GANDI has the possibility to filter molecules according to the characteristics defined in Table 3. The hydrogen bond donors and acceptors correspond to the pharmacophore definitions (see 5.2.7). For the acceptors the number of heavy atoms and for the donors the hydrogens are counted. Rings are the number of smallest set of rings (SSSR's). Rotatable bonds are all single bonds which are not part of a ring, not between sp2 hybridized groups (e.g. single bonds between two phenyl rings are not counted as rotatable) and bonds connecting an atom with no other atom neighbor (e.g. halogens). Similar to the filtering with the pharmacophore module mentioned earlier, too stringent constraint defintions might lead to a difficult search for feasible molecules and no viable solutions might be found during the optimization. If the latter should occur, the constraints should be softened and GANDI rerun.

Name	Tag (input file)	required format in input file
Molecular weight	MW	MW [min] [max]
# hydrogen bond donors	HBDONOR	HBDONOR [min] [max]
# hydrogen bond acceptors	HBACCEPTOR	HBACCEPTOR [min] [max]
Formal charge	FCHARGE	FCHARGE [min] [max]
# rotatable bonds	ROTB	ROTB [min] [max]
# rings	RING	RING [min] [max]

Table 3: Possible constraints defined in the input.

3.6 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 (referred to as $(\mu + \lambda)$ selection). The merging procedure starts with the most favorable 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 more favorable score. The similarity between two individuals is calculated with Equation 9. The similarity cutoff is specified in the input file (`SIMCUT [real]` and see 4.6.3.2). 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 merging.

3.7 Convergence check

Two checks are performed if the genetic algorithm has converged. A first and simple check is carried out every fixed amount of steps (`INTCONVSTEP [integer]`) and evaluates how many individuals contain identical docked fragment poses (i.e. the same genes). This is done due to the fact that the genetic algorithm can converge to few solution with different linkers. An upper limit can be specified that signifies the amount of shared docked fragment poses in the current population (`INTMAXCONV [real]`), which is calculated with Equation 17 by comparing all individuals to one another.

$$ratio = \frac{2 * \#\text{ identical genes}}{n_{individuals} * (n_{individuals} - 1) * n_{ligandsize}} \tag{17}$$

Where $n_{individuals}$ is the number of individuals, $n_{ligandsize}$ equals the number of docked fragment poses per individual and # identical genes is the number of identical genes values found when comparing all individuals to one another. The simple convergence check thus works on the level of a single island and stops the optimization before converging too much towards a single solution. This can at least in part be prevented by exchanging individuals between islands (see below), which introduces new genetic information to an island. If intra-island convergence occurs, the optimization for this particular island

is stopped, however the remaining islands proceed up to the next migration step (see below), where the genetic algorithm is stopped (i.e. the convergence of a single island halts the overall optimization process). This form of convergence can occur when few fragments are docked into the receptor binding site and only few of these present viable sets to be connected.

A second, computationally more expensive test for convergence is performed before exchanging individuals (see below) and examines the content of islands to one another. This test is carried out every fixed iteration step (`EXCHANGESTEP [integer]`). For this step all islands are compared to one another and the ratio of similar individuals in two islands is calculated with Equation 9. The cutoff what constitutes a pair of similar individuals is an input parameter (`CONVSIMCUT [real]`) as well as the maximal ratio that are allowed to be similar (`MAXCONV [real]`). The optimization is stopped for all islands if two islands are found that exceed the allowed ratio of similar individuals.

3.8 Migration

The main difference between a classic and an island genetic algorithm is, that in the latter multiple populations are evolved simultaneously in separated islands. After a user-defined number of iterations (4.6.2.5), 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 (`EXCHANGEMODE ALL`), which can lead to homogenization of the genetic material of the islands if the number of individuals that are exchanged is set too high, i.e. all islands become essentially the same.

- Neighbor
  Only neighboring islands exchange individuals (`EXCHANGEMODE NEIGHBOR`).

- Random
  Every island chooses one island randomly with which individuals are exchanged (`EXCHANGEMODE RANDOM`).

Running GANDI with n number of islands that do not exchange individuals during the entire course of the optimization equals to running n "standard" genetic algorithms.

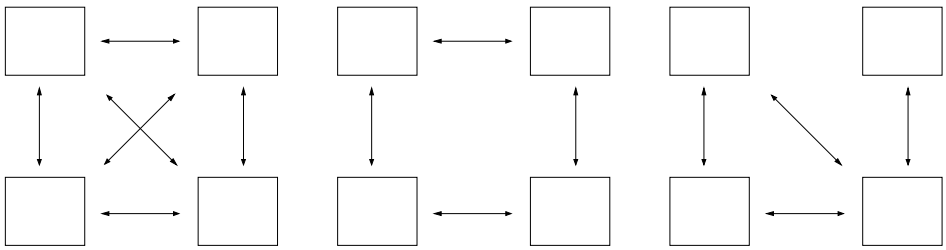


Figure 6: Exchange models: all with all, neighbor and random

3.9 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. The information on which docked fragments and linkers were used during the build-up of the molecule is stored in the header of the MOL2-file. This includes the energy values read in from the MOL2-files for the docked fragments and the calculated interaction energies between receptor and linkers for the latter. Intermediate molecules can be saved to disk during optimiziation (`SAVINGSTEP [integer]`). Conversely to the saving of the final molecules where the entire set of molecules (i.e. over all islands) is clustered (and re-ranked if Pareto-based scoring is enabled), the intermediate molecules are neither clustered nor re-ranked over the entire set. As mentioned earlier, during Pareto-based scoring an archive of all non-dominated solutions is maintained separately for every island and is also written to disk at the end of optimization (labelled with the tag `arch`).

4 Input file examples

Some helpful hints on the build-up of the input file (for complete examples see the test cases supplied with the distribution):

- Lines in the input file are parsed one line at a time and if a # is encountered at the beginning of a whitespace separated word (or the beginning of the line) the remainder of the line is discarded.
- If a tag is defined twice in the input file, which should be avoided, the first value read will be overwritten with the second.
- The tags with which variables can be set are not case sensitive (but file names are).

Whenever a certain tag is not specified in the input file, GANDI will use its default value listed in the appendix section B, e.g. GANDI is run for the default number of iterations and islands using only the force-field based energy function.

