Current and Future Directions for Genetic Algorithms in DNA Array Analysis

E.C. Keedwell and A.Narayanan
{E.C.Keedwell, A.Narayanan}@ex.ac.uk
School of Engineering, Computer Science and Mathematics, Harrison Building,
University of Exeter, Exeter, EX4 4QF, United Kingdom

Abstract. DNA arrays are currently one of the fastest-growing areas of bioinformatics as they allow biologists unprecedented access to the workings of an organism. Genetic algorithms in recent years have found a number of applications in the analysis of DNA arrays. This paper outlines some of the successful genetic algorithm approaches, some problems with their application in this domain and some potential solutions to these problems. Also, the future possibilities for both genetic algorithms and DNA arrays are considered.

1. Introduction – DNA Array Analysis

DNA array analysis and transcriptomics are among the fastest growing areas of bioinformatics and for good reason. Whereas biologists were previously restricted to monitoring a handful of genes at one time, DNA arrays allow them to simultaneously measure the expression levels of thousands of genes through their corresponding mRNA (the transcript of a gene prior to translation to protein). The complete collection of mRNAs and their alternative splice forms is usually referred to as the transcriptome, the set of instructions for creating all of the different proteins found in an organism. DNA arrays cover two types of technology, depending on how DNA nucleotide sequences are put onto the chip. 'Microarrays' use pre-synthesized DNA (about 100 bases) for probing, whereas 'gene chips' use *in situ* synthesized oligonucleotide probes (e.g. 25 bases for Affymetrix gene chips). Generally speaking, the analysis of these DNA arrays falls into two areas, classification and temporal analysis. The task for classification is to group a set of DNA arrays (samples - often patients with a diagnosis) into separate groups or classes, according to the profiles of their genes. In temporal analysis the same organism is sampled repeatedly to determine the change in activity of genes over time. Here the analysis technique can look at clustering the genes according to the change in value over time or more ambitiously, attempt to reverse engineer the genetic network, a set of causal connections between genes. However, the major problem is that typically thousands of genes are measured (Affymetrix gene chips can already measure the full transcriptome equivalent of the human genome of 30,000 or so genes) for each sample, and only a few dozen samples exist. In database terms, there is often a 1:100 ratio between rows of a database (samples, or individuals) and attributes (or genes). A typical DNA array database, for instance, consists of about 100 rows (samples) and minimally 10,000 columns (genes). Clustering and other standard data analysis techniques often face difficulty in returning statistically significant results given the sparsity of samples in comparison to attributes. This also assumes that the data is reliable and free of noise, which is currently not the case given the early stage of DNA array technology.

The sparsity and noise problems have led some researchers to look for alternative, 'softer' analytical methods, such as neural networks and decision trees (see Narayanan, Keedwell and Olsson (2002) for an overview of these alternative approaches). Genetic algorithms (GA), however, have also been at the forefront of these approaches and the following sections discuss various aspects of GA approaches to these DNA array analyses and future directions for GAs in this domain.

2. Genetic Algorithms in DNA Array Analysis

2.1 Success - GA Applications to DNA array Data

Whilst genetic algorithms have found applications to a number of problems in bioinformatics, it seems that the majority of this research has focussed on analysing DNA array data. Ando and Iba (2000, 2001a, 2001b) have applied genetic algorithms in a number of ways to DNA array data to discover classification rules and gene expression pathways. Similarly Keedwell and Narayanan (2003) have created a neural-genetic hybrid which can classify data and reverse engineer gene networks from DNA array data. Along with the application of neural networks and decision trees in bioinformatics, the application of genetic algorithms to gene expression data is still at a preliminary stage. Yet the domain appears highly suitable for the application of GAs. So why are there not more genetic algorithm, or more generally evolutionary computation, solutions to DNA array problems?

Let us first sketch out the domain in more detail:

1. DNA arrays are a major step forward in the process of understanding the underlying genetics of disease. They can discover the simultaneous expression values of thousands of genes where previously only a handful could be identified.

Therefore they represent an unprecedented view of the mechanics of life at the most fundamental level.

