*Gene interaction networks boost genetic algorithm performance in biomarker discovery*

*Charalampos Moschopoulos\*, Dusan Popovic\*, Rocco Langone, Johan Suykens, Bart De Moor, Yves Moreau*

Department of Electrical Engineering (ESAT),

STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics / iMinds Medical IT

KU Leuven

Leuven, Belgium

{cmoschop, dusan.popovic, rocco.langone, johan.suykens, bart.demoor, yves.moreau}@esat.kuleuven.be

*Abstract*—**In recent years, the advent of high-throughput techniques led to significant acceleration of biomarker discovery. In the same time, the popularity of machine learning methods grown in the field, mostly due to inherit analytical problems associated with the data resulting from these massively parallelized experiments. However, learning algorithms are very often utilized in their basic form, hence sometimes failing to consider interactions that are present between biological subjects (i.e. genes). In this context, we propose a new methodology, based on genetic algorithms, that integrates prior information through a novel genetic operator. In this particular application, we rely on a biological knowledge that is captured by the gene interaction networks. We demonstrate the advantageous performance of our method compared to a simple genetic algorithm by testing it on several microarray datasets containing samples of tissue from cancer patients. The obtained results suggest that inclusion of biological knowledge into genetic algorithm in the form of this operator can boost its effectiveness in the biomarker discovery problem.**

Keywords—biomarker discovery; genetic algorithm; gene interaction network; microarray gene expression datasets

# Introduction

In the high-throughput era, many biological components of a cell can be simultaneously examined. Very often, the goal of such analyses is to reveal molecular mechanisms behind certain diseases and/or to detect disease hallmarks. Biomarkers are measurable indicators of a specific biological state, and they can be used to determine the presence or the stage of a disease [1]. Automatic discovery of these has the potential to aid drug development via reduction of the clinical trials costs and also many times can help in development of diagnostic applications for detection or prediction of a disease status [2]. However, the process of biomarker mining from high-throughput data is not trivial for several reasons. This data is characterized by extreme dimensionality and high levels of measurement noise [3]. In parallel, often restricted number of available samples makes classical statistical techniques unsuitable for this kind of computational problems.

Gene expression analysis has been one of the most promising resources for biomarker discovery [4,5], as monitoring genes that are differentially expressed between normal and diseased tissue samples might indicate potential predictors. Various computational methods have been applied to facilitate this process, including many machine learning-based approaches [6,7]. Additionally, the genetic algorithms (GAs) have proved their efficacy in this field as they scale well with increasing dimensionality of the feature selection problems and sometimes display better performance that other popular feature selection strategies [8,9]. However, none of the aforementioned methods directly takes into consideration the network interaction properties of genomic data that can be derived from various biological databases.

\* Equally contributing first authors.

In this contribution, we present an enhanced version of a simple genetic algorithm that utilizes an additional operator based on the gene interaction network, aiming to detect the most informative set of biomarkers that characterize certain biological condition. In order to prove the efficacy of the proposed method, we perform extensive experiments using 3 different gene expression datasets on cancer containing samples of normal and diseased tissue. We compare our results with that obtained with a simple GA across four algorithms used in the subsequent classification step. Finally, to facilitate better insight into different aspects of the classification performance we provide several complementary metrics: balanced accuracy, Sensitivity, Specificity, Precision, Negative predictive value, Mathews correlation coefficient and area under the ROC curve, all obtained on independent testing partitions extracted from the data sets.

# METHOD

Genetic algorithms (GAs) are stochastic optimization methods that simulate the process of natural evolution [10, 11]. Based on this paradigm, they use three simple operators (selection, crossover and mutation), which allow them to evolve and reach near-optimal solutions [12, 13]. GAs have been successfully utilized in many different scientific fields, for solving optimization and search problems. They also have been used in many bioinformatics applications, including RNA structure prediction, multiple sequence alignment, microarray data classification and others [14, 15].

