# Breast Cancer Biomarker Selection Using Multiple Offspring Sampling

A. LaTorre[1], J.M. Peña[1], S. González[1], O. Cubo[1], F. Famili[2]

[1] Computer Architecture Department, Universidad Politécnica de Madrid
Boadilla del Monte, Madrid, 28660, Spain
{atorre, jmpena, sgonzalez, ocubo}@fi.upm.es
[2] Institute for Information Technology, National Research Council,
Ottawa Ontario, K1A 0R6, Canada;
fazel.famili@nrc-cnrc.gc.ca

**Abstract.** Biomarkers are biochemical facets that can be used to measure different aspects of a disease. In the last years, there has been much interest in biomarkers of different cancer variants for predicting future patterns of disease. However, DNA Biomarker selection is a difficult task as it involves dealing with a special type of datasets, microarrays, that consists of a large number of features with small number of samples. This paper proposes a new approach for biomarkers selection by means of an innovative parallel evolutionary algorithm that performs wrapper feature selection from thousands of genes to achieve a small set of most relevant ones. To test our method, the well known Van't Veer dataset on Breast Cancer [1] has been considered. Preliminary results outperform those reported by Van't Veer both in accuracy and the number of genes selected.

## 1 Introduction

Biomarkers are biochemical facets or features that can be used to measure different aspects of a disease, like the risk to develop it, its progress or the effects of particular treatments. Disease markers can be studied at many molecular levels, ranged from genomic, epigenomic, proteomics, cellular and morphologic, to genetic factors. These factors predispose patients to the disease or indicate its occurrence. In particular, genetic biomarkers are DNA subsequences that have biological significance, in terms of disease evolution, drug tolerance or response to specific treatments.

There has been much interest in biomarkers of cancer variants in predicting future patterns of disease, especially as cancer treatment has made such positive strides in the last few years. The hope that prognosis and disease treatment could be predicted using these information patterns pushes forward the research in this particular field.

During the last few years, early cancer diagnosis has been based on the concentration of serum antigens, like CEA (Carcinoembryonic antigen), in blood [2]. CEA and other antigens are nonspecific for cancer and can be produced by normal organs as well. Their application is restricted in use and no treatment is ever based solely on a CEA. Usually, alterations above normal can spur further diagnostic testing to catch the disease at an early stage. These serum biomarkers can be partially effective in preliminary diagnosis or, additionally, as a way of determining the adequacy of postoperative therapy [3].

As an alternative to serum antigens, DNA biomarkers could provide predictive capabilities in the evaluation of the evolution of the disease and prognosis [4]. In addition, and even more important, they could lead us to the development of effective treatments using appropriate drugs and therapies.

DNA biomarker selection is difficult to perform. The machine learning analogy for biomarkers selection is feature subset selection (FSS) on microrrays. Microarrays are datasets with the problem of curse of dimensionality (large number of features with small number of samples). FSS approaches are divided into wrapper and filter methods. Wrapper methods provide better results but they have two major issues to be considered: (i) a robust and coherent validation method should be applied to ensure quality and fairness of the internal classifier, and (ii) the size of the search space grows exponentially according to the number of features.

In this paper, a new approach is presented on biomarkers selection. This approach is based on an innovative parallel evolutionary algorithm that performs wrapper feature selection from thousands of genes to achieve a small set of most relevant ones, keeping the best prediction quality. This new technique is a two-stage method (depicted on figure 1). Preliminary feature filtering and data preprocessing is followed by the actual biomarkers selection using Multiple Offspring Sampling (MOS). This method has been tested using the well known Van't Veer dataset on Breast Cancer [1]. The selection obtained includes fewer genes than the ones reported by Van't Veer getting better prediction results.

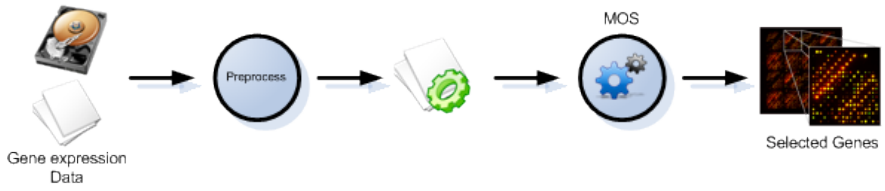


**Fig. 1.** Overview of MOS applied to Biomarker Selection

## 2 Feature Subset Selection (FSS)

The FSS problem [5,6,7] deals with the search of the best subset of variables to train a classifier. This is a very important issue in several areas of knowledge discovery such as machine learning, optimization, pattern recognition and statistics. The goal behind FSS is the appropriate selection of a relevant subset of features upon which to focus the attention of a classification algorithm, while ignoring the rest. The FSS problem is based on the fact that the inclusion of more variables in a training dataset does not necessarily improve the performance of the model. We can distinguish two different kinds of variables:

