BMC Bioinformatics

Methodology article

The C^1C^2: A framework for simultaneous model selection and assessment

Martin Eklund*, Ola Spjuth and Jarl ES Wikberg

Address: Department of Pharmaceutical Pharmacology, Uppsala University, Box 591, BMC, SE-751 24 Uppsala, Sweden

Email: Martin Eklund* - martin.eklund@farmbio.uu.se; Ola Spjuth - ola.spjuth@farmbio.uu.se; Jarl ES Wikberg - jarl.wikberg@farmbio.uu.se

* Corresponding author

Published: 2 September 2008

BMC Bioinformatics 2008, **9**:360 doi:10.1186/1471-2105-9-360

Received: 7 April 2008

Accepted: 2 September 2008

This article is available from: http://www.biomedcentral.com/1471-2105/9/360

Abstract

Background: There has been recent concern regarding the inability of predictive modeling approaches to generalize to new data. Some of the problems can be attributed to improper methods for model selection and assessment. Here, we have addressed this issue by introducing a novel and general framework, the C^1C^2, for simultaneous model selection and assessment. The framework relies on a partitioning of the data in order to separate model choice from model assessment in terms of used data. Since the number of conceivable models in general is vast, it was also of interest to investigate the employment of two automatic search methods, a genetic algorithm and a brute-force method, for model choice. As a demonstration, the C^1C^2 was applied to simulated and real-world datasets. A penalized linear model was assumed to reasonably approximate the true relation between the dependent and independent variables, thus reducing the model choice problem to a matter of variable selection and choice of penalizing parameter. We also studied the impact of assuming prior knowledge about the number of relevant variables on model choice and generalization error estimates. The results obtained with the C^1C^2 were compared to those obtained by employing repeated *K*-fold cross-validation for choosing and assessing a model.

Results: The C^1C^2 framework performed well at finding the true model in terms of choosing the correct variable subset and producing reasonable choices for the penalizing parameter, even in situations when the independent variables were highly correlated and when the number of observations was less than the number of variables. The C^1C^2 framework was also found to give accurate estimates of the generalization error. Prior information about the number of important independent variables improved the variable subset choice but reduced the accuracy of generalization error estimates. Using the genetic algorithm worsened the model choice but not the generalization error estimates, compared to using the brute-force method. The results obtained with repeated *K*-fold cross-validation were similar to those produced by the C^1C^2 in terms of model choice, however a lower accuracy of the generalization error estimates was observed.

Conclusion: The C^1C^2 framework was demonstrated to work well for finding the true model within a penalized linear model class and accurately assess its generalization error, even for datasets with many highly correlated independent variables, a low observation-to-variable ratio, and model assumption deviations. A complete separation of the model choice and the model assessment in terms of data used for each task improves the estimates of the generalization error.

Background

A common task in computational biology/bioinformatics and computational chemistry/chemometrics (hereafter abbreviated BBCC) is to model a dependent variable from a set of independent variables; this gives insight into the workings of the process being modeled and enables prediction of future observations. Typical examples include analyzing potential drug activity through proteochemometrics and quantitative structure-activity relationship (QSAR) modeling [1-3], discovering gene regulatory binding-site modules [4], and predicting clinical outcomes of cancer from gene expression data [5]. However, recent articles have indicated that predictive modeling approaches have not fully fulfilled expectations for solving real problems. This issue has for instance been discussed in the fields of QSAR [6] and microarray gene expression data modeling [7,8]. While some of the problems may be attributed to incorrect use and interpretation of the models, others can be ascribed to improper model selection and assessment. Our aim is here to address the latter issue by introducing the C^1C^2, a general framework for model choice and assessment.

Let $\boldsymbol{D} = \{\boldsymbol{X}_n, \boldsymbol{\gamma}_n\}$ be a dataset, where $\boldsymbol{X}_n = (\boldsymbol{x}'_1, ..., \boldsymbol{x}'_n)'$ is an $n \times p_X$ matrix whose ith row, $\boldsymbol{x}_i$, is the value of a p_X-vector of independent variables associated with γ_i, the ith row of the $n \times 1$ matrix, $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_n)'$. A statistical model, $\mathcal{M}_l$ can be used to characterize the relation between $\boldsymbol{X}_n$ and $\boldsymbol{\gamma}_n$. In general, given the dataset $\boldsymbol{D}$, $\mathcal{M}_l$ must be chosen from a set of models, $\mathcal{M} = \{\mathcal{M}_1, ..., \mathcal{M}_m, ..., \mathcal{M}_M\}$ according to some criterion (typically the minimization of a loss function). The most common way to select $\mathcal{M}_l$ from $\mathcal{M}$ in BBCC is to use K-fold cross-validation; that is, the dataset $\boldsymbol{D}$ is split into K mutually exclusive subsets, $\boldsymbol{D}_1$,..., $\boldsymbol{D}_k$,..., $\boldsymbol{D}_K$, of approximately equal size and $\mathcal{M}_l^0(\boldsymbol{D})$ is picked to minimize the function:

$$C^0(\mathcal{M}_m) = \frac{1}{n}\sum_{k=1}^{K}\sum_{i=1}^{n_{D_k}} L(\gamma_i, \gamma_i(x, \mathcal{M}_m, \boldsymbol{D}_{-k})), \qquad (1)$$

where L is a loss function and $\boldsymbol{D}_{-k}$ denotes that the kth subset was excluded from $\boldsymbol{D}$ during the model fitting; n_{D_i} is the number of observations in subset D_i; and $m = 1,..., M$, where M is the number of models in $\mathcal{M}$. This model selection method may seem straightforward and intuitive, however it neglects the fact that all the data at hand is used to make the model choice. Thus, we no longer have an independent testset to assess the chosen model by. The result is that, typically, $C^0(\mathcal{M}_l)$ underestimates the generalization error (see for instance [9]), defined as the expected prediction error over an independent test sample. This problem has been highlighted in relatively recent works [9,10], but was noted initially in 1974 [11]. To obtain a more accurate generalization error estimate, the model selection process must be separated from the model assessment in terms of the data that is used. Ideally, if data were abundant and easily produced, we would set aside a large test dataset and use it to assess – but not to choose! – the model $\mathcal{M}_l$, and subsequent model refinements could be assessed with new, unseen data. In practice, this is however often impossible since BBCC data is typically scarce, and expensive to produce. The luxury of large independent testsets can thus rarely be afforded. To tackle this problem, Freyhult et al. [9] suggested using a K-fold cross-validatory assessment of an H-fold cross-validatory model choice, $\mathcal{M}_l^\dagger$, as a way of simultaneously choosing $\mathcal{M}_l$ and assessing its performance; thereby separating the model selection from its assessment. The model is assessed by the function

$$C^\dagger = \frac{1}{n}\sum_{k=1}^{K}\sum_{i=1}^{n_{D_k}} L(\gamma_i, \hat{\gamma}_i(x_i, \mathcal{M}_l^\dagger(\boldsymbol{D}_{-k}), \boldsymbol{D}_{-k})), \qquad (2)$$

where $\mathcal{M}_l^\dagger(\boldsymbol{D}_{-k})$ is the cross-validatory choice of $\mathcal{M}_l$ based on $\boldsymbol{D}_{-k}$; that is, the model $\mathcal{M}_l$ that minimizes the function

$$C^\dagger_{\boldsymbol{D}_{-k}}(\mathcal{M}_m) = \frac{1}{n_{\boldsymbol{D}-k}}\sum_{h=1}^{H}\sum_{j=1}^{n_{D_h}} L(\gamma_j, \hat{\gamma}_j(x_j, \mathcal{M}_m, \boldsymbol{D}_{-hk})), \qquad (3)$$

where $\boldsymbol{D}_{-hk}$ denotes the dataset $\boldsymbol{D}$ with the kth and hth subsets omitted. In the present work we build on and generalize this idea into the C^1C^2 framework. In general, the number of models in $\mathcal{M}$ is huge, thus it is unfeasible to go through even a small subset of them manually. Hence, for a framework such as the C^1C^2 to be useful in practice, automated methods for searching the model space $\mathcal{M}$ are necessary; in this sense the C^1C^2 is similar to the automatic modelling approaches taken in for instance [12-14]. Here, the specific use of the C^1C^2 is demonstrated by applying it together with two search methods to simulated and real-world datasets. The results are compared to those obtained by employing the function (1) for model selec-

tion and assessment. In the interest of clarity, we have restricted our attention to the study of model choice and assessment within a linear model class, $\mathcal{M}^{ridge}$ (defined below) for a quadratic loss function. We discuss the results of the demonstrations, the pros and cons of the generality of the C^1C^2, and set out some directions for further research.

