12-7-2010

Feature selection for classification using an ant colony system

Nadia Abd-Alsabour
Bond University

Marcus Randall
Bond University, marcus_randall@bond.edu.au

Feature Selection for Classification Using an Ant Colony System

Nadia Abd-Alsabour
School of information technology, Bond University
QLD, Australia
nadia.abdalsabour@ieee.org

Marcus Randall
School of information technology, Bond University
QLD, Australia
mrandall@bond.edu.au

Abstract— **Many applications such as pattern recognition require selecting a subset of the input features in order to represent the whole set of features. The aim of feature selection is to remove irrelevant or redundant features while keeping the most informative ones. In this paper, an ant colony system approach for solving feature selection for classification is presented. The proposed algorithm was tested using artificial and real-world datasets. The results are promising in terms of the accuracy of the classifier and the number of selected features in all the used datasets. The results of the proposed algorithm have been compared with other results available in the literature and found to be favorable.**

Keywords- Ant colony optimisation, feature selection.

I. INTRODUCTION

Pattern classification is the assignment of an input pattern to one of several predefined categories/classes [1]. Pattern recognition can be viewed as a two-fold task: a) learning (sometimes called the training phase) the invariant and common properties of a set of samples (training set) that characterise a class, and b) deciding (through the classifier) that a new sample is a possible member of the class by noting that it has properties common to those of the set of samples [2].

The basic component of any pattern recognition system is the classifier whose task is to partition the feature space into class-labelled decision regions, one region for each category [1], [3]. The performance of classifiers is sensitive to the choice of the features that are used for constructing those classifiers. The choice of the features that are presented to the classifiers affects the following important properties:

- The accuracy of the classifiers,
- the time needed for learning the classifiers, and
- the number of examples needed for learning the classifiers [1].

Increasing the amount of training data may affect the time needed for the learning and may lead to finding a less optimal solution. This is because many features in the training data may be irrelevant to the classification task. For instance, data that records the day of the week on which a bank loan application was completed is unlikely to be relevant to the success of the application. Moreover, other features may be redundant. Such features may slow down and mislead the learning step and do not contribute to the classification process. Hence, feature selection (FS) from the original set of features is highly desirable in many situations in order to remove any irrelevant or redundant features [4].

Feature selection is the problem of selecting a subset of features without reducing the accuracy of representing the original set of features. Feature selection is used in many applications to remove irrelevant and redundant features where there are high dimensional datasets. These datasets may contain a high degree of irrelevant and redundant features that may decrease the performance of the classifiers.

The main approaches that are used for solving feature selection problems can be classified into filter or wrapper approaches, depending on whether or not feature selection is done independently of the classifier. Some researchers use hybrid methods to take advantage of these two approaches and to handle large datasets [5].

Feature selection can be seen as an optimisation problem that involves searching the space of possible feature subsets to identify the optimal one. Many optimisation techniques such as genetic algorithms (GAs) [1], tabu search (TS), simulated annealing (SA) and ant colony optimisation algorithms (ACO) have been used for solving feature selection.

This paper introduces an implementation for ACO (using Ant Colony System (ACS)) to solve feature selection problems as explained in Section IV.

The rest of this paper is organised as follows. Section II introduces the basics of ACO. Section III addresses the fundamentals of ACS on which the proposed algorithm was developed. The fourth section briefly describes some of ACO algorihtms used to solve feature selection problem. The fifth section explains the proposed algorithm. Section VI details the experiments carried out and presents the results. The discussion of the results is given in Section VII. Section VIII concludes this paper and highlights future work in this area.

II. ANT COLONY OPTIMISATION

Real ants are able to find the shortest path between their nest and food sources because of the chemical substance (pheromone) that they deposit on their way. The pheromone evaporates over time so the shortest paths will contain more pheromone and will subsequently attract a greater number of ants.

978-0-7695-4295-9/10 $26.00 © 2010 IEEE
DOI 10.1109/eScienceW.2010.23

ACO algorithms simulate the foraging behavior of some ant species [6]. ACO algorithms use two factors for guiding the search process. These are: 1) the pheromone values (numerical values as a simulation for the pheromone that real ants deposit on their way to and from their nest), and 2) heuristic information for the selection of solution component values. There are two types of heuristic information used by ACO algorithms; static heuristic information (that is computed at initialisation time and then remains unchanged throughout the whole algorithm's run, such as the distances between cities in the travelling salesman problem (TSP)) and dynamic heuristic information (that depends on the partial solution constructed so far and therefore is computed at each step of an ant's walk) [7].