**Irrelevant features.** This variable has no relation with the target of the classifier.

**Redundant features.** There exist subsets of variables with variables whose information can be deduced from other variables from that subset. The inclusion of all variables within one of those sets will not improve the final model.

### 2.1 Classical Solutions

The literature describes several approaches to solve this problem. To achieve the best possible performance with a particular learning algorithm on a particular training set, a feature subset selection method should consider how the algorithm and the training set interact. There are two alternatives to consider this interaction which have been called *filters* or *wrappers*, respectively.

*Filter methods* [8,9] are mathematical expressions that evaluate each available feature. We can sort all features using this evaluation to obtain a ranking of features and cut this rank when desired. Correlation between each feature and the target variable is a classical example of a filter measure. Filter FSS is based on an estimation of the performance of the algorithm (without its actual execution) based on statistical or information-based relationships among the selected features, including the classification label.

*Wrapper methods* [7,10] use the induction algorithm itself to evaluate the performance of each subset. We train the model with each candidate subset of features and use any resulting quality measure to evaluate each candidate feature selection. In the wrapper approach, FSS becomes an optimization problem for finding the best set of features, using the induction algorithm as a black box.

Filter methods are, in practice, faster than wrapper ones and obtain good enough results in some datasets. Wrapper methods potentially achieve better feature selections but their computational cost is higher. There are two main aspects that deeply influence the computational cost of these techniques: (i) the optimization algorithm could be more or less exhaustive. For example, forward selection, backward elimination, and their stepwise variants can be viewed as simple hill-climbing techniques in the space of feature subsets; (ii) the robustness of the validation method applied to evaluate the quality of the results obtained by each candidate selection. It includes the measure to use, but also the validation schema (leave-one-out, cross-validation, bootstrap, ...). These validation methods behave differently in terms of variance, bias, and complexity.

An accurate FSS technique based on wrapper approaches that combines both a powerful search method and a robust validation approach is still a challenge, particularly in high dimensional datasets. An appropriate alternative is using a *hybrid approach*. The most common one is the use of a filter to reduce the number of features (features are ranked based on their representativness and the worst are removed), and a wrapper to perform the final selection. This represents a balance between the number of features to make the wrapper technique reasonable in computational time and the number of features included in the optimal subset selection.

### 2.2 Heuristic Wrapper Approaches

As it has been said before, wrapper methods use the final model as an internal evaluation step for feature selection. The wrapper trains a model and uses the accuracy of the model

as the fitness value of the subset used for training. These methods need a search schema that guides the generation and selection of subsets of features.

An exhaustive search generates and evaluates all possible subsets of features and so the algorithm always finds the best subset. The main disadvantage of this approach is its complexity. The application of this algorithm is unfeasible even with small-medium size datasets.

Most common approaches are greedy algorithms due to its low computational cost and good results in general. Four greedy approaches can be distinguished [11,12]:

**Sequential Forward Selection.** The search starts with an empty subset and adds the best feature in each step until no improvement can be done.

**Sequential Backward Elimination.** This approach starts with all features selected and deletes the worst one. The deletion of variables stops when no improvement can be done.

**Sequential Floating Forward Selection.** The algorithm starts with an empty subset and the best feature is added in each step (the same as SFS). After adding the variables it tries to delete one of the previously selected ones (a backward step) if this improves the current solution.

**Sequential Floating Backward Elimination.** It starts with all features and deletes the worst. After deleting each variable, the algorithm tries to add one of the previously discarded ones.

Genetic Algorithms [13,14] have been proposed as an alternative to FSS in regular datasets. Although it is a more powerful explorative method, the results with standard datasets are similar to the greedy alternatives. However, these algorithms may behave differently with horizontal datasets (e.g. microarrays).