Within biomarker discovery, GAs were applied before, producing solutions with fixed number of selected biomarkers [16, 17]. A similar approach has been generalized in [8], where a simple GA explicitly lead to a smallest possible solution (in terms of the number of biomarkers selected) through explicit encoding of shrinkage pressure in its fitness function. Our method is based on the simple GA setup as introduced in the aforementioned work [8, 9], with enhancement in form of an additional operator that uses the knowledge contained in gene interaction networks. In this setup, a gene of the GA represents the presence or absence of a biological gene expression among the features of the classification algorithm by taking one of two possible binary values (1/0). Therefore, length of each GA chromosome corresponds to a number of biological genes that can be found on a microarray chip.

The fitness function encodes optimization goals and therefore represents a central GA design choice. Here it is constructed to capture two desired aspects of discovered biomarker panels – parsimony and classification performance. This is accomplished by taking balanced accuracy obtained by cross validation as the indicator of fitness while penalizing solutions that contain many genes. Formally, the fitness function is defined as:

 , (1)

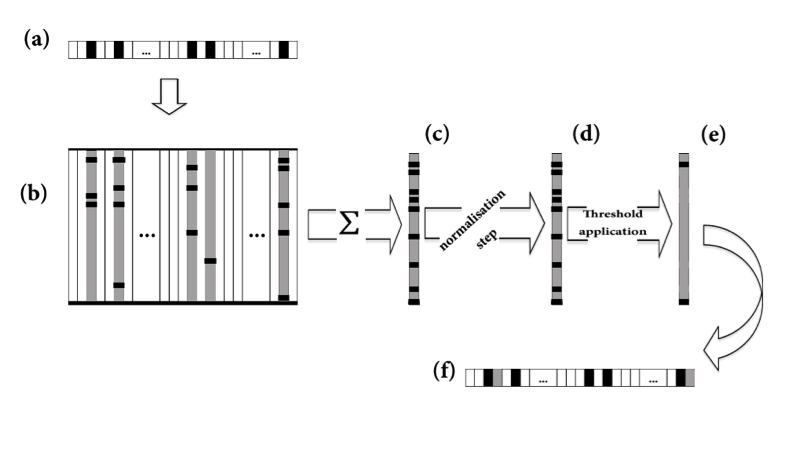
where *Fc* stands for the value of the fitness function corresponding to the chromosome *c, Bc*  for the balanced accuracy, *Sc* for the number of genes used as predictors and *Si* for the mean number of genes that were selected in the initial population of the GA, across all chromosomes. The value one is added for the computational convenience, i.e. to assure that all fitness values remain positive. The balanced accuracy (*Bc*) is obtained by 10-fold cross validation of the one nearest neighbour (1-NN) classifier on a partition of the original dataset corresponding to the selected features. We choose 1-NN over more complex classification algorithms as it is very fast to evaluate and still able to capture non-linear relationships in the data. It also does not require any parameter tuning and it asymptotically achieves Bayes error within a constant factor [18]. Additionally, the 1-NN algorithm has been shown as able to consistently outperform several more complex classification algorithms in some applications [19], and it was already successfully utilized in a similar context [8]. The parsimony in terms of selected genes (i.e. shrinkage) is enforced by the explicit penalization of long and rewarding of short solutions, proportionally to the relative gain/loss in number of selected features compared to the initial population (the last term of the equation 1).

The proposed new operator uses information provided by the genetic interaction network, and it is triggered after the execution of crossover and mutation operators. The “birds-eye” view on its functioning mechanism is provided on Fig. 1. Assume the GA chromosome solution presented at Fig. 1 (a), where the black cells encode the selected genes. In the first step, the interaction weights (from interaction matrix) of their adjacent genes are summed up. In this way, the genes most “relevant” to the activated set receive the highest scores. The weight values are then scaled such to sum to one.

