ELSEVIER

Analytica Chimica Acta 348 (1997) 71–86

ANALYTICA CHIMICA ACTA

Genetic algorithms as a method for variable selection in multiple linear regression and partial least squares regression, with applications to pyrolysis mass spectrometry

David Broadhurst[a], Royston Goodacre[a], Alun Jones[a], Jem J. Rowland[b], Douglas B. Kell[a,*]

[a]*Institute of Biological Sciences, University of Wales, Aberystwyth, Dyfed SY23 3DA, UK*
[b]*Department of Computer Science, University of Wales, Aberystwyth, Dyfed SY23 3DA, UK*

Received 11 July 1996; accepted 31 December 1996

Abstract

Four optimising methods for variable selection in multivariate calibration have been described: one for determining the optimal subset of variables to give the best possible root-mean-square error of prediction (RMSEP) in a multiple linear regression (MLR) model, the second for determining the optimal subset of variables which produce a model with RMSEP less than or equal to a given value. Algorithms three and four were identical to algorithms one and two, respectively, except that this time they use a cost function derived from a partial least squares (PLS) model rather than an MLR model. Applied to a typical set of pyrolysis mass spectrometry data the first variable selection method is shown to reduce the RMSEP of the optimal MLR or PLS model significantly when the number of variables is decreased by approximately half. Alternatively, the number of variables may be reduced substantially (> 10-fold) with no loss in RMSEP.

Keywords: Wavelength selection; Evolutionary computing; Multivariate calibration; Chemometrics; Model selection

1. Introduction

When dealing with mathematical models which are built with little, if any, *a priori* information about the system under analysis it is often difficult to decide how many variables to measure in order to build an adequate model. Often too many variables are measured. This usually happens when the modeller is unsure as to which are important or because the experimental procedure provides measurement of a large number of variables automatically, as typically occurs in spectroscopy [1]. It is very tempting for the modeller to create a model using all available variables. This, however, creates several data analysis problems:

- some of the variables may be completely irrelevant to the objectives of the model, and cloud any meaningful relationships that exist between other variables;
- in order to obtain reliable parameter estimates the number of observations made on each variable should be significantly greater than the number of variables. Failure to adhere to this criterion may lead to serious overfitting in multivariate calibration models [1–6];

*Corresponding author. Tel.: +44 1970 622334; fax: +44 1970 622354; e-mail: dib93/rrg/auj/jjr/dbk@aber.ac.uk.

PII S0003-2670(97)00065-2

- variables may be correlated, in which case replicated information is redundant;
- the signal to noise ratio of certain variables may be so low that their inclusion in the model may be questioned (and will certainly lead to a poorer model), especially if other 'cleaner' correlated variables are available [6];
- when the model parameters are optimised using iterative methods, a greater number of parameters can result in a more complex error surface to be optimised. This complexity may effect the overall convergence time of the model.

For the above reasons it is advantageous to select the 'best' variables prior to the modelling process [7,8]. The general approach to variable selection is to minimise a cost function, where the cost function calculates some metric to decide which subset of the available variables produces the 'best' model. Some sort of optimisation algorithm is then needed to select the subsets to be tested by the cost function.

The simplest method of selection would be to examine all possible combinations of the variables exhaustively. If there are p initial variables and the best m are required then this would result in $p!/(m!(p-m)!)$ possible subsets [9]. If p and m are large then this is computationally expensive (and in most situations practically impossible). Disqualifying this search method means that there is no guaranteed way of finding the optimal variable subset for a given model [10]. However, as a rule, adequate local solutions can be found in a relatively short time. The three most popular optimisation strategies are *forward selection* (FS), *backward selection* (BS) and *stepwise multiple regression* (SMR).

1.1. Forward selection

Forward selection begins by looking at each individual variable. The one that evaluates the lowest cost function is kept. This variable is then examined in conjunction with each of the remaining $(p-1)$ variables until the pair that minimises the cost function is found. This pair is retained and tried in conjunction with each of the remaining $(p-2)$ variables. This process continues until the optimum subset is found or a stopping criterion is met.

1.2. Backward selection

This algorithm operates in the opposite direction. Starting with p variables, each one is omitted in turn and the optimum subset retained. Then with this $(p-1)$ subset each of the remaining variables is omitted in turn and again the optimum subset kept. This process is repeated until an optimum subset of m variables is found or a stopping criterion is met.

1.3. Stepwise multiple regression

Stepwise multiple regression is a modified form of forward selection. The model starts out by including only one variable, and more variables are subsequently added, but at each stage a backward elimination-style test is also applied. If a variable is added, but becomes less important as a result of subsequent additions, SMR will allow its removal from the model.

With the above methods each variable is studied independently, and no consideration is given to variable interaction. For example, variables p_i and p_j may be unimportant separately but in conjunction they may provide useful information. Thus with these methods only a very small part of the available search space is examined. In order to study the importance of uncorrelated groups of variables as well as individual variables a more global optimisation method needs to be used, where variables are selected or rejected simultaneously.

The simplest simultaneous selection method is the exhaustive-subset search. However this has already been eliminated as an option due to computational overheads. Other methods are based on a stochastic search of the problem space. The simplest but least effective stochastic search method involves generating a population of n subsets, each subset containing a random combination of variables taken from the total set. Each subset is then applied to the cost function. This function returns a single numerical value which is proportional to the 'utility' of the given subset. The subset with the best cost function response is considered to be the optimum. This method has obvious flaws, the main one being that for the method to be useful the population of subsets has to be large, again leading to high computational costs.

Another stochastic optimisation technique which is reasonably new to statistical modelling is the *genetic*

algorithm (GA) [11,12]. Again in this algorithm a population of n subsets is created, each containing a random combination of variables. The cost function for each subset is then evaluated in turn. Then, using techniques loosely based on biological genetics and evolution, a new population is created.

First, each subset is considered as a string of m 1's and 0's, where m is the total number of variables to choose from. The state of each variable is represented by a '1' (selected to be in the model) or a '0' (not selected). In genetic terms each variable is called a *gene* and a set of variables is called a *chromosome*. For example, in a variable selection problem starting with 8 variables, one possible chromosome would be 00110101. This can be translated such that variables 3,4,6, and 8 are to be used in the modelling process and variables 1,2,5, and 7 are to be omitted.

A weighted random selection is applied to the original population where the probability of a particular subset (chromosome) being selected is a function of its cost function response. Thus chromosomes with a good cost function response will have a greater chance of selection. Using this method two of the chromosomes are selected and 'mated', swapping sections of their respective gene sequences. This process produces two new 'child' chromosomes inheriting characteristics of their 'parents'. These child chromosomes are then subjected to random mutation where the state of each gene may be changed from a '1' to a '0' or vice versa. The probability of this change is normally very small.

The process of selection followed by reproduction followed by mutation is then repeated until n new chromosomes are created. The cost function is then evaluated for each of the chromosomes and the whole process repeats itself.

The algorithm continues until a stopping criterion is reached. For example, this may be that a given cost function response is met, a certain number of generations has passed, or the chromosomes have converged to a similar configuration.

The five steps of encoding into chromosomes, initial population selection, evaluation of the cost function, reproduction, and testing for the stopping criterion are the basic building blocks for all GAs. However, there are various ways of carrying out each step. A full explanation of the subject of GAs can be found in [11,12]. Tutorial reviews of their applications in chemometrics and analytical chemistry are available [13,14], while the use of GAs for variable selection in spectroscopy has been described, for instance, by [15–17]. Other studies of wavelength selection in spectroscopy include [1,15,18–26].