4.1 Creating the potential energy grids

Table 4 shows a basic example of an input file how it might be used when running GANDI the first time in a project. Note that only the receptor, the docked fragments and the linkers are defined, which have to be defined in every input file (no default). For the docked fragments and linkers no additional specification are made and GANDI thus assumes that all connection vectors are valid, i.e. all heavy atom – hydrogen atom vectors are potentially used both for the linkers and docked fragments. In addition, no tags are specified to write the grids, which is not necessary as this is the default.

```
4.6.4.1   RECEPTOR ./protein/1ke5.mol2
          10
          11
          .
          .
          137
          138
          END
          #
          #
          #
4.6.6.1   DOCKEDFRAGMENTS
          fragment_1_clus_pproc.mol2
          fragment_2_clus_pproc.mol2
          .
          .
          fragment_n_clus_pproc.mol2
          END
          #
          #
4.6.6.3   LINKERS
          linker_1.mol2
          linker_2.mol2
          .
          .
          linker_m.mol2
          END
```

Table 4: A basic input file

4.2 Using a template to bias the design

Table 5 is an example where GANDI was run previously and the potential energy grids do exists and do not have to be re-calculated again. However, as mentioned previously the receptor, docked fragments and the linkers have to be specified again (omitted here for brevity, but would include the same definitions as in the previous example in Table 4). In this GANDI run a template molecules is provided

which would bias the design towards the user-defined inhibitor. This run includes the 3D similarity based scoring function (default is not to use it) and the force-field based scoring function, which is not specified in the input as it is turned on by default. The default scoring mode is Pareto-based scoring. Alternatively, the $Dist_{MQN}$ or Sim_{2D} scoring function (3.4.1.3 and 3.4.1.4) could be used to bias the design.

	`# Reading the potential energy grids`
4.6.4.2	`VDWGRIDACCESS r`
4.6.4.3	`CLBGRIDACCESS r`
	`#`
	`#`
4.6.3.3	`SIM3DWEIGHT 1`
	`TEMPLATE ./1ke5_ligand.mol2`
	`#`

Table 5: Using a template to bias the design

4.3 Pharmacophore example

Table 6 (left panel) shows an example of a three point pharmacophore defined for a protease. The `ANY` tag corresponds to any atom that should be at that location (here the S1 pocket) and the `EXCLUSION` avoids that molecules are built towards a specific pocket (S1'). The third point consists of three hydrogen bond donor or acceptors that should be satisfied. However, only one of the latter has to be satisified to contribute positively and molecules that conform to all three are not ranked higher. As mentioned

```
.
.
PH4WEIGHT 1
#
PH4
ANY 1.5 -3.3550 -2.7610 -8.0110
EXCLUSION 5.5 6.7360 -0.6920 -5.9810
HBACCEPTOR 3 120 2769 OR
HBDONOR 3 120 2774 OR
HBACCEPTOR 3 120 2787
END
.
.
```

```
.
.
PH4WEIGHT FILTER
#
PH4
HBACCEPTOR 3 120 2769 OR
HBDONOR 3 120 2774 OR
HBACCEPTOR 3 120 2787
END
.
.
```

Table 6: Pharmacophore based scoring (left) and filtering (right panel).

earlier, the pharmacophore module can also be used to filter out molecules that do not conform to the latter (an example input is shown in Table 6, right). Individuals that do not conform to the pharmacophore are immediately removed. If the pharmacophore is defined too strictly (i.e. defining too many features or to strict distance cutoffs), the genetic algorithm might find it difficult to find viable solutions. Should this occur one might try to loosen the angle and distance cutoffs or use fewer features to filter molecules.

4.4 Working with zones

The main reason for working with zones is to obtain molecules that are made up of fragments binding to user-defined pockets (i.e. zone). This is thus another approach to guide the design of GANDI by requesting fragments to be included in the ligand. When zones are defined, GANDI reserves genes for these mandatory zones and these genes only sample the docked fragments of that particular zone, thus making the search a lot more efficient (for the advantage of working with zones see also 6.1.4).

The first step to working with zones is to define which are mandatory with the tag `MANDATORYZONES` followed by the number of zones and the zone index. Assigning a docked fragment position to a zone is simply done by adding a line immediately after the file name and before the fragment vector definition starting with `Z` and the zone index. One can also define fewer zones than the number of docked fragments per ligand, e.g. by only defining one zone as mandatory in a subpocket and the rest as undefined (not adding a `Z [integer]` line). Docked fragments defined as belonging to a particular zone will be reserved for a single gene and will thus never be connected to one another. The remaining "undefined" docked fragments are then used for the remaining genes, e.g. when using one mandatory zone with ligand size equals to 2 (two docked fragments are connected) would leave one gene to sample the undefined fragments.

4.6.6.2
4.6.6.1

```
MANDATORYZONES 3 1 2 3
#
DOCKEDFRAGMENT
Fragment1_zone1_clus_pproc.mol2
Z 1
Fragment2_zone1_clus_pproc.mol2
Z 1
:
:
Fragment1_zone2_clus_pproc.mol2
Z 2
Fragment2_zone2_clus_pproc.mol2
Z 2
:
:
Fragment1_zone3_clus_pproc.mol2
Z 3
Fragment2_zone3_clus_pproc.mol2
Z 3
:
:
END
```

Table 7: Example: Defining zones

4.5 Using constraints

Using constraints (hard cutoff filters) is simply done by specifying the features that serve to discard molecules (see Table 8) As mentioned earlier for the pharmacophore filtering, too restrictive settings can lead to a difficult search for feasible molecules and the cutoffs might have to be relaxed if no or only few molecules are produced.

```
:
:
CONSTRAINT
ROTB 0 10
RING 0 10
FCHARGE -1 1
END
:
:
```

Table 8: Example: Defining zones

4.6 Input values

4.6.1 Running GANDI in parallel

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_[compiler]_[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 7). The number of threads (`NUMTHREADS [integer]`) used by GANDI should 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 9). It should be noted that for test cases with large amounts of data (docked fragments and linkers), memory access can become the rate limiting step. Increasing the number of threads (e.g. more than 4) can then even lead to a decrease in speedup.

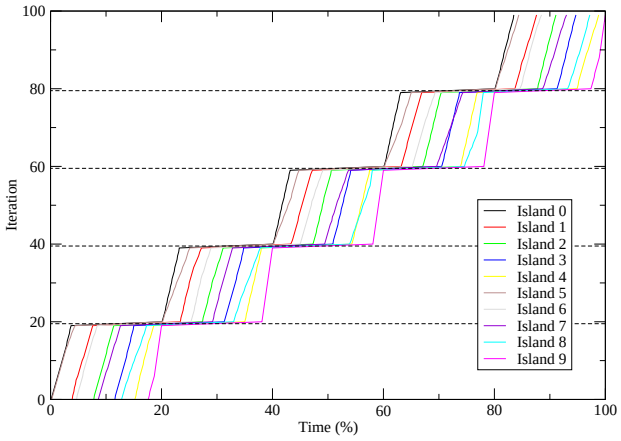


Figure 7: Running GANDI with 2 threads, 10 islands and 100 iterations in total with exchange of individuals every 20 steps. Synchronization of the optimization occurs before the exchange of individuals.

