Biomarker Discovery Based on Large-Scale Feature Selection and MapReduce

Ahlam Kourid(✉) and Mohamed Batouche

Computer Science Department, College of NTIC, Constantine 2 University – A. Mehri, 25000, Constantine, Algeria
ahlem.kou@gmail.com, mohamed.batouche@univ-constantine2.dz

Abstract. Large-scale feature selection is one of the most important fields in the big data domain that can solve real data problems, such as bioinformatics, where it is necessary to process huge amount of data. The efficiency of existing feature selection algorithms significantly downgrades, if not totally inapplicable, when data size exceeds hundreds of gigabytes, because most feature selection algorithms are designed for centralized computing architecture. For that, distributed computing techniques, such as MapReduce can be applied to handle very large data. Our approach is to scale the existing method for feature selection, Kmeans clustering and Signal to Noise Ratio (SNR) combined with optimization technique as Binary Particle Swarm Optimization (BPSO). The proposed method is divided into two stages. In the first stage, we have used parallel Kmeans on MapReduce for clustering features, and then we have applied iterative MapReduce that implement parallel SNR ranking for each cluster. After, we have selected the top ranked feature from each cluster. The top scored features from each cluster are gathered and a new feature subset is generated. In the second stage, the new feature subset is used as input to the proposed BPSO based on MapReduce which provides an optimized feature subset. The proposed method is implemented in a distributed environment, and its efficiency is illustrated through analyzing practical problems such as biomarker discovery**.**

Keywords: Feature selection · Large-scale machine learning · Big data analytics · Bioinformatics · Biomarker discovery

1 Introduction

With the progress of high technology in several fields that produce an important volume of data such as Microarray and Next generation sequencing in bioinformatics [1], deal with high dimensional data becomes a challenge for several tasks in machine learning. Feature selection is one of the techniques of reduction dimensionality [2] that is effective in removing irrelevant data; increasing learning accuracy, therefore becomes very necessary for machine learning tasks. Scalability can become a problem for even simple and centralized approaches, for that feature selection methods based on parallel algorithm will be the mainly choice for dealing with large-scale data. Many parallel algorithms are implemented using different parallelization techniques

A. Amine et al. (Eds.): CIIA 2015, IFIP AICT 456, pp. 81–92, 2015.
DOI: 10.1007/978-3-319-19578-0_7

such as MPI (The Message Passing Interface), and MapReduce. MapReduce is a programming model for distributed computation, derived from the functional programming concepts, and is proposed by Google for large-scale data processing in a distributed computing environment [3].

Recent comparisons studies of feature selection methods in high-dimensional data have shown that the combination of K-means clustering and filter method based SNR (Signal to Noise Ratio) score combined with binary PSO is a graceful method for classification problem [4]. The method is applied for classification of DNA microarray data. To resolve redundancy in gene expression values one approach i.e. sample based clustering by using k-means clustering algorithm is used and the genes (features) are being grouped into number of clusters. After clustering SNR ranking is being used to rank each gene (feature) in every cluster. The gene subset selected by taking the top scored gene (feature) from each cluster is validated with an SVM classifier, and will be taken as the initial search space to find the optimized subset by applying PSO and the optimized subset is used to train different classifier such as SVM [4]. However, the existing method is limited over large scale datasets. In order to overcome that problem we present our method that is suitable for very large data and that has the potential for parallel implementation, based on parallel Kmeans on MapReduce for clustering a huge amount of features, so similar features having the same characteristics will be grouped in the same cluster, and on an iterative MapReduce that implement parallel SNR ranking for each cluster. Finally, the top non-redundant ranked features selected are input to BPSO on MapReduce to select the relevant features.

2 Parallel Programming Paradigm and Framework

In order to implement our approach to cope with large scale data sets, we are using Hadoop platform and MapReduce as parallel programming paradigm .

