ON THE GRADIENT-BASED ALGORITHM FOR MATRIX FACTORIZATION APPLIED TO DIMENSIONALITY REDUCTION

Vladimir Nikulin and Geoffrey J. McLachlan
Department of Mathematics, University of Queensland, Brisbane, Australia

Keywords: Matrix factorisation, Gradient-based optimisation, Cross-validation, Gene expression data.

Abstract: The high dimensionality of microarray data, the expressions of thousands of genes in a much smaller number of samples, presents challenges that affect the applicability of the analytical results. In principle, it would be better to describe the data in terms of a small number of metagenes, derived as a result of matrix factorisation, which could reduce noise while still capturing the essential features of the data. We propose a fast and general method for matrix factorization which is based on decomposition by parts that can reduce the dimension of expression data from thousands of genes to several factors. Unlike classification and regression, matrix decomposition requires no response variable and thus falls into category of unsupervised learning methods. We demonstrate the effectiveness of this approach to the supervised classification of gene expression data.

1 INTRODUCTION

The analysis of gene expression data using matrix factorization has an important role to play in the discovery, validation, and understanding of various classes and subclasses of cancer (Brunet et al., 2004). One feature of microarray studies is the fact that the number of samples collected is relatively small compared to the number of genes per sample which are usually in the thousands. In statistical terms this very large number of predictors compared to a small number of samples or observations makes the classification problem difficult. An efficient way to solve this problem is by using dimension reduction statistical techniques.

In principle, it would be better to describe the data in terms of a small number of metagenes, which could reduce noise while still capturing the invariant biological features of the data (Tamayo et al., 2007). As it was noticed in (Koren, 2009), latent factor models are generally effective at estimating overall structure that relates simultaneously to most or all items.

The SVM-RFE (support vector machine recursive feature elimination) algorithm was proposed in (Guyon et al., 2002) to recursively classify the samples with SVM and select genes according to their weights in the SVM classifiers. However, it was noted in (Zhang et al., 2006) that the SVM-RFE approach used the top ranked genes in the succeeding cross-validation for the classifier. This cross-validation (CV) scheme will generate a biased estimation of errors. In the correct CV scheme it is necessary to repeat feature selection for any CV loop which may be very expensive in terms of computational time.

The SVM-RFE employs very simple concept to select the given number of top-ranked genes. A deficiency of this approach is that the features could be correlated among themselves (Peng and Ding, 2005). For a long time, people already realized the "*p* best features are not the best *p* features". For example, if gene *i* is ranked high by the SVM-RFE, other genes highly correlated with gene *i* are also likely to be selected. It is frequently observed that simply combining a "very effective" gene with another "very effective" gene does not form a better feature selection.

Matrix factorization, an unsupervised learning method, is widely used to study the structure of the data when no specific response variable is specified. In contrast to the SVM-RFE, we can perform dimension reduction using matrix factorization only once.

Note that the methods for non-negative matrix factorization (NMF) which was introduced in (Lee and Seung, 2000) are valid under the necessary condition that all the elements of all input and output matrices are non-negative. In Section 2.1 we formulate our general method for matrix factorization, which is significantly faster compared to NMF.

Nikulin V. and J. McLachlan G. (2010).
ON THE GRADIENT-BASED ALGORITHM FOR MATRIX FACTORIZATION APPLIED TO DIMENSIONALITY REDUCTION.
In *Proceedings of the First International Conference on Bioinformatics*, pages 147-152
DOI: 10.5220/0002736601470152

2 METHODS

Let $(\mathbf{x}_j, \mathbf{y}_j)$, $j = 1, \ldots, n$, be a training sample of observations where $\mathbf{x}_j \in \mathbb{R}^p$ is p-dimensional vector of features, and $\mathbf{y}_j$ is a multi-class label. Boldface letters denote vector-columns. Let us denote by $\mathbf{X} = \{x_{ij}, i = 1, \ldots, p, j = 1, \ldots, n\}$ the matrix containing the observed values on the p variables.

