ISBN 978-82-575-0997-2
ISSN 1503-1667

Norwegian University of Life Sciences
NO–1432 Ås, Norway
Phone +47 64 96 50 00
www.umb.no, e-mail: postmottak@umb.no

Julia Isaeva

Philosophiae Doctor (PhD) Thesis 2011:34

Norwegian University of Life Sciences • Universitetet for miljø- og biovitenskap
Department of Chemistry, Biotechnology and Food Science
Philosophiae Doctor (PhD) Thesis 2011:34

Multivariate analysis as a tool for understanding and reducing complexity of mathematical models in systems biology

Multivariat analyse som verktøy til forståelse og reduksjon av kompleksitet av matematiske modeller i systembiologi

Julia Isaeva

Multivariate analysis as a tool for understanding and reducing complexity of mathematical models in systems biology

Multivariat analyse som verktøy til forståelse og reduksjon av kompleksitet av matematiske modeller i systembiologi

Philosophiae Doctor (PhD) Thesis

Julia Isaeva

Dept. of Chemistry, Biotechnology and Food Science
Norwegian University of Life Sciences

Ås 2011

Thesis number 2011: 34

ISSN 978-82-575-0997-2

ISBN 1503-1667

Acknowledgments

This work was carried out at the Norwegian University of Life Sciences during the period 2007 - 2011. It was funded by the Norwegian State Educational Loan Fund and the Norwegian University of Life Sciences.

I am heartily grateful to my main supervisor, Assoc. Prof. Solve Sæbø, whose guidance, patience and ability to structure occasionally overwhelming work have been invaluable from the first to the final stage of this project.

I owe my deepest gratitude to Prof. Harald Martens for his infinite reserve of ideas and continual encouragement. His constant excitement and enthusiasm about the work have been contagious and inspirational. A special mention should be made of our long talks during his visit to Germany, which were of immense value and gave extra depth and life to my work.

I would also like to express my warmest thanks to my co-supervisor, Prof. John Wyller, for his moral support, kindness and constructive advice. In addition, I would like to acknowledge all my colleagues who contributed at one stage or another with programming, ideas, feedback and support.

It is my great pleasure to thank Prof. Arcadi Ponossov and Prof. Are Aastveit who made this thesis possible by giving me a chance to get a PhD degree at this university. My big thanks also goes to Prof. Olaf Wolkenhauer for his hospitality and financial support during my stay in his research group at the University of Rostock, in Germany. Stefan Pauleweit is kindly thanked for his help in organising that stay.

I would also like to thank all my international friends who made the years of my PhD truly unforgettable, in particular Anna M., Anna O., Irina and Maryna for their readiness to help any moment. I also wish to deeply thank my lovely flatmates – Mausi, Cindy and Papi – who have become my second family during the last year. I am very happy to have them in my life with their endless support, thoughtful advice and simply long talks around our table after sometimes hard days.

Finally, I would like to express my gratitude to my friends and family in Russia for their love, especially to my mom. I am forever indebted to her for her unceasing belief in me and her continuous support. Without her I would have never been able to reach this point of my life.

Julia Isaeva
Ås, July 2011

List of papers

I. J. Isaeva, S. Sæbø, J.A. Wyller, K.H. Liland, E.M. Faergestad, R. Bro and H. Martens (2010). Using GEMANOVA to explore the pattern generating properties of the Delta-Notch model, *Journal of Chemometrics,* **24** (10), 626-634, doi: 10.1002/cem.1348.

II. J. Isaeva, S. Sæbø, J.A. Wyller, O. Wolkenhauer and H. Martens (2011). Nonlinear modelling of curvature by bi-linear metamodelling, *Chemometrics and Intelligent Laboratory Systems*, doi: 10.1016/j.chemolab.2011.04.010.

III. J. Isaeva, S. Sæbø, J.A. Wyller, S. Nhek and H. Martens (2011). Fast and comprehensive fitting of complex mathematical models to massive amounts of empirical data, *Chemometrics and Intelligent Laboratory Systems*, doi: 10.1016/j.chemolab.2011.04.009.

IV. J. Isaeva, M. Martens, S. Sæbø, J.A. Wyller and H. Martens (2011). The modelome of line curvature: Many nonlinear models approximated by a single bi-linear metamodel with verbal profiling, submitted to *Physica D: Nonlinear Phenomena.*

V. H. Martens, I. Måge, K. Tøndel, J. Isaeva, M. Høy and S. Sæbø (2010). Multilevel binary replacement (MBR) design for computer experiments in high-dimensional nonlinear systems, *Journal of Chemometrics,* **24** (11-12), 748-756, doi: 10.1002/cem.1366.

Summary

In the area of systems biology, technologies develop very fast, which allows us to collect massive amounts of various data. The main interest of scientists is to receive an insight into the obtained data sets and discover their inherent properties. Since the data often are rather complex and intimidating equations may be required for modelling, data analysis can be quite challenging for the majority of bio-scientists who do not master advanced mathematics. In this thesis it is proposed to use multivariate statistical methods as a tool for understanding the properties of complex models used for describing biological systems.

The methods of multivariate analysis employed in this thesis search for latent variables that form a basis of all processes in a system. This often reduces dimensions of the system and makes it easier to get the whole picture of what is going on. Thus, in this work, methods of multivariate analysis were used with a descriptive purpose in Papers I and IV to discover effects of input variables on a response.

Often it is necessary to know a functional form that could have generated the collected data in order to study the behaviour of the system when one or another parameter is tuned. For this purpose, we propose the Direct Look-Up (DLU) approach that is claimed here to be a worthy alternative to the already existing fitting methods due to its high computational speed and ability to avoid many problems such as subjectivity, choice of initial values, local optima and so on (Papers II and III).

Another aspect covered in this thesis is an interpretation of function parameters by the custom human language with the use of multivariate analysis. This would enable mathematicians and bio-scientists to understand each other when describing the same object. It was accomplished here by using the concept of a metamodel and sensory analysis in Paper IV. In Paper I, a similar approach was used even though the main focus of the paper was slightly different. The original aim of the article was to show the advantages of the multi-way GEMANOVA analysis over the traditional ANOVA analysis for certain types of data. However, in addition, the relationship between human profiling of data samples and function parameters was discovered.

In situations when funds for conducting experiments are limited and it is unrealisable to study all possible parameter combinations, it is necessary to have a smart way of choosing a few but most representative conditions for a particular system. In Paper V Multi-level Binary Replacement design (MBR) was developed as such, which can also be used for searching for a relevant parameter range. This new design method was applied here in Papers II and IV for selection of samples for further analyses.

Sammendrag

(Norwegian summary)

Teknologiutviklingen innenfor systembiologien er nå så rask at det gir mulighet til å samle svært store datamengder på kort tid og til relativ lav pris. Hovedinteressen til forskerne er typisk å få innsikt i dataene og deres iboende egenskaper. Siden data kan være ganske komplekse og ofte beskrives ved kompliserte, gjerne ikke-lineære, funksjoner, kan dataanalyse være ganske utfordrende for mange bioforskere som ikke behersker avansert matematikk. I dette arbeidet er det foreslått å bruke multivariat statistisk analyse for å komme nærmere en forståelse av egenskapene av kompliserte modeller som blir brukt for å beskrive biologiske systemer.

De multivariate metodene som er benyttet i denne avhandlingen søker etter latente variabler som utgjør en lineær basis og tilnærming til de komplekse prosessene i et system. Dermed kan man oppnå en forenkling av systemet som er lettere å tolke. I dette arbeidet ble multivariate analysemetoder brukt i denne beskrivende hensikten i Artikler (Papers) I og IV til å oppdage effekter av funksjonsparametre på egenskapene til komplekse matematiske modeller.

Ofte er det nødvendig å finne en matematisk funksjon som kunne ha generert de innsamlede dataene for å studere oppførselen av systemet. Med den hensikt foreslår vi en metode for modelltilpasning ved DLU-metoden (the Direct Look-Up) som her påstås å være et verdifullt alternativ til de eksisterende estimeringsmetodene på grunn av høy fart og evne til å unngå typiske problemer som for eksempel subjektivitet, valg av initialverdier, lokale optima, m.m (Artikler II og III).

Et annet aspekt dekket i denne avhandlingen er bruken av multivariat analyse til å gi tolking av matematiske funksjonsparametre ved hjelp av et dagligdags vokabular. Dette kan gjøre det enklere for matematikere og bioforskere å forstå hverandre når de beskriver det samme objektet. Det var utført her ved å benytte ideen om en metamodell og sensorisk analyse i Artikkel IV. I Artikkel I var en lignende metode også brukt for å få sensoriske beskrivelser av bilder generert fra differensiallikninger. Hovedfokuset i Artikkel I var imidlertid et annet, nemlig å vise fordelen ved multi-way GEMANOVA-analyse fremfor den tradisjonelle ANOVA-analysen for visse datatyper. I denne artikkelen ble GEMANOVA brukt til å avdekke sammenhengen mellom kompliserte kombinasjoner av funksjonsparametrene og bildedeskriptorer.

I situasjoner der ressurser til å utføre eksperimenter er begrenset og det er umulig å prøve ut alle kombinasjoner av parametre, er det behov for metoder som kan bestemme

et fåtall av parameterinnstillinger som er mest mulig representative for et bestemt system. I Artikkel V ble derfor Multi-level Binary Replacement (MBR) design utviklet som en sådan, og den kan også brukes for å søke etter et relevant parameterrom for datasimuleringer. Den nye designmetoden ble anvendt i Artikler II og IV for utvelgelse av parameterverdier for videre analyser.

Contents

Introduction

1 Motivation

In today's science it is not a rare occasion to have large sets of data collected from a conducted experiment or, simply, from an observation of some phenomenon. Physics, biology, chemistry, astronomy etc., all of these sciences nowadays have modern technologies that make it possible, in most cases, to obtain far more data than the human mind is able to handle [1]. For instance, population growth curves in biology [2] (yeast cells in a nutritive solution, fruit-flies in a milk environment, a human population etc. [3]), concentration of a product in kinetic reactions [4], regulatory mechanisms [5], temporal change of light absorbency by the 2-Dimensional Gel Electrophoresis (2DGE) [6] and much more.

To understand the nature of processes and to discover the underlying phenomena, these data have to undergo different types of analyses. This usually gives an experimentalist an overview of effects of different parameters, as well as combinations of them. Yet another aim of every analysis is to be able to foresee a behavioural change of a system when the original conditions are altered. The latter is important in many branches of science in terms of economy: it prevents a scientist from conducting an experiment or introducing a new technology with *a priori* known "bad" outcome.

Analysis of data is usually done by model fitting, that is, by finding a functional relationship between explanatory and response variables. A model *per se* is a simplification of the real world reflecting the main processes by means of the mathematical language. Data modelling has been a research focus for many years [7], and the list of various methods that have been developed is voluminous. However, there can be made a distinction between two main types of modelling: so-called hard and soft modelling. The former is based on an existing theory and binds the data with it, whereas the latter, on the contrary, has no assumptions (or as few as possible) and is data-driven [1]. Hard modelling is sometimes referred to as *bottom-up* and builds mechanistic theories or statistical assumptions into mathematical models. In this connection, mechanistic and statistical types can be distinguished for hard modelling. Both of them are a bit disliked by bio-scientists due to their complicated theory (mathematical formulae and statistical distributions of the error), which is alien to non-mathematicians as well as to non-statisticians.

Hard mechanistic approaches try to model processes in details, thereby, often providing complex dynamical systems containing a large number of parameters. Even though all the model parameters are meaningful and their effects are known, it is quite difficult to interpret the entire system and get a general picture of it.

A statistical approach, in its turn, is focused on handling uncertainty. It helps to understand whether the error of measurements is due to a random noise, or whether there is a structure in it and possibly if some important processes are missed from the scientist's view.

Hard modelling, both mechanistic and statistical, has a strong theory behind it and, therefore, is more traditional and trusted by users unlike soft modelling. The latter consists in finding covariation patterns between variables by analysing tables of data [1] and is sometimes called as a *top-down* approach. Top-down means that an insight into a system is gained by its gradual decomposition into sub-systems. This thesis will focus on soft modelling based on the multivariate analysis. The need for this arises from the fact that in modern science, quite often, one can afford to have more than one observed/measured variable at a time. This may increase chances of analysts to get a better picture of processes in a system. Multivariate analysis is more and more used for data reduction and simplification of data structures by means of finding latent variables that describe the underlying processes. These latent variables are usually fewer than the original variables in the model and may provide a simpler overview of a system.

However, pure mathematicians and statisticians may argue that soft modelling is lacking theoretical aspects, and, therefore, require some other methods for model assessment. It is, indeed, based on a simple linear algebra (matrix algebra) and does not involve any statistical assumptions about error distribution. Nevertheless, it does include verifying of the results (whether the found patterns are valid or it is an apparent error) by simple statistical techniques (e.g., cross-validation). Besides, it facilitates simple graphical interpretation, which can be used for classification of samples (grouping) and prediction of new observations. The only danger with such an analysis is over-fitting of data. One might get overwhelmed by the results and might unintentionally impose this model on the noise [8]. This can happen when a too detailed model is considered, and can lead to poor prediction. However, when being cautious, multivariate analysis is very useful and is preferred by bio-scientists due to its simple mathematical background and for being more comprehensible.

The aim of this thesis is to show that, with the help of multivariate analysis, mathematical models from systems biology can be understood by a wide audience despite complexity of the models and a number of parameters. It has been shown that results from multivariate analyses can be used multidisciplinary and reduce the gap between communities of bio-scientists and data analysts, both mathematicians and statisticians.

