Genomics 100 (2012) 65–71

Contents lists available at SciVerse ScienceDirect

Genomics

journal homepage: www.elsevier.com/locate/ygeno

Review

A survey of application: Genomics and genetic programming, a new frontier

Mohammad Wahab Khan *, Mansaf Alam

Department of Computer Science, Jamia Millia Islamia, Maulana Mohammad Ali Jauhar Marg, New Delhi 110025, India

ARTICLE INFO

Article history:
Received 24 December 2011
Accepted 29 May 2012
Available online 5 June 2012

Keywords:
Genomics
Genetic programming
Genetic network
Gene expression data
SNP

ABSTRACT

The aim of this paper is to provide an introduction to the rapidly developing field of genetic programming (GP). Particular emphasis is placed on the application of GP to genomics. First, the basic methodology of GP is introduced. This is followed by a review of applications in the areas of gene network inference, gene expression data analysis, SNP analysis, epistasis analysis and gene annotation. Finally this paper concluded by suggesting potential avenues of possible future research on genetic programming, opportunities to extend the technique, and areas for possible practical applications.

Contents

1. Introduction

1.1. Genetic programming

John Holland's pioneering *Adaptation in Natural and Artificial Systems* described how the evolutionary process in nature can be applied to solving problems using what is now called the *genetic algorithm* [1]. Genetic algorithm uses the principles of evolution such as reproduction, selection, crossover and mutation (collectively known as genetic operators) to discover better solution to a given problem that has random starting set of solutions.

In a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0 s and 1 s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations have been produced,

Abbreviations: GP, Genetic programming; SNP, Single nucleotide polymorphism; HS, Hierarchical clustering; GRN, Gene regulatory network; GPNN, Genetic programming neural network; GA, Genetic algorithm.

* Corresponding author. Fax: +91 11 26980014.
E-mail address: bioinventor@gmail.com (M.W. Khan).

0888-7543/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
doi:10.1016/j.ygeno.2012.05.014

or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.

Genetic programming is an extension of the genetic algorithm that automatically solves problems without requiring the user to know or specify the form or structure of the solution in advance. Genetic programming is an automatic domain-independent method for solving problems [2,3]. Starting with thousands of randomly created computer programs, it applies the Neo-Darwinian principle of natural selection, recombination (crossover), mutation, gene duplication, gene deletion, and certain mechanisms of developmental biology. It thus breeds better population over many generations. Genetic programming starts from a high-level statement of a problem's requirements and attempts to produce a computer program that solves the problem. Moreover it bears a strong resemblance to genetic algorithms (GA's). However despite the resemblance they also have some differences that can be summarized as follows:

- GA's generally make use of chromosomes as fixed length and structure in linear form whilst GP typically codes solutions as tree structured and variable length chromosomes.
- GP typically incorporates a domain specific syntax that governs meaningful arrangements of information on the chromosome. For GA's, the chromosomes are typically syntax free.
- GP makes use of genetic operators that preserve the syntax of its tree-structured chromosomes during 'reproduction'.
- GP solutions are often coded in a manner that allows the chromosomes to be executed directly using an appropriate interpreter. GA's are rarely coded in a directly executable form.

Genetic programming is a domain-independent method that genetically breeds populations of computer programs to solve problems by executing the following steps:

(1) Generates an initial population of random computer programs composed of the primitive functions and terminals of the problem.
(2) Iteratively performs generations consisting of the following sub-steps until the termination criterion of the problem is satisfied:
  (a) Executes each program in the population and determine how fit it is at solving the specific problem.
  (b) Creates a new generation of the population of programs by applying the following two primary operations to program(s) that are selected from the population with a probability based on fitness (i.e., the fitter the program, the more likely it is to be selected).
    (i) Reproduction: copies a selected program to the new population.
    (ii) Crossover: creates two new offspring programs for the new population by genetically recombining randomly chosen parts of two selected programs.
(3) The single best computer program produced anytime during the run is typically designated as the result of the run. This result may be a solution (or approximate solution) to the problem. A flowchart of a typical GP algorithm is shown in Fig. 1.

In applying genetic programming with automatic function definition to solving a problem, Koza J R [2] uses five major preparatory steps. These steps involve determining terminal set, function set, fitness function, control parameter, termination criteria.

[1]- Terminal set selection: a set of either variable atoms (representing, perhaps, the inputs, sensors, detectors, or state variables of some system) or constant atoms (such as the number 3 or the Boolean constant NIL).
[2]- Selection of function set: a set of domain specific functions used in conjunction with the terminal set to construct potential solutions to a given problem. For symbolic regression this could consist of a set of basic mathematical functions, arithmetic function, recursive and iterative function, Boolean and conditional operators.
[3]- Assignment of fitness function: the fitness measure is implied in terms of "what needs to be done" not "how to do it". Fitness is a numeric value assigned to each member of a population to provide a measure of the appropriateness of a solution to the problem in question. The fitness measure may incorporate any measurable, observable, or calculable behaviour or characteristic or combination of behaviour's or characteristics.
[4]- Control parameters: this includes the population size and the crossover and mutation probabilities.
[5]- The termination criterion: the termination criterion may include a maximum number of generations to be run as well as a problem-specific success predicate.