Number of threads	Speedup
1	1.00
2	1.64
3	1.79
4	2.19
5	2.41

Table 9: 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.

4.6.2 Optimizer

The following input values control the extend to which chemical space is sampled with GANDI. In general a "large" setup with multiple islands, large number of individuals and many iterations will lead to an increase of the sampling and should thus lead to improve results.

4.6.2.1 Population `NUMISLANDS [integer]` defines how many islands, corresponding to simultaneous genetic algorithms, should be run at the same time. `NUMINDIVIDUALS [integer]` specifies how many individuals a single island should be made up of.

4.6.2.2 Iterations An important setting of GANDI is how long and how thorough chemical space should be searched, which is predominantely controlled by three numbers: the number of iterations for the genetic algorithm `NUMITERATIONS [integer]`, the tabu search `NUMLINKERINTERATIONS [integer]` and how often GANDI should try generating a single molecule with non overlapping docked fragments `NUMINDITERATIONS [integer]`. The convergence checks explained below help to stop the optimization process before GANDI converges too much towards a single solution in chemical space.

4.6.2.3 Size The ligand size `LIGANDSIZE [integer]`, which remains constant throughout the entire run, corresponds to the number of docked fragments that should be connected together.

4.6.2.4 Reproduction Possible values for the selection mode are tournament selection (`SELECTIONMODE TOURNAMENT`) and rank-based roulette wheel selection (`SELECTIONMODE RANKROULETTE`). responsible for selecting individuals for reproduction.

4.6.2.5 Migration The exchange of individuals between islands is defined by which islands exchange individuals with one another (`EXCHANGEMODE [ALL/NEIGHBOR/RANDOM]`), after how many iterations of the genetic algorithm exchanges occur (`EXCHANGESTEP [integer]`) and how many individuals should migrate (`EXCHANGERATIO [real]`, see 3.8). Migration can help an island to escape a local minima by introducing new genetic information (swapping individuals between islands). However, if the exchange occurs too often or too many individuals are exchanged, homogenization of the genetic information of two islands can occur (i.e. they become too similar). The latter is checked in an inter-island convergence test described below.

4.6.2.6 Convergence Two convergence checks are implemented in GANDI (intra- and inter-island) that serve as termination criteria too avoid that either a single islands is dominated by a single solution or that two islands become too similar (see also 3.7. The intra-island convergence test is controlled by two parameters, namely how often the check should be performed (at what iteration intervall `INTCONVSTEP [integer]`) and how many genes of two individuals on average can be identical (`INTMAXCONV[real]`). The second, inter-island test for convergence is controlled by the ratio of individuals of two islands that can be similar (`MAXCONV [real]`) and the cutoff of what constitutes similar molecules (`CONVSIMCUT [real]`) calculated with Equation 9.

4.6.2.7 Random seed The random seed (`SEED [integer]`) is used to initialize the random number generators responsible for selecting islands in the random exchange mode (3.8), individuals during the reproduction procedure (3.2) and linkers during the tabu search (3.3). As genetic algorithms are stochastic processes, it can be of advantage to re-run GANDI with the same setup but different random seeds.

4.6.3 Varia

4.6.3.1 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 (`SAVESTEP [integer]`).

4.6.3.2 Similarity The similarity cutoff (`SIMCUT [real]`) determines the maximal similarity of two molecules residing in the same island (see 3.6) determined with equation 9 with the exponential factor (`SIMEXPFACTOR [real]`) corresponding to γ. This value should be decreased, if GANDI produces molecules that are too similar, i.e. leading to more diverse islands. The maximal squared distance between any two atoms for which the similarity should be calculated is specified with (`SIMSQDISTCUT [real]`). 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 3.6), which decreases the computational cost without loosing too much accuracy. The final similarity cutoff (`FSIMCUT [real]`) determines if an inter-island clustering should be performed after the actual optimization procedure. One of two individuals of distinct islands is removed (and not stored to disk) if they are more similar than the user defined cutoff value. This last clustering step is only performed if the final similarity cutoff value is > 0.

4.6.3.3 Scoring The weights of the scoring function terms of equation 15 are:

Tag	Value	Scoring function term
EFFWEIGHT	[real]	Force-field based energy
SIM3DWEIGHT	[real]	3D structural similarity
SIM2DWEIGHT	[real]	2D fingerprint similarity
PH4WEIGHT	[real]	Pharmacophore similarity
BURIALWEIGHT	[real]	Burial
MQNWEIGHT	[real]	Fingerprint distance

Table 10: Activating scoring functions with weights

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 $w_{3D} = 0$, $w_{2D} = 0$. and $w_{MQN} = 0$ Additional parameters can be set for the force-field based scoring function. These include weights of the internal (`INTENERWEIGHT [real]`) and linker energy (`LINKERENERWEIGHT [real]`). The former encompasses van der Waals, coulombic, angle and dihedral penalties and the latter only van der Waals and coulombic energies, see also section 3.4). The internal energy comprises all interactions between all possible pairs of docked fragments and linkers, as well as angle and dihedral penalties

arising from newly formed bonds. The linker energy is the interaction energy between linker fragments and the receptor. The internal and linker energies are not calculated if their corresponding weight factor is set to zero. For Pareto-based scoring the actual weights do not have any particular meaning and can thus be set to "1".

4.6.3.4 Constraints Constraints were previously discussed in section 3.5 and listed in Table 3. The constraints section in the input file starts with the `CONSTRAINTS` and ends with the `END` tag. In between these two constraints are defined by first specifying what property should be limited followed by the minium and maximum value (both have to be defined).

4.6.3.5 Print level The print level (`PRINTLEVEL [integer]`) determines the output verbosity and increases with increasing number (from 0-4). Where values above 2 are used for debugging purposes.

4.6.3.6 Parameter file `PARAMETERFILE [file name]` specifies the location and name of the parameter file (see 5)

4.6.3.7 Output directory `OUTPUTDIR [name]` specifies the output folder to which the molecules will be saved. If the directory does not exist, GANDI will try to create it and will exit if the folder cannot be created or files cannot be written to it.