2 Background

Our world is multivariate, and there are no processes that depend only on one unique variable [9]. There are always some correlations present between observed properties, and certain values of one variable are linked with those of another, or a set of them. Therefore, it is important not to lose any significant information when trying to analyse such data. For this purpose, multivariate analysis is broadly used. Multivariate analysis is the analysis of data obtained from simultaneous measurements on many variables [10]. For instance, students' exam marks for different subjects, a set of body measurements of patients, collection of climatic conditions etc. In the example with the exam marks, it would be of interest to know how a certain result on one exam will affect such on another; whether there is any relation of marks to the order of the exams etc.

There exists a long list of multivariate methods for data analysis, and there is no recipe for which method is the best and gives the most appropriate results in a given case. It is mostly an analyst's preference that decides the choice of method. However, data organisation has to be thought through thoroughly, and potential problems of handling certain data with one or another method should be exposed.

Main objectives of multivariate methods can be divided into three groups [9]:

- data description
- data discrimination/classification
- prediction.

Data description methods are explorative and aim at finding main patterns in data tables and positive or negative correlations between diverse variables. It helps to "look inside" the data and discover the effect of each variable on the response. The main example of explorative methods is the Principal Component Analysis (PCA).

Data discrimination is intended for revealing grouping of samples and variables with similar properties. For example, in Paper IV we saw a clear separation of curves into sigmoids and archoids. Unlike discrimination, **classification of data** is "supervised" and data clusters are known *a priori*. In this case it is possible to determine which group a new data point belongs to. As traditional methods for data classification SIMCA (Soft Independent Modelling of Class Analogy) [9] and DPLSR

(Discriminant Partial Least Squares Regression) [8] can be considered, whereas PCA can be of great help for data discrimination.

Prediction is an essential element in data analysis. In the world of expensive equipment and reagents, it is important to build an experiment in an accurate way and not waste money. That is why, it is useful to know how a system will behave when one or several conditions are changed. This is possible by building a reasonable model of the existing data and a further prediction of a probable result under new circumstances. A common method for doing that is the Partial Least Squares Regression (PLSR) [11].

Even though the multivariate methods addressed in this thesis have different goals at the end, they have similar principles. All of them are based on matrix algebra and are aimed at finding new, *latent* variables as ”cores” of system processes. *Latent* means that one cannot measure/observe these variables directly [8]. They are obtained as linear combinations of the manifest (observed) variables and reflect the underlying structure of the data. By means of using simple matrix manipulations, multivariate analysis may be understood by far more people than both hard mathematical and statistical modelling and, therefore, attracts a larger number of bio-scientists.

For a successful performance of multivariate analysis, it is preferable to have both much information about essential properties (many variables) and a large number of objects. The former is important for discovering true interdependencies between variables and, optionally, their relevance to some response variable(s), whereas the latter is important for reducing estimation error and for model validation. Unfortunately, in real data experiments a shortage of one or the other is common, which makes it more difficult to find an appropriate model for a given data set.

At the beginning of either type of multivariate analysis, all data obtained should be organised in a matrix or in a cube. In Papers II-V of this thesis we have dealt with two-way data, i.e., with data matrices, whereas Paper I was focused on a multi-way analysis that is described separately below.

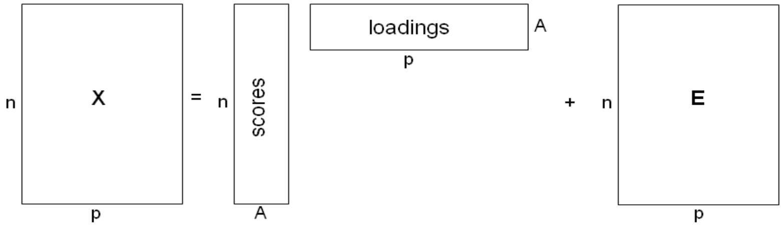
As the number of multivariate methods is very large, only descriptions of those that were used in the enclosed papers are given here. All these methods were implemented here with the purpose of complexity reduction of the data sets and prediction of new data.

Principal Component Analysis (PCA)

PCA concerns the analysis of a single data matrix (two-way table), and, as was mentioned above, is usually used for descriptive purposes and data exploration. It implies finding new, latent variables that describe most of the data variation. These new variables are called *Principal Components* (*PCs*) and lie in directions of the maximal variation of the data. They describe the data variation in a descending order: first PC is found along the direction of largest variation; second PC - in the direction of second largest variation but orthogonal to the first, and so on. When the remaining variation is small enough, it is considered that the optimal number (A) of PCs is found, and the information that is left is regarded as noise. Often, A is much smaller than the number p of original variables. PCs constitute a new, orthogonal basis for the variable space and are obtained as linear combinations of the original variables. Coefficients of the latter in the space formed by the new variables are called *loadings*, namely, a set of loadings is a transformation matrix from old variables to the new ones. The projection of the observed variables onto the new basis yields *scores* – the observed values along the PCs. Together with loadings, the vectors of scores comprise an explained part of the data, the structure, whereas the remaining variability is regarded as noise [9]. This can be expressed as:

$$\mathbf{X} = \mathbf{TP}' + \mathbf{E} = Structure + Noise \tag{1}$$

or graphically



where $\mathbf{X}$ is the observed data matrix, $\mathbf{T}$ and $\mathbf{P}$ represent scores and loadings respectively, $\mathbf{E}$ is a residual matrix, n is the number of samples, p is the number of variables and A is the number of PCs.

From Eq. (1) one can notice that matrix $\mathbf{X}$ is represented linearly with respect to both matrices $\mathbf{T}$ and $\mathbf{P}$. That is why PCA is referred to as a bi-linear method.

Plots from PCA analysis are of great value for data exploration in the way it may reveal patterns of covariation between variables or between samples. For instance, a score plot gives an idea of which samples are similar and which are different. In this context, PCA can also be used as a pre-step to other methods like clustering, classification or regression. Loading plots, in its turn, show an analyst what variables are related to each other and in which way – positively or negatively. If variables are measured on different scales, it is more recommended to study a correlation loading plot, which is scale invariant due to transformation of the original loadings into correlation coefficients between the input and latent variables [12]. Both score and loading plots were widely used in Paper IV to get a first overview of the sensory data on curves. These two plots can be combined into one (bi-plot) and give information about the influence of certain variables on different samples.

In this thesis, PCA is extensively applied both as a descriptive tool in Papers I and IV, as a data compression method and as a basic principle for developing a new curve fitting method in Papers II and III, and for building the concept of a *metamodel* (Papers II and IV), which is described below.

Partial Least Squares Regression (PLSR)

PLSR, unlike PCA, concerns the analysis of two data matrices, a predictor variables matrix ($\mathbf{X}$) and a response matrix ($\mathbf{Y}$) that may have one or several variables [11]. In case of multiple responses, instead of modelling one response at a time (PLS1), all variables can be taken into account simultaneously (PLS2), which provides the information about their interdependence.

The main principle of this method is the same as for PCA and consists in finding latent variables that describe the essential structure of the data. However, in case of PLSR, a matrix of responses is also taken into consideration so that the covariance between $\mathbf{X}$ and $\mathbf{Y}$ matrices is maximised. As a matter of fact, PCA may be regarded as a special case of the PLSR analysis with no $\mathbf{Y}$-variables [8].

The process of building a PLSR model is iterative, that is, components are extracted one by one through deflation of both data matrices. As a result of this, sets of *loading weight* vectors and corresponding to them score vectors are attained for $\mathbf{X}$ and $\mathbf{Y}$ matrices. Loading weights for $\mathbf{X}$ matrix represent components of a PLSR model and are constructed in such a way that they span the direction of the maximal covariance between matrices of predictor and response variables. Despite the difference in construction, PLSR components are also called PCs by analogy with PCA.

PLSR is used both for pattern revelation and prediction of new data. For the first purpose, score and loading plots are used much in the same way as for PCA. The only difference is that the loading plots also contain information about $\mathbf{Y}$-variables ($\mathbf{Y}$-loadings). In this way, explanatory variables are related to the responses, and it is easily seen whether there is any effect of one or another variable on a certain outcome. When a proper model of the $\mathbf{Y} \sim \mathbf{X}$ relationship is found, it is quite often used for prediction of new responses from a new set of input $\mathbf{X}$-variables.

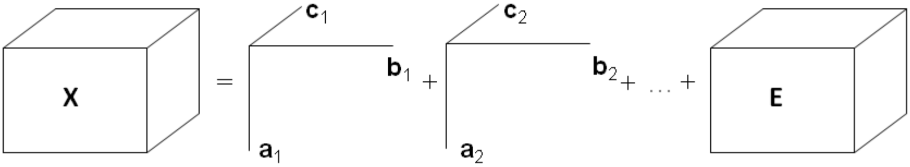
For a model, to be suitable for prediction, it has to be realistic, that is, $\mathbf{X}$ and $\mathbf{Y}$ matrices have to be collections of *essential* properties and responses, and a new data set should be obtained under the same circumstances. If the original $\mathbf{X}$ and $\mathbf{Y}$ are representative and span X- and Y-spaces rather extensively, every new observation should be predicted easily. In order to check whether a found model is reasonable (neither too complex nor too simple and does not give any strange results), different techniques have been developed and are applied for a model validation. There exist two main types of such methods: cross-validation [13] and test-set validation. The first one is internal and considers parts of a data set to be unknown. The latter is external and applied when another (independent) data set is available. In both cases, ”unknown” values are predicted and compared with the true ones by means of the Root Mean Square Error (RMSE) [8] or the coefficient of determination for prediction (R^2_{pred}) [14]. Validation of a model is an absolutely necessary procedure since having a poor model may lead to wasting a large amount of time and money.

In this thesis, PLSR was used in Paper IV in order to examine a relationship between the metascores and sensory evaluation. Later on, the established model was applied for prediction of function parameters from the estimated sensory values.

Multi-way analysis

All the methods described so far deal with data organised in two-way tables (matrices). However, quite often it would be more appropriate to structure them as a cube (three-way) or a hypercube (N-way) [15]. Such data organisation is used, e.g., in food science when a certain property of a product is observed under several levels of various factors (temperature, light, moisture etc.) as, for example, in [16]. Each dimension of a cube is called a *mode,* and all the data for one of the mode levels is called a *slice* or a *slab*.

The two-way methods mentioned above (PCA, PLSR) can also be applied to N-way data with the requirement for the latter to be unfolded in advance [17]. Unfolding means reshaping of a (hyper)cube into a matrix, and it is indeed tempting to do so since two-way multivariate methods are well-known and easy to interpret. Nevertheless, there is no agreement in what way data should be unfolded (along which dimension). Moreover, such a re-organisation leads to an information loss about correlation between slabs. For this purpose, multi-way analysis was developed with PARAFAC (PARAllel FACtor analysis) as one of the main N-way methods [15, 18]. PARAFAC is referred to as a tri-linear generalisation of the bi-linear PCA due to the similar principle: it projects data down onto several latent variables, thereby reducing dimensionality of the data. In a three-way case, a PARAFAC model has a set of three loading vectors for each component; and, if one of the cube modes represents samples, then the corresponding loadings are called scores by analogy with the PCA. A three-way PARAFAC model can be illustrated by the following figure:



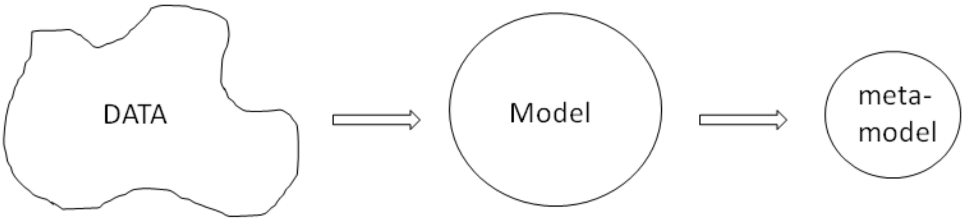
Due to the ability to analyse data in the original structure without unfolding, PARAFAC is often called the three-way advantage, although it can also be applied to an N-way data (that is, a hypercube of N dimensions).

The number of factors in each component of a PARAFAC model is equal to the number of the data modes. This can be changed if its alternative - GEneralised Multiplicative ANalysis Of VAriance (GEMANOVA) - is used [15, 16, 18]. GEMANOVA can "eliminate" individual factors from each component by setting all levels of the corresponding loading vectors to be equal to one. In this case, first component may, e.g., contain two modes, second - all N modes, third - only one and so on. If each component includes all the modes, then GEMANOVA is identical to PARAFAC. To assess the goodness of fit, a model-based bootstrap, which is described in [19, 20], can be applied.

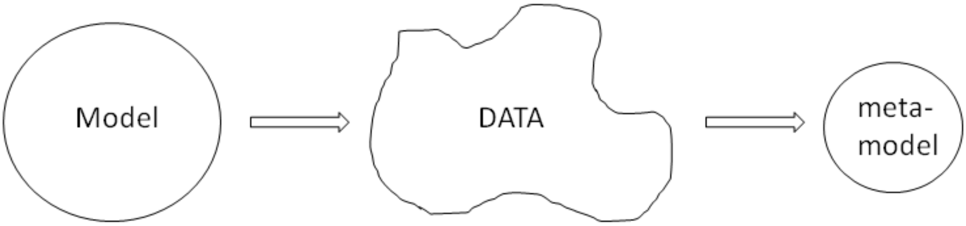
GEMANOVA is focused on finding higher-order interactions, which are inherent to the majority of the real world processes, and, therefore, is suitable for analysing a tangled structure of complex systems. Thus, for example, in Paper I GEMANOVA was applied for studying a mathematical model of a dynamical system – the Delta-Notch model, which has five parameters for each of 2 500 cells. Some complex interactions between parameters of this model were found by using the named method.