One of the recent trends in ACO is to solve industrial problems proving that it is useful for real-world applications [8]-[9]. Recently, researchers have adopted ACO to solve feature selection problems ([10]-[14]). We will highlight the main differences between these algorihtms and the proposed algorithm in the following sections.

In traditional ACO algorithms, the pheromone values are associated with the nodes or the edges of the construction graph representing the problem (depending on the chosen problem representation), which may also contain heuristic information representing prior information about the given problem [7]-[8].

In the proposed algorithm, a construction graph is not used. Pheromone is associated with each feature. Although there is no heuristic information known in advance for this type of optimisation problems, it is used in the proposed algorithm in computing the move probability that is used by each ant to select a particular feature in each construction step. The proportion of ants that choose a particular feature is used as the heuristic information. This is explained in detail in Section IV.

III. Ant Colony System

In this section, a variant of ACO is explained, namely ant colony system (ACS) for solving the travelling salesman problem. This is because many aspects of the application have commonality with feature selection problem.

Ant colony system is considered one of the most successful ACO algorithms [15]. Since its appearance, it has been used to solve a variety of optimisation problems. ACS is a constructive algorithm where for each iteration, each ant chooses the next city to be visited (*j*) using the pseudorandom proportional rule that is computed according to the following equation:

$$\arg\max_{C_{il} \in N(s^p)} \{\tau.\eta_{il}^{\beta}\} \tag{1}$$

This equation is used by each ant at each construction step to choose the next city depending on a random variable q uniformly distributed in [0, 1] and a parameter q_0 $(0 \leq q_0 \leq 1)$. The equation is used if $q \leq q_0$.

After each construction step, the local pheromone update is performed by all ants to the last edge traversed according to the following equation:

$$\tau_{ij} = (1-\varphi).\tau_{ij} + \varphi.\tau_0 \tag{2}$$

where $\varphi \in (0,1]$ and τ_0 is the initial pheromone.

The local pheromone update leads to decreasing the pheromone values on the edges and hence encourages subsequent ants to choose other edges and to therefore produce different solutions. This is essential to prevent pre-mature convergence.

At the end of each iteration, the pheromone values are updated by the best ant only according to the following equation:

$$\tau_{ij} = \begin{cases} (1-\rho).\tau_{ij} + \rho.\Delta\tau_{ij} & \text{if } (i,j) \in \text{best tour}, \\ \tau_{ij} & \text{otherwise} \end{cases} \tag{3}$$

where $\Delta\tau_{ij} = 1/L_{best}$ and L_{best} is the length of the tour constructed by the best ant.

IV. Related Work

In this section, we explain briefly some of the ACO algorithms that have been used to solve feature selection problem.

An ACO approach called 'antselect' for variable selection in quantitative structure activity relationship (QSAR) has been developed by Izrailev and Agrafiotis [10]. In antselect, a weight is associated with each feature and the selection of each variable depends on it, i.e., the variable with a larger weight has a higher probability of being selected than the ones with smaller weights.

In Gunturi et al. [11], antselect was used with two extra features:

- Only variables with an inter-correlation coefficients less than 0.75 are selected, and
- The number of variables to be selected is fixed at the beginning of the selection process.

Shen et al. [12] express the feature selection problem in binary notation. The move probability for any feature is zero or one. A one means that this feature will be selected and zero means it will not. Shi et al. [13] used this algorithm with a different dataset. The work of Shamsipur et al. [14] is similar to Shen et al. [12] but the move probability depends on the weights associated with the variables and the number of the selected variables by an ant is randomly chosen.

V. The Proposed Algorithm

The proposed algorithm is a wrapper-based system that deals with the problem of feature selection as a binary problem where a set of binary bits (of a length equivalent to the number of the features in the given dataset) is associated with each ant to represent its feature selection. If the n^{th} bit is a 1 this means that the feature number n in the given dataset is selected, otherwise this feature is not selected. Thus, the concept of path in the traditional ACO algorithms is not meaningful here. At the start of the algorithm, the bits are randomly initialised to zeros and ones.