4.6.3.8 Variables GANDI allows the definition of string variables which will be replaced in sections following the defintion. The syntax for the variable definition is `VARIABLE NAME REPLACEMENT` and the subsequent use is (similar to Linux shell) `${NAME}`. Where the latter will be replaced by the term defined with `REPLACEMENT` from the definition. Please not that the dolar sign and brackets are mandatory for the replacement.

4.6.4 Receptor

4.6.4.1 Binding site Defines the location and name of the receptor MOL2-file as well as the list of binding site residues terminated by and `END` tag (`RECEPTOR [filename] [residue list] END`). The residues are defined by their sequence ID defined in the MOL2 file, e.g. the first number ("1") in the second line:

```
TRIPOS<SUBSTRUCTURE>
   1 MET_23 1 GROUP **** A **** 1
```

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. For an example of how the receptor section is defined see section 4.

4.6.4.2 Van der Waals energy grid GANDI defines a margin (`VDWGRIDMARGIN [real]`) around the binding site residues for which the potential energy grid will be calculated. In addition, the resolution of the grid can be defined (`VDWGRIDSPACING [real]`). resolution. The grid has to be only written once (`VDWGRIDACCESS w`) for every protein,

after which it can be simply read (`VDWGRIDACCESS r`), which reduces the computational cost.

4.6.4.3 Coulombic energy grid The coulombic energy grid is handled similar as the van der Waals grid (`CLBGRIDMARGIN [real]`, `CLBGRIDSPACING [real]` and `CLBGRIDACCESS [w/r]` with the exception that the value of the dielectric constant can be defined (`DIELCONST [real]`) and whether or not GANDI should use a distance-dependent dielectric model to account for electrostatic interactions (`DISTDEPDIEL [y/n]`).

4.6.5 Cutoff values

4.6.5.1 Heavy atom clash The number of small and severe clashes between the heavy atoms of two docked fragments is calculated with equation 18 and 19 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 (`CLASHSCALFACTOR [real]` and `SEVCLASHSCALFACTOR [real]`) 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 (`MAXNUMCLASH [integer]` and `MAXNUMSEVCLASH [integer]`).

$$\text{small if: } r_{ij}^2 < r_{small}^2 = (c_{small} * (R_i^{vdW} + R_j^{vdW}))^2 \quad (18)$$

$$\text{severe if: } r_{ij}^2 < (c_{severe} * r_{small})^2 \quad (19)$$

4.6.5.2 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.

4.6.5.3 Vector tolerances and linker lookup table GANDI builds a look-up table with all angles and distances of all linker vectors on start-up. For the linker, distances (and angles) between the two vector origins (i.e. heavy atoms) are calculated and stored in a table for faster look-up (bin sizes defined by `TOLERANCEANGLE [real]`, `TOLERANCEDIHEDRAL [real]`, `TOLERANCESQDIST [real]`). In addition, if both angles of a linker connection are above a certain cutoff (default 175 degrees, `COLINEARITY [real]` to change), the dihedral angle is assumed to be colinear and the dihedral angle is set to "0". The same check is performed when connections of docked fragments are evalued.

The distances and angles of all pairs of connection vectors for all sets of docked fragments are calculated 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). In order to provide a more accurate result reflecting

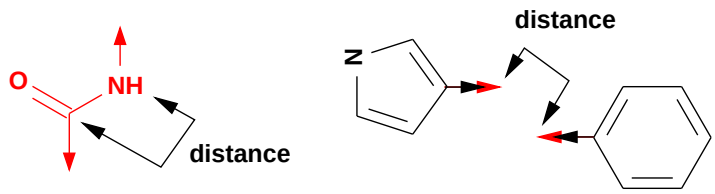


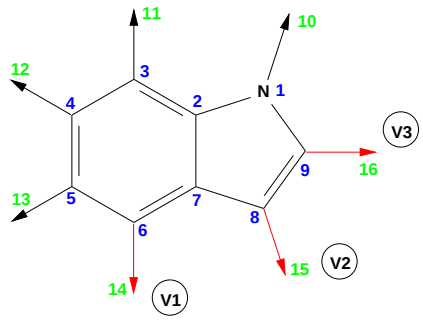
Table 11: Linker (left) and docked fragment (right) distances, with vector extension (red arrows) for docked fragments

heavy atom – heavy atom distances, the heavy → hydrogen atom vectors of docked fragments are expanded to *vector length* (`VECTORLENGTH [real]`) and the distances between the two new positions is calculated and used together with the angle constraints to search the linker look-up table (see table 11).

4.6.6 Docked Fragments and Linkers

4.6.6.1 Docked Fragment Section The list of docked fragments to be used is defined with the tag `DOCKEDFRAGMENTS`, followed by the list of files and terminated with an `END` tag (see Table 4). No defaults exists for the list of docked fragments, i.e. the list of docked fragments has to be defined in every input file.

If only the names of the fragment files are specified GANDI assumes that all possible connection vectors should be used for a given docked fragment. Two additional vector definitions are possible, which are the explicit definition and using the tag `DAIM`. For the former, the vectors are defined by the user by first providing the number of connection vectors to be used, followed by the connection vector definition 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). An explicit vector definition is given in Figure 12.



Filename	./indole.mol2
Number of vectors	3
Vector definition	6 14 8 15 9 16 (= V1, V2, V3)

Table 12: Explicit vector definition

When the file name of a docked fragment is followed by a `DAIM` tag, 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 tree methods of which vectors are used can be specified individually for every docked fragment (and linker discussed below), e.g. the user can choose to explicitly define the connection vectors for one fragment and not specify anything for the remaining fragments resulting in GANDI using all connection vectors for the latter and the explicityl defined for the former.

In addition the docked fragment section can include zone definitions for a given fragment (see Table 4.4 for an example). Zone definitions *must* proceed the connection vector definition of that fragment. Similar to the connection vector definiton discussed before, the user can choose to assign only a subset of fragments to zones. Fragments with no zone definition are assigned to zone "-1" by GANDI. These fragments can be connected to any fragment of any zone, in particular also other fragments from zone "-1". In contrast, fragments with an explicit zone definition will *never* be connected to one another, which is e.g. helpful to avoid that fragment poses in a particular protein subpocket are connected to one another.

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.

4.6.6.2 Further docked fragment options There additional options which influence how docked fragments are handled, but are defined outside of the docked fragment section discussed before. One option includes the definition of which zones are mandatory, i.e. fragments from which zones should be included in every ligand. This is defined by providing the tag `MANDATORYZONES` the number of zones followed by the zone IDs, where the latter are defined by the user in the docked fragment section discussed before (for an example see 4.4).