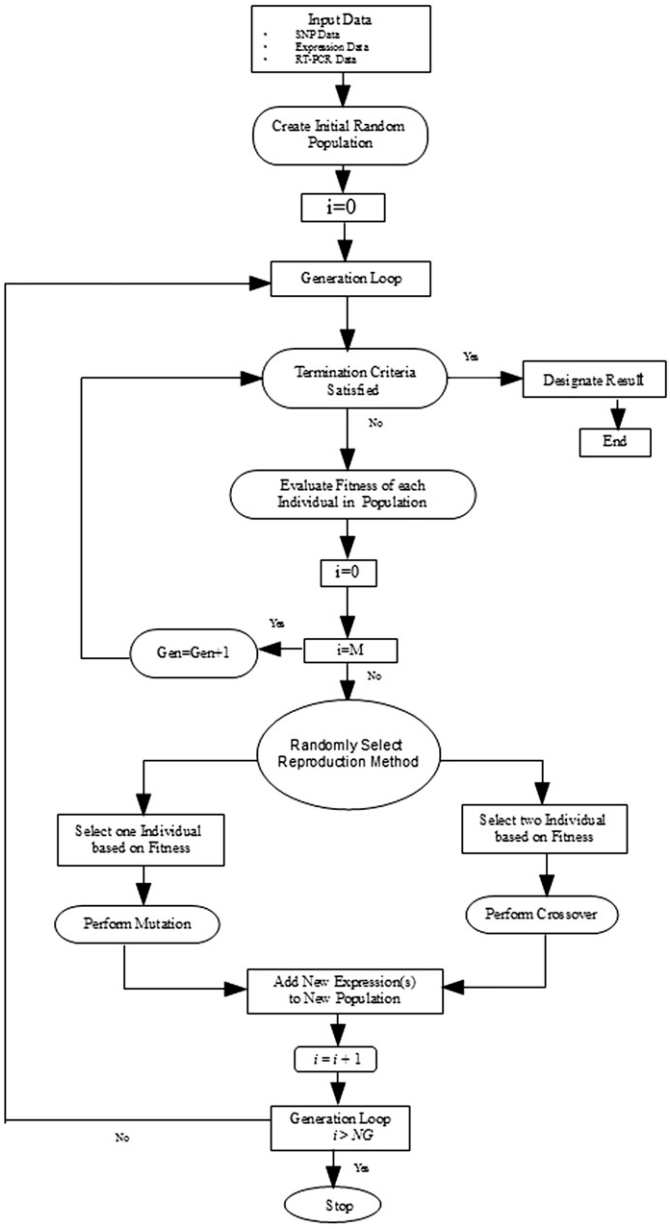

Fig. 1. Typical genetic programming algorithm flow sheet.

The first two preparatory steps define the primitive set for GP, and therefore indirectly define the search space GP will explore. The third, fitness step helps to execute the best elements in the search space

whereas the remaining two affect the quality and speed of search. The process of mechanically creating a computer program that fits certain numerical data is sometimes called system identification or symbolic regression.

In order to further understand the coding procedure and the genetic operators used for GP, consider the problem of predicting the numeric value of an output variable, Z, from two input variables x and y.

One possible symbolic representation for Z in terms of x and y would be,

$$y = (x + y) * 2 \ldots\ldots\ldots\ldots \quad (1)$$

Fig. 2 demonstrates how this expression may be represented as a tree structure.

With this tree representation, the genetic operators of crossover and mutation must be posed in a fashion that allows the syntax of resulting expressions to be preserved.

Fig. 3 shows a valid crossover operation where the two parent expressions are given by:

$$\text{Parent } 1 : Z = (x + y) * 2$$

$$\text{Parent } 2 : P = (a - b)/(c + d)$$

Parent 1 has input variables 'x' and 'y' and a constant '2' while parent 2 has four input variables 'a', 'b' 'c' and 'd'. Both expressions attempt to predict the process output, 'Z'. If the '*' from parent 1 and the '/' from parent 2 are chosen as the crossover points, then the two offspring are given by:

$$\text{Offspring } 1 : Z = (x + y) * (c + d)$$

$$\text{Offspring } 2 : Z = (a - b)/2$$

It is assumed that by recombining relevant sub-trees, it is possible to produce new expressions that provide fitter solutions. In order to provide population diversity and allow the exploration of areas of the solution space not represented in the initial population, a mutation operator may also be used. Mutation merely consists of randomly changing a function, input or constant in one of the mathematical expressions making up the present population.

Additional information on genetic programming can be found in books such as Banzhaf et al. [3]; books in the series on genetic programming from Kluwer Academic Publishers [4].

GP is inspired by Darwinian natural evolution, and has been applied successfully to a large number of difficult problems, such as automatic design, pattern recognition, robotic control, optimization, Biology, financial trading and forecasting. In the beginning of 90s Koza founded the GP which has grown exponentially since their introduction [2]. GP plays an important role in medicine, biology and bioinformatics. Its application in medicine, biology and bioinformatics has been studied by Handley S [5] and Koza J [6]. They used GP to make predictions about the behaviour and properties of biological systems and principally proteins. Furthermore, Howard Oakley, a practicing medical doctor, used GP to model blood owe in toes [7] as part of his long term interests in frost bite.