2.1 Metamodel

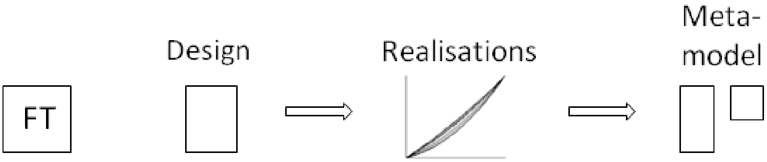
As was mentioned above, a model is an abstraction of the real world. However, models themselves can be simplified by means of metamodelling. "Meta", from Greek, stands for "after" and in the data analysis means modelling of a model (approximation of a model) [21]. Generally, it leads to a significant reduction of complexity and dimensionality of the data:



This illustration is valid for real data when modelling comes after observations (*a posteriori*). In case of having simulated data, a model is known *a priori* (e.g., functional models in Papers II-IV), and a metamodel is then built on the basis of the simulations:



or more particular for this thesis:



Here FT (function type) represents some mathematical model, realisations attained by a given design compose data, and metamodels are formed by sets of scores and loadings from PCA on the simulated data.

The aim of metamodelling is to obtain simpler (than original) models in terms of structure but with a minimal loss of information. In this thesis, metamodels were used in Papers II and III as a basis for a new curve fitting method described below, and in Paper IV – for a compact representation of the phenomenon of curvature and mapping a human profiling of curvature into the mathematical language.

3 Paper summaries

Paper I – *Using GEMANOVA to explore the pattern generating properties of the Delta-Notch model*

The aim of this paper was to explore a complex nonlinear mathematical model of dynamics – the Delta-Notch model – by means of multi-way analysis (GEMANOVA). Delta and Notch are two signalling proteins in a cell, responsible for its colour, and they influence the level of each other both in one cell and in adjacent cells [22]. The data were represented as 2D hexagonal lattices of 2 500 cells, each dependent on five state parameters. The lattices were generated from a quarter fractional factorial design with two levels for each of the model parameters and contained cells of different shades of grey. In total, 26 cell grids were used and evaluated by a sensory panel with twelve descriptors portraying the patterns. Obtained values were organised in a five-way array and analysed with GEMANOVA. It was shown that the latter is more suitable for analysis of such data than standard statistical methods (particularly, ANOVA [23]) due to its ability to capture an N-way structure and find higher-order interactions without overloading a model with too many parameters. The GEMANOVA analysis revealed significant interactions between the system parameters, and the results were validated by non-parametric bootstrapping. Further, new data were generated by computer simulations in order to check the veracity of the established GEMANOVA models. It was noted that the majority of the parameter effects found by GEMANOVA were correct. Besides, the data simulations revealed the presence of the bifurcation point, which was confirmed by numerical approximations from [22].

Paper II – *Nonlinear modelling of curvature by bi-linear metamodelling*

In Paper II we have developed a new method for fitting nonlinear models to data. Existing methods for estimation of nonlinear functions usually require assumptions about functional form and parameters. Moreover, they are typically iterative and it is necessary to choose a set of initial values, which can be extremely difficult without any prior knowledge about the data. Most of the methods, in addition, have a local optima problem: if a choice of starting values is made without a proper attendance, the final solution may be false due to the convergence of the search criterion to a local, instead of the global, minimum (maximum). The new method (Direct Look-Up, DLU), proposed in this paper, is based on a *modelome* – collection of realisations of 38 simple mathematical functions from different application fields. The set of simulations for each function is further approximated by a bi-linear metamodel (PCA model), that is, by a set of score and loading vectors. When having a new curve, it is simply projected onto each metamodel, and a list of most plausible functions along with the parameter estimates is obtained. In that way, the DLU method avoids problems with local optima and does not require any prior assumptions, including initial values. The method was demonstrated on a computer simulated noise-free curve of the Hill function type with random parameter values. The true (Hill) function was one of the suggested by the method models for the given curve, and parameter estimates were rather accurate.

Paper III – *Fast and comprehensive fitting of complex mathematical models to massive amounts of empirical data*

This article is subsequent to Paper II and extends the method's technique to the level where it is able to handle noisy data. Firstly, the DLU method was compared to the traditional method for curve fitting – Iterative Least Squares (ILS) [24] on an example of a set of artificial curves, but this time with homoscedastic noise and missing data points. Parameter estimates for both of the methods were almost identical, although estimation errors in case of ILS were much larger due to the fact that ILS did not converge in 27% of the cases. This points at the obvious advantage of the DLU over it. Moreover (and most importantly), performance time of curve fitting with the DLU approach was reduced by factor 24 in comparison to ILS, which is extremely relevant when having large data sets. At last, the DLU method was tried on a real, highly noisy data set containing 174 216 curves (time series) over 200 time points. The estimated function type agreed with the initial guess of the experimentalist, and the reconstructed from the estimated values data looked very similar to the original one. The only problem encountered was the difficulty in handling more than 90 000 curves simultaneously, but this was a computer capacity problem only.

Paper IV – *The modelome of line curvature: Many nonlinear models approximated by a single bi-linear metamodel with verbal profiling*

The focus of Paper IV was on the further exploration of the concept of a metamodel, nonlinear phenomenon of curvature and making the latter more accessible to a general audience. In contrast to Papers II and III, the metamodel built here was global, namely, it was constructed for all the models in the modelome jointly. Only 12 PCs were needed to describe the whole collection of curves by means of PCA, which indicates a significant reduction of dimensionality. Further, using the MBR design from Paper V on the metascores, 32 extensively spanning the curvature space curves were chosen and evaluated by a sensory panel with 14 descriptors. The sensory evaluation was repeated four months later including curves of four new function types with the purpose of verifying whether the established metamodel captures the entire curvature phenomenon. PCA and PLSR analyses on the sensory values and metascores have shown high efficiency of the evaluation and have found a nonlinear model of their relationship to each other, which allowed us to give meaning to meta-PCs by the words-descriptors and to predict sensory values for the rest of the curves in the modelome. The latter led to the opportunity of mapping function parameters into the custom language, defined by the descriptors, i.e., interpretation of pure mathematical parameters by words used in the everyday life.

Paper V – *Multi-level binary replacement (MBR) design for computer experiments in high-dimensional nonlinear systems*

This paper describes a new method for design of experiments for several factors, with more than two levels for each of them. The most traditional way to do it is a factorial design [23]. However, if multi-level multi-factor design is to be performed, the total cost of the experiments can be very high. Therefore, it is important to reduce the size of the design in such a way that the chosen factor levels span the parameter space quite extensively. If only two levels for each factor are available, then fractional factorial design can be used. For other situations, it is proposed here to employ the MBR design method, which consists in recoding each multi-level factor into a set of binary variables yielding a design with only two-level factors. A traditional fractional factorial design is imposed to give a requested number of design points or resolution. The design points are then recoded back to the original multi-level factors in order to run the experiments. The efficiency of such a procedure was shown on an example of computer simulations for a growth curve. Moreover, it was demonstrated that, by means of the MBR design method, it is possible to search for a relevant range of parameter values, which is extremely important when experiments are very costly.

4 Discussion

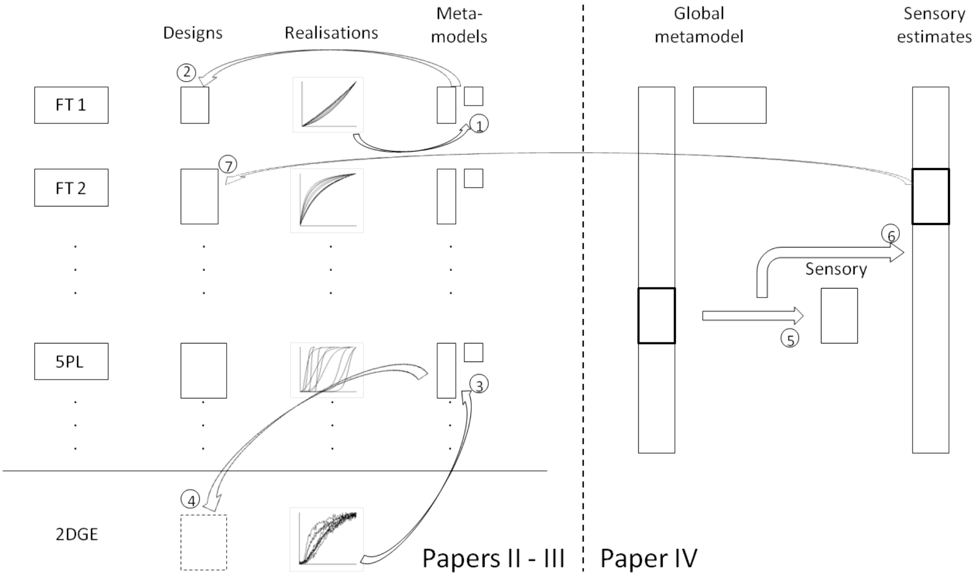
4.1 Contribution

The aim of this thesis was to demonstrate effectiveness and ease of usage of multivariate analysis methods when studying complex mathematical models from systems biology. The latter are often so complicated that it is only in mathematicians' power to deal with them. In connection with this, the gap between math-oriented scientists and bio-scientists becomes larger and larger as complexity of systems increases. As a step towards reducing this gap, sensory evaluations of the outcomes of tangled processes were run and analysed by means of multivariate methods in Papers I and IV. It was shown there that frightful mathematical functions and their parameters can be easily interpreted by custom words used by ”normal” people. It means that, whenever a biologist and a mathematician have a conversation, they can describe the same object in the way they are most comfortable with, and they will still understand each other. All what they need for this, is an appropriate multivariate model mapping their two languages one into another, be it a two-way or an N-way model.

However, mathematics is not the only field that scare many people away: statistics with its endless number of distributions, hypotheses testing and error evaluation is also alien to the majority of bio-scientists. When it comes to modelling their data, which are very often in a large amount, it is difficult to make any assumptions about the error distribution and initial values, especially if an experiment is conducted for the first time and there is no prior knowledge about the data. Even though the era of modelling is not new at all and a long list of methods has been developed during many years, these difficulties are still faced by analysts along with many more problems encountered such as local optima, handling of noisy data, subjectivity in the choice of methods and models, long performance time and so on. Therefore, there was a strong need for a novel method that would solve at least some of the named problems. The DLU approach was proposed as such in Papers II and III, which consists in a simple projection of a new data set onto a bi-linear metamodel of the realisations of simple mathematical functions. Here no assumptions have to be made, i.e., a chance to get stuck in a local optimum is rather low. The computational time is extremely short in comparison to the traditional fitting methods, which is of great importance in modern technology that allows a researcher to have massive amounts of data.

Multivariate analyses have been applied here for a better understanding of complex models that would undergo either mathematical or statistical modelling. Both of them are tied to a strong theory that bio-scientists are not familiar with. In contrary, multivariate methods are based on the elementary mathematics verifying the results by simple statistical procedures. This gives a much easier and quicker overview of a system than most of the advanced methods. Multivariate analyses are accompanied by rather interpretable graphics that give an analyst a clear insight into the system processes. Thus, for example, in Paper I, effects of the parameters of a highly nonlinear mathematical model were easily seen from informative GEMANOVA plots.

By means of the bi-linear PCA, the concept of a metamodel was developed further and defined for curvature in Papers II - IV. A general picture of these three papers can be depicted in the following way with "FT" as a notation for a function type:



In Papers II and III a metamodel for each of 38 function types from the modelome was established separately. In this case, as was mentioned above, all metamodels are represented by a set of scores and loadings. Then, for either simulated curve, parameter values can be estimated by its projection onto the metamodel for each function (arrows 1 and 2). When studying real data from the 2DGE experiment, five-parameter logistic (5PL) function turned out to be the one with the best fit, and through a projection of the data set of curves onto the metamodel of the named function, parameter estimates for each curve were obtained in a short period of time (arrows 3 and 4). Furthermore, a global (joint) metamodel for all the curves in the modelome together was built in Paper IV. Employing the MBR design, a set of 64 curves was chosen as a representative collection from the modelome, and these were evaluated by a sensory panel. By establishing a reasonable model mapping metascores into the sensory values for the selected samples (arrow 5), it became possible to predict sensory evaluation for all the curves in the modelome (arrow 6). In this thesis it was implemented only for two models – the logistic function and the error function. At last, a model imitating a relationship between parameter values of the named functions and sensory estimates of their realisations was constructed (arrow 7). It means that, given a function type, parameter values for any curve can be predicted with a certain precision by its human profiling.

Certainly, to capture the entire curvature phenomenon, the parameter space for each function had to be spanned quite densely and extensively. In those cases when a function depends on two or three parameters, it is of no problem to sample parameter space quite densely, but an increase of a number of parameters can lead to a combinatorial explosion. It is not a big issue if one has to deal just with computer simulations (computer capacity can easily be extended), but, when it comes to real world data, it may be very costly, and even unrealisable, to conduct experiments for all possible situations. Moreover, the relevant parameter range is often unknown, which makes it even more difficult. That is why, it is important to plan experiments beforehand by locating the interval of relevance for each parameter and choosing such combinations of parameter values that represent the whole parameter space as widely as possible. For this purpose, the MBR design method was developed in Paper V and employed in Paper II for simulations of one of the functions and in Paper IV for choosing the curves for the sensory evaluation so that they fill up the entire room of curves up to a considerable extent.

4.2 Future perspectives

There is, of course, still much that can be done for bringing closer various societies of scientists. Hard modelling should be more comprehensibly taught to bio-scientists; soft modelling should be proved to mathematicians and statisticians to be an efficient tool for analysing data etc. Development of the dictionary between absolutely different scientific languages is an area of great interest, and a first step towards this has been done in this thesis. The next step could be an improvement of accuracy of such translation and including more and more complex ”words” – mathematical functions – into the dictionary. Currently, our modelome consists of only simple functions that are smooth and monotonous and have not more than one inflection point. However, real processes are rarely described by such elementary models, and therefore, it is necessary to develop the modelome further to the level of sums and products of several functions.

The DLU approach proposed in this thesis grants an analyst with a list of plausible functions for his/her data. To be able to choose the most suitable of them, one should know to what extent their properties differ. For this purpose the metamodel can be of great use. The parameter spaces of two functions can be mapped into each other along with sensory evaluation of the corresponding curves. Then it should be straight forward to discover diverging properties between the functions.