Another check includes the search for equivalent vectors (`MAPVECEQUIVALENCE [y/n]`) in docked fragments. If activated, GANDI will look for equivalent connections 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 [13] (Figure 8). If the similarity according to Equation 9 is ≥ 0.99 the two vectors are considered equivalent and the new connection vector is added to the fragment definition. This test is only helpful if a vectors are defined explicitely or with the `DAIM` option discussed before. 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.

A second test examines if the specified connection vectors are accessible (`MAPBUMPCHECK [n/y cutoff]`), 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").

4.6.6.3 Linker section The linker section is identical to the docked fragment section (4.6.6.1) described above. The main difference in the MOL2-files of the docked fragments

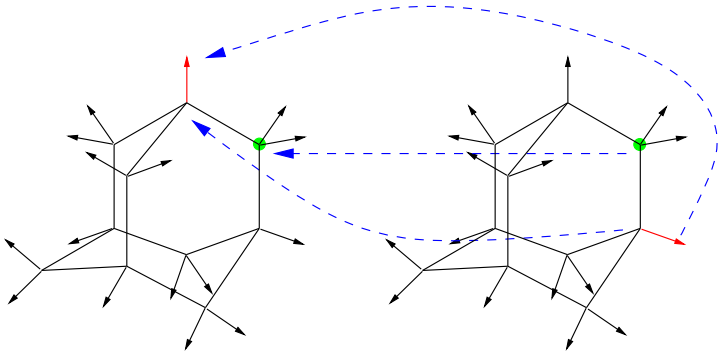


Figure 8: Superpositioning example adamantane: vectors under investigation are red, neighboring atoms are green.

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, bump checking or definition of mandatory zones is possible for linkers fragments.

5 Parameter File

5.1 Structure

```
# The following parameters are MSI CHARMm
#
5.2.1
#          atom     element         van der Waals         mass        atom
#          type     number    radius        energy_min                h.type
111
1          B        5         1.17          0.01          10.81       any
2          C        6         1.870         0.1410        12.01100    sp2
5          C5R      6         2.040         0.0500        12.01100    sp2
7          C5RP     6         2.040         0.0500        12.01100    bar
.          .        .         .             .             .           .
.          .        .         .             .             .           .
.          .        .         .             .             .           .
109        NO2      7         1.83          0.09          14.00670    sp2
110        NR56     7         1.830         0.0900        14.00670    sp2
111        BUMP     1000      2.060         0.2600        0           any
#
5.2.2
#          atom     atom      bond
#          type     type      length
1054
1          B        CT        1.58
2          B        N         1.42
3          B        OE        1.36
4          B        OT        1.325
.          .        .         .
.          .        .         .
.          .        .         .
1052       MZR      XCL       2.350
1053       MZR      XF        1.902
1054       MZR      XI        2.660
#
#
5.2.3
# Angle section
# number of angle parameters
# atom 1   atom 2   atom 3    equilibrium   tolerance
# h.type   h.type   h.type    angle
40
sp3        sp3      sp3       109.471       15.0
sp3        sp2      sp3       120.000       15.0
sp3        sp       sp3       180.000       15.0
sp2        sp3      sp3       109.471       15.0
.          .        .         .             .
.          .        .         .             .
.          .        .         .             .
bar        bar      sp3       120.000       15.0
bar        bar      sp2       120.000       15.0
bar        bar      bar       120.000       15.0
#
#
5.2.4
# Dihedral section
# number of dihedral parameters
# atom 1   atom 2   atom 3    atom 4        equilibrium   tolerance   equilibrium   tolerance ...
# h.type   h.type   h.type    h.type        angle                     angle
#
26
s          sp2      sp2       s             180.0         15.0        360.0         15.0
s          sp2      sp2       sp2           180.0         15.0        360.0         15.0
s          sp2      sp2       sp3           180.0         15.0        360.0         15.0
.          .        .         .             .             .           .             .
.          .        .         .             .             .           .             .
```

Table 13: Excerpt of parameter file

5.2.5

```
# number of forbidden connections
# list of forbidden connections
57
OC            OW
OC            OT
OC            OS
OC            OH2
OC            OE
OC            OC
OE            OW
OE            OT
OE            OS
OE            OH2
.             .
.             .
.             .
```

5.2.6

```
# number of forbidden subgraphs
# list of forbidden subgraphs
10
NC(R)(R)N
OC(R)(R)O
NC(R)(R)O
OC(R)(R)N
C(NC(=O))(=O)
N(X:X)(R)(R)(R)
N(X=X)(R)(R)(R)
N-N
N-O
O-O
.             .
.             .
.             .
```

5.2.7

```
# number of pharmacophore group definitions
# pharmacophore group definitions
11
hydrophobic   C
hbacceptor    O=X
hbacceptor    O(X)
hbacceptor    N(X)(X)(X)
hbacceptor    N(=X)X
hbacceptor    N(:X):X
hbdonor       HO
hbdonor       HN
hbdonor       HS
weakhbdonor   HC:N
weakhbdonor   HCC=O
```

Table 14: Excerpt of parameter file - part 2

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 Angle Section

Encompasses the (hard) cutoff values for the angle penalty mentioned earlier. For the angles the general form of these parameters, which uses the reduced atom hybridizaton type definition described before in the atom section 5.2.1, is:

```
[type1] [type2] [type3] [equilibrium angle] [angle tolerance]
```

Thus, angles that are outside the allowed region (absolute difference) lead to the addition of the user-defined penalty (`BONDVIOLATIONPENALTY [real]`) to the internal force-field energy.

$$|\text{equilibrium angle} - \text{measured angle}| \geq \text{angle tolerance} \tag{20}$$

In case no angle terms are specified (setting the number to "0" and commenting the parameter section), the angle penalty is ignored in the calculation. Angle penalties are calculated if the corresponding terms are specified in the parameter file. Individuals with missing angle terms are ignored during optimization and a list of missing angle terms is printed at the end of the optimization.

The use of a penalty function with a reduced set of atom types is due to usually strongly penalizing terms in force-field based scoring functions (such as CHARMm). The use of such functions is not ideal when fragment poses generated by SEED are used due to its discrete sampling procedure, i.e. molecules will always have (mostly) small angle and dihedral distortions which can usually easily be removed with a brief minimization.

