Journal of Computer-Aided Molecular Design, 7 (1993) 587–619
ESCOM

J-CAMD 215

Sample-distance Partial Least Squares: PLS optimized for many variables, with application to CoMFA

Bruce L. Bush* and Robert B. Nachbar, Jr.

Merck Research Laboratories, Building 50SW-100, Merck & Co., Inc., 126 East Lincoln Avenue, Rahway, NJ 07065, U.S.A.

Received 7 December 1992
Accepted 22 March 1993

Key words: Partial least squares; Structure-activity relationship; Molecular modeling; CoMFA; Factor analysis

SUMMARY

Three-dimensional molecular modeling can provide an unlimited number m of structural properties. Comparative Molecular Field Analysis (CoMFA), for example, may calculate thousands of field values for each model structure. When m is large, partial least squares (PLS) is the statistical method of choice for fitting and predicting biological responses. Yet PLS is usually implemented in a property-based fashion which is optimal only for small m. We describe here a sample-based formulation of PLS which can be used to fit any single response (bioactivity). SAMPLS reduces all explanatory data to the pairwise 'distances' among n samples (molecules), or equivalently to an n-by-n covariance matrix **C**. This matrix, unmodified, can be used to fit all PLS components. Furthermore, SAMPLS will validate the model by modern resampling techniques, at a cost independent of m. We have implemented SAMPLS as a Fortran program and have reproduced conventional and cross-validated PLS analyses of data from two published studies. Full (leave-each-out) cross-validation of a typical CoMFA takes 0.2 CPU s. SAMPLS is thus ideally suited to structure–activity analysis based on CoMFA fields or bonded topology. The sample-distance formulation also relates PLS to methods like cluster analysis and nonlinear mapping, and shows how drastically PLS simplifies the information in CoMFA fields.

INTRODUCTION

Molecular modeling and statistical methods

As 3D molecular modeling becomes widely available to workers in structural biology, properties calculated from atomic coordinates are used increasingly to explain and predict biological activity. Molecular modeling can produce any desired number of explanatory descriptors for each structure – far more than the number of activity data to be explained. For example, Comparative

*To whom correspondence should be addressed.

Abbreviations: PLS, partial least squares; SAMPLS, sample-distance partial least squares; CoMFA, comparative molecular field analysis.

Molecular Field Analysis (CoMFA) [1,2] bases its predictions upon field values calculated at each point of a 3D grid around the molecular structures. The number m of explanatory variables may run into thousands, whereas the number n of compounds rarely exceeds 100. Moreover, field values are highly correlated, having been derived from a much smaller number of molecular descriptors such as atomic charges and positions. In this situation, conventional statistical methods like multiple regression are vulnerable to overfitting: producing a formula which fits the training data but is unreliable for prediction.

Linear regression by partial least squares (PLS) [3–5] is designed to avoid such overfitting. The method reduces the explanatory data to a small number of components, or linear combinations, which are strongly correlated with the responses. Since the pioneering work of H. and S. Wold [3a,b], PLS has been applied in the laboratories of S. Wold [6–8] and others [9,10] to a variety of experimental data, especially spectra of mixtures. It receives increasing use in chemometrics and pharmaceutical research; see for example de Meo et al. [11]. Within the molecular modeling package SYBYL [12], PLS is the recommended regression method for analysis of CoMFA fields.

The first PLS component (h = 1) is a 'trend vector' [13], or gradient of the observed response(s) in the space of explanatory properties. (We adopt the terminology of Geladi and Kowalski [4]: n samples, m explanatory properties per sample, h components.) The next component (h = 2) is the trend within a subspace orthogonal to the first; and so forth.

To avoid overfitting and produce a robust model, it is crucial to validate each component, stopping as soon as trends become ambiguous. Modern validation techniques repeat the analysis many times on different input data or different regroupings of samples [14,15]. In standard implementations of PLS, the process becomes time-consuming for large m.

As far as we know, all previous descriptions [4,5,16] and implementations [12,17] of PLS are *property-based.* They employ m-vectors and m-column matrices throughout each step of fitting, prediction, and validation. This approach is efficient and compact if m is small, as when analyzing a few experimental properties. When PLS is called upon to deal with thousands of properties, however, the property-based formulation is less efficient.

We have derived a *sample-based* implementation of PLS which we term 'SAMPLS'. This approach is based on an n-by-n matrix of 'distances' between pairs of samples, or, equivalently, a matrix of covariances (similarities). Applied to CoMFA fields, for example, a SAMPLS calculation which begins with this similarity matrix gives exactly the same results as a property-based PLS analysis of the CoMFA fields themselves. Full cross-validation of a CoMFA/PLS analysis, which takes over an hour under a standard PLS implementation, requires tenths of a second with SAMPLS. Time and space requirements increase only as n^2. Indeed, a typical CoMFA study could be fully validated on a programmable calculator.

In the course of a CoMFA study, the researcher may need to explore many conformations and superpositions of the compounds. The time consumed by cross-validation becomes a substantial hurdle. Some statistical packages (including SYBYL QSAR) therefore suggest partial cross-validation, which randomly assigns the n samples to a smaller number, g, of groups. However, this short-cut introduces uncontrolled variation which can complicate the search for a reliable model. During one of the first CoMFA studies in our laboratories, for example, the guiding quantity, cross-validated R^2, dropped suddenly from 0.26 to −0.19, simply because the compounds were reordered (K. Prendergast, unpublished results). The speed of SAMPLS allows routine use of leave-each-out cross-validation (g = n) and thus avoids this source of random effects.

Aside from its efficiency, the sample-distance approach provides insight into the workings of PLS regression, especially as applied to molecular design. The authors of the chemometric system SPECTRE [18] comment that '[t]he main disadvantage of the PLS method is that the latent variables are abstract and difficult to interpret. ...'. When the samples are chemical compounds, it is rarely possible to synthesize abstract combinations of molecular properties or field patterns. Instead, SAMPLS expresses its predictions in terms of displacements between real chemical groups, e.g., 'halfway between cyclohexyl and phenyl' or 'modify B in analogy to the change from A to methyl-A'. A sample-distance approach has recently been found to be useful even for analysis of symbolic data, namely peptide and protein sequences [19,20].

SAMPLS is fully equivalent to PLS. This fact demonstrates that partial least squares belongs to the family of data analysis techniques which depend on intersample 'distances' rather than on individual explanatory properties. Other linear techniques in this family include principal components analysis and regression (PCA, PCR) [4,19]; nonlinear techniques include clustering [19,21] and nonlinear mapping [22,23]. By contrast, conventional multiple regression and projection pursuit regression [24] fall outside the family discussed here. Those methods reweight individual explanatory properties, or combine them into non-additive 'interaction' terms.

Several scatterplots based on intersample distances in 'sample space' are illustrated below. A quick look at such a plot may help the researcher to choose an appropriate statistical method, whether linear (like PLS), or nonlinear (like clustering). Such a plot is a useful supplement to 3D maps of the molecular fields generated by CoMFA and by related methods [25].

In its current form, the sample-distance method applies only to the case of a single response. This restriction poses little difficulty for biological applications.

Plan of paper

This paper begins by summarizing the standard property-based form of PLS. We then reformulate this algorithm in terms of the n-by-n intersample covariance matrix $\mathbf{C} = \mathbf{X}\mathbf{X}^{\mathbf{T}}$, where $\mathbf{X}$ is the n-by-m matrix of explanatory data. We have implemented the property-based approach as an RS/1 [26] procedure '#PLS', and the sample-based approach as the stand-alone program 'SAMPLS'.

We test SAMPLS against RS/1 PLS and SYBYL PLS on data from two published studies. The first test is based on results of research at Fisons plc [27] into the bioeffectiveness of 14 anti-allergics, analyzed in terms of m = 3 physicochemical properties. In the second study, a CoMFA analysis of the binding of 21 steroids to human corticosteroid-binding globulins (CBGs) [1,28], m is approximately 1000. SAMPLS begins with a distance matrix generated by SYBYL from the CoMFA fields, whereas SYBYL QSAR uses the fields directly.

All of these PLS methods give identical results, as long as the data are consistently scaled throughout fitting and cross-validation. Three-axis scatterplots illustrate how these PLS results derive from the 'distances' between test compounds.

The Discussion compares the statistics generated by the two equivalent approaches to PLS. The simplicity of the sample-distance approach affords insights into the workings of PLS, revealing some of its limitations. In particular, biological applications may call for *non*linear techniques; we suggest directions for combining the strengths of the PLS and similarity analysis. We then explore the sensitivity of PLS to CoMFA parameters and its *in*sensitivity to details of molecular structure. We conclude with a few words on efficiency.

METHODS

The following overview defines the nomenclature of this paper, and may provide a useful perspective on PLS.

Overview and nomenclature

Partial least squares is an iterative process in which each *cycle* generates a *component* (h = 1, 2, 3, ...) of the regression model. Each component *fits* a portion of the original data in the *training set*, leaving a smaller *residual* for the next cycle, and generating numerical coefficients which can be used in subsequent predictions. The residual may be expressed in the quantity *R-squared*, or R^2, which increases toward unity ($R^2 = 1$) as the residuals vanish.

If continued until no further changes occur in coefficients or residuals, the process generates a *full least squares* (LS) fit. At this point, h equals the number of independent explanatory properties, which is generally the lesser of n (the number of samples) and m (the number of explanatory data per sample).

PLS (*partial* least squares) stops at a smaller value of h chosen by the researcher. The rationale for stopping is that on each cycle the prediction coefficients become more vulnerable to 'noise'. Although the fit to the training set improves, predictions will probably become worse. With a well-chosen number of components, PLS can often strike a useful balance between precision in fitting the training set and robustness in predicting new samples.

Many workers choose that stopping value of h by *cross-validation* [14]. This simulates actual prediction by subdividing the training samples into g *groups* or subsets. Each group is 'predicted' on the basis of a fit to the other groups. The scatter among quasi-predictions is summarized by the *cross-validated* or '*predictive*' R^2 defined below. When h is too large, this quantity decreases, indicating 'overfitting'. Partial cross-validation involves some randomness in assigning samples to groups. In *full (leave-each-out) cross-validation*, however, each of the n samples is omitted exactly once, so that no randomness affects the assignments. Even this level of cross-validation may not suffice when the training samples are analogs of a few carefully selected 'leads' rather than being independently chosen – a common circumstance in synthetic medicinal chemistry. Such situations may call for *leave-k-out* cross-validation, which requires even more trials (of order n^k).

Finally, *bootstrapping* [14] is a closely related method which estimates the reliability of individual predictions or statistical parameters. Bootstrapping entails resampling in sets of n, allowing samples to enter a set more than once. The resampling is expensive and, as a practical matter, necessarily random: systematic coverage would require at least an astronomical $[(2n-1)!/(n-1)!n!]$ resamplings (approximately 2^n). Clearly, all of these techniques can benefit greatly from an efficient method for repeating the regression on different sets of samples.

Notation: We adopt the convention that data are arranged with one sample in each of n rows; explanatory properties are arrayed in m columns. We use lower-case bold for column vectors (**y**, **t**, **w**, **p**), and upper-case bold for n-by-m and n-by-n matrices (**X**, **C**, **D**). Superscript **T** denotes transposition, yielding row vectors ($\mathbf{y^T}$, $\mathbf{w^T}$) and m-by-n matrices ($\mathbf{X^T}$). Roman symbols are scalars or matrix elements (a, b, β, c*, y*, D_{ij}). Each PLS cycle (h) begins with an n-by-m block $\mathbf{X}_h$ of explanatory data, and an n-by-1 matrix (column vector) $\mathbf{y}_h$ of responses. At the end of cycle h,

both $\mathbf{X}_h$ and $\mathbf{y}_h$ are updated, leaving residuals $\mathbf{X}_{h+1}$ and $\mathbf{y}_{h+1}$ for use in the next cycle. A summary measure of the residual after cycle h is

$$R^2 = 1-(\mathbf{y}_{h+1})^2/(\mathbf{y}_h)^2$$

The cycles end at a specified h(max), or when $\mathbf{y}_{h+1}$ is sufficiently small.

The procedures described below use *centered* property values, which are deviations from the true average value of that property over the *fitted* (training) samples. Whereas centering is only a matter of bookkeeping, *scaling* the property columns affects the results of PLS or principal components regression. SAMPLS allows any desired scaling of the data, but it cannot rescale during validation.

Standard property-based formulation of PLS

A concise description of property-based PLS appears in the appendix to the tutorial by Geladi and Kowalski [4]. The general PLS algorithm fits multiple responses (a 'Y-block') self-consistently. This requires a convergent inner iteration within each PLS cycle. Höskuldsson [5] briefly describes the simpler case of one response variable. The latter article scales some internal quantities differently, step 5 below, but its conventions result in the same updates (steps 7–8) and predictions (step 9). We follow Geladi's conventions.