It occurred to us that a combination of GAs and some kind of supervised learning multivariate calibration method would provide an excellent approach to the problem of model selection, and the examples given here are with particular reference to pyrolysis mass spectrometry. Four algorithms are described:

- one for determining the optimal subset of variables to give the best possible root-mean-square error of prediction (RMSEP) in a multiple linear regression (MLR) model;
- the second for determining the optimal subset of variables which produce a model with RMSEP less than or equal to a given value. This value is generally the best RMSEP using all the variables. This method may also be expected to improve generalisation according to the parsimony principle [6] and is useful when the acquisition of extra variables is associated with a cost.
- algorithms three and four are identical to algorithms one and two, respectively, except that this time they use a cost function derived from a partial least square regression (PLS) model rather than an MLR model.

2. Methods

2.1. Linear modelling

The simplest method of producing a multivariate calibration model is to use MLR [27]. A fixed regressor model is used, of the form

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{b} + \boldsymbol{e},$$

where $\boldsymbol{b}$ is the unknown parameter vector, the $\boldsymbol{X}$ matrix and $\boldsymbol{y}$ vector are the measured calibration data for regressor variables $\boldsymbol{x}$ and response variable y respectively. The error vector, $\boldsymbol{e}$ consists of systematic modelling errors and random measurement errors assumed to have normal distribution and expected value $E(e)=0$. Estimates of the parameter values are

determined by minimising $\boldsymbol{e}$. This is simply done by solving the equation

$$\boldsymbol{b}' = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y},$$

where $\boldsymbol{b}'$ is the least squares estimate of $\boldsymbol{b}$. This produces a model: $\boldsymbol{z}=\boldsymbol{X}\boldsymbol{b}'$, where $\boldsymbol{z}$ is the predicted response vector given the calibration matrix $\boldsymbol{X}$ [27].

A possible problem with this method of parameter estimation is that $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ (the covariance matrix) can be ill-conditioned. Ill-conditioning occurs when there is (approximate) multicollinearity in the x variables (i.e. there is high correlation between some of the columns of $\boldsymbol{X}$). Any high correlation will result in a numerically unstable (or incalculable due to singularity) inverse matrix creating large errors in the parameter estimates [28]. Multicollinearity is most readily observed when the number of rows exceeds the number of columns in $\boldsymbol{X}$ (i.e. there are fewer regression variables than measured observations). In this case there is no way the regressor variables can be independent [9].

In spectroscopy, multicollinearity of the regressor variables is highly probable and often the number of observations is less than the number of regressor variables. Thus the only way to build stable MLR models is to apply variable selection techniques. Combining GAs and MLR should provide a reasonable modelling tool in many areas of multivariate analysis. The application of this hybrid method has already proved to be effective in the task of wavelength selection in spectroscopy [16,17].

Another approach to linear modelling involves projecting the x variables onto a set of orthogonal latent variables. In principal components regression, PCR [4,9], the projection is carried out using a basis set formed from the eigenvectors of $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$. The latent variables are then used in a MLR model with the y variable. In PLS [4,29], the basis set is formed by looking at both the $\boldsymbol{X}$ matrix and the $\boldsymbol{y}$ vector, then again a simple linear regression is performed.

In PLS the latent variables (factors) are extracted from the calibration data one at a time in order of decreasing relevance to the model. Due to this ranking and the fact that each variable is independent, the problem of latent variable selection is quite trivial. A model is built using just the first and most important factor and then evaluated, a second model is then built using the first two factors and again evaluated. The process is repeated adding more and more factors until the 'best' model is created. In the present implementation, variables were mean centred and scaled to unit variance.

It is often assumed that because the regressor variables are projected onto, possibly a small number of, orthogonal latent variables there will be no need to apply selection algorithms to the regressor variables. However, even though the effects of multicollinearity in the calibration data are greatly reduced by projection they are not completely removed. Also, when dealing with quite noisy data, the removal of irrelevant variables can only improve the overall model with or without latent variables.

Although PCR is simpler to calculate, PLS has usually proved to be more effective [30] and is thus more widely used. Therefore, in the following algorithms only MLR and PLS models were investigated.

2.2. Model validation

In order to assess the utility of a newly formed model we need to obtain a value which gives a measure of its performance when presented with new data. A commonly used measure is the RMSEP [31]. This is simply the mean squared difference between the predicted response of a model, z, and the true response, y. Thus,

$$\mathrm{RMSEP} = \sqrt{\sum_{i=1}^{n}(y_i - z_i)^2/n},$$

where n is the number of objects.

When a model is built using the calibration data there is a danger that the model will *over-fit* the data. That is, the model will succeed in finding a relationship between the calibration data $\boldsymbol{X}$ and $\boldsymbol{y}$ but it will have poor predictive power. For a model to be truly valid it must be specific enough to describe the desired relationship but also general enough to ignore chance relationships such as noise.

In order to avoid overfitting, some sort of model validation is needed. The two most popular methods are *data-splitting* and *full cross validation*.

In data-splitting, the measured data, $\boldsymbol{X}$ and $\boldsymbol{y}$, are split into two sets, commonly called the training set and test set ($\boldsymbol{X}_{\mathrm{train}}$, $\boldsymbol{X}_{\mathrm{train}}$ and $\boldsymbol{X}_{\mathrm{test}}$, $\boldsymbol{y}_{\mathrm{test}}$). The training set is then used to calibrate the model as described in Section 2.1, and the test set is left untouched until the

model has been built. Once built two measures of the model's utility can be calculated. Both measures use the RMSEP equation. Firstly X_{train} is applied to the model resulting in the prediction vector Z_{train} which leads to a calibration RMSEP (which will now be called RMS error in calibration RMSEC). Then X_{test} is applied to the model creating the vector Z_{test} which leads to a model independent estimation of the RMSEP. The RMSEP is normally greater than the RMSEC but gives a far greater indication of the model's utility.

The manner in which the data are split into the two groups is very important and has been the subject of several papers [32–35]]. Both groups must be chosen to contain a representative spread of the measured (x and y) data to provide accurate measures of model validity.

The drawback to the data-splitting method of validation is that, by partitioning the data into two groups, information from which the model could be extracted is lost. This could result in a much poorer model. If the number of measured observations is very large then this may not be a problem. However, often the number of observations is limited.

If data splitting is impractical, *full cross validation* methods can be used, of which the PRESS (PREdiction Sum of Squares) statistic is an example [7]. Consider a data set consisting of n objects in which the first object is withheld. The parameters for the model are then estimated. When the model is formulated, the response to the deleted object is predicted ($z_{i,-i}$). The first object is then replaced in the data set and the second object removed; again the model parameters and the response for the missing object are estimated. This process is repeated until all n objects have at some point been eliminated from the modelling process. This results in the production of n model fits and n predictions from which n residuals, $E_{i,-i}=y_i-z_{i,-i}$ ($i=1,2,\ldots,n$), can be calculated. The PRESS statistic can then be calculated using the definition

$$\text{PRESS} = \sum_{i=0}^{n} (y_i - z_{i,-i})^2.$$

Although full cross validation methods use more of the available data for building the model, they have the disadvantage that n 'artificial' models have to be built before a measure of overall model quality can be calculated. This may cause serious time delays in an iterative variable selection algorithm. Also it has been suggested that the statistical theory behind this method of validation is not yet fully understood and can possibly overestimate the model's utility [33].