In the next step, the resulting probabilities are multiplied by the operator’s basic probability of activation per gene (in this work 1%) and the total number of activated genes in a chromosome (Fig. 1 (d)). Finally, we generate a vector of numbers randomly sampled from the standard uniform distribution (U(0,1)) of the same length as GA chromosome; after which each gene whose probability of selection surpasses a corresponding randomly generated threshold is added to the chromosome (Fig. 1 (f)). It is worth to note that this design can be (and it was in this case) fully vectorized, considerably speeding up the GA in which it is used.

We hypothesize that using information captured by the interaction network might facilitate “hill climbing” behaviour of a genetic algorithm in a vicinity of a local optimum. That is, if the GA triggers the relevant pathway(s), this enhancement can help in fine-tuning of the gene panel used. In this setup, the extent of potential gains in GA performance strongly depends on quality of information that is embedded in a network. For example, if a network is highly corrupted by the noise, proposed operator practically adds random genes to each generated solution; degrading overall performance of the algorithm. However, we do not expect that performance of the enhanced GA could drop bellow that of simple GA even if interaction network is completely uninformative, as than genes added by the operator would be pruned-off in subsequent steps of GA execution.

We utilize the experimentally verified version of String 9.1 network [20] to make sure that only the proven links between genes are present. This network is consisted of approximately 20,000 genes (human genome). Also, the String network subsumes the data from other interaction databases, providing a global view on all the available interaction information. Moreover, for each interaction it provides a continuous score, which is representative of the interaction reliability, as described extensively in [20]. These interaction matrix weights are calculated by combining the probabilities from the different evidence channels, correcting for the probability of randomly observing an interaction (detailed treatment can be found in [21]). Finally, the String database is considered as one of the best resources of this kind of data [22] and it is widely used in common practice for various biological analyses [23].



1. The functionality of the proposed operator. (a) A chromosome instance of GA. Black cells represent activated genes. (b) The respective columns of activated genes of the genetic interaction network adjacency matrix (this is a sparse matrix having genes both in rows and columns, thus of approximate size 20.000 x 20.000). Black cells represent the weight scores of the adjacent genes to the activated set. (c) Vector containing the summed weights of adjacent genes. (d) Normalized vector. The genes surpassing randomly assigned thresholds (e) are added to the chromosome solution (f).

# EXPERIMENTS

## Microarray gene expression datasets

In our experiments we used 3 publicly available datasets from the GEO database [24]. All three gene expression datasets were generated on Affymetrix U133 plus 2.0 whole genome microarray platform and pre-processed by RMA [25]. The tissue status (diseased or healthy sample) was considered as an outcome of interest in all cases.

The first dataset contains normal and two distinctive classes of hepatocellular carcinomas (HCC), which are highly associated with survival of the patients [26]. This dataset includes 16559 genes and 182 samples. Through the rest of the paper we refer to this data as to hepatocellular carcinoma dataset. The second dataset contains paired normal and tumor tissue samples obtained at the time of surgery from resected pancreas of 36 pancreatic cancer patients [27]. All the patients were suffering from PDAC. In total, this dataset includes 19898 genes and 78 genechip hybridizations have been performed. In the rest of the paper we refer to this data as to pancreatic cancer dataset. The third dataset contains deregulated genes in cardia and non-cardia gastric cancer, obtained from cardia and non-cardia gastric tumors [28]. This dataset includes 12101 genes and 268 samples. In the rest of the paper we refer to this data as to gastric cancer dataset.

## Benchmark setup

The benchmark covers eight combinations of 2 feature selection methods (proposed enhanced GA and simple GA [8]) and 4 classifiers (Decision tree [29], Naive Bayes [30], Random Forest [31] and Neural Network [32]) to rule-out possibility that the obtained result comes from a specific classifier-feature selection interaction. We chose to use these particular classifiers for their diversity and a relative robustness with regard to parameter settings. In particular, the decision tree is based on the CART algorithm with pruning. Random forest is composed out of 100 trees and the number of variables considered in each node is the square root of their total number (default value, [31]). Finally, we use a back-propagation trained, one hidden layer feed-forward neural network with 5 hidden units.

