Programming for Chemical and Life Science Informatics – I573

Final Project Report

Topic: Genetic Algorithm for Feature Selection

Sashikiran Challa (schalla)

Introduction and Background:

Genetic Algorithm (GA), Simulated annealing, are stochastic methods available for the optimization of various parameters and selection of variables. Genetic Algorithm's have been widely used in the field of QSAR modeling. More specifically, one application of GA is to search a descriptor space to find optimal subsets of descriptors that can be used to build predictive models. In this project an attempt was made to examine how fast GA reaches the global best model.

GAs belong to the class of evolution based search algorithms. GAs were developed by John Holland in an attempt to explain the adaptive processes of natural systems and to design artificial systems based on these natural systems. [2] Given a problem and a population of individuals, a GA will evaluate each individual as a potential solution according to predefined evaluation or objective function. The evaluation function assigns a value to each individual and then this value is used by the fitness function to determine which individuals will breed to produce the next generation. Breeding comprises of crossover and mutation. This process of evaluation of individuals, selection of individuals for breeding, generation of new population is repeated until the value of the objective function converges.

A QSAR model is a mathematical relationship between a set of physicochemical descriptors (structural, geometric, etc) and a property (biological activity, solubility, boiling point, etc). A model can be built using linear methods or non-linear methods. In general linear method is used if the property to be predicted is a structural property like solubility, boiling point. Non- linear methods are used if the property to be predicted is a biological activity [1].

If the observation's response (i.e. observed variable) is dependent on just a single independent variable simple linear regression is used. But if it is dependent on several independent variables, multiple linear regression method is used. A multiple linear relationship can be modeled by

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots\ldots\ldots + \beta_p x_{ip} + \epsilon_i \quad i=1,2,\ldots.n$$

y_i is the response for the i^{th} observation and $x_{i1,}x_{i2,}\ldots\ldots x_{ip}$ are the independent variables for the i^{th} observation and n is the number of observations**.** β_0, β_1,…., β_p are the coefficients to be estimated. ϵ_i is the error term. Precisely y_i, β_0, $\beta 1$, …..β_p are estimated by the multiple linear regression technique. Once the parameters are estimated the quality of the model is evaluated by either R^2 value or root mean square error (RMSE). R^2 is obtained by dividing the difference between sum of squares total and sum of squares error by sum of squares total. It is the proportional improvement in prediction from the regression model, compared to the mean model. It ranges from 0 to 1. RMSE is the square root of the variance of the residuals. It indicates how close the observed values are close to the model's predicted values. Good models are characterized by high value of R^2 and low values of RMSE.

$$R^2 = \frac{Sumofsquarestotal - sumofsquareserror}{sumofsquarestotal} = \frac{\Sigma(y-\bar{y})2 - \Sigma(y-\hat{y})2}{\Sigma(y-\bar{y})2}$$

$$RMSE = \sqrt{\frac{\sum(y-\hat{y})2}{n}}$$

Data and Method:

The data provided was for 79 compounds with values of 43 different descriptors, and also values of their boiling points. Correlation and variance tests which are usually performed to filter out correlated descriptors were not performed since the number of descriptors provided was small.

The whole project was implemented in R, a language and environment for statistical computing and graphics. As the first step of this project Brute-force evaluation was implemented. The data files were read into R and using the *combinations* function available in gtools package all the different combinations of 3, 4, 5 subsets descriptor sets were obtained. A linear model was built on all the combinations obtained using the *lm* function in R. Simultaneously objective function's values (RMSE) values were evaluated for every combination.

The second step of this project was to implement GA. R code was written for the same. When using GA for descriptor selection a chromosome or an individual is simply a subset of descriptors (of user specified length). A population is defined as a collection of chromosomes or individuals. Initial population of descriptor subsets was generated by randomly selecting from the pool of descriptors. The number of descriptor subsets was chosen as 100 and the size of the subsets were 3, 4 and 5. A multiple linear regression

model was built on every subset in the population and their RMSE values were also calculated. Then using a fitness function each individual's fitness is evaluated. Two different fitness functions (1/RMSE[i]) and (2-(RMSE[i]/RMSE_avg)) were used. All the individuals with fitness value higher than the average fitness value are taken into mating population also called parent population. This process is called Elitism. This ensures that all the best individuals in a generation are taken into mating population. Rest of the mating population was filled using Roulette-Wheel selection method. On the mating population single-point crossover and mutations were implemented with 0.9 and 0.1 probabilities respectively. The resulting child population's RMSE values were evaluated. It was made sure that the maximum fittest individual between the initial population and the child population was taken into the child population. The child population was then made into second generation population and whole process was repeated for 1000 generations. Convergence rate was set to 200, i.e. when the same minimum RMSE value occurs for 200 generations then the GA would come to an end.

