Docking algorithm

Ligand representation

Following the formalism of genetic algorithms ligand is represented by the chromosome each gene of which codes certain degree of freedom sampled during the docking run. Physically, ligand viewed as a set of bar-joint rigid molecular fragments, in which internal degrees of freedom are placed on torsionally flexible chemical bonds. This representation is coded in an ordered linear chromosome, in which three genes stand for translations and three – for rotations of a root moiety of a ligand, N-1 genes (where N – is the number of rigid fragments) code internal (torsional) degrees of freedom, and if ligand contains flexible rings – for each ring the gene is reserved, which stores its conformation. Translational genes store Cartesian coordinates of a root fragment, rotation of a root is presented by Euler angles, torsional genes store the torsion angles of corresponding rotatable bonds. Choice of a root fragment of a molecule is not arbitrary – it is determined during the first stage of genetic algorithm as a fragment, which interacts most preferably with the protein. Flexible rings of a ligand are treated specially – all possible conformations of each ring are built and enumerated; thus ring conformation is stored as an integer in a corresponding gene. Currently five- and six-member rings are treated flexibly. Ring conformations are generated using the ring-flapping algorithm described elsewhere[1].

Types of energy calculations

Three distinct scoring functions are used by Lead-Finder (ranking, dG- and VS-scoring functions) which include the same set of energy contributions but weighted with different coefficients. However, since calculation of all energy terms is quite time-consuming, a number of approximate schemes to calculate scoring function are applied during the docking run, each of which uses more or less detailed representation of particular terms, and thus is more or less precise and computationally demanding. The most detailed and precise type of energy calculations corresponds to exact scoring function representation and are calculated only once (for each ligand pose) at the end of docking run.

Less precise representation of scoring function is applied at the stage of post-docking optimization of generated ligand poses, and thus is abbreviated as pdo-type energy. This type of scoring function accounts for the following components of protein-ligand interaction: van der Waals and Coulomb energy, formation of H-bonds, coordination with metal, volume-based non-polar solvation, penalties for shielding protein H-bond donors and acceptors. Internal ligand energy – pair-wise atomic interactions, 1-4 interactions and torsional energy – is also accounted in pdo-type energy. Surface component of solvation, embedding of polar ligand atoms into hydrophobic environment, electrostatic component of ligand desolvation and internal energy of ligand in solution are ignored at this stage to speed up calculations. Pdo-type energy is calculated several thousands to several tenths of thousands times during docking run.

Next is the so called ‘fast’ energy, which is calculated routinely during the genetic algorithm run and during preliminary docking stage in case when ligand conformation does not return very high energies. Fast-type energy includes van der Waals and Coulomb protein-ligand interactions,

simplified H-bonding energy. Internal ligand energy is presented by pair atomic interactions truncated at 4 Å; 1-4 interactions and torsional energy are dismissed. Fast-type energy is calculated from several hundreds of thousands to several millions of times during docking run.

When ligand energy is quite high (caused by overlap with protein or ligand itself) it is treated with coarse-type energy, which is the fastest to calculate. Protein-ligand interactions are presented only with van der Waals energy, and internal ligand energy is presented only with interactions of overlapping atoms. Coarse-type energy is mostly applied for fast pose optimization and can be called up to tenths millions times during docking.

Types of pose optimizations

Local pose optimization is viewed as a valuable component[2] of genetic algorithm, which aids faster evolution of individuals (ligand poses) in addition to common genetic operations such as recombination and mutation. By analogy with energy calculations a number of pose optimization algorithms are applied during docking run to balance between speed and accuracy. The most frequently used type of local optimization procedures during our docking is the so called pseudo Solis-Wets (PSW) optimization. It is based on random displacement in each degree of freedom (initial step is chosen specially for each gene) and following chosen direction in case energy of a new ligand pose is better. When after a series of trials no improvement in energy is detected the step is reduced, and trials are repeated until step size reaches the specified limit. This way of PSW optimization is referred by us as complete; it normally takes 100-300 iterations to reach local minimum. Fast PSW optimization implies exit from optimization cycle after a specified number of inefficient trials. During docking run complete and fast PSW optimizations are applied with relative probabilities 0.05 and 0.95; however, at the stage of initial pool generation (preliminary docking stage) use of complete PSW dominates. Overall, about 80% of all energy calculations during docking are called by PSW optimizations.