Multi-way analysis is not as well known as two-way methods and is not widely used, however, it has a great potential. Since data often has a (hyper)cube structure as a result of the experimental design, it is important to learn more how to analyse such data without losing relevant structure information.

As was mentioned above, a model is just a simplification of the real world, and we do not claim that it can describe absolutely all properties of a biological system. Nevertheless, a reasonable model can mimic the underlying phenomena present in the data and narrow the region of study. With constant improvement and extension of methods, it is easy to get lost in the world of multivariate analysis. However, knowing in detail just a few number of methods and applying them with good care may provide one with an appropriate model that captures the essential information about the observed data and help to foresee the results of further experiments.

References

[1] H. Martens and A. Kohler. Mathematics and measurements for high-throughput quantitative biology. *Biological Theory*, 4(1):29–43, 2009.

[2] J. Warringer, D. Anevski, B. Liu, and A. Blomberg. Chemogenetic fingerprinting by analysis of cellular growth dynamics. *BMC Chemical Biology*, 8:3–12, 2008.

[3] R. Pearl. *The Biology of Population Growth.* Ayer Co Pub, 1977.

[4] J.I. Steinfeld, J.S. Francisco, and W.L. Hase. *Chemical Kinetics and Dynamics.* Prentice Hall Englewood Cliffs (New Jersey), 1989.

[5] F. Jacob and J. Monod. Genetic regulatory mechanisms in the synthesis of proteins+. *Journal of Molecular Biology*, 3(3):318–356, 1961.

[6] H. Grove, E.M. Faergestad, K. Hollung, and H. Martens. Improved dynamic range of protein quantification in silver-stained gels by modelling gel images over time. *ELECTROPHORESIS*, 30:1856–1862, 2009.

[7] J.F. Rusling, T.F. Kumosinski, and ScienceDirect (Online service). *Nonlinear Computer Modeling of Chemical and Biochemical Data.* Academic Press, 1996.

[8] H. Martens and M. Martens. *Multivariate Analysis of Quality: An Introduction.* John Wiley & Sons Inc, 2001.

[9] K.H. Esbensen, D. Guyot, F. Westad, and L.P. Houmøller. *Multivariate Data Analysis - In Practice: An Introduction to Multivariate Data Analysis and Experimental Design.* Multivariate Data Analysis, 2002.

[10] S. Kotz, N.L. Johnson, and C.B. Read. *Encyclopedia of Statistical Sciences*, volume 6. Wiley, 1985.

[11] H. Martens and T. Naes. *Multivariate Calibration.* John Wiley & Sons Inc, 1989.

[12] H. Martens and M. Martens. Modified Jack-knife estimation of parameter uncertainty in bilinear modelling by partial least squares regression (PLSR). *Food quality and preference*, 11(1-2):5–16, 2000.

[13] M. Stone. Cross-validatory choice and assesment of statistical predictions. *Journal of the Royal Statistical Society, Series B—Methodological*, 36:111–147, 1974.

[14] D.C. Montgomery, E.A. Peck, and G.G. Vining. *Introduction to Linear Regression Analysis*. Wiley Interscience Publication, 2001.

[15] R. Bro. *Multi-way Analysis in the Food Industry. Models, Algorithms and Applications*. PhD thesis, Royal Veterinary and Agricultural University, 1998.

[16] R. Bro and M. Jakobsen. Exploring complex interactions in designed data using GEMANOVA. Color changes in fresh beef during storage. *J. Chemom.*, 16(6):294–304, 2002.

[17] E.M. Faergestad, S. Sæbø, Ø. Langsrud, M. Høy, A. Kohler, K.H. Liland, K. Hollung, J. Almergren, E. Anderssen, and H. Martens. Analysis of megavariate data in functional genomics. *Comprehensive Chemometrics (Walczak B, Tauler Ferré R, Brown S, eds)*, 4:221–278, 2009.

[18] A. Smilde, R. Bro, and P. Geladi. *Multi-way Analysis with Applications in the Chemical Sciences*. John Wiley & Sons, Ltd., 2004.

[19] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC: New York, USA, 1998.

[20] K.H. Liland and E.M. Faergestad. Testing effects of experimental design factors using multi-way analysis. *Chemometrics and Intelligent Laboratory Systems*, 96(2):172–181, 2009.

[21] J.P.C. Kleijnen. *Design and Analysis of Simulation Experiments*. Springer Verlag, 2007.

[22] J.R. Collier, N.A.M. Monk, P.K. Maini, and J.H. Lewis. Pattern formation by lateral inhibition with feedback: a mathematical model of Delta-Notch intercellular signalling. *Journal of Theoretical Biology*, 183(4):429–446, 1996.

[23] D.C. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons Inc, 2008.

[24] P.J. Bickel and K.A. Doksum. *Mathematical Statistics: Basic Ideas and Selected Topics*. Prentice-Hall, Inc., 1977.

Paper I

Received: 6 July 2009, Revised: 4 June 2010, Accepted: 20 July 2010, Published online in Wiley Online Library: 11 October 2010

(wileyonlinelibrary.com) DOI: 10.1002/cem.1348

Using GEMANOVA to explore the pattern generating properties of the Delta-Notch model[†]

Julia Isaeva[a]*, Solve Sæbø[a], John Andreas Wyller[b], Kristian Hovde Liland[a], Ellen Mosleth Faergestad[c], Rasmus Bro[d] and Harald Martens[e]

In the area of systems biology, increasingly complex models are developed to approximate biological processes. The complexity makes it difficult to derive the properties of such models analytically. An alternative to analytical considerations is to use multivariate statistical methods to reveal essential properties of the models. In this paper it is shown how the properties of a relatively complex mathematical model for describing cell-pattern development, the Delta-Notch model, can be explored by means of statistical analyses of data generated from the model. ANOVA is a well-known and one of the most commonly used methods for analyzing data from designed experiments, but it turns out that it is not always appropriate for finding and exploring higher-order interactions. For this purpose a multiplicative alternative—GEMANOVA—was used in the present paper for studying the Delta-Notch model, for which the properties depend on higher order interactions between the model parameters. It is shown here how a forward selection strategy combined with bootstrapping can be used to identify GEMANOVA models with reasonable fit to the data, and it is demonstrated how new insight about the Delta-Notch model can be gained from interpreting the GEMANOVA output. Copyright © 2010 John Wiley & Sons, Ltd.

Keywords: GEMANOVA; dynamical systems model; multivariate analysis; sensory data

1. INTRODUCTION

In the area of systems biology, there is an increasing focus on developing mathematical models that to some extent describe biological processes (see for example [1–4]). This modelling approach reflects a so-called reductionist view of science, namely, that the road towards understanding a biological system goes through a causal understanding of the elements of the process. Some, therefore, refer to this as the "bottom-up" way of doing science. The opposite approach is the "top-down" method characterized by studying the global patterns of a system, typically through observational studies. Through an iterative process involving hypothesis formulation, observation and testing, the aim is to obtain a causal understanding of the process, slowly working towards the elements of the process. Hence, the aim of both approaches is the same, but they attack the problem from opposite directions. The former is the typical mathematical approach, whereas the latter is the statistical counterpart.

Historically, there has been a big "gap" between the steps of the process, at which these two approaches give us insight. The detailed mathematical models tend to grow into intractably large systems if submodels are put together in an attempt to build more global systems. The number of parameters soon becomes so large that it is impossible to obtain a purely mathematical understanding of the properties of the model. On the other hand, the statistical approaches are typically based on assumptions like linearity and normality, which may be justifiable in order to study a biological process at a global scale. Furthermore, statistics is about finding associations, but unless carefully planned experiments can be performed, the causality question is much more difficult to answer. In summary we may say that the strength of the "bottom-up" approaches is the ability to study causality properties for sub-elements of complex biological systems, whereas the "top-down" approaches are more suited for studying the global

* Correspondence to: Julia Isaeva, Norwegian University of Life Sciences, Department Chemistry, Biotechnology and Food Science.
E-mail: julia.isaeva@umb.no

a Julia Isaeva, Solve Sæbø, Kristian Hovde Liland
Department of Chemistry, Biotechnology and Food Science, Norwegian University of Life Sciences, P.B. 5003, N-1432 Aas, Norway

b John Andreas Wyller
Department of Mathematical Sciences and Technology, Norwegian University of Life Sciences, P.B. 5003, N-1432 Aas, Norway

c Ellen Mosleth Faergestad
Nofima Mat AS, Norwegian Institute of Food, Fisheries and Aquaculture Research, N-1430 Aas, Norway

d Rasmus Bro
University of Copenhagen, Department of Food Science, Rolighedsvej 30, DK-1958 Frederiksberg C, Denmark

e Harald Martens
Centre of Integrative Genetics/IMT, Norwegian University of Life Sciences, N-1432 Aas, Norway

[†] This paper is submitted for the Special Issue "Proceedings of the 11th Scandinavian Symposium on Chemometrics, SSC11".

properties of the systems. In order to fill the gap between the global and the detailed understanding of a system, it may, therefore, be beneficial to combine statistical and mathematical methods. The approach described in this paper is just one step towards fulfilling this goal.

Another example of this is the paper by Veflingstad [5] where a non-linear model for the dynamics of two pattern generating proteins, Delta and Notch in a discrete cell network, was studied [6]. Even though this model only contained two state variables and five parameters, it turned out difficult to relate the steady states to the parameter settings and the choices of initial conditions. Only a few studies are published on the mathematical exploration of the pattern generating properties of this model (see [6,7]), and these studies are for rather limited cell networks due to the complexity of the model. The way Veflingstad dealt with this was to consider the model as a data generating system. An experimental design was put up in order to explore the impact of different parameter settings on the resulting steady states. Furthermore, multivariate statistical method (Partial Least Squares Regression, PLSR) was used to relate steady state categories to the parameter settings [8]. This combination of dynamic modelling and statistics was a completely new approach towards increased understanding of the Delta-Notch model.

Many biological processes are described by very complicated mathematical models usually containing large number of parameters to be estimated [1–4]. It may be desirable to simplify the model somehow by reducing the number of parameters, e.g., by neglecting some factors in the biological model. It is important to do so without loosing essential model properties. Thus, it should be ensured that the disregarded parameters have only a slight influence on the model. For recognizing whether a factor is significant or not, statistical methods may be used. However, the statistical toolbox also contains a wide range of other methods suitable for the interactions between different factors and the importance of the each term in a model.

ANalysis Of VAriance (ANOVA) is a frequently used method for analyzing data from designed experiments and may be effective for screening main and interaction effects of various factors to some experimental output. A typical search procedure for a good model is based on a forward selection scheme starting with the inclusion of significant main effects, then second-order interactions, and so on. Alternatively, a backward elimination procedure may be adopted leaving out non-significant higher-order interactions first [9]. The aim is usually a simple model, mostly with main effects and as few interactions as possible, and usually the highest order interactions are regarded as part of the noise. However, in reality it may occur that the behavior of a system is defined by complex interactions and not only by main effects. A large number of main effects and interaction effects included in the model will typically lead to increased estimation errors for the effects and few degrees of freedom left for the error sum of squares. So unless the number of replicates is sufficiently large, the ANOVA method may, therefore, fail to discover higher order interactions that are truly present.

The potentially large number of parameters in ANOVA models is partly a result of its hierarchical structure and the assumption of additivity of effects. That is, in a customary way of fitting a model, if a high order interaction is included to a model, e.g., $A * B * C$, then all lower-order terms containing A, B and C exclusively, should also be in the model even if they are not statistically significant (this is also referred to as the principle of functional marginality (e.g [10]). It is of course possible to omit lower order terms from the model (unrestricted selection) which will lead to a lower number of parameters to be estimated. However, using the principle of functional marginality is usually advised, otherwise the model might be forced to go through certain points [10].

An alternative to the additive ANOVA approach is to arrange the data in an N-dimensional hypercube with one dimension for each experimental factor. This hypercube then is decomposed into a series of outer products (tensor products) of latent vectors resembling the Principal Component representation of two-dimensional arrays. The importance of the various factors can then be derived from these latent components, as described below. This method, known as GEMANOVA (GEneralized Multiplicative ANOVA) [11,12], was used to analyze simulated data from the Delta-Notch model.

GEMANOVA has, through its multiplicative structure based on tensor outer products, the potential for discovering higher-order interaction effects in a parsimonious way. The number of parameters needed in GEMANOVA to describe the data may be much less than when using ANOVA. Veflingstad [5] showed that higher-order interactions seem to be relevant; and since complex interactions might have more influence on the model than the main effects, ANOVA would be less useful for exploring the system, and the model obtained would be difficult to interpret. The fact that GEMANOVA models are multiplicative in contrast to the additive ANOVA models means that higher-order interactions in GEMANOVA and ANOVA models are not identical; and leaving out main effects from ANOVA will not give the same effects estimates.

The simulated data from the Delta-Notch model follow a quarter fractional factorial design, which leaves many missing observations in the data hypercube. GEMANOVA handles missing data by expectation maximization [13]. This means that the position and amount of missing data will not bias the results unless crucial information is lacking. Increased amounts of missing data will, though, lead to higher variability on the estimated parameters. Using traditional ANOVA for analyzing fractional factorial design data necessarily leads to confoundings between certain main effects and interactions that makes it impossible to distinguish the effect of interactions from the main effects [14]. The usual assumption, made in these cases, namely, that higher-order interactions are not important and can be neglected, appears to be a dangerous assumption for the Delta-Notch data. The formulation of the GEMANOVA model also facilitates the estimation of higher-order interactions without having replicates. For estimating highest order interactions in a full ANOVA model, replicates are required, which may be costly to obtain. The absence of replicates in the Delta-Notch model data was one of the reasons for using GEMANOVA in the analysis.

