Research

New feature subset selection procedures for classification of expression profiles

Trond Hellem Bø and Inge Jonassen

Address: Department of Informatics, University of Bergen, N-5020 Bergen, Norway.

Correspondence: Trond Hellem Bø. E-mail: trondb@ii.uib.no

Published: 14 March 2002

Genome ***Biology*** 2002, **3(4)**:research0017.1–0017.11

The electronic version of this article is the complete one and can be found online at http://genomebiology.com/2002/3/4/research/0017

(Print ISSN 1465-6906; Online ISSN 1465-6914)

Received: 5 December 2001
Revised: 30 January 2002
Accepted: 8 February 2002

Abstract

Background: Methods for extracting useful information from the datasets produced by microarray experiments are at present of much interest. Here we present new methods for finding gene sets that are well suited for distinguishing experiment classes, such as healthy versus diseased tissues. Our methods are based on evaluating genes in pairs and evaluating how well a pair in combination distinguishes two experiment classes. We tested the ability of our pair-based methods to select gene sets that generalize the differences between experiment classes and compared the performance relative to two standard methods. To assess the ability to generalize class differences, we studied how well the gene sets we select are suited for learning a classifier.

Results: We show that the gene sets selected by our methods outperform the standard methods, in some cases by a large margin, in terms of cross-validation prediction accuracy of the learned classifier. We show that on two public datasets, accurate diagnoses can be made using only 15-30 genes. Our results have implications for how to select marker genes and how many gene measurements are needed for diagnostic purposes.

Conclusion: When looking for differential expression between experiment classes, it may not be sufficient to look at each gene in a separate universe. Evaluating combinations of genes reveals interesting information that will not be discovered otherwise. Our results show that class prediction can be improved by taking advantage of this extra information.

Background

Microarrays can be used to obtain simultaneous measures of transcript abundance for thousands of genes. A number of projects have applied this technology to study differences between diseased and healthy tissue (for example [1]) and differences between different types and subtypes of diseases (for example [2-4]). The aim is to help improve the understanding of the diseases at a molecular level and to develop new diagnostic and prognostic tools. Microarray experiments can also help identify marker genes.

Typically, such experiments have taken on the order of 50-100 samples of different patients and used microarrays to measure the abundance (or relative abundance) of 5,000-10,000 genes. In some cases the samples are labeled with information about disease (for example, healthy/diseased, disease type). In other cases, such labels are not given and the aim can be to discover groupings of the samples, and clustering and class-discovery methods can be applied. When labels are given, the aim is to find sets of genes that distinguish well between samples with different labels. The

identity of such genes can help understand disease mechanisms. Classifiers employing expression data for the identified genes can be used for diagnostic and prognostic applications, that is, class prediction.

Finding sets of genes with expression values that allow class separation may be achieved by the use of supervised or unsupervised methods. The most common practice is to apply a direct approach, where the class labels of the samples are used in the search for separating genes (supervised). However, there exist methods where partitions of the samples are found in an unsupervised manner together with gene sets that support these partitions, for instance by clustering [5]. If any of the partitions correlate sufficiently with known class labels, the gene set that supports this partitioning may be reported as relevant for separating the sample classes.

Developing class predictors is in machine-learning terminology a 'supervised learning problem'. The expression data constitute a training set of labeled examples where each example is the expression profile for one sample. The aim is to develop a model that correctly predicts the labels of new examples. In order to test models it is standard practice to not include all examples in the training set so that the resulting models can be tested on the examples not used to build the model (the test set).

The models to be built will use some features of the examples. One problem with gene expression data is that each example has too many features, and many of them are noisy and irrelevant for the learning problem at hand. This is a common problem in machine learning and pattern recognition and a number of approaches have been proposed to select a subset of the features to be used. The problem of finding the best subset is commonly referred to as the feature subset selection (FSS) problem.

One approach is to consider each feature individually and see whether it distinguishes examples with different class labels. Let us restrict ourselves to the problem where we have only two classes (for example, healthy/diseased). A perfect feature would, for example, have high values for class 1 and low values for class 2, and could be used by itself to predict class membership of new examples. Such features rarely exist, and more commonly one needs to find a set of features that makes it possible to make a decision boundary that separates most class 1 examples from most class 2 examples.

A few groups have published results obtained using different FSS procedures on microarray data. These methods evaluate each feature (gene) with respect to how well it distinguishes between class 1 and class 2. Then they rank all genes according to the result and select the top K genes as the feature subset to be used. Some also employ a method to remove redundancy in the selected gene set; for example, some genes may behave very similarly and effectively provide the same information [6]. Other groups have reduced the dimensionality (number of features) by singular value decomposition (SVD), also referred to as principal component analysis (PCA), and used, for example, the first ten principal components as the feature subset [4,7].

The methods considering each gene separately potentially miss sets of genes that together allow good separation between the classes while each of the genes individually does not. We interpreted the results obtained by Xiong *et al.* [7] to indicate that this may indeed be the case. They obtained significantly higher accuracy using the seven first principal components than using 42 genes that each separate well between the classes.

Here we investigate new FSS methods that analyze pairs of genes, making it possible to find pairs that distinguish well between sample classes. Additionally we investigate the so-called forward selection method for FSS, where a good feature set is constructed by a 'greedy' selection procedure [8]. The results of these procedures are compared to results for previously reported FSS methods and the results show that our new FSS method gives more stable and better classification accuracy than methods evaluating each gene separately. The prediction accuracies achieved with our pair-based methods are also comparable to the best results reported in other papers.

We also list the genes chosen by our FSS methods and study the differences compared with sets chosen by other methods. Furthermore, we seek an explanation for the difference in classification accuracy achieved.

Finally, we discuss the implications of our findings. Apparently our FSS procedure provides an approach to finding gene sets that allows good separation between different classes and reveals better prediction results than other methods. Further research in this direction is required - considering wrapped feature selection systems, for example. The importance of considering combinations of genes in the feature selection process may contribute to new approaches to understanding diseases. Additionally, the importance of gene combinations may inspire new ways of designing microarray experiments for diagnostic and prognostic purposes.