Results

Algorithm

Let C^1, $C^2 \in \boldsymbol{C} = \{C_1, ..., C_J\}$, where $\boldsymbol{C}$ is a set of model assessment criteria and C^1, C^2 represent two specific criteria (i.e. $C^1 = C_i$, $C^2 = C_j$, $i, j = 1, ..., J$). Further, let $S \in \boldsymbol{S}$, where $\boldsymbol{S}$ is a set of search methods; let $L \in \boldsymbol{L}$, where $\boldsymbol{L}$ is a set of loss functions; let $\boldsymbol{G}$ denote a sequence of data processing steps (e.g. mean-centering, transformations, whitening, etc) and let $\boldsymbol{G}'$ contain the results of $\boldsymbol{G}$ applied to $\boldsymbol{D}_{-k}$ (the roles of $\boldsymbol{G}$ and $\boldsymbol{G}'$ are exemplified in the discussion following the pseudococe below); let R be a positive integer and K an integer between 1 and n, where n is the number of observations. The C^1C^2 procedure is outlined with the following pseudocode:

Initiate $\mathcal{M}$, $\boldsymbol{G}$, L, C^1, C^2, R, and K.

for (r in 1,..., R) {

  a. Partition data, $\boldsymbol{D} = \{\boldsymbol{D}_k\}_{k = 1,..., k}$.

  for (k in 1,..., K) {

    b. Apply $\boldsymbol{G}$ to $\boldsymbol{D}_{-k}$. Save results in $\boldsymbol{G}'$.

    c. Search $\mathcal{M}$ using the data $\boldsymbol{D}_{-k}$ and C^2 as objective function. Assume $\mathcal{M}_l$ is found to maximize (or minimize) C^2. Save $\mathcal{M}_l$.

    d. Apply $\boldsymbol{G}$ to $\boldsymbol{D}_k$ using $\boldsymbol{G}'$.

    e. Assess $\mathcal{M}_l$ using C^1 and $\boldsymbol{D}_k$. Save assessment result.

  }

}

The data partitioning in (a) separates data for the model choice from data for the model assessment. Note that the partitioning is dependent on the choice of C^1 and does not necessarily need to be done in a cross-validation fashion. For instance, the choice C^1 = ".632 estimator" [15,16], partitions the data by independently sampling n rows from $\boldsymbol{D}$ with replacements and lets the observations not included among the sampled observations constitute the test set. The output from the C^1C^2 is also dependent on the choice of C^1; for example, the choice $C^1 = C^2$ = Bayesian Information Criterion (BIC, see [17] and Methods) would not give a direct estimation of the generalization error, but rather an assessment of model overfitting. To clarify the roles of $\boldsymbol{G}$ and $\boldsymbol{G}'$, we give the following example: Let $\boldsymbol{G}$ only contain a processing step that scales to unit variance. In (b) $\boldsymbol{G}$ is applied to $\boldsymbol{D}_{-i}$ and the standard deviation of each column of $\boldsymbol{D}_{-k}$ is saved in $\boldsymbol{G}'$. In (d), $\boldsymbol{G}'$ is applied to $\boldsymbol{D}_k$, that is, the columns in $\boldsymbol{D}_k$ are scaled using the standard deviations calculated in (b). This treatment of $\boldsymbol{G}$ ensures that $\boldsymbol{D}_k$ indeed constitutes an independent testset. The 'for loop' over r is introduced to enable calculation of confidence intervals for estimates and, by averaging the estimates over R repetitions, it permits reduction of the variance in parameter and error (or overfitting) estimates by a factor of $1/R$.

Figure 1 gives a graphical gives a graphical view of the C^1C^2 framework. We emphasize that the generality of the framework allows C^1, C^2, and S to be chosen to fit the problem at hand. Adequate choices of C^1, C^2, and S make the model selection and assessment more accurate and faster, which we will discuss below.

Datasets used in the testing

Both the simulated and the real data used for evaluating a new method or algorithm should reflect typical dataset properties found in real-world application domains. Examples of such properties in BBCC are multicollinearity, a large number of independent variables relative to the number of observations, and binary and categorical independent variables.

Simulated data

We simulated datasets as follows:

Let $\Delta = (\delta_i)_{i = 1,..., p}$, where $\delta_i = \begin{cases} 1 \text{ iff } \boldsymbol{x}_i \text{ is in the model} \\ 0 \text{ else} \end{cases}$, represents a subset of $\boldsymbol{X}_n$, and let $\beta_p(\Delta) = (\beta_i(\delta_i))_{i = 1,..., p}$, where $\beta_i(\delta_i) = \begin{cases} \beta_i \neq 0 \text{ iff } \delta_i = 1 \\ 0 \text{ else} \end{cases}$, is a vector of regression coefficients. The data matrix, $\boldsymbol{X}_n$ was sampled from a twenty-dimensional multivariate normal distribution. Thus, $\boldsymbol{x}_i \sim N_{20}(\boldsymbol{0}_{20}, \Sigma_{20})$, $i = 1, ..., n$, where $\boldsymbol{0}_{20}$ is a twenty-dimensional vector of zeroes and Σ_{20} is either the $\boldsymbol{I}_{20}$ identity matrix or a covariance matrix, $\boldsymbol{S}_{20}$, estimated from a real in-house QSAR dataset that originated from HIV protease inhibitors. The HIV QSAR dataset contains highly correlated independent variables resulting in an $\boldsymbol{S}_{20}$ with

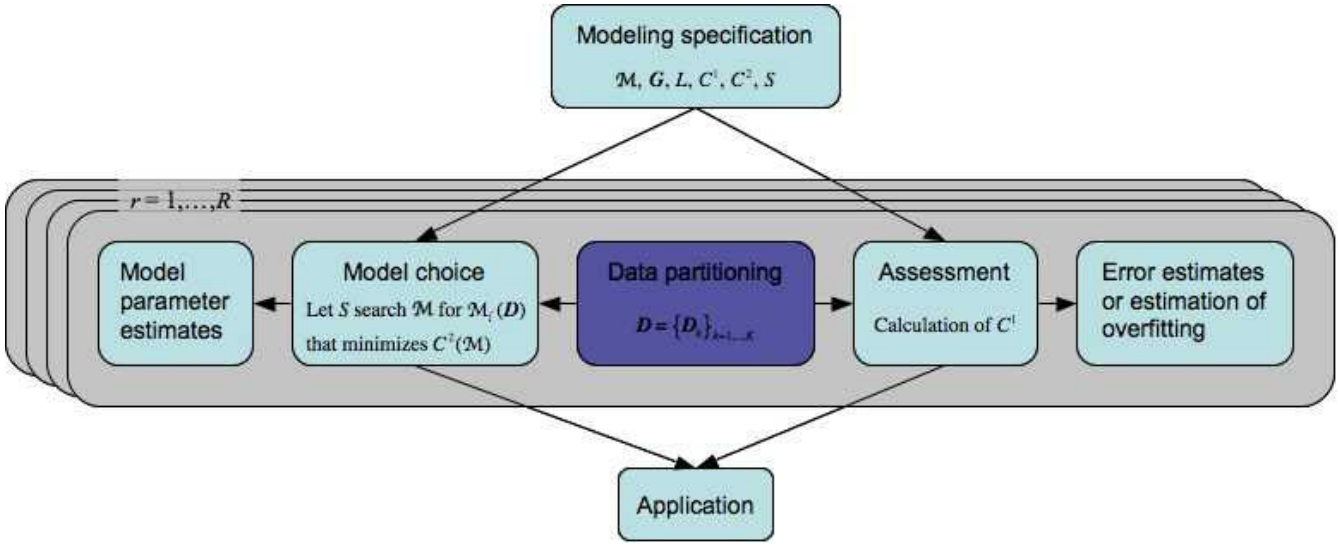

Figure 1
The C^1C^2 framework. The data partitioning in step (a) in the pseudocode separates the model choice from its assessment, which is highlighted in purple in the figure. The left side of the figure relates to steps (b) to (d) in the pseudocode, and the right side to step (e); i.e. the left side relates to choosing the model and saving the parameter estimates, and the right side to assessing the model and saving the assessment results.

many large absolute values in the off-diagonal elements. Three δ_i in Δ were chosen to be nonzero and equal to one; the positions were chosen at random to be 11, 14, and 18 (but remained fixed throughout the experiment for evaluation purposes). The corresponding regression coefficients, $\beta_{11}(\delta_{11})$, $\beta_{14}(\delta_{14})$, and $\beta_{18}(\delta_{18})$, were, for simplicity, all set equal to 1. The variables 11, 14, and 18 were slightly correlated with an estimated covariance matrix $\Sigma_{(11,14,18)} = \begin{pmatrix} 4.07 & 15.28 & -0.01 \\ 15.28 & 3423.20 & -0.33 \\ -0.01 & -0.33 & 0.01 \end{pmatrix}$..