In total, three publicly available data sets containing gene expression profiles were used for the algorithm benchmarking. Each of the datasets has been initially randomly partitioned into the three segments - one for the feature selection, one for the classifier training and one for the testing. It is important to stress that both of the genetic algorithm-based approaches rely on a cross validation of a *feature selection partition*, thus they do not utilize information from the classifier-training or the testing part. To assure stability of the results, this whole procedure is repeated one hundred times for all datasets, each time with a different random data split.

To measure added value of the new operator, we used exactly the same configuration for the enhanced and simple GA. The code implementing GAs is based on the SpeedyGA.m 1.2 Matlab script [33]. Each GA population is consisted of 100 chromosomes, and to initialize a population, we randomly generate individuals with chance of 1% for each feature (gene) to be set to 1. The probability of mutation per bit of individual chromosome has been set to 0.5 divided by the maximal length of chromosome. We use uniform crossover, with the probability of reproduction without it set to 0.3. Selection is preformed proportionally to the sigma-scaled value [34] of the fitness function using the stochastic universal sampling. We restrict the maximal number of generations to 200 and keep track on the best solution over all generations. As suggested by the evolution curves displayed on Fig. 2, this number of generations is adequate for GA in terms of convergence.

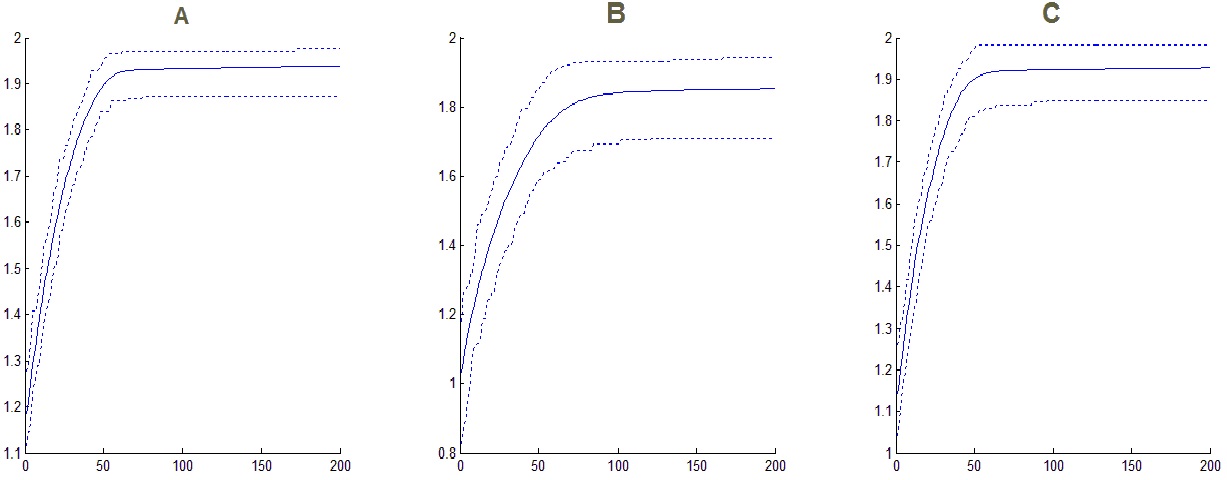


Fig 2. Convergence of enhanced GA. Dotted lines are minimal/maximal and solid line are mean values of the best obtained fitness in each generation, across all 100 runs of benchmark. The panel (a) corresponds to Hepatocellular carcinoma dataset, (b) to Pancreatic cancer dataset and (c) to Gastric cancer dataset.