Property-based PLS: fitting

Initialize $\mathbf{X}$, $\mathbf{y}$: centered values of properties

Cycle: for h = 1, 2, 3, ..., h(max)

$\mathbf{w} = \mathbf{X}^T \mathbf{y}/a$ (1)

where a is chosen to normalize $\mathbf{w}$:

$\mathbf{w}^T \mathbf{w} = 1$; $a^2 = \mathbf{y}^T \mathbf{X} \mathbf{X}^T \mathbf{y}$ (2)

$\mathbf{t} = \mathbf{X} \mathbf{w}$ (3)

$\mathbf{p} = \mathbf{X}^T \mathbf{t}/(\mathbf{t}^T \mathbf{t})$ (4)

Rescale: Geladi convention (has no effect on fit or prediction):

$p = |\mathbf{p}|$; that is, $p^2 = \mathbf{p}^T \mathbf{p}$

$\mathbf{w} = p\mathbf{w}$

$\mathbf{t} = p\mathbf{t}$

$\mathbf{p} = \mathbf{p}/p$ (5)

$b = (\mathbf{t}^T \mathbf{y})/(\mathbf{t}^T \mathbf{t})$ (6)

Save for prediction: b_h, $\mathbf{p}_h$, $\mathbf{w}_h$

Update $\mathbf{X}_{h+1} = \mathbf{X}_h - \mathbf{t}_h \mathbf{p}_h^T$ (7)

Update $\mathbf{y}_{h+1} = \mathbf{y}_h - b_h \mathbf{t}_h$ (8)

End.

The prediction cycle begins with a single m-vector $\mathbf{x}^*$ of explanatory properties. (The asterisk indicates a quantity related to one prediction sample. Each property value is centered with respect to the *training* set.) The predicted response y*, also centered with respect to the *training* set, is built up component by component.

Property-based PLS: prediction

Initialize: $y^* = 0$; $\mathbf{x}^*$ = explanatory data (centered)

Cycle: for h = 1, 2, 3, ..., h(max)

Increment prediction: $y^* = y^* + b\ \mathbf{x}^{*T}\ \mathbf{w}$ (9)

Update explanatory data: $\mathbf{x}^{*T} = \mathbf{x}^{*T} - \mathbf{x}^{*T}\ \mathbf{w}\ \mathbf{p}^T$ (10)

End.

The predicted response for each sample is thus independent of other prediction samples, and is linear in the property vector $\mathbf{x}^*$. This prediction cycle can also be applied formally to a *training* sample, to regenerate the fitted response value y. This fitted response is not linear in the properties of the training sample, however, since the properties of all the training compounds enter implicitly into **w** and **p**.

By combining steps above, we can calculate vectors **p** and **w** from one another through the m-by-m matrix $(\mathbf{X}^T\ \mathbf{X})$. This is the covariance of explanatory properties, which can be used for principal components analysis and in experimental design [4,5]. Matrix $(\mathbf{X}^T\ \mathbf{X})$ changes from cycle to cycle as **X** is updated, so there is no advantage to precomputing it.

A clue to avoiding all matrix updates appears when we rewrite steps 7 and 8 entirely in terms of **t**:

Update $\mathbf{X} = \mathbf{X} - (1/\mathbf{t}^T\ \mathbf{t})\ \mathbf{t}\ \mathbf{t}^T\ \mathbf{X}$ (7')

Update $\mathbf{y} = \mathbf{y} - (1/\mathbf{t}^T\ \mathbf{t})\ \mathbf{t}\ \mathbf{t}^T\ \mathbf{y}$ (8')

(This form is valid regardless of how **t** is normalized.) Both updates thus consist of projecting away from each column of **y** and **X** any portion (component) parallel to **t**. The update in the prediction cycle works similarly. This makes clear why each new component **t** is orthogonal to those preceding. It also offers a way to eliminate both m-vectors, **w** and **p**, from PLS, and leads directly to a formulation in n-space which avoids matrix updates entirely.

Sample-distance formulation of PLS

The sample-distance approach (SAMPLS) combines PLS steps to make use of the *intersample* covariance $\mathbf{C} = \mathbf{X}\ \mathbf{X}^T$. This n-by-n matrix, applied to the n-vector **y** of responses, yields **t** directly, bypassing all m-vector calculations. (**X** is normally used in centered form, so that **C** is truly a covariance. However, the SAMPLS algorithm below does not require that the columns of **X** be centered.) Matrix **C** is essentially an overlap matrix (see Eq. 11'). It can also be calculated from the matrix **D** of intersample distances, as discussed below.

Sample-based PLS: fitting

Initialize: $\mathbf{y}_1 = \mathbf{y}_0$: observed responses (centered).

Calculate the covariance matrix

$\mathbf{C} = \mathbf{X}\ \mathbf{X}^T$ (11)

that is, for each pair of samples (i,j), sum over all properties k:

$C_{ij} = \Sigma_k\ X_{ik}\ X_{jk}$ (11')

Keep **X** for prediction only.

Cycle: for h = 1, 2, 3, ..., h(max)
Working vector: $\mathbf{s} = \mathbf{C}\,\mathbf{y}$ (12)
Center **s**, and
Orthogonalize **s** to previous $\mathbf{t}_g$:
For g = 1, ..., (h-1),
$\mathbf{s} = \mathbf{s} - (\mathbf{t}_g^T\mathbf{s} / \mathbf{t}_g^T\,\mathbf{t}_g)\,\mathbf{t}_g$ (13)
Keep during the fitting cycle:
$\mathbf{t}_h = \mathbf{s}$;
$t_h^2 = (\mathbf{t}^T\,\mathbf{t})$;
$\beta = \mathbf{t}^T\,\mathbf{y} / t^2$ (14)
Save for prediction:
$\beta_h = \beta$
$\mathbf{y}_h = \mathbf{y}$
$\mathbf{z}_h = \mathbf{C}\mathbf{t}_h$
$\alpha_h = \mathbf{z}_h^T\mathbf{t}_h$
Update $\mathbf{y}_{h+1} = \mathbf{y}_h - \beta_h\,\mathbf{t}_h$ (15)
No need to update: **X** or **C**.
End.

Scalar β corresponds to b in the property-based algorithm, but it is scaled differently, as is vector **t**.

Predictions can be made for one sample at a time, as before. First convert the m-vector of explanatory data **x*** into an n-vector $\mathbf{c}^* = \mathbf{X}\,\mathbf{x}^*$, which relates the new sample to the training samples. Only n-dimensional quantities are needed during the cycle.

Sample-based PLS: prediction

If the explanatory data for a prediction sample are available at the time of the regression, the predicted response y* can be evaluated by carrying **c*** as an extra row of the **C** matrix. This is the method used by program SAMPLS. The steps are very similar to steps 12–15 of the regression. Scalars s* and t* play roles paralleling those of **s** and **t**. While orthogonalizing **s** to the previous $\mathbf{t}_h$, we also subtract from s* the corresponding fraction of each earlier t*. The resulting t* for the new cycle is then scaled by β to yield the current component y* of the prediction. In this method, vector **z** is not needed.

Begin prediction: y* = 0 (centered on average over training set)
Cycle: for h = 1, 2, 3, ..., h(max)
Working scalar for prediction sample: $s^* = \mathbf{c}^{*T}\mathbf{y}$ (12')
During orthogonalization, subtract from s*:
For g = 1, ..., (h-1),
$s^* = s^* - (\mathbf{t}_g^T\mathbf{s}/\mathbf{t}_g^T\mathbf{t}_g)\,t^*_g$ (13')
Keep during the prediction cycle:
$t^*_h = s^*$ (14')
Build up prediction: $y^*_{h+1} = y^*_h + \beta_h\,t^*_h$ (15')

Alternatively, if prediction is performed at a later time, the prediction cycle formally resembles the property-based method, using the saved vectors **y** and **z** instead of **w** and **p**. (Note that $\mathbf{y}_h$ is the residual saved from the *beginning* of fitting cycle h.) The covariance vector must be decremented after each cycle.

Initialize: $y^* = 0$;
  $\mathbf{x}^*$ = explanatory data (centered as was **X**);
  $\mathbf{c}^* = \mathbf{X}\,\mathbf{x}^*$ = covariance vector
Cycle: for h = 1, 2, 3, ..., h(max)
  Increment prediction: $y^* = y^* + \beta_h\,\mathbf{c}^{*T}\,\mathbf{y}_h$ (16)
  Decrement covariance vector:
    $\mathbf{c}^{*T} = \mathbf{c}^{*T} - (1/\alpha_h)\,\mathbf{c}^{*T}\,\mathbf{y}_h\mathbf{z}_h^T$ (17)
End.

An even more compact form is possible. Sum, over all components, the factors multiplying the *original* $\mathbf{c}^{*T}$:

$$\begin{aligned}\mathbf{v} = {} & \beta_1\mathbf{y}_1 \\ & + \beta_2(1-(1/\alpha_1)\mathbf{y}_1\mathbf{z}_1^T)\mathbf{y}_2 \\ & + \beta_3(1-(1/\alpha_1)\mathbf{y}_1\mathbf{z}_1^T)(1-(1/\alpha_2)\mathbf{y}_2\mathbf{z}_2^T)\mathbf{y}_3 \\ & + \ldots\end{aligned} \tag{18}$$

The result is a single n-dimensional *prediction vector* **v**. It summarizes the PLS model; the predicted activity of any new sample, based on its covariance vector $\mathbf{c}^{*T}$, is just $(\mathbf{c}^{*T}\mathbf{v})$.

The SAMPLS program can perform full least-squares regression and principal components regression, as well as PLS. In full least squares, the PLS cycles are continued until $\beta_h = 0$, at which point **v** is just $(\mathbf{C}^{-1}\,\mathbf{y}_0)$. In general this occurs when h exceeds the lesser of m and n. To perform principal components regression, SAMPLS iterates steps 12 and 13 above, normalizing **t** and feeding it back in place of **y** until no further changes occur. This stationary point for cycle h is independent of $\mathbf{y}_h$, and is a characteristic of the unmodified **C**, being its eigenvector of h-th greatest eigenvalue. The rest of the fitting cycle proceeds as shown, as does the prediction cycle.

A key to the efficiency of SAMPLS is that neither **C** nor **X** is updated during fitting or prediction. An informal explanation runs as follows. One could indeed update **X** in each cycle. Then, to maintain the definition $\mathbf{C} = \mathbf{X}\,\mathbf{X}^T$, one would update **C** by projecting **t** out of all rows and columns, as in Eq. 7':

(*Not necessary*:) $\mathbf{C} = (\mathbf{1}-\mathbf{t}\mathbf{t}^T)\,\mathbf{C}\,(\mathbf{1}-\mathbf{t}\mathbf{t}^T)$

However, this update step is unnecessary as long as **t** is updated, as may be seen from the following informal line of reasoning. (See the Appendix for a more formal proof that SAMPLS is equivalent to standard PLS regression.) **C** enters into the fit only through the vectors $\mathbf{t}_h$. The product (**C y**) (step 12) is the same whether or not **C** is updated, since all previous components have been projected from **y** (step 15). Step 13 then projects those components out of this product. (Centering **t** effectively projects out the zero-th component, a constant vector.) The resulting $\mathbf{t}_h$ is the same within a scalar factor as in the property-based algorithm (step 5).

Even more important for efficiency is that SAMPLS must also *validate* the model without modifying **C** or referring to **X**. Clearly, validation methods which vary only the response data require only the original **C**. Cross-validation and bootstrapping can also be performed without repeatedly modifying **C**. Both of these procedures merely reweight the samples of the training set, using integer weights. Whenever SAMPLS calculates a vector or matrix product or an average over samples, it must form weighted sums over the n samples. The sum of weights appears in the denominator of any average. In particular, each sample (row) must be weighted when centering **y** initially, and when centering and orthogonalizing **t** during each cycle of PLS. Once **t** and **y** have been recentered, there is no need to recenter rows or columns of **C**.

Although reweighting *samples* (rows) poses no difficulty, SAMPLS cannot efficiently reweight or rescale individual *properties* (columns) of **X**. This would change the relative importance of the properties, and require that **C** be recalculated from **X**. During validation, for example, the SYBYL QSAR module rescales all properties over each revalidation group. SAMPLS cannot reproduce the results of this procedure. Strong arguments indicate that such rescaling is undesirable in principle (see Discussion).

SAMPLS can begin with intersample distances **D** instead of the covariance matrix **C**. Define intersample distance as a sum over the properties k = 1, 2, ..., m:

$$D^2_{ij} = (D_{ij})^2 = \Sigma_k (X_{ik} - X_{jk})^2 \tag{19}$$

D_{ij} is simply the Euclidean distance between the property-vectors of samples i and j. In CoMFA applications, the sum over k involves all the grid points and all the field types (steric, electrostatic, and so on). Workers in pattern analysis sometimes call D_{ij} the *Mahalanobis distance* [29] of order 2.