Datasets were generated assuming that γ_n followed a linear model according to: $\gamma_n = \boldsymbol{X}_n \beta_p(\Delta) + \varepsilon_n$, where $\varepsilon_i \sim N(0,1.5)$. Four datasets were simulated in order to evaluate the C^1C^2s performance in settings where $n < p_x$, $n > p_x$, and where the observations were sampled from an orthogonal multivariate normal distribution or not, according to the following schema:

1. $n = 15$, $\Sigma_{20} = \boldsymbol{I}_{20}$

2. $n = 200$, $\Sigma_{20} = \boldsymbol{I}_{20}$

3. $n = 15$, $\Sigma_{20} = \boldsymbol{S}_{20}$

4. $n = 200$, $\Sigma_{20} = \boldsymbol{S}_{20}$

The simulated datasets are available in CSV format from Additional files 1, 2, 3, 4.

The Selwood dataset

This is a real dataset, made available from a website [18] and originally published in 1990 [19]. It is a widely studied dataset (see [20,21] and references therein). It contains one dependent variable, 53 independent variables, and 31 observations. The 53 independent variables correspond to numerical descriptions of molecules (antifilarial antimycin analogues) designed to capture their physicochemical properties. The dependent variable is the *in vitro* antifilarial activity of the molecules. This dataset exhibits extremely strong correlations between the independent variables and contains real valued, binary and categorical independent variables. It is known from previous studies that this dataset contains nonlinearities, but that decent models can be found using linear methods.

Testing

To demonstrate the use of the C^1C^2, it was applied to the simulated and real-world datasets described above (hereafter referred to as "the datasets"). Below we describe the choices for R, K, $\mathcal{M}$, $\boldsymbol{G}$, L, C^1, C^2, and S and the motivation for each selection.

Choice of R *and* K

The larger the choice of R, the higher the accuracy in the estimate of the generalization error; the choice of R is thus

constrained by time and is dependent on the size of the dataset and the computational complexity of the choices of $\mathcal{M}$, $\boldsymbol{G}$, L, C^1, C^2, and S. The choice of K is a trade-off between bias and variance; the larger the K, the more variance and the less bias in the estimates of the generalization error [22]. R was here set to 12 and K to 5.

Choice of $\mathcal{M}$

We make the assumption that a normal linear model forms a reasonable approximation of the data. This model is given by: $\boldsymbol{\gamma}_n = \boldsymbol{X}_n \beta_p + \varepsilon_n$, where the subscripts denote the number of rows in a matrix, β_p are regression coefficients, and $\varepsilon_n \sim N(0, \sigma^2 \boldsymbol{I})$. Further, because $n < p$ and the columns in $\boldsymbol{X}_n$ are highly correlated in some of the datasets, we decide to use the ridge estimator, $\hat{\beta}_p^{ridge}$ (see Methods), of the regression coefficients, β_p. Let $\mathcal{M}^{ridge}$ be the linear class of models given by: $\boldsymbol{\gamma}_n = \boldsymbol{X}_n \beta_p + \varepsilon_n$, where β_p is estimated by $\hat{\beta}_p^{ridge}$. We thus choose $\mathcal{M} = \mathcal{M}^{ridge}$. The problem of model choice within $\mathcal{M}^{ridge}$ reduces to the problem of variable selection, i.e. choosing which subset of the p columns in $\boldsymbol{X}_n$ to include in the model, and the problem of choosing the ridge penalizing parameter λ (see Methods). Hence, letting $\Delta = (\delta_i)_{i=1,\ldots,p}$ (see simulated data above) represent a subset of $\boldsymbol{X}_n$, we want to choose Δ and λ using the C^1C^2 framework. A choice of Δ and λ for given values of r and k will be termed "an estimate" of Δ and λ, respectively, and be denoted $\hat{\Delta}$ and $\hat{\lambda}$. Averages of estimates over the K folds and the R repeats in the C^1C^2 are denoted $\bar{\hat{\Delta}}$ and $\bar{\hat{\lambda}}$, respectively.

Choice of G

As the columns in the Selwood dataset are measured in different units using different scales, we choose to make $\boldsymbol{G}$ contain mean centering and scaling to unit variance processing steps.

Choice of L

We use the standard quadratic loss function given by:

$$L(\gamma_i, \gamma_i(\boldsymbol{x}_i, \mathcal{M}_m)) = (\gamma_i - \gamma_i(\boldsymbol{x}_i, \mathcal{M}_m))^2.$$

Choice of C^1 *and* C^2

Others [12,13] have suggested choosing $C^1 = C^2$ = cross-validation. Here, we choose C^1 = cross-validation and C^2 = BIC. Hence, in this demonstration we assess a model choice $\mathcal{M}_l^{ridge} \in \mathcal{M}^{ridge}$ according to:

$$C^1 = \frac{1}{n} \sum_{k=1}^{K} \sum_{i=1}^{n_{D_k}} L(\gamma_i, \gamma_i(x_i, \mathcal{M}_l^{ridge,BIC}(\boldsymbol{D}_{-k}), \boldsymbol{D}_{-k})), \tag{4}$$

where $\mathcal{M}_l^{ridge,BIC}$ $(\boldsymbol{D}_{-k})$ is the $\mathcal{M}_l^{ridge}$ chosen according to BIC based on $\boldsymbol{D}_{-k}$; that is, the value of $\mathcal{M}_l^{ridge}$ that optimizes the function:

$$C^2_{\boldsymbol{D}_{-k}}(\mathcal{M}_m^{ridge}) = \log \mathrm{P}(\boldsymbol{D}_{-k} \mid \beta^{ridge}, \mathcal{M}_m^{ridge}) - \frac{df}{2} \log n_{\boldsymbol{D}_{-k}}, \tag{5}$$

$m = 1,\ldots, M$, where M is the number of models in $\mathcal{M}^{ridge}$. df in (5) is the number of free parameters in the model $\mathcal{M}_m^{ridge}$ (note that this is not equal to the number of parameters in the model $\mathcal{M}_m^{ridge}$, see for instance [16]). The choice of C^1 is motivated by that we wish to get a direct estimate, $\hat{\varepsilon}_{gen}$, of the generalization error, ε_{gen}, of our model choice. Provided that the assumptions behind BIC are fulfilled, the choice C^2 = BIC has several advantages over C^2 = cross-validation, including: a reduction of bias in parameter estimates [22], a reduction of variance by the Rao-Blackwell type relation derived in [23], and a drastic reduction of the computational cost of the procedure.

Choice of S

A genetic algorithm (GA) was chosen as a search method because it is very easy to adapt to different situations and in general effective for nondeterministic polynomial-time hard combinatorial problems, such as the problem of estimating Δ [24]. A trial solution in the GA is here a varying length chromosome that contains a real-valued number representing λ and a vector of integers representing the indices of the δ_i in Δ that are nonzero. The fitness function is our choice of C^2. For the simulated datasets, we also chose to run the C^1C^2 with a brute force search method: for each $\lambda \in \{0, 0.01, 0.02, \ldots, 10\}$ an exhaustive search in the Δ space (i.e. an all-subset regression) was performed. This enabled comparisons between the GA method and a search method guaranteed to find the optimal model (given a specific objective function and the resolution and limits of λ).

Some remarks regarding the demonstration

To enable comparisons with the estimates of Δ, λ, and ε_{gen} obtained with repeated K-fold cross-validation, the demonstration described above was repeated with the func-

tion (1) used as a criterion for model choice and for assessing the model. Note that, since the C^1C^2 includes the 'for loop' over r, (1) was repeated $R = 12$ times, each time with a new, independent data partitioning. This was done to facilitate an impartial comparison between the two methods.