2. The data is amenable to machine learning analysis. The output of an actual DNA array is noisy and full of statistical effects such as print-head and slide effects where the mechanics of the DNA array process affect the data. However, a number of statistical algorithms can be used to mitigate these effects and the output is often a reasonably noise-free set of floating point or binary numbers.

3. The data is prevalent on the internet. A variety of sites offer experimental DNA array data online and therefore this is an easy source of real data to test machine learning approaches.

Therefore it is understandable that many of the GA bioinformatics applications so far relate to the analysis of DNA array data. The success of the GA on this type of data can be attributed to the fact that they are ideally suited to this type of task. In most DNA array data there are a vast number of variables (genes) and only a small number of records (samples). This is due to the fact that DNA arrays are costly to experiment with and therefore samples are at a premium. A number of gene reduction approaches (such as principal component analysis) can be used to reduce the dimensionality of DNA array data, but GAs remain one of the few approaches which can act directly on the full data. The ability of GAs to efficiently search these large spaces is one of the reasons for their success in this field.

2.2 Problems – Why GAs don't work

The reasons why GAs are difficult to apply to DNA array tasks are similar to those that can be used against the algorithm in other applications.

Firstly, GAs being a stochastic algorithm, are not guaranteed to find the optimal answer for a given problem, even assuming that there is an optimum solution to be found. In many DNA array experiments there are a number of equally good solutions and therefore this can mean that the GA will arrive at one of a number of these solutions depending on the random seed.

Secondly, the GA uses a large number of objective function evaluations to achieve the final solution. Whilst this number of evaluations is only a minute fraction of the total possible combinations of genes, depending on the structure of the chromosome, this can incur overly long running times for the algorithm. It is fair to say that this is reasonably algorithm and application dependent, but it remains a hurdle for some GA applications.

Finally, there is a less tangible obstacle to the uptake of genetic algorithm technologies in DNA arrays and perhaps the broader topic of bioinformatics, and that is the perception of GAs. In a field where real genetics are the primary concern, the biologically-inspired operators used in GAs can plant the first seeds of doubt about the algorithm for biologists. The way GA specialists use the terms selection, crossover and mutation can cause difficulty for a variety of people when they try and square them with the actual methods of the same name in nature.

2.3 Solutions - How these problems can be solved

As advocated in a number of recent papers including Moore (2003) and Rowland (2003), the problem of the stochastic element in genetic algorithms can be mitigated by using correct cross-validation and sampling techniques to verify results. By using a minimum three-fold cross-validation procedure for classification tasks, (that is training, tuning and testing on three separate datasets and repeating the process for each dataset), a true notion of classification accuracy can be obtained. This method is advantageous in that it can be used on relatively small sample numbers which is almost always the case in DNA array experiments

The second problem of complexity is more difficult to alleviate as genetic algorithms currently are one of the most efficient search tools for this purpose. However, GA research and computational power are continuing apace so this will ease some of the problems of computational complexity. Another interesting avenue of research is the use of hybrid techniques which can enhance genetic algorithms and reduce the number of model evaluations required to discover an interesting and hopefully near-optimal solution. As previously described the representation of the problem to the GA can also significantly reduce complexity issues. For instance, when reverse engineering genetic networks, it is not necessary to use the GA to evolve a complete set of connections between genes. Chaos theory has postulated (Kauffman, 1996) and experimental biology (Thieffry et al, 1998) has confirmed that only a small number (< 6) genes effect each other in the genetic network. Therefore by making use of these biologically determined constraints the complexity of the GA can be reduced dramatically.

The final problem is not so easily solved as it requires that GAs are perceived correctly. Asserting that evolutionary methods are simply search and optimisation techniques rather than a biological metaphor is especially difficult in this domain.

3. The Future - Multi-Objective GAs

Despite the problems shown above, GAs remain one of the most likely techniques to discover genuinely interesting information and structures from DNA array data. Problems 1 and 3 however could be mitigated by using multi-objective approaches to DNA array data analysis. Multi-objective algorithms offer the end user a much broader perspective on the problem being solved as they allow them to see the trade-off between conflicting objectives in the problem. The use of multi-objective techniques is now *de rigeur* in engineering disciplines as it is understood that there is potentially more than one answer to a particular problem. Therefore in a biology scenario where there are many seemingly optimal answers, the discovery of a set of solutions rather than a single one (which can change with random seed) could find success. In addition to this, the answers that constitute the pareto-front are likely to have some similarity in their structure and therefore this can allay the fears of biologists who doubt the consistency of the genetic algorithm.