A distance matrix **D** may be available independently of the explanatory properties **X**, or we may wish to perform a PLS analysis based on a set of distances *as if* they had been derived from properties. In either case, **C** can easily be derived from **D**. The formula is given by Lewi and Moereels [19] in the context of cluster analysis, and by Crippen in connection with 'distance geometry embedding' [30], where **C** is called the *metric matrix*. Only squared distances enter the calculation:

$$C_{ij} = -(1/2)\,[(D_{ij})^2 - \langle (D_{i*})^2 \rangle - \langle (D_{*j})^2 \rangle + \langle\langle (D_{**})^2 \rangle\rangle] \tag{20}$$

Brackets <...> indicate averages over row i or column j, and double brackets <<...>> an average over all i and j.

Likewise, all information needed to predict the response of the new sample resides in the vector **d*** of distances from a new sample to the training samples. This vector is easily converted to **c*** for use as described above, without reference to the training data **X**. In this sense, PLS prediction, like the PLS fit, occurs entirely in the subspace of the training samples, no matter how many properties m are involved.

Implementations

RS/1 implementation of PLS

We have implemented PLS as an interactive RS/1 procedure ('#PLS'). The dialog is patterned

after the supplied RS/1 linear regression procedure $FITMULTIPLE. Internally, #PLS uses the prescriptions of Geladi and Kowalski for fitting and prediction. Unlike SAMPLS, #PLS allows for several response properties in the 'Y-block'. Centering and scaling are controlled by the user. To assist in choosing the number of components h for the final conventional PLS regression, two distinct measures of goodness of fit are provided when each component is extracted from the X-and Y-blocks. The first is a sequential F-test; the second is a predictive error sum of squares based on cross-validation. The user specifies g, the number of cross-validation groups. Because partial cross-validation (g less than n) requires that groups be defined randomly, only the results of *full* cross-validation can be expected to agree with SYBYL or other programs. As in SAMPLS, cross-validation is performed *without* rescaling.

SAMPLS

Program SAMPLS is written in Fortran-77, and runs under both VAX VMS and Silicon Graphics Irix operating systems. Calculations are performed on single-precision (REAL*4) arrays, accumulating products in double precision. Output can be formatted as annotated tables or can be directed to files in a simplified format for analysis and plotting.

Response data are read first, in a flexible format which can handle table files written from SYBYL QSAR. The response data may come from their own file, or may be one column in the file of explanatory properties. To omit a sample from the fitting phase, the user marks its line in this file of response data. Such omitted samples are always included in the prediction set, at negligible cost of time.

The second step is to read either a file of explanatory properties, or a file of precomputed intersample distances. If the former, the user may designate omitted properties interactively. In either case, the program calculates **C** while compressing the omitted samples.

The user now specifies h(max), chooses whether to scale the properties, and selects either PLS or PCR regression. In contrast to the SYBYL command or our RS/1 procedure, there is no need to re-run SAMPLS for different choices of h. SAMPLS always performs complete 'leave-each-out' cross-validation. Predictions and residuals are tabulated for all h, allowing the user to choose an appropriate column.

SYBYL QSAR: PLS (including CoMFA)

All SYBYL procedures described below employed the QSAR module from SYBYL version 5.41 for VAX/VMS.

Test computations

All calculations were performed on VAX computers running version 5.4-1A of the VMS operating system. Timings for SYBYL and SAMPLS were measured on a 23-MIPS VAX 8840.

The three programs performed unscaled and scaled PLS regressions to anti-allergic potency (LOGPR) for n = 14 compounds, as described by Gould et al. [27] and listed in Table 1. These were analyzed up to h(max) = m = 3 components, with full 14-group cross-validation. Molecular properties SIGMA, LOGP, MR, and LOGPR were used exactly as shown (e.g., LOGPR values to two decimal places).

The second test was a CoMFA study of data reported by Dunn et al. [28] on binding of 21

steroids to corticosteroid-binding globulin (CBG). (See column 'y(obs)' of Table 8.) CoMFA fields were recalculated by SYBYL directly from the superposed molecules, following the protocol in the SYBYL tutorial, described in Cramer et al. [1]. Molecular structures and command scripts were taken directly from the SYBYL demonstration files. Steric and electrostatic fields on a 2-Å grid provided a total of 996 explanatory variables per compound. SYBYL and SAMPLS performed PLS analysis up to h(max) = 4 components with full 21-group cross-validation. All calculations used options MINIMUM_SIGMA = 0.0, SCALING = NONE.

The SYBYL calculations used these CoMFA fields directly. To allow us to make a direct comparison with SAMPLS, Dr. David Patterson of Tripos Associates, Inc. kindly created a modified version of SYBYL which generates a distance matrix file from the CoMFA fields. He also provided an example of the file itself for the steroid CoMFA example (see Table 2).

The distributed version of SYBYL can provide the data needed by SAMPLS indirectly, if need be. We used SYBYL QSAR to compute principal components, or 'factors'. An auxiliary program converted a listing of f factors, each scaled by the square root of its eigenvalue, into intersample distances. In the limit f = n = 21, these distances agreed with the specially provided distance matrix (data not shown).

RESULTS

Anti-allergic potency

In the unscaled analyses, all three programs give identical results. Table 3 reproduces the output of the SAMPLS run, showing cross-validated and conventional fits for all three

TABLE 1
ANTI-ALLERGIC COMPOUNDS: POTENCY AND EXPLANATORY PROPERTIES

	LOGPR[a]	SIGMA[b]	LOGP[b]	MR[b]
1 01-NHEt	−2.15	−0.61	4.399	9.633
2 02-NHC_5H_{11}	−1.28	−0.61	5.986	11.024
3 03-NHCOMe	−1.19	0.00	3.390	9.669
4 04-H	−1.00	0.00	3.301	8.337
5 05-Dioxinyl	−0.75	−0.05	2.243	10.321
6 06-CHO	−0.63	0.42	3.197	8.836
7 07-Me	−0.42	−0.17	3.950	8.800
8 08-OEt	−0.40	−0.24	4.233	9.417
9 09-NHPh	−0.15	−0.40	5.849	11.216
10 10-NHCONHPh	−0.05	−0.20	5.119	12.085
11 11-Br	0.02	0.23	4.210	9.113
12 12-Cl	0.04	0.23	4.060	8.828
13 13-SEt	0.35	0.03	4.655	10.070
14 14-CF_3	0.40	0.54	4.270	8.847

[a] Bioactivity ($\log_{10}$(potency ratio)), as described in Gould et al. [27].
[b] Physicochemical properties of substituents (Ref. 27): SIGMA = Hammett sigma (electronic), LOGP = log partition coefficient (octanol:water), MR = molar refractivity.

TABLE 2
STEROID CoMFA INTERSAMPLE DISTANCES

2	1	66.8529	11	3	63.5424	15	4	69.0803	18	6	117.617	20	18	77.1764
3	1	74.6242	11	4	78.1931	15	5	87.4966	18	7	121.718	20	19	54.5375
3	2	66.6293	11	5	114.93	15	6	102.788	18	8	137.985	21	1	109.089
4	1	103.865	11	6	113.722	15	7	82.3854	18	9	69.1177	21	2	101.355
4	2	87.6198	11	7	117.036	15	8	118.095	18	10	80.3767	21	3	104.025
4	3	96.9566	11	8	126.419	15	9	64.8757	18	11	49.5414	21	4	79.995
5	1	127.741	11	9	56.4095	15	10	79.106	18	12	101.479	21	5	103.372
5	2	122.401	11	10	66.5727	15	11	88.8906	18	13	60.7083	21	6	108.69
5	3	123.73	12	1	115.237	15	12	69.0339	18	14	79.8342	21	7	83.373
5	4	107.952	12	2	101.483	15	13	93.8933	18	15	96.314	21	8	111.007
6	1	129.838	12	3	107.263	15	14	55.3131	18	16	92.6599	21	9	75.3868
6	2	121.363	12	4	59.6872	16	1	96.9244	18	17	109.669	21	10	82.4841
6	3	125.555	12	5	111.26	16	2	85.103	19	1	92.7374	21	11	94.012
6	4	116.674	12	6	119.315	16	3	94.1781	19	2	70.6926	21	12	79.5935
6	5	79.2901	12	7	107.814	16	4	61.1376	19	3	80.2013	21	13	93.9718
7	1	129.15	12	8	106.725	16	5	108.863	19	4	97.1336	21	14	71.3946
7	2	123.987	12	9	91.4603	16	6	114.72	19	5	126.153	21	15	41.3574
7	3	125.184	12	10	101.91	16	7	88.9952	19	6	125.79	21	16	57.1387
7	4	104.453	12	11	93.1318	16	8	105.287	19	7	128.864	21	17	70.8087
7	5	53.0272	13	1	69.56	16	9	92.8868	19	8	141.805	21	18	106.176
7	6	76.2312	13	2	54.7877	16	10	99.3481	19	9	84.2121	21	19	109.933
8	1	139.563	13	3	63.3057	16	11	76.213	19	10	82.5945	21	20	107.167
8	2	131.754	13	4	87.1788	16	12	78.9393	19	11	72.8225			
8	3	135.015	13	5	118.98	16	13	80.9603	19	12	109.331			
8	4	114.462	13	6	113.868	16	14	44.6859	19	13	81.0736			
8	5	140.879	13	7	118.026	16	15	68.9679	19	14	89.7876			
8	6	140.82	13	8	124.814	17	1	122.484	19	15	103.717			
8	7	131.448	13	9	65.6764	17	2	109.53	19	16	97.5441			
9	1	87.7785	13	10	72.649	17	3	115.709	19	17	116.213			
9	2	68.2268	13	11	41.6685	17	4	73.0613	19	18	72.24			
9	3	79.3106	13	12	100.715	17	5	111.175	20	1	83.7501			
9	4	91.0443	14	1	94.8851	17	6	116.032	20	2	67.6633			
9	5	104.211	14	2	81.3948	17	7	97.5418	20	3	67.01			
9	6	103.297	14	3	88.4686	17	8	104.014	20	4	100.355			
9	7	108.075	14	4	46.772	17	9	95.9127	20	5	128.674			
9	8	131.85	14	5	91.0853	17	10	101.875	20	6	131.243			
10	1	95.6638	14	6	107.361	17	11	101.492	20	7	129.888			
10	2	78.2871	14	7	87.8148	17	12	58.0576	20	8	138.334			
10	3	86.5496	14	8	114.181	17	13	104.961	20	9	86.1714			
10	4	101.582	14	9	83.9364	17	14	78.0609	20	10	81.5721			
10	5	113.591	14	10	95.7627	17	15	73.0677	20	11	67.3914			
10	6	112.094	14	11	71.032	17	16	75.6158	20	12	111.896			
10	7	114.146	14	12	69.2299	18	1	69.5306	20	13	75.0692			
10	8	136.517	14	13	79.7811	18	2	32.4343	20	14	92.188			
10	9	38.4303	15	1	108.294	18	3	68.7743	20	15	105.121			
11	1	67.9422	15	2	97.4114	18	4	87.3675	20	16	94.6855			
11	2	38.9469	15	3	100.387	18	5	118.519	20	17	119.312			

Distance in Eq. 19 for each pair of compounds in Table 8.

components. Table 4 assembles the output of six runs of SYBYL PLS which provide the corresponding information. The RS/1 procedure gives identical results (not shown). Cross-validation indicates that the PLS models with h = 1 and 2 components are worse than purely random (negative cross-validated R^2). Even full LS fit, h = m = 3, gives a mediocre cross-validated R^2 of 0.30.

Rescaling the explanatory data of Table 1, so that each property has unit variance, substantially changes the relative importance of the properties and produces a much better PLS fit. Again, SAMPLS and the property-based RS/1 procedure give identical regressions (Tables 5 and 6). SYBYL also gives identical results (Table 7) for the final PLS regression. A small discrepancy does appear in the cross-validation. SYBYL gives cross-validated R^2 = (0.09, 0.39, 0.31) for the three components, as against (0.11, 0.41, 0.31) given by the other two programs. This discrepancy has nothing to do with the PLS regressions, which are completely equivalent. Rather, it reflects the way all SYBYL statistical procedures implement the option SCALING=AUTOSCALE during validation. We verified, by explicitly omitting each of the 14 samples in turn, that SYBYL (version 5.41) rescales the property columns anew over each 13-sample subset. (This has no effect on the 3-component PLS model, h = 3, which is a full least squares fit and is thus unaffected by scaling.) By contrast, SAMPLS and RS/1 PLS retain the same scaling throughout – arguably a more consistent procedure, as discussed below.