Special algorithm is applied to find the closest local minimum for a given pose. This algorithm marks degrees of freedom (DOF) displacement along which gains in energy. Further PSW-like optimization proceeds in a subspace spanned along these degrees of freedom. After a specified number of failed trials the subspace reduces to those DOF, which gave improvement on the previous productive step. Optimization and reduction of subspace are continued until one DOF is left; scanning the last DOF returns local energy minimum.

The most complete local optimization algorithm is designed to scan local minima surrounding the given pose and choose the optimal. This is achieved by scanning subspaces spanned over all single and double terminal DOF (two rotatable bonds counting from terminal fragments). Smoothed energy profiles are built for each subspace and for each mapped minima the local optimization algorithm (described in the previous paragraph) is run. Depending on the discreteness of the built energy profiles more or less precise optimization is achieved. The most precise settings are used for post-docking pose optimization. Less robust optimization is used for saving individuals in elite niches (see below) during docking, or in initial pool generation.

Initial pool generation

Input pool of individuals for docking algorithm is not arbitrary – it is preliminary optimized to speed up *in silico* evolution. Before docking algorithm is run, ligand conformation in solution is optimized – its energy will be used in energy calculations and its conformation will serve as a source for generating pool of ligand structures for docking. Initial pool of ligand structures is generated by randomizing translation and orientation coordinates of a molecule. Root of a ligand is determined as a fragment for which fast optimization of randomly generated individuals yields better results. For relatively big ligands (containing more than 7 rotatable bonds) root is not determined, reduced ligand (obtained by truncating terminal DOFs) is used instead.

Pool of 10000-100000 of random individuals is generated afterwards, each of which is subjected to a set of optimization procedures: first, all structures are minimized with complete PSW algorithm; then structures demonstrating promising binding of some fragments are optimized with the most complete algorithm. Ligand poses are energy sorted and clustered by geometry. Input pool for genetic algorithm is filled first by best individuals from found clusters and then by other structures according to their energy. Reduced ligands are reconstructed by building up and optimizing terminal degrees of freedom (which were preliminary stripped off).

When reference ligand structure is provided (to specify active site location or to verify docking accuracy in benchmarking study) it is optimized at this initial stage and its ΔG-score (reference energy) is calculated.

Genetic algorithm

Overall implementation of genetic algorithm looks as follows:

1. initial pool (population) of individuals is generated
2. while convergence of population is not achieved
    a. common operators (crossover, mutations, optimizations) are applied
    b. specific operators (crossover with individuals from elite niches, etc) are applied
    c. individuals are divided into niches
    d. individuals are selected for further rounds of evolution
3. docked poses are optimized, ranked and ΔG-score (and VS-score) is calculated.

Initial pool generation and optimization types were described above. Crossover is two-point (spans between two chosen genes). Mutations are generated with Cauchy distribution. Probability of mutation depends on the stage of the docking run – at the beginning it is relatively low (0.05), and grows up to 0.5 along with maturation of the population. Number of offsprings for a given individual is exponential to its rank. Additionally, three worst individuals are subjected to randomization followed by fast PSW optimization. Best individual in the niche is subjected to complete PSW optimization when it is changed.

Current implementation of the genetic algorithm uses such notion as a niche to cluster individuals with close genotype and to restrict their expansion. Niche is represented by individuals whose genetic distance (defined as a weighted difference in gene values) from the best individual is less than specified value. Niche size (maximal number of individuals in the niche) is restricted. Thus, when new individuals are generated, selection is preceded by clustering current populations into niches. Selection is performed by sorting offsprings according to their score and filling up niches. If the best individual of the best niche remains unchanged for certain number of selection rounds, it is transferred to the list of elite niches (other individuals from the niche are automatically erased). Elite individuals do not participate in docking directly, however they can form descendants with individuals from current population. All individuals within specified genetic distance from either elite individual are automatically removed from the population, however, when this individual has better energy than the elite one, the latter is replaced by the former. Number of elite niches is restricted to 20.

After convergence of population (specified number of generations is performed) ligand poses are optimized using more precise form of the scoring function (as described above) and the most complete optimization algorithm. Obtained poses are ranked and ΔG of binding is calculated using exact-type scoring function.

---

[1] A.W.R. Payne and R.C. Glen Molecular recognition using a binary genetic search algorithm J. Mol. Graphics, 1993, Vol. 11, 74-91.