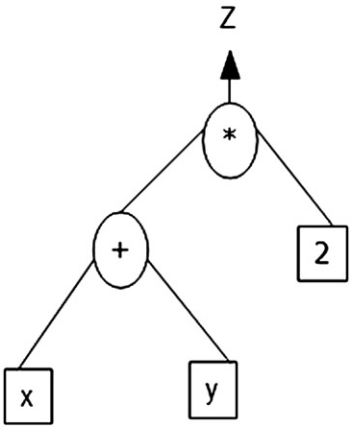

Fig. 2. Representation of a numeric expression using a tree structure.

2. Genetic programming applications in genomics

The following section presents a review of genomics applications of GP. The results of the literature survey have been arranged into the following broad categories:

(1): Genetic network inference
(2): Gene expression data classification
(3): SNP analysis
(4): Epistasis analysis
(5): Gene annotation

2.1. Genetic network inference

Network biology involves the use of networks to represent complexity, computes biological relationships, and seeks to uncover biological principles and insights. The approach of inferring gene regulatory networks (GRNs) has been flourishing for many years, and new methods from mathematics, information science, engineering and social sciences have been applied. The goal of developing models of biological systems is to balance simplicity with acceptability. If a model is too simplistic, extensions to biology may be feeble, and findings will lack influence. Conversely, overly elaborate models that include unnecessary details will obscure the essential dynamics of the process under investigation. Furthermore, to communicate to a biological audience, the computational tasks in a simulation study need to address identifiable aspects of biological phenomena. A variety of genetic regulatory network (GRN) models have been developed to simulate aspects of biological development, including the effect of interplay between multiple interacting mechanisms [8], the role played by physical interactions [9] and the dynamics of inter cellular communication [10]. GP has been implemented to create the transcriptional gene regulatory network of the lac operon [11]. Transcriptional interactions are not sufficient to illustrate the dynamics of central dogma of molecular biology. A new genetic programming (GP) approach has been used for evolving genetic networks that demonstrates desired dynamics when simulated as a discrete stochastic process [12]. Their representation of genetic networks is based on a biochemical reaction model including key elements such as transcription, translation and post-translational modifications. Despite these there are certain drawbacks as they are slow and feasible for small data-sets. For large number of gene interactions and fast parameter estimation, a unified approach to infer gene regulatory networks using the S-system model proposed [13]. That is based on the GP embedded multidimensional optimization algorithm. Due to the imprecise nature of biological experiments, biological data are often characterized by the presence of redundant and noisy data, which are usually derived from errors associated with data collection, such as contaminations in laboratorial samples. Gene expression data represent an example of noisy biological data that suffer from this problem [14]. To mitigate the effects of noise, GP embedded Kalman filtering method was used [15]. Simulations with synthetic and yeast data demonstrate the effectiveness of the proposed algorithm. In general, the statistics of the noise in the micro array measurements are not known. Thus, the Kalman filter may not be appropriate for estimating parameters. For robust estimation of parameters even without the knowledge of the noise statistics a joint GP and H∞ filtering approach is applied to infer the GRN [16], where H∞ filtering provides optimal parameter estimations under uncertainties. Filtering methods [15,16] were only used

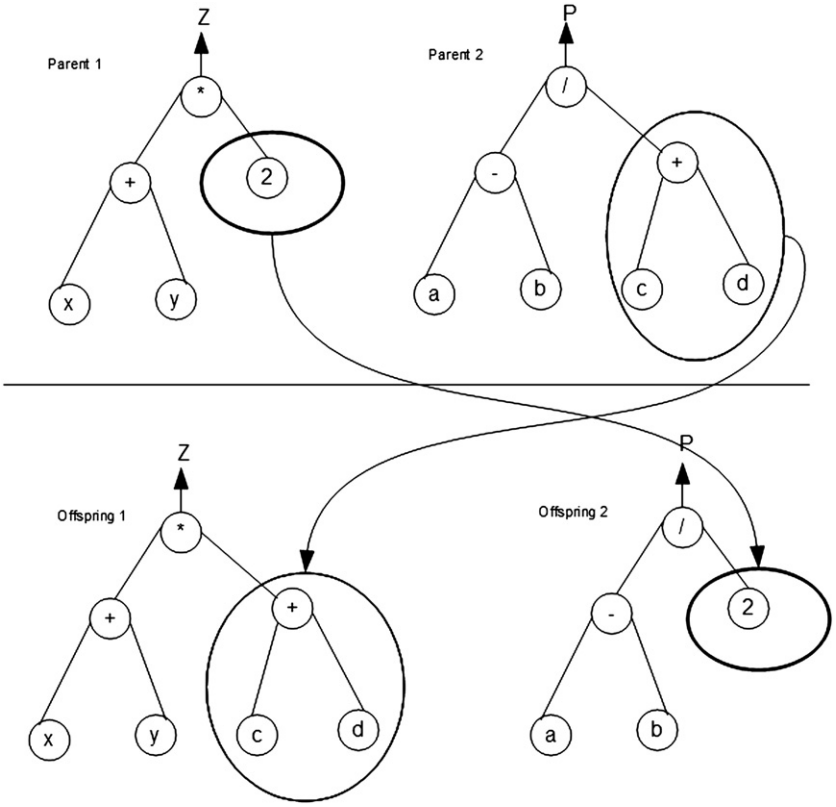

Fig. 3. A typical crossover operation.

to cover the noise in data while in many cases system noise was also working as stumbling block in fine tuning of network process. To mitigate this effect a noisy sigmoid model, to include both system noise and measurement noise, came up to existence [17].