2.1 MAPREDUCE

MapReduce is a functional programming model that is well suited to parallel computation. The model is divided into two functions which are map and reduce .In MapReduce; all data are in the form of keys with associated values. The following notation and example are based on the original presentation [3]:

A. **Map Function**

A map function is defined as a function that takes a single key-value pair and outputs a list of new key-value pairs. The input key may be of a different type than the output keys, and the input value may be of a different type than the output values:

$$\text{Map} :(\text{K1}, \text{V1}) \rightarrow \text{list} ((\text{K2}, \text{V2})) \quad (1)$$

B. **Reduce Function**

A reduce function is a function that reads a key and a corresponding list of values and outputs a new list of values for that key. The input and output values are of the same type.

$$\text{Reduce :(K2, list (V2))} \rightarrow \text{list (V2)} \tag{2}$$

2.2 HADOOP Platform

Hadoop is an open source Java based framework to store and process large amounts of data. It allows distributed processing of data which is present over clusters using functional programming model. MapReduce is the most important algorithm implemented in Hadoop. Each Map and Reduce is independent of other Maps and Reduces. Processing of data is executed in parallel to other processes. A job scheduler or job tracker tracks MapReduce jobs which are being executed. Tasks like Map, Reduce and Shuffle are accepted from Job Tracker by a node called Task Tracker. Hadoop architecture is defined as follows: Hadoop consists of two components, the Hadoop Distributed File System (HDFS) and MapReduce, performing distributed processing by single-master and multiple-slave servers. There are two elements of MapReduce, namely JobTracker and TaskTracker, and two elements of HDFS, namely DataNode and NameNode. [5].

3 Scaling Up Feature Selection Algorithm

For scaling up the existing method for feature selection, we propose an approach based on MapReduce which is composed of two stages. The first stage consists in filtering the set of features by selecting the top scored features whereas the second stage optimizes the obtained subset of selected features.

3.1 Filtering the Set of Features

This stage is scalable and implements K-means clustering on MapReduce and SNR ranking on MapReduce for each cluster. It is designed for the purpose of eliminating redundancy in features and selecting the top scored features [6]. And it is composed of the following steps:

Step1: clustering features (genes) with parallel K-means on MapReduce. As by applying clustering technique we can group similar type of features in the same cluster, so that best features from each cluster can be selected.
Step2: mappers read lines (features) and compute SNR score for each feature.
Step3: according to the paradigm shuffle and sort in MapReduce, the final output file contains ranked SNR values. Top ranked features are selected in two cases:

- One output file: the top ranked features are selected from this file.
- Multiple output files: each file is ranked by SNR value, for that Terasort can be used to rank all SNR values from these files. Terasort is a standard map/reduce sort, and it is implemented as benchmark in hadoop [7].

Step4: After that the best scored feature in a cluster is selected, and go to step 2 for the next cluster. We can assure that applying SNR and selecting the best scored feature from each cluster the resultant feature gene subset have no redundancy.
Step5: top features (genes) ranked from each cluster are aggregated and validated with SVM classifier using the evaluation method 10 foldCV.

The system architecture for the proposed method in stage-I- is illustrated in **Fig.1**.

3.2 Optimizing the Subset of Selected Features

This stage aims to select an optimized subset of features from the subset selected in the previous stage. It is parallel, and can provide scalability to a certain degree because of the SVM classifier which is sequential. In this stage; we have used four MapReduce jobs described by the following steps:

Step1: the subset of features selected and validated in the previous stage, is the input to the novel BPSO proposed based on MapReduce, we have divided the particles of swarm into groups, so that the input file contains particles defined by their groups, in the first MapReduce job, mappers evaluate fitness (accuracy of SVM) of particles in parallel.
Step2: in the second MapReduce job, mappers read output file from the first job and emit the group identifier as key in order to group particles. Reducers evaluate Gbestg of each group in parallel, and emit "one" as key and the Gbestg of the group with fitness of Gbestg of the group as value.
Step3: the third MapReduce job evaluates the Gbestglobal, which is the maximum of all Gbestg of each Group. The output file of this job contains the Gbestglobal and its fitness.
Step4: the file output of the first job in HDFS is the input of the fourth job, in this job mappers read the output file of the third job that contains Gbestglobal and its fitness from HDFS, in order to evaluate the new positions and the new velocities in parallel. The output of this job is the new swarm for the next iteration.

The system architecture for the proposed method in stage-II- is illustrated in **Fig.2**.