If higher-order interactions are important, effect plots from GEMANOVA may be easier to interpret than ANOVA-based interaction plots. The latter look much complicated, especially for higher-order interactions, whereas it is clearly seen from GEMANOVA plots which level of variable has more influence on the whole system. GEMANOVA plots show, whether the factor has a positive or negative effect when going from low to high levels in its values.

One disadvantage of GEMANOVA is the lack of good model fit evaluation criteria. For instance, there are no uncertainty measures directly available for the model parameter estimates, which makes significance testing more complicated. In this paper bootstrapping of residuals is used for obtaining uncertainty measures and for testing. Bootstrapping in GEMANOVA was first described by Faergestad *et al*. [15].

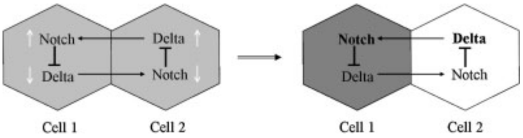


Figure 1. Schematic illustration of lateral inhibition mediated by Delta-Notch signalling [3]. The Notch concentration in cell 1 increases with the growth of the Delta level in the neighbouring cell. It causes decrease in Delta-activity in the first cell and later on an increment of the Notch level in cell 2. After some time, the cells obtain different shades of grey: dark grey means high concentration of Notch and low concentration of Delta and vice versa for white cells.

In Section 2 of this paper, we present the Delta-Notch model for data generation and the GEMANOVA model as a general method for exploring the properties of dynamical mathematical models. Moreover, we present a strategy for searching for a good GEMANOVA model using a forward selection search and significance testing using bootstrapping. In Section 3 we give the results from the GEMANOVA analysis. In order to verify some of our findings, we generate in Section 4 more data from the Delta-Notch model using a finer scale on some of the parameters. We close this article with a discussion of our findings in Section 5.

2. METHODS

2.1. The Delta-Notch model

The pattern-generating ability of two signalling proteins (Delta and Notch) controlling cell differentiation [6] in a 2D hexagonal lattice is modelled. The concentration of these two proteins determines the colour of each cell. More precisely, the cell with a high concentration of Notch will be of black colour, whereas a cell with a high concentration of Delta will be white. If there is much Delta and little Notch in a cell, the neighbouring cells tend to have little Delta and much Notch. The mechanism of interaction between Delta and Notch in a 1D cell chain is shown in Figure 1. For a 2D hexagonal lattice, the way the change of concentration of one protein triggers the change in another one, is similar. In case a cell gets perturbed with an increased level of Delta, an increase in Notch level is observed in the adjacent cells. Further, due to lateral inhibition [6], the concentration of Delta in those cells is decreasing with consequent diminution of Notch level in the center cell. In the case where a 2D hexagonal lattice of cells with equal levels of Delta and Notch (all cells having the same colour) is slightly perturbed in Delta-Notch concentrations, the protein concentrations tend to converge into a steady state where the cells obtain different shades of grey. So, if one cell obtains light grey colour, its neighbours will tend to become darker. The patterns arising depend on the initial conditions and parameter values.

The following five assumptions about the model were formulated by Collier in [6]:

1. Cells interact through Delta-Notch signalling only with cells with which they are in direct contact.
2. The rate of production of Notch activity is an increasing function of the level of Delta activity in neighbouring cells.
3. The rate of production of Delta activity is a decreasing function of the level of activated Notch in the same cell.

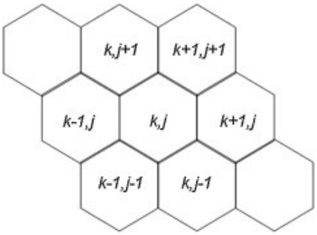


Figure 2. The scheme for indexing a 2D hexagonal array of cells.

4. Production of Notch and Delta activity is balanced by decay described by simple exponential decay with fixed rate constants.
5. The level of activated Notch in a cell determines the cell's fate: low levels lead to adoption of the primary fate, high levels to adoption of the secondary fate.

The non-linear dynamic model of how each cell interacts with its six neighbours has five control parameters. Different combinations of the parameters give different patterns (Veflingstad [5]). The production rates of the proteins Delta and Notch are expressed in terms of sigmoidal function given as

$$S(x, \theta, p) = \frac{x^p}{x^p + \theta^p},$$

where x is the amount of Delta or Notch, the parameter θ is a threshold parameter for the sigmoid curve, and p is a steepness-parameter.

The Delta-Notch model may be defined as follows:

$$\frac{dD_k}{dt} = \mu\left[1 - S(N_k, \theta_N, p_N) - D_k\right]$$

$$\frac{dN_k}{dt} = S\left(\{D\}_k, \theta_D, p_D\right) - N_k$$

Here μ is defined as a ratio of decay-rates for Delta and Notch; θ_D and θ_N are the threshold-parameters for Delta and Notch respectively; p_D and p_N are the steepness-parameters for Delta and Notch respectively; D_k and N_k are concentrations of proteins Delta and Notch in cell $k = 1, 2, \ldots$; and $\{D_k\}$ refers to the average of Delta concentration in 6 neighbouring cells and is defined by

$$\{D\}_k = \frac{1}{6}\left(D_{k,j+1} + D_{k,j-1} + D_{k-1,j} + D_{k-1,j-1} + D_{k+1,j+1} + D_{k+1,j}\right),$$

where indexes are according to Figure 2.

2.1.1. *The data*

In order to explore the properties of the Delta-Notch model, data were simulated using different parameter settings. All simulations were initiated from a cell-grid where all cells were grey (equal amount of Delta and Notch in all cells, i.e., a homogeneous steady state). But in order to obtain a pattern, a small perturbation was imposed on the homogeneous background state of the cells (the balance between Delta and Notch was disturbed by a small amount in all cells). The perturbation is defined by two parameters: "*PertSize*" and "*PertDir*". Here *PertSize* is the amount of the perturbation (percentage of Delta in steady state), whereas

Table I. Description of the attributes used by judges to evaluate the images

Name	Description	Low (1.0)	High (9.0)
Whiteness	Average color (NCS-system)	No white	White
MultiShade	How many shades of grey	No shades	Many shades
Contrast	How well the pattern is defined	Hardly	Clearly
Sharpness	Blurred, indistinct pattern	None	Clear
StraightLines	Presence of straight lines, direction is irrelevant	None	Many
PatternWhite	White pattern on black background	No clear white pattern	Clear white pattern
PatternBlack	Dark pattern on light background	No clear dark pattern	Clear dark pattern
Curls	Presence of connected paths that cross	None	Many
Continuous	Degree of continuous regions	None	High
Regular	Degree of order	None	High
Associations	Degree of associations	None	Many
MentalLoad	Visual burden during analysis of image	None	High

PertDir is the direction of the perturbation: less than or more than homogeneous steady state value ($-1/+1$). A more detailed description of how the data were simulated is given in [5].

Two levels were chosen for each of the five model parameters and the two perturbation parameters: low and high; and a 2^{7-2} fractional design was run. This gave 32 different images after convergence of the models. For six of the parameter settings, the system converged back to the state where all cells were grey with no distinct pattern across the cell grid. These were regarded as missing values with respect to pattern descriptors, as presented below. In addition, six "centerpoints" (intermediate values) were run, but these were not used in the GEMANOVA analyses presented here since GEMANOVA, as to yet, does not handle centerpoints.

The patterns emerging across the 2D grid are assumed to depend on the parameter settings, but, due to random perturbations, the exact patterns are not reproduced in consecutive runs with the same parameter setting. Hence, patterns are difficult to quantify numerically. Therefore, a sensory strategy, known from food research, was used to summarize the patterns. Twelve predefined descriptors were defined to describe the images. These descriptors were: "Whiteness", "MultiShade", "Contrast", "Sharpness", "StraightLines", "PatternWhite", "PatternBlack", "Curls", "Continuous", "Regular", "Associations" and "MentalLoad" (see Table I, [5]). The images were presented to eleven sensory judges who evaluated each image for each descriptor on a scale from 1.0 to 9.0. GEMANOVA was then used to relate parameter settings to the prescribed descriptor values. Figure 3 shows examples of grids of cells obtained under different initial conditions.

To get an overview of the associations between the descriptors and the parameter settings a PCA was run using the average judge score for all descriptors and the results are summarized as a correlation loadings plot in Figure 4. As can be read out of the Figure, the variability in the data was mostly described by two components.

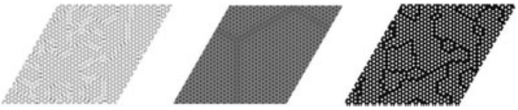


Figure 3. Examples of patterns with high scores for descriptors: "Whiteness" (left), "Straight Lines" (middle) and "Continuous" (right). White colour indicates low level of Notch and high level of Delta. Black colour is vice versa.

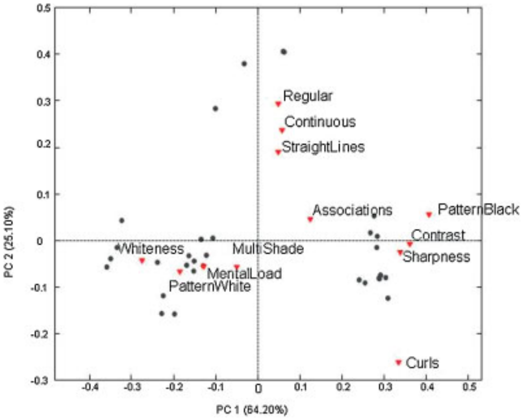


Figure 4. Correlation loadings plot from a PCA analysis of the sensory scoring data of the generated images.

PCA yielded a clear grouping of the descriptors, and, in order to illustrate the use of GEMANOVA as a tool for relating patterns to parameter settings, two descriptors were selected for further analysis. The descriptor "Whiteness" was chosen from the group "Whiteness", "MultiShade", "MentalLoad" and "PatternWhite"; and "StraightLines" was chosen from the group "StraightLines", "Continuous" and "Regular".

2.2. GEMANOVA

GEMANOVA is a relatively new method of analyzing data that are organized into an N-way array. It is suitable for data mostly influenced by complex interactions between factors.

GEMANOVA is based on the N-way method known as PARAFAC (PARAllel FACtor analysis) [12,16]. As an example, consider a situation where a response variable x is assumed to be influenced by three factors, hence, the data can be arranged into a 3-way cube ($N = 3$). A 3-way PARAFAC model is defined element-wise by

$$x_{ijk} = \sum_{q=1}^{Q} a_{iq} b_{jq} c_{kq} + e_{ijk}, \tag{1}$$

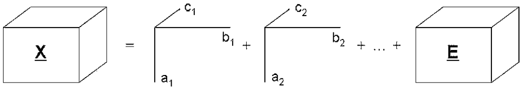


Figure 5. A PARAFAC model.

or by using tensor products by

$$\underline{\mathbf{X}} = \sum_{q=1}^{Q} \underline{\mathbf{a}}_q \otimes \underline{\mathbf{b}}_q \otimes \underline{\mathbf{c}}_q + \underline{\mathbf{E}}, \tag{2}$$

where *i*, *j* and *k* denote the level of the three factors; *Q* is the number of model components; $\underline{\mathbf{E}}$ is a residual three-way array containing terms e_{ijk} of unexplainable variation; $\underline{\mathbf{a}}_q$, $\underline{\mathbf{b}}_q$ and $\underline{\mathbf{c}}_q$ are the loadings vectors for the three so-called *modes* of component *q*; and ⊗ denotes the tensor outer product [11].

Figure 5 shows a pictorial rendition of (2).

GEMANOVA is different from PARAFAC in a sense that certain effects can be "eliminated" by setting all levels of the corresponding loading vector equal to one. This means that a given component in the model may depend on only a subset of the factors. Here are some examples of some model equations that may occur for a three-way array of data with modes *a*, *b* and *c*:

$$\underline{\mathbf{X}} = \underline{\mathbf{a}} \otimes \underline{\mathbf{b}} \otimes \underline{\mathbf{c}} + \underline{\mathbf{E}}$$
$$\underline{\mathbf{X}} = \underline{\mathbf{a}}_1 \otimes \underline{\mathbf{b}}_1 \otimes \underline{\mathbf{c}}_1 + \underline{\mathbf{a}}_2 + \underline{\mathbf{E}}$$
$$\underline{X} = \underline{\mathbf{a}}_1 \otimes \underline{\mathbf{c}}_1 + \underline{\mathbf{b}}_2 + \underline{\mathbf{E}}$$

The first model is a one-component model with one three-way interaction; the second is a two-component (two effects) model, where the second component contains only the mode *a* (modes *b* and *c* are set to one); and, hence, this is similar to a main effect in classical ANOVA. The third model has a first component with two modes (*a* and *c*),—a two-way interaction effect, and a second component with only the *b*-mode,—a main effect.

To show that GEMANOVA yields simplifications of models, let us compare a non-hierarchical ANOVA-model with a third-order interaction only and a one-component GEMANOVA model.

$$x_{ijk} = d_{ijk} + e_{ijk} \tag{3}$$
$$x_{ijk} = a_i b_j c_k + e_{ijk} \tag{4}$$

where d_{ijk} is a three-way interaction and e_{ijk} are residual terms.

The ANOVA model is described by (3) and contains *IJK* parameters, whereas the GEMANOVA model is defined by (4) and has $I + J + K$ parameters. Hence, a reduction of model parameters is obtained if one of the modes/factors has more than two levels [11]. Furthermore, the inclusion of the highest order interaction in an ANOVA model usually implies that all lower order interactions also should be included, and this will increase the parameter number dramatically (considering functional marginality here [10]).