Comparison of the enhanced and simple Genetic algorithm combined with 4 different classifiers, with regard to the average values of balanced accuracy (Acc.), Sensitivity (Sens.), Specificity(Spec.), Precision (Prec.), Negative Predictive Value (NPV), Mathews Correlation Coefficient (MCC) and Area under the ROC curve (AUC) obtained on the test set. The classifiers used are Decision Trees (DT), Naive Bayes (NB), Random Forest (RF) and Neural Network (NN). Shaded fields indicate the feature selection/classifier combination that gave rise to the highest average value of given performance measure that is obtained for single data set.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Enhanced genetic algorithm** | | | | **Simple genetic algorithm** | | | |
| **Dataset** |  | **DT** | **NB** | **RF** | **NN** | **DT** | **NB** | **RF** | **NN** |
| **Hepatocellular carcinoma** | **Acc** | 0.9676 | 0.9875 | 0.9873 | 0.9873 | 0.9699 | 0.9858 | 0.9828 | 0.9828 |
| **Sens.** | 0.9603 | 0.9868 | 0.9848 | 0.9848 | 0.9691 | 0.9843 | 0.9833 | 0.9833 |
| **Spec.** | 0.9750 | 0.9882 | 0.9897 | 0.9897 | 0.9706 | 0.9873 | 0.9824 | 0.9824 |
| **Prec** | 0.9755 | 0.9887 | 0.9900 | 0.9900 | 0.9723 | 0.9876 | 0.9833 | 0.9833 |
| **NPV** | 0.9626 | 0.9872 | 0.9855 | 0.9855 | 0.9709 | 0.9849 | 0.9840 | 0.9840 |
| **MCC** | 0.9367 | 0.9754 | 0.9750 | 0.9750 | 0.9415 | 0.9720 | 0.9665 | 0.9665 |
| **AUC** | 0.9698 | 0.9965 | 0.9991 | 0.9917 | 0.9737 | 0.9974 | 0.9979 | 0.9885 |
| **Pancreatic cancer** | **Acc** | 0.7824 | 0.8417 | 0.8391 | 0.8391 | 0.7845 | 0.8439 | 0.8348 | 0.8348 |
| **Sens.** | 0.7585 | 0.8747 | 0.8656 | 0.8656 | 0.7796 | 0.8637 | 0.8577 | 0.8577 |
| **Spec.** | 0.8067 | 0.8085 | 0.8122 | 0.8122 | 0.7900 | 0.8229 | 0.8110 | 0.8110 |
| **Prec** | 0.8092 | 0.8354 | 0.8360 | 0.8360 | 0.8015 | 0.8433 | 0.8316 | 0.8316 |
| **NPV** | 0.7869 | 0.8756 | 0.8679 | 0.8679 | 0.7980 | 0.8647 | 0.8606 | 0.8606 |
| **MCC** | 0.5798 | 0.6965 | 0.6904 | 0.6904 | 0.5840 | 0.6969 | 0.6801 | 0.6801 |
| **AUC** | 0.8022 | 0.8846 | 0.9101 | 0.7956 | 0.7969 | 0.8928 | 0.9083 | 0.7909 |
| **Gastric cancer** | **Acc** | 0.8857 | 0.9226 | 0.9331 | 0.9331 | 0.8883 | 0.9194 | 0.9245 | 0.9245 |
| **Sens.** | 0.8782 | 0.9285 | 0.9341 | 0.9341 | 0.8920 | 0.9197 | 0.9244 | 0.9244 |
| **Spec.** | 0.8934 | 0.9167 | 0.9322 | 0.9322 | 0.8848 | 0.9192 | 0.9246 | 0.9246 |
| **Prec** | 0.8966 | 0.9209 | 0.9341 | 0.9341 | 0.8891 | 0.9215 | 0.9264 | 0.9264 |
| **NPV** | 0.8831 | 0.9293 | 0.9352 | 0.9352 | 0.8945 | 0.9216 | 0.9261 | 0.9261 |
| **MCC** | 0.7756 | 0.8477 | 0.8678 | 0.8678 | 0.7802 | 0.8410 | 0.8508 | 0.8508 |
| **AUC** | 0.8988 | 0.9644 | 0.9750 | 0.9492 | 0.9013 | 0.9633 | 0.9690 | 0.9520 |