In traditional DNA array analysis it may appear that there is little scope for multi-objective optimisation, as classification models and gene networks use a single measure, accuracy, on the data to determine their optimality. However, a number of strategies exist to break down the problem into a number of objectives. One such strategy would be to use model complexity as a secondary objective in addition to accuracy and thereby converting a single-objective problem to a multi-objective one. The domination principle clearly works for this formulation for the classification of DNA arrays, as a solution which considers many genes is likely to be less favourable than one which requires a small number of genes and has the same accuracy. It is impossible to tell, however, the necessary number of genes to attain a certain accuracy on the dataset and this is why a multi-objective approach is required.

Using multi-objective technology, the biologist therefore can pick a solution according to its complexity which is most consistent with the biological experiment being conducted. If, for instance, two genes are suspected to be involved in a disease those solutions with 1-3 genes can be most closely scrutinised. Also, it may be that whilst the 2 gene solution is good, the 1 and 3 gene solutions could also be interesting, if not better than that 2 gene solution.

4. The Future - DNA arrays and genetic networks

The combination of classification and temporal studies enables individuals to be classified and compared based on their genetic networks – a model which dictates the expression levels of genes in the body based on expression levels at a previous timestep. This type of study would overcome some of the diagnostic obstacles of DNA array classification, for instance determining at what point in the development of a cancer the sample has been taken. In addition to this, the progression of diseases can be mapped and key points in their development identified. What this means for the GA, however, is even greater complexity. A two dimensional problem (genes * sample (classification) or genes * time (temporal)) has become a three dimensional one (genes * sample * time). This even greater complexity is bound to ensure that genetic algorithm researchers have plenty of work in bioinformatics in the future. The future challenge for genetic algorithms is the reverse engineering of gene networks from temporal DNA array data that samples across individuals as well as time. Such networks then provide clinicians with focused drug targets that will enable them to control the development of a disease as well as provide individualised drug treatments depending on the stage of disease reached (pharmacogenomics).

5. References

Ando, S., Iba H.,. (2001a) "Inference of Gene Regulatory Model by Genetic Algorithms", Proceedings of Conference on Evolutionary Computation 2001 pp712-719

Ando, S., Iba H., (2001b) "The Matrix Modeling of Gene Regulatory Networks - Reverse Engineering by Genetic Algorithms-", Proceedings of Atlantic Symposium on Computational Biology, and Genome Information Systems & Technology 2001.

Ando, S., Iba H., (2000) "Inference of Gene Regulatory Model by Genetic Algorithms", Proceeding of International Symposium on Adaptive Systems

Kauffman, S. (1996) At Home in the Universe: The Search for Laws of Self-Organization and Complexity. Penguin Books

Keedwell E., Narayanan A., (2003) “Genetic algorithms for gene expression analysis" in Applications of Evolutionary Computing LNCS 2611 Gunther Raidl et al (Eds.), proceedings of EvoBIO2003 1st European Workshop on Evolutionary Bioinformatics pp 76-86

Moore, J. (2003) “Cross Validation Consistency for the Assessment of Genetic Programming Results in Microarray Studies” Applications of Evolutionary Computing LNCS 2611 Gunther Raidl et al (Eds.), proceedings of EvoBIO2003 1st European Workshop on Evolutionary Bioinformatics pp 99-106

Narayanan, A., Keedwell, E.C., Olsson, B. (2002) “Artificial Intelligence Techniques for Bioinformatics”, Applied Bioinformatics 1(4): 191-222.

Rowland, J. (2003) “Generalisation and Model Selection in Supervised Learning with Evolutionary Computation” Applications of Evolutionary Computing LNCS 2611 Gunther Raidl et al (Eds.), proceedings of EvoBIO2003 1st European Workshop on Evolutionary Bioinformatics pp 119-130

Thieffry, D., Huerta, A.M., Perez-Rueda, E., Collado-Vides, J. (1998) “From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in Escherichia coli” BioEssays Vol 20. pp 433-440 John Wiley and Sons Inc.