### 2.3 FSS applied to Microarray Analysis

The analysis of gene expression using microarray data has become popular in the past few years. Microarrays are applied to a wide variety of problems in life and medical sciences. An important issue is patients' diagnosis for some specific disease. Because of the cost and effort required to gather this information, microarray datasets have only a low number of samples or observations (10-100). However, each sample has a large number of numerical expression levels of genes (10000-30000). This extreme asymmetry, referred as the "curse of dimensionality" [15], is the typical property of most microarray datasets, and needs modified computational techniques to be analyzed.

An important task in classification is to reduce the high dimensionality feature space, that is, for example, applying dimensionality reduction or feature subset selection techniques.

Feature selection applied to microarray data has primarily been studied in a supervised learning context, where predictive accuracy is commonly used to evaluate feature subsets. Specifically, (penalized or non-penalized) logistic regression algorithms were used by [16,17]. Even new algorithms based on logistic regression (Recursive Feature Elimination) were proposed [18] to obtain the best genes selection. Other supervised methods have also been considered [19,1] for cancer diseases.

Considering both wrapper and filter feature selection, Inza and Larrañaga present a comparison between both models in DNA microarray domains [20]. Different methods using both models have been proposed [21,22,1] trying to exploit benefits from both approaches with significative results.

## 3 Breast cancer dataset description

Van't Veer dataset [1] on Breast Cancer [3] has been considered to validate our approach. As we know, Van't Veer researches were approved by FDA (Food and Drug Administration) and were applied in a genetic test, named MammaPrint, that predicts whether patients will suffer breast cancer relapse or not.

Data is divided into two groups, learning and validation instances. The training data consists of 78 patients, 34 of which are patients that developed distance metastases within 5 years (poor prognosis). The rest of the dataset (44 patients) are the ones who remained healthy from the disease after their initial diagnosis for an interval of 5 years (good prognosis). The second group of patients (validation dataset) consists of 19 patients, 12 patients with poor prognosis and 7 with good prognosis.

DNA microarray analysis was used to determine the mRNA expression levels of approximately 24500 genes for each patient. All the tumours were hybridized against a reference pool made by pooling equal amounts of RNA from each patient.

### 3.1 Preprocessing

Obviously, the original data contains many redundancies and also incorrect or missing values, depending on some factors. So, as a first step, certain preprocessing was performed in order to clean up and prepare the data. Variables with low internal variance or low correlation with outcome were also discarded.

Several preprocessing algorithms have been carried out through the training data. Firstly, replicated genes are discarded. Next, patients with more than 80% of missing gene values are also discarded. All data have been background corrected, normalized and log-transformed using Lowess Normalization [23]. Missing values were estimated using a 15-weighted nearest neighbours algorithm [24] (kNN Impute).

### 3.2 Preliminary Filtering

Filter scoring tries to identify genes that are differentially expressed in the categories of the problem. The first step of the filter procedure is to rank the features in terms of the values of the used univariate scoring metric. In a second step, the $d$ features with the highest scoring metric are chosen to induce the LR model. For this contribution, Pearson measure has been selected.

$$r(j) = \frac{\sum_{i=1}^{N} (x_{ij} - \overline{x}_j) \cdot (y_i - \overline{y})}{(n-1) \cdot s_j \cdot s_y} \tag{1}$$

[3] available at `http://www.rii.com/publications/2002/vantveer.html`

where $\overline{x}_i$ is the mean value, $s_j$ is the standard deviation of expression levels, and $y_i$ and $\overline{y}$ are the class value and class mean respectively.

Next, a ranking list ordered by Pearson correlation is generated. With this list, a group of 1000 best genes has been selected. A large number of pre-candidate genes has been selected to provide enough alternatives to the wrapper search in the second stage. As mentioned before, this means that the search space in the wrapper method is large and potentially very complex. The proposed wrapper method to select biomarkers from a 1000 candidate genes is based on an innovative evolutionary technique that allows optimal values to be found on complex and large search spaces.

## 4 MOS: Multiple Offspring Sampling

Multiple Offspring Sampling is introduced as a variant of classic population-based evolutionary algorithms. This new approach proposes the simultaneous use of different *techniques* (a proper definition of technique in the context of *MOS* will be given in subsection 4.2) to create new individuals (candidate solutions).

To show how *MOS* modifies the behaviour of classic Evolutionary Algorithms (EA), we should first present a general schema of EA functioning, which will be given in the next subsection. Afterwards, Multiple Offspring Sampling will be presented.