The demonstration of the C^1C^2 framework can be compared with the work of for instance Nicolotti and Carotti [20], where a genetic algorithm was employed to estimate Δ. In contrast to that approach, the C^1C^2 framework completely separates model choice and assessment whereby more accurate generalization error estimates in general are achieved. Further, the use of specific ad hoc objective functions is avoided by choosing C^2 to be a formally derived model selection criterion, and simultaneous estimation of model parameters other than Δ (for example, the ridge parameter λ in the demonstration) can be afforded. Typically, in works that have employed a search method for estimating Δ, a given number of nonzero δ_i in Δ is assumed (see for instance [20,25]). Therefore it was of interest to investigate the effect of this assumption on producing good estimates of Δ and ε_{gen}. This can be tested for the simulated datasets in the demonstration, where the number of nonzero δ_i is known. The C^1C^2 was therefore applied to the simulated datasets both with an assumption about the number of nonzero δ_i and without the assumption. For simplicity (however somewhat unrealistically), we assumed the correct number of nonzero δ_i.

Results of the testing

Simulated datasets

The four simulated datasets in combination with the use of either the C^1C^2 or repeated K-fold cross-validation for model choice and assessment, the GA or the brute-force search method, and either with or without the assumption of prior knowledge of the number of nonzero δ_i constitute a two-level, five-factor, full factorial experimental design. The C^1C^2 and the repeated K-fold cross-validation were applied four times to each factor combination, thus providing four replicates of the whole demonstration for the simulated data. The design can be analyzed within the normal linear model

$$w_{iv} = \gamma_0 + \gamma_1 z_{1i} + \gamma_2 z_{2i} + \gamma_3 z_{3i} + \gamma_4 z_{4i} + \gamma_5 z_{5i} + \eta\, i \ (i = 1,\ldots,128), \quad (6)$$

where z_{ji}, $j = 1,2,3,4,5$, are factors corresponding to C^1C^2 or repeated K-fold cross-validation model choice and assessment, brute force or GA search, $\Sigma_{20} = \boldsymbol{I}_{20}$ or $\Sigma_{20} = \boldsymbol{S}_{20}$ in the multivariate normal distribution from which the data was sampled, 200 or 15 observations, and assuming three nonzero δ_i or no such assumption, respectively. i goes from 1 to 128 in (6) as there are 32 factor combinations in four replicates. w_{iv}, $v = 1,2,3$, are response variables defined according to the following: the Euclidean norm $w_{i1} = \left\| \bar{\hat{\Delta}} - \Delta \right\|_i$ was used to measure how well Δ on average was estimated, $w_{i2} = \bar{\hat{\lambda}}_i$ was used as a response variable in the λ case (as the correct choice of λ is not known), and $w_{i3} = \left| \overline{\hat{\varepsilon}_{gen} - \tilde{\varepsilon}_{gen}} \right|_i$ was used to measure how well the generalization error ε on average was estimated; $\hat{\varepsilon}_{gen}$ denotes the estimate of ε_{gen} for given values of r and k; $\tilde{\varepsilon}_{gen} = \frac{1}{N}\sum_{j=1}^{N}(y_{j,ext} - y_{j,ext}(\boldsymbol{x}_{j,ext}, \mathcal{M}_l))^2$ denotes the generalization error estimate obtained by using the corresponding choice of model, $\mathcal{M}_l$, to predict the response values in a large ($N = 500{,}000$) external test set, generated in the same way as the dataset used for choosing the model and estimating $\hat{\varepsilon}_{gen}$; the bar denotes the average over the $R \cdot K$ individual estimates. The generalization error can be decomposed into three parts: one irreducible error (corresponding to the error added when simulating the data), the squared bias, and the variance. The latter two are dependent on the model choice and consequently the generalization error is dependent on the model choice. We here assume that the large-sample estimate of the generalization error, $\tilde{\varepsilon}_{gen}$, closely represents the true generalization error, ε, for a given model choice.

The results for choosing a model $\mathcal{M}_l^{ridge} \in \mathcal{M}^{ridge}$ for the simulated datasets are available in Additional file 5, where $\left\| \bar{\hat{\Delta}} - \Delta \right\|$, $\bar{\hat{\lambda}}$, and $\left| \overline{\hat{\varepsilon}_{gen} - \tilde{\varepsilon}_{gen}} \right|$ are tabulated for each factor combination and replicate. The parameter estimates for fitting the model (6) using $\left\| \bar{\hat{\Delta}} - \Delta \right\|$, $\bar{\hat{\lambda}}$, and $\left| \overline{\hat{\varepsilon}_{gen} - \tilde{\varepsilon}_{gen}} \right|$ as response variables are shown in Tables 1, 2, and 3, respectively. All fitted models were highly significant ($F_{5,122} = 26.1$, p-value $< 2.2 \times 10^{-16}$ with $\left\| \bar{\hat{\Delta}} - \Delta \right\|$ as response; $F_{5,122} = 47.7$, p-value $< 2.2 \times 10^{-16}$ with $\bar{\hat{\lambda}}$ as response; and $F_{5,122} = 12.1$, p-value $= 1.6 \times 10^{-9}$ with $\left| \overline{\hat{\varepsilon}_{gen} - \tilde{\varepsilon}_{gen}} \right|$ as response); residual plots showed no large deviations from the assumptions of normality of error distribution (an asymptotic normal distribution of the response variables is warranted by the central limit theo-

Table 1: Coefficient estimates of model (6) with $w_{i1} = \left\| \bar{\hat{\Delta}} - \Delta \right\|_i$, $i = 1, ..., 128$, as a response variable.

	Estimate	Std.Error	t-value	Pr(>|t|)
intercept	-0.01992	0.04606	-0.433	0.6661
c1c2	-0.04337	0.03761	-1.153	0.2511
ga	0.15683	0.03761	4.170	5.72e-05
cor	0.07211	0.03761	1.918	0.0575
15	0.21324	0.03761	5.670	9.75e-08
all	0.32754	0.03761	8.710	1.78e-14

c1c2 – the C^1C^2 was used (as opposed to repeated K-fold cross-validation), **ga** – the GA search method was used (as opposed to the brute force search method), **cor** – correlated independent variables in the dataset (as opposed to uncorrelated,) **15** – $n = 15$ observations in the dataset (as opposed to $n = 200$), **all** – no assumption regarding the number of nonzero δ_i (as opposed to the assumption of three $\delta_i = 1$).

rem), homoscedasticity, and independent errors (data not shown). A few outliers were however observed, probably resulting from "unfortunate" data partitions.

Selwood dataset

Applying the C^1C^2 to the Selwood dataset yielded $\hat{\Delta}$ that on average contained 11.4 ± 3.5 nonzero $\hat{\delta}_i$. The 11 most frequently picked variables were the partial atomic charge for atoms ATCH1, ATCH3, and ATCH6; electrophilic superdelocalizability for atom ESDL3; the van der Waal's volume VDWVOL; the surface area SURF_A; the principal moments of inertia MOFI_Y and MOFI_Z; the principal ellipsoid axis PEAX_X; the partition coefficient LOGP; and the sum of F substituent constant SUM_F (see [19] for more details about the variables). The estimation of λ by $\bar{\hat{\lambda}}$ was 2.50 ± 0.09 and the estimation of the generalization error by $\bar{\hat{\varepsilon}}_{gen}$ was 0.42 ± 0.038, where $\bar{\hat{\varepsilon}}_{gen}$ is an average over the $R \cdot K$ $\hat{\varepsilon}_{gen}$ produced in the C^1C^2.

Table 2: Coefficient estimates of model (6) with $w_{i2} = \bar{\hat{\lambda}}_i$, $i = 1, ..., 128$ as a response variable.

	Estimate	Std.Error	t-value	Pr(>|t|)
Intercept	0.02864	0.04732	0.605	0.546181
c1c2	-0.04804	0.03863	-1.244	0.216065
ga	-0.07329	0.03863	-1.897	0.060193
cor	0.56058	0.03863	14.510	< 2e-16
15	0.14307	0.03863	3.703	0.000321
all	0.11504	0.03863	2.977	0.003506

See Table 1 for notation explanation.