For gene expression studies, the number p of genes is typically in the thousands, and the number n of experiments is typically less than 100. The data are represented by an expression matrix $\mathbf{X}$ of size $p \times n$, whose rows contain the expression levels of the p genes in the n samples. Our goal is to find a small number of metagenes or factors. We can then approximate the gene expression patterns of samples as a linear combinations of these metagenes. Mathematically, this corresponds to factoring matrix $\mathbf{X}$ into two matrices

$$\mathbf{X} \sim \mathbf{AB}, \tag{1}$$

where the matrix $\mathbf{A}$ has size $p \times k$, with each of the k columns defining a metagene; entry a_{if} is the coefficient of gene i in metagene f. The matrix $\mathbf{B}$ has size $k \times n$, with each of the $\mathbf{B}$ columns representing the metagene expression pattern of the corresponding sample; entry b_{fj} represents the expression level of metagene f in sample j.

2.1 Main Model

Let us consider

$$L(\mathbf{A}, \mathbf{B}) = \frac{1}{p \cdot n} \sum_{i=1}^{p} \sum_{j=1}^{n} \left(E_{ij}^2 + R_{ij}\right), \tag{2}$$

where $E_{ij} = x_{ij} - \sum_{f=1}^{k} a_{if} b_{fj}$,

$$R_{ij} = \sum_{f=1}^{k} \left(\frac{c_a a_{if}^2}{n} + \frac{c_b b_{fj}^2}{p} \right), \tag{3}$$

where c_a and c_b are non-negative constants (known as ridge parameters).

Remark 1. The target of the important regularization term in (3) is to ensure stability of the model. We used in our experiments relatively small values $c_a = c_b = 0.001$, which cannot be regarded as optimal.

The target function (2) needs to be minimised. It includes in total $k \cdot (p + n)$ regulation parameters and may be unstable if we minimise it without taking into account the mutual dependence between elements of the matrices $\mathbf{A}$ and $\mathbf{B}$.

As a solution to the problem, we can go consequently through all the differences E_{ij}, minimising them as a function of the particular parameters which are involved in the definition of E_{ij}. Compared to usual gradient-based optimisation, in our optimisation model we are dealing with two sets of parameters, and we should mix uniformly updates of these parameters, because these parameters are dependent.

Algorithm 1. Matrix factorization.

```
1: Input: $\mathbf{X}$ - matrix of microarrays.
2: Select $\ell$ - number of global iterations; $k$ - number of factors; $\lambda > 0$ - initial learning rate, $0 < \xi < 1$ - correction rate, $c_a > 0$ and $c_b > 0$ -ridge parameters, $L_S$ - initial value of the target function.
3: Initial matrices $\mathbf{A}$ and $\mathbf{B}$ may be generated randomly.
4: Global cycle: repeat $\ell$ times the following steps 5 - 17:
5:   genes-cycle: for $i = 1$ to $p$ repeat steps 6 - 15:
6:     tissues-cycle: for $j = 1$ to $n$ repeat steps 7 - 15:
7:       compute prediction $S = \sum_{f=1}^{k} a_{if} b_{fj}$;
8:       compute error of prediction: $E = x_{ij} - S$;
9:       internal factors-cycle: for $f = 1$ to $k$ repeat steps 10 - 15:
10:        compute $\alpha = a_{if} b_{fj}$;
11:        update $a_{if} \Leftarrow a_{if} + \lambda \left(E \cdot b_{fj} - \frac{1}{n} c_a a_{if}\right)$;
12:        $E \Leftarrow E + \alpha - a_{if} b_{fj}$;
13:        compute $\alpha = a_{if} b_{fj}$;
14:        update $b_{fj} \Leftarrow b_{fj} + \lambda \left(E \cdot a_{if} - \frac{1}{p} c_b b_{fj}\right)$;
15:        $E \Leftarrow E + \alpha - a_{if} b_{fj}$;
16:  compute $L = L(\mathbf{A}, \mathbf{B})$;
17:  $L_S = L$ if $L < L_S$; otherwise: $\lambda \Leftarrow \lambda \cdot \xi$.
18: Output: $\mathbf{A}$ and $\mathbf{B}$ - matrices of metagenes or latent factors.
```

The following partial derivatives are necessary for Algorithm 1 (see steps 11 and 14):

$$\frac{\partial E_{ij}^2}{\partial a_{if}} = -2 \cdot E_{ij} b_{fj} + \frac{2 c_a a_{if}}{n}, \tag{4a}$$

$$\frac{\partial E_{ij}^2}{\partial b_{fj}} = -2 \cdot E_{ij} a_{if} + \frac{2 c_b b_{fj}}{p}. \tag{4b}$$

Similar to the standard gradient-based optimisation applied to the squared loss function we can optimise here value of the step-size. However, taking into account the complexity of the model, it will be better to maintain fixed and small values of the step size or learning rate. In all our experiments we conducted matrix factorization with the above Algorithm 1 using 100 global iterations with the following regulation parameters: $\lambda = 0.01$ - initial learning rate, $\xi = 0.75$ -correction rate. Figure 1(a) illustrates convergence of Algorithm 1. Figure 2 illustrates matrices $\mathbf{B}$ as an outcome of Algorithm 1.