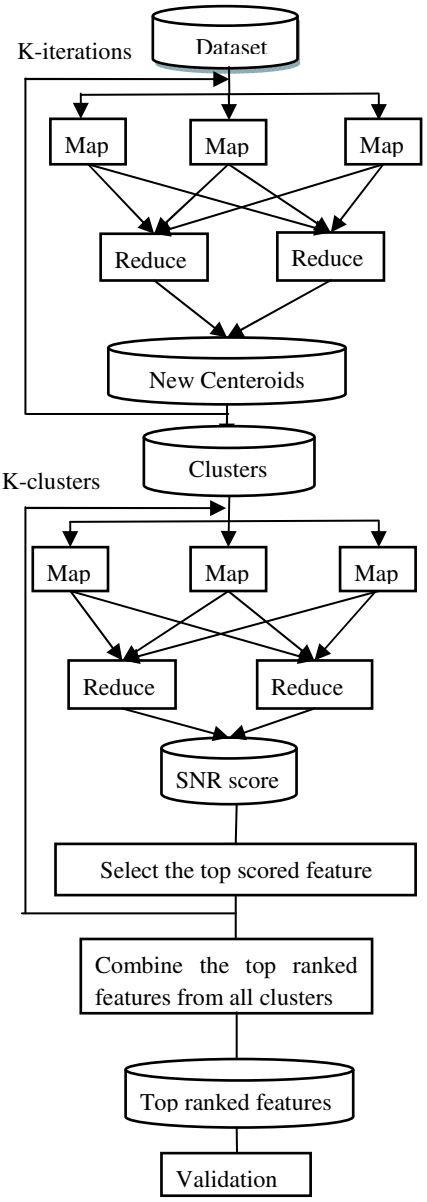

Fig. 1. System architecture of the proposed method stage-I-

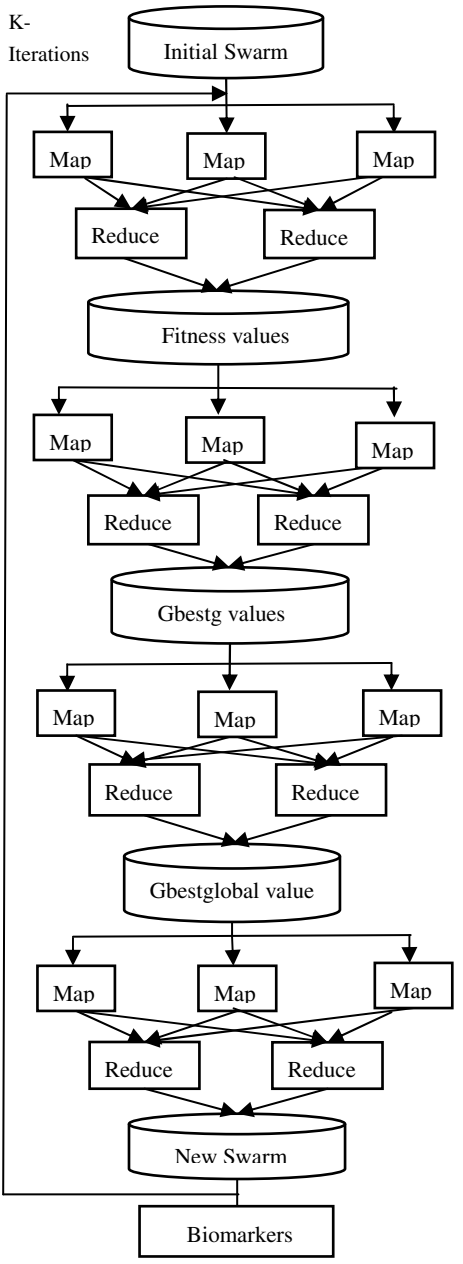

Fig. 2. System architecture of the proposed method stage-II-

4 Implementation of the Proposed Approach

In order to implement the proposed approach for scaling up the existing method for feature selection, we describe in the following the algorithms and map/reduce functions.

4.1 The First Stage

In this stage, we are using K-means along with SNR ranking on MapReduce which are defined as follows:

Algorithm 1. Kmeans on MapReduce.

Input : Training data (features) .
Output: Clusters.

Algorithm 1.1. k-means::Map

Input: Training data x ∈ D, number of clusters k, distance measure d
1: If first Map iteration **then**
2: Initialize the k cluster centroids C randomly
3: Else
4: Get the k cluster centroids C from the previous Reduce step.
5: Set $S_j = 0$ and $n_j = 0$ for j = {1, ... , k}
6: For each $\mathbf{x_i} \in$ **D do**
7: y_i = arg $\min_j d(\mathbf{x_i}, \mathbf{c_j})$
8: $S_{y_i} = S_{y_i} + x_i$
9: $n_{y_i} = n_{y_i} + 1$
10:For each j ∈ {1, … , k} **do**
11: Output(j, <S_j, n_j>)

Algorithm 1.2. k-means::Reduce

Input : List of centroid statistics – partial sums and counts [<S_j^l , n_j^l>] – for each centroid j ∈ {1, ..., k}
1:For each j ∈ {1, ..., k} **do**
2: Let λ be the length of the list of centroid statistics
3 : $n_j = 0$, $S_j = 0$
4 : **For each** l ∈ {1, ..., k} **do**
5 : $n_j = n_j + n_j^l$

6 : $S_j = S_j + S_j^l$

7 : $\mathbf{c_j} = \frac{S_j}{n_j}$

8 : Output (j, $\mathbf{c_j}$)

The whole clustering is run by a Master, which is responsible for running the Map (cluster assignment) and Reduce (centroid re-estimation) steps iteratively until k-means converges [8].