Results:

By the brute force method the following RMSE values and model descriptors were obtained.

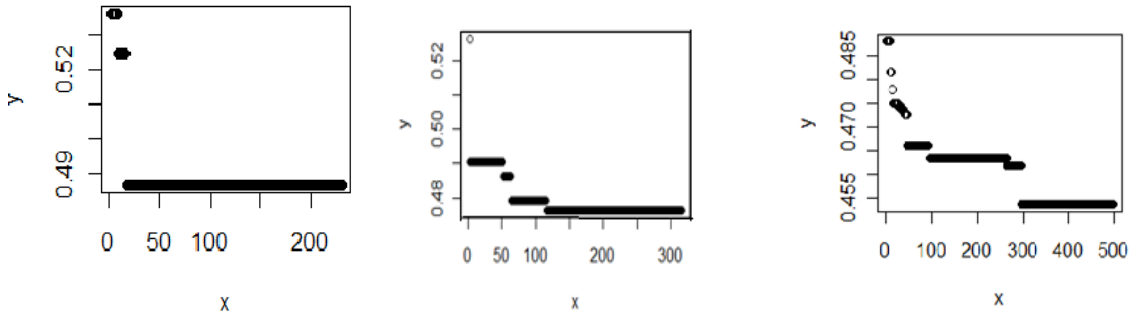
	3-subset	**4-subset**	**5-subset**
Brute-Force	**0.4865257**	**0.4762986**	**0.4463216**
Descriptors	**dMDEN.33, dFNHS.1, dRNH.1**	**dMOMI.6, dFNHS.1, dRNH.1, dS5CH.17**	**dMDEC.12, dMDEN.33, dFNHS.1,dRNH.1, d1SP3.1**
Time Taken	**57 secs**	**8mins 55secs**	**69 mins 58secs**

By using GA following RMSE values were obtained:

Generations reported were obtained by subtracting 200 from the generation at which GA ended.

GA	**3-subset**	**4-subset**	**5-subset**
1/RMSE[i] as fitness function	**0.4865247**	**0.476446**	**0.4463216**
Descriptors	**dMDEN.33, dFNHS.1,dRNH.1**	**dMOMI.6, dFNHS.1, dRNHS.1, dMDEN.33**	**dFNHS.1,dMDEC.12, d1SP3.1, dRNH.1, dMDEN.33**
No.of generations	**95, (4.51mins)**	**117, (4.95mins)**	**76, (5.6mins)**

(2-(RMSE[i]/RMSE_avg)) as fitness function **Descriptors** **No.of generations**	**0.4865257** **dRNH.1, dNDB.13, dELEC.0** **800 generations**	**0.484001** **dFNHS.1, dTHWS.1, dRNH.1, dMOMI.0** **800 generations**	**0.4712232** **dRNH.1, dMDEN.33, dCHDH.2, dMDEC.22, dFNHS.1** **800 generations**