2.3. Model selection and validation

The average score across the judges was used with only one descriptor at a time as a response variable and with the four most relevant parameters (according to the results from Veflingstad [5]) as modes in the GEMANOVA model (θ_D, θ_N, p_D and p_N). A problem with the GEMANOVA model is that there is no exact way of testing significance of the modes or determining the number of components. An alternative for model selection could be to evaluate the models in terms of predictive power as was done by [17]. However, the absence of replicates in our case makes cross-validation not an option and other model fit criteria must therefore be used. Another problem is the large number of candidate models to evaluate. The number of candidates rapidly increases with the number of modes and the number of components.In order to find a reasonable model that does not over-fit the data, we thus developed a forward selection strategy combined with bootstrap testing. The strategy is as follows: Start by fitting all possible one-component models (involving from one to the maximum number of modes) and compute the residual sums of squares (ssq) for each model. The ssq-values are plotted against the number of estimated parameters to guide the selection of a small set of models with small ssq-values to be extended to two-component models. The selected models are then extended with a second component involving all combinations of the modes. A new evaluation of ssq can then be performed if further components need to be added. At the end, the ssq for all models considered are plotted versus number of parameters to select a final set of models for further analysis with bootstrapping as described next.

2.3.1. *Bootstrapping*

A non-parametric model-based bootstrap, as described in Liland and Faergestad [18], was used for significance testing. In this procedure a GEMANOVA model is first fitted to the original data. Then the fits and the residual for every data point are calculated. In the bootstrap loop random samples of the residuals (with replacement) are added to the fits, and the model is refitted to these "bootstrap samples". This procedure is repeated *B* times yielding a set of *B* estimated models. The significance of a given mode is determined by comparing the estimates based on the original data with the distribution of estimates based on the bootstrap data sets.

One way of determining if there is a consistent positive or negative slope between two levels of a factor is to count how many of the bootstrapped models have the opposite sign of the slope compared to model for the original data. If we denote the sign of the slope between the levels by $d_{f,i,j}$, where *f* is the component number, and *i* and *j* denote the number of the levels, we can construct a hypothesis for testing the sign of the slope:

$$\begin{array}{l} H_0 : d_{f,i,j} = 0 \\ H_1 : d_{f,i,j} \neq 0 \end{array}.$$

This will have an estimated *p*-value of rejecting the null hypothesis H_0 when H_0 is in fact true:

$$\hat{p} = \frac{1 + \#\left(d_{f,i,j} \neq d^b_{f,i,j}\right)}{B + 1},$$

where $d^b_{f,i,j}$ is computed from the *b*-th bootstrap replicate. A $\hat{p}$-value close to 0 would indicate that there is a consistent direction in the effect levels that has not been generated by random noise in the data.

Using a non-parametric bootstrap means we do not have to make any assumptions about the distribution of the data. With the model-based bootstrap, we re-sample from the residuals, which is a standard bootstrapping technique for linear models (Efron and Tibshirani, [19]), but should be equally applicable in GEMANOVA models. In contrast to using jack-knifing for estimating uncertainty, non-parametric model-based bootstrapping does not remove any portion of the data. This can be of critical importance, especially in designed experiments, as key structural information connected to a limited number of samples could potentially be lost when removing observations, which might lead to unrealistic estimation of the uncertainty.

The current implementation of GEMANOVA and its underlying PARAFAC algorithm is prone to errors due to non-unique ordering of component, and possible pair-wise flipping of signs, and false convergence of models. In an automated process like the bootstrap, all these problems could lead to faulty bootstrap replicates giving unreliable results. Both the shifting ordering and sign-flipping can be almost completely eliminated from the bootstrap if the initial PARAFAC model produced in the original GEMANOVA modelling is used as a starting point for the estimation algorithm for the bootstrapped models instead of random initializations.

False convergence is sometimes a problem in complex models or when using data with missing observations. This can be seen, for instance, when models have poorer fit to the data or need a much higher number of iterations than could be expected, when two components are equal but with opposite signs, or when two components are small and almost identical. In the bootstrap algorithm used in this paper, bootstrap models having an ssq considerable larger than the ssq for the original model, or a very large number of iterations before convergence, are discarded as false convergences.

3. GEMANOVA RESULTS

The results of GEMANOVA analysis are presented as figures showing the estimated effects of the different levels of each factor. From the plots it can be observed which level of the factor that gives high scores for the given pattern descriptor. Also it is possible to make conclusions on how the parameters influence the score of the given descriptor, for instance, which parameters influence the scoring of "Whiteness".

GEMANOVA was run for the Whiteness descriptor using the stepwise forward method. The ssq-values versus the number of parameters are shown in Figure 6.

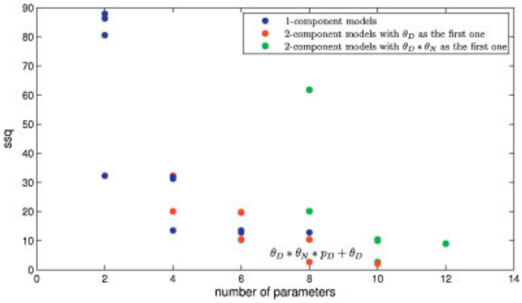


Figure 6. The ssq of one- and two-component GEMANOVA models for "Whiteness" vs number of parameters.

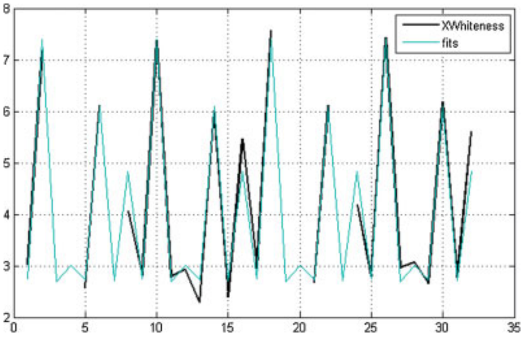


Figure 7. The fits vs the data for the "$\theta_D * \theta_N * p_D + \theta_D$"-model for "Whiteness".

The forward selection strategy pointed to the the two-component model "$\theta_D * \theta_N * p_D + \theta_D$" as a good model with small ssq and not too many parameters. This model was picked out for bootstrapping in order to test the significance of the included parameters. (Since running the bootstrap takes much time (approximately 5 hours for $B = 1000$), it was applied for only one model for each attribute.) As can be seen from the Figure 6, increasing model complexity does not reduce the ssq much. Further, choosing simpler models (with six or four parameters) gave considerable increase in the ssq-values. Moreover, Figure 7 shows that the model has a very good fit to the data. From this point of view, "$\theta_D * \theta_N * p_D + \theta_D$" might be the best model to investigate further.

The results from the bootstrapping of this model are shown in Figure 8. The small p-values mean that the modes of the chosen model are significant for the Whiteness descriptor. It is also confirmed by the fact that the histograms for the estimated effects from the bootstrap models are not overlapping for the low and the high levels of the parameters. One can see from the figure that a high level of θ_D and low level of θ_N and p_D, images will most likely be bright and, hence, will get high scores from judges with regard to "Whiteness". All parameters in the selected model were found to be significant, and no further simplification of the model seemed to be necessary.

The same procedure for seeking a model was used for the response "StraightLines". In this case the, "$\theta_N * p_D + \theta_N * p_N$"-model was selected. As can be seen from Figure 9, the parameters p_D and p_N seem to be not significant. Therefore, a model without p_N in the second component was tried in order to find a better model. A new bootstrapping gave the results that the second component, consisting only of θ_N, had a large p-value (equal to 0.246). Finally a model with only one component, containing θ_N and p_D, was analyzed. As can be seen from Figure 10, both parameters included in the model were then significant. This illustrates how the forward selection procedure combined with bootstrapping can identify a good model with a good fit and with only significantly contributing parameters. Comparing the fits of both of the models vs the data, one can see that the two-component model describes the data slightly better (Figure 11), so the final decision on which model to choose should in this case probably be based on a follow up study with more refined scales for the θ_N and the p_N parameters.

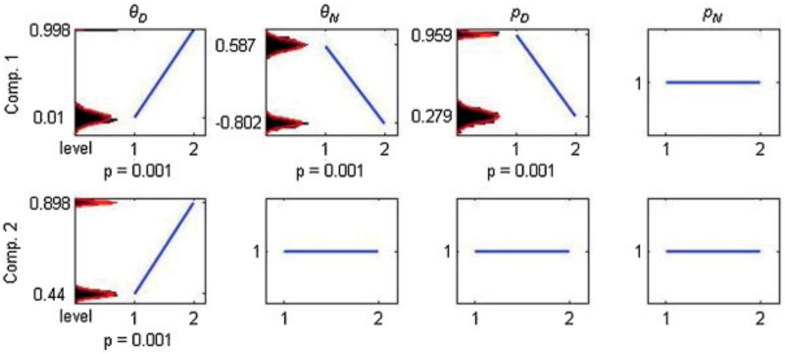


Figure 8. Effects and bootstrap plots for the "$\theta_D * \theta_N * p_D + \theta_D$" model for "Whiteness". Each row of the figure corresponds to a component of the GEMANOVA model. The first component has three modes (loadings for p_N are set equal to 1 for both levels). The second component has only contribution from the θ_D mode. The blue line segments connect the estimated effects for the low and high levels of the parameters. The *p*-values indicate the significance of each mode to the model. The histograms are the distributions of the effect estimates as found by bootstrapping. As can be seen, to get the maximum effect for the descriptor "Whiteness" (that is brighter images and high scores), one should keep θ_D at high level, whereas θ_N and p_D should be at the low level.

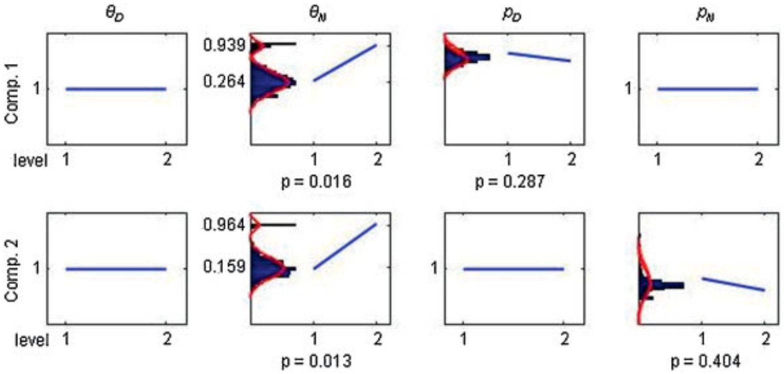


Figure 9. Effects and bootstrap plots for the "$\theta_N * p_D + \theta_N * p_N$" model for the descriptor "Straight Lines". p_D and p_N seem to be non-significant and may be left out. θ_N has a positive effect in the model and provide high scores "StraightLines".

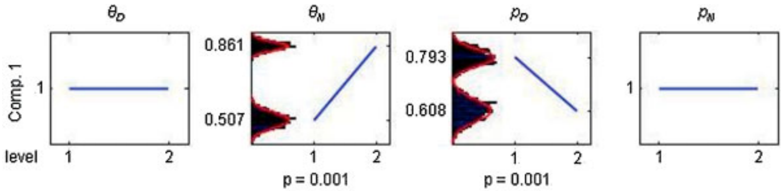


Figure 10. Effects and bootstrap plots for the "$\theta_N + p_D$" model for the descriptor "StraightLines". Both θ_N and p_D are significant for this model according to *p*-values and non-overlapping histograms for the bootstrap estimated effects. A high level of θ_N and a low level of p_D appears to give high scores for "StraightLines".

4. SIMULATION

In order to check some of the GEMANOVA results, some pattern simulations were performed. Different sets of parameter values that according to the GEMANOVA results were supposed to give high scores for the descriptors were chosen for the simulations.

As concluded from Figure 8, a high level of θ_D and a low level of θ_N and p_D should give high scores for the "Whiteness" descriptor. Therefore, in the simulations θ_D and θ_N were set equal to 0.8 and 0.2 respectively. When it comes to the parameter p_D, this was tested at a more refined scale between 1 to 10 in order to explore the influence of this parameter to the Whiteness of the patterns.. The parameter p_N did not contribute to the scores for "Whiteness"

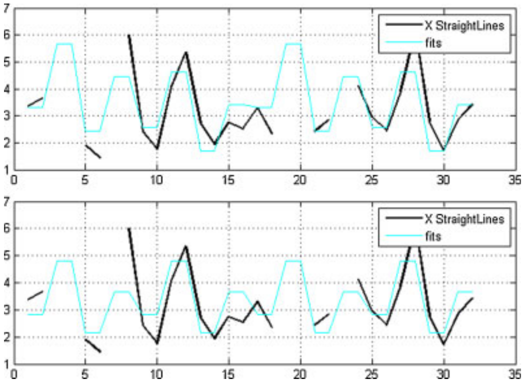


Figure 11. The fits vs the data for two models for "StraightLines": "$\theta_N * p_D + \theta_N * p_N$"- and "$\theta_N * p_D$"-models on the upper and lower plots respectively.

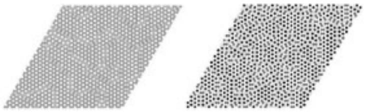


Figure 12. Patterns simulated for the set of the parameters $\theta_D = 0.8$, $\theta_N = 0.2$. For the left image $p_D = 1$, for the right one $p_D = 10$.

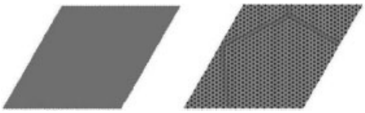


Figure 13. Patterns generated under similar initial conditions: $\theta_D = 0.6$, $\theta_N = 0.7$, $p_D = 3$. The difference was only in levels of p_N. For the image on the left hand side $p_N = 1$, for the right hand side image $p_N = 10$.

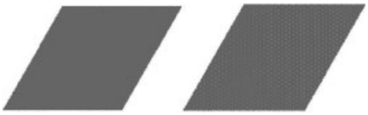


Figure 14. Images for patterns with the following set of the parameters: $\theta_D = 0.6$, $\theta_N = 0.7$, $p_D = 3$. For the left picture $p_N = 5.4$, for the right one $p_N = 5.5$.

according to the GEMANOVA results, and this parameter was, therefore, held constant throughout the simulations.