Results

To evaluate our methods we applied them to two public datasets ([9,10]; see also Materials and methods) and used two standard methods for comparison. The comparison is indirect, meaning that we use the average error rate of a learned classifier as a quality measure on the feature selection procedure. Here we describe our methods and the results we have achieved using them. We also show that our methods reveal better results than two standard methods.

We apply a novel FSS procedure for ranking genes based on relevance for separating two experiment classes. Rather than evaluating each gene independently of the other genes, we consider gene pairs. Each gene pair is evaluated by how well it separates two classes, assigning a separation score to the pair. For a comparison study we use a greedy method from machine learning called 'forward selection' and a gene-ranking method based on evaluating each gene separately. We will subsequently refer to the latter as 'individual ranking'.

Gene pair ranking

We propose a filter method for evaluating pairs of genes by how well they separate two classes. We give each pair of genes a score reflecting how well the pair in combination distinguishes two experiment classes. Figure 1 shows an example of a pair of genes separating two classes well, along with the diagonal linear discriminant (DLD) axis and the decision boundary given by the axis (for an introduction to DLD, see [11]). We evaluate a gene pair by computing the projected coordinates of each experiment on the DLD axis using only these two genes. We then take the two sample *t*-statistic on the projected points as the pair score. We propose two alternative methods for selecting a feature subset based on pair scores, one exhaustive method called 'all pairs' and one faster method called 'greedy pairs'.

The all-pairs procedure considers all pairs of genes. Given pair scores for all pairs we select the top-ranked disjoint pairs in a greedy manner. First, the pair with highest pair score is selected, then all pairs containing any of these two genes are removed from the list. Then the highest-scoring pair from the remaining list is chosen, and so on.

By removing the already selected genes from the gene set, we do not take the risk that one exceptionally high-scoring gene can drag along several 'bad' companion genes. If such a high-scoring gene was left in the gene set, it would probably be responsible for many of the top-ranked pairs. By removing selected genes from the gene set, a high-scoring gene will only cause its best available companion to join it in the set of selected genes.

As an alternative, less computationally expensive, method we try an approach evaluating only a subset of all gene pairs (greedy pairs). The greedy-pairs approach first ranks all genes on the basis of individual *t*-score. Subsequently, this procedure first selects the best gene g_i ranked by gene *t*-score, then finds the gene g_j that together with g_i maximizes the pair *t*-score. These two genes are then removed from the gene set and the procedure is repeated on the remaining set until we have selected the desired number of genes. This approach is computationally much faster than all pairs, as only a subset of all gene pairs will be evaluated, but it may miss some pairs with high score.

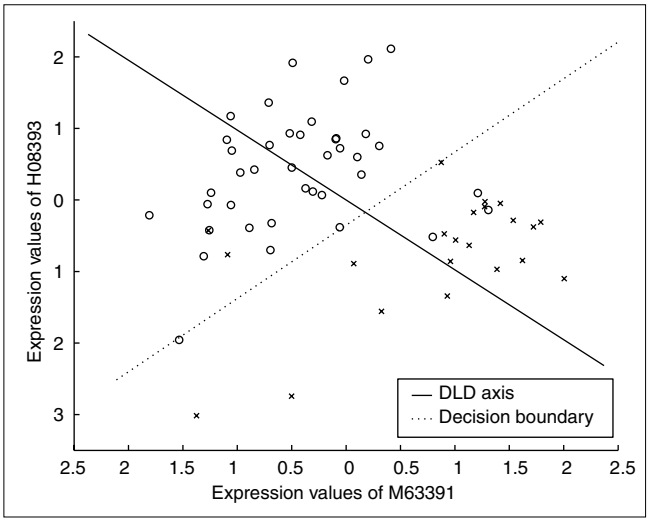

Figure 1
Example of a good pair of genes in the colon dataset. The expression values give almost full separation between normal and tumor tissues. Along the *x*-axis (horizontal) are the expression values of M63391, and along the *y*-axis (vertical) are the expression values of H08393. The points marked 'x' are normal tissues, and the tumor tissues are marked by an 'o'. The expression values have been through the preprocessing steps described in the text. Also plotted is the DLD axis and the class-decision boundary for these two genes. Note that the DLD axis and the decision boundary are orthogonal, but as a result of different scaling on the axes it does not appear so in the plot.

Prediction accuracy results

For testing our methods we used two public datasets, one representing colon (tumor/normal) tissues [10], and the other representing acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML) [9] (see also Materials and methods). For both these datasets we perform cross-validation tests on the two-class problem, distinguishing between ALL and AML in the ALL/AML dataset and normal and tumor in the colon dataset. By dividing the data into a training set and a test set several times, we compare the average performance of different prediction methods on the test set using four different feature subset selection (FSS) procedures on the training set. In this study we use two linear discriminant methods: diagonal linear discriminant (DLD) and Fisher's linear discriminant (FLD), and one local method; *k* nearest neighbors (*k*NN) (for *k*NN and FLD, see [12]). The FSS procedures we use are all pairs, greedy pairs, forward selection and individual ranking. The application of several prediction methods is to see if the differences between the FSS procedures are specific to a particular prediction method rather than more general. Instead of comparing the different prediction methods we compare the ability of the different FSS procedures to find feature subsets that generalize the class differences. A comparison is done for feature subsets of size 2, 4, 6,..., *m*, where *m* is the number of experiments in the dataset. We also leave out different portions

from the training data (1, 33%, 50%), to see what effect this has on prediction accuracy.

The results are summed up in plots like the one shown in Figure 2, with the number of genes along the *x*-axis and the prediction accuracy along the *y*-axis. The curves in Figure 2 show performance using the different FSS procedures and DLD prediction. Plots in Figure 2a,b illustrate prediction performance on the colon dataset when one example is left out at each round (often referred to as leave-one-out cross-validation or LOOCV; Figure 2a) and when leaving out 31 examples (50%; Figure 2b). Plots in Figure 2c,d show prediction performance for the ALL/AML dataset leaving out one (Figure 2c) and 36 (50%; Figure 2d) of the examples from the training set. The complete set of plots, showing results for all four FSS procedures and all three prediction methods, are available as Additional data files.