Steroid CoMFA

SAMPLS and SYBYL give exactly the same results. The issue of rescaling does not arise because CoMFA field data are not scaled. The output from the SAMPLS run, which is based on the distance matrix, is reproduced in Table 8. The cross-validated R^2 of 0.687 for two components decreases slightly when further components are added. SYBYL results for the four-component cross-validation and the final two-component regression are assembled in Table 9. The program discards the cross-validation predictions for h = 1, 2, 3, offering only the summary statistics shown in the table.

Timing

All programs handled the anti-allergic example in a few seconds of CPU time: SAMPLS in one run, the other programs in six separate runs. In the more demanding Steroid CoMFA test, SYBYL performed a single conventional PLS regression in 134 CPU s. A full 21-group, 4-component cross-validation took essentially 21 times as long: 2912 CPU s, or well over an hour of real time. SAMPLS required 0.19 CPU s to calculate and write out the full analysis.

Visual representations

Quasi-3D scatterplots of these test results show how PLS and PCR regression make use of intersample distances. Figures 1 and 2 depict the unscaled and scaled analyses of the anti-allergic compounds. The horizontal plane represents the explanatory data. Each sample is positioned in terms of the two dominant 'factors' of the covariance matrix **C**, that is, the principal components PC1 and PC2. These positions retain the actual intersample distances rather faithfully, since the

TABLE 3
ANTI-ALLERGIC COMPOUNDS: SAMPLS ANALYSIS OF UNSCALED DATA

PLS fit of 14 samples by 3 components...

	y(obs):	Average	Variance	SD
		−0.515	0.505	0.711
Cross-validated fit SAMPLS v0.5				
Scaling option: None				
R^2:		−0.6022	−0.1183	0.3087
SE of estimate:		0.9365	0.8172	0.6739

Sample	y(obs)	Prediction or Fit (ih = ..)		
		1	2	3
1 01-NHEt	−2.150	−0.417	−0.518	−1.272
2 02-NHC_5H_{11}	−1.280	0.111	−0.045	−0.550
3 03-NHCOMe	−1.190	−0.809	−0.669	−0.589
4 04-H	−1.000	−0.321	−0.753	−0.946
5 05-Dioxinyl	−0.750	−1.658	−1.876	−1.658
6 06-CHO	−0.630	−0.559	−0.087	0.147
7 07-Me	−0.420	−0.737	−0.698	−1.151
8 08-OEt	−0.400	−0.641	−0.730	−1.019
9 09-NHPh	−0.150	−0.796	−0.242	−0.487
10 10-NHCONHPh	−0.050	−1.280	−0.849	−0.051
11 11-Br	0.020	−0.490	−0.026	−0.044
12 12-Cl	0.040	−0.593	−0.084	−0.173
13 13-SEt	0.350	−0.322	−0.205	−0.178
14 14-CF_3	0.400	−0.575	0.726	0.691

Conventional fit SAMPLS v0.5				
Scaling option: None				
R^2:		0.2631	0.4749	0.6661
SE of estimate:		0.6351	0.5600	0.4683

Sample	y(obs)	Prediction or Fit (ih = ..)		
		1	2	3
1 01-NHEt	−2.150	−0.782	−1.129	−1.606
2 02-NHC_5H_{11}	−1.280	−0.116	−0.575	−0.813
3 03-NHCOMe	−1.190	−0.816	−0.822	−0.674
4 04-H	−1.000	−0.860	−0.660	−0.959
5 05-Dioxinyl	−0.750	−1.319	−1.543	−1.032
6 06-CHO	−0.630	−0.636	−0.236	−0.031
7 07-Me	−0.420	−0.696	−0.640	−1.006
8 08-OEt	−0.400	−0.619	−0.690	−0.938
9 09-NHPh	−0.150	−0.039	−0.396	−0.388
10 10-NHCONHPh	−0.050	−0.211	−0.620	−0.050
11 11-Br	0.020	−0.334	−0.038	−0.034
12 12-Cl	0.040	−0.398	−0.067	−0.138
13 13-SEt	0.350	−0.270	−0.233	−0.116
14 14-CF_3	0.400	−0.116	0.439	0.575

TABLE 4
ANTI-ALLERGIC COMPOUNDS: SYBYL PLS ANALYSIS OF UNSCALED DATA

Cross-validation			
R^2:	−0.60221	−0.11828	0.30868
SE of predictions (cross-validated):	0.93653	0.81721	0.67390
	LOGPR_cv (1)	LOGPR_cv (2)	LOGPR_cv (3)
1 01-NHEt	−0.41750	−0.51836	−1.27159
2 02-NHC_5H_{11}	0.11122	−0.04531	−0.54969
3 03-NHCOMe	−0.80870	−0.66874	−0.58915
4 04-H	−0.32064	−0.75301	−0.94583
5 05-Dioxinyl	−1.65761	−1.87581	−1.65796
6 06-CHO	−0.55858	−0.08699	0.14686
7 07-Me	−0.73684	−0.69828	−1.15139
8 08-OEt	−0.64137	−0.72985	−1.01893
9 09-NHPh	−0.79553	−0.24225	−0.48689
10 10-NHCONHPh	−1.28032	−0.84863	−0.05095
11 11-Br	−0.48993	−0.02591	−0.04368
12 12-Cl	−0.59298	−0.08422	−0.17283
13 13-SEt	−0.32207	−0.20528	−0.17811
14 14-CF_3	−0.57511	0.72560	0.69128
Conventional fit			
R^2:	0.26313	0.47493	0.66610
SE of estimate:	0.63513	0.55997	0.46834
	LOGPR_fit (1)	LOGPR fit (2)	LOGPR_fit (3)
1 01-NHEt	−0.78163	−1.12857	−1.60566
2 02-NHC_5H_{11}	−0.11574	−0.57514	−0.81284
3 03-NHCOMe	−0.81554	−0.82214	−0.67399
4 04-H	−0.86029	−0.65989	−0.95885
5 05-Dioxinyl	−1.31851	−1.54317	−1.03248
6 06-CHO	−0.63603	−0.23619	−0.03062
7 07-Me	−0.69562	−0.64025	−1.00587
8 08-OEt	−0.61877	−0.69025	−0.93827
9 09-NHPh	−0.03917	−0.39555	−0.38793
10 10-NHCONHPh	−0.21066	−0.62034	−0.05047
11 11-Br	−0.33420	−0.03800	−0.03391
12 12-Cl	−0.39805	−0.06664	−0.13828
13 13-SEt	−0.27005	−0.23255	−0.11579
14 14-CF_3	−0.11574	0.43867	0.57496

full 'sample space' has only m = 3 dimensions. Note how the unscaled data (Fig. 1) seem to cluster around sample 7. Scaling the properties to unit variance (Fig. 2) spreads out the sample positions.

The full length of each vertical bar, as marked by sample number '1' through '14', indicates the observed response. (Greatest potency, i.e. most negative LOGPR, is at top.) Principal compo-

TABLE 5
ANTI-ALLERGIC COMPOUNDS: SAMPLS ANALYSIS OF SCALED DATA

PLS fit of 14 samples by 3 components...

	y(obs):	Average	Variance	SD
		−0.515	0.505	0.711
Cross-validated fit SAMPLS v0.5				
Scaling option: Scale to var = 1, mean = 0				
R^2 squared:		0.1136	0.4066	0.3087
SE of estimate:		0.6966	0.5953	0.6739

Sample	y(obs)	Prediction or Fit (ih = ..)		
		1	2	3
1 01-NHEt	−2.150	−1.142	−1.180	−1.272
2 02-NHC_5H_{11}	−1.280	−0.594	−0.522	−0.550
3 03-NHCOMe	−1.190	−0.422	−0.605	−0.589
4 04-H	−1.000	−0.384	−0.911	−0.946
5 05-Dioxinyl	−0.750	−0.619	−1.046	−1.658
6 06-CHO	−0.630	0.298	0.130	0.147
7 07-Me	−0.420	−0.763	−1.093	−1.151
8 08-OEt	−0.400	−0.841	−0.997	−1.019
9 09-NHPh	−0.150	−1.046	−0.481	−0.487
10 10-NHCONHPh	−0.050	−0.874	−0.289	−0.051
11 11-Br	0.020	−0.052	−0.038	−0.044
12 12-Cl	0.040	−0.089	−0.163	−0.173
13 13-SEt	0.350	−0.427	−0.178	−0.178
14 14-CF_3	0.400	0.665	0.659	0.691

Conventional fit SAMPLS v0.5				
Scaling option: Scale to var = 1, mean = 0				
R^2:		0.4822	0.6651	0.6661
SE of estimate:		0.5324	0.4472	0.4683

Sample	y(obs)	Prediction or Fit (ih = ..)		
		1	2	3
1 01-NHEt	−2.150	−1.359	−1.589	−1.606
2 02-NHC_5H_{11}	−1.280	−1.165	−0.796	−0.813
3 03-NHCOMe	−1.190	−0.520	−0.691	−0.674
4 04-H	−1.000	−0.533	−0.945	−0.959
5 05-Dioxinyl	−0.750	−0.736	−1.090	−1.032
6 06-CHO	−0.630	0.117	−0.039	−0.031
7 07-Me	−0.420	−0.721	−0.987	−1.006
8 08-OEt	−0.400	−0.797	−0.927	−0.938
9 09-NHPh	−0.150	−0.851	−0.383	−0.388
10 10-NHCONHPh	−0.050	−0.623	−0.086	−0.050
11 11-Br	0.020	−0.060	−0.024	−0.034
12 12-Cl	0.040	−0.078	−0.124	−0.138
13 13-SEt	0.350	−0.320	−0.116	−0.116
14 14-CF_3	0.400	0.435	0.588	0.575

nents regression (PCR) with h = 2 components must choose a best-fit plane through these data. Because PLS is not constrained to use PC1 and PC2 as its first two components, the PLS fit, a hyperplane in (n+1) = 4 dimensions, will not look like a plane in this quasi-3D projection. The successive PLS approximations to the response for h = 1, 2, and 3 are marked on the bars.

TABLE 6
ANTI-ALLERGIC COMPOUNDS: RS/1 PLS ANALYSIS OF SCALED DATA

Cross-validated responses

	Actual LOGPR	Predicted		
		h = 1	h = 2	h = 3
01-NHEt	−2.15	−1.142101	−1.179706	−1.271595
02-NHC_5H_{11}	−1.28	−0.594130	−0.522461	−0.549689
03-NHCOMe	−1.19	−0.421908	−0.605442	−0.589152
04-H	−1.00	−0.384279	−0.910933	−0.945829
05-Dioxinyl	−0.75	−0.618518	−1.045870	−1.657957
06-CHO	−0.63	0.297794	0.130346	0.146862
07-Me	−0.42	−0.762689	−1.093365	−1.151387
08-OEt	−0.40	−0.841272	−0.997069	−1.018927
09-NHPh	−0.15	−1.046017	−0.481088	−0.486887
10-NHCONHPh	−0.05	−0.873701	−0.289432	−0.050947
11-Br	0.02	−0.051991	−0.037760	−0.043676
12-Cl	0.04	−0.088979	−0.163437	−0.172832
13-SEt	0.35	−0.427440	−0.178241	−0.178109
14-CF_3	0.40	0.664575	0.658509	0.691282

Summary statistics

h	X-block residual	Y-block residual	Seq. F	Prob. > *F*	cv SSE LOGPR	cv R^2 LOGPR
0	1.000000	0.505319				
1	0.574174	0.261647	12.106958	0.004071	5.823165	0.114
2	0.136616	0.169243	6.551815	0.025018	3.898088	0.407
3	4.373906e-34	0.168724	0.033816	0.857446	4.541390	0.309

Analysis of Variance
Response LOGPR; 2 components extracted

Source	df	Sum Sq.	Mean Sq.	F-ratio	Signif.
Intercept	1	3.713150	3.713150	18.56	0.0012
Regression	2	4.368995	2.184497	10.92	0.0024
Error	11	2.200155	0.200014		
Total	14	10.282300			

$R^2 = 0.6651$
Adj. $R^2 = 0.6042$ Cross-validated $R^2 = 0.4066$ (14 groups)

RMS error = 0.447229

TABLE 6 (continued)

Conventional regression, h = 2

	Actual LOGPR	Fitted LOGPR	Residual LOGPR
01-NHEt	−2.15	−1.589252	−0.560748
02-NHC_5H_{11}	−1.28	−0.796008	−0.483992
03-NHCOMe	−1.19	−0.691461	−0.498539
04-H	−1.00	−0.945311	−0.054689
05-Dioxinyl	−0.75	−1.090401	0.340401
06-CHO	−0.63	−0.039166	−0.590834
07-Me	−0.42	−0.986520	0.566520
08-OEt	−0.40	−0.926732	0.526732
09-NHPh	−0.15	−0.382583	0.232583
10-NHCONHPh	−0.05	−0.086220	0.036220
11-Br	0.02	−0.023689	0.043689
12-Cl	0.04	−0.124297	0.164297
13-SEt	0.35	−0.115934	0.465934
14-CF_3	0.40	0.587575	−0.187575

These plots make clear why scaling the explanatory properties improves the model of anti-allergic response. When the samples are projected onto a horizontal space according to their *unscaled* explanatory properties (Fig. 1), close-neighboring samples have very different observed responses: for example, (7, 12, 14) or (1, 13). The fitted value for at least one sample in each of these clusters will be a bad compromise. Indeed, PLS produces a dubious fit, as judged by cross-validated R^2. The scatterplot also hints that the poorly fit compounds are by no means 'outliers' in terms of their *explanatory* properties. Omitting them might better be termed 'silencing dissenters' than 'dropping outliers'. When samples are repositioned horizontally according to *scaled* explanatory properties (Fig. 2), there is a clear trend to better activity at the lower left (PC1<0, PC2<0). The PLS fit becomes much better.