The identification and characterization of genes that influence the risk of common, complex multifactorial disease primarily through interactions with other genes and environmental factors remains a statistical and computational challenge in genetic epidemiology. To deal with this issue, new statistical approaches GPNN have been developed [18]. Genetic programming neural network (GPNN) is a novel pattern recognition approach for detecting gene–gene and gene–environment interactions in studies of human disease. GPNN has high power to detect two and three-locus interactions in moderate sample sizes. Higher order interactions will require larger sample sizes.

Currently, the mainstream methods for modelling either pre require the regulatory relationship or are not able to demonstrate the dynamics of the regulatory network. New differential equations based GP derived method has been devised for adjustment of continuous external changes, search for the regulatory models suitable for the experiment data using genetic operators, and realize the prediction for the random regulatory relationship between genes [19]. In this method GP plays a crucial role to conform the vicinity dynamics, and depicts the regulatory models suitable for the experiment data.

Biology presents incomparable, but desirable characteristics compared to engineered systems. The emerging consensus coming from biological research points to the importance of dynamic networks of gene activity (GRN). Such networks are concerned with gene regulation and protein synthesis. From the system point of view, biological development could be viewed as a multi-layered system with each layer abstracted by different system processes. Bringing these concepts together, one can hypothesize that every multi-cellular organism could be viewed as a multi-layered system that develops from the zygote cell via the mechanisms of gene regulation and cell signalling. In considering biologically inspired artificial designs, one of the important questions concerns the amount of biological realism that should be included. For such type of biosystem creation GP has been used for the design of artificial genetic regulatory network (AGRN). Electronic circuitry of AGRN is based on cell signalling and regulatory mechanisms [20].

Modularity pervades all levels of biological process and is the building block in biological construction. In the engineering field, it is also an important aspect in designing large systems. GP helps to investigate the modularity in a developmental gene regulation network model for bio-inspired circuit constructions. To achieve scalable combinatorial systems for gene regulation in cellular system, modular design is done by genetic programming [21].

Gene gates model the basic regulatory mechanism which involves the production of proteins (translation) from DNA through the production of RNA (transcription). In such stochastic gene gates model, transcription and translation are considered a single action. A feature-based fitness function is applied in a genetic programming system to synthesize stochastic gene regulatory network models whose behaviour is defined by a time course of protein expression levels [22].

2.2. Gene expression data classification

Due to recent advances in DNA microarray technology, it became possible to obtain gene expression profiles of samples from different disease/diagnostic classes. Several classification algorithms (k-NN, SVM, CART, NSC, Neural Network, FLDA, and DLDA) based on statistical analysis have been applied on these profiles, in an attempt to achieve accurate and automatic class prediction. The classification methods are relatively well established, however, the complexity of the problems rooted in the microarray technology hinders the applicability of the classification methods as diagnostic and prognostic predictors or class-discovery tools in medicine. Furthermore, the

question of classification "which method is better" does not have a simple answer. To overcome this problem GP has played a good role to identify the best solution. In classification of gene expression data, genetic programming acts as a classifier as well as a gene selection algorithm. Conventional ensemble approaches construct various classifiers by estimating the similarity on the output patterns of them, and combine them with several fusion methods. Since they measure the similarity indirectly, it is restricted to evaluate the precise diversity among base classifiers. Ensemble is a representative technique for improving classification performance by combining a set of classifiers. It is required to maintain the diversity among base classifiers for effective ensemble. An effective genetic programming ensemble method comprehensively obtained the classification rules which directly estimate the similarity between them [23]. Genetic programming implicit majority voting technique was implemented for the prediction of the class of a test sample to obtain the accuracy on two microarray data-sets. The method was applied to cancer gene expression data. The accuracy obtained with majority voting is better than the average accuracy of the rules in a voting group [24].

Many newly discovered genes are of unknown function. DNA microarrays are a method for determining the expression levels of all genes in an organism for which a complete genome sequence is available. By comparing the expression changes under different conditions it should be possible to assign functions to these genes. However, many hundreds of thousands of data points may be produced over a series of experiments. Genetic programming provided simple explanatory rules for gene function from such data-sets, where previous approaches had not succeeded [25]. It was not only to derive classifier rules with extremely high classification accuracy, but the structures of the rules themselves have been shown to lead to the discovery of previously unsuspected biological insights into the functioning of an organism at the whole-genome level.