For the colon dataset, the pair-based methods achieve much better results than individual ranking and forward selection. When leaving out only one example from the training set there is no clear difference in prediction accuracy, except that forward selection gives worse results than the other methods (Figure 2a). Leaving out larger portions of the data,

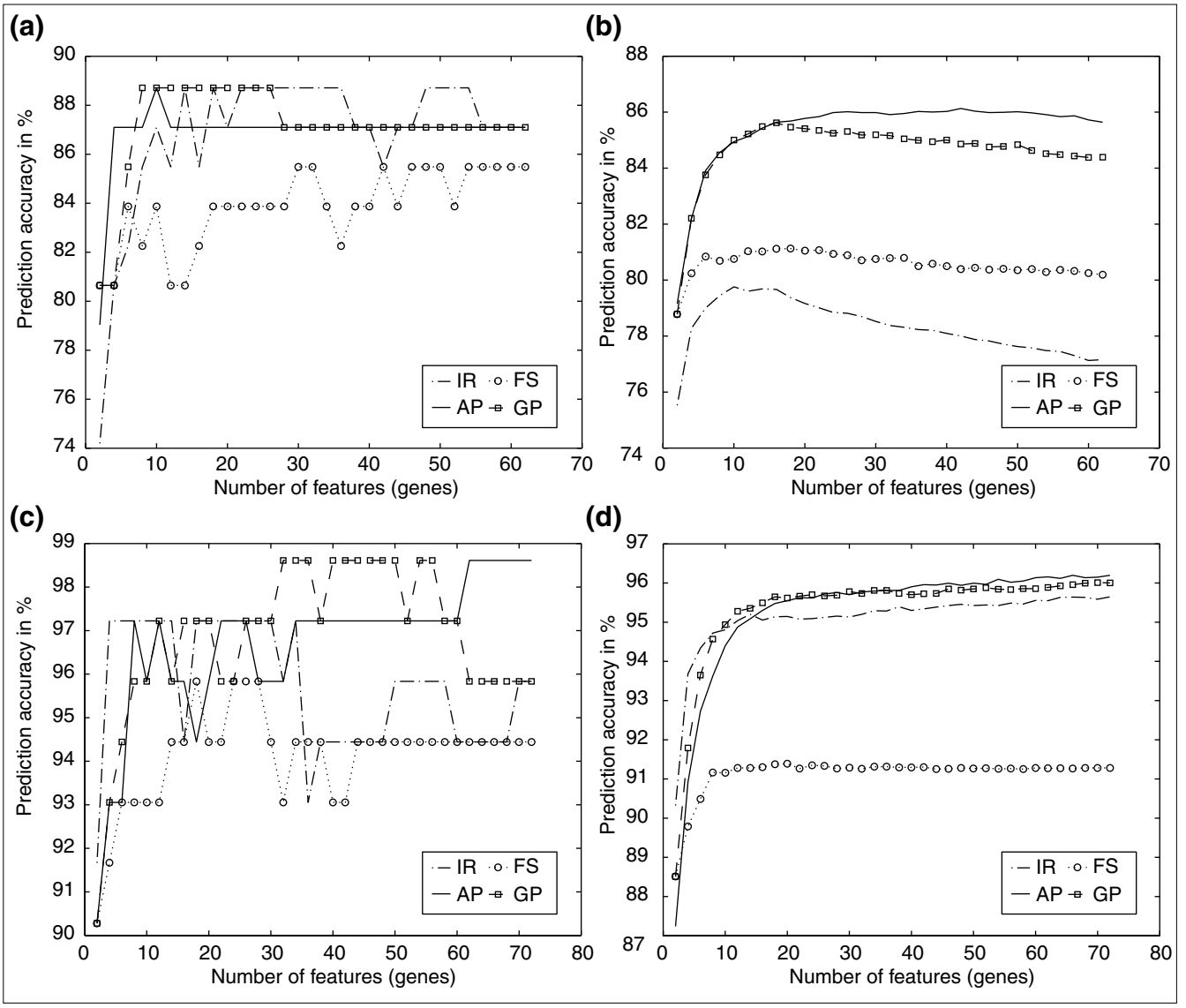

Figure 2
Plots of prediction accuracy on the colon and ALL/AML dataset using four different FSS procedures and DLD prediction. **(a)** Colon dataset: LOOCV and DLD prediction. **(b)** Colon dataset: L-31-OCV and DLD prediction. **(c)** ALL/AML dataset: LOOCV and DLD prediction. **(d)** ALL/AML dataset: L-36-OCV and DLD prediction. Along the *x*-axis are the number of genes in the feature subsets, and average prediction accuracy is shown along the *y*-axis. The FSS procedures individual ranking (IR), all pairs (AP), forward selection (FS) and greedy pairs (GP) are explained in the text.

the pair-based methods give superior prediction accuracy, compared to both individual ranking and forward selection (Figure 2b). For this dataset, the all-pairs ranking gives slightly better results than the greedy-pairs ranked genes.

Using FLD and 3NN (*k*NN where *k*=3) prediction, the tendency is also that pair-based methods give the best prediction results, so the good results achieved with the pair-based methods are not just specific for DLD prediction. Comparing our results to those of Xiong *et al.* [7], we achieve results comparable to what they achieved using the first seven principal components and FLD prediction. Using FLD prediction and the 20 top-ranked genes, selected by all-pairs ranking, we achieve a prediction accuracy of 87.8% leaving out 32% (20 examples) and 85.9% leaving out 50% (31 examples) from the training data. The corresponding results achieved by Xiong *et al.* were 87.0% and 85.7%. Weston *et al.* [13] report an average error rate of 12.8% on the colon data when leaving out 12 examples at each cross-validation iteration using the top 15 genes from their own feature selection method and a support vector machine [14] for prediction. Using the top 14 genes from all-pairs ranking and DLD prediction we achieve an error rate of 12.2%.

For the ALL/AML dataset there is smaller difference in results using the pair-based methods (all pairs, greedy pairs) and individual ranking, but it is still in favor of the pair based methods. On this dataset, forward selection gives prediction results far worse than the other methods. When leaving out 1, 24 (33%) or 36 (50%) of the examples from the training set, the pair-based methods give slightly better results compared to individual ranking using all three prediction methods. Generally the prediction accuracy rises a bit faster with the number of features using ranking by gene *t*-score than using pair-based ranking. However, the pair-based methods give slightly better prediction accuracy when the number of genes in the feature subset increases. In all cases except one (LOOCV and FLD prediction), all pairs or greedy pairs has the best maximum prediction accuracy. For LOOCV and FLD prediction, individual ranking, all pairs and greedy pairs had the same maximum prediction accuracy.