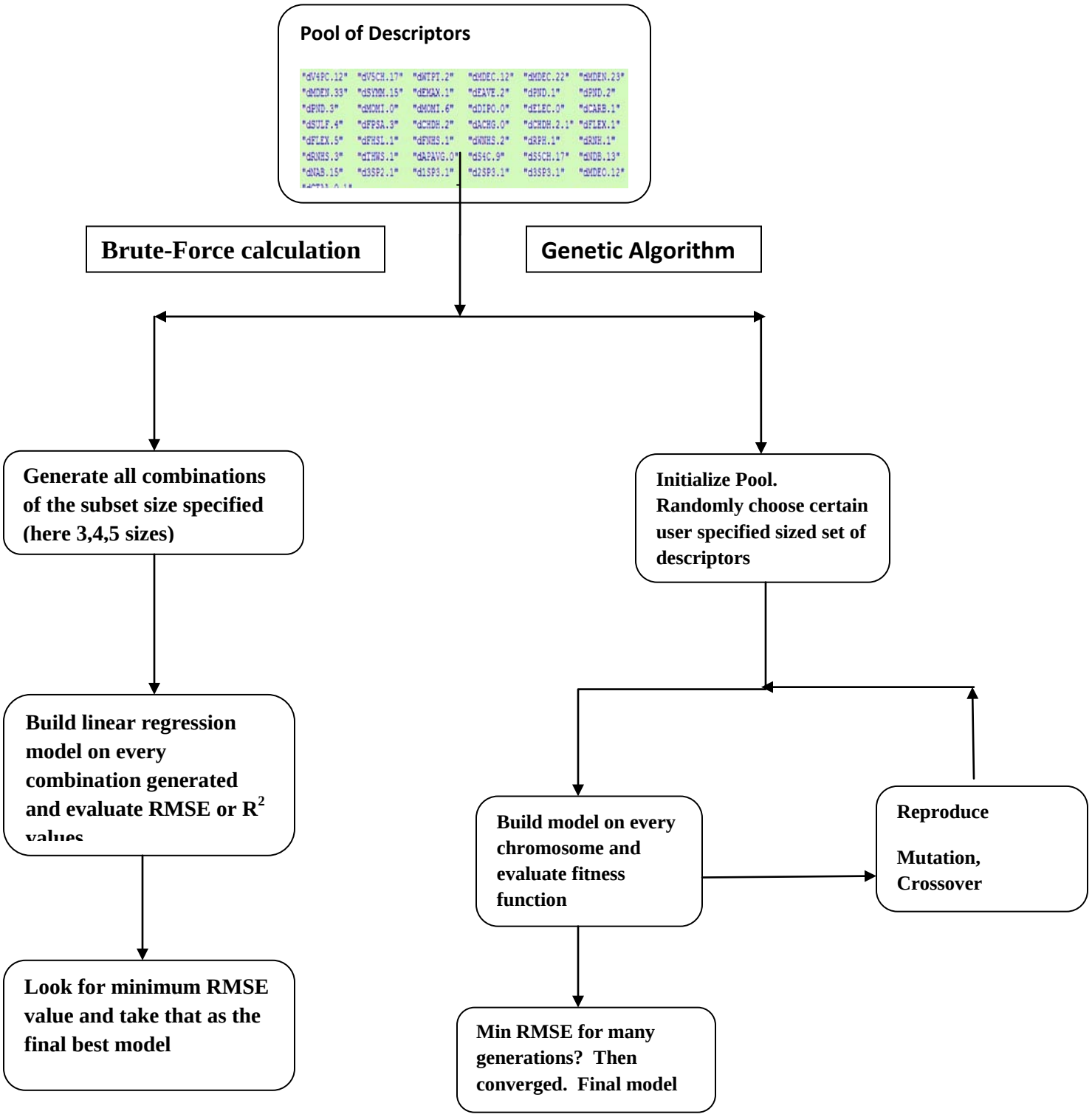
X-axis=generations, Y-axis=RMSE,

Plots of generations vs RMSE values. (3, 4, 5-subsets respectively)

From the results above it can be said that 5-subset descriptor is the best for it has the minimum RMSE value. And the fitness function (2-(RMSE[i]/RMSE_avg)) is not a good fitness function, for even after running for 1000 generations, it is not reaching the global optimal value. This is referred to as slow-finishing which is reasoned to be because of a single individual dominating all the generations. [3]

GAs may or may not reach the global optimal values, depending on the fitness function, parent selection method employed. Future work to this project could involve using different model building methods like SVM Regression, Neural Networks, and also using different parent selection techniques like Tournament Selection, Universal Stochastic Sampling, etc., and look at the best descriptor sets obtained.

[Acknowledgements: My Sincere thanks to Professor Guha for providing me with all the study material required to understand and implement this project. My thanks to Jae Hong for introducing me to QSAR modeling. Also my thanks to Kuldeep.J who helped me to understand the concept of Genetic Algorithm.]



Work-flow for the Project

References:

[1] Guha, R.; Statistical and Optimization Techniques, Chapter 2, PhD Thesis.

[2] Yasri, A.; Hartsough, D.; Toward an Optimal procedure for Variable Selection and QSAR Model Building. J. Chem. Inf. Comput. Sci., 2001, 41(5), 1218-1227

[3] Franco Busetti; Genetic Algorithms Overview. (A Tutorial on Genetic Algorithms)

[4] Viswesh, Vekatraman.; Andrew Roland, D.; Zheng, Rong.; Evaluation of Mutual Information and Genetic Programming for feature selection in QSAR. J. Chem. Inf. Comput. Sci., 2004, 44(5), 1686-1692

[5] David, Beasley.; David, Bull, R.; Ralph Martin, R.; An Overview of Genetic Algorithms: Part 1 Fundamentals. University Computing, 1993, 15(2), 58-69