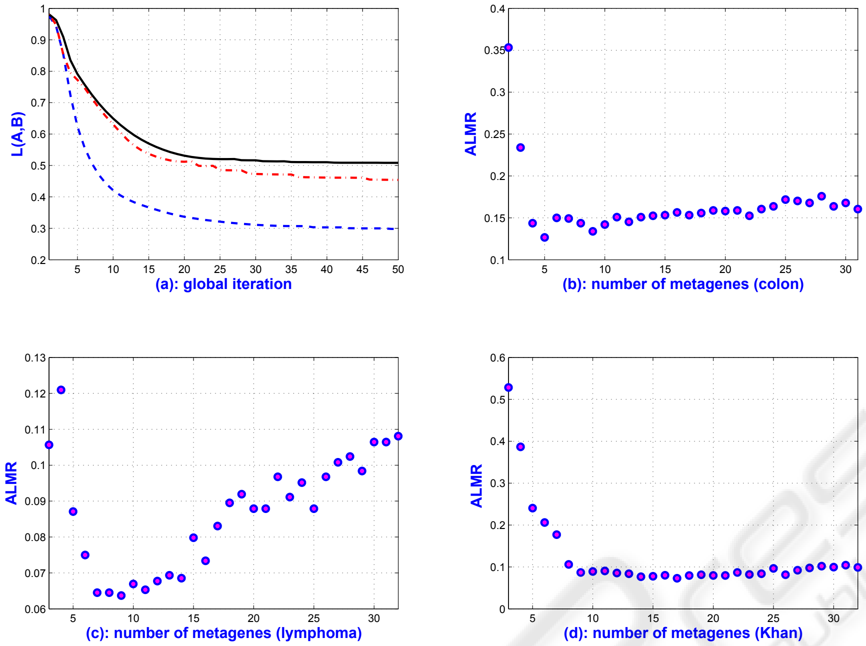

Figure 1: From the top: (a) behavior of the target (2) as a function of global iteration (see global cycle in Algorithm 1), used $k = 10$ - number of metagenes; dashed blue, solid black and dot-dashed red lines correspond to the colon, leukaemia and lymphoma cases. LMRs as a function of number of metagenes, where the following schemes were used in the experiments: (b) $SCH(20, SVM, LOO)$; (c-d) $SCH(20, MLR, LOO)$, see Section 4.

2.2 Multinomial Logistic Regression (MLR)

Following (Bohning, 1992) let us consider maximisation of the log-likelihood applied to the matrix **B** which was produced as an outcome of Algorithm 1:

$$R(\mathbf{W}) = \sum_{j=1}^{n} \left[\sum_{\ell=1}^{m} y_{j\ell} u_{j\ell} - \log(1 + \sum_{\ell=1}^{m} \exp(u_{j\ell}) \right],$$

where $y_{j\ell} = 0$ for all $\ell = 1, \dots, m+1$, besides one i with $y_{ji} = 1$, $\mathbf{W}$ is a $m \times k$ matrix of linear coefficients, the matrix $\mathbf{U}$ with elements $u_{j\ell}$ is defined as a product **WB**

Our task is to find a solution for the equation $\nabla R(\mathbf{W}) = 0$, which represents necessary condition of an optimum.

The following equation is called Newton's step and may be used as a base for the iterative algorithm

$$\mathbf{W}_c^{(s+1)} = \mathbf{W}_c^{(s)} - \nabla^2 R(\mathbf{W}^{(s)})^{-1} \nabla R(\mathbf{W}^{(s)}), \qquad (5)$$

where ∇R is a mk- dimensional vector column (gradient), and $\nabla^2 R$ is a mk- dimensional squared matrix of second derivatives (known as Hessian matrix). All necessary details regarding computation of the gradient vector ∇R and Hessian matrix $\nabla^2 R$ in the terms of Kronecker products may be found in (Bohning, 1992). There is a direct correspondence between matrix $\mathbf{W}$ and the mk-dimensional vector-column $\mathbf{W}_c$, where first m elements of $\mathbf{W}_c$ coincide with elements of the first row of the matrix $\mathbf{W}$ and so on.