Selected gene sets

Given the all-pairs ranking method, we study which genes show up as highly relevant in pairs. For this study we use the data for all experiments in each of the colon and ALL/AML datasets. We ranked all pairs of genes for each dataset and list the top-ranked 25 disjoint pairs in Tables 1 and 2.

For the colon dataset it is interesting that, taking all the available data into account, the top-ranked gene pairs contain several genes that are not among the top genes when ranked individually. In fact, only 24 out of the top 50 genes ranked by pair *t*-score are among the top 50 genes ranked by gene *t*-score. Only 32 out of 50 genes are in the top 100 ranked by gene *t*-score.

For the ALL/AML dataset, 31 out of 50 genes, ranked by pair *t*-score, are among the top 50 ranked by gene *t*-score. Forty-two out of 50 were among the top 100 ranked by gene *t*-score. The difference from the colon data set was that the all-pairs ranking method selects far more genes not from the top individually ranked genes. This could account for why we do not see a larger difference in prediction accuracies for this dataset.

Discussion

The recurring question when working with diseases and gene-expression profiles is which genes are involved and which genes are suitable as marker genes. The focus of this paper is on methods for finding gene sets that are suitable for discriminating between experiment classes, like disease/normal or between different subclasses of a disease.

We propose a conditional gene relevance measure, the pair *t*-score, where genes are scored by how well they separate experiment classes in the context of some other gene. This approach may inspire some insight into the biological mechanisms behind a disease as we consider genes in combination with others. We believe our approach is a step in the right direction to find the genes that are interesting for separating experiment classes but do not show up as interesting on an individual basis. Considering combinations of genes in the feature selection process may contribute to new approaches to understanding disease. Additionally, the importance of gene combinations may inspire new ways of designing microarray experiments for diagnostic and prognostic purposes. Once a set of marker genes has been identified, this makes cheap production of diagnostic microarrays possible, as such a microarray needs only a relatively small number of marker genes spotted on it. Alternatively, other methods of measuring gene expression can be used, as the number of genes to be monitored is rather small. For the ALL/AML and colon datasets we demonstrate that quite accurate diagnoses can be achieved using only the gene-expression levels of 20-30 and 15-20 genes, respectively.

We do not claim that our pair-based methods will find all interesting genes, as there may be relevant genes that are significant by themselves but may not appear in any of the high-scoring pairs. However, we demonstrate that looking at gene pairs will reveal some extra information about class differences. Our methods are not meant as a substitute for single gene evaluation, but rather as a supplement to already existing methods.

An interesting point regarding cross-validation is that when leaving one example out during training, the performance curves shows a different picture from the one we get when leaving out larger portions of the data. Leaving out larger portions from the training data gives smoother performance curves, showing a clearer picture of differences in prediction performance. It is also interesting to study how much the

Table 1

Top-ranked 50 genes (25 pairs) for ALL/AML class separation using AP (all pairs) ranking

Pair rank	Gene ID	Pair *t*-score	Gene *t*-score	Gene rank	Gene annotation
1	M84526_at	16.16	12.88	1	DF D component of complement (adipsin)
1	M92287_at	16.16	8.87	6	CCND3 Cyclin D3
2	M23197_at	15.43	11.72	2	CD33 CD33 antigen (differentiation antigen)
2	M31523_at	15.43	11.0	3	TCF3 Transcription factor 3 (E2A immunoglobulin enhancer binding factors E12/E47)
3	U46499_at	13.56	9.14	5	Glutathione S-transferase, microsomal
3	M31303_rna1_at	13.56	6.52	36	Oncoprotein 18 (Op18) gene
4	M63138_at	13.53	9.46	4	CTSD Cathepsin D (lysosomal aspartyl protease)
4	HG1612-HT1612_at	13.53	8.47	10	Macmarcks
5	X62320_at	12.58	8.38	11	GRN Granulin
5	Z14982_rna1_at	12.58	5.54	93	MHC-encoded proteasome subunit gene LAMP7-E1 (LMP7)
6	M31211_s_at	12.41	8.61	9	MYL1 Myosin light chain (alkali)
6	X62654_rna1_at	12.41	7.32	17	ME491 gene extracted from *H. sapiens* gene for Me491/CD63 antigen
7	M27891_at	12.15	8.83	7	CST3 Cystatin C (amyloid angiopathy and cerebral hemorrhage)
7	U89922_s_at	12.15	5.24	115	LTB Lymphotoxin-beta
8	X59417_at	12.14	7.99	12	Proteasome iota chain
8	X52056_at	12.14	6.52	37	SPI1 Spleen focus forming virus (SFFV) proviral integration oncogene spi1
9	M19507_at	12.09	7.03	24	MPO Myeloperoxidase
9	M89957_at	12.09	6.92	28	IGB Immunoglobulin-associated beta (B29)
10	M84371_rna1_s_at	11.9	7.18	23	CD19 gene
10	U16954_at	11.9	6.43	40	(AF1q) mRNA
11	M63379_at	11.74	7.56	13	CLU Clusterin (complement lysis inhibitor; testosterone-repressed prostate message 2; apolipoprotein J)
11	M83667_rna1_s_at	11.74	7.52	14	NF-IL6-beta protein mRNA
12	M16038_at	11.72	7.31	18	LYN V-yes-1 Yamaguchi sarcoma viral related oncogene homolog
12	Y08612_at	11.72	6.31	47	Rabaptin-5 protein
13	D88422_at	11.68	8.72	8	Cystatin A
13	M11722_at	11.68	7.18	21	Terminal transferase mRNA
14	X66401_cds1_at	11.67	5.89	69	LMP2 gene extracted from *H. sapiens* genes TAP1, TAP2, LMP2, LMP7 and DOB
14	Y00433_at	11.67	4.72	182	GPX1 Glutathione peroxidase 1
15	M63959_at	11.61	6.59	34	LRPAP1 Low density lipoprotein-related protein-associated protein 1 (alpha-2-macroglobulin receptor-associated protein 1)
15	X51521_at	11.61	6.44	38	VIL2 Villin 2 (ezrin)
16	Z15115_at	11.37	7.49	15	TOP2B Topoisomerase (DNA) II beta (180 kDa)
16	U10868_at	11.37	5.55	92	ALDH7 Aldehyde dehydrogenase 7
17	Y12670_at	11.25	5.67	82	LEPR Leptin receptor
17	U77948_at	11.25	5.6	87	KAI1 Kangai 1 (suppression of tumorigenicity 6, prostate; CD82 antigen (R2 leukocyte antigen detected by monoclonal antibody IA4))
18	U46751_at	11.2	5.8	73	Phosphotyrosine independent ligand p62 for the Lck SH2 domain mRNA
18	L06797_s_at	11.2	5.77	74	Probable G protein-coupled receptor LCR1 homolog
19	M95678_at	11.19	5.73	77	PLCB2 Phospholipase C, beta 2
19	U72936_s_at	11.19	5.38	101	X-linked helicase II
20	S76617_at	11.16	6.43	41	BLK Protein-tyrosine kinase blk
20	L09209_s_at	11.16	5.9	68	APLP2 Amyloid beta (A4) precursor-like protein 2
21	M55150_at	11.11	6.7	32	FAH Fumarylacetoacetate
21	M96803_at	11.11	4.74	180	SPTBN1 Spectrin, beta, non-erythrocytic 1
22	X17042_at	11.08	6.93	27	PRG1 Proteoglycan 1, secretory granule
22	X99920_at	11.08	5.15	124	S100 calcium-binding protein A13
23	S50223_at	10.86	6.65	33	HKR-T1
23	U82759_at	10.86	4.39	229	GB DEF Homeodomain protein HoxA9 mRNA
24	J03589_at	10.83	5.62	85	Ubiquitin-like protein GDX
24	X12447_at	10.83	5.16	122	ALDOA Aldolase A
25	X74262_at	10.8	5.89	70	Retinoblastoma binding protein p48
25	L19437_at	10.8	5.13	127	TALDO Transaldolase