For the above reasons the algorithms presented in this paper use the data-splitting method of model validation.

2.3. Model validation and genetic algorithms

In the work presented in this paper, a GA is used to find the optimum subset of regressor variables for a given modelling method based on the results of cost function evaluations for all candidate genetic chromosomes (variable subsets). The chromosomes will be passed to a modelling subroutine where the relevant columns of the calibration data (X_{train} and X_{test}) are extracted and used, together with y_{train} and y_{test}, to calibrate and validate a model. The cost function evaluation will therefore be based on the RMSEP of this model.

At the end of an evolutionary stage, each chromosome in the genetic population will have an associated cost function value. This value is then used to determine the probability of its associated chromosome being used to create the next generation. As new generations are created the cost function value is optimised.

If the cost function is based on the RMSEP calculated for a given chromosome then the RMSEP is not independent of the overall modelling process. It is therefore questionable whether this is a suitable measure of validity for the final mathematical model (with the optimal variables selected). It is necessary to find a new independent measure of model validity.

It is proposed that the available data are split into three sets: a training set (X_{train}, y_{train}) a model validation set (X_{mv}, y_{mv}) and an independent test set (X_{its}, y_{its}). The training set is used to calibrate the model, producing an RMSEC value. The model validation set is used to test the models validity, producing an RMSEP_{mv} value. The independent test set used to test the validity of the final model, producing an RMSEP_{its} value.

2.4. Software design and implementation

The software described herein was written in-house using Microsoft Visual C++ (version 2.1) running under Microsoft Windows NT version 3.51 on an IBM-compatible PC.

C++ is an object-oriented programming language. Object-oriented programming involves deconstructing the system to be modelled into component objects, each of which has a set of data elements and a set of functions that may operate on these data. At its most basic level an object-oriented system can be described as a set of objects that communicate with each other to achieve some goal. Each object may be considered as a small virtual computer with its own memory and its own instruction set. Communication is achieved by sending messages between objects which are interpreted in a way dependent on the object's instruction set.

In this application, five object types (commonly known as classes) are defined as follows (the term in the parentheses is the class name):

1. Matrix (matrix).
2. MLR Model (mlr).
3. PLS Model (pls).
4. Chromosome (chrom).
5. Population (pop).

2.4.1. Matrix (matrix)

Data elements

- An array of double precision numbers

Member functions

- All the basic numerical operators associated with matrix algebra.

2.4.2. MLR Model (mlr)

Data elements

- Data sets ($\mathbf{X}_{train}$, $\mathbf{y}_{train}$) ($\mathbf{X}_{mv}$, $\mathbf{y}_{mv}$) and ($\mathbf{X}_{its}$, $\mathbf{y}_{its}$). Each of these is a *matrix* object.
- Maximum number of regressor variables, *var_max*. As explained earlier MLR requires that the inverse of the covariance matrix must be calculable, therefore the number of variables selected for a model, *var_ used*, must be less than or equal to the number of measured observations.

Member functions

- Calibration and validation of the model for a given chromosome, *run(*chrom x)*. The model object is given access to the data from a given chrom object *x*, relevant columns of the calibration data ($\mathbf{X}_{train}$ and $\mathbf{X}_{test}$) are extracted and a model is built and validated. A cost function is then evaluated producing a measure of chromosome fitness. All the calculated values are then passed back to the *chrom* object.

2.4.3. PLS model (pls)

Data elements

- Data sets ($\mathbf{X}_{train}$, $\mathbf{y}_{train}$) ($\mathbf{X}_{mv}$, $\mathbf{y}_{mv}$) and ($\mathbf{X}_{its}$, $\mathbf{y}_{its}$). Each is a *matrix* object.
- Maximum number of factors, *fact_max*. This value is used when trying to find the optimum number of factors to be used by the PLS model. A model is built and validated using 1 factor, then 2 factors, then 3 factors etc., continuing until *fact_max* factors have been used. Then the 'best' number of factors is chosen by finding the minimum $RMSEP_{mv}$ from all the validated models.

Member functions

- Identical to mlr.

2.4.4. Chromosome (chrom)

Data elements

- An array of characters, *string(chrom_length)*. This array represents the gene sequence of the chromosome. Each element can either be a '1' or a '0'.
- Chromosome length *chrom_length*. This is equal to the total number of available variables.
- Cost function evaluation, *cost*, RMSEC value, *cal_error*; $RMSEP_{mv}$ value, *mv_error*; $RMSEP_{its}$ value, *its_error*. These variables are used to store the results obtained from the model built when the chromosome is passed to the model object.

Member functions

- None.

2.4.5. Population (pop)

Data elements

- An array of *chrom* objects, *pool(p)*.

- A *model* object, *mod*.
- Chromosome length *chrom_length*. This is equal to the total number of available variables.
- Population size *p*. This value is dependent on the total number of available variables, as the initial random population should provide good coverage of the total problem space. Computation time also has to be taken into account, so *p* should not be too large.
- Percentage of the population retained after each evolutionary stage, *%r*. In the original Goldberg algorithm [11], the chromosomes of a given generation are completely replaced by a population of child chromosomes. In our algorithm the top *%r* chromosomes of a generation are kept for the next generation. In this way very good chromosomes are not lost in a single evolutionary stage.
- Maximum number of generations *G*. The GA will stop either when the population has converged or when the population has been through *G* evolutionary stages. As with the Goldberg GA [11] repetition of existing chromosomes has been allowed. Convergence is defined as to have taken place when the top *%r* chromosomes of a given population are identical.
- Probability of crossover *P(c)*.
- Probability of mutation *P(m)*.

Member functions

- Creation of the initial population, *initialise('file_name')*. The parameter, *file_name*, is the name of a file containing all the information needed to initialise all its data elements.
- Evaluate all the chromosomes in a population, *eval*().
- Create a new population, *reproduce*().

2.5. The model selection algorithms

Four algorithms will be described

1. GAMLR-E: Variable selection using a GA with the aim of minimising the $RMSEP_{mv}$ for a given MLR model.
2. GAMLR-V: Variable selection using a GA with the aim of minimising the number of regressor variables used to produce a MLR model with a predetermined maximum $RMSEP_{mv}$, *mv_max* (the maximum $RMSEP_{mv}$ determined from analysis of algorithm 1).
3. GAPLS-E: Variable selection using a GA with the aim of minimising the $RMSEP_{mv}$ for a given PLS model.
4. GAPLS-V: Variable selection using a GA with the aim of minimising the number of regressor variables used to produce a PLS model with a predetermined maximum $RMSEP_{mv}$, *mv_max* (the maximum $RMSEP_{mv}$ determined when using all the available regressor variables).

All four methods use the same basic GA, differing only in the model object used in *pop* (*mlr* or *pls*) and the particular cost function subroutine used by this model.

2.5.1. Cost function for GAMLR1

```
if (var_used > var_max)
{
cost = 1e10^6;
}
else
{
cost = mv_error;
}
```

2.5.2. Cost function for GAMLR2

```
if (var used > var_max)
{
cost = 1e10^6;
}
else
{
if (mv_error > mv_max)
{
cost = var_used/chrom_length;
}
else
{
cost = mv_error;
}
}
```

2.5.3. Cost function for GAPLS1

```
cost = mv_error;
```

2.5.4. Cost function for GAPLS2

```
if (mv_error > mv_max)
{
cost = var_used/chrom_length;
}
else
{
cost = mv_error;
}
```

2.6. Data sets used

Two data sets were used to verify the effectiveness of the four model selection methods. The first data set was obtained using pyrolysis mass spectrometry (PyMS) on binary mixtures of the protein lysozyme with glycogen [36]. The second data set was also created using PyMS, in this case for the analysis of fermentation broths for a drug of commercial interest [37].

