Genetic Algorithm Tool

10 May 2016

DTC Laboratory
Dept. of Pharm. Tech.,
Jadavpur University,
Kolkata, West Bengal,
INDIA

Genetic Algorithm (GA) Tool for QSAR model development

A genetic algorithm (GA) is a search heuristic method that mimics the process of natural selection. Where the exhaustive search is impractical, heuristic methods are used to speed up the process of finding a satisfactory solution. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, crossover, mutation, and selection.

The evolution usually starts from a population of randomly generated individuals, and is an iterative process, with the population in each iteration is known as a generation. In each generation, the fitness of every individual in the population is evaluated; the fitness is usually the value of the objective function in the optimization problem being solved. The more fit individuals are stochastically selected from the current population, and each individual's genome is modified (recombined and possibly randomly mutated) to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population [1] [2].

Theoretical background and the Algorithm

Initialization of genetic algorithm

Initially many individual solutions are usually randomly generated to form an initial population, allowing the entire range of possible solutions. Occasionally, the solutions may be "seeded" in areas where optimal solutions are likely to be found.

Selection

During each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a fitness function, which evaluates each individual and based on this fitness function the best individuals are selected.

In Genetic algorithm v4.0 tool, the fitness function is based on the prediction error-based metrics, i.e., mean absolute error (*MAE*) and that were recently proposed in the literature [3]. Here, ” stands for the standard deviation of the absolute error (*AE*) values. The fitness function [4] (*equation* 1), is defined as follows:

$$F = \sum_{i=0}^{i=k} \frac{\hat{P}_{i,\mathrm{Threshold}} - P_i}{\hat{P}_{i,\mathrm{Threshold}} - \hat{P}_{i,\mathrm{Ideal}}} \quad \ldots (1)$$

where,

F = Fitness Score,

k = Number of parameters (here, k=2),

P_i = Value of respective parameter for a equation,

$\hat{P}_{i,Ideal}$ = Ideal value of that parameter,

$\hat{P}_{i,Threshold}$ = Threshold value of that parameter.

The metrics involved in this fitness function along with their ideal and threshold value are shown in Table 1. Here, higher fitness score signifies better fitness of the QSAR model in terms of actual prediction quality.

Table 1. The parameters along with their ideal and threshold value as employed in the fitness function of the *in-house* genetic algorithm tool.

Parameter	Ideal value	Threshold value
MAE	0.0	0.15× *TrainYObsRange**
MAE+3*SD	0.0	0.25× *TrainYObsRange*

*TrainYObsRange = Range of training set observed response values

Genetic operators

The next step is to generate a second-generation population of solutions from those selected through a combination of genetic operators: *crossover* (also called as recombination), and *mutation*.

For each new solution to be produced, a pair of "parent" solutions is selected for breeding from the pool selected previously. By producing a "child" solution using the above methods of crossover and mutation, a new solution is created which typically shares many of the characteristics of its "parents".Generally the average fitness will have increased by this procedure for the population, since only the best solutions from the first generation are selected for breeding, along with a small proportion of less fit solutions. These less fit solutions ensure genetic diversity within the genetic pool of the parents and therefore ensure the genetic diversity of the subsequent generation of children. Although crossover and mutation are known as the main genetic operators, it is possible to use other operators such as regrouping, colonization-extinction, or migration in genetic algorithms. It is worth tuning the parameters such as the *mutation probability, crossover* *probability* and *population size* to find reasonable settings for the problem class being worked on.

Termination

This process is repeated until a termination condition has been reached. Common terminating conditions are:

1. A solution is found that satisfies minimum criteria.

2. Fixed number of generations (usually user defined) is reached.

Genetic algorithm (GA) Tools

Genetic algorithm v4.1 (*with Process validation*) perform the genetic algorithm for selection of significant variables (*descriptors*). This tool develops model from information provided in training set and then determines corresponding training set prediction quality via different validation parameters. The fitness function (*fitness score; equation 1*) used to select the best solutions (*i.e. descriptor combination*) is well explained before. Here, the *fitness score* is directly proportional to the quality of QSAR model. And one can select the best model by selecting the QSAR model with highest fitness score. Some validation parameters such as *R^2, R^2 Adjusted, SEE, Q^2 (LOO), SDEP, Rm^2 metrics and prediction quality based on MAE-based criteria* [3, 5] are also calculated and are displayed in the output file. One can judge the robustness of the QSAR model by analyzing these validation parameters.

Genetic algorithm (GA) Tool Folder

The program folder consists of three folders "Data", "Lib" and "Output". For convenience, user may keep input file in "Data" folder and may save output files in "Output" folder, since by default, clicking on the browse button will open these folders. "Lib" folder consists of library files required for running the program. Hence try not to move or delete or rename these library files.

"Lib" folder also consist of a descriptor database file ("*DescriptorDatabase.xlsx*"; snapshot 1A) with basic information about descriptors calculated using cerius2, dragon and PaDEL software. This information is used to display brief description about each descriptors selected after running GeneticAlgorithm in the output file. The database file can be updated by the user, either by inserting information about a descriptor in any of the current sheets or add another sheet for descriptors calculated using different software, if required. Since the program identifies the descriptor from its Descriptor Symbol/Name (*first column*), it's important that it should be accurately typed or copy-pasted without any extra space and it is also case sensitive. Further take care that no cell should be left blank. If any information is not available, type "*Information Not Available*".

Input file format (*i.e. training set file format)*

Three different file types are allowed *i.e.* xlsx , xls and csv as input file. The input file (see snapshot 2) should consist of compound numbers (*first column*), descriptor values and the endpoint values (*last column*) for each object/compound. The format in which this information should be placed in the file is as follows:

First Row: Header *i.e.* name for each column, for instances, descriptor names, endpoint name. It can be numerical, alphabet or alphanumerical in nature.