Table 2

Top-ranked 50 genes (25 pairs) for colon tumor/normal class separation using AP (all pairs) ranking

Pair rank	Gene ID	Pair *t*-score	Gene *t*-score	Gene rank	Gene annotation
1	M63391	10.02	5.57	2	Human desmin gene, complete cds
1	H08393	10.02	5.47	4	Collagen alpha 2(XI) chain (*H. sapiens*)
2	X12671	9.85	5.37	5	Human gene for heterogeneous nuclear ribonucleoprotein (hnRNP) core protein A1
2	Z50753	9.85	5.09	9	*H. sapiens* mRNA for GCAP-II/uroguanylin precursor
3	R87126	9.58	6.37	1	Myosin heavy chain, nonmuscle (*Gallus gallus*)
3	X63629	9.58	4.86	12	*H. sapiens* mRNA for p cadherin
4	M36634	9.27	4.65	15	Human vasoactive intestinal peptide (VIP) mRNA, complete cds
4	H11084	9.27	3.55	65	Vascular endothelial growth factor (*Cavia porcellus*)
5	J05032	8.96	5.2	8	Human aspartyl-tRNA synthetase alpha-2 subunit mRNA, complete cds
5	U19969	8.96	3.05	132	Human two-handed zinc finger protein ZEB mRNA, partial cds
6	J02854	8.94	5.28	7	Myosin regulatory light chain 2, smooth muscle isoform (human) (contains element TAR1 repetitive element)
6	R54097	8.94	3.93	46	Translational initiation factor 2 beta subunit (human)
7	H06524	8.89	4.18	29	Gelsolin precursor, plasma (human)
7	U22055	8.89	3.77	51	Human 100 kDa coactivator mRNA, complete cds
8	M76378	8.74	4.81	13	Human cysteine-rich protein (CRP) gene, exons 5 and 6
8	T62947	8.74	4.12	34	60S ribosomal protein L24 (*Arabidopsis thaliana*)
9	D21261	8.67	3.46	76	SM22-alpha homolog (human)
9	H20709	8.67	2.69	203	Myosin light chain alkali, smooth-muscle isoform (human)
10	X86693	8.64	4.16	30	*H. sapiens* mRNA for hevin like protein
10	D14812	8.64	2.66	211	Human mRNA for ORF, complete cds
11	H09719	8.35	2.57	237	Tubulin alpha-6 chain (*Mus musculus*)
11	L07648	8.35	2.31	321	Human MX11 mRNA, complete cds
12	X12369	8.25	3.27	97	Tropomyosin alpha chain, smooth muscle (human)
12	R98842	8.25	3.05	131	Prothymosin alpha (*H. sapiens*)
13	J04102	8.11	3.06	128	Human erythroblastosis virus oncogene homolog 2 (ets-2) mRNA, complete cds
13	U14631	8.11	2.84	164	Human 11 beta-hydroxysteroid dehydrogenase type II mRNA, complete cds
14	T63133	8.06	2.85	160	Thymosin beta-10 (human)
14	T61661	8.06	2.51	255	Profilin I (human)
15	T92451	8.06	4.12	33	Tropomyosin, fibroblast and epithelial muscle-type (human)
15	U09587	8.06	3.46	74	Human glycyl-tRNA synthetase mRNA, complete cds
16	T71025	8.0	4.34	24	Human
16	L11706	8.0	3.18	104	Human hormone-sensitive lipase (LIPE) gene, complete cds
17	Z48541	7.96	3.14	120	*H. sapiens* mRNA for protein tyrosine phosphatase
17	D25217	7.96	2.54	249	Human mRNA (KIAA0027) for ORF, partial cds
18	M76378	7.94	5.04	10	Human cysteine-rich protein (CRP) gene, exons 5 and 6
18	T56604	7.94	3.89	48	Tubulin beta chain (*Haliotis discus*)
19	X54942	7.93	4.42	23	*H. sapiens* ckshs2 mRNA for Cks1 protein homolog
19	R44301	7.93	3.36	85	Mineralocorticoid receptor (*H. sapiens*)
20	T90280	7.86	3.52	70	Ribophorin II precursor (human)
20	T51534	7.86	3.01	137	Cystatin C precursor (human)
21	R96357	7.81	2.89	156	Polyadenylate-binding protein (*Xenopus laevis*)
21	R46753	7.81	2.64	216	Cyclin-dependent kinase inhibitor 1 (*H. sapiens*)
22	M76378	7.75	4.51	20	Human cysteine-rich protein (CRP) gene, exons 5 and 6
22	D00860	7.75	3.38	81	Ribose-phosphate pyrophosphokinase I (human)
23	X14958	7.56	4.53	19	Human hmgI mRNA for high mobility group protein Y
23	X87159	7.56	2.63	221	*H. sapiens* mRNA for beta subunit of epithelial amiloride-sensitive sodium channel
24	T51023	7.55	4.12	32	Heat shock protein HSP 90-beta (human)
24	D31716	7.55	2.83	168	Human mRNA for GC box binding protein, complete cds
25	M26383	7.52	5.53	3	Human monocyte-derived neutrophil-activating protein (MONAP) mRNA, complete cds
25	T47377	7.52	4.11	35	S-100P protein (human)