Table 3: Coefficient estimates of model (6) with $w_{i3} = \overline{\left| \hat{\varepsilon}_{gen} - \tilde{\varepsilon}_{gen} \right|}_i$, $i = 1, ..., 128$ as a response variable.

	Estimate	Std.Error	t-value	Pr(>|t|)
intercept	0.034642	0.004841	7.157	6.73e-11
c1c2	-0.024003	0.003952	-6.073	1.47e-08
ga	-0.001149	0.003952	-0.291	0.771710
cor	0.006198	0.003952	1.568	0.119423
15	0.013469	0.003952	3.408	0.000888
all	-0.012089	0.003952	-3.059	0.002732

See Table 1 for notation explanation.

Applying repeated K-fold cross-validation for model choice and assessment to the Selwood dataset gave on average 14.1 ± 4.8 selected variables. The 14 most frequently picked variables included the same 11 variables picked by the C^1C^2 (see above) plus DIPMOM (the dipole moment), ATCH7 (partial charge of atom 7), and DIPV_Y (the dipole moment vector in the Y-direction). The estimation of λ by $\bar{\hat{\lambda}}$ was 3.01 ± 0.22 and the estimation of the generalization error by $\bar{\hat{\varepsilon}}$ was 0.35 ± 0.041, where $\bar{\hat{\varepsilon}}$ is an average over the $R \cdot K$ $\hat{\varepsilon}$ produced in the repeated K-fold cross-validation.

Implementation

Computer programs to implement the C^1C^2 were written in Java (Sun Microsystems [26]) as a part of the library P, that will serve as the data analysis plugin for Bioclipse [27]. P is available under the GSPL license from the website [28]; it is open source and free for academics. It has a modular architecture that enables plugging in new features, including modeling methods, model selection criteria, and search procedures. P relies on a modified version of the JGap library (available from the website [29]) for the genetic algorithm computations (the modifications are available under the LGPL license from the website [30]). The R-package, pvclust [31,32], was used for the cluster analysis (see Discussion).

Discussion

Simulated datasets

The model (6) fitted to $w = \left\| \bar{\hat{\Delta}} - \Delta \right\|$ (see Table 1) showed a relatively clear significant difference (on the 90% level) in average Δ estimates depending on whether the data came from a multivariate normal distribution with $\Sigma_{20} = I_{20}$ or $\Sigma_{20} = S_{20}$. Furthermore, we observed significant positive impacts on average Δ estimates with more observations and knowledge about the number of nonzero δ_i. All these findings were expected; highly correlated variables

should provide worse estimates of Δ, whereas more observations and trustworthy prior knowledge should provide better estimates. A significant improvement in average Δ estimates was observed when using the brute-force search compared to the GA. The GA on average selected slightly more variables than needed and than what the brute-force method did. No clear significant difference could be seen between using the C^1C^2 rather than repeated *K*-fold cross-validation.

It can be shown that the optimal choice of λ (in terms of minimized expected generalization error) tends to zero as the number of observations tends to infinity and decreases with decreasing number of variables (see for instance [33]) and with decreasing correlations between the independent variables. The model (6) fitted with $w = \bar{\hat{\lambda}}$ as a response (see Table 2) showed that the average estimated λ was significantly smaller for the data that came from a multivariate normal distribution with $\Sigma_{20} = \boldsymbol{I}_{20}$ compared to $\Sigma_{20} = \boldsymbol{S}_{20}$, when more observations were used, and when prior knowledge about the number of nonzero δ_i in Δ was assumed. Although the true value of λ is not known, these results are thus consistent with theory and provide evidence that both the C^1C^2 and repeated *K*-fold cross-validation gave reasonable estimates of λ in the demonstration. However, the average λ estimates are not equal to zero for all orthogonal datasets, presumably due to the stochastic nature of the GA and to errors in $\boldsymbol{\gamma}_n$. No significant differences were observed between using the GA or the brute force search methods or between the C^1C^2 and the repeated *K*-fold cross-validation.

Fitting model (6) with $w = \overline{\left|\hat{\varepsilon}_{gen} - \tilde{\varepsilon}_{gen}\right|}$ as a response (see Table 3) showed that the average error estimates were significantly worsened with the assumption of a given number of nonzero δ_i and that no significant difference was observed when using the GA or the brute force method, or when the independent variables in the dataset were correlated or not. These findings might seem confusing given that the assumption of a given number of nonzero δ_i, the use of the brute-force search method, and uncorrelated independent variables all improved model selection. The findings can be explained by the fact that, in general, without an assumption of a given number of nonzero δ_i, when using the GA for searching the model space, and when independent variables were correlated, more nonzero δ_i are on average selected (see Table 1). Thus the chances of also selecting the correct ones improve. This implies that it is worse to estimate at least one $\delta_i = 1$ with $\hat{\delta}_i = 0$ than to estimate all $\delta_i = 1$ with $\hat{\delta}_i = 1$ and at least one $\delta_i = 0$ with $\hat{\delta}_i = 1$. This makes sense, because the former models are incorrect, whereas the latter ones contain the true model, but are inefficient due to their unnecessary large size. The average error estimates were significantly improved with a large number of observations and when the C^1C^2 was employed to produce the estimates compared to when the repeated *K*-fold cross-validation was used (see Fig. 2 and Table 3). The latter result seems contradictory with that no clear difference was found between the average Δ estimates produced with the C^1C^2 and those obtained with the repeated *K*-fold cross-validation (see above). It can however be explained by studying the $R \cdot K$ individual Δ estimates, where a clear (99% level) positive effect could be observed when using the C^1C^2 compared to the repeated *K*-fold cross-validation. The *individual* Δ estimates were thus worse when repeated *K*-fold cross-validation was used, resulting in worse generalization error estimates. However *the average* Δ estimates from the respective method were almost the same. This observation is seconded by the higher confidence intervals of the average Δ estimates produced with repeated *K*-fold cross-validation (see Additional file 5). The finding that the C^1C^2 produces more accurate generalization error estimates than repeated *K*-fold cross-validation is consistent with the results presented in for instance [9] and provides evidence for that a complete separation of the data used for model choice and the data used for model assessment is necessary to obtain better estimates of the generalization error.

Selwood dataset

The result of estimating Δ was, expectedly, less clear when applying the C^1C^2 and repeated *K*-fold cross-validation to real-world data; the Selwood dataset is particularly difficult to model due to the extremely high correlations between variables (many variable pairs have correlation coefficients > 0.95), the low observation-to-variable ratio, and deviations from the linearity and homoscedasticity assumptions (see [20]). 11.4 out of the 53 variables were on average selected by the C^1C^2 and 14.1 by *K*-fold cross-validation. Interestingly, the 11 most frequently picked variables selected by the C^1 C^2 is a proper subset of the 14 most recurrently selected variables by *K*-fold cross-validation. Hierarchically clustering the 14 most frequently picked variables chosen by *K*-fold cross-validation (which thus includes the 11 variables selected most often by the

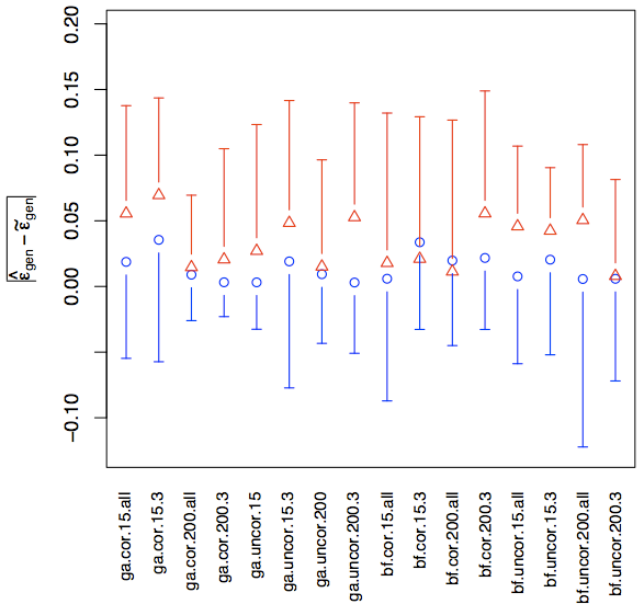

Figure 2

Generalization errors obtained with the C^1C^2 and repeated *K*-fold cross-validation. The figure shows $\overline{\left|\hat{\varepsilon}_{gen} - \tilde{\varepsilon}_{gen}\right|}$, where $\hat{\varepsilon}_{gen}$ were produced using the C^1C^2 (blue) and repeated *K*-fold cross-validation (red) for all other factor combinations in model (6). The plot is based on pooled $\overline{\left|\hat{\varepsilon}_{gen} - \tilde{\varepsilon}_{gen}\right|}$ over the four replicates for each method. The bars show the 95% confidence interval, calculated from the pooled results (the confidence intervals are only shown in one direction to avoid cluttering). The factor combinations in model (6) are coded as: **ga** – the GA search method was used, **bf** – the brute force search method was used, **uncor** – orthogonal independent variables in the dataset, **cor** – correlated independent variables in the dataset, **15** – *n* = 15 observations in the dataset, **200** – *n* = 200 observations in the dataset, **all** – no assumption regarding the number of nonzero δ_i, **3** – three $\delta_i = 1$ were assumed.