The steroid CoMFA responses (CBG binding) are plotted against principal components in Fig. 3. The general trend to better activity at the right corresponds to fairly good PCR or PLS regressions with one or two components. Note how sample 9 and 10 fall well below any plane through the observations (in this projection). However, apparent positions may be misleading, since the scatterplot projects nearly 1000 dimensions into 2, with high distortion of intersample distances. (This distortion can be confirmed by regenerating distances with 2, 3, or even 10 factors.)

PLS 'rotates' the samples in n-space to fit a plane as well as possible, as shown in Fig. 4. This plot distributes the sample points horizontally according to the PLS components rather than the principal components. Samples 9 and 10, for example, move to the less-active region of the map. The predictions of a 2-component PLS model must now appear as a plane. The fact that this rotation has been chosen for best fit to a plane emphasizes the need for validation. This two-component CoMFA model (conventional R^2 of 0.897) fares well in cross-validation (cross-validated R^2 of 0.687). Nonetheless, the model should be tested on compounds that vary along novel

TABLE 7
ANTI-ALLERGIC COMPOUNDS: SYBYL PLS ANALYSIS OF SCALED DATA

Cross-validation			
R^2:	0.09213	0.39327	0.30968
SE of predictions (cross-validated):	0.70498	0.60194	0.67390
	LOGPR cv (1)	LOGPR cv (2)	LOGPR cv (3)
1 01-NHEt	−1.12525	−1.21166	−1.27159
2 02-NHC_5H_{11}	−0.56450	−0.49515	−0.54969
3 03-NHCOMe	−0.42433	−0.61118	−0.58915
4 04-H	−0.37854	−0.91433	−0.94583
5 05-Dioxinyl	−0.69526	−1.21984	−1.65796
6 06-CHO	0.29844	0.13188	0.14686
7 07-Me	−0.76321	−1.09989	−1.15139
8 08-OEt	−0.83983	−0.99812	−1.01893
9 09-NHPh	−1.03940	−0.47706	−0.48689
10 10-NHCONHPh	−0.94690	−0.22913	−0.05095
11 11-Br	−0.05860	−0.04023	−0.04368
12 12-Cl	−0.09297	−0.16862	−0.17283
13 13-SEt	−0.42683	−0.17830	−0.17811
14 14-CF_3	0.57432	0.64564	0.69128
Conventional fit			
R^2:	0.48222	0.66508	0.66610
SE of estimate:	0.53240	0.44723	0.46834
	LOGPR fit (1)	LOGPR fit (2)	LOGPR fit (3)
1 01-NHEt	−1.35868	−1.58925	−1.60566
2 02-NHC_5H_{11}	−1.16515	−0.79601	−0.81284
3 03-NHCOMe	−0.52002	−0.69146	−0.67399
4 04-H	−0.53272	−0.94531	−0.95885
5 05-Dioxinyl	−0.73616	−1.09040	−1.03248
6 06-CHO	0.11669	−0.03917	−0.03062
7 07-Me	−0.72137	−0.98652	−1.00587
8 08-OEt	−0.79652	−0.92673	−0.93827
9 09-NHPh	−0.85078	−0.38258	−0.38793
10 10-NHCONHPh	−0.62271	−0.08622	−0.05047
11 11-Br	−0.05980	−0.02369	−0.03391
12 12-Cl	−0.07832	−0.12430	−0.13828
13 13-SEt	−0.31957	−0.11593	−0.11579
14 14-CF_3	0.43509	0.58757	0.57496

'dimensions' not explored in the training set. This CoMFA/PLS model did in fact yield useful predictions for additional compounds, as discussed in Ref. 1.

These static plots may suggest even more powerful ways to visualize and interact with the data.

TABLE 8
SAMPLS ANALYSIS OF STEROID CoMFA

PLS fit of 21 samples by 4 components....					
Average:	−6.148	0.000	0.000	0.000	0.000
Cross-validated fit SAMPLS v1.0					
Scaling option: None (use distances)					
R^2:	0.000	0.639	0.687	0.615	0.584
SD:	1.173	0.705	0.656	0.728	0.756
SE:	1.173	0.724	0.692	0.790	0.846
Sample	y (obs)	Prediction or Fit (ih = ..)			
		1	2	3	4
1 Aldesterone	−6.279	−7.390	−7.659	−7.758	−7.796
2 Deoxycorticosterone	−7.653	−7.293	−7.456	−7.548	−7.592
3 Deoxycortisol	−7.881	−7.099	−7.086	−7.057	−7.027
4 Dihydrotestosterone	−5.919	−5.960	−6.524	−6.619	−6.661
5 Estradiol	−5.000	−5.053	−4.949	−5.377	−5.073
6 Estriol	−5.000	−5.334	−5.250	−5.687	−5.615
7 Estrone	−5.000	−4.649	−4.589	−4.984	−4.833
8 Etiocholanolone	−5.255	−5.550	−5.739	−5.677	−5.699
9 Pregnenolone	−5.255	−6.415	−5.563	−5.829	−5.851
10 17-OH-Pregnenlone	−5.000	−6.568	−5.642	−5.838	−5.822
11 Progesterone	−7.380	−7.022	−6.982	−6.950	−6.891
12 Androstanediol	−5.000	−5.614	−5.765	−5.641	−5.750
13 Hydroxyprogesterone	−7.740	−6.911	−6.820	−6.815	−6.833
14 Testosterone	−6.724	−5.859	−6.317	−6.391	−6.396
15 Androstenediol	−5.000	−5.293	−4.784	−4.669	−4.668
16 Androstendione	−5.763	−5.912	−6.498	−6.718	−6.901
17 Androsterone	−5.613	−5.110	−4.862	−4.809	−4.803
18 Corticosterone	−7.881	−7.157	−7.350	−7.555	−7.631
19 Cortisol	−7.881	−7.049	−6.955	−6.907	−6.805
20 Cortisone	−6.892	−7.409	−7.569	−7.669	−7.674
21 Dehydroepiandrostrone	−5.000	−5.249	−4.562	−4.273	−4.226

DISCUSSION

Equivalence of sample-based and property-based PLS

The tests confirm that SAMPLS reproduces the results of property-based PLS regression. Agreement is exact for any scaling of the data, as long as that scaling is retained throughout the analysis.

A small discrepancy occurs in the cross-validation of the scaled anti-allergic model, because SYBYL rescales *during* cross-validation. We believe that it is fundamentally sound to retain the original scaling during cross-validation. Consider what happens if a property happens to be nearly (not exactly) constant over all but one of the training samples. During cross-validation this one sample is 'predicted' from the others. If we rescale the property values to unit variance over

TABLE 8
(continued)

PLS fit of 21 samples by 4 components....					
Average:	−6.148	0.000	0.000	0.000	0.000
Conventional fit SAMPLS v1.0					
Scaling option: None (use distances)					
R^2:	0.000	0.748	0.897	0.938	0.964
SD:	1.173	0.589	0.376	0.292	0.221
SE:	1.173	0.604	0.397	0.316	0.248
Sample	y (obs)	Prediction or Fit (ih = ..)			
		1	2	3	4
1 Aldesterone	−6.279	−7.285	−7.012	−6.500	−6.075
2 Deoxycorticosterone	−7.653	−7.392	−7.581	−7.661	−7.692
3 Deoxycortisol	−7.881	−7.338	−7.565	−7.749	−7.884
4 Dihydrotestosterone	−5.919	−5.932	−6.439	−6.231	−6.180
5 Estradiol	−5.000	−4.837	−5.034	−5.275	−4.949
6 Estriol	−5.000	−5.028	−5.056	−5.352	−5.042
7 Estrone	−5.000	−4.574	−4.854	−5.117	−4.947
8 Etiocholanolone	−5.255	−5.207	−5.473	−4.935	−5.167
9 Pregnenolone	−5.255	−6.279	−5.369	−5.512	−5.590
10 17-OH-Pregnenlone	−5.000	−6.342	−5.163	−5.222	−5.236
11 Progesterone	−7.380	−7.092	−7.097	−7.112	−7.101
12 Androstanediol	−5.000	−5.418	−5.411	−5.087	−5.105
13 Hydroxyprogesterone	−7.740	−7.080	−7.174	−7.366	−7.553
14 Testosterone	−6.724	−5.943	−6.546	−6.558	−6.581
15 Androstenediol	−5.000	−5.194	−4.738	−4.747	−4.826
16 Androstendione	−5.763	−5.868	−6.365	−6.262	−6.342
17 Androsterone	−5.613	−5.094	−5.178	−5.209	−5.553
18 Corticosterone	−7.881	−7.338	−7.636	−7.859	−7.926
19 Cortisol	−7.881	−7.334	−7.520	−7.683	−7.718
20 Cortisone	−6.892	−7.415	−7.344	−7.185	−7.009
21 Dehydroepiandrostrone	−5.000	−5.127	−4.562	−4.493	−4.639

the other (n−1) samples, we assign a huge value of that property to the 'predicted' sample. This leads to an infinite (positive or negative) prediction, and thus to infinite cross-validated errors. The same logical inconsistency can arise during ordinary prediction. To avoid infinite predictions, we must either scale according to some external criterion (as for CoMFA fields), or else scale, once and for all, according to *all* values known – prediction samples as well as training samples.

SAMPLS does not calculate any statistical quantities related to the m explanatory properties. The most useful of these would be the m-vector of prediction coefficients. This is simply $(\mathbf{X}^T \mathbf{v})$, where $\mathbf{v}$ is the SAMPLS prediction vector of Eq. 18. Either vector, $\mathbf{v}$ or $(\mathbf{X}^T \mathbf{v})$, completely summarizes a PLS or PCR prediction. The former is more compact when there are more properties m than samples n. For a detailed description of other intermediate quantities related to

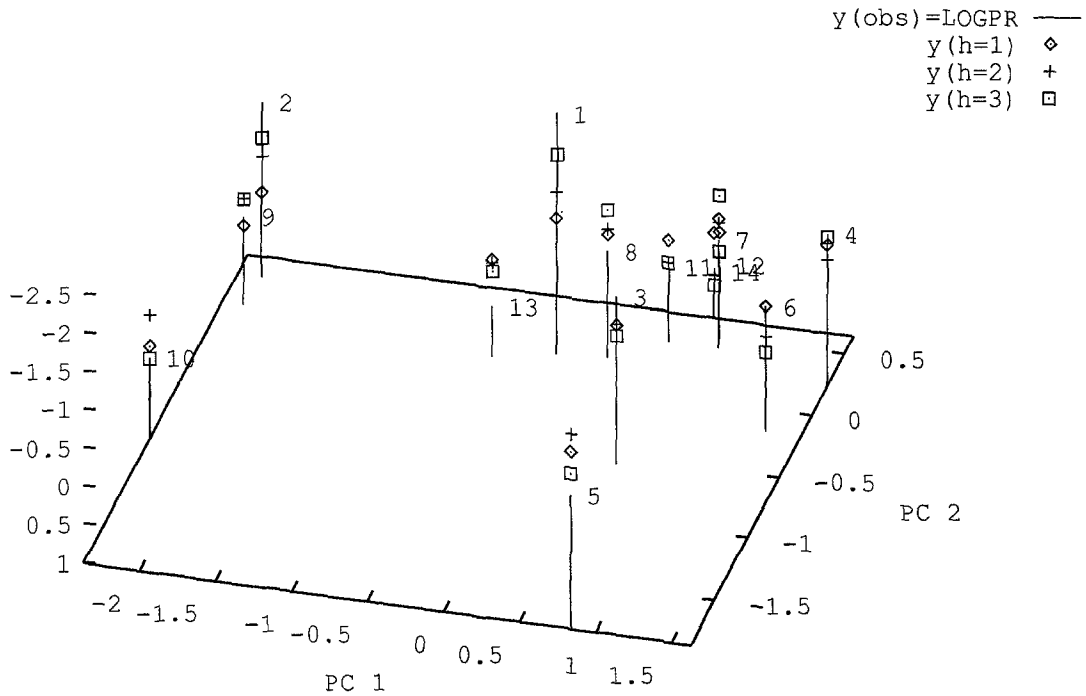

Fig. 1. Anti-allergic compounds: unscaled data and PLS analysis. Horizontal plane: scatterplot of samples, along principal components PC1 and PC2. Vertical displacement: observed bioactivity (LOGPR) and PLS fits to bioactivity.

individual properties, and the dozens of relationships among them, the interested reader may consult Glen et al. [16].