prediction accuracy drops when leaving out more than one example from the training set. The all-pairs and greedy-pairs methods are here shown to be more robust than individual ranking and forward selection in terms of loss of prediction accuracy, as shown in the plots in Figure 2.

Further work will include integrating our methods into the J-Express software package [15], to make them available to the biological community. Other methods for evaluating combinations of genes will be investigated, and we will consider other prediction methods, like support vector machines and artificial neural networks [12], testing whether better prediction accuracy can be achieved. We will also consider feature redundancy, which might lead to even smaller feature sets than we currently achieve. In these studies we will also include additional datasets.

Conclusions

We propose a new method for evaluating combinations of genes. Comparing the top-ranked genes ranked by gene-pair evaluation to the gene sets selected by two standard methods, we demonstrate that the gene sets selected by the pair-based approaches are more robust in terms of their ability to distinguish experiment classes. These results indicate that in evaluating each gene independently one risks overlooking some interesting genes. The improved prediction accuracy indicates that the genes we find might be at least as interesting to study as the top-ranked genes found by an independent evaluation of each gene. Looking at gene pairs, we find some genes that are not obviously good discriminators alone, but discriminate well when used in pairs with other genes. Thus, we discover some interesting genes that will not be discovered unless one looks at gene combinations.

We expect that some high-scoring pairs can occur just by chance because of the vast number of pairs evaluated. But if the pairs we select are high scoring by chance, we do not expect an improvement in prediction accuracy. Thus, by demonstrating a better ability to generalize class differences when using high-scoring pairs, we also show that these pairs give some extra information on the experiment classes.

Materials and methods

Datasets

For testing purposes we use the datasets published by Golub *et al.* [9] and Alon *et al.* [10]. The dataset published by Golub *et al.* [9] consists of 72 samples, of which 47 were acute lymphoblastic leukemia (ALL) and 25 samples were acute myeloid leukemia (AML). Gene-expression levels were measured using Affymetrix high-density oligonucleotide arrays containing 7,129 genes. The Alon *et al.* data [10] consists of 62 samples, of which 22 are normal and 40 are colon tumor tissue. Gene-expression levels were measured using Affymetrix oligonucleotide arrays complementary to more than 6,500 genes. Published along with the Alon *et al.* paper was a dataset [10] containing expression levels for the 2,000 genes with highest minimal intensity across the samples, and this is the dataset we study in this paper. We refer to the Golub *et al.* dataset [9] as 'ALL/AML' and the Alon *et al.* dataset [10] as 'colon'. Before analysis, we carried out the following preprocessing steps on both datasets: base 10 logarithmic transformation; and for each gene, subtract the mean and divide by the standard deviation.

Because the ALL/AML dataset contains a lot of negative intensity values, we first use the following preprocessing steps (similar to those proposed by Dudoit *et al.* [16]) on the ALL/AML dataset: thresholding with a floor of 1; and filtering by excluding genes with $max/min \leq 5$ and $(max - min) \leq 500$. This leaves us with a dataset of 3,934 genes.

Prediction methods

We use three prediction methods: one local method, *k* nearest neighbors (*k*NN), and two linear discriminant methods, diagonal linear discriminant (DLD) and Fisher's linear discriminant (FLD). We consider an expression profile $\mathbf{x} = (x_1, x_2, ..., x_n)$ to which we want to assign a class label (1 or 2). Formally we wish to estimate the function $f(\mathbf{x}) \rightarrow \{1,2\}$ based on the available training data with as small error rate as possible. There are many approaches to estimate *f*, making different assumptions about the distribution of the classes. *k*NN only considers the neighborhood around **x**, whereas DLD and FLD aim to find a best possible linear separating rule between classes based on all the available training data.

Many variants of the *k*NN algorithm exist. We choose to find the *k* nearest neighbors by Euclidean distance, using the selected feature subset, and let each of the *k* neighbors get a vote of weight 1. We then predict **x** to be of the class getting the majority of the votes. Thus if $k = 3$ and there were two examples of class 1 and one example of class 2 among the three nearest neighbors of **x**, we would predict that **x** belongs to class 1. *k*NN is a local method as it only considers the neighborhood of the experiment **x**, and does not consider the information in the rest of the training data. We use a fixed value of $k = 3$ in our tests.

DLD and FLD are two discriminant methods for which a discriminant axis **a** is computed on the basis of the available training data. The prediction using **a** is class 1 if

$$\mathbf{a}^T\left(\mathbf{x} - \frac{1}{2}(\mu_1 + \mu_2)\right) > 0 \quad (1)$$

and class 2 otherwise. The parameters μ_1 and μ_2 are here the mean expression profile of classes 1 and 2, respectively. The difference between DLD and FLD is in how the discriminant axis is computed. The DLD axis is computed as

$$\mathbf{a} = \mathbf{S}^{-1}(\mu_1 - \mu_2) \quad (2)$$

where **S** is the diagonal variance matrix whose elements are the common variance estimate

$$\sigma^2_{g_i} = \frac{(n_1 - 1)\sigma^2_{1,g_i} + (n_2 - 1)\sigma^2_{2,g_i}}{(n_1 + n_2 - 2)} \quad .$$

To compute the FLD axis, **S** is substituted by the covariance matrix **Σ** in equation 2. **S** and **Σ** will have the same values on the diagonal, while **Σ** will also contain covariances between genes.