Note that the inverse Hessian matrix in (5) may not exist, and, anyway, computation of an inverse matrix may be a very difficult task in the case if dimension is high. We propose the following alternative update procedure: $\mathbf{W}_c^{(s+1)} = \mathbf{W}_c^{(s)} - \mathbf{v}$, where $\mathbf{v}$ is mk-dimensional vector-column as a solution of the regularised squared minimisation problem

$$\mu \mathbf{v}^T \mathbf{v} + \|\nabla^2 R(\mathbf{W}^{(s)}) \mathbf{v} - \nabla R(\mathbf{W}^{(s)})\|^2,$$

where μ is a ridge parameter for the regularisation term. Note that regularisation term is a very important in order to stabilise solution and reduce overfitting, see for more details (Mol et al., 2009). We used in our experiments value $\mu = mk/100$.

3 DATA

The colon dataset[1] is represented by a matrix of 62 tissue samples (40 negative and 22 positive) and 2000

[1] http://microarray.princeton.edu/oncology/affydata/index.html

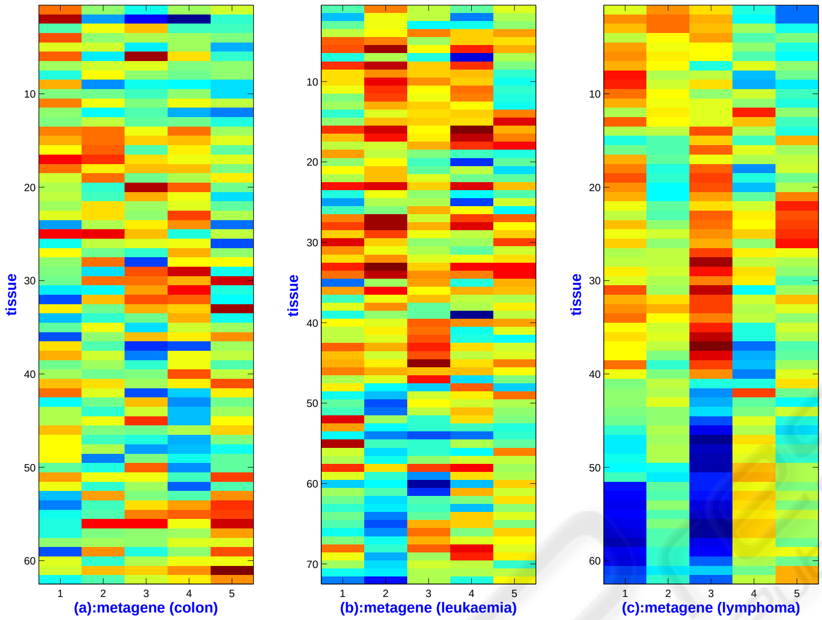

Figure 2: Images of the matrix **B** in (1), $k = 5$: (a) colon (sorted from the top: 40 positive then 22 negative), (b) leukaemia (sorted from the top: 47 ALL, then 25 AML) and (c) lymphoma (sorted from the top: 42 DLCL, then 9 FL, last 11 CLL). All three matrices were produced using Algorithm 1 with 100 global iterations as it was described in Section 2.1.

genes. The microarray matrix for this set thus has $p = 2000$ rows and $n = 62$ columns.

The leukaemia dataset[2] contains the expression levels of $p = 7129$ genes of $n = 72$ patients, among them, 47 patients suffer from acute lymphoblastic leukaemia (ALL) and 25 patients suffer from the acute myeloid leukaemia (AML).

We followed the pre-processing steps of (Dudoit et al., 2002) applied to the leukaemia set: 1) thresholding: floor of 1 and ceiling of 20000; 2) filtering: exclusion of genes with max / min ≤ 2 and (max - min) ≤ 100, where max and min refer respectively to the maximum and minimum expression levels of a particular gene across a tissue sample. This left us with $p = 1896$ genes. In addition, the natural logarithm of the expression levels was taken.

The lymphoma dataset[3] contains the gene expression levels of the three most prevalent adult lymphoid malignancies: (1) 42 samples of diffuse large B-cell lymphoma (DLCL), (2) 9 samples of follicular lymphoma (FL), and (3) 11 samples of chronic lymphocytic leukaemia (CLL). The total sample size is $n = 62$ and $p = 4026$ genes. More information on these data may be found in (Alizadeh et al., 2000).