### 4.1 Evolutionary Algorithms

Evolutionary algorithms (like Genetic Algorithms (GAs)), in a general schema, are divided into different phases:

① Creation of the initial population $P_0$.
② Evaluation of the initial population $P_0$.
③ Checking of the algorithm termination (convergence or generation limit), if so then finish, otherwise continue.
④ Generation, using some individuals from $P_i$, of new individuals for the next generation, called offspring population $O_i$.
⑤ Evaluation of the new individuals in $O_i$.
⑥ Combination of offspring and previous population to define the next population $P_{i+1}$.
⑦ Go back to ③.

Based on this schema, different evolutionary algorithms and approaches have been developed. For example, in step ⑥ classical GAs take the offspring as the next population ($P_{i+1} = O_i$). Other approaches, like steady state algorithms generate only one offspring individual that replaces the worst individual in $P_i$, and intermediate approaches, based on elitism, take the best individuals from both $O_i$ and $P_i$ to generate $P_{i+1}$.

In step ④, there have been also many different approaches in the literature. Some examples are based on selecting different genetic operators, or using statistical approaches for modelling the population and later sampling the offspring (e.g. estimation of distribution algorithms by [25]).

### 4.2 Multiple Offspring Basics

We introduce Multiple Offspring Sampling (MOS) approach as a combined alternative in the way steps ④ and ⑥ are performed. MOS proposes the definition of multiple mechanisms to generate new individuals, and make them compete during the evolution process. Each mechanism creates its own offspring $O_i^{(j)}$ ($i$ is the generation and $j$ is the mechanism).

These MOS mechanisms, or techniques, as they are named at the beginning of section 4, could be defined as a mechanism to create new individuals, i.e., (a) a particular evolutionary algorithm model, (b) with an appropriate coding, (c) using specific operators (if required) and (d) configured with its necessary parameters.

According to the above definition we can consider different parameters and thus divide MOS into several categories. A rough taxonomy of how MOS can be divided could be:

- Algorithm-based MOS: different algorithms (GAs, EDAs) are used to create new individuals.
- Coding-based MOS: different codings (genotypes) can be used to represent one candidate solution (phenotype) of the problem.
- Operator-based MOS: for a single coding of candidate solutions there could exist different genetic operators (if working with GAs) that could be used simultaneously.
- Parameter-based MOS: different values for evolutionary parameters (crossover and mutation ratios, selection mechanisms, etc.) are used within each technique.
- Hybrid MOS: a combination of any of the previous.

In the particular case of the experimentation performed for this study, two different genotype encodings are considered.

As a solution, the phenotype, can participate in multiple genotype recombinations, a group of functions is required to transform genotypes between two different encodings.

Once the offspring population is created by each of the techniques being used, the quality of these populations is evaluated by means of several possible measures. The most obvious of these measures is the average fitness of the population, but more sophisticated measures could be proposed to take into account not only the current performance of the technique but its capability.

Finally, in phase ⑥, previous population $P_i$ and all the offsprings $O_i^{(j)}$ are merged to produce the next population $P_{i+1}$. This process is usually done by using an elitist population merge function.

The calculation of the amount of new individuals created in each generation, for $n$ different offspring sampling methods, is obtained using a Participation Function (***PF***). Different functions have been proposed in other scenarios by [26], where the first approach to Algorithm-MOS was introduced under the combination of two different Evolutionary Algorithms: GAs and EDAs. From these functions, a dynamic one was selected for being used in our studies. This function dynamically adjusts the participation of each technique according to the quality of the offspring populations calculated before.

### 4.3 MOS for Biomarker Selection

Previous subsections have introduced MOS as an innovative parallel genetic algorithm that is able to exploit the benefits of using different techniques to produce a new offspring based on current population. In the case of this study, two different codings were used.

First coding is simply a binary vector of length the number of features in the learning dataset. Each of these binary values tells if that feature will or will not be selected by the algorithm.

Second coding is a condensed version of the first one, consisting of a vector of integer numbers where each number represents a gene being selected by the algorithm. This messy coding was firstly introduced by [27] and since then reliably applied to a wide range of optimization problems [28,29].

These two codings coexist all along the evolutionary process, each of them taking more participation in different phases of the execution of the genetic algorithm and helping the GA to outperform itself when using just a single genetic representation (coding).