When only using a subset $\mathbf{G} = \{g_{f1},...,g_{fl}\}$ of the genes ($l<n$), some modifications to the equations above have to be done. For DLD it is sufficient to define a feature subset matrix $\mathbf{F} = Diag(\{0, 1\}^n)$, where $\mathbf{F}_{ii} = 1$ if and only if $g_i \in \mathbf{G}$. Equation 1 can then be replaced with

$$\mathbf{a}^T \mathbf{F} \left(\mathbf{x} - \frac{1}{2} (\mu_1 + \mu_2)\right) > 0 \qquad (3)$$

This equation corresponds to setting every element $\mathbf{a}_i$ where $g_i \notin \mathbf{G}$ to 0.

For FLD the matrix $\mathbf{\Sigma}^{-1}$ will contain information on gene redundancy in the whole gene set. This is not the case for $\mathbf{S}^{-1}$, which is a diagonal matrix. It would therefore not be correct to do the same modifications to the equations as for DLD. For a feature subset of l genes, an l x l covariance matrix Σ' has to be computed, considering only the genes in **G**. Furthermore will the vectors $\mathbf{a}'$, μ'_1 and μ'_2 be l x 1 instead of n x 1 vectors. $\mathbf{x}'$ is now the vector containing only the expression values of the genes in **G**. The equations stay the same as before, except that **Σ**, **a**, μ_1 μ_2 and **x** are now substituted with $\mathbf{\Sigma}'$, $\mathbf{a}'$, μ'_1, μ'_2 and $\mathbf{x}'$.

Feature selection

When considering which feature (gene) subset to select for class prediction or for study in the wet lab, we need some method of eliminating the least interesting and highlight the most interesting before a choice is made. A natural choice is to rank the features after some relevance measure and then select some of the features with highest score for further use.

As the main focus of this paper is class prediction, we will discuss feature selection in this context. In the machine-learning literature, Kohavi and John [8] divide features into two main categories: irrelevant and relevant, of which the relevant genes can be separated into strongly relevant and weakly relevant. Strongly relevant features are the obvious ones, distinguishing well between classes independently of other features. Weakly relevant features, on the other hand, are not that obvious, and may not be relevant except in the context of other features. These features might be difficult to discover, as combinations of genes have to be evaluated in some manner. Furthermore there may be redundancy among the features. The machine-learning feature selection problem is to select the minimum optimal feature set, meaning that we want to find the smallest subset of features that gives optimal prediction accuracy. No algorithm exists for solving this problem except trying all feature subsets and choosing the best, which is not feasible if there are more than a trivial number of features. For microarrays, where the number of features is in the thousands, this is not an alternative at all. The approach normally used is filter methods or wrapper methods. Filter methods evaluate each feature independently of the prediction method to be used, whereas wrapper methods evaluate the feature set relative to the prediction method.

Standard methods

A large number of measures have been proposed for scoring genes, starting with Golub *et al.* [2] that proposed using $|\frac{\mu_1 - \mu_2}{\sigma_1 + \sigma_2}|$. Other gene measures in the literature include both non-parametric measures like the TNoM score of Ben-Dor *et al.* [17] and information gain (proposed by Xing *et al.* [6]), and parametric measures like *t*-score [7], Fisher score [13], naive Bayes global relevance [18] and between- to within-variance ratio [16]. These methods reward genes that allow (approximate) separation of experiment classes on the basis of the gene-expression levels. For instance, the measure proposed by Golub *et al.* [2] rewards genes where the mean expression levels in the two classes are far apart but at the same time the standard deviation in each class is small.

Instead of comparing different relevance measures, we choose to use *t*-score as our relevance measure, as the focus of this paper is on FSS methods rather than relevance measures. Given the DLD axis, we try the greedy feature-selection method called forward selection. The greedy forward selection first selects the best gene, meaning the gene with highest *t*-score, and in the subsequent steps adds the gene that leads to the highest two-sample *t*-statistic score. The *t*-score at step i is computed on the projected data points on the DLD-axis using only the expression values of the i selected genes.

Evaluating feature subset selection (FSS) performance

To evaluate the robustness of our pair-based approaches, we evaluate their performance compared to individual ranking and forward selection by cross-validation. To rank genes on an individual basis, we use the two-sample *t*-statistic on the expression values of the gene across the experiments.

Given a training set, we study how well the different FSS procedures do in finding feature subsets that generalize the differences between two experiment classes. Success is measured by how accurately a prediction method predicts the correct class labels of a set of labeled experiments (the test set) on the basis of the selected feature subset only. The FSS

procedures are compared by studying the performance of a prediction method in the context of which FSS method is used to select feature subsets. Thus, if we use the same prediction method, but we get varying prediction accuracy results using alternative FSS procedures, we can draw conclusions about how 'good' each of the FSS methods is. It is important to note that the training set and test set should have no common examples. Otherwise, the learning algorithm (either the FSS procedure or the prediction method) could take advantage of information on the examples of which it is going to predict the class labels during the learning process. The procedure for cross-validation is as follows.

(A) Repeat a number of times the following procedure:
   1. Partition the examples randomly into a training set and a test set, including **x**% of the examples in the training set;
   2. Use the FSS procedure on the training set;
   3. Train a classifier using the training set and only the selected features;
   4. Use the classifier to predict the class labels of each example in the test set and count the number of correct and false predictions.

(B) Output the percentage of correct predictions.

Using different FSS procedures in the feature selection step, we plot the performance of the three classifiers *k*NN, DLD and FLD for feature subsets of size 2, 4, 6, ..., m. The result is plots like the ones shown in Figure 2, showing a comparison of the prediction accuracy using the four different FSS procedures. We also perform tests for different values for x, typically 50% and 66%. In addition we do LOOCV tests.