Khan dataset (Khan et al., 2001) contains 2308 genes and 83 observations, each from a child who was determined by clinicians to have a type of small round blue cell tumour. This includes the following four classes: neuroblastoma (N), rhabdomyosarcoma (R), Burkitt lymphoma (B) and the Ewing sarcoma (E). The numbers in each class are: 18 - N, 25 - R, 11 - B and 29 - E.

We applied double normalisation to the data. Firstly, we normalised each column to have means zero and unit standard deviations. Then, we applied the same normalisation to each row.

4 EXPERIMENTS

After decomposition of the original matrix **X** according to (1), we used the leave-one-out (LOO) classification scheme, applied to the matrix **B**. This means that we set aside the ith observation and fit the classifier by considering remaining $(n-1)$ data points. The experimental procedure has heuristic nature and its performance depends essentially on the initial settings (see, also, (Brunet et al., 2004)). Let us denote a classification scheme by

$$SCH(nrs, Model, LOO), \qquad (6)$$

where we conducted nrs identical experiments with randomly generated initial settings. Any particular

[2] http://www.broad.mit.edu/cgi-bin/cancer/publications/

[3] http://llmpp.nih.gov/lymphoma/data/figure1

Table 1: Some selected experimental results, where "*NM*" is the number of mis-classified samples; "*ls*", "*SVM*" and "*MLR*" indicate "*lscov*" function in Matlab, linear SVM and multinomial logistic regression.

Data	Model	k	AMR	NM	m	AUC
Colon	ls	5	0.1129	7	1	0.8823
Colon	SVM	5	**0.0968**	6	1	0.8818
Leukaemia	SVM	3	0.0139	1	1	0.9916
Leukaemia	ls	4	0.0139	1	1	0.9957
Leukaemia	SVM	35	**0.0**	0	1	1.0
Lymphoma	MLR	12	0.0322	2	2	-
Khan	MLR	21	0.0241	2	3	-

experiment includes two steps: 1) dimensionality reduction with Algorithm 1; 2) LOO evaluation with classification Model. In most of our experiments with the scheme (6) we used $nrs = 20$. We conducted experiments with three different classifiers: 1) Matlab-based *lscov* function, 2) linear *SVM*, and 3) we used *MLR* in application to the lymphoma and Khan sets. These classifiers are denoted by "Model" in (6).

We used two evaluation criteria: 1) LOO misclassification rate (LMR) and 2) area under receiver operation curve (AUC), where the last one was used only in application to colon and leukaemia set with binary labels.

By definition, $LMR = \frac{1}{m}\sum_{j=1}^{m} I\{q_j \neq y_j\}$, where q_j is the prediction for the label y_j, I is an indicator function.

Remark 2. Figures 1(b-c) illustrate average LMRs as a function of numbers of metagenes. As it may be expected, results corresponding to the small number of metagenes are poor because of over-smoothing. Then, we have some improvement to some point. After, that point the model suffers from overfitting. This property may be used for the selection of the number k of metagenes.

Table 1 represents some best results. It can be seen that our results are competitive with those in (Dettling and Buhlmann, 2003), (Peng, 2006), where the best reported result for the colon set is LMR = 0.113, and LMR = 0.0139 for the leukaemia set.

The appearance of the images in Figure 2 is very logical. We sorted the tissues in order to consider visual differences between the patterns. In the case of the colon data, Figure 2(a), we cannot see clear separation of the negative and positive classes. In contrast, in the case of leukaemia, Figure 2(b), metagene N2 (from the left) separates top the 42 tissues from the remaining 25 tissues with only one exception. It is tissue N58 (from the top) -the only one mis-classified tissue in Table 1 (cases $k = 3,4$). Similarly, in the case of lymphoma, Figure 2(c), metagene N1 (from the left) separates clearly CLL from the two remaining classes. Further, metagene N3 separates DLCL from the two remaining classes.

Remark 3. In the Fig. 1(b-c) we have plotted the average LMRs estimated using LOO cross-validation under assumption that the matrix factorization will be the same during the n validation trials as chosen on the basis of the full data set. However, there will be a selection bias in these estimates as the matrix factorization should be reformed as a natural part of any validation trial; see, for example, (Ambroise and McLachlan, 2002). But, since the labels y_t of the training data were not used in the factoring process, the selection bias should not be of a practical importance.