Genetic programming method exquisitely was used for the medical diagnosis evolving classifier [26] and identification of individualized feature selection in breast cancer [27] using gene expression data. To find out the hidden relationship between genetic material and tumour pathologies, genetic programming came across to draw the model from two oncology data-sets, first healthy and cancerous colon tissues second, acute myeloid and lymphoblastic leukaemia [28]. To study gene function and gene regulation information various clustering methods existed at present usually need manual operations, which are difficult for gene expression data. To handle the data at large scale, high-dimension, and noise, a novel genetic programming (GP) clustering system based on hierarchical statistical model (HS-model) was proposed for the whole intact yeast gene data without dimensionality reduction [29]. That unambiguously deals with characteristic of gene expression data. Feature selection and classification of multiclass microarray data-sets through sub assembling have been done by genetic programming using greedy algorithm-based methods [30].

2.3. SNP analysis

Genomic studies provide large volumes of data with the number of single nucleotide polymorphisms (SNPs) ranging into thousands. The analysis of SNPs permits determining relationships between genotypic and phenotypic information as well as the identification of SNPs related to a disease. Continually growing wealth of information and advances in biology, calls for the development of approaches for discovery of new knowledge. Genetic epidemiologists have taken the challenge to identify genetic polymorphisms involved in the development of diseases. Many have collected data on large numbers of genetic markers but are not familiar with available methods to assess their association with complex diseases. Statistical methods have been developed for analysing the relation between large numbers of genetic and environmental predictors for disease or disease-related variables in genetic association studies. One such area is the identification of gene or SNP patterns impacting cure or drug development for various diseases. For selections of significant genes/SNPs the Genetic algorithm-based gene selection (GAGS), Feature set intersection approach (CFS), Weighted decision-tree-based gene selection (WDTGS) approaches [31] performed far better than the information gain (IG) and standard regression (REG) approaches [32] in terms of all three-quality measures, i.e., cross-validation accuracy, specificity, and the number of significant genes/SNPs. The GA–CFS–WDTGS approaches were uniquely able to identify some gene/SNPs that could not be identified by the IG and REG approaches. To find SNPs from a hay stack of SNPs a combinatorial approach genetic programming optimized neural network (GPNN) [33] was employed. This algorithm uses the logistic regression analysis, neural networks, including the parameter decreasing method (PDM) and genetic programming optimized neural networks(GPNN) and several non-parametric methods, which include the set association approach, combinatorial partitioning method (CPM), restricted partitioning method (RPM), multifactor dimensionality reduction (MDR) method and the random forests approach to select and model important SNPs. The GPNN combinatorial method gives more insight in combination patterns for sets of genetic and/or environmental predictor variables that may be related to the outcome variable. Non-parametric methods are widely used for studying populations that take on a ranked order. They may be applied in situations where less is known about the application in question. Another justification for the use of non-parametric methods is simplicity and increased robustness. Non-parametric models differ from parametric models in that the model structure is not specified a priori but is instead determined from data. The term non-parametric is not meant to imply that such models completely lack parameters but that the number and nature of the parameters are flexible and not fixed in advance. More information about comparative analysis of parametric and non-parametric methods in SNP analysis could be found in A Geert Heidema et al. [33].

2.4. Epistasis analysis

In human genetics it is now possible to measure large numbers of DNA sequence variations across the human genome. Given current knowledge about biological networks and disease processes it seems likely that disease risk can best be modelled by interactions between biological components, which may be examined as interacting DNA sequence variations. The machine learning challenge is to effectively explore interactions in these data-sets to identify combinations of variations which are predictive of common human diseases. The identification of risk-associated genetic variants in common diseases remains a challenge to the biomedical research community. Common statistical approaches that exclusively measure main effects hindered the detection of interactions between some of these variants. Thus, detecting and interpreting interactions are a challenging task from the statistical and computational perspectives. However, certain methods in computing science may be the key player to improve our understanding on the mechanisms of genetic disease by detecting interactions even in the presence of very low heritability. Genetic programming has been implied to induce a decision tree to detect interactions in genetic variants which uses cross-validation strategy for estimating classification and prediction errors and tests for consistencies with experimental data [34]. To have better estimates, a new consistency measure that takes into account interactions and had been used in a genetic programming environment was proposed. GP approach has been used to examine the role that an expert knowledge aware initializer can outperform both a random initializer and an enumerative initializer [35]. One attractive feature of the probabilistic initializer is that it is easily integrated into already existing approaches. The study of common, complex multifactorial diseases in genetic epidemiology is complicated by nonlinearity in the genotype-to-phenotype mapping relationship that is due, in part,

to epistasis or gene–gene interactions. Moore et al. [36] have recently shown that Symbolic Discriminant Analysis (SDA), which uses a GP approach to generate models, was able to successfully model predictors of atrial fibrillation in a well characterized data-set which included a two-way epistasis interaction. This has been shown to be an effective strategy for modelling epistasis. Genetic "Mask" as a novel building block exploits expert knowledge in the form of a reconstructed relationship between two attributes [37]. The availability of "Mask" building blocks improves SDA performance and supports the idea that preprocessed data improves GP performance. Genetic programming was also used to determine the epistasis genetic risk factor in human disease. With the help of statistical expert knowledge computational evolution system is able to construct the disease susceptibility, gene–gene interactions and protein–protein interaction [38]. To understand the parameter estimation and system identification for epistatic interaction genetic programming has been used. An analytical tool ATHENA (Analysis Tool for Heritable and Environmental Network Associations) that incorporates grammatical evolution neural networks (GENN) has been used to detect interactions among genetic factors [39].