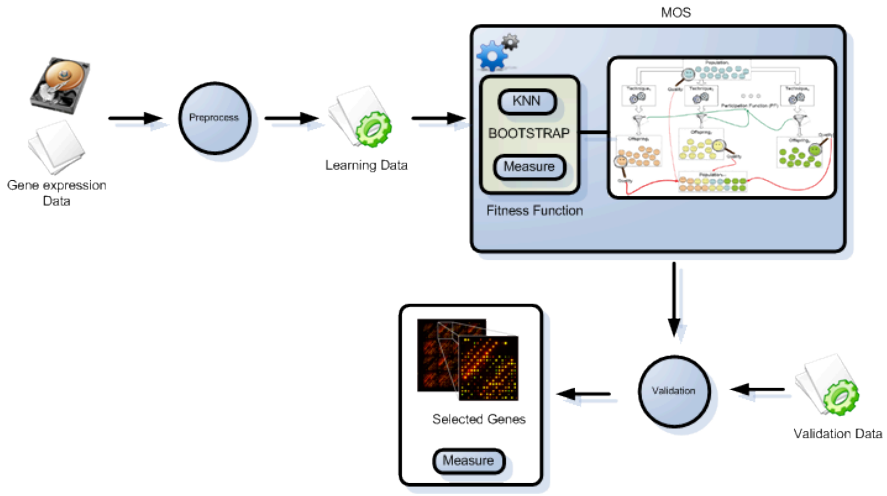


**Fig. 2.** A detailed view of MOS applied to Biomarker Selection

## 5 Experimentation scenario, results and discussion

This section provides an overview of the whole process followed in this research along with the results obtained and a discussion about these results.

Figure 4.3 clearly depicts the followed process:

① Firstly, a preprocessing phase is performed to select the best 1000 genes considering their position in a ranked list ordered by Pearson correlation, as explained in section 3.2.
② Then, a MOS algorithm is executed to select most relevant genes from these 1000 genes previously selected. MOS will evaluate each generated individual with a bootstrap (200 iterations) using a KNN algorithm, trying to optimize a fitness "measure", the AUC in this experimentation, as it has empirically demonstrated to behave quite well for this problem. For KNN, two different distance measures have been considered: traditional Euclidean distance and Chebyshev distance. First one has been selected because it has been widely used in previous works and that lets us to fairly compare our approach with others. Second one has been used due to its capability to penalize selections of genes with large distance among only few of them (even between just two of them), a characteristic we wanted to exploit in our experiments.
③ Finally, an external validation process is performed considering only selecting genes to learn a KNN algorithm and a validation dataset different from that used to learn and not seen by the algorithm until now.

The experiments were executed on a 13 dual Xeon cluster at 2.40 GHz, using a parallel asynchronous genetic algorithm implemented in GAEDALib coded by [30] with the configuration described in table 1(b).

**Table 1.** Experimental scenario

(a) GA configuration

| | |
|---|---|
| **(Global) Pop. size** | 390 |
| **Termination** | Pop. convergence |
| **Convergence %** | 98 % |
| **Individuals selection** | Roulette wheel |
| **Crossover %** | 90 % |
| **Mutation %** | 1 % |

(b) Parallel configuration

| | |
|---|---|
| **Paradigm** | islands model |
| **Model** | asynchronous |
| **Topology** | mesh |
| **Migration rate** | 10 gens. |
| **Migration pop.** | Top 20 % |
| **Nodes** | 26 |

Table 2 summarizes the results obtained in this experimentation. Fourteen different configurations were tested. For each of the two distance measures considered, seven different fitness functions were tested. First function only tried to maximize the AUC, regardless of the number of selected genes. With such a great degree of freedom the algorithm tends to select a huge number of genes. For this reason, a new fitness function was introduced (see equation 2) that tries to avoid this problem. This fitness function tries to lower the number of variables as much as possible but not more than the pivot value that acts as a center of gravity for the number of variables. Then, six new configurations were executed, with the only difference being in the pivot value.

$$\textit{fitness} = AUC * \frac{1}{abs(\#genes - pivot) + 1} \tag{2}$$

Results, in general, outperform those of Van't Veer both in prediction accuracy and smaller number of genes selected. Best results are achieved with Chebyshev distance and the penalized fitness function with pivot equal to 40, although there are not great differences among all the configurations with penalized fitness function regardless of the distance measure used. This makes us think that the optimal number of genes must be within this range ([20, 60]).

Table 2 presents the average results of ten executions for each configuration. Several executions of the algorithm (with different configurations) returned a selection of genes with an impressive 94% of accuracy in external validation and with just 20 genes selected in the best case.