4.1 Computation Time

A Linux computer with speed 3.2GHz, RAM 16GB, was used for most of the computations. The time for 300 global iterations with Algorithm 1 (used special code written in C) in the case of $k = 11$ was between 10 and 15 sec. Based on our experiments, 20 global iterations with non-negative matrix factorization (Lee and Seung, 2000) for the same task as above requires about 25 min.

5 CONCLUSIONS

Microarray data analysis is challenging the traditional machine learning techniques due to the availability of a limited number of training instances and the existence of large number of genes, together with the inherent various uncertainties. In many cases machine learning techniques rely too much on the gene selection, which may cause selection bias. Generally, feature selection may be classified into two categories based on whether the criterion depends on the learning algorithm used to construct the prediction rule. If the criterion is independent of the prediction rule, the method is said to follow a filter approach, and if the criterion depends on the rule, the method is said to follow a wrapper approach (Ambroise and McLachlan, 2002). The objective of this study is to develop a filtering machine learning approach and produce a robust classification for microarray data.

Based on our experiments, the proposed matrix factorisation performed an effective dimensional reduction as a preparation step for the following supervised classification. Classifiers built in metagene, rather than original gene, space are more robust and reproducible because the projection can reduce noise more than simple normalisation. Algorithm 1, as a main contribution of this paper, is conceptually simple. Consequently, it is much faster compared to popular NMF. Stability of the algorithm depends essentially on the properly selected learning rate, which must not be too big. We can include additional functions so that the learning rate will be reduced or increased depending on the current performance.

There are many advantages to such a metagene approach. By capturing the major, invariant biological features and reducing noise, metagenes provide descriptions of data sets that allow them to be more easily combined and compared. In addition, interpretation of the metagenes, which characterize a subtype or subset of samples, can give us insight into underlying mechanisms and processes of a disease.

The results that we obtained on three real datasets confirm the potential of our approach.

REFERENCES

Alizadeh, A., Eisen, M., Davis, R., Ma, C., Lossos, I., Rosenwald, A., Boldrick, J., Sabet, H., Tran, T., and Yu, X. (2000). Distinct types of diffuse large b-cell-lymphoma identified by gene expression profiling. *Nature*, 403:503–511.

Ambroise, C. and McLachlan, G. (2002). Selection bias in gene extraction on the basis of microarray gene expression data. *Proceedings of the National Academy of Sciences USA*, 99:6562–6566.

Bohning, D. (1992). Multinomial logistic regression algorithm. *Ann. Inst. Statist. Math.*, 44(1):197–200.

Brunet, J., Tamayo, P., Golub, T., and Mesirov, J. (2004). Metagenes and molecular pattern discovery using matrix factorisation. *Proceedings of the National Academy of Sciences USA*, 101(12):4164–4169.

Dettling, M. and Buhlmann, P. (2003). Boosting for tumor classification with gene expression data. *Bioinformatics*, 19(9):1061–1069.

Dudoit, S., Fridlyand, J., and Speed, I. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of Americal Statistical Association*, 97(457):77–87.

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422.

Khan, J., Wei, J., Ringner, M., Saal, L., Ladanyi, M., Westermann, F., Berthold, F., and Schwab, M. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 7(6):673–679.

Koren, Y. (2009). Collaborative filtering with temporal dynamics. In *KDD*, pages 447–455.

Lee, D. and Seung, H. (2000). Algorithms for non-negative matrix factorisation. In *Advances in Neural Information Processing Systems*.

Mol, C., Mosci, S., Traskine, M., and Verri, A. (2009). A regularised method for selecting nested groups of relevant genes from microarray data. *Journal of Computational Biology*, 16(5):677–690.

Peng, H. and Ding, C. (2005). Minimum redundancy and maximum relevance feature selection and recent advances in cancer classification. In *SIAM workshop on feature selection for data mining*, pages 52–59.

Peng, Y. (2006). A novel ensemble machine learning for robust microarray data classification. *Computers in Biology and Medicine*, 36:553–573.

Tamayo, P., Scanfeld, D., Ebert, B., Gillette, M., Roberts, C., and Mesirov, J. (2007). Metagene projection for cross-platform, cross-species characterization of global transcriptional states. *Proceedings of the National Academy of Sciences USA*, 104(14):5959–5964.

Zhang, X., Lu, X., Shi, Q., Xu, X., Leung, H., Harris, L., Iglehart, J., Miron, A., and Wong, W. (2006). Recursive svm feature selection and sample classification for mass-spectrometry and microarray data. *BMC Bioinformatics*, 7(197).