For all values of p_D, the images appeared to be bright images likely to get a high Whiteness score (Figure 12), but the images appeared to be slightly darker for high values of this parameter. These simulations supported the GEMANOVA results in its conclusion that keeping a high level of θ_D and low level of θ_N and p_D yields high scores in "Whiteness".

For "StraightLines" GEMANOVA gave the conclusion that high scores are associated with θ_N at a high level and p_D and p_N at the low level. However, such parameters combination yields pictures being homogeneously grey (initial state is a steady state), corresponding to a missing value combination in the descriptors. But bootstrapping revealed that there might be a better model only with one component with the modes θ_N and p_D, indicating that the parameters p_N and θ_D are not influential. Hence, the GEMANOVA results are somewhat inconclusive with respect to the importance of the parameter p_N. Therefore, it was decided to investigate its importance by means of simulating patterns holding θ_N and p_D as fixed while varying p_N. If p_N does not play any significant role for generating patterns, then images should be similar in the sense of appearance of straight lines no matter the value of p_N. So the following values of the first three model parameters were chosen: $\theta_D = 0.6$, $\theta_N = 0.7$, $p_D = 3$. The remaining parameter, p_N, was tested at multiple values between 1 and 10. Figure 13 shows that indeed different patterns arise for different values of p_N. Grey images were observed for values of p_N up to about 5 or 6. For slightly large values the patterns became grainy. It means that for values from 1 to 5 the initial state is confirmed a steady state, stable to small perturbations, and somewhere on the interval between 5 and 6 this stability is disturbed and a bifurcation point is likely to present. In order to localize the bifurcation point, further simulation were done on an even more refined scale on interval from 5 to 6. From Figure 14 it can be observed that the bifurcation point has a value somewhere between 5.4 and 5.5. These results were checked analytically with the condition for stability of a steady state (Collier [6]). It has been discovered that for a two-dimensional array of hexagonal cells the homogeneous steady state is linearly stable, if and only if

$$|(fg)'(x_0)| < 2,$$

where

$$f(D_k) = fg(N_k) = S(\{D\}_k, \theta_D, p_D)$$
$$g(N_k) = 1 - S(N_k, \theta_N, p_N)$$

and $(g(x_0), x_0)$ is a homogeneous steady state under the assumption, that the concentrations of Delta and Notch do not vary from cell to cell [6]. The inequality was numerically checked in Matlab using various values of p_N on the range from 5 to 6, and also these results point to the presence of a bifurcation point between 5.4 and 5.5.

5. DISCUSSION

In this paper descriptors of image patterns generated from the Delta-Notch model were analyzed as sensory data. The data were generated under different initial conditions, evaluated by sensory judges and organized into a multidimensional array. The purpose was to explore the properties of the system, to find out, which parameters do not influence the data and which of them play significant roles. This may be important for the further simplification of the mathematical model of the data and for obtaining a better understanding of its properties. Since the Delta-Notch levels are controlled locally in each cell by five parameters and the grid pattern consists of 2500 cells, more than ten thousand parameters control the whole system. Hence, it has been a challenge to obtain an understanding of the pattern-generating properties of the model. Using multivariate statistical methods is a new way of relating parameter settings to more global properties (such as patterns) of a complex system of interacting components. This can, for instance, be done with statistical methods such as ANOVA, PLS, PARAFAC or GEMANOVA.

Firstly, PCA was used to investigate correlation between parameters and descriptors, to classify groups of attributes. Based on these results, a set of descriptors was chosen for further investigation with GEMANOVA in this paper. Two descriptors and four parameters were selected: "Whiteness" and "StraightLines", and θ_D, θ_N, p_D and p_N. Some descriptors from the group "Associations", "PatternBlack", "Contrast" and "Sharpness" were indeed analyzed as well. The results are not presented in the paper since no good model was obtained. Two-component models had quite large residuals even with all the modes included. Some of them did not even converge. According to the PCA results, those attributes should have been mostly described by the first component. It might be, that PCA did not capture significance of the third component and that group of descriptors is lying on the third axis. If so, then GEMANOVA analysis was the right about including more than two components to the model. But at the same time, adding more components to the "StraightLine"-model did not give a significant reduction in ssq. Therefore, only two-components models were shown here.

GEMANOVA was chosen as a method for further analysis due to reasons described in the introduction. The Delta-Notch model seems to be defined mostly by higher-order interactions (as shown in [8]), which are quite difficult to investigate with ANOVA. Two attributes were explored with a forward selection/bootstrap scheme to obtain good GEMANOVA models. There is no precise criteria for evaluating GEMANOVA models (since no assumptions about data and errors distribution and variance are made). Therefore, it should be decided beforehand what is a good model: the one with small residuals, the one with the fewest parameters etc.

The GEMANOVA results pointed to a two-component GEMANOVA model for the attribute "Whiteness" involving an interaction between three of the parameters. Hence, the statistical analysis pointed directly to the parameter combination yielding high score with respect to the global property of pattern whiteness.

For "StraightLines" GEMANOVA pointed out as the best model (in a sense of small residuals and a small number of parameters) the one that gave grey images after simulation. Since scores for this attribute are not defined for such images, there was a missing value for this parameter setting in the original data. Due to the fact that GEMANOVA in the estimation process imputes "neutral" values for the missing values, it will be the other corners of the multidimensional data cube that control the estimation of the multiplicative parameter effects. The imputations may be regarded as neutral from an estimation point of view, but the actual value being imputed is in this case not interpretable. The bootstrap testing revealed, that the parameter p_N was non-significant, which led to a reduced model for "StraightLines". But simulations gave the opposite conclusion, that p_N does influence the degree of "StraightLines". So, in summary, the GEMANOVA analysis (and probably most other statistical analyses) is highly influenced by the presence of missing values corresponding to "basins" in the parameter space, for which the response variable (here "StraightLines") is not defined. However, the statistical analysis did anyhow lead to the discovery of a bifurcation point on the interval from 5.4 to 5.5, where the steady state is changing from being stable to being unstable. Probably selecting values of p_N larger than 6 for data generation would lead to a model with clear significance of p_N.

Besides presenting the GEMANOVA approach for exploring the properties of a complex mathematical model, this paper has also shown that the interplay between multivariate statistical modelling and purely mathematical considerations may be fruitful for this purpose. The GEMANOVA analysis with subsequent simulations revealed the presence of the bifurcation point, and this was confirmed by numerical approximations based on purely mathematical properties pointed out by Collier [6]. The complexity of the Delta-Notch model can, of course, not be captured by a simple statistical model, be it an additive or a multiplicative one, but combining statistical and mathematical tools iteratively seems to be a path for gaining new knowledge about the Delta-Notch model.

REFERENCES

1. Turing AM. The chemical basis of morphogenesis. *Phil Trans Roy Soc London* 1952; **237**: 37–72.
2. Zwietering MH, Jongenburger I, Rombouts FM, Riet K van't. Modeling of the bacterial growth curve. *Appl Environ Microbiol* 1990; **56**: 1875–1881.
3. Peschel M, Mende W. *The Predator-Prey Model: Do We Live in a Volterra World*? Academie Verlag: Berlin, 1986.
4. Stanbury JB, Wyngaarden JB, Fredrickson DS, Goldstein JL, Brown MS, (eds). *The Metabolic Basis of Inherited Disease* (5th edn). McGraw-Hill: New York, 1983.
5. Veflingstad SR. The search for relations between structure and behaviour in models of gene regulatory networks. PhD thesis, The Norwegian University of Life Sciences, Ås, Norway 2006.
6. Collier JR, Monk NAM, Maini PK, Lewis JH. Pattern formation by lateral inhibition with feedback: a mathematical model of Delta-Notch in intercellular signalling. *J Theor Biol* 1996; **183**: 429–446.
7. Gosh R, Tomlin C. Symbolic reachable set computation of piecewise affine hybrid automata and its applications to biological modelling: Delta-Notch protein signalling. *IEEEPrcSystemsBiol* 2004; **1**: 170–183.
8. Martens H, Veflingstad SR, Plahte E, Martens M, Bertrand D, Omholt SW. The genotype-phenotype relationship in multicellular pattern-generating models—the neglected role of pattern descriptors. *BMC Systems Biology* 2009; **3**: 87.
9. Montgomery DC, Peck EA. *Itroduction to Linear Regression Analysis* (2nd edn). Wiley: New York, USA, 1992.
10. Cederkvist HR, Aastveit AH, Næs T. The importance of functional marginality in model building—A case study. *J Chemometrics and Intelligent Laboratory Systems* 2007; **87**(1): 77–20.
11. Bro R, Jakobsen M. Exploring complex interactions in designed data using GEMANOVA. Color changes in fresh beef during storage. *J Chemometrics* 2002; **16**: 294–304.
12. Bro R. Multi-way analysis in the food industry. Models, algorithms and applications. PhD thesis, The Royal Veterinary and Agricultural University, Copenhagen, Denmark, 1998.
13. Tomasi G, Bro R. Parafac and missing values. *J Chemometrics and Intelligent Laboratory Systems* 2005; **75**: 163–180.
14. Montgomery DC. *Design and Analysis of Experiments* (6th edn). Wiley: New York, USA, 2005.
15. Faergestad EM, Langsrud Ø, Høy M, Hollung K, Sæbø S, Liland KH, Kohler A, Giskehaug L, Almergren J, Anderssen E, martens H. Analysis of megavariate data in functional genomics. In *Comprehensive Chemometrics* (Vol. 4), Brown S, Tauler R, Walczak R (eds). Elsevier: Oxford, 2009; 221–278.
16. Smilde A, Bro R, Geladi P. *Multi-way Analysis with Applications in the Chemical Sciences*. Wiley: Chichester, England, 2005; 59–66, 340–349.
17. Ebrahimi D, *et al*. Generalized multiplicative analysis of variance of kill kinetics data of antibacterial agents. *J Chemometrics and Intelligent Laboratory Systems* 2008; **92**: 101–109.
18. Liland KH, Færgestad EM. Testing effects of experimental design factors using multi-way analysis. *J Chemometrics and Intelligent Laboratory Systems* 2009; **96**: 172–181.
19. Efron B, Tibshirani RJ. *An Introduction to the Bootstrap*. Chapman and Hall/CRC: New York, USA, 1998.

Paper II

ARTICLE IN PRESS

CHEMOM-02359; No of Pages 11

Chemometrics and Intelligent Laboratory Systems xxx (2011) xxx–xxx

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemolab

Nonlinear modelling of curvature by bi-linear metamodelling

Julia Isaeva [a,*], Solve Sæbø [a], John A. Wyller [b,c], Olaf Wolkenhauer [d], Harald Martens [b]

[a] Department of Chemistry, Biotechnology and Food Science, Norwegian University of Life Sciences, P.B. 5003, N-1432 Ås, Norway
[b] Centre for Integrative Genetics (CIGENE)/IMT, Norwegian University of Life Sciences, N-1432 Ås, Norway
[c] School of Mathematical Sciences, University of Nottingham, NG7 2RD, UK
[d] Department of Systems Biology and Bioinformatics, University of Rostock, 18051 Rostock, Germany

ARTICLE INFO

Article history:
Received 22 December 2010
Received in revised form 12 April 2011
Accepted 19 April 2011
Available online xxxx

Keywords:
Curve-fitting
Nonlinear
Modelling
Non-iterative
Bi-linear
Multivariate metamodeling

ABSTRACT

The phenomenon of line curvature – that a smooth line *z*=*f(x)* deviates from being straight – is often observed in scientific data. Fitting nonlinear mathematical models to curves today requires slow iterative search processes prone to errors due to local optima. A new generic method, the direct look-up method, is presented for mathematical description of such curvature with less subjectivity and simpler parameter estimation. The new modelometric method is based on bi-linear metamodelling to emulate a whole set of potentially relevant nonlinear models capable of describing curvature. A comprehensive set of 38 nonlinear mathematical models was here collected from different scientific disciplines. Each model can generate a wide range of monotonous sigmoid or arched output shapes depending on their set of input parameter values. For each nonlinear model, its model phenome – its repertoire of possible output curves – was established once and for all by computer simulations statistically designed to fill the relevant model parameter space at a chosen resolution. This simulated curve set was compressed by Principal Component Analysis. The resulting set of 38 bi-linear metamodels emulates the input–output behaviour of the nonlinear models. Then, to parameterise new curves, the input data of each curve were fitted to all relevant nonlinear models via their metamodels. Models with good enough fit were listed as plausible, and their unknown parameter values were estimated from their closest known simulation design points. Thereby, the slow, iterative nonlinear curve fitting was replaced by a fast linear projection with a simple look-up quantification. The traditional problem of choosing initial values to avoid local optima was eliminated. The multivariate metamodelling allowed a wide set of nonlinear curvature descriptions to be handled.

1. Introduction

1.1. Linear and nonlinear curve-fitting

In different fields of experimental science researchers often collect continuous batches of data where a dependent variable (response variable) z is related to an independent variable denoted by x. Then the data may be denoted as vector sets $[x, z]$. The relationship between x and z may be visualised as curves and represented symbolically by

$$z = f(x). \tag{1}$$

If f is a smooth function of x, mathematical modelling is useful for quantifying it. But if relationship $z=f(x)$ is a curve that requires nonlinear mathematical modelling, the process of obtaining model parameters can be slow, cumbersome and uncertain. We here present a faster, safer and less subjective way to parameterise data with curvature in $z=f(x)$ relationships. For simplicity, we focus on the simplest case, with only one x-variable and one z-variable.