5.2.4 Dihedral Section

Contains parameters for the dihedral penalty function where the same specifications and rules as in 5.2.3 apply. The difference here is that four atom types have to be defined and more than one angle and tolerance is allowed, e.g.:

```
sp3 sp3 sp3 sp3 60.0 15.0 120.0 15.0 180.0 15.0 ...
```

Identical to the angle section before, in case no dihedral terms are specified (setting the number to "0"), the dihedral penalty is ignored in the calculation.

5.2.5 Forbidden Connections

This section comprises a list of atom type pairs that should never be connected to one another.

5.2.6 Forbidden Subgraph

GANDI also checks for forbidden substructures in newly create molecules and if detected ignores these during optimization. GANDI uses a modified version of Peter Kolb's adaptation of the VF2 algorithm[18] to perform subgraph matching. One of the main differences is that GANDI does not check the entire molecule for forbidden graphs, but focuses only on newly formed bonds. This ensures that molecules are not eliminated from the search which contain fragments with forbidden subgraphs if the user decides to includes any for design reasons.

Bonds:	
-	single
=	double
#	triple
:	aromatic
%	amide
~	any
Atoms:	
	any atom symbol
R	any atom
X	any atom except hydrogens
c,x,n,s,o,p	ring atoms connected by aromatic bonds (lower case)
Other:	
(	start of new subgroup
)	end of subgroup
integers	ring opening and closures (1-9), e.g.: C1CCCCC1 is a cyclohexane

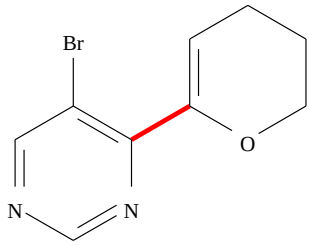
Table 15: SMILES definition

The way in which a user can define a forbidden subgraph is by means of SMILES notation in this section. The SMILES definition and the subgraph matching employed by GANDI is based upon atom elements and their connectivity (i.e. number and nature of connecting bonds). However GANDI does not perform any aromaticity, hybridization or net charge analysis. Subgraph matching algorithm proceeds in such a way that the very first bond of every SMILE is matched against all newly formed bonds. This ensures that the user can explicitly state which substructure should not be formed by a new bond without eliminating fragments from the search. Allowed characters in the GANDI SMILES notation are given in Table 15.

If no explicit bond symbol are stated between atom symbols, GANDI assumes single bonds. The only exception are aromatic ring atoms where the bonds are assumed to be aromatic (":"). An example is given in Table 16.

5.2.7 Pharmacophore definition

This section defines the pharmacophore groups which are used for the pharmacophore scoring function (3.4.1.5) and for the constraints (3.5), if defined by the user. The pharmacophores that can be defined by the user are hydrogen bond acceptors and donors (strong and weak), hydrophobic groups as well as one custom definition. The other pharmacophore definitions are computed internally by GANDI (ring centroids, "any" types etc.). The definition of a these user-defined pharmacophores is atom centered and only the first atom of the pharmacophore group definition corresponds to the respective group, e.g. for the definition `hbdonor HN` the hydrogen would be labeled as donor. The SMILES definitions are identical as mentioned for the forbidden substructure section (5.2.5). The



C(C(=C)N)(=C)(O)

Table 16: left: GANDI molecule with newly formed bonds in red; right: SMILES notation of subgraph which would avoid such a molecule. In the above SMILES notation the first bond would be between *C(C* matched against the new bond in red.

substructure matching when first assigning the pharmacophore group definitions requires that the matched atom in the molecule has not more connecting atoms as the corresponding SMILES atom (except for definitons consisting of a single atom such as `C`). With this matching procedure ternary amines can be distinguished from quaternary amines, e.g. `N(R)(R)(R)` would not match a quaternary amine and can thus be defined as a hydrogen bond acceptor. Currently, GANDI assigns pharmacophore points to fragments, linkers and the receptor before optimization to save time.

6 Help guide & Troubleshooting

6.1 Help guide

This section provides the user with a few tips of how GANDI can be run.

6.1.1 Filtering

Although it happens that GANDI produces interesting results with the first setup used on a project, it can be useful to analyze the results obtained and then re-run GANDI with a more restrictive search to eliminate unwanted solutions. In general, GANDI could be first run without any filters (constraints, pharmacophore-based filtering) activated to see if solutions are found and then gradually introduce the latter to form a more focussed search. Using filters can help to narrow the search in chemical space for novel molecules. However, there are tradeoffs when doing so, the first being that only molecules will be produced that conform to the users expectations, i.e. alternate solutions are ignored that can sometimes provide valuable input for novel design strategies. Secondly, severe restrictions of the search with stringent filtering criteria can lead to few or even no molecules being produced. However, applying filters of course leads to a reduction in search space, which should allow GANDI to converge more easily to a given minima of the respective scoring functions. For these reasons it can make sense to first run GANDI with an unbiased setup, analyze the results and re-run GANDI with a more focused approach (or different setups exploring different design ideas).

6.1.2 Parameter settings of GANDI

GANDI has a number of input parameters that influence the search in chemical space and have been mentioned in the previous sections, whose influence on the outcome can be difficult to understand at first. Generally, a larger number of individuals and islands provides a more thorough search. Similarly a large number of iterations of the genetic algorithm and the tabu search leads to a prolonged optimization. Both settings of course lead to an increased computational cost.

6.1.3 Fragment definition

6.1.4 Why working with zones

The advantage of using zones, apart from the fact that molecules fill the desired pockets, lies in the often drastic reduction of the search space. For every defined zone (`MANDATORYZONE [# zones] [ZONE IDs]` in the input file) GANDI only samples the defined zones for a particular gene. For example in the case of proteases, which often show clearly defined pockets responsible for the substrate specificity, one might want to design an inhibitor that binds to a predfined set of these pockets. The example of a trypsin inhibitor is shown in Figure 9, where the protein was co-crystallized with an inhibitor with three distinct substructures. If the goal is to design similar molecules, GANDI can be run connecting three fragments (ligand size is three) of which each binds to a distinct pocket. The docked fragment poses should then be split into three sets (according to the zones) and labelled in the input file (see the example in 7). If GANDI is setup as described, the

first gene will encode the first zone (1), the second gene the second zone and so forth. Furthermore, when starting the optimizer (random initialization of individuals) or during the run, the permissable ranges of the respective genes are only the enumerated docked fragments. Assuming there are equal number of docked fragments n in all three pockets the theoretical number of docked fragment pose combinations is n^3 when using the zone definition. However, if GANDI uses a blind search without zone definition this increases to $(3n)^3$, i.e. a factor of 9. This increase can even be larger for e.g. a growing exercise where one fragment is fixed in zone 1 and the number of poses in zone 1 is a lot smaller that the total number $n_{Z1} << n_{tot}$ (and ligand size 2). The fraction of the number of combinations without and with zones is $\frac{n_{tot}^2}{n_{Z1} * n_{tot}}$ n_{tot}, e.g. with 10^5 fragment poses the reduction in search space is almost 10^5.