The SAMPLS algorithm presented here is restricted to the case of a single response. Property-based PLS, on the other hand, can generate a linear model for several responses simultaneously. This procedure may be able to increase the reliability of the signal (the regression) when the sources of noise in the various responses are independent. SAMPLS cannot perform such a simultaneous PLS analysis; it can only detect patterns strong enough to emerge in a single response. This restriction may not be a great hindrance for biochemical and biological applications, however.

Understanding PLS in terms of sample covariance **C**

The SAMPLS formulation leads to several observations which may clarify PLS. They can only be sketched here, but many of the comments will suggest generalizations of PLS or guidelines for using it safely.

(1) PLS depends on explanatory data only through covariance **C** *or intersample distances* **D**

The SAMPLS algorithm makes this explicit, but the statement holds for any implementation of PLS, no matter how complex it appears. Methods which adjust the explanatory factors, on the other hand, lie outside the family of sample-distance techniques.

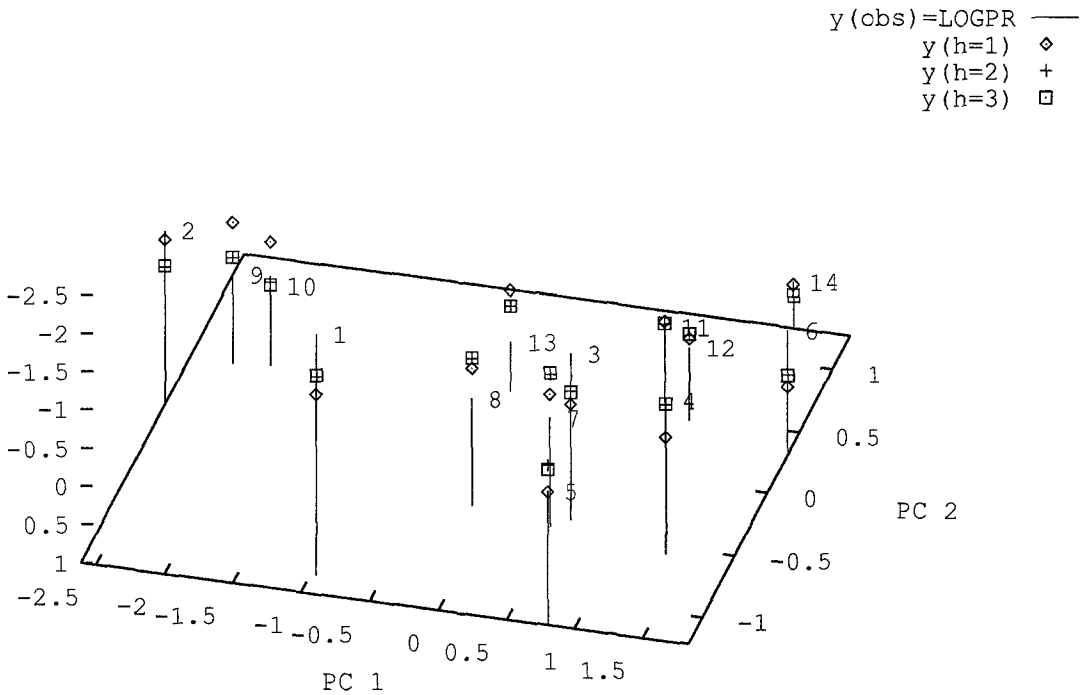

Fig. 2. Anti-allergic compounds: scaled data and PLS analysis. Same axes as Fig. 1. Scaling the data changes the intersample distances and the positions in the horizontal plane.

(2) PLS performs a partial inversion of **C**

A complete least-squares fit generates the prediction vector

$$\mathbf{v} = \mathbf{C}^{-1}\,\mathbf{y} \tag{21}$$

(If m<n, $\mathbf{C}^{-1}$ may be defined as the generalized inverse.) *Partial* least squares (PLS) approximates **v** without actually inverting matrix **C**. The first approximation is a scalar multiple of vector $(\mathbf{C}\,\mathbf{y}_0)$, namely:

$$\mathbf{v}_1 = [(\mathbf{y}_0^{\mathrm{T}}\,\mathbf{C}\,\mathbf{y}_0) / (\mathbf{y}_0^{\mathrm{T}}\,\mathbf{C}\,\mathbf{C}\,\mathbf{y}_0)]\,\mathbf{C}\,\mathbf{y}_0$$

The second approximation adds a scalar multiple of $(\mathbf{C}\,\mathbf{C}\,\mathbf{y}_0)$, and so forth. The direct powers $\mathbf{C}^n$ are not actually computed.

(3) PLS is equivalent to conjugate directions

Calculating $(\mathbf{C}^{-1}\,\mathbf{y})$ in this way is the same as determining **v** by conjugate directions minimization of $(\mathbf{C}\,\mathbf{y}-\mathbf{v})^2$, starting at $\mathbf{v} = 0$. Directions of residuals $\mathbf{y}_h$ are 'conjugate': each $\mathbf{y}_h$ is orthogonal to all previous $\mathbf{t}_g = \mathbf{C}\,\mathbf{y}_g$, so $(\mathbf{y}_h^{\mathrm{T}}\,\mathbf{C}\,\mathbf{y}_g) = 0$.

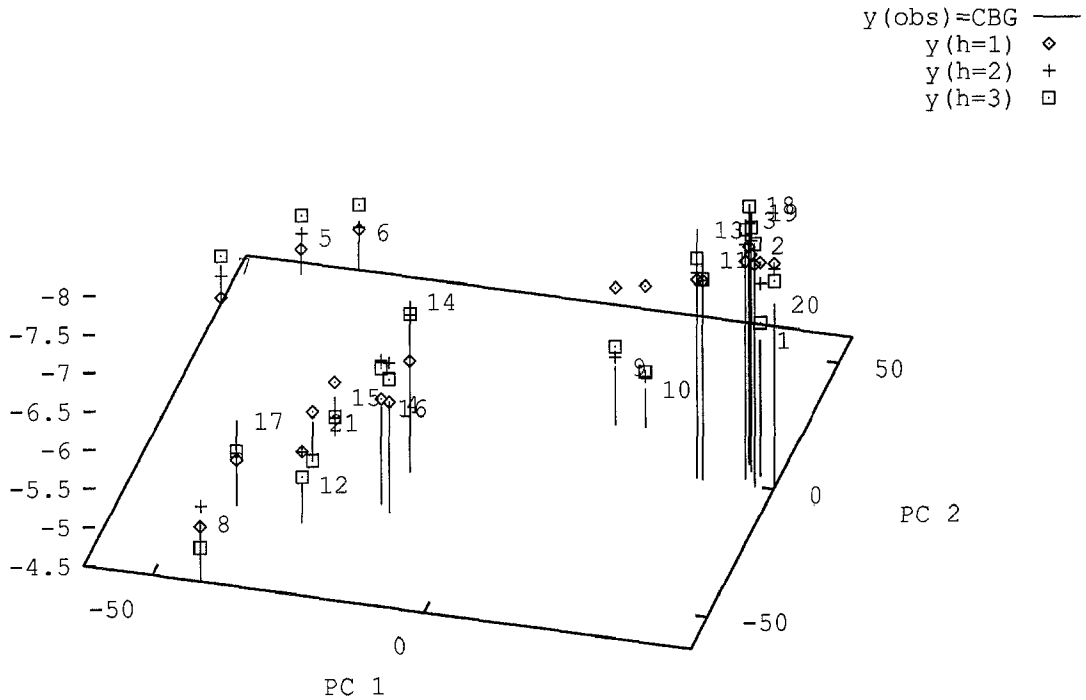

Fig. 3. Steroid CoMFA: unscaled data and PLS fit. Horizontal plane: scatterplot of intersample distances, derived from CoMFA field values, along principal components PC1 and PC2. Vertical displacement: observed bioactivity (CBG binding) and PLS fits to bioactivity.

(4) The first components extracted by PLS or PCR are more stable than full least squares

The uncertainty in a least-squares prediction vector $\mathbf{v} = \mathbf{C}^{-1}\mathbf{y}$ is dominated by the *smallest* eigenvalues of **C**. If the eigenvalues span a large range, the problem is 'ill-conditioned', because the small eigenvalues are sensitive to noise in either the response **y** or the explanatory properties **C**. The first few components determined by either PLS or PCR, on the other hand, are dominated by the most stable principal components of **C** (those having *large* eigenvalues λ_1, λ_2, ...). Indeed, PCR begins by projecting **y** along the first principal component of **C**, and multiplying the result by $(1/\lambda_1)$. PLS uses all of **y**, multiplying it by factor β_1 which is in general larger than $(1/\lambda_1)$. As a result, PCR is even more stable than PLS, but in general PLS yields a greater 'signal' as measured by the magnitude of prediction vector **v**.

(5) Predictions of full LS or PCR are additive in the training observations, but PLS predictions are not

For full LS or PCR, the prediction vector **v** depends additively on the observations y_i, that is, linearly on vector **y** (Eq. 21). But the PLS prediction vector contains **y** in various combinations ($\mathbf{y}^T\,\mathbf{C}^n\,\mathbf{y}$) in both numerator and denominator. (These nonlinear combinations 'force' the PLS prediction toward the observations.) In any of these methods, the prediction is linear in the properties $\mathbf{c}^*$ of the *prediction* sample, and the model is summarized by **v** (Eq. 18).

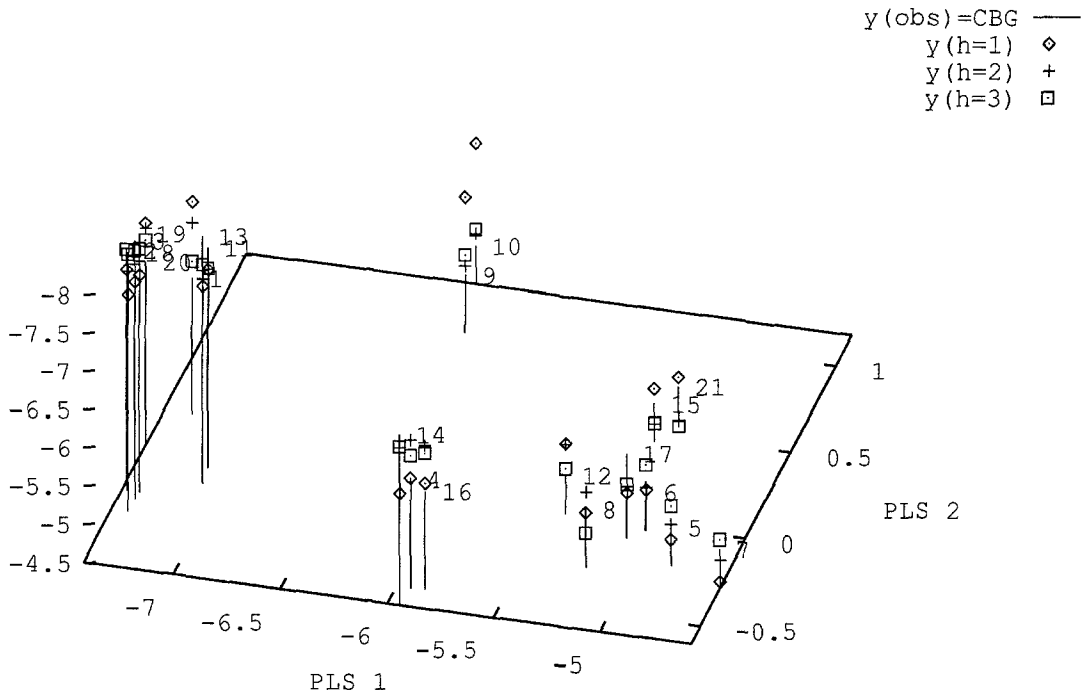

Fig. 4. Steroid CoMFA: PLS components and PLS fit. Horizontal plane: scatterplot of samples along the directions of the first two PLS components. Vertical displacement: observed bioactivity and PLS fits, as in Fig. 3.