Cross-validation procedure

The prediction accuracy estimate $\hat{p}$ is the average performance over all iterations performed in (A). Thus $\hat{p}$ is a random variable drawn from a distribution with mean p, the true prediction accuracy, and a standard deviation σ. The more iterations of the above cross-validation procedure used to compute $\hat{p}$ the smaller σ will become and the better $\hat{p}$ will estimate the true value of p. Empirical studies showed us that taking a randomly chosen test set at each iteration produced $\hat{p}$ curves that sometimes had large deviations from the true p curve, even when taking 62 iterations for the colon dataset as done by Xiong *et al.* [7]. We also tried an approximation to another cross-validation procedure called k-fold cross-validation. In k-fold cross-validation the dataset is split randomly into k equally large subsets. Then k iterations of cross-validation is carried out, at each round taking one of the subsets as the test set, and the rest of the data as the training set. As the datasets we used could not always be split into equally large subsets, that is, m is not a multiple of the training set size s, we used a modified version of the k-fold approach. We solve this by making the test sets one by one, randomly drawing s experiments out for a test set at each iteration. In addition, we keep a count on how many times each experiment has been left out so far. By not allowing any experiments to be left out, the i + 1th time before all experiments has been left out i times, we ensure that all experiments are left out the same number of times in the long run. Compared with the other approach, we found that using the approximate k-fold cross-validation, $\hat{p}$ converged much faster towards p, giving better estimates using the same number of iterations. Nonetheless, we chose to run $5m$ iterations using the approximate k-fold cross-validation approach, such that the prediction accuracy estimates should show a trustworthy comparison of the different FSS procedures.

Additional data files

Additional data files are available giving plots of average prediction accuracy performance, using three prediction methods and holding back different portions of the data from the training set. L-24-OCV stands for leave-24-out cross validation, and so on. Each curve shows the performance for feature sets of varying size selected by four feature selection methods. These files are also available at our website [19].

ALL/AML dataset plots using: 3NN prediction and LOOCV; 3NN prediction and L-24-OCV; 3NN prediction and L-36-OCV; DLD prediction and LOOCV; DLD prediction and L-24-OCV; DLD prediction and L-36-OCV; FLD prediction and LOOCV; FLD prediction and L-24-OCV; FLD prediction and L-36-OCV.

Colon dataset plots using: 3NN prediction and LOOCV; 3NN prediction and L-20-OCV; 3NN prediction and L-31-OCV; DLD prediction and LOOCV; DLD prediction and L-20-OCV; DLD prediction and L-31-OCV; FLD prediction and LOOCV; FLD prediction and L-20-OCV; FLD prediction and L-31-OCV.

Acknowledgements

I.J. has been supported by grants from the Norwegian research council. We thank Bjarte Dysvik and Karl-Henning Kalland for helpful discussion.

References

1. Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, Levine AJ: **Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays.** *Proc Natl Acad Sci USA* 1999, **96:**6745-6750.
2. Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeeck M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, *et al.*: **Molecular classification of cancer: class discovery and class prediction by gene expression monitoring.** *Science* 1999, **286:**531-537.
3. Alizadeh AA, Eisen MB, Davis RE, Ma C, Lossos IL, Rosenwald A, Boldrick JC, Sabet H, Tran T, Yu X, *et al.*: **Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling.** *Nature* 2000, **403:**503-511.
4. Khan J, Wei JS, Rigner M, Saal LH, Ladanyi M, Wetsermann F, Berthold F, Schwab M, Antonescu CR, Peterson C, *et al.*: **Classification and**

diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nat Med* 2001, **7:**673-679.
5. Getz G, Levine E, Domany E: **Coupled two-way clustering analysis of gene microarray data.** *Proc Natl Acad Sci USA* 2000, **97:**12097-12084.
6. Xing EP, Jordan MI, Karp R: **Feature selection for high-dimensional genomic microarray data.** In *Proceedings of Eighteenth International Conference on Machine Learning, 2001*. San Francisco: Morgan Kaufmann; 2001.
7. Xiong M, Jin L, Li W, Boerwinkle E: **Computational methods for gene expression-based tumor classification.** *BioTechniques* 2000, **29:**1264-1270.
8. Kohavi R, John GH: **Wrappers for feature subset selection.** *Artificial Intelligence* 1997, **97:**273-324.
9. **Supplementary datasets and prediction results for Golub *et al*. 'Molecular classification of cancer: class discovery and class prediction by gene expression monitoring'** [http://www-genome.wi.mit.edu/MPR/data_set_ALL_AML.html]
10. **Data pertaining to the article 'Broad patterns of gene expression revealed by clustering of tumor and normal colon tissues probed by oligonucleotide arrays'** [http://microarray.princeton.edu/oncology/affydata/index.html]
11. Mardia KV, Kent JT, Bibby JM: *Multivariate Analysis*. London: Academic Press, 1979.
12. Ripley BD: *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press, 1996.
13. Weston J, Mukherjee S, Chapelle O, Pontil M, Poggio T, Vapnik V: **Feature selection for SVMs**. In *Advances in Neural Information Processing Systems 13*. 11th edition. Edited by Solla SA, Leen TK, Muller K-R. Cambridge, MA: MIT Press, 2001.
14. Vapnik V: *Statistical Learning Theory*. New York: John Wiley and Sons, 1999.
15. Dysvik B, Jonassen I: **J-Express: exploring gene expression data using Java.** *Bioinformatics* 2001, **17:**369-370.
16. Dudoit S, Fridlyand J, Speed T: *Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data.* Technical report 576. Berkeley, CA: Department of Statistics, University of California, 2000. [http://www.stat.berkeley.edu/~sandrine/tecrep/576.pdf]
17. Ben-Dor A, Bruhn L, Friedman N, Nachman I, Schummer M, Yakhini Z: **Tissue classification with gene expression profiles.** *J Comput Biol* 2000, **7:**559-583.
18. Chow ML, Moler EJ, Mian IS: **Identifying marker genes in transcription profiling data using a mixture of feature relevance experts.** *Physiol Genomics* 2001, **5:**99-111.
19. **Supplementary information: New feature subset selection procedures for classification of expression profiles** [http://www.ii.uib.no/~trondb/featureselection/]