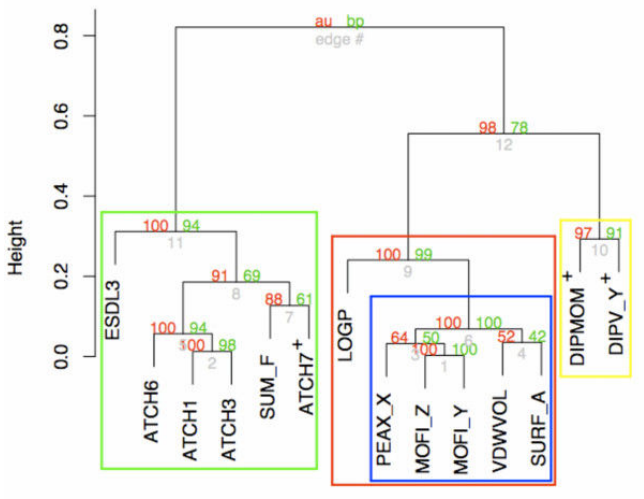

Figure 3

Cluster dendrogram of the 14 selected variables from the Selwood dataset using repeated *K*-fold cross-validation. Three distinct clusters can be noted (shown in red, green, and yellow rectangles). One sub-cluster can be seen within the red cluster (shown in a blue rectangle). The red and green numbers are *p*-values of a given cluster; they indicate how well the cluster is supported by data (see [31] for details). +Additional variables selected by repeated *K*-fold cross-validation compared to the C^1C^2.

C^1C^2) using the absolute correlation as a distance measure revealed three distinct clusters and one subcluster (see Fig. 3). Good models (in terms of estimated generalization error) for the Selwood dataset can be achieved by selecting LOGP and one variable from the set of variables in the blue subcluster (PEAX_X, MOFI_Y, MOFI_Z, VDWVOL, and SURF_A) and one from the set of variables in the green cluster (ESDL3, ATCH1, ATCH3, ATCH6, ATCH7, and SUM_F). LOGP appears to be sufficiently different from the rest of the variables in the red cluster to improve model performance. The variables in the respective blue and green clusters are highly correlated and it is sufficient to have one variable from each cluster in a model.

Models containing LOGP and one variable each from the green and blue clusters have high predictive power and comply to the QUICK rules for credible predictive models proposed previously [21]. Furthermore, these models have been found credible in the works of others [20,25,34]. The C^1C^2 chose 11 variables belonging to the green, red, and blue clusters, whereas *K*-fold cross-validation chose an additional three variables: ATCH7 in the green cluster, and DIPMOM and DIPV_Y belonging to the third distinct (yellow) cluster. Variables from the yellow cluster do not improve the internal predictive ability when testing models containing LOGP and one variable from the respective green, blue, and yellow clusters on the whole Selwood dataset (data not shown); this result is supported by the work of Nicolotti and Carotti (see Table 1 in [20]). More variables in the Selwood dataset were thus on average selected with repeated *K*-fold cross-validation than when using the C^1C^2 (the difference was significant on the 80% level, tested by a one-sided Welch's *t*-test), including two that not seem to improve the predictive ability of the models. The generalization estimates obtained with the *K*-fold cross-validation was lower than those obtained with the C^1C^2 (significant on the 70% level). Although these differences are not highly significant, it is tantalizing to arrive at the conclusion that the models selected by repeated *K*-fold cross-validation in this

particular case are more prone to overfitting and that this is reflected in the lower generalization error estimates.

The C^1C^2

We have here introduced the C^1C^2 framework for simultaneous model choice and assessment. The main idea is a complete separation of the choice of a model and its assessment in terms of the data used for each task. The C^1C^2 was applied to the problem of choosing a model, $\mathcal{M}_l^{ridge} \in \mathcal{M}^{ridge}$. Previously, others have described methods that, within the linear model, tackle the problem of regression coefficient shrinkage and variable selection simultaneously, for example, the lasso [35]. However, the C^1C^2 framework is general and is easily applied to other settings. For instance, different choices of C^2 are favorable in different situations; Akaike's information criterion (AIC) [36] is known to be consistent within the linear model if the true model is very complex, whereas BIC is favorable within linear models of finite dimension [37], and cross-validation is preferable to use in situations where the degrees of freedom of a model is difficult to define, and so forth. The search method can also be tailored to the problem at hand; for instance, brute-force methods are advantageous for small problems, whereas GAs are faster and thus applicable to larger problems. Moreover, if required, $\mathcal{M}$ can be chosen to contain nonlinear models, L can be chosen to be exponential in order to increase the penalty on outliers, and instead of using the search method to produce an estimate of Δ (as we did in the demonstration) we can let $\mathbf{G}$ contain a dedicated variable selection method. The cost of this generality is uncharacterized convergence rates (in finite time) of the parameter estimates, which is coupled to the need of employing a general search method (like a GA) rather than solving standard convex problems. Running the C^1C^2 R times enables averaging of estimates and calculation of confidence intervals, but renders problems in choosing which out of the R models to use for interpretation and future predictions. A potential remedy to these problems is, instead of choosing *a* model, to employ *all* chosen models in a stacking-like schema (see [38] for details on stacking). Testing this idea and further testing of the C^1C^2 for other choices of $\mathcal{M}$, $\mathbf{G}$, L, C^1, C^2, and S will be pursued in future research.

Conclusion

We have presented some evidence that suggests that the C^1C^2 works well in terms of choosing the correct model and produce good estimates of the generalization error. It was demonstrated to perform well within a penalized linear model, even for "difficult" datasets with highly correlated independent variables, a low observation-to-variable ratio, and deviations from model assumptions (see Table 4 for a summary of the findings in the demonstrations). However, more research is needed to fully assess the methods performance for more general, for instance nonlinear, models and to provide theoretical insight to frameworks such as the C^1C^2. The C^1C^2 is general and reasonable choices of $\mathcal{M}$, $\mathbf{G}$, L, C^1, C^2, and S help in achieving as unbiased estimates with as low a variance to as low a computational cost as possible. A framework that completely separates model choice from assessment in terms of used data, like the C^1C^2, should always be employed for model selection and assessment in order to avoid positive bias in the generalization error estimates and, ultimately, to avoid false conclusions and using dubious models to direct further research.

Methods

Bayesian Information Criterion (BIC)

Suppose we have a set of candidate models, $\mathcal{M}$ and corresponding model parameters, θ_m, and we wish to choose the best model among $\mathcal{M}$. Assuming we have a prior distribution, $P(\theta_m|\mathcal{M}_m)$ for the parameters of each model, $\mathcal{M}_m \in \mathcal{M}$, the posterior probability of a given model is:

Table 4: Summary of the demonstrations of the C^1C^2.