2.7. Preparation of the lysozyme and glycogen mixtures

Binary mixtures were prepared such that 5 ml of a solution contained 0–100 μg of the determinand lysozyme (from chicken egg white, Sigma), in steps of 5 μg, in 20 μg glycogen (oyster type II, Sigma) [36].

2.8. Preparation of the fermentation samples

Samples were taken from fermentation broths. The identity of the producing organism and the structure of the metabolite of interest are proprietary; the micro-organism and the product are therefore coded M and P respectively.

Samples were taken aseptically from fermentations and frozen until they were analysed by PyMS. The medium used to grow mutants derived from organism M was a complex medium containing mixed sugars and hydrolysed protein, and samples were taken at different times. The amount of P was determined using high performance liquid chromatography (HPLC). The error in these values was typically 2–5% [37].

2.9. Pyrolysis mass spectrometry

5 μl aliquots of the above samples were evenly applied onto iron–nickel foils to give a thin uniform surface coating. Prior to pyrolysis the samples were oven-dried at 50°C for 30 min. Each fermentation sample was analysed in triplicate and each lysozyme/glycogen mixture in quadruplicate. The pyrolysis mass spectrometer used in this study was a Horizon Instruments PYMS-200X; for full operational procedures see [36,38,39]. The sample tube carrying the foil was heated, prior to pyrolysis, at 100°C for 5 s. Curie point pyrolysis was at 530°C for 3 s, with a temperature rise time of 0.5 s. These conditions were used for all experiments. The data from PyMS were collected over the *m/z* range 51 to 200 and may be displayed as quantitative pyrolysis mass spectra; the abscissa represents the *m/z* ratio whilst the ordinate contains information on the ion count for any particular *m/z* value ranging from 51–200 (e.g., as in Fig. 1). In all cases, data were normalised to a value of 2^{16} for the total ion count, to remove the effects of sample size *per se*.

Prior to any analysis the mass spectrometer was calibrated using the chemical standard perfluorokerosene (Aldrich), such that *m/z* 181 was one tenth of *m/z* 69.

2.10. Model data

2.10.1. Data set 1: lysozyme in glycogen

The total data set consisted of the four normalised replicate pyrolysis mass spectra derived from the mixtures containing 0, 5, 10, ..., 90, 95 and 100 μg of lysozyme ($\boldsymbol{X}$-matrix), and the accompanying actual (true) amount of lysozyme in the mixtures ($\boldsymbol{y}$-vector). Thus the number of regressor variables equals 150 (the mass range of the pyrolysis mass spectra), and the total number of objects equals 84. The $\boldsymbol{X}$-data were mean centred and scaled in proportion to the reciprocal of their standard deviations.

The above data set was then partitioned equally into the three following sub-sets (values in the parentheses are the dimensions of each matrix)

- $\mathbf{X}_{train}$, $\mathbf{y}_{train}$ (28×150, 28×1).
- $\mathbf{X}_{mv}$, $\mathbf{y}_{mv}$ (28×150, 28×1).
- $\mathbf{X}_{its}$, $\mathbf{y}_{its}$ (28×150, 28×1).

The data were partitioned using the 'multiplex' algorithm [35]. This algorithm systematically places

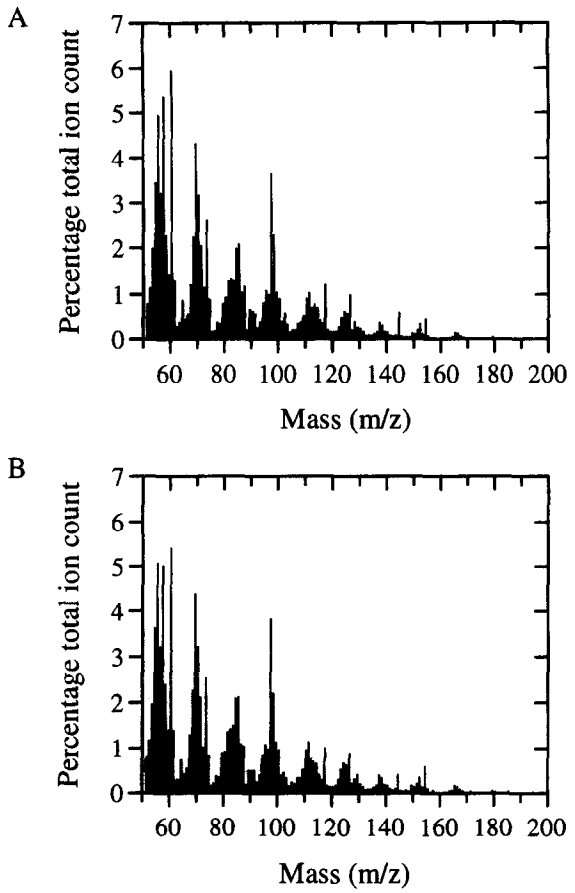

Fig. 1. Normalized pyrolysis mass spectra of 45 μg lysozyme mixed with 20 μg glycogen (A) and 50 μg lysozyme mixed with 20 μg glycogen (B). It may be observed that the differences are relatively modest as judged by eye.

data points into the above sets such that the training set range covers both the model validation set range and the independent test set range, and that all three sets are representative. The software used to implement this algorithm was written by Dr. Alun Jones and runs under Microsoft Windows NT on an IBM-compatible PC. The data were partitioned by the *y*-data.

2.10.2. Data set 2: drug P in M

The total data set consisted of the three normalised triplicate pyrolysis mass spectra derived from 39 mixtures containing between 0 and 86.59 μg ml^{-1} of drug P in M (*X*-matrix), and the accompanying actual (true) amount of P in M (*y*-vector). Thus the number of regressor variables equals 150 (the mass range of the pyrolysis mass spectra), and the number of objects equals 116 (one object was lost during experimentation). The *X*-data were mean centred and scaled in proportion to the reciprocal of their standard deviations.

The above data set was then partitioned equally using the 'multiplex' algorithm [35] into the three following subsets (values in the parenthesis are the dimensions of each matrix):

- $\mathbf{X}_{train}$, $\mathbf{y}_{train}$ (38×150, 38×1)
- $\mathbf{X}_{mv}$, $\mathbf{y}_{mv}$ (39×150, 39×1)
- $\mathbf{X}_{its}$, $\mathbf{y}_{its}$ (39×150, 39×1).

The data were again partitioned using the 'multiplex' algorithm [35].

3. Results

3.1. Parameter values for the genetic algorithm

All the results were obtained using the following parameter values in the GA:

Chromosome length	150
Number of generations	200
Chromosome population size	200
Percentage of each population kept	20%
Probability of crossover	0.7
Probability of mutation	0.1

The choices were governed by a combination of known useful defaults [12], a number of preliminary experiments, and the processing power available.