The pheromone values are associated with the features. At each construction step, each ant selects a feature out of all the features with the probability computed according to the following equation:

$$P_i = \tau_i \, . \, \Delta\tau_i \tag{4}$$

where τ_i is the pheromone value associated with feature *i* and $\Delta\tau_i$ is the proportion of ants that have selected this feature. It acts as heuristic information that represents the desirability of feature *i*. In ACO algorithms, the design of the move probability is critical. Here both the pheromone values and heuristic information are used to compute it.

After each construction step, the local pheromone update is performed by all ants according to the following equation:

$$\tau_i = (1-\varphi) \, . \, \tau_i \quad + \quad \varphi . \tau_0 \tag{5}$$

At the end of each iteration, the global pheromone update is performed according to the following equation:

$$\tau_i = (1-\rho) \, . \, \tau_i \quad + \quad (\rho \, . 1 / L_{best})^{\beta} \tag{6}$$

where L_{best} is the number of features selected by the best ant and ρ is the evaporation rate. This equation is used by the best ant at the end of each iteration and is applied to all features that it has chosen.

The updating of the pheromone values is important in order to reinforce those features that lead to high quality feature subsets. Features that belong to good solutions will contain larger pheromone values. Consequently, these features tend to be selected more often.

The main steps of the proposed algorithm are as follows:

- Initialisation: in this phase, the parameters of the proposed algorithm are set.
- Construct a solution by each ant: at each construction step, each ant selects the next feature using the move probability that is calculated based on the pheromone values and the heuristic information according to Equation 4. Each feature can be selected at most once. An ant's solution represents its own feature subset. The length of each ant's feature subset does not need to be equal to other ants' solutions.
- Build the support vector machine (SVM) model: each ant passes its feature subset to the classifier and receives its accuracy.
- Update the best solution and the best ant based on the accuracy of the classifier.
- Update the pheromone values using Equation 6 by the best ant to its subset.
- If a predefined maximum number of iterations is reached, the algorithm halts and outputs the best solution encountered.

VI. Computational Experience

In order to test the proposed algorithm, a SVM learning algorithm is used. Although not all machine learning algorithms require the phase of feature selection, feature selection is important in building a SVM-based classification [16].

We used the accuracy of the classifier as a fitness function where each ant evaluates its solution based on its ratio of correct classifications. Classifier accuracy indicates how accurately a given classifier will label future data on which the classifier has not been trained. One of the ways used to estimate classifier accuracy is *k*-fold cross validation in which the initial data are randomly partitioned into *k* mutually exclusive folds s_1, s_2, …, s_k each of which is of approximately equal size. Training and testing is performed *k* times. The classifier of the first iteration is trained on folds s_2, s_3, …., s_k and the first fold is kept for testing. In iteration *i*, the fold s_i is the test fold and the remaining folds are collectively used to train the classifier [4]. In this case, the classifier accuracy is computed according to the following equation:

$$\text{classifier accuracy} = \frac{\text{Overall number of correct classifications from } k \text{ iterations}}{\text{The total number of samples in the initial data}} \tag{7}$$

A. Datasets

In order to test the proposed algorithm, ten experiments using ten artificial and real world datasets were carried out. The datasets comprise two categories: datasets from statistical datasets [17] and datasets from the UCI (University of California, Irvine) machine learning repository. The latter is a collection of databases, domain theories, and data generators that are used all over the world as a primary source of machine learning datasets for the empirical analysis of machine learning algorithms [18]. The details of the datasets are shown in Table I.

TABLE I. THE DETAILS OF THE USED DATASETS.

Dataset Name	No. of features	No. of classes	Class label type	No. of instances
Statistical datasets				
Backache	32	2	Numeric	180
Prnn_virus3	17	4	Numeric	38
Prnn_viruses	17	6	Numeric	61
Analcatdata_authorship	70	4	Numeric	841
Analcatdata_marketing	32	5	Nominal	364
UCI datasets				
Kdd_synthetic_control	61	6	Nominal	600
Sonar	60	2	Numeric	208
Vehicle	18	4	Numeric	846
Dermatology	34	6	Nominal	366
Wine	13	3	Nominal	178

B. Method

In our experiments, the following two systems were developed:

- SVM: that uses the entire set of features (without the phase of feature selection) [19], and
- SVM-FS: that uses a subset of features selected by the proposed algorithm (with the phase of feature selection).