# RESULTS AND DISCUSSION

Table 1 enlists the average values of various classification metrics obtained by the application of different combinations of feature selection methods and classifiers. It is immediately apparent that, regardless to methods used, the pancreatic dataset poses the most challenging problem, while hepatocellular carcinoma is the easiest to learn from. Also, the enhanced version of GA outperforms the simple one in almost all cases, irrespective to the classifier used and the performance metric considered. The exception from this is the pancreatic cancer dataset, where for some of the metrics simple GA version prevails. However, that is not a case for the most general measure used, i.e. for the area under the ROC curve. Furthermore, it seems that the combination of enhanced GA and Naive Bayes or Random Forests generally displays the best performance. The classification pipeline that relies on Naive Bayes works best on the hepatocellular carcinoma dataset, while Random forest and Neural Network classifiers are performing better on the Gastric carcinoma dataset. On the Pancreatic carcinoma dataset, Naive Bayes and Random Forest classifiers behave similarly.

The observed difference in performance between the two methods is apparently not very large in all of the cases. The one reason for this is the fact that these data sets pose relatively easy classification problems, letting relatively simple methods to distinguish informative from non-informative features. Secondly, having in mind that this particular GA is designed and tuned especially for the biomarker discovery task [8], the difference in performance, even being relatively small, clearly indicates that addition of gene interaction data via the new operator has potential to improve GA-based prediction.

Figure 3 depicts median lengths of final genetic signatures produced by the enhanced and simple GA on three data sets. Typically, the enhanced GA selects few predictors more than its simple counterpart. This result was somewhat expected as the proposed operator works against parsimony by activating additional genes in each solution during each generation. Therefore, if the size of a signature is of critical importance in particular application, the discussed effect can be reversed by increasing parsimony pressure compared to that in simple GA. For example, the weight of size penalty term in the fitness function (here -1) could be increased in relation to basic probability of activating additional operator (here 1%) to compensate for the effect.

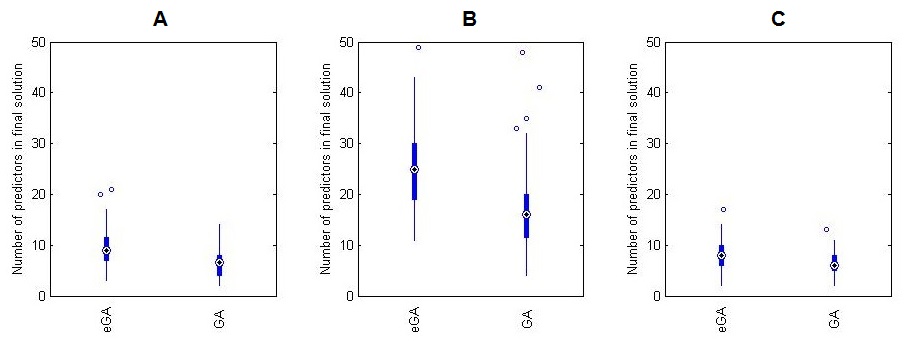


Fig 3. Final number of genes selected by the enhanced (eGA) and the simple genetic algorithm (GA) during 100 randomization runs. The panel (a) corresponds to Hepatocellular carcinoma dataset, (b) to Pancreatic cancer dataset and (c) to Gastric cancer dataset.

# CONCLUSIONS

In this paper we proposed a new operator for genetic algorithms, based on knowledge derived from genetic interaction networks and demonstrated it in the context of biomarker discovery problem. We prove efficacy of this enhanced GA by testing it against its simple counterpart on three different gene expression datasets. Besides that, we prove the robustness of the proposed method with regard to choice of algorithm used in the classification step. In the future we plan to create similar operators for GA in order to integrate gene association’s knowledge derived from other biological resources such as pathways, text mining, structural properties and more.