2.5. Gene annotation

Gene annotation is used to refer to the prediction and annotation of a coding transcript on a region of the genome, but as the complexity of the functional features on the genome increases, users require prediction of non-coding RNAs, alternatively spliced transcripts, pseudo genes, and conserved elements. Eight years after the initial draft sequence of the human genome was published, the exact number of coding genes present on this sequence is still unclear. Since new sequencing technologies have reduced the cost of sequencing and dramatically increased the speed, we can expect an enormous expansion in the amount of available genomic and transcript sequence data. To gain insight into the functional information contained within these new sequences, the features within the sequence need to be accurately annotated. As stated in the literature Brunel [40], the human genome project is officially completed. Out of all the genes sequenced, approximately 40% of these genes code for a protein that has an unknown function. To obtain the gene function without experimental equipment, a mathematical discriminate function using artificial intelligence algorithm GP has been used to find a discriminate function that predict the gene action into some biological function [41]. Promoters or splicing sites control gene expression and are important for successful gene prediction, and can be recognized by certain patterns or motifs that are conserved within a species. To solve the problem of promoter identification in eukaryotes the Boolean GP approach was applied [42] and at the same time provided accurate classification results and the same time generated solutions that were easily interpreted with known promoter characteristics. GP can be used to classify a given gene sequence as either constitutively or alternatively spliced. Wiehe et al. [43] has used a feature matrix of sequence properties such as nucleotide composition or exon length, was passed to the GP system "Discipulus". Their classifier yields highly accurate sensitivity and specificity predictions on the retained introns and cassette exon data. Genetic programming has been implied to create regular expressions as mRNA motifs to predict human exon splitting. RNAnet {http://bioinformatics.essex.ac.uk/users/wlangdon/rnanet/} allows the user to calculate correlations of gene expression, both between genes and between components within genes [44]. The universe of functionally important non-transcribed RNAs is rapidly expanding but their systematic identifications in genomes remain challenging. Non-coding RNAs (ncRNA) are transcripts, whose function lies in the RNA sequence itself and not as information carriers for protein synthesis. Although long believed to be a minor gene class, recent discoveries have revealed that ncRNA genes are far more prevalent than previously believed and that beyond these three classic tasks, there are also many other known now (rRNA and tRNA protein synthesis) [45,46].

Several methods exist for predicting non-coding RNA (ncRNA) genes in Escherichia coli [47,48]. A new machine learning algorithm called GPboostReg has been implemented to use automatic discovery of sequence patterns to predict ncRNA genes [49]. GPboostReg is applied to create classifiers that predict whether or not a sequence is an ncRNA gene. To create the classifiers, GPboostReg combines genetic programming and boosting algorithms. The effects of genes on phenotype are mediated by processes that are typically unknown but whose determination is desirable. The conversion from gene to phenotype is not a simple function of individual genes, but involves the complex interactions of many genes; it is what is known as a nonlinear mapping problem. This nonlinear mapping problem has been solved with the help of GP. Encoding of preferentially selected data of cellular and higher-order activities by genes is seen as directly analogous to computer programs. This analogy is of utility in biological genetics and in problems of genotype–phenotype mapping [50].

3. Conclusion

This survey has revealed that GP is used in genomics and currently focussed on typical genetic analysis and gene network inference. While computer scientists have concentrated on gaining a fundamental understanding of the algorithm and improving its performance, the biologist community is addressing practical issues, often by introducing accepted biological understanding and mechanism. Perhaps the most promising research direction appears to be the application of GP techniques to gene regulatory network inference problem. While computational considerations currently limit the accuracy and robustness of problems that can be addressed, these will inevitably be lifted as processor speeds continue to increase. Further potential avenues of research include the investigation of gene expression data and gene–gene interaction at cell organization level, and also investigation of other algorithms capable of performing model optimization. With the increasing availability of time series microarray data, the algorithm could be applied to construct models to characterize cancer evolution and serve as the basis for developing new regulatory therapies. Genetic programming which is well proficient evolutionary algorithm could be helpful to get the solution from the problems of emerging field of next generation sequencing. It is emphasized that GP is a young field of research, whose practitioners are still exploring its capabilities and limitations.

Acknowledgments

I would like to thank Mr. Danishuddin for their valuable assistant to me in drafting and reviewing the manuscript. I am thankful for his courageous motivation to me for exploring the research.