From table 2 we can also observe that there exists a strong correlation between the optimization measure (AUC) and the validation measure (accuracy) (0.92 for Pearson correlation). This property is quite desirable for an optimization measure when training an algorithm that will be validated with unknown data.

Finally, the penalizing method appears to be very restrictive and makes the algorithm to adjust perfectly to the selected value of pivot. This behaviour must be studied and some modifications may be introduced to allow a certain level of flexibility for the number of features selected.

**Table 2.** Summary of results: all reported values are the average of ten executions

| | **AUC** | **Accuracy** | **Size** |
|---|---|---|---|
| Chebyshev Distance - Not Penalized | 0.75 | 0.79 | 317.70 |
| Chebyshev Distance - Penalized (centered on 0) | 0.60 | 0.71 | 2.45 |
| Chebyshev Distance - Penalized (centered on 20) | 0.76 | 0.81 | 20.00 |
| Chebyshev Distance - Penalized (centered on 30) | 0.73 | 0.79 | 30.00 |
| Chebyshev Distance - Penalized (centered on 40) | 0.76 | **0.84** | 40.00 |
| Chebyshev Distance - Penalized (centered on 50) | 0.75 | 0.81 | 50.00 |
| Chebyshev Distance - Penalized (centered on 60) | 0.75 | 0.81 | 60.00 |
| Euclidean Distance - Not Penalized | 0.74 | 0.77 | 131.25 |
| Euclidean Distance - Penalized (centered on 0) | 0.66 | 0.73 | 2.35 |
| Euclidean Distance - Penalized (centered on 20) | 0.76 | 0.80 | 20.00 |
| Euclidean Distance - Penalized (centered on 30) | 0.75 | 0.82 | 30.00 |
| Euclidean Distance - Penalized (centered on 40) | 0.75 | 0.81 | 40.00 |
| Euclidean Distance - Penalized (centered on 50) | 0.80 | 0.82 | 50.00 |
| Euclidean Distance - Penalized (centered on 60) | 0.76 | 0.81 | 60.00 |

## 6 Conclusions and future work

This paper introduces an innovative and robust method to perform FSS on large microarray data sets (1000 features or more).

It also presents a validation mechanism that consists of: (i) an internal validation process to avoid overfitting to learning data (bootstrap with 200 iterations in this ex-

perimentation) and (ii) an external validation to evaluate the quality of the selection of genes.

Results demonstrate the effectiveness of this method, with an average accuracy of 84% in the best configuration, and several selections of genes with an accuracy of 94%. The number of genes selected is also fewer than those reported by Van't Veer, which makes this approach outperform previous works both in accuracy and selections of genes' size.

Future works will include analysis of the relations among different selections of genes with similar performance and a study of the behaviour of the algorithm when learning with measures others than AUC.

## Acknowledgements

This research project is funded by the Spanish Ministry of Science TIN2007-67148.

## References

1. van 't Veer, L.J., Dai, H., van de Vijver, M.J., He, Y.D., Hart, A.A., Mao, M., Peterse, H.L., van der Koõy, K., Marton, M.J., Witteveen, A.T., Schreiber, G.J., Kerkhoven, R.M., Roberts, C., Linsley, P.S., Bernards, R., Frieñd, S.H.: Gene expression profiling predicts clinical outcome of breast cancer. Nature **415**(6871) (2002) 530–536
2. Wang, D., Knyba, R., Bulbrook, R., Millis, R., Hayward, J.: Serum carcinoembryonic antigen in the diagnosis and prognosis of women with breast cancer. Eur J Cancer Clin Oncology **1**(20) (1984) 25–56
3. et al., J.M.T.: Serum markers and prognosis in locally advanced breast cancer. Tumori **6**(91) (2005) 522–552
4. Hartwell, L., Mankoff, D., Paulovich, A., Ramsey, S., Swisher, E.: Cancer biomarkers: a systems approach. Nature Biotechnology **8**(24) (2006) 905–913
5. Almuallim, H., Dietterich, T.: Learning with many irrelevant features. In: Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91). Volume 2., Anaheim, California, AAAI Press (1991) 547–552
6. Blum, A., Langley, P.: Selection of relevant features and examples in machine learning. Artificial Intelligence **97**(1–2) (1997) 245–271
7. John, G., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: International Conference on Machine Learning. (1994) 121–129 Journal version in AIJ.
8. Ben-Bassat, M.: Use of distance measure, information measures, and error bounds on feature evaluation. In Krishnaiah, P.R., Kanal, L.N., eds.: Classification, Pattern Recognition and Reduction of Dimensionality. North–Holland Publishing Company, Amsterdam (1987) 773–791
9. Jeffery, I.B., Higgins, D.G., Culhane, A.C.: Comparison and evaluation of methods for generating differentially expressed gene lists from microarray data. BMC Bioinformatics **7** (2006) 359+
10. Kohavi, R., John, G.: Wrappers for feature subset selection. Artificial Intelligence **97**(1–2) (1997) 273–324
11. Kittler, J.: Feature set search algorithms. Pattern Recognition and Signal Processing (1978) 41–60
12. Somol, P., Pudil, P., Novovicová, J., Paclík, P.: Adaptive floating search methods in feature selection. Pattern Recognition Letters **20**(11–13) (1999) 1157–1163