Both the C^1C^2 and repeated K-fold cross-validation performed well at finding the true Δ (even when independent variables are highly correlated and when $n < p$).
The C^1C^2 and repeated K-fold cross-validation produced reasonable estimates of λ.
Prior information about the number of important independent variables improves model choice but can reduce the accuracy of generalization error estimates.
Correlated independent variables and using the genetic algorithm worsened the model choice significantly, but not the generalization error estimates.
The C^1C^2 compares favourably with repeated K-fold cross-validation for assessing the generalization error.

n denotes the number of observations in a dataset, p the number of variables, Δ represents a given subset of the p variables, and λ the ridge regression parameter.

$$\mathrm{P}(\mathcal{M}_m \mid \boldsymbol{D}) \propto \mathrm{P}(\mathcal{M}_m) \cdot \mathrm{P}(\boldsymbol{D} \mid \mathcal{M}_m) \propto$$
$$\mathrm{P}(\mathcal{M}_m) \cdot \int \mathrm{P}(\boldsymbol{D} \mid \theta_m, \mathcal{M}_m)\mathrm{P}(\theta_m \mid \mathcal{M}_m) d\theta_m.$$

To choose a model in a Bayesian setting, the posterior odds, given by:

$$\frac{\mathrm{P}(\mathcal{M}_l|\boldsymbol{D})}{\sum_{m=1}^{M}\mathrm{P}(\mathcal{M}_m|\boldsymbol{D})} = \frac{\mathrm{P}(\mathcal{M}_l)\mathrm{P}(\boldsymbol{D}|\mathcal{M}_l)}{\sum_{m=1}^{M}\mathrm{P}(\mathcal{M}_m)\mathrm{P}(\boldsymbol{D}|\mathcal{M}_m)} \qquad (7)$$

are formed for all models $\mathcal{M}_l \in \mathcal{M}$, and $\mathcal{M}_l$ is picked to maximize equation (7). If all models in $\mathcal{M}$ are given equal prior probabilities, the problem of choosing the model $\mathcal{M}_l$ is reduced to calculating the integrals, $\mathrm{P}(\boldsymbol{D}|\mathcal{M}_m)$. It can be shown [39,40] that the Bayesian Information Criterion (BIC) approximates the logarithm of this integral with an $O(1)$ error term, that is:

$$\log \mathrm{P}(\boldsymbol{D} \mid \mathcal{M}_m) = BIC + O(1),$$

where

$$BIC = \log \mathrm{P}(\boldsymbol{D} \mid \theta_m, \mathcal{M}_m) - \frac{df_m}{2} \log n.$$

In the latter expression, $\hat{\theta}_m$ is the maximum likelihood estimate, df_m is the number of free parameters in model $\mathcal{M}_m$ (note that this in general is not equal to the number of parameters in the model), and n is the number of observations [17]. Thus, BIC is an approximation to the Bayes solution, but valid outside the Bayesian context. This is true because the leading terms in the approximation do not depend on the prior densities of the model parameters, θ_m. BIC is, as opposed to nonparametric approaches such as cross-validation, model based and therefore relies on the assumptions made in the modeling. BIC is derived under the assumption that the data comes from a distribution in the exponential family (see [41] for more about the assumptions behind BIC and a comparison with Akaike's Information Criterion).

Ridge regression

The ordinary least squares (OLS) estimator of the regression coefficients β in the standard linear model is efficient (i.e. has the minimum possible variance) within the class of linear and unbiased estimators. However, when the independent variables are correlated, the variance of the OLS estimator is generally high. In these situations, ridge regression [42] can yield improved parameter estimates by minimizing a penalized residual sum of squares, given by: $RSS\ (\lambda) = (\gamma - X\beta)^T (\gamma - X\beta) + \lambda\beta^T\beta$. Finding the minimum of this expression gives the ridge solution: $\hat{\beta}^{ridge} = (X^T X + \lambda I)^{-1} X^T \gamma$, where I is the $p \times p$ identity matrix. The solution thus adds a positive constant to the diagonal of $X^T X$ before inversion; this makes the problem nonsingular, even if $X^T X$ is not of full rank. While this introduces bias into the coefficient estimates, variance is often greatly reduced.

Note that $\hat{\beta}^{ridge}$ is a linear function in γ, thus it is straightforward to define the effective degrees of freedom of the ridge regression fit, (df(λ)) as:

df (λ) = tr $[X(X^T X + \lambda I)^{-1} X^T]$ [16]. The degrees of freedom of the fit are needed for carrying out model selection according to, for instance, BIC. Linearity in γ also enables easy implementation (no quadratic programming required as, for instance, is necessary with the lasso).

Genetic algorithm (GA)

A GA (see [43] for more details) is a stochastic search technique for finding exact or approximate solutions to optimization and search problems. A typical genetic algorithm is defined by a genetic representation of a given solution (normally termed a *chromosome* in the GA context). That is, a vector, $\mathbf{w}_t^i$ specifies the numerical representation of the ith chromosome at generation t, and an objective function (or *fitness function*), $f(\boldsymbol{w}_t^i) \rightarrow /$ ℝ/evaluates the fitness of a chromosome. The GA is initiated by setting up a random population that contains a number of trial chromosomes. New solutions are generated by mutation or recombination of existing solutions and are selected for the next generation with a probability given by: $p(\boldsymbol{w}_t^i) = f(\boldsymbol{w}_t^i) \Big/ \sum_j^a f(\boldsymbol{w}_t^i)$. The process is continued through a number of generations until an optimal or acceptable solution has been found. Genetic algorithms of this type can be shown to converge with a probability of one to the global optimal solution as $t \rightarrow \infty$.

Brute force search

A **brute force search** systematically tests an exhaustive list of all possible candidates for the solution to a given search or optimization problem and checks whether each candidate satisfies the problem's statement.

Availability and requirements

• Project name: P

• Project homepage: http://www.genettasoft.com/p/P.zip

• Operating systems: Platform independent (interpreted language)

• Programming language: Java

• Requirements: Java 5 or higher. A modified version of the JGAP package http://jgap.sourceforge.net/ for genetic algorithms. The modifications are distributed under the LGPL license and are available at http://www.genettasoft.com/p/JGAPm.zip. log4j, available from http://logging.apache.org/log4j/.

• Licence: GSPL (see http://www.genettasoft.com/gspl/gspl1_1.pdf)

• Restrictions to use for commercial purposes: licence needed

Authors' contributions

ME devised and implemented the proposed C^1C^2 framework. OS was involved in program design and aided with the implementation. JESW supervised the project. All authors read and approved the final manuscript.

Additional material

Additional file 1

The first of the four simulated datasets used in the demonstrations. This dataset with 15 observations was sampled from a multivariate normal distribution with mean $\mathbf{0}_{20}$ *and covariance matrix* $\mathbf{I}_{20}$ *(see article for details).*
Click here for file
[http://www.biomedcentral.com/content/supplementary/1471-2105-9-360-S1.csv]

Additional file 2

The second of the four simulated datasets used in the demonstrations. This dataset with 200 observations was sampled from a multivariate normal distribution with mean $\mathbf{0}_{20}$ *and covariance matrix* $\mathbf{I}_{20}$ *(see article for details).*
Click here for file
[http://www.biomedcentral.com/content/supplementary/1471-2105-9-360-S2.csv]

Additional file 3

The third of the four simulated datasets used in the demonstrations. This dataset with 15 observations was sampled from a multivariate normal distribution with mean $\mathbf{0}_{20}$ *and covariance matrix* $\mathbf{I}_{20}$ *(see article for details).*
Click here for file
[http://www.biomedcentral.com/content/supplementary/1471-2105-9-360-S3.csv]

Additional file 4

The fourth of the four simulated datasets used in the demonstrations. This dataset with 200 observations was sampled from a multivariate normal distribution with mean $\mathbf{0}_{20}$ *and covariance matrix* $\mathbf{S}_{20}$ *(see article for details).*
Click here for file
[http://www.biomedcentral.com/content/supplementary/1471-2105-9-360-S4.csv]

Additional file 5

C^1C^2 and repeated K-fold cross-validation estimates of Δ, λ, and ε for the simulated data. *The factor combinations in model (6) are coded as:* ***c1c2*** *– the C^1C^2 was used,* ***k-fold*** *– repeated K-fold cross-validation was used,* ***ga*** *– the GA search method was used,* ***bf*** *– the brute force search method was used,* ***15*** *– n = 15 observations in the dataset,* ***200*** *– n = 200 observations in the dataset,* ***uncor*** *– orthogonal independent variables in the dataset,* ***cor*** *– correlated independent variables in the dataset,* ***all*** *– no assumption regarding the number of nonzero δ_i,* ***3*** *– three $\delta_i = 1$ were assumed. All values are means ± 95% confidence intervals, assuming normal distributions for* $\left\| \bar{\hat{\Delta}} - \Delta \right\|$, $\bar{\hat{\lambda}}$, *and* $\overline{\left| \hat{\varepsilon}_{gen,i} - \tilde{\varepsilon}_{gen,i} \right|}$.
Click here for file
[http://www.biomedcentral.com/content/supplementary/1471-2105-9-360-S5.xls]

Acknowledgements

Supported by the Swedish VR (04X-05975). We extend out gratitude to three anonymous reviewers who helped to improve this paper.