3.2. Data set 1

3.2.1. Algorithm GAMLR-E

The genetic algorithm GAMLR-E was applied to data set 1 five times. The stopping criterion used was to allow a maximum number of generations (200). By this time the GA was observed to converge to a constant $RMSEP_{mv}$. Each run took approximately 1 hour using a 90 MHz Intel Pentium processor. The results are shown in Table 1. The best results obtained gave an $RMSEP_{its}$ of 3.91%, using a subset of 23 regressor variables. It is interesting to note that each of the five results yielded quite different variable subsets. Although some variables were selected in

Table 1
Comparison of the quality of the MLR models produced by five runs of algorithm GAMLR-E on the lysozyme/glycogen data. The aim of this variable selection genetic algorithm is to optimise the percentage root-mean-square-error of prediction for an MLR model. Values for the percentage root-mean-square-error in calibration, % RMS-error of prediction in model validation (used as the cost function by the algorithm), and % RMS-error of prediction for an independent test set were calculated for each model and the $RMSEP_{its}$ used as the final measure of model utility. The bold faced type indicates the optimal model

Expt. number	Number of variables	% RMSEC	% $RMSEP_{mv}$	% $RMSEP_{its}$
1	**23**	**0.528105**	**4.02438**	**3.91973**
2	21	2.69248	4.29649	6.6678
3	22	2.34557	4.71219	5.15602
4	23	0.511461	4.34628	4.94728
5	19	2.5276	4.54529	5.39324

more than one of the final subsets, none of these variables was selected in all of the subsets (Fig. 2). This suggests that for this data set the model error surface has many local minima, probably resulting from the large amount of multicollinearity in the regressor variables and the normalization used.

3.2.2. *Algorithm GAPLS-E*

The genetic algorithm GAPLS-E was applied to data set 1 five times using the same stopping criterion as GAMLR-E. Each run took approximately 4 h using a 90 MHz Intel Pentium processor. The maximum number of factors from which the optimum for each model is found, *fact_max*, was set to 16. This value was chosen to be quite low because previous research into modelling these PyMS data using PLS suggested that only a small number of factors is needed [36,38,39], and second, as *fact_max* increases the computation time for each chromosome evaluation increases proportionally. Thus, to reduce overall computation time, *fact_max* must be kept reasonably low.

The results of the five runs are shown in Table 2. The best results obtained gave an $RMSEP_{its}$ of 3.77%, using a subset of 71 regressor variables and 5 factors, compared with the PLS model built using all 150 regressor variables which produced an $RMSEP_{its}$ of 7.52% using 2 factors.

Fig. 3 shows the model predictions versus known amount of lysozyme in glycogen for (a) the PLS model using all 150 regressor variables, (b) the best model produced by GAMLR-E, and (c) the best model produced by GAPLS-E. All three methods produce good models; however the two GAs produce significantly better results.

Again there was no strong correlation between the variable subsets chosen in each of the five experiments (data not shown).

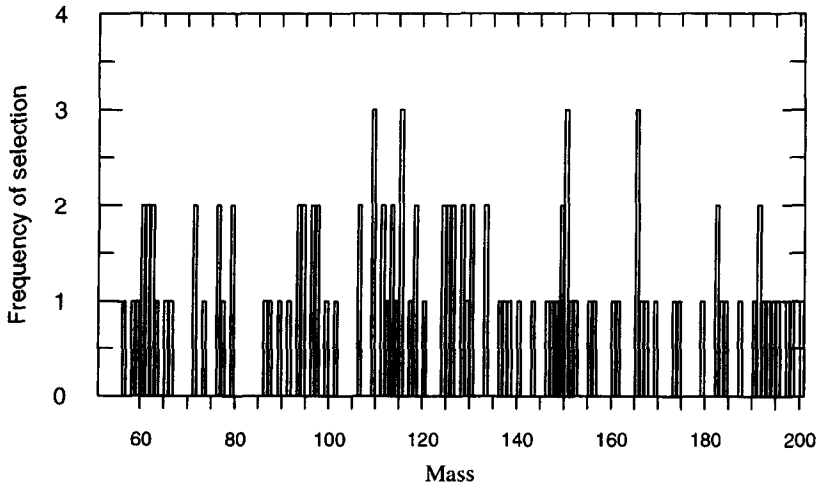

Fig. 2. Frequency of selection of individual variables upon repeated training of GAMLR-E on the lysozyme/glycogen dataset. 5 GAs were trained for 200 generations as described in the text, and the frequency of selection in the final chromosome plotted.

Table 2
Comparison of the quality of the PLS models produced by five runs of algorithm GAPLS-E on the lysozyme/glycogen data. The aim of this variable selection genetic algorithm is to optimise the % RMSEP for a PLS model. Values for the RMSEC, $RMSEP_{mv}$ and $RMSEP_{its}$ were calculated for each model and the $RMSEP_{its}$ used as the final measure of the utility of each model. The bold faced type indicates the optimal model

Expt. number	Number of variables	Number of factors	% RMSEC	$\%RMSEP_{mv}$	$\%RMSEP_{its}$
1	67	5	2.63314	3.19682	3.89642
2	72	5	2.77041	3.10056	5.37667
3	**71**	**5**	**2.29251**	**3.05553**	**3.77118**
4	70	5	2.47559	3.07834	3.86994
5	68	5	2.61594	3.19676	4.94723
No selection [a]	150	2	6.56	7.48	7.52

[a] The model statistics for a PLS model built using all the 150 available regressor variables.