Algorithm 2. SNR on MapReduce

Input : Clusters .
Output: Feature subset of top scored features from clusters.

- List: contains target classes of samples in order.
- Record: contains values of samples for $feature_i$.
- DFS: is a distributed directory system for storage of output and input files of MapReduce.
- ID_feature: is an identifier characterizes each feature.
- $file\_cluster_i$:contains features of cluster i.

Clustering features withAlgorithm 1.
For each cluster i do
DFS.put ($file\_cluster_i$)

Map function (parallel over features) (key: ID_feature, value: record)
List= [class1, class2, class2……]
Iterate over record and list
compute μ1, μ2
compute σ1, σ2
compute SNR
Output (SNR, (ID_feature, record))

Reduce Function (key: SNR, value :(ID_feature, record)
Output (SNR, (ID_feature, record))
Select top scored feature.
DFS.delete ($file\_cluster_i$).

Aggregation and validation of the top scored features selected.

4.2 The Second Stage:

In this stage, we are using Binary PSO on MapReduce which is composed of four MapReduce jobs. In Hadoop, a mechanism of JobControl classes is provided to execute the four jobs sequentially.

Algorithm 3. PSO on MapReduce

Input : Initial swarm of particles and the subset of top features selected and validated .
Output: Best solution Gbest.

- GroupP: we have defined at the beginning several groups of particles, GroupP is the identifier of each group.
- P: position of a particle, Pbest: best position of a particle, FitPbest: fitness of Pbest, Gbest: global position of particles, FitGbest: fitness of Gbest, V: velocity of a particle.

First job
Mapper (key: ID_particle, value: (P, Pbest, FitPbest, Gbest, FitGbest, V, GroupP) (parallel mappers)
Initial Pbest, FitPbest, Gbest, FitGbest are empty.
fitness (): function of evaluation of the fitness of the designed particle (accuracy SVM) and take as input P and features selected.
If fitness (P) >FitPbest
Pbest=P.
FitPbest= fitness (P).
Emit(ID_particle, (P, Pbest, FitPbest, Gbest, FitGbest, V, GroupP))
Reducer (key: ID_particle, value: (P, Pbest, FitPbest, Gbest, FitGbest, V,GroupP) (parallel reducers)
Emit(ID_particle, (P, Pbest, FitPbest, Gbest, FitGbest, V,GroupP)) (file-output1 in HDFS)

Second job
Mapper (key: ID_particle, value: (P, Pbest, FitPbest, Gbest, FitGbest, V, GroupP) (parallel mappers)
Emit(GroupP, (ID_particle ,P, Pbest, FitPbest, Gbest, FitGbest, V))
Reducer (key: GroupP, value: (ID_particle, P, Pbest, Gbest, FitGbest, V) (parallel reducers)
Initial Gbestg is empty.
Cpt: number of 1 in Gbest, initialized to 0.
For all values
Gbestg = maximum of all Gbest with minimum number of Cpt (in case of equality between Gbest).
FitGbestg= FitGbest of Gbestg.
Emit (ONE, (Gbestg, FitGbestg)) (file-output2 in HDFS)

Third job
Mapper (key: ONE, value: (Gbestg, FitGbestg) (parallel reducers)
Emit (ONE, (Gbestg, FitGbestg))
Reducer (key: ONE, value: (Gbestg, FitGbest) (parallel reducers)
Initial Gbestglobal is empty. Cpt1: number of 1 in Gbestg, initialized to 0.
For all values
Gbestglobal = maximum of Gbestg with minimum number of Cpt1 (in case o equality between Gbestg)
FitGbestglobal= FitGbest of Gbestglobal.
Emit (Gbestglobal, (FitGbestglobal)) (file-output3 in HDFS)

Fourth job
Mapper (key: ID_particle, value: (P, Pbest, FitPbest, Gbest, FitGbest, V, GroupP) (parallel mappers)
Read file-output3 from HDFS
Gbest = Gbestglobal
FitGbest = FitGbestglobal
V'= New_Velocity (V, P, Pbest, Gbest) /* New_Velocity is a function for the evaluation of the new velocity*/
P'= New_Position (P, V') /* New_Position is a function for the evaluation of the new position*/
P=P', V=V'
Emit(ID_particle, (P, Pbest, FitPbest, Gbest, FitGbest V, GroupP))
reducer (key: ID_particle, value: (P, Pbest, FitPbest, Gbest, FitGbest, V, GroupP) (parallel reducers)
Emit(ID_particle, (P, Pbest, FitPbest, Gbest, FitGbest, V, GroupP))

Repeat the execution of jobs K-iterations.