##### Acknowledgment

BDM and YM are full professors at the Katholieke Universiteit Leuven, Belgium. Research supported by: Research Council KU Leuven: GOA/10/09 MaNet, CoE PFV/10/016 SymBioSys, several PhD/postdoc & fellow grants; Industrial Research fund (IOF): IOF/HB/13/027 Logic Insulin, IOF: HB/12/022 Endometriosis, IOF/KP (Identification and development of new classes of immunosuppressive compounds and discovery of new key proteins involved in the T and B-cell activation); Flemish Government: FWO: PhD/postdoc grants, projects: G.0871.12N (Neural circuits); IWT: PhD Grants; TBM-Logic Insulin (100793), TBM Rectal Cancer (100783), TBM IETA (130256), O&O ExaScience Life Pharma, ChemBioBridge; Hercules Stichting: Hercules III PacBio RS, Hercules 1: The C1 single-cell auto prep system, BioMark HD System and IFC controllers (Fluidigm) for single-cell analyses; iMinds: Medical Information Technologies SBO 2014; Art&D Instance; VLK Stichting E. van der Schueren: rectal cancer, VSC Tier 1: exome sequencing; Federal Government: FOD: Cancer Plan 2012-2015 KPC-29-023 (prostate), COST: Action BM1104: Mass Spectrometry Imaging