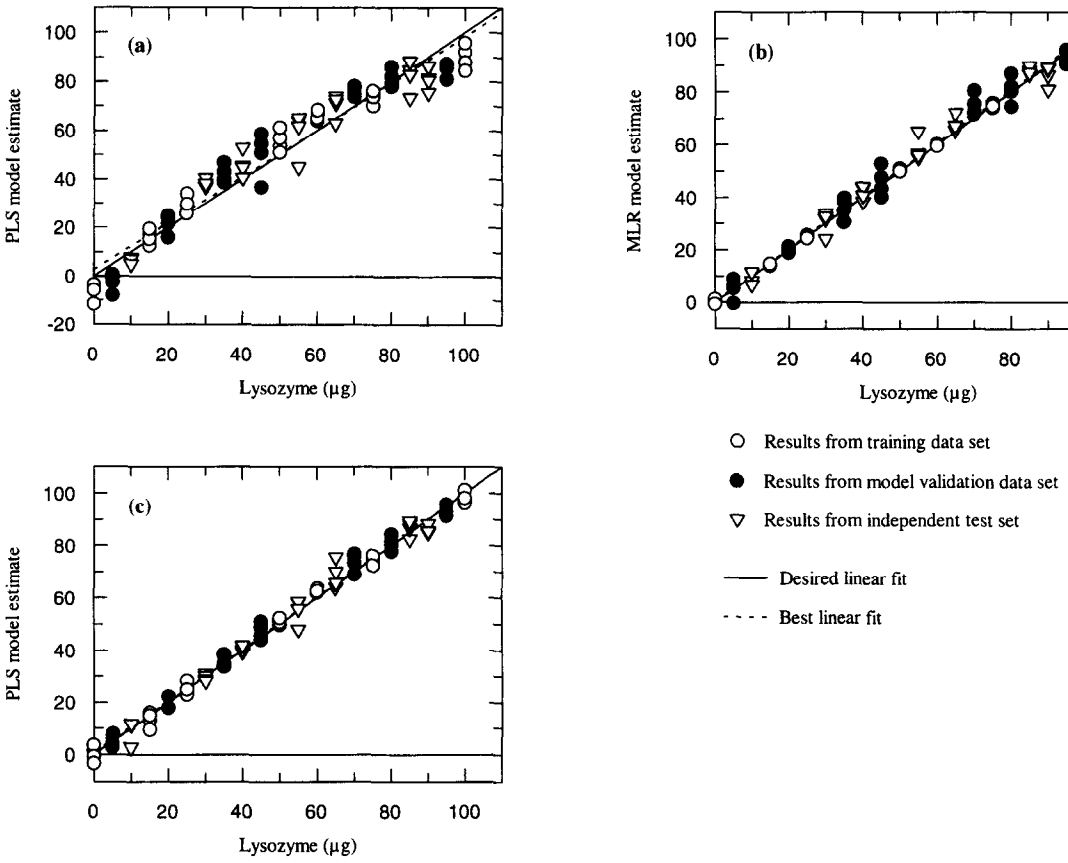

Fig. 3. Model predictions versus known amount of lysozyme in glycogen for (a) the PLS model using all 150 regressor variables (%RMSEC=6.56; $\%RMSEP_{mv}$=7.48; $\%RMSEP_{its}$=7.52) (b) the best model produced by GAMLR-E (%RMSEC=0.53; $\%RMSEP_{mv}$=4.02; $\%RMSEP_{its}$=3.92), and (c) the best model produced by GAPLS-E (%RMSEC=2.29; $\%RMSEP_{mv}$=3.06; $\%RMSEP_{its}$ =3.77). ○=Results from training data set, •=Results from model validation data set, ▽=Results from independent test data set, ———=Desired linear fit, - - - -=Best linear fit.

Table 3
Comparison of the quality of the MLR models produced by five runs of algorithm GAMLR-V on the lysozyme/glycogen mixture data. The aim of this variable selection genetic algorithm is to optimise the number of selected regressor variables for an MLR model given a maximum allowable model RMSEP. For this set of experiments the maximum RMSEP was set to 5% (Chosen by examining the results in Table 1). Values for the RMSEC, $RMSEP_{mv}$ and $RMSEP_{its}$ were calculated for each model and the $RMSEP_{its}$ used as the final measure of the utility of each model. The bold faced type indicates the optimal model

Expt. number	Number of variables	RMSEC	$RMSEP_{mv}$	$RMSEP_{its}$
1	16	2.37686	4.55442	6.52044
2	12	1.92472	4.63949	5.05392
3	17	1.4685	4.41333	4.95204
4	**14**	**2.3492**	**3.82072**	**4.02591**
5	15	2.05237	4.37819	4.88966

3.2.3. Algorithm GAMLR-V

The genetic algorithm GAMLR-V was applied to data set 1 five times. The stopping criterion used was to allow a maximum number of generations (200). By this time the GA was observed to converge to a constant cost value. Each run took approximately 1 hour using a 90 MHz Intel Pentium processor. The threshold value for the $RMSEP_{mv}$ was set to 5%, since this was approximately the worst $RMSEP_{its}$ result from the GAMLR-E experiments. This value was chosen because the aim of this GA is not to produce the best model error but to reduce the number of regressor variables while retaining a reasonable model.

The results of the five runs are shown in Table 3. The best result produced a subset of 12 variables with a corresponding $RMSEP_{its}$ of 5.05%.

Again there was no strong correlation between the variable subset chosen in each of the five experiments (data not shown).

3.2.4. Algorithm GAPLS-V

The genetic algorithm GAPLS-V was applied to data set 1 five times using the same stopping criterion as GAMLR-V. Each run took approximately 4 h using a 90 MHz Intel Pentium processor. The maximum number of factors, *fact_max*, was 16. The threshold value for the $RMSEP_{mv}$ was set to 7.5%, the $RMSEP_{mv}$ for the PLS model built with all the regressor variables.

The results of the five runs are shown in Table 4. The best result produced a subset of 8 variables with a corresponding $RMSEP_{its}$ of 7.11%.

Again, there was no strong correlation between the variable subset chosen in each of the five experiments (data not shown).

3.3. Data set 2

3.3.1. Algorithm GAMLR-E

The genetic algorithm GAMLR-E was applied to data set 2 five times. The stopping criterion used was to allow a maximum number of generations (200). By this time the GA was observed to converge to a constant $RMSEP_{mv}$. Each run took approximately 1.5 h using a 90 MHz Intel Pentium processor. The

Table 4
Comparison of the quality of the PLS models produced by five runs of algorithm GAPLS-V on the lysozyme/glycogen mixture data. The aim of this variable selection genetic algorithm is to optimise the number of selected regressor variables for a PLS model given a maximum allowable model RMSEP. For this set of experiments the maximum RMSEP was set to 7.5% (the $RMSEP_{mv}$ for the PLS model built using all variables). Values for the RMSEC, $RMSEP_{mv}$ and $RMSEP_{its}$ were calculated for each model and the $RMSEP_{its}$ used as the final measure of the utility of each model. The bold faced type indicates the optimal model

Expt. number	Number of variables	Number of factors	RMSEC	$RMSEP_{mv}$	$RMSEP_{its}$
1	**8**	**2**	**7.02031**	**5.7696**	**7.11044**
2	8	2	6.92487	7.24001	7.51208
3	6	2	6.78251	7.34695	9.0413
4	8	2	7.27413	7.06115	7.16085
5	8	2	6.87259	6.50481	7.41126
No selection [a]	150	2	6.56	7.48	7.52

[a] The model statistics for a PLS model built using all the 150 available regressor variables.

Table 5
Comparison of the quality of the MLR models produced by five runs of algorithm GAMLR-E on the Drug P in M data. The aim of this variable selection genetic algorithm is to optimise the % RMSEP for an MLR model. Values for the RMSEC, $RMSEP_{mv}$ and $RMSEP_{its}$ were calculated for each model and the $RMSEP_{its}$ used as the final measure of the utility of each model. The bold faced type indicates the optimal model

Expt. number	Number of variables	RMSEC	$RMSEP_{mv}$	$RMSEP_{its}$
1	31	**4.710397**	**29.76779**	**36.16139**
2	33	3.372574	30.18265	40.25185
3	33	1.984519	31.20229	41.28882
4	30	3.171707	24.06169	43.20344
5	28	4.55253	24.38563	40.94605

results are shown in Table 5. The best results obtained gave an $RMSEP_{its}$ of 36.16%, using a subset of 31 regressor variables.

There was no strong correlation between the variable subset chosen in each of the five experiments (data not shown).

3.3.2. Algorithm GAPLS-E

The genetic algorithm GAPLS-E was applied to data set 2 five times using the same stopping criterion as GAMLR-E. Each run took approximately 6 h using a 90 MHz Intel Pentium processor. The maximum number of factors from which the optimum for each model is found, *fact_max*, was set to 16.

The results of the five runs are shown in Table 6. The best results obtained gave an $RMSEP_{its}$ of 26.81%, using a subset of 52 regressor variables and 2 factors, compared with the PLS model built using all 150 regressor variables which produced an $RMSEP_{its}$ of 26.65% using 2 factors.

Fig. 4 shows the model predictions versus known amount of drug P in M for (a) the PLS model using all 150 regressor variables, (b) the best model produced by GAMLR-E, and (c) the best model produced by GAPLS-E. Both the GAMLR-E algorithm and the GAPLS-E algorithm produced poor models, not managing to improve on the PLS model using all the variables. However, the results do show the significance of using three data sets in the model construction. Looking at both Tables 5 and 6 the difference between $RMSEP_{mv}$ and $RMSEP_{its}$ for a given model can be as much as 10%. The GAs used optimise the $RMSEP_{mv}$. Thus the final model produced may have selected variables that fit both the desired relationship and also noise in both the training set and the model validation set.