5 Results and Experiments

We have used tow datasets of cancer RNA-seq gene expression data (gastric cancer, ESCA (esophageal carcinoma)): gastric dataset derived from the main source of gene expression data Omnibus. The last, ESCA derived from TCGA (Cancer Genome Atlas), and four gene expression microarray datasets (tow ovarian cancer datasets, gastric cancer dataset, ESCC dataset (esophageal squamous cell carcinoma)) derived from Omnibus. Our approach is implemented on two-node cluster (master and slave), both master machine and slave machine are equipped with dual core processor and 4GB RAM memory for master node, and 2 GB for slave node. The operating system installed on the two nodes is Linux Ubuntu 13.10. The experiment is done using hadoop-1.2.1 and mahout 0.9 [9]. The cluster is configured in fully-distributed mode [10]. We have used support vector machine (SVM) to obtain classification accuracy, and the cross validation method 10 foldCV for performance evaluation of the classifier SVM. In order to improve the scalability of our method we have used a synthetic dataset (duplicate genes of each dataset), the size of data increased reaches 5GB for each dataset. Experiment is done with 5 clusters and 10 clusters. The performance of

our method is compared to other approaches in the literature: an approach Based on Neighborhood Rough Set and Probabilistic Neural Networks Ensemble is proposed for the classification of Gene Expression Profiles [11], in [12] authors proposed a new selection method of interdependent genes via dynamic relevance analysis for cancer diagnosis. However, in the work presented in [13] a sequential forward feature selection algorithm to design decision tree models is suggested for the identification of biomarkers for Esophageal Squamous Cell Carcinoma. The obtained results are shown on Table 1, Table 2 and Table 3.

Table 1. Accuracy of SVM and number of genes selected in our method with normal datasets and comparison with other approaches

dataset	Ng	BPSO on MapReduce				[11]		[12]		[13]	
		Se	**Sp**	**Acc**	**#**	**Acc**	**#**	**Acc**	**#**	**Acc**	**#**
Ovrian [11]	15154	1	0,98	99	3	96	9	-	-	-	-
Gastric [12]	4522	1	1	100	2	-	-	96	14	-	-
ESCC [13]	22477	0,96	0,96	96	2	-	-	-	-	97	2
Ovrian	54675	1	1	100	2	/		/		/	
Gastric	21475	1	1	100	1						
ESCA	26540	1	1	100	2						

Ng: number of genes, **Se:** sensibility, **Sp:** specificity, **Acc:** accuracy (%), **#:** number of genes selected.

Table 2. Accuracy of SVM and number of genes selected in our method with large-scale datasets and comparison with other approaches

dataset	Size dataset	BPSO on MapReduce				[11]		[12]		[13]	
		Se	**Sp**	**Acc**	**#**	**Acc**	**#**	**Acc**	**#**	**Acc**	**#**
Ovarian [11]	5GB	1	0,98	99	3	96	9	-	-	-	-
Gastric [12]	5GB	1	1	100	2	-	-	96	14	-	-
ESCC [13]	5GB	0,96	0,96	96	2	-	-	-	-	97	2
Ovarian	5GB	1	1	100	2	/		/		/	
Gastric	5GB	1	1	100	1						
ESCA	5GB	1	1	100	2						

Table 3. List of biomarkers discovered

Type of cancer	Biomarkers	Related to cancer
Gastric cancer	**VSIG2** (V-set and immunoglobulin domain containing 2)	Selected from 22 gastric cancer biomarkers [14].
	D26129_at (RNS1 Ribonuclease A (pancreatic))	Considered among the non-regulated genes in gastric cancer [15].
	M62628_s_at (Alpha-1 Ig germline C-region membrane-coding region)	
Ovarian cancer	**METTL7A** (methyltransferaselike 7A)	Selected among the 28 genes markers linked to cancer [16].
	GALC (galactosylceramidase)	Selected Among the new differentially expressed genes in cell lines MKN45 gastric cancer [17].
Esophageal cancer	**ADAM12** (ADAM Metallopeptidase Domain 12)	Biomarkers of two types of cancer, breast cancer and bladder cancer [18].
	GPR155 (G protein-coupled receptor 155)	Melanomabiomarker for mouse [19].
	SH3BGRL (SH3 domain binding glutamate-rich protein)	Selected from 20 potential biomarkers of breast cancer [20]