In this paper, we focus on testing the effect of feature selection on the performance of the classifier. The performance of SVM itself was not optimised, although further investigation is required since it affects the performance of the whole system. We used the default values to its parameters in both cases, with and without the use of feature selection. The C-classification SVM of package e1071 of the R language with the default values to its parameters was used.

In these two systems, 5-fold cross validation was used. The number of ants was set to the number of the features in the given dataset. The initial pheromone was set to 1. The number of iterations is 10 iterations. ρ was set to 0.4 for all experiments except for datasets that contain more than 30 features for which it was 0.3. β was set to 0.2. φ was set to 0.2 and α to 1. After many experiments, these values of the parameters were used since they give the best performance of the proposed algorithm.

C. Results

Table II shows the results of these two systems using the above mentioned datasets. The results for the algorithm represent the average of ten independent runs. These systems are implemented using the R language [20]-[21] and the WEKA machine learning tool [22]-[23]. All the experiments were run on a PC with a 2 GHz CPU and 2 GB RAM.

TABLE II. THE ACCURACY OF SVM WITH AND WITHOUT THE USE OF FEATURE SELECTION.

Dataset Name	No. of original features	Avg. no. of selected features using ACS_SVM	SVM (without FS)	SVM-FS (with FS)
Statistical datasets				
Backache	32	18.8	0.9	0.9222
Prnn_virus3	17	9.6	0.9474	0.9842
Prnn_viruses	17	10.2	0.0164	0.9082
Analcatdata_authorship	70	36.8	1	1
Analcatdata_marketing	32	24.6	0.6401	0.6692
UCI datasets				
Kdd_synthetic_control	61	30.2	1	1
Sonar	60	43.4	0.9808	1
Vehicle	18	15.6	0.8463	0.8584
Dermatology	34	23.2	0.9888	0.9955
Wine	13	8.6	1	1

The previous results show that SVM-FS with the proposed algorithm for performing feature selection outperforms SVM that uses all the features in all the datasets (the accuracy of SVM (the last column) is larger than that of SVM with all features (the fourth column)). The number of features selected by the proposed algorithm (the third column) is significantly smaller than the total number of the features in the original datasets (the second column) in all of the datasets. It should be noted that the proposed algorithm can give better results if we set the parameters individually for each dataset.

Table III compares some of the results of the proposed algorithm with the results of other feature selection algorithms from the literature as follows:

1. Genetic algorithm (GA) for feature selection with the use of DistAI classifier (GADistAI). These results are taken from Yang and Honavar [1]. GAs are one of the most common approaches for feature selection.
2. Particle swarm optimisation (PSO). These results are taken from Tu et al. [24].
3. Chaotic binary particle swarm optimisation (CBPSO). These results are taken from Chuang et al. [25].

TABLE III. COMPARISON BETWEEN THE PROPOSED ALGORITHM AND OTHER FEATURE SELECTION ALGORITHMS

Dataset Name	**PSO_SVM**		**CBPSO_KNN**		**GADistAI**		**ACS_SVM** (proposed)	
	Avg. no. of selected features	accuracy	Avg. no. of selected features	accuracy	Avg. no. of selected features	accuracy	Avg. no. of selected features	accuracy
Sonar	34	96.15	27	93.27	32.2	0.84	43.4	1
Vehicle	-	-	10	74.35	9.4	0.501	15.6	0.8584
Wine	8	1	8	99.44	8.2	0.967	8.6	1

This comparison shows that the proposed algorithm outperforms the other approaches in solving feature selection problems in terms of the accuracy of the classifier. The accuracy of SVM compared with the proposed algorithm (the last column) is higher than that of the other approaches (the third, the fifth, and the seventh columns) in all of the datasets used in this comparison.