3.3.3. Algorithm GAMLR-V

The genetic algorithm GAMLR-V was applied to data set 2 five times. The stopping criterion used was to allow a maximum number of generations (200). By this time the GA was observed to converge to a constant cost value. Each run took approximately 1.5 h using a 90 MHz Intel Pentium processor. The threshold value for the $RMSEP_{mv}$ was set to 25%, as this was approximately the worst $RMSEP_{its}$ result from the GAMLR-E experiments.

The results of the five runs are shown in Table 7. The best result produced a subset of 21 variables with a corresponding $RMSEP_{its}$ of 29.8%.

Table 6
Comparison of the quality of the PLS models produced by five runs of algorithm GAPLSE on the Drug P in M data. The aim of this variable selection genetic algorithm is to optimise the % RMSEP for a PLS model. Values for the RMSEC, $RMSEP_{mv}$ and $RMSEP_{its}$ were calculated for each model and the $RMSEP_{its}$ used as the final measure of the utility of each model. The bold faced type indicates the optimal model.

Expt. number	Number of variables chosen	Number of factors	RMSEC	$RMSEP_{mv}$	$RMSEP_{its}$
1	**52**	**2**	**18.15354**	**15.94651**	**26.81851**
2	40	2	18.08919	16.05765	29.32971
3	54	2	17.41474	16.04009	27.80314
4	54	2	16.82047	16.01987	28.88494
5	54	2	17.74341	15.92745	27.63066
No selection [a]	150	2	18.70379	21.85767	23.65989

[a] The model statistics for a PLS model built using all the 150 available regressor variables.

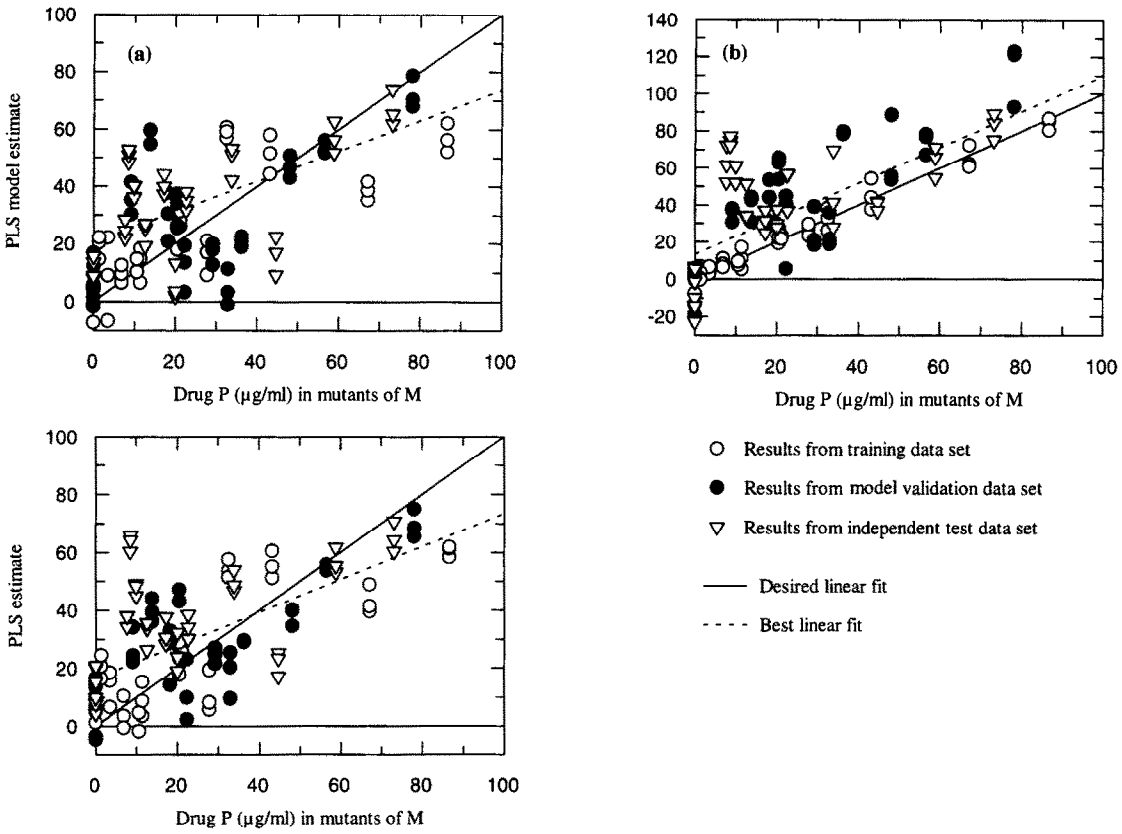

Fig. 4. Model predictions versus known amount of drug P in M for (a) the PLS model using all 150 regressor variables (%RMSEC=18.7; $\%RMSEP_{mv}$=15.9; $\%RMSEP_{its}$=26.8) (b) the best model produced by GAMLR-E (%RMSEC=4.71; $\%RMSEP_{mv}$=29.77; $\%RMSEP_{its}$=36.16), and (c) the best model produced by GAPLS-E (%RMSEC=18.15; $\%RMSEP_{mv}$=15.95; $\%RMSEP_{its}$=26.82). ○=Results from training data set, •=Results from model validation data set, ▽=Results from independent test data set, ———=Desired linear fit, - - - -=Best linear fit.

Table 7
Comparison of the quality of the MLR models produced by five runs of algorithm GAMLR-V on the drug M in P data. The aim of this variable selection genetic algorithm is to optimise the number of selected regressor variables for an MLR model given a maximum allowable model RMSEP. For this set of experiments the maximum RMSEP was set to 25% (Chosen by examining the results in Table 1). Values for the RMSEC, $RMSEP_{mv}$ and $RMSEP_{its}$ were calculated for each model and the $RMSEP_{its}$ used as the final measure of the utility of each model. The bold faced type indicates the optimal model

Expt. number	Number of variables	RMSEC	$RMSEP_{mv}$	$RMSEP_{its}$
1	15	9.908754	24.99203	40.00901
2	**21**	**6.08971**	**26.37244**	**29.81002**
3	17	8.060885	21.2908	36.17023
4	18	7.587493	24.09643	36.87597
5	15	9.870897	24.12415	45.63229

There was no strong correlation between the variable subset chosen in each of the five experiments (data not shown).

3.3.4. Algorithm GAPLS-V

Finally, the genetic algorithm GAPLS-V was applied to data set 1 five times using the same stopping criterion as GAMLR-V. Each run took approximately 6 h using a 90 MHz Intel Pentium processor. The maximum number of factors, *fact_max*, was 16. The threshold value for the $RMSEP_{mv}$ was set to 22%, the $RMSEP_{mv}$ for the PLS model built with all the regressor variables.