13. Inza, I., Larrañaga, P., Sierra, B.: Feature subset selection by bayesian networks: a comparison with genetic and sequential algorithms. International Journal of Approximate Reasoning **27**(2) (2001) 143–164
14. Jain, A., Zongker, D.: Feature selection: Evaluation, application, and small sample performance. IEEE Trans. Pattern Anal. Mach. Intell. **19**(2) (1997) 153–158
15. usa H. Asyali, Dilek Colak, O.D., Inan, M.S.: Gene expression profile classification: A review. Current Bioinformatics **1**(1) (2006) 55–73
16. Shevade, S.K., Keerthi, S.S.: A simple and efficient algorithm for gene selection using sparse logistic regression. Bioinformatics **19**(17) (2003) 2246–2253
17. Weber, G., Vinterbo, S.A., Ohno-Machado, L.: Multivariate selection of genetic markers in diagnostic classification. Artificial Intelligence in Medicine **31**(2) (2004) 155–167
18. Shen, L., Tan, E.C.: Dimension reduction-based penalized logistic regression for cancer classification using microarray data. IEEE/ACM Trans. Comput. Biol. Bioinformatics **2**(2) (2005) 166–175
19. Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., Lander, E.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science **286** (1999) 531–537
20. Inza, I., Larrañaga, P., Blanco, R., Cerrolaza, A.J.: Filter versus wrapper gene selection approaches in dna microarray domains. Artificial Intelligence in Medicine **31**(2) (2004) 91–103
21. Mamitsuka, H.: Selecting features in microarray classification using roc curves. Pattern Recogn. **39**(12) (2006) 2393–2404
22. Statnikov, A., Tsamardinos, I., Dosbayev, Y., Aliferis, C.F.: Gems: a system for automated cancer diagnosis and biomarker discovery from microarray gene expression data. Int J Med Inform **74**(7-8) (2005) 491–503
23. Quackenbush, J.: (Microarray data normalization and transformation - nature genetics)
24. Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., Altman, R.B..: Missing value estimation methods for dna microarrays. Bioinformatics **17**(6) (2001) 520–525
25. Larrañaga, P., Lozano, J.: Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation. Kluwer Academic Publisher (2002)
26. Robles, V., Peña, J., Larrañaga, P., Pérez, M., Herves, V.: GA-EDA: A New Hybrid Cooperative Search Evolutionary Algorithm. In: Towards a New Evolutionary Computation. Advances in Estimation of Distribution Algorithms. Volume 192 of Studies in Fuzziness and Soft Computing. Springer (2006) 187–220
27. Goldberg, D.E., Deb, K., Kargupta, H., Harik, G.: Rapid accurate optimization of difficult problems using fast messy genetic algorithms. In Forrest, S., ed.: Proceedings of the Fifth International Conference on Genetic Algorithms, San Mateo, CA (1993) 56–64
28. Watson, R.A., Hornby, G.S., Pollack, J.B.: When food is better than sex: Messy variations on the ga. In: Proceedings of the 5th International Conference of the Society for Adaptive Behavior (SAB'98), University of Zurich, Switzerland (1998)
29. Fenton, P., Walsh, P.: A comparison of messy ga and permutation based ga for job shop scheduling. In Beyer, H., O'Reilly, U., eds.: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2005, Washington DC, USA, ACM Press (2005) 1593–1594
30. Díaz, P.: Diseño e implementación de una librería de algoritmos evolutivos paralelos. Master's thesis, Facultad de Informática, Universidad Politécnica de Madrid (2005)