References

1. Kontijevskis A, Prusis P, Petrovska R, Yahorava S, Mutulis F, Mutule I, Komorowski J, Wikberg JES: **A look inside HIV resistance through retroviral protease interaction maps.** *PloS Computational Biology* 2007, **3(3):**.
2. Wikberg JES, Lapinsh M, Prusis P: **Proteochemometrics: A tool for modelling the molecular interaction space.** In *Chemogenomics in Drug Discovery - A Medicinal Chemistry Perspective* Edited by: Kubinyi H, Müller G. Weinheim , Wiley-VCH; 2004:289-309.
3. Hansch C: **A Quantitative Approach to Biochemical Structure-Activity Relationships.** *Accounts of Chemical Research* 1969, **2:**232-239.
4. Hvidsten TR, Wilczynski B, Kryshtafovych A, Tiuryn J, Komorowski J, Fidelis K: **Discovering regulatory binding-site modules using rule-based learning.** *Genome Res* 2005/06/03 edition. 2005, **15(6):**856-866.
5. van't Veer LJ, Dai H, van de Vijver MJ, He YD, Hart AA, Mao M, Peterse HL, van der Kooy K, Marton MJ, Witteveen AT, Schreiber GJ, Kerkhoven RM, Roberts C, Linsley PS, Bernards R, Friend SH: **Gene expression profiling predicts clinical outcome of breast cancer.** *Nature* 2002, **415(6871):**530-536.
6. Johnson SR: **The Trouble with QSAR (or How I Learned To Stop Worrying and Embrace Fallacy).** *J Chem Inf Model* 2008:25-26.
7. Michiels S, Koscielny S, Hill C: **Prediction of cancer outcome with microarrays: a multiple random validation strategy.** *The Lancet* 2005, **365(9458):**488-492.
8. Ntzani EE, Ioannidis JPA: **Predictive ability of DNA microarrays for cancer outcomes and correlates: an empirical assessment.** *The Lancet* 2003, **362(9394):**1439-1444.
9. Freyhult E, Peteris P, Lapinsh M, Wikberg JES, Moulton V, Gustafsson MG: **Unbiased descriptor and parameter selection confirms the potential of proteochemometric modelling.** *BMC Bioinformatics* 2005, **6(50):**1-14.

10. Golbraikh A, Tropsha A: **Beware of q2!** *J Mol Graph Model* 2002/02/23 edition. 2002, **20(4):**269-276.
11. Stone M: **Cross-Validatory Choice and Assessment of Statistical Predictions.** In *Journal of the Royal Statistical Society Series B (Methodological) Volume 36*. Issue 2 Royal Statistical Society; 1974:111-147.
12. Cartmell J, Enoch S, Krstajic D, Leahy DE: **Automated QSPR through Competitive Workflow.** *J Comput Aided Mol Des* 2005, **19(11):**821-833.
13. Cartmell J, Krstajic D, Leahy DE: **Competitive Workflow: novel software architecture for automating drug design.** *Curr Opin Drug Discov Devel* 2007, **10(3):**347-352.
14. Obrezanova O, Gola JM, Champness EJ, Segall MD: **Automatic QSAR modeling of ADME properties: blood-brain barrier penetration and aqueous solubility.** *J Comput Aided Mol Des* 2008, **22(6-7):**431-440.
15. Efron B, Tibshirani R: **An Introduction to the Bootstrap.** New York , Chapman & Hall/CRC. ; 1993.
16. Hastie T, Tibshirani R, Friedman J: **The Elements of Statistical Learning.** In *Springer series in statistics* New York , Springer-Verlag; 2001:533.
17. Schwarz G: **Estimating the Dimension of a Model.** In *The Annals of Statistics Volume 6*. Issue 2 Institute of Mathematical Statistics; 1978:461-464.
18. **The Selwood dataset** [http://www.ndsu.edu/qsar_soc/resource/datasets/selwood.htm]
19. Selwood DL, Livingstone DJ, Comley JCW, O'Dowd AB, Hudson AT, Jackson P, Jandu KS, Rose VS, Stables JN: **Structure-activity relationships of antifilarial antimycin analogs: a multivariate pattern recognition study.** *Journal of Medicinal Chemistry* 1990, **33(1):**136-142.
20. Nicolotti O, Carotti A: **QSAR and QSPR studies of a highly structured physicochemical domain.** *J Chem Inf Model* 2006/01/24 edition. 2006, **46(1):**264-276.
21. Todeschini R, Consonni V, Mauri A, Pavan M: **Detecting "bad" regression models: multicriteria fitness functions in regression analysis.** *Analytica Chimica Acta* 2004, **515(1):**99-208.
22. Burman P: **A Comparative Study of Ordinary Cross-Validation, v-Fold Cross-Validation and the Repeated Learning-Testing Methods.** In *Biometrika Volume 76*. Issue 3 Biometrika Trust; 1989:503-514.
23. Efron B: **The Estimation of Prediction Error: Covariance Penalties and Cross-Validation.** In *Journal of the American Statistical Association Volume 99*. American Statistical Association; 2004:619-632.
24. Amaldi E, Kann V: **On the Approximability of Minimizing Nonzero Variables Or Unsatisfied Relations in Linear Systems.** *Theoretical Computer Science* 1997, **209:**237–260.
25. Kubinyi H: **Variable Selection in QSAR Studies. II. A Highly Efficient Combination of Systematic Search and Evolution.** *QSAR & Combinatorial Science* 1994, **13(4):**393-401.
26. **Java - The Source for Java Developers** [http://java.sun.com/]
27. Spjuth O, Helmus T, Willighagen EL, Kuhn S, Eklund M, Wagener J, Murray-Rust P, Steinbeck C, Wikberg JE: **Bioclipse: an open source workbench for chemo- and bioinformatics.** *BMC Bioinformatics* 2007/02/24 edition. 2007, **8:**59.
28. **P** [http://www.genettasoft.com/p/P.zip]
29. **JGAP - Java Genetic Algorithms Package** [http://jgap.sourceforge.net/]
30. **JGAPm** [http://www.genettasoft.com/p/JGAPm.zip]
31. Shimodaira H: **Approximately unbiased tests of regions using multistep-multiscale bootstrap resampling.** *Annals of Statistics* 2004, **32:**2616-2641.
32. **pvclust** [http://www.is.titech.ac.jp/~shimo/prog/pvclust/]
33. Skurichina M: **Stabilizing Weak Classifiers - Regularization and Combining Techniques in Discriminant Analysis.** *Volume PhD*. Vilnius State University; 2001.
34. Cho SJ, Hermsmeier MA: **Genetic Algorithm Guided Selection: Variable Selection and Subset Selection.** *J Chem Inf Comput Sci* 2002, **42(4):**927 -9936.
35. Tibshirani R: **Regression Shrinkage and Selection via the Lasso.** In *Journal of the Royal Statistical Society Series B (Methodological) Volume 58*. Issue 1 Royal Statistical Society; 1996:267-288.
36. Akaike H: **A new look at the statistical model identification.** *IEEE transactions on automatic control* 1974, **19(6):**716 -7723.
37. Shao J: **An asymptotic theory for linear model selection.** *Statistica Sinica* 1997, **7:**221-264.
38. Wolpert D: **Stacked Generalization.** *Neural Networks,* 1992, **5:**241-259.
39. Kass RE, Wasserman L: **A Reference Bayesian Test for Nested Hypotheses and its Relationship to the Schwarz Criterion.** In *Journal of the American Statistical Association Volume 90*. Issue 431 American Statistical Association; 1995:928-934.
40. Wasserman L: **Bayesian model selection and model averaging.** In *Mathematical Psychology Symposium* Bloomington, Indiana ; 1999.
41. Kuha J: **AIC and BIC - Comparisons of Assumptions and Performance.** *Sociological Methods & Research* 2004, **33(2):**188-229.
42. Hoerl AE, Kennard RW: **Ridge Regression: Biased Estimation for Nonorthogonal Problems.** In *Technometrics Volume 12*. Issue 1 American Statistical Association; 1970:55-67.
43. Goldberg DE: **Genetic Algorithms in Search, Optimization and Machine Learning.** Boston , Addison-Wesley Longman Publishing Co., Inc. ; 1989:372.