VII. Discussion

The proposed algorithm deals with the feature selection problem as a binary problem and integrates this into an ACS framework. The proposed algorithm is general as it can be used with any dataset as shown in the previous sections (since these datasets are from different domains). This is motivated by the need for feature selection in different application areas. One example, the dermatology dataset, is illustrated here. It is known that the differential diagnosis of erythemato-squamous diseases is a real problem in dermatology. They all share the clinical features of erythema and scaling, with very few differences. The diseases in this group are psoriasis, seboreic dermatitis, lichen planus, pityriasis rosea, cronic dermatitis, and pityriasis rubra pilaris. Usually a biopsy is necessary for the diagnosis but unfortunately these diseases share many histopathological features as well. Another difficulty for the differential diagnosis is that a disease may show the features of another disease at the beginning stage and may have the characteristic features at the later stages. Patients were first evaluated clinically with 12 features. Afterwards, skin samples were taken for the evaluation of 22 histopathological features. The values of the histopathological features are determined by an analysis of the samples under a microscope. All of that makes selecting the most informative features is highly desirable in many applications related to this domain [26].

The merit of the proposed algorithm is that it is a combination between the traditional ACO and the ACO algorithms that deals with feature selection as a binary problem.

The main characteristics of the proposed algorithm are as follows:

- The move probability is not zero or one as in Shen et al. [12] or uses only the weight (pheromone) associated with each feature as in Izrailev and Agrafiotis [10]. Yet, it uses a similar equation to ACS where two factors are used to compute it. These two factors are the old pheromone value associated with a feature and the heuristic information. This heuristic information indicates how often a particular feature has been chosen by different ants. Therefore the value of the move probability depends on the result of the equation.
- It uses the local pheromone update as in ACS. Using the local pheromone update leads to decreasing the pheromone values on the features that encourages subsequent ants to choose other features and subsequently produce diverse solutions.
- A different global pheromone update equation rather than the usual one is used for feature selection. This equation is used by the best ant as in ACS. The global pheromone update in this algorithm uses the old pheromone value associated with the feature multiplied by the inverse of the length of the best ant's solution.
- The length of the solution of each ant is not fixed as in Gunturi et al. [11] and in Shamsipur et al. [14] allowing each ant to construct its solution without any prior restriction. The only criterion used in evaluating the constructed solution by each ant is the accuracy of the classifier.

VIII. Conclusions and Future Work

In this paper, the feature selection problem for classification was solved using an ant colony system approach on an SVM classifier with ten artificial and real world datasets. The proposed algorithm deals with the feature selection problem as a binary one. We used heuristic information in order to guide the search process besides the pheromone values as in most of the conventional ACO algorithms. The results are promising in terms of the solution quality and the number of selected features in all the datasets.

Although the obtained results are encouraging, further investigation into the adjustment of the values of the large number of parameters that the proposed algorithm has, is required. These are its parameters besides the parameters of SVM. All of these parameters need to be adjusted so that the performance of the proposed algorithm is enhanced.

In order to sensibly attempt this parameter exploration, as well as to be able to solve large real-world feature selection problems in reasonable amounts of computational time, the use of parallel and high performance computation will be necessary. It should be noted that by using ensemble feature selection the performance of the proposed algorithm could be enhanced, especially when dealing with large datasets. Recently, the ensemble feature selection technique has been used to select different feature subsets for each base learner to construct an ensemble classifier. This is exemplified by Li et al. [27] who they used dynamic Adaboost learning with feature selection based on a parallel genetic algorithm. Another research direction is to use high performance computing techniques such as parallel multicategory support vector machines [28]. This is because using support vector machines for large datasets is very computationally intensive and using high performance computing techniques can improve its performance. For example, the experiments of Zhang et al. [28] showed that using parallel multicategory support vector machines enhanced the accuracy of the classification.

ACKNOWLEDGMENT

The authors would like to thank Dr. Andrew Lewis for his valuable comments on this paper.

REFERENCES

[1] Yang, J., Honavar, V.: Feature subset selection using a genetic algorithm, IEEE Intelligent Systems, vol. 13, no. 2, pp. 44–49, (1998)

[2] Bandyopadhyay, S., and Pal, S.: Classification and Learning Using Genetic Algorithms-Applications in Bioinformatics and Web Intelligence, Springer-Verlag Berlin Heidelberg, (2007)

[3] Duda, R.O., Hart, P.E.: Pattern classification and scene analysis, John Wiley & Sons, Inc. (1973)

[4] Han, J., and Kamber, M.: Data mining concepts and techniques, Academic Press, (2001)

[5] Liu, H., Motoda, H.: Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic, Norwell, MA, (1998)