First column: Serial number/Compound number (*only numerical values*)

Subsequent columns: Property/Independent variables/Descriptor values; each column will consist of each descriptor values for all the nanoparticles. These values should be numerical values and not alphabets or alphanumerical values.

Last column: Endpoint values/Dependent variables (*only numerical values*)

How to run the program

It's simple! Just click/double click on the jar file (*GeneticAlgorithm.jar*) present in the GA folder. A window will open as shown in *Snapshot 1*, with few queries (*given below*) that should be provided by the user before clicking on 'Submit' button to run the program.

"*Select Training set File*": Click on 'browse' button to select the *training set file*. By default, it will open the "Data" folder present in GA program folder. So for convenience, user can keep the input file in the "Dataset" folder.

"*Select Test set File*": Click on 'browse' button to select the *test set file*. By default, it will open the "Data" folder present in GA program folder. So for convenience, user can keep the input file in the "Dataset" folder.

"*Select Output Directory*": Click on 'browse' button to select the destination/output file directory and define output file name. By default, it will open the "Output" folder present in GA program folder. So for convenience, user can save the output files in the "Output" folder.

Optionally, user can perform *data pretreatment* of dataset to remove constant and inter-correlated descriptors prior to Genetic Algorithm execution. If user selects the checkbox labeled, as "*Data Pre-treatment*" then the following information has to be provided:

*Enter Variance cut-off: Enter the variance cut-off value based on which the constant variables will be removed. By default, the cut-off value is set to *0.001*.

*Enter correlation coefficient cut-off: Enter the inter-correlation coefficient cut-off value based on which the inter-correlated variables will be removed. By default, the cut-off value is set to *1.0*.

"*Total number of Iterations*": User should mention the number of iteration (generation) the algorithm will run. This is one of the stopping criteria (*default value: 100*). Other stopping criteria used in this program is that the algorithm will stop if there is no change in fitness function value (less than 0.001 differences in fitness value) for successive 10 generations.

"Equation Length": This is equivalent to number of descriptor which will be present in the generated models and also in the final best QSAR models (*default value: 3*).

At present, the algorithm used in this tool does not include addition and deletion as operators. Hence the equation length remains same throughout the algorithm.

"Cross-Over Probability": This value corresponds to the probability value (*range 0 to 1*) of performing cross-over operation within chromosomes (in this case various generated/selected equation) in each generation. The default value is 1. In this version, user cannot change the default value.

"Mutation Probability": This value corresponds to the probability value (*range 0 to 1*) of performing mutation operation within chromosomes in each generation. The default value is 0.3, due to its low probability of occurring in nature.

"Initial number of equations generated": This value corresponds to the number of equation generated in the first step (*first generation*). (*Fixed value*: 100)

"Number of equations selected in each generation": This value corresponds to the number of best equation selected in each generation.

Optionally, user can also perform process validation by selecting the checkbox labeled with "Process Validation" and then mention the number of random models to be generated (*the default value is 10*).

1. *Output text file (_GA.txt) :* The generated text file will consist of the resultant MLR equation, fitness function information (*i.e.* fitness score and LOF value), and validation parameters like r2, r2 adjusted, SEE, q2, and SDEP after the successful execution of the GA. It will also consist of brief description about each selected descriptors (*if the selected descriptor information is present in the database*)

2. *_Filecode*.csv*:* It consists of the compound no./serial no.(*first column*), selected descriptors (*subsequent columns*) and endpoint (*last column*) information. For each GA run, a new xlsx/xls/csv file with a different file name (*comprising of a file code*) is generated. This file will consist of information about compound number, descriptors and property/activity values of respective selected best equation. About 'file code' it is well explained in a note below.

3. *_Filecode_ProcessValid*.csv: The process validation results will consist of r, r^2, q^2 values for the original and all the generated random models; average r, r^2 and q^2 of random models; and cRp^2 value.

References:

1. http://en.wikipedia.org/wiki/Genetic_algorithm

2. https://engineering.purdue.edu/~lips/Publications/proddesgn/Encyc.pdf

3. Roy, Kunal*, Rudra Narayan Das, Pravin Ambure, and Rahul B. Aher. "Be aware of error measures. Further studies on validation of predictive QSAR models." *Chemometrics and Intelligent Laboratory Systems,* Volume 152, 15 March 2016, Pages 18–33. doi:10.1016/j.chemolab.2016.01.008

4. Ambure, Pravin, and Kunal Roy. "Understanding the structural requirements of cyclic sulfone hydroxyethylamines as hBACE1 inhibitors against Aβ plaques in Alzheimer's disease: a predictive QSAR approach." *RSC Advances* 6, 34 (2016): 28171-28186. DOI: 10.1039/C6RA04104C

5. Roy, K.; Mitra, I., On various metrics used for validation of predictive QSAR models with applications in virtual screening and focused library design. *Combinatorial Chemistry & High Throughput Screening* 14, (6), 450-474.

Java External Library Used

Apache POI – the Java API for Microsoft Documents

- Available at http://poi.apache.org/

XMLBeans

- Available at http://xmlbeans.apache.org/

Contact us at the following addresses:

Dr. Kunal Roy,
Drug Theoretics and Cheminformatics Lab.,
Dept. of Pharmaceutical Technology,
Jadavpur University,
Kolkata, West Bengal,
INDIA-700032
Email Id: kunalroy_in@yahoo.com

Software Developer details:

Pravin Ambure,
Research Scholar,
Drug Theoretics and Cheminformatics Lab.,
Dept. of Pharmaceutical Technology,
Jadavpur University,
Kolkata, West Bengal,

INDIA-700032

E-mail Id: ambure.pharmait@gmail.com