(6) Linear models are not *local interpolations*

A smooth local interpolation at point y_p might take the following form, weighting the y-value of each nearby training sample i by a function of distance d(i,p):

$$y_p^* = \Sigma_i [y_i \cdot f(d(i,p))]/(\text{norm}) \tag{22}$$

where

$$\text{norm} = \Sigma_i[f(d(i,p))]$$

Choosing f to be a compact, peaked function like a Gaussian does indeed lead to local smoothing or interpolation, generating a curved (nonlinear) prediction surface which passes near the observed points. But if linear full LS or PCR is expressed in Eq. 22, the 'interpolating' function is $f(d) = d^2$. Influence increases with distance; thus linear models are extremely sensitive to outliers, and the prediction goes to infinity (plus or minus) at large distances. As discussed above, a PLS prediction can also be expressed in terms of the distances to all training samples, but PLS cannot even be written additively in the form of Eq. 22.

(7) Sample-based PLS can assess the reliability of individual *PLS predictions by varying response data*

This is the approach espoused by Efron et al. [14a,b] whenever the strict requirements of

TABLE 9
SYBYL PLS ANALYSIS OF STEROID CoMFA

Cross-validation

	LOGPR.cv (1)	LOGPR.cv (2)	LOGPR.cv (3)	LOGPR.cv (4)
R^2	0.63855	0.68714	0.61471	0.58438
SE of predictions	0.72377	0.69183	0.79000	0.84575

Conventional regressions

	LOGPR.fit (1)	LOGPR.fit (2)	LOGPR.fit (3)	LOGPR.fit (4)
R^2	0.74794	0.89710	0.93827	0.96439
SE of estimate	0.60441	0.39675	0.31622	0.24756

Cross-validated predictions (h = 4) *and regression* (h = 2)

	Cross-validated prediction (h = 4)	Conventional regression (h = 2)
1 Aldesterone	−7.79560	−7.01192
2 Deoxycorticosterone	−7.59182	−7.58064
3 Deoxycortisol	−7.02732	−7.56454
4 Dihydrotestosterone	−6.66056	−6.43948
5 Estradiol	−5.07336	−5.03369
6 Estriol	−5.61521	−5.05647
7 Estrone	−4.83280	−4.85395
8 Etiocholanolone	−5.69852	−5.47254
9 Pregnenolone	−5.85126	−5.36893
10 17-OH-Pregnenlone	−5.82184	−5.16308
11 Progesterone	−6.89122	−7.09713
12 Androstanediol	−5.74987	−5.41102
13 Hydroxyprogesterone	−6.83337	−7.17354
14 Testosterone	−6.39641	−6.54601
15 Androstenediol	−4.66786	−4.73752
16 Androstendione	−6.90056	−6.36492
17 Androsterone	−4.80254	−5.17826
18 Corticosterone	−7.63150	−7.63595
19 Cortisol	−6.80527	−7.52026
20 Cortisone	−7.67433	−7.34373
21 Dehydroepiandrostrone	−4.22563	−4.56241

classical statistics (independent, normally-distributed random 'noise') cannot be assured. Using SAMPLS, one may add random noise to the response vector **y**, or reweight samples (bootstrap) and accumulate an ensemble of prediction vectors (**v** or $\mathbf{w} = \mathbf{X}^T \mathbf{v}$) or predictions y*. This procedure assumes only that measurement variation is uncorrelated across *samples*. Note that we cannot simply estimate the scatter in a prediction from the variability of individual prediction *coefficients* (elements of **v**), since the coefficients are jointly derived from the same observations,

X and **y**. (In the property-based approach, for the same reason, scatter in individual elements of **w** cannot be combined to estimate reliability of predictions.)

Linear models of bioactivity versus similarity analysis

Linear regression and similarity analysis have different strengths and weaknesses for modeling bioactivity. Similarity analysis is very conservative. It bestows favorable predictions only upon compounds which closely resemble known active compounds. This is a form of local nonlinear interpolation, fitting a smooth curved prediction surface through the known bioactivities. The ultimate of such local analysis is to 'cluster', that is, to predict response on the basis of the nearest training samples, irrespective of trends.

Linear models (such as PLS) are far from conservative. They project *trends* far from the training set, and employ *group additivity* to combine the effects of several modifications. These are sensible extrapolations, especially when analyzing spectra or other physical measurements. In modeling bioactivity, however, it is well to keep in mind that binding affinity almost always 'saturates' or approaches a limiting value for large modifications of the compound. Furthermore, linear models are very sensitive to the choice of compounds, especially to *multiplicity*. For example, three benzodiazepine compounds with moderately good bioactivity (say, 1 log-unit better than average, or a factor of 10 in affinity) will influence a PLS model as strongly as a single compound with very high activity (three log-units, or a factor 10^3 in affinity). Similarity analysis is usually much less sensitive to the choice of compounds.

In practice, therefore, linear regression is usually applied in a somewhat localized way, by informally clustering beforehand or by excluding outliers. A more formal approach, combining advantages of similarity analysis with those of linear models, would be to reweight compounds according to local density in sample-space, then employ a smoothing function f(d) having a fairly long 'tail'. The regression would display nearly linear behavior within a cluster, while approaching a constant value (representing 'no information') far from the training samples. The iterative framework of PLS might provide a useful starting point for such an approach.

CoMFA fields and PLS analysis

The points above apply to PLS analysis of any data. The special behavior of PLS with CoMFA fields deserves brief discussion.

(1) CoMFA fields are suitable data for linear regression to bioactivity, if they include the important physical interactions

Each CoMFA field value is (nominally) a local interaction energy of the molecule with a possible receptor fragment. Adding together these energies is a reasonable way to represent binding free energy (or logarithm of potency), as long as the CoMFA fields include all important contributions. However, binding within an aqueous medium requires energy to desolvate the ligand and receptor. Even the simplest models of solvation [31,32] involve large energies which are *quadratic* (second power) in atomic charges. (Charges of either sign are 'hydrophilic', whereas uncharged entities are 'hydrophobic'.) The standard CoMFA fields, steric and electrostatic, are respectively independent of charge and linear (first power) in the charge. Trying to fit solvation

phenomena with these two CoMFA fields can only amplify noise in the data. A promising method for incorporating a 'hydrophobic' field into CoMFA has been described by Kellogg et al. [33]. The CoMFA studies of Klebe and Abraham [2], utilizing binding geometries modeled upon crystal structures of inhibitor complexes, also indicate that whereas 'the steric and electrostatic field contributions ... [predict] the enthalpic contribution ...', they do not suffice for predictions of free enthalpy (free energy) of binding.

(2) PLS reduces CoMFA fields to a series of overlap 'volumes'

PLS ignores all details of shape except as they affect overlap volume in the general sense. Each element of correlation matrix **C** is a sum over space of the products of two various CoMFA fields, which is effectively a weighted volume of overlap. For short-range steric fields having a constant value, for example, the integral is just the steric overlap of the molecules.

(3) Distance cut-offs and other details of implementation, rather than details of molecular structure, can dominate CoMFA/PLS results

The short-range repulsive peak of the steric field would dominate the overlap integrals, and thus the PLS results. In practice, the field value is truncated to a maximum value; the choice of this limiting value may well dominate the PLS model. Electrostatic fields are the opposite case: the long-range overlap of monopole fields diverges to infinity, and even for dipole fields the long-range overlap is not absolutely convergent. The choice of grid dimensions, distance cut-off, or 'distance-dependent dielectric' may well influence the result more than do the atomic charges.

(4) CoMFA modeling and 'field fitting' could be performed without a grid, in terms of an interatomic 'energy'

CoMFA fields are usually generated by summing spherically symmetric *atomic* fields. The overlap integral of two such fields is a function only of interatomic distance. Depending on the mathematical form of the atomic fields, the overlap is often a universal function which need only be multiplied by the 'weights' or 'charges' of the two atoms. Therefore, *except for effects of the cut-offs* mentioned above, PLS reduces the CoMFA fields to a similarity measure or 'energy' between molecules, which is an *additive* function of interatomic 'potentials'. (CoMFA may well require a nonadditive 'energy' function to give sensible results: for example, steric overlap should not depend on the exact positions of interior atoms. Only cut-offs can provide such nonadditivity.)

Re-implementing CoMFA without the need for a grid might allow rapid 'field fitting': that is, adjustment of molecular conformations in such a way as to improve the predictions. Even if it is technically feasible, however, 'field fitting' risks overinterpreting the data.

(5) CoMFA and PLS treat favorable and unfavorable energies on the same footing; thermodynamics does not

Thermodynamically, a state of highly unfavorable (positive) energy contributes very *little* to free energy, because it is exponentially unlikely to be observed ($E \exp(-E) \rightarrow 0$ for $E >> 1$). For this reason, CoMFA usually invokes special rules for highly unfavorable energy contributions; for example, by truncating only positive values of steric energy, and by ignoring electrostatic energy

when steric energy is highly positive. Here again, physically reasonable results depend on details of implementation.

Computational speed

For analyses involving small m, both conventional PLS and SAMPLS are so fast that speed is no consideration. The standard property-based algorithm needs roughly (4·m·n·h) multiplications to generate an h-component PLS regression. SAMPLS needs (n·n·h) multiplications for the regression, after a preliminary calculation of covariance **C** which takes (n·n·m) multiplications. (SAMPLS also requires (m · n) multiplications to convert the regression coefficients back into property space, if this is desired for CoMFA mapping or other purposes). The first component produced by SAMPLS is therefore slower than in the property-based approach (speed factor 4/n), but when m is large, subsequent components are faster by a factor of (4 m/n).

The real difference in speed and compactness comes when validating a model. The property-based approach uses the whole n-by-m table, whereas SAMPLS avoids all m-vector manipulations. Thus SAMPLS has an intrinsic advantage of (4 m/n) in speed for each additional validation group. For CoMFA fields, this could be (4000/20) or 200-fold.

In practice, we experienced a far larger speed-up in the steroid CoMFA test. A fully cross-validated 4-component run of SAMPLS (m = 992) took 0.19 CPU s, compared to 2800 CPU s (approximately 15 000 times as long) using SYBYL PLS. Indeed, SYBYL users often reduce the number of explanatory variables by setting a 'minimum sigma' threshold of 2 kcal/mol. Unlike the use of SAMPLS, which involves no approximations, this is a drastic step (effectively ignoring many polar interactions), and it does indeed reduce time drastically. When this threshold is imposed on the steroid CoMFA, only m = 105 field properties enter the PLS analysis, which then takes only 37 CPU s. Even so, the statistics change very little. This underscores how *in*sensitive PLS can be to the details of CoMFA fields.

Beginning with **D** or **C**, SAMPLS performs analysis and validation trials very quickly, in a time which is independent of the number of explanatory properties. Memory requirements are also reduced. The advantage applies as well to topological 'trend vector analysis' and other approaches which can involve thousands of possible descriptors. Although SAMPLS is a linear procedure, it might also serve as the inspiration for an efficient nonlinear structure-activity method.

The SAMPLS program and the RS/1 implementation of PLS will be submitted to the Quantum Chemistry Program Exchange, Chemistry Department, Indiana University, Bloomington, IN 47405, U.S.A.

CONCLUSIONS

A sample-based formulation of partial least squares, implemented as program SAMPLS, is shown to yield the same results as property-based PLS in both fitting and cross-validation. Computational speed is independent of the number of properties, making SAMPLS ideal for CoMFA studies or topological analysis.

PLS reduces the information in CoMFA fields to a matrix of molecular similarities, and then fits the reduced data with a linear model. This drastic reduction achieves stable results, but may eliminate

important molecular details, or allow local peaks of field 'energy' to overshadow more subtle features. Ad hoc rules designed to control such effects may themselves dominate the PLS model.

SAMPLS appears to be a substantial technical advance for rapid exploratory modeling of bioactivity. Thermodynamic laws and physicochemical considerations indicate, however, that progress in molecular design may also require new definitions of molecular similarity, and a nonlinear analysis of molecular fields.

Note added in proof

Lindgren, Geladi and Wold have recently published [34] a 'Kernel Algorithm for PLS' in terms of square matrices indexed by *property* (m-by-m matrices, in our notation). Their approach complements the *sample-distance* algorithm presented here: it is completely general, but requires updating the square matrices component-by-component and during validation for each resampling. Good, So and Richards [35] compare PLS/CoMFA analysis of the steroid data in Ref. 1 to PLS and neural network analyses of an *independently derived* similarity matrix. All methods give comparably good regressions, as proposed above in the discussion of 'CoMFA fields and PLS analysis'.

ACKNOWLEDGEMENTS

We thank Dr. David Patterson, Tripos Associates, Inc., for providing a test version of the SYBYL program which generates a file of intersample distances computed from the tabulated properties. It is a pleasure to acknowledge stimulating conversations with Dr. D. Patterson, and with Dr. J. Andrew Dearing (Shell International, The Hague, Netherlands) and Dr. Philip Jonathan (Shell Development, Sittingbourne Research Centre, U.K.). We thank the reviewers for careful reading and useful comments.