##### References

1. N. Rifai, M. A. Gillette, and S. A. Carr, "Protein biomarker discovery and validation: the long and uncertain path to clinical utility," Nat Biotechnol, vol. 24, pp. 971-83, Aug 2006.
2. B. Williams-Jones, "History of a gene patent: tracing the development and application of commercial BRCA testing," Health Law J, vol. 10, pp. 123-46, 2002.
3. J. H. Phan, R. A. Moffitt, T. H. Stokes, J. Liu, A. N. Young, S. Nie, et al., "Convergence of biomarkers, bioinformatics and nanotechnology for individualized cancer treatment," Trends Biotechnol, vol. 27, pp. 350-8, Jun 2009.
4. L. J. van't Veer and R. Bernards, "Enabling personalized cancer medicine through analysis of gene-expression patterns," Nature, vol. 452, pp. 564-570, 2008.
5. J. Hoefkens, "Towards unbiased biomarker discovery," Drug Discovery World, vol. Spring, pp. 19-24, 30/3/2010 2010.
6. A. M. Glas, A. Floore, L. J. M. J. Delahaye, A. T. Witteveen, R. C. F. Pover, N. Bakx, et al., "Converting a breast cancer microarray signature into a high-throughput diagnostic test," BMC genomics, vol. 7, p. 278, 2006.
7. R. Brandt, R. Grutzmann, A. Bauer, R. Jesnowski, J. Ringel, M. Lohr, et al., "DNA microarray analysis of pancreatic malignancies," Pancreatology, vol. 4, pp. 587-97, 2004.
8. D. Popovic, A. Sifrim, G. A. Pavlopoulos, Y. Moreau, and B. De Moor, "A simple genetic algorithm for biomarker mining," in Pattern Recognition in Bioinformatics, ed: Springer, 2012, pp. 222-232.
9. C. Moschopoulos, D. Popovic, A. Sifrim, G. Beligiannis, B. De Moor, and Y. Moreau, "A genetic algorithm for pancreatic cancer diagnosis," in Engineering Applications of Neural Networks, ed: Springer, 2013, pp. 222-230.
10. T. Bäck, Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms vol. 996: Oxford university press Oxford, 1996.
11. T. Bäck, D. B. Fogel, and Z. Michalewicz, Evolutionary computation 2: advanced algorithms and operators vol. 2: CRC Press, 2000.
12. D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning.: Addison-Wesley, 1989.
13. Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs. New York: Springer-Verlag, 1999.
14. S. Bandyopadhyay and S. K. Pal, Classification and learning using genetic algorithms: Springer Heidelberg, 2007.
15. C. Gondro and B. P. Kinghorn, "A simple genetic algorithm for multiple sequence alignment," Genet Mol Res, vol. 6, pp. 964-82, 2007.
16. L. Jourdan, C. Dhaenens, and E.-G. Talbi, "A genetic algorithm for feature selection in data-mining for genetics," Proceedings of the 4th Metaheuristics International ConferencePorto (MIC’2001), pp. 29-34, 2001.
17. T. Jirapech-Umpai and S. Aitken, "Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes," BMC bioinformatics, vol. 6, p. 148, 2005.
18. C. J. Stone, "Consistent nonparametric regression," The annals of statistics, pp. 595-620, 1977.
19. C. Stanfill and D. Waltz, "Toward memory-based reasoning," Communications of the ACM, vol. 29, pp. 1213-1228, 1986.
20. A. Franceschini, D. Szklarczyk, S. Frankild, M. Kuhn, M. Simonovic, A. Roth, et al., "STRING v9. 1: protein-protein interaction networks, with increased coverage and integration," Nucleic acids research, vol. 41, pp. D808-D815, 2013.
21. C. Von Mering et al. "STRING: known and predicted protein–protein associations, integrated and transferred across organisms." Nucleic acids research 33.suppl 1 (2005): D433-D437.
22. T. Klingstrom and D. Plewczynski, "Protein-protein interaction and pathway databases, a graphical review," Brief Bioinform, vol. 12, pp. 702-13, Nov 2011.
23. N. C. Santos, M. O. Pereira, and A. Lourenco, "Pathogenicity phenomena in three model systems: from network mining to emerging system-level properties," Brief Bioinform, Oct 7 2013.
24. R. Edgar, M. Domrachev, and A. E. Lash, "Gene Expression Omnibus: NCBI gene expression and hybridization array data repository," Nucleic Acids Res, vol. 30, pp. 207-10, Jan 1 2002.
25. R. A. Irizarry, B. Hobbs, F. Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U. Scherf, et al., "Exploration, normalization, and summaries of high density oligonucleotide array probe level data," Biostatistics, vol. 4, pp. 249-64, Apr 2003.
26. J. S. Lee, I. S. Chu, A. Mikaelyan, D. F. Calvisi, J. Heo, J. K. Reddy, et al., "Application of comparative functional genomics to identify best-fit mouse models to study human cancer," Nat Genet, vol. 36, pp. 1306-11, Dec 2004.
27. L. Badea, V. Herlea, S. O. Dima, T. Dumitrascu, and I. Popescu, "Combined Gene Expression Analysis of Whole-Tissue and Microdissected Pancreatic Ductal Adenocarcinoma identifies Genes Specifically Overexpressed in Tumor Epithelia-The authors reported a Combined Gene Expression Analysis of Whole-Tissue and Microdissected Pancreatic Ductal Adenocarcinoma identifies Genes Specifically Overexpressed in Tumor Epithelia," Hepato-gastroenterology, vol. 55, p. 2016, 2008.
28. G. Wang, N. Hu, H. H. Yang, L. Wang, H. Su, C. Wang, et al., "Comparison of global gene expression of gastric cardia and noncardia cancers from a high-risk population in china," PLoS One, vol. 8, p. e63826, 2013.
29. L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, (1984). Classification and regression trees. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software. ISBN 978-0-412-04841-8.
30. T. Bayes, (1763), "An essay towards solving a Problem in the Doctrine of Chances." , Philosophical Transactions of the Royal Society of London, Vol. 53, p. 370
31. L. Breiman (2001). “Random Forests.” Machine Learning 45 (1): 5–32.
32. D. E. Rumelhart, G. E. Hinton, R. J. Williams, "Learning representations by back-propagating errors". Nature 323 (6088): 533–536. , 1986
33. K. M. Burjorjee and J. B. Adviser-Pollack, Generative fixation: a unified explanation for the adaptive capacity of simple recombinative genetic algorithms: Brandeis University, 2009.
34. M. Mitchell, An introduction to genetic algorithms: MIT press