[6] Dorigo, M., Bonabeou, E., and Theraulaz, G.: Inspiration for optimization from social insect behavior, Nature, vol. 406, pp. 39-42, (2000)

[7] Dorigo, M., and Stutzle, T.: Ant Colony Optimization, MIT Press, (2004)

[8] Dorigo, M., Birattari, M., and Stutzle, T.: Ant colony optimization-artificial ants as a computational intelligence technique, IEEE computational intelligence magazine, (2006)

[9] Dorigo, M., Caro, G.D., and Sampels, M.: Ant Algorithms, Springer-Verlag (2002)

[10] Izrailev, S., Agrafiotis, D.: Variable selection for QSAR by artificial ant colony systems, SAR QSAR in environmental research, vol. 13, pp. 417-423, (2002)

[11] Gunturi, S., Narayanan, R., Khandelwal, A.: In silico ADME modeling 2: Computational models to predict human serum albumin binding affinity using ant colony systems, Bio-organics & Medicinal Chemistry, vol. 14, pp. 4118-4129, (2006)

[12] Shen, Q., Jiang, J.H., Tao, J.C., Shen, G.L., and Yu, R.Q.: Modified Ant Colony Optimization Algorithm for Variable Selection in QSAR Modeling: QSAR Studies of Cyclooxygenase Inhibitors. J Chem. Inf. Model, vol. 45, pp. 1024-1029, (2005)

[13] Shi, W., Shen, Q., Kong, W., and Ye, B.: QSAR analysis of tyrosine kinase inhibitor using modified ant colony optimization and multiple linear regression, European Journal of Medicinal Chemistry. vol. 42, pp. 81-86, (2007)

[14] Shamsipur, M., Zare-Shahabadi, V., Hemmateenejad, B., and Akhond, M.: Ant colony optimization: a powerful tool for wavelength selection, Journal of Chemometrics, vol. 20, pp. 146-157, (2006)

[15] Dorigo, M., and Gambardella, L.M.: Ant colony system: A cooperating learning approach to the traveling salesman problem, IEEE Transactions on evolutionary computation, vol.1, no. 1, pp. 53-66, (1997)

[16] Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning-Data Mining, Inference, and Prediction, Second Edition, Springer, (2008)

[17] http://www.cs.waikato.ac.nz/ml, viewed 25th June 2010.

[18] Murphy, P.M., Aha, D.W.: UCI Repository of machine learning databases, Department of information and computer science, University of California, Irvine, CA, http://www.ics.uci.edu/_mlearn/MLRepository.html, viewed 25th June 2010.

[19] Abd-Alsabour, N.: Feature Selection for Classification Using an Ant System Approach, in Hinchey, M., et al. (Eds.): IFIP Advances in Information and Communication Technology, vol. 329, pp. 233-241, (2010)

[20] R: A Language and Environment for Statistical Computing 2006 [http://www.R-project.org]. R Foundation for Statistical Computing, Vienna, Austria.

[21] Dalgaard, P.: Introductory statistics with R, Springer, (2008)

[22] http://www.cs.waikato.ac.nz/ml/weka, viewed 25th June 2010.

[23] Witten, I., and Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco, (2005)

[24] Tu, C.J., Chuang, L.Y., Chang, J.Y, and Yang, C.H.: Feature Selection using PSO-SVM, International Journal of Computer Science, vol. 18, pp. 33-1, (2007)

[25] Chuang, L.Y., Li, J.C., and Yang, C.H.: Chaotic Binary Particle Swarm Optimization for Feature Selection using Logistic Map, Proceedings of the International Conference of Engineers and Computer Scientists, vol. I, Hong Kong (2008)

[26] http://archive.ics.uci.edu/ml/datasets/Dermatology, viewed 10th October 2010.

[27] Li, R., Lu, J., Zhang, Y., Zhao, T.: Dynamic Adaboost learning with feature selection based on parallel genetic algorithm for image annotation, Knowledge-Based Systems, vol. 23, pp. 195–201, (2010)

[28] Zhang, C., Li, P., Rajendran, A., Deng Y., and Chen D.: Parallelization of multicategory support vector machines (PMC-SVM) for classifying microarray data, BMC Bioinformatics, vol. 7, Suppl. 4, (2006)