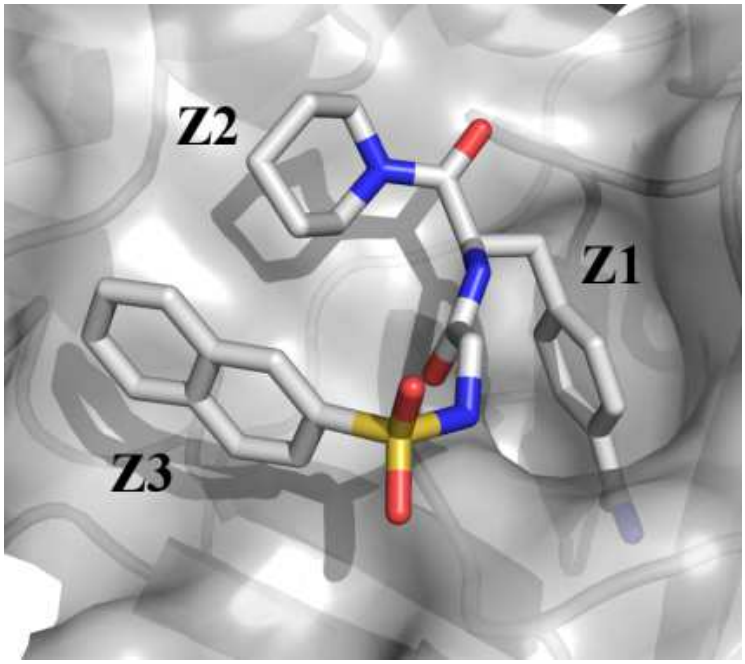


Figure 9: Example of how to derive zones (PDB code: 1PPC)

6.2 FAQ

- `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 3.9).

- `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 (3.6): 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 (4.6.5.1).
  - Increasing the connection vector angle and distance tolerances (4.6.5.3).
  - Increase the number of individual trials and tabu search iterations (4.6.2.2).

- `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.

6.3 Error messages

Following is a list of the more commonly occuring 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.

- `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 4.6.6.1).

- `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.

- `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.

- `WARNING, The file switch for ... is neither 'w' nor 'r'.`
  Reason : Incorrect switch setting.

- `WARNING, caught exception while reading ...`
  Reason : Encountered premature end of file or file read error.

- `WARNING, Number of islands <= 0, exiting`
  Reason : At least one island has to be defined.

- `WARNING, Number of individuals <= 0, exiting`
  Reason : The genetic algorithm needs multiple individuals to work efficiently.

- `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.

- `WARNING, Number of iterations <= 0`
  Reason : The number of iterations must be larger than zero.

- `WARNING, number of mandatory docked fragments is larger than ligand size , exiting !`
  Reason: GANDI enforces that generated molecules contain a binding pose of every docked fragment, thus the number of mandatory docked fragments cannot exceed the ligand size.

- `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.

- `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.

- `WARNING, ... grid margin should be > 0 ! : ... , exiting`
  Reason : The safety margin around the vdW or coulombic grid was not defined properly.

- `WARNING, ... grid spacing should be > 0 ! : ... , exiting`
  Reason : The grid spacing of the vdW or coulombic grid was not defined properly.

- `WARNING, Number of docked fragments/linkers should be > 0 ! : ... , exiting`
  Reason : No docked fragments or linkers were defined.

- `WARNING, There is more than one substructure in ...`
  Reason : One of the specified linkers contains more than one molecule.

- `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.

- `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 Nicolas Majeux, Peter Kolb, Philip Schütz, Pietro Alfarano, Claus Ehrhardt, Armin Widmer and Tim Knehans for useful discussions and comments.

References

[1] F. Dey and A. Caflisch. Fragment-based de novo ligand design by multiobjective evolutionary optimization. *Journal of Chemical Information and Modeling*, 48(3):679–690, 2008.

[2] D. E. Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning*. Addison-Wesley, Reading MA, 1989.

[3] J. H. Holland. *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, MI, 1975.

[4] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.*, 13:533–549, 1986.

[5] Fred Glover. Tabu search– part i. *ORSA J. Comput.*, 1(3):190–206, 1989.

[6] Fred Glover. Tabu search– part ii. *ORSA J. Comput.*, 1(1):4–32, 1990.

[7] Mark A. Murcko and Guy W. Bemis. The properties of known drugs. 1. molecular frameworks. *J. Med. Chem.*, 39:2887–2893, 1996.

[8] Mark A. Murcko and Guy W. Bemis. Properties of known drugs. 2. side chains. *J. Med. Chem.*, 42:5095–5099, 1999.

[9] Peter Ertl. Cheminformatics analysis of organic substituents: identification of the most common substituents, calculation of substituent properties, and automatic identification of drug-like bioisosteric groups. *J. Chem. Inf. Comput. Sci.*, 43:374–380, 2003.

[10] M. Vieth, M.G. Siegel, R.E. Higgs, I.A. Watson, D.H. Robertson, K.A. Savin, G.L. Durst, and P.A. Hipskind. Characteristic physical properties and structural fragments of marketed oral drugs. *J. Med. Chem.*, 47(1):224–232, 2004.

[11] Charles Darwin. *On The Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. John Murray, London, 1859.

[12] Paul Bryant Grosso. *Computer simulations of genetic adaptation: parallel subcomponent interaction in a multilocus model*. PhD thesis, Computer and Communication Sciences Department, University of Michigan, 1985.

[13] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Cryst.*, A32:922–923, 1976.

[14] Kong T Nguyen, Lorenz C Blum, Ruud Van Deursen, and Jean-Louis Reymond. Classification of organic molecules by molecular quantum numbers. *Chemmedchem*, 4(11):1803–1805, 2009.

[15] Manfred Hendlich, Friedrich Rippmann, and Gerhard Barnickel. Ligsite: automatic and efficient detection of potential small molecule-binding sites in proteins. *Journal of Molecular Graphics and Modelling*, 15(6):359 – 363, 1997.

[16] Carlos M. Fonseca and Peter J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16, 1995.