The results of the five runs are shown in Table 8. The best result produced a subset of 7 variables with a corresponding $RMSEP_{its}$ of 22.4%, and as before there was no strong correlation between the variable subset

Table 8
Comparison of the quality of the PLS models produced by five runs of algorithm GAPLS-V on the lysozyme/glycogen mixture data. The aim of this variable selection genetic algorithm is to optimise the number of selected regressor variables for a PLS model given a maximum allowable model RMSEP. For this set of experiments the maximum RMSEP was set to 22% (the $RMSEP_{mv}$ for the PLS model built using all variables). Values for the RMSEC, $RMSEP_{mv}$ and $RMSEP_{its}$ were calculated for each model and the $RMSEP_{its}$ used as the final measure of the utility of each model. The bold faced type indicates the optimal model

Expt. number	Number of variables	Number of factors	RMSEC	$RMSEP_{mv}$	$RMSEP_{its}$
1	6	2	23.2693	20.66878	24.88209
2	5	2	22.0307	19.54071	28.78681
3	7	**1**	**24.4616**	**22.66555**	**22.42222**
4	5	1	20.02679	21.60076	28.23605
5	6	1	24.29657	21.57894	24.41206
No selection [a]	150	2	18.69731	21.8501	23.65169

[a] The model statistics for a PLS model built using all the 150 available regressor variables.

chosen in each of the five experiments (data not shown).

The results from this algorithm are good. The number of variables was reduced significantly and the $RMSEP_{its}$ is even better than the models built using GAMLR-E, GAPLS-E and PLS model using all available variables.

4. Conclusion

Four optimising methods for variable selection in multivariate calibration have been described: one for determining the optimal subset of variables to give the best possible root-mean-square error of prediction (RMSEP) in a multiple linear regression (MLR) model, the second for determining the optimal subset of variables which produce a model with RMSEP less than or equal to a given value. Algorithms three and four were identical to algorithms one and two, respectively, except that this time they use a cost function derived from a PLS model rather than an MLR model.

Two data sets were used to verify the effectiveness of the four model selection methods. The first data set was obtained using pyrolysis mass spectrometry (PyMS) on binary mixtures of the protein lysozyme with glycogen [36]. The second data set was also created using PyMS, in this case for the analysis of fermentation broths for a drug of commercial interest [37].

The results obtained from applying the four optimising algorithms to the two data sets showed that the hybrid genetic algorithm/multivariate calibration is a valuable approach to the problem of model selection. Both the MLR and PLS models produced were of similar standard which suggests that once a suitable subset of orthogonal regressor variables has been selected nothing substantial is gained by projecting these variables onto a set of latent variables. When priority is given to reducing the number of regressor variables in a model rather than minimising the RMSEP both algorithms performed extremely well on average reducing the number of variables used from 150 to fewer than 20.

Acknowledgements

DB, AJ, JJR and DBK thank the Chemicals and Pharmaceuticals Directorate of the UK BBSRC, and RG thanks the Wellcome Trust (grant 0426151Z/94/Z) for financial support. We thank Dr Mark Neal for useful discussions.

References

[1] P.J. Brown, Measurement, Regression, and Calibration, Clarendon Press, Oxford, 1993.
[2] P.J. Brown, C.H. Spiegelman, M.C. Denham, Philosophical Transactions Of the Royal Society Of London Series a-Physical Sciences and Engineering, 337 (1991) 311–322.

[3] H. Mark, Appl. Spectros., 42 (1988) 1427–1440.
[4] H. Martens, T. Næs, Multivariate Calibration, Wiley, 1989.
[5] C.E. Miller, Chemometrics Intell. Lab. Syst., 30 (1995) 11–22.
[6] M.B. Seasholtz and B. Kowalski, Anal. Chim. Acta, 277 (1993) 165–177.
[7] A.J. Miller, Subset selection in regression, Chapman & Hall, London, 1990.
[8] D.B. Kell and B. Sonnleitner, Trends Biotechnol., 13 (1995) 481–492.
[9] W.J. Krzanowski, Principles of Multivariate Analysis - A User's Perspective, Oxford University Press, 1988.
[10] M.R.J. Garey, Computers and Intractability: A guide to the the theory of NP-Completeness, Freeman, 1979.
[11] D.E. Goldberg, Genetic Algorithms in search, optimization and machine learning, Addison-Wesley, 1989.
[12] J.H. Holland, Adaption in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, MIT Press, 1992.
[13] C.B. Lucasius and G. Kateman, Chemometrics Intell. Lab. Syst., 19 (1993) 1–13.
[14] C.B. Lucasius and G. Kateman, Chemometrics Intell. Lab. Syst, 25 (1993) 99–145.
[15] U. Horchner and J.H. Kalivas, Anal. Chim. Acta, 311 (1995) 1–13.
[16] D. Jouan-Rimbaud, D.L. Massart, R. Leardi and O.E. de Noord, Anal. Chem., 67 (1995) 4295–4301.
[17] C.B. Lucasius, M.L.M. Beckers and G. Kateman, Anal. Chim. Acta, 286 (1994) 135–153.
[18] F.E. Dowell, Optical Engineering, 33 (1994) 2728–2732.
[19] P.J. Brown, J. Chemometrics, 6 (1992) 151–161.
[20] U. Horchner and J.H. Kalivas, J. Chemometrics, 9 (1995) 283–308.
[21] J.H. Kalivas, N. Roberts and J.M. Sutter, Anal. Chem., 61 (1989) 2024–2030.
[22] N.J. Messick, U.B. Horchner, J.H. Kalivas, Abstracts Of Papers Of the American Chemical Society, 208 (1994) 136-CHED.
[23] J. Sutter, J.H. Kalivas, Abstracts Of Papers Of the American Chemical Society, 195 (1988) 193-CHED.
[24] J.M. Sutter, J.H. Kalivas, Abstracts Of Papers Of the American Chemical Society, 203 (1992) 24-COMP.
[25] R.G. Brereton and A.K. Elbergali, J. Chemometrics, 8 (1994) 423–437.
[26] F. Lindgren, P. Geladi, A. Berglund, S. Jostrom and S. Wold, J. Chemometrics, 9 (1995) 331–342.
[27] T.H. Wonnacott, R.J. Wonnacott, Regression: a second course in statistics, Wiley, 1981.
[28] G.A.F. Seber, C.J. Wild, Nonlinear regression, Wiley, New York, 1989.
[29] F.L. Bookstein, in: J. Kmenta, J.B. Ramsey (Eds.), Evaluation of econometric models, Academic Press, 1980, pp. 75–90.
[30] S. de Jong, J. Chemometrics, 7 (1993) 1–7.
[31] D.M. Allen, Technometrics, 13 (1971) 469–475.
[32] R.R. Picard and R.D. Cook, J. Amer. Statist. Assoc., 79 (1984) 575–583.
[33] R.R. Picard and K.N. Berk, The American Statistician, 44 (1990) 140–147.
[34] R.D. Snee, Technometrics, 19 (1977) 415–428.
[35] A. Jones, D.B. Kell and J.J. Rowland, Anal. Chim. Acta, submitted.
[36] R. Goodacre, M.J. Neal and D.B. Kell, Anal. Chem., 66 (1994) 1070–1085.
[37] R. Goodacre, S. Trew, C. Wrigley-Jones, M.J. Neal, J. Maddock, T.W. Ottley, N. Porter and D.B. Kell, Biotechnol. Bioeng., 44 (1994) 1205–1216.
[38] R. Goodacre, M.J. Neal, D.B. Kell, L.W. Greenham, W.C. Noble and R.G. Harvey, J. Appl. Bacteriol., 76 (1994) 124–134.
[39] R. Goodacre, S. Trew, C. Wrigley-Jones, G. Saunders, M.J. Neal, N. Porter and D.B. Kell, Anal. Chim. Acta, 313 (1995) 25–43.