References

[1] John H. Holland, Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, University of Michigan Press, Ann Arbour, MI, 1975 The second edition is currently available from The MIT Press 1992.
[2] J.R. Koza, Genetic Programming: on the Programming of Computers by Means of Natural Selection, MIT Press, USA, 1992.
[3] W. Banzhaf, P. Nordin, R.E. Keller, F.D. Francone, Genetic Programming — An Introduction, Morgan Kaufmann and Heidelberg:dpunkt, San Francisco, CA, 1998.
[4] W.B. Langdon, Genetic Programming and Data Structures: Genetic Programming + Data Structures = Automatic Programming! Kluwer, Amsterdam, 1998.
[5] Simon Handley, Automatic learning of a detector for alpha-helices in protein sequences via genetic programming, in: Stephanie Forrest (Ed.), Proceedings of the 5th International Conference on Genetic Algorithms, ICGA-93, 1993, pp. 271–278.
[6] John R. Koza, David Andre, Classifying protein segments as trans-membrane domains using architecture-altering operations in genetic programming, in: Peter J. Angeline, K.E. Kinnear Jr. (Eds.), Advances in Genetic Programming 2, chapter 8, MIT Press, USA, 1996, pp. 155–176.

[7] Howard Oakley, Two scientific applications of genetic programming: stack alters and non-linear equation fitting to chaotic data, in: Kenneth E. Kinnear Jr. (Ed.), Advances in Genetic Programming, chapter 17, MIT Press, USA, 1994, pp. 369–389.
[8] P. Hogeweg, Shapes in the shadow: evolutionary dynamics of morphogenesis, Artif. Life (USA) (2000) 85–101.
[9] K. Fleischer, Investigations with a multicellular developmental model, in: C.G. Langton, T. Shimohara (Eds.), Artificial Life V, The MIT Press/Bradford Books, Cambridge, MA, 1996, pp. 229–236.
[10] C. Furusawa, K. Kaneko, Complex organisation in multicellularity as a necessity in evolution, Artif. Life (USA) 6 (2000) 265–281.
[11] John R. Koza, Guido Lanza, William Mydlowec, Automatic creation of a genetic network for the lac operon from observed data by means of genetic programming, First International Conference on Systems Biology (ICSB), November 14–16 2000.
[12] André Leier, P. Dwight Kuo, Wolfgang Banzhaf, Kevin Burrage, Evolving noisy oscillatory dynamics in genetic regulatory networks EuroGP, Lecture Notes in Computer Science, vol. 3905, Springer, 2006, pp. 290–299.
[13] Haixin Wang, Lijun Qian, Edward Dougherty, Inference of gene regulatory networks using S-system: a unified approach, Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, 2007.
[14] Giampaolo L. Libralon, São Carlos, André C. Ponce Leon Ferreira, Ana C. Lorena, Ensembles of pre-processing techniques for noise detection in gene expression data, Proceeding ICONIP'08 - Volume Part I, Springer-Verlag, Berlin, Heidelberg, 2009.
[15] Haixin Wang, Lijun Qian, Edward Dougherty, Inference of gene regulatory networks using genetic programming and kalman filter, IEEE International Workshop on Genomic Signal Processing and Statistics, GENSIPS '06, May 2006, pp. 27–28.
[16] Haixin Wang, Lijun Qian, E. Dougherty, Modelling genetic regulatory networks by sigmoidal functions: a joint genetic algorithm and Kalman filtering approach, Third International Conference on Natural Computation, ICNC 2007, vol. 2, August 24–27 2007, pp. 324–328.
[17] Lijun Qian, Haixin Wang, Edward R. Dougherty, Inference of noisy nonlinear differential equation models for gene regulatory networks using genetic programming and Kalman filtering, IEEE Trans. Signal Process. 56 (7) (July 2008) 3327–3339.
[18] Alison A. Motsinger, Stephen L. Lee, George Mellick, Marylyn D. Ritchie, GPNN: power studies and applications of a neural network method for detecting gene–gene interactions in studies of human disease, BMC Bioinformatics 7 (2006) 39, http://dx.doi.org/10.1186/1471-2105-7-39.
[19] Jing Liu, Aiguo Wu, Modelling gene regulatory network based on genetic programming, 2010 International Conference on Electrical and Control Engineering (ICECE), June 2010, pp. 5341–5344.
[20] Song Zhan, Julian F. Miller, Andy M. Tyrrell, An evolutionary system using development and artificial Genetic Regulatory Networks for electronic circuit design, Biosystems 98 (3) (2009) 176–192.
[21] Song Zhan, Julian F. Miller, Andy M. Tyrrell, Modular design from gene regulation in a cellular system, IEEE Congress on Evolutionary Computation (CEC 2010), IEEE Press, July 18–23 2010.
[22] Janine Imada, Evolutionary synthesis of stochastic gene network models using feature-based search spaces, January 28 2009.
[23] Jin-Hyuk Hong, Sung-Bae Cho, Ensemble Genetic Programming for Classifying Gene Expression Data, Proceedings of the Second Asian-Pacific Workshop on Genetic Programming, December 6–7 2004.
[24] Topon Kumar Paul, Yoshihiko Hasegawa, Hitoshi Iba, Classification of gene expression data by majority voting genetic programming classifier, 2006 IEEE Congress on Evolutionary Computation Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada July 16–21, 2006.