[17] Carlos M. Fonseca and Peter J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In *Genetic Algorithms: Proceedings of the Fifth International Conference*, pages 416–423. Morgan Kaufmann, 1993.

[18] Pasquale Foggia. *The VFLib Graph Matching Library, version 2.0*. http://amalfi.dis.unina.it/graph/db/vflib-2.0/doc/vflib.html.

A Changelog

Version	Release Date	Changes
2.0	X.X.2014	• Minor bug fixes, expection handling and other features (output directory, variable definitions etc.)
pre - 2.0	X.X.2011	• Angle and dihedral terms exchanged for penalty function • Pareto scoring mode • Pharmacophore-based, MQN and burial scoring function • Constraints- and Pharmacophore-based filtering • Exchanged mandatory fragments with mandatory zone option • Reading of gzip'ed docked fragment files • Input file format changed • Inter- and intra-island convergence check
1.3.	X.X.2010	• Angle and dihedral terms in energy function and sections 5.2.3 and 5.2.4); • Exhaustive search in case of low number of found matching linkers • SMILES parser and subgraph matching to avoid forbidden substructures (section 5.2.6) • Change of vector tolerances and matching linker look up (section 4.6.5.3)

Version	Release Date	Changes
1.2	27.2.2009	• Switch for zero atom linker (section 4.6.6.3) • Weighting terms for internal and linker energies (section 4.6.3.3) • Change in memory allocation to reduce usage • Final clustering of all molecules (inter-island) after optimization (section 4.6.3.2)
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 –

B Input Parameters

Manual section	Input tag	Default value	Range / Comment
4.6.1	NUMTHREADS	1	integer > 0
4.6.2.1	NUMISLANDS	10	integer > 0
4.6.2.1	NUMINDIVIDUALS	100	integer > 0
4.6.2.2	NUMITERATIONS	1000	integer > 0
4.6.2.2	NUMINDITERATIONS	20	integer > 0
4.6.2.2	NUMLINKERITERATIONS	50 ????	integer > 0
4.6.2.3	LIGANDSIZE	2	integer > 1
3.2.1	SELECTIONMODE	RANKROULETTE	rankroulette, tournament
3.2.3	MUTATIONP	0.2	real $\in [0,1]$
3.2.3	MUTATIONSIGMA	0.1 ????	real > 0
3.2.2	CROSSOVERP	0.8	real $\in [0,1]$
3.8	EXCHANGEMODE	RANDOM	random, neighbor, all
3.8	EXCHANGESTEP	100	integer > 0
3.8	EXCHANGERATIO	0.05	real ≥ 0
4.6.3.1	SAVINGSTEP	10000	integer > 0
4.6.2.7	RANDSEED	12345	integer > 0
4.6.3.2	SIMCUT	0.8	real $\in [0,1]$
4.6.3.2	FSIMCUT	0.99	real $\in [0,1]$
4.6.3.2	SIMEXPFACTOR	0.9	real > 0
4.6.3.2	SIMSQDISTCUT	16	real ≥ 0
3.4.2	SCORINGMODE	PARETO	pareto, weightedsum
4.6.3.3	EFFWEIGHT	1	real
4.6.3.3	SIM2DWEIGHT	0	real
4.6.3.3	SIM3DWEIGHT	0	real
4.6.3.3	BURIALWEIGHT	0	real
4.6.3.3	PH4WEIGHT	0	real
4.6.3.3	MQNWEIGHT	0	real
3.4.1.5	PH4	–	terminate list with "END"
3.5	CONSTRAINT	–	terminate list with "END"
3.4.1.2 / 3.4.1.3	TEMPLATE	–	terminate list with "END"

Table 17: Allowed input arguments, default values and admissable ranges

	Input tag	Default value	Range / Comment
3.4.1.1	INTENERWEIGHT	0.5	real ≥ 0
3.4.1.1	LINKERENERWEIGHT	1.0	real ≥ 0
3.4.1.1	BONDVIOLATIONPENALTY	5.0	real ≥ 0
3.4.1.1	LIGEFF	NONE	none, heavyatom ,weight
4.6.3.5	PRINTLEVEL	1	{0,1,2,3,4}
4.6.3.6	PARAMETERFILE	gandi.prm	absolute or relative path
4.6.4.1	RECEPTOR	–	terminate list with "END"
4.6.4.2	VDWGRIDMARGIN	10	real ≥ 0
4.6.4.2	VDWGRIDSPACING	0.3	real > 0
4.6.4.2	VDWGRIDACCESS	r	r, w
4.6.4.2	VDWGRIDFILE	vdw_grid.dat	absolute or relative path
4.6.4.3	DIELCONST	4.0	real ≥ 0
4.6.4.3	DISTDEPDIEL	1	1 (==yes, 0 (==no)
4.6.4.3	CLBGRIDMARGIN	10.0	real ≥ 0
4.6.4.3	CLBGRIDSPACING	0.5	real > 0
4.6.4.3	CLBGRIDACCESS	r	r, w
4.6.4.3	CLBGRIDFILE	clb_grid.dat	absolute or relative path
4.6.5.1	CLASHSCALFACTOR	0.89	real ≥ 0
4.6.5.1	SEVCLASHSCALFACTOR	0.5	real ≥ 0
4.6.5.1	MAXNUMCLASH	10	integer ≥ 0
4.6.5.1	MAXNUMSEVCLASH	0	integer ≥ 0
4.6.5.3	TOLERANCESQDIST	1.5	real > 0
4.6.5.3	TOLERANCEANGLE	20	real > 0
4.6.5.3	TOLERANCETORSION	20	real > 0
4.6.5.3	VECTORLENGTH	1.6	real > 0
4.6.5.3	COLINEARITY	175	real > 0
4.6.6.1	DOCKEDFRAGMENTS	–	terminate list with "END"
4.6.6.2	MANDATORYZONES	0	(== no mandatory maps)
4.6.6.2	MAPVECEQUIVALENCE	n	n, y
4.6.6.2	MAPBUMPCHECK	n	n, y [cutoff]
4.6.6.3	LINKERS	–	terminate list with "END"
4.6.6.3	ZEROATOMLINKER	y	y, n
3.7	MAXCONV	0.05	real $\in [0,1]$
3.7	CONVSIMCUT	0.95	real $\in [0,1]$
3.7	INTMAXCONV	0.5	real $\in [0,1]$
3.7	INTCONVSTEP	10	integer > 0
4.6.3.8	VARIABLE	-	string
4.6.3.7	OUTPUTDIR	-	string

Table 18: Part2: Allowed input arguments, default values and admissable ranges