REFERENCES

1 Cramer, III, R.D., Patterson, D.E. and Bunce, J.E., J. Am. Chem. Soc., 110 (1988) 5959.
2 Klebe, G. and Abraham, U., J. Med. Chem., 36 (1993) 37.
3 a. Wold, S., Ruhe, A., Wold, H., Dunn, III, W.J., SIAM J. Sci. Stat. Comput., 5 (1984) 735.
3 b. Wold, S., Albano, C., Dunn, III, W.J. and Edlund, U., In Kowalski, B.R. (Ed.) Chemometrics – Mathematics and Statistics in Chemistry (NATO ASI Ser. C 138), Reidel, Dordrecht, 1984, pp. 17–95.
4 Geladi, P. and Kowalski, B.R., Analyt. Chim. Acta, 185 (1986) 1.
5 Höskuldsson, A., J. Chemometr., 2 (1988) 221.
6 Stahle, L. and Wold, S., J. Pharmacol. Methods, 16 (1986) 91.
7 Hellberg, S., Sjöström, M., Skagerberg, B. and Wold, S., J. Med. Chem., 30 (1987) 1126.
8 Johnsson, J., Eriksson, L., Hellberg, S., Sjöström, M. and Wold, S., Acta Chem. Scand., 43 (1989) 286.
9 Hartauer, K.J. and Guillory, J.K., Pharm. Res., 6 (1989) 608.
10 Dousseau, F. and Pezolet, M., Biochemistry, 29 (1990) 8771.
11 DeMeo, G., Pedini, M., Ricci, A., Bastianini, L., Jacquignon, P.C., Bonelli, D., Clementi, S. and Cruciani, G., Farmaco, 45 (1990) 313.
12 SYBYL: Tripos Associates, Inc., 1699 South Hanley Road, Suite 303, St. Louis, MO 63144, U.S.A.
13 Carhart, R.E., Smith, D.H. and Venkataraghavan, R., J. Chem. Inform. Comput. Sci., 25 (1985) 64.
14 a. Efron, B. and Tibshirani, R., Science, 253 (1991) 390.
14 b. Efron, B. and Gong, G., Am. Statist., 37 (1983) 36.
15 Stone, A. and Jonathan, P., J. Chemometr., in press.

16 Glen, W.G., Dunn, III, W.J. and Scott, D.R., Tetrahedron Comput. Methodol., 2 (1989) 349.
17 Glen, W.G., Sarker, M., Dunn, III, W.J. and Scott, D.R., Tetrahedron Comput. Methodol., 2 (1989) 377.
18 Katsumi, H., Yoshida, M., Kikuzono, Y., Takayama, C. and Marsili, M., Analyt. Sci., 7 (Suppl.) (1991) 719.
19 Lewi, P.J. and Moereels, H., Trends Analyt. Chem., 10 (1991) 283.
20 van Heel, M.J., J. Mol. Biol., 220 (1991) 887.
21 Everitt, B.S., Cluster Analysis, Halsted, New York, 1980.
22 Kowalski, B.R. and Bender, C.F., J. Am. Chem. Soc., 94 (1972) 5632.
23 Hudson, B., Livingstone, D.J. and Rahr, E., J. Comput.-Aided Mol. Design, 3 (1989) 55.
24 Friedman, J.H. and Stuetzle, W., J. Am. Stat. Assoc., 76 (1981) 817.
25 a. Norinder, U., J. Comput.-Aided Mol. Design, 5 (1991) 419.
25 b. Kim, K.H. and Martin, Y.C., J. Med. Chem., 34 (1991) 2056.
26 RS/1: BBN Software Products Corp., Cambridge, MA.
27 Gould, K.J., Manners, C.N., Payling, D.W., Suschitzky, J.L. and Wells, E., J. Med. Chem., 31 (1988) 1445.
28 Dunn, J.F., Nisula, B.C. and Rodbard, D., J. Clin. Endocrinol. Metab., 1981 (1981) 63.
29 Mahalanobis, P.C., Proc. Natl. Inst. Sci. (India), 122 (1936) 122.
30 Crippen, G.M., Distance Geometry and Conformational Calculations (Chemometrics Research Studies Series, No. 1), Research Studies Press, New York, 1981.
31 Gilson, M. and Honig, B., Proteins, 4 (1988) 7.
32 Harvey, S.C., Proteins, 5 (1989) 78.
33 Kellogg, G.E., Semus, S.F. and Abraham, D.J., J. Comput.-Aided Mol. Design, 5 (1991) 545.
34 Lindgren, F., Geladi, P. and Wold, S., J. Chemometr., 7 (1993) 45.
35 Good, A.C., So, S-S. and Richards, W.G., J. Med. Chem., 36 (1993) 433.

APPENDIX

Proof of equivalence of SAMPLS with property-based PLS

To prove that property-based PLS and sample-based PLS produce equivalent regressions, it suffices to show that they yield the same $\mathbf{t}_h$, within a scalar factor, for all h. The update of residual $\mathbf{y}$ (and, in the property-based algorithm, the update of $\mathbf{X}$) depend only on the direction of $\mathbf{t}$, not its magnitude.

Consider the operator which projects into the space orthogonal to vector $\mathbf{t}_h$:

$$\mathbf{Q}_h = \mathbf{1} - (1/\mathbf{t}_h^T\mathbf{t}_h)\mathbf{t}_h\mathbf{t}_h^T \tag{A.1}$$

For any vector $\mathbf{v}$, the product $\mathbf{Q}_h\mathbf{v}$ is orthogonal to $\mathbf{t}_h$ (that is, $\mathbf{t}_h^T\mathbf{Q}_h\,\mathbf{v} = 0$). Like any projection matrix, $\mathbf{Q}$ is symmetric ($\mathbf{Q} = \mathbf{Q}^T$) and idempotent ($\mathbf{Q} = \mathbf{QQ} = \mathbf{QQQ}$ and so forth to any power). Furthermore, *if* $\mathbf{t}_h$ and $\mathbf{t}_g$ are orthogonal *then* their projection matrices $\mathbf{Q}_h$ and $\mathbf{Q}_g$ commute (that is, $\mathbf{Q}_h\mathbf{Q}_g = \mathbf{Q}_g\mathbf{Q}_h$). These are general properties of projections, quite apart from the choice of PLS vectors. With this notation, the update steps in the property-based algorithm may be written (steps 7', 8')

$$\mathbf{X}_{h+1} = \mathbf{Q}_h\mathbf{X}_h \tag{A.2}$$

$$\mathbf{y}_{h+1} = \mathbf{Q}_h\mathbf{y}_h \tag{A.3}$$

Combining cycles 1 through (h-1), the expression for the updated matrix $\mathbf{X}_h$ becomes

$$\mathbf{X}_h = \mathbf{Q}_{h-1}\mathbf{Q}_{h-2}\ldots\mathbf{Q}_2\mathbf{Q}_1\mathbf{X}_1 \tag{A.4}$$

and similarly for the residual $\mathbf{y}_h$

$$\mathbf{y}_h = \mathbf{Q}_{h-1}\mathbf{Q}_{h-2}\ldots\mathbf{Q}_2\mathbf{Q}_1\mathbf{y}_1 \tag{A.5}$$

Substituting (A.4) and (A.5) into the calculation of $\mathbf{t}_h$ in the property-based algorithm, namely

$$\mathbf{t}_h = \text{scalar}\cdot\mathbf{X}_h\mathbf{X}_h^T\mathbf{y}_h \tag{A.6}$$

gives $\mathbf{t}_h$ in terms of the original $\mathbf{X}$ and $\mathbf{y}$ as:

$$\mathbf{t}_h = \text{scalar}\cdot\mathbf{Q}_{h-1}\mathbf{Q}_{h-2}\ldots\mathbf{Q}_2\mathbf{Q}_1\mathbf{X}_1(\mathbf{Q}_{h-1}\mathbf{Q}_{h-2}\ldots\mathbf{Q}_2\mathbf{Q}_1\mathbf{X}_1)^T\mathbf{Q}_{h-1}\mathbf{Q}_{h-2}\ldots\mathbf{Q}_2\mathbf{Q}_1\mathbf{y}_1 \tag{A.7}$$

This can be simplified further. Because $\mathbf{Q}^T = \mathbf{Q}$, and using the fact that $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$ for any two matrices $\mathbf{A}$, $\mathbf{B}$, we have

$$\mathbf{t}_h = \text{scalar}\cdot\mathbf{Q}_{h-1}\mathbf{Q}_{h-2}\ldots\mathbf{Q}_2\mathbf{Q}_1\mathbf{X}_1(\mathbf{X}_1^T\mathbf{Q}_1\mathbf{Q}_2\ldots\mathbf{Q}_{h-2}\mathbf{Q}_{h-1})\mathbf{Q}_{h-1}\mathbf{Q}_{h-2}\ldots\mathbf{Q}_2\mathbf{Q}_1\mathbf{y}_1 \tag{A.8}$$

Up to this point we have not assumed the orthogonality of the $\mathbf{t}_h$. Equation A.8 shows explicitly that for any h, $\mathbf{t}_h$ is orthogonal to all preceding $\mathbf{t}_g$. By the same argument, any pair of vectors $\mathbf{t}_g$ is orthogonal for g = 1,2,...,(h-1), and therefore all projections $\mathbf{Q}_g$ commute. We may thus permute and combine the projections which are applied to $\mathbf{y}_1$ in Eq. A.8. The result is

Property-based method:

$$\mathbf{t}_h = \text{scalar}\cdot\mathbf{Q}_{h-1}\mathbf{Q}_{h-2}\ldots\mathbf{Q}_2\mathbf{Q}_1\ (\mathbf{X}_1\mathbf{X}_1^T)\ \mathbf{Q}_{h-1}\mathbf{Q}_{h-2}\ldots\mathbf{Q}_2\mathbf{Q}_1\mathbf{y}_1 \tag{A.9}$$

We have reparenthesized for clarity, using the associative property of matrix multiplication.

The simplification of Eq. A.8 into Eq. A.9 is the crux of the sample-based method. Equation A.9 expresses $\mathbf{t}_h$ in terms of the updated vector $\mathbf{y}$ and the *original* matrix $\mathbf{C}_1 = (\mathbf{X}_1\mathbf{X}_1^T)$. The sample-based method uses the identical expression. To see this, write Eq. 15 in terms of projection operators as

$$\mathbf{y}_{g+1} = \mathbf{Q}_g\mathbf{y}_g \tag{A.10}$$

for any g. Iterating gives $\mathbf{y}_h$ in terms of the original response vector $\mathbf{y}_1$,

$$\mathbf{y}_h = \mathbf{Q}_{h-1}\mathbf{Q}_{h-2}\ldots\mathbf{Q}_2\mathbf{Q}_1\mathbf{y}_1 \tag{A.11}$$

The definition of $\mathbf{t}_h$ in the sample-based method (Eqs. 12 and 13) is also an iterated projection:

$$\mathbf{t}_h = \mathbf{Q}_{h-1}\mathbf{Q}_{h-2}\ldots\mathbf{Q}_2\mathbf{Q}_1\ \mathbf{C}_1\mathbf{y}_h \tag{A.12}$$

or, substituting from Eq. A.11

Sample-based method:

$$\mathbf{t}_h = \mathbf{Q}_{h-1}\mathbf{Q}_{h-2}\cdots\mathbf{Q}_2\mathbf{Q}_1\,\mathbf{C}_1\,\mathbf{Q}_{h-1}\mathbf{Q}_{h-2}\cdots\mathbf{Q}_2\mathbf{Q}_1\mathbf{y}_1 \tag{A.13}$$

This is *formally* identical to Eq. A.9. It remains only to prove that the projection operators in Eqs. A.9 and A.13 are the same.

This proof proceeds by induction. First, verify that both methods give the same $\mathbf{t}_1$. The property-based method (Eqs. 1 and 3) gives

$$\mathbf{t}_1 = \mathbf{X}_1\mathbf{w} = \text{scalar}\cdot\mathbf{X}_1(\mathbf{X}_1^T\mathbf{y}_1) \tag{A.14}$$

The sample-based method (Eqs. 11 and 12) gives the same, within a scalar factor:

$$\mathbf{t}_1 = \mathbf{C}_1\mathbf{y}_1 = (\mathbf{X}_1\mathbf{X}_1^T)\mathbf{y}_1 = \mathbf{X}_1(\mathbf{X}_1^T\mathbf{y}_1) \tag{A.15}$$

Now assume that the two methods produce identical $\mathbf{t}_g$ within a scalar factor for all g = 1,2,...,(h−1). The projection operators $\mathbf{Q}_1$, $\mathbf{Q}_2$, ..., $\mathbf{Q}_{h-1}$ defined from these vectors are therefore identical. The two methods then define $\mathbf{t}_h$ through identical expressions (Eqs. A.9 and A.13). Each method then uses Eq. A.3 to define $\mathbf{y}_{h+1}$. By induction, both methods produce the same **t** and **y** for all components h.