[25] Richard J. Gilbert, Jem J. Rowland, Douglas B. Kell, Genomic computing: explanatory modeling for functional genomics GECCO, Morgan Kaufmann, 2000, pp. 551–557.
[26] Stephan M. Winkler, Michael Affenzeller, Stefan Wagner, Using enhanced genetic programming techniques for evolving classifiers in the context of medical diagnosis, Genet. Program. Evolvable Mach. 10 (2) (June 2009) 111–140.
[27] Leonardo Vanneschi, Francesco Archetti, Mauro Castelli, Ilaria Giordani, Classification of oncologic data with genetic programming, Journal of Artificial Evolution and Applications, vol. 2009, Hindawi Publishing Corporation, 2009.
[28] Leonardo Vanneschi, et al., Identification of individualized feature combinations for survival prediction in breast cancer: a comparison of machine learning techniques, 8th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, EvoBIO 2010, LNCS, vol. 6023, Springer, April 7–9 2010, pp. 110–121.
[29] Guiquan Liu, Xiufang Jiang, Lingyun Wen, A clustering system for gene expression data based upon genetic programming and the HS-model, Third International Joint Conference on Computational Science and Optimization (CSO), vol. 1, May 28–31 2010, pp. 238–241.
[30] Kun-Hong Liu, Chun-Gui Xu, A genetic programming-based approach to the classification of multiclass microarray datasets, Bioinformatics 25 (3) (2009) 331–337.
[31] Shital C. Shah, Andrew Kusiak, Data mining and genetic algorithm based gene/SNP selection, Artif. Intell. Med. 31 (2004) 183–196.
[32] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1-2) (1997) 273–324.
[33] A. Geert Heidema, et al., The challenge for genetic epidemiologists: how to analyse large numbers of SNPs in relation to complex diseases, BMC Genet. 7 (2006) 23, http://dx.doi.org/10.1186/1471-2156-7-23.
[34] K. Jesús, Estrada-Gil1, GPDTI: a genetic programming decision tree induction method to find epistatic effects in common complex diseases, Bioinformatics 23 (13) (2007) i167–i174.
[35] Casey S. Greene, Bill C. White, Jason H. Moore, Using expert knowledge in initialization for genome-wide analysis of epistasis using genetic programming, GECCO, ACM, 2008, pp. 351–352.
[36] J.H. Moore, et al., Symbolic modelling of epistasis, Hum. Hered. 63 (2) (2007) 120–133.
[37] Ryan J. Urbanowicz, Bill C. White, Jason H. Moore, Mask Functions for the Symbolic Modelling of Epistasis Using Genetic Programming GECCO, ACM, 2008, pp. 339–346.
[38] Kristine A. Pattin, et al., Exploiting expert knowledge of protein–protein interactions in a computational evolution system for detecting epistasis genetic programming theory and practice VIII, Genetic and Evolutionary Computation, vol. 8, Springer, 20–22 May 2010, pp. 195–210.
[39] Emily Rose Holzinger, et al., Initialization parameter sweep in ATHENA: optimizing neural networks for detecting gene–gene interactions in the presence of small main effects, GECCO '10: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, ACM, 7–11 2010, pp. 203–210.
[40] J.C. Werner, Genetic programming applied to gene function identification, Data mining Cup 2001 of The Seventh ACM SIGKDD International Conference on Knowledge discovery and data mining, August 26–29 2001.
[41] James Cunha, Terence C. Fogarty Werner, Genetic programming applied to gene function identification, Cup 2001 of The Seventh ACM SIGKDD International Conference on Knowledge discovery and data mining — USA/SanFrancisco, August 26–29 2001.
[42] Ivana Vukusic, Sushma Nagaraja Grellscheid, Thomas Wiehe, Applying genetic programming to the prediction of alternative mRNA splice variants, Genomics 89 (2007) 471–479.
[43] Singer X.J. Wang, Peter Lichodzijewski, Boolean genetic programming for promoter recognition in eukaryotes, Congress on Evolutionary Computation, 4151, IEEE, 2005.
[44] William B. Langdon, J. Rowsell, A.P. Harrison, Creating regular expressions as mRNA motifs with GP to predict human exon splitting, GECCO '09: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, ACM, July 8–12 2009, pp. 1789–1790.
[45] S.R. Eddy, Non-coding RNA genes and the modern RNA world, Nat. Rev. Genet. 2 (2001) 919–929.
[46] J.S. Mattick, RNA regulation: a new genetics? Nat. Rev. Genet. 5 (2004) 316–323.
[47] K.M. Wassarman, F. Repoila, C. Rosenow, G. Storz, S. Gottesman, Identification of novel small RNAs using comparative Genomics and microarrays, Genes Dev. 15 (2001) 1637–1651.
[48] S. Chen, et al., Bioinformatics based approach to discover small RNA genes in the Escherichia coli genome, Biosystems 65 (2002) 157–177.
[49] Pa Sætrom, et al., Predicting non-coding RNA genes in Escherichia coli with boosted genetic programming, Nucleic Acids Res. 33 (10) (2005) 3263–3270.
[50] Douglas B. Kell, Genotype–phenotype mapping: genes as computer programs, Trends Genet. 18 (11) (November 2002).