6 Conclusion and Future Work

In this paper, we presented a large-scale feature selection based on MapReduce for biomarker discovery. From the obtained results and comparative analysis we can conclude that our method performs well, and gives better performance than centralized approaches. For that, our method can be applied to handle large-scale datasets and to overcome the challenge of feature selection in Big Data, especially for biomarker discovery in bioinformatics. Our method is auto-scalable and can be executed in a distributed environment with any number of nodes. Our future work is to implement our approach on Spark for better performance in time execution.

References

1. Jay, S., Hanlee, J.: Next-generation DNA sequencing. Nature Biotechnology 26(10), 1135–1145 (2008)
2. Yvan, S., Inaki, I., Larranaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics 23(19), 2507–2517 (2007)
3. Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. In: OSDI 2004: Sixth Symposium on Operating System Design and Implementation, Sponsored by USENIX, in Cooperation with ACM SIGOPS, pp. 137–150 (2004)
4. Barnali, S., Debahuti, M.: A Novel Feature Selection Algorithm using Particle Swarm Optimization for Cancer Microarray Data. Procedia Engineering 38, 27–31 (2012)
5. Azli, A., et al.: Distributed visual enhancement on surveillance video with Hadoop Mapreduce and performance evaluation in pseudo distributed mode. Australian Journal of Basic and Applied Sciences 8(9), 38 (2014)
6. Kourid, A.: Iterative MapReduce for Feature Selection. International Journal of Engineering Research & Technology 3(7) (2014)
7. White, T.: Hadoop the definitive guide. O'Reilly Media (2012)
8. Bekkerman, R., Bilenko, M., Langford, J.: Scaling up Machine learning. Cambridge University Press (2011)
9. Sean, O., et al.: Mahout in action. Manning Publications (2011)
10. Gaizhen, Y.: The Application of MapReduce in the Cloud Computing. In: Intelligence Information Processing and Trusted Computing (IPTC), pp. 154–156. IEEE (2011)
11. Yun, J., Guocheng, X., Na, C., Shan, C.: A New Gene Expression Profiles Classifying Approach Based on Neighborhood Rough Set and Probabilistic Neural Networks Ensembl. In: Lee, M., Hirose, A., Hou, Z.-G., Kil, R.M. (eds.) ICONIP 2013, Part II. LNCS, vol. 8227, pp. 484–489. Springer, Heidelberg (2013)
12. Sun, X., et al.: Selection of interdependent genes via dynamic relevance analysis for cancer diagnosis. Journal of Biomedical Informatics 46(2), 252–258 (2013)
13. Tung, C.W., et al.: Identification of Biomarkers for Esophageal Squamous Cell Carcinoma Using Feature Selection and Decision Tree Methods. The ScientificWorld Journal (2013)
14. Yang, S., Chung, H.C., et al.: Novel biomarker candidates for gastric cancer. Oncology Reports 19(3), 675–680 (2008)
15. Geetha Ramani, R., Gracia Jacob, S.: Benchmarking Classification Models for Cancer Prediction from Gene Expression Data: A Novel Approach and New Findings. Studies in Informatics and Control 22(2), 133–142 (2013)
16. Li, X., et al.: SSiCP: a new SVM based Recursive Feature Elimination Algorithm for Multiclass Cancer Classification. Bio-Medical Materials and Engineering 23, S1027–S1038 (2014)
17. Tuan, T.F., et al.: Putative tumor metastasis-associated genes in human gastric cancer. International Journal of Oncology 41(3), 1068–1084 (2012)
18. Fröhlich, C., et al.: Molecular Profiling of ADAM12 in Human Bladder Cancer. Clinical Cancer Research 12(24), 7359–7368 (2006)
19. Hacker, E., et al.: Reduced expression of IL-18 is a marker of ultraviolet radiation-induced melanomas. Int. J. Cancer 123(1), 227–231 (2008)
20. Mayer, M.: Breast Cancer Prognostic Biomarkers. Accelerating science (2014)