Simple additive (linear) functions of the straight-line type

$$z = b_0 + x{\cdot}b_1 \tag{2}$$

can give a reasonable local approximation to smooth, weakly nonlinear curve types $z=f(x)$. Moreover, linear functions are easy to deal with statistically since all their model parameters can be estimated in a single step by regression, usually by versions of ordinary least squares (OLS) or weighted least squares (WLS) regression [1]. Single-step linear regression models may, therefore, for instance, be used for development of calibration models [2], which can then be applied for predicting z_i from new values x_i or vice versa. However, graphical displays of z vs x from a complex system often show a curved line that our background knowledge tell us cannot be meaningfully represented by the straight-line model. A curved polynomial model

$$z = b_0 + x{\cdot}b_1 + x^2{\cdot}b_2 \tag{3}$$

may possibly give adequate fit to the data. But although polynomials are also additive and, hence, solvable by a simple one-step regression,

[*] Corresponding author. Tel.: +47 6496 6243.
E-mail address: julia.isaeva@umb.no (J. Isaeva).

0169-7439/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.chemolab.2011.04.010

the polynomial coefficients or parameters b_0, b_1 and b_2 can seldom give insight into the underlying causal processes. To give scientific meaning to strongly curved relationships $z=f(x)$ usually require nonlinear mathematical modelling. Examples of this come from physiological data (response $z=f$(stimulus x) [3,4]) , growth curve data (cell count $z=f$(time x) [5]), kinetic reactions (concentration $z=f$(time x) [6]), regulatory mechanisms (rate $z=f$(state x) [7]), as well as statistical distributions (cumulative frequency $z=f$(level x) [8]). If response z in Eq. (1) was generated by x by a nonlinear process, we may approximate this known or unknown nonlinear phenomenon by one or more nonlinear mathematical models

$$z = F_m(x; \mathbf{p}_m), m = 1, 2, \ldots, M, \tag{4}$$

where $F_m(\cdot)$ is a mathematically defined function and $\mathbf{p}_m = \left[p_{m,1}, p_{m,2}, \ldots\right]$ is the vector of parameters for this model m for a given data set [x, z]. If a good choice of the functional form $F(\cdot)$ is available, one can use it to summarise lots of input data [x, z] in terms of a few estimated parameters $\mathbf{p}_m$ for interpretation. But if the functional form of the causal phenomenon is uncertain, and a mechanistic modelling is called for, then a range of different mathematical models $F_m(\cdot)$, $m=1, 2, \ldots, M$, should be tried, to see which of the alternative nonlinear models describe the data sufficiently well. This can be difficult in practise: Fitting nonlinear models to observed curves can be challenging, both in terms of computational load and uncertainty of the parameter estimates due to local optima, especially if the number of curves is very high.

The fitting of a nonlinear model to input curve data may sometimes be simplified by linearising the input variables in nonlinear transformations (e.g. $z=log(z_{input})$ or $x=1/x_{input}$) and then applying linear regression to the linearised variables. But the price of this may be a more complicated heteroscedastic error structure in the transformed data [9], that may generate high statistical estimation errors unless handled correctly. Moreover, many nonlinear models are simply impossible to linearise well enough to allow single-step regression fitting.

A more versatile curve fitting approach is presented by numerical approximation. The iterative search process is intended to find parameter combinations $\mathbf{p}_m$ in $z = F_m(x; \mathbf{p}_m)$ that minimise or maximise a chosen criterion. The most common criterion is the least squares (LS), in which the sum-of-squares of the lack-of-fit between data z and model $F_m(x; \mathbf{p}_m)$ is minimised [1]. But other criteria are also common. For instance, in mathematical statistics, maximum likelihood (ML, [1]) estimator and the Bayes estimator [1] employ distribution-based criteria that typically depend on certain conditions for or assumptions about $F(\cdot)$ and the parameters $\mathbf{p}$, such as assumptions about the distribution of z given x (ML and Bayes) and certain prior assumptions about the parameters $\mathbf{p}$ (Bayes). To optimise the criterion, the parameter estimation requires numerical optimisation [10,11] using a numerical search process, e.g. an iterative, so-called “hill-climbing” method.

1.2. Problems with nonlinear curve fitting

Iterative fitting of nonlinear models to data is important but wrought with problems. Curve-fitting methods like simplex optimisation [10] or conjugated gradient optimisation [12] can be computationally time-consuming. Iterative hill-climbing methods are, therefore, impractical in cases where a massive number of observed data sets [x, z_i^{obs}], $i=1, 2, \ldots, N$, are to be described by nonlinear modelling, as in the case of Isaeva et al. [13], where >170,000 curves [14] had to be fitted to a five-parameter logistic (5PL) function [15]. Parallel computer processing does not help much since it is difficult to foresee how many clock cycles are needed in each iterative curve fitting process. Faster estimation methods for nonlinear functions are, therefore, needed.

To avoid over-parameterisation, i.e., overly optimistic fit to the empirical data, it is, of course, important to assess the modelling critically, both by cross-validation and by test data acquired independently later on. But more importantly, it can often be difficult to select the right mathematical form of the function $z = F(x; \mathbf{p})$ by looking at the input data, especially if the system is complex and the data are noisy, which is often the case in real experiments. So it may be a good idea to try several different mathematical models $z = F_m(x; \mathbf{p}_m)$, $m=1, 2, \ldots, M$, for each input data set [x, z_i^{obs}]. On the other hand, the high computational load in nonlinear curve fitting makes it tempting for scientists to fit only one chosen nonlinear model, $z = F_{chosen}(x; \mathbf{p}_{chosen})$, to their data — even in situations where it is not clear that the chosen mathematical form is the best causal representation of the system at hand. The model may serve its immediate purpose of summarising the data, but that goal may be too narrow: An apparently not-too-bad fit to the data may be deceptive and misleading. Another reason for shunning away from trying more than one model may be the fear of making false discovery if trying too many alternative explanatory models on too few data. This is *per se* a sound scepticism, but hardly a big problem unless the number of alternative models is so high that it approaches the number of data points in [x, z], and the data are noisy. Another possible reason why scientists may stick to only one nonlinear model is that the scientists' choice of a mathematical model is a human activity with a strongly cultural, even subjective component. In the end, a choice of a nonlinear mathematical model form usually has to be made. But it would be desirable to delay that decision until it is clear how different, potentially plausible models fit the observed data.

Another problem with iterative numerical estimation methods is the risk of getting stuck in local optima, which implies a sensitivity to the starting values chosen for the parameters prior to the search process. In statistics this is a well known problem in numerical ML-estimation. But also the Markov Chain Monte Carlo methods [16–18] frequently used in Bayes approximations may fail to converge (or converge very slowly) towards the desired target [19]. Unfortunately, local numerical sensitivity analysis does not reveal that the attained optimum is local rather than global. Hence, iterative search processes can give erroneous results without the user knowing it, due to local optima. A remedy is to repeat the search process from different starting points, chosen, e.g., according to some statistical design, and retain the optimal solution with best fit to the data. But that increases the computational load and still cannot guarantee that the global optimum has been found. It would be desirable to have an estimation method that overcomes the problem of local suboptimal solutions and that makes the choice of starting values irrelevant.

1.3. Nonlinear curve fitting by bi-linear metamodelling

In order to deal with the problems listed above, an improved method for fitting nonlinear models to curves should be fast and accurate, and without a need for prior unnatural assumptions and conditions (robust method); it should be objective in the sense that a wide range of functions may be considered; and it should not have any numerical instabilities or problems with local optima and choice of parameter start values.

The new generic method to be presented here is based on the fact that a nonlinear mathematical model can usually be emulated by a simpler multivariate metamodel [20,21]. For a given problem type or phenomenon (e.g., smooth curves), a wide range of potentially relevant nonlinear models are collected. For each nonlinear model, its so-called *model phenome* (the collection of all relevant types of outputs from the model, [20]) is computed, once and for all, by extensive simulations that span the model's design parameter space at sufficient density according to a statistical design. A

Please cite this article as: J. Isaeva, et al., Nonlinear modelling of curvature by bi-linear metamodelling, Chemometr. Intell. Lab. Syst. (2011), doi:10.1016/j.chemolab.2011.04.010

compressed bi-linear metamodel is generated by Principal Component Analysis (PCA) of the many calibration curves in this model phenome. Each new unknown curve can then be fitted by the nonlinear functions via these metamodels: Simple SIMCA classification [22] (linear projection onto these bi-linear metamodels) reveals the best-fitting models for this curve. The unknown nonlinear model parameters are then estimated, using the known parameter values of the simulated curve(s) in the model phenome data base, that resemble the unknown curve the most.

An implicit version of the new parameter estimation method was used by Kohler et al. [23] for optimised correction of infrared cell spectra by Extended Multiplicative Signal Correction (EMSC) and by Kohler et al. [24] for correction of time warping problems in mass spectra. The method is here demonstrated for nonlinear modelling of a commonly observed nonlinear phenomenon: curvature (arch- or sigmoid-like shapes) in observations $z=f(x)$. The method is illustrated for two situations:

1. The exact mathematical form of the underlying causal structure $f(\cdot)$ is unclear, so that many different nonlinear models $z=F_m(x;\mathbf{p}_m)$, $m=1, 2, \ldots, M$, must be tested.
2. The choice of nonlinear model $z=F_{chosen}(x;\mathbf{p}_{chosen})$ has already been made, and the problem is to estimate the unknown parameters $\mathbf{p}_{chosen}$ in a fast way without danger of local optima.

A comprehensive set of potentially relevant, but mathematically different nonlinear models will be identified, each capable of generating a wide range of similar-looking smooth curves $z=F_{chosen}(x;\mathbf{p}_{chosen})$ with 0 or 1 inflection point, depending on their parameter values. The models range from kinetic and regulatory models in biology and chemistry, via cumulative statistical distributions to trigonometric functions. The models are very different from each other in terms of parameters, and one might say that parameter estimates from this new method, as from fitting data with polynomials, do not give any insight into the phenomenon either. However, such a collection of models, yielding a set of several alternative solutions, might open up new underlying processes that a scientist did not think of. Hence, in addition to presenting a useful estimation method, the present paper also addresses the mathematical richness with which the phenomenon of curvature can be addressed.

In the next sections we describe the new method, the direct look-up (DLU) method, which we claim largely satisfies these requirements (Section 2). We then illustrate the method on simulated data in Section 3 and discuss our results to identify pros and cons of the method in Section 4.

2. Theory

2.1. Choice of relevant nonlinear mathematical models

The general idea behind the new approach is to construct a large "library" of alternative nonlinear functions with an extensive set of realisations – a model phenome – for each function. One would then be able to compare a new set of input data from an experiment with the library curves and determine the functions and their parameter values that best describe the data, simply by finding the closest matches in the library. To make this approach feasible, the library has to be organised in such a way that it holds as many potential functions as possible, and the look-up process must be fast. Although the look-up principle is simple for one given function, there are many challenges that need special attention in order to make it work effectively. For instance, very many potential curves could be added to the library, and a number of alternative functions may more or less describe the same phenomenon. Hence, in the process of building the library we have to deal with issues like: sampling, experimental design, storage problems, pattern recognition and nearest neighbour localisation. The steps towards building a fully functional look-up library are described in the following.

2.1.1. Choice of relevant models

To describe the phenomenon of a simple curvature in relationships $z=f(x)$, we limit ourselves to be able to fit smooth monotonous curves with 0 or 1 inflection points. This includes increasing or decreasing curves of an arch-like or sigmoid character. A wide range of function types $z=F_m(x;\mathbf{p}_m)$ were collected from different fields of science. Most of the functions in the library yield curves of a sigmoidal form since such curves are commonly observed in many natural processes (e.g., growth of bacteria [5]). The set is not intended to be complete, but should cover a very wide range of curvature, and thus form the basis for developing a full library of mathematical functions yielding simple curvature.

The left panel of Fig. 1 illustrates the range of curves attainable with one of the nonlinear functions, the logistic function

$$z=F(x;p_1,p_2)=\frac{1}{1+exp}(-p_1x+p_2). \tag{5}$$

The curves were simulated from Eq. (5) using different parameter combinations (see Table B.1). This function can generate many different kinds of curves, ranging from approximately straight lines to almost step functions.

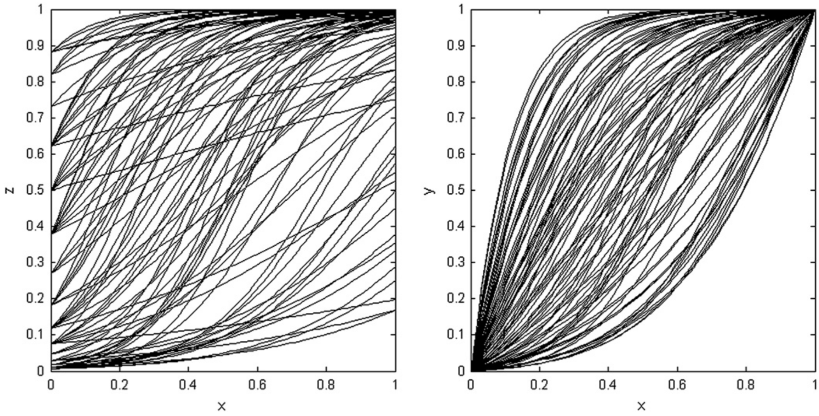


Fig. 1. Examples of curves generated by one model: the logistic function – Model 24 in Table B.1 in Appendix B – under different parameter combinations. One hundred curves were randomly selected from the 654 curves stored in the model phenome library for this function. On the left panel – curves before preprocessing; on the right panel – after preprocessing.

Please cite this article as: J. Isaeva, et al., Nonlinear modelling of curvature by bi-linear metamodelling, Chemometr. Intell. Lab. Syst. (2011), doi:10.1016/j.chemolab.2011.04.010

The first version of the library consists of a variety of function types, and amongst them are: the Hill function, the logistic function and the Michaelis–Menten equation from chemistry/biology, some trigonometric functions and some cumulative distribution functions (CDF's) from statistics, e.g.:

$$\text{Hill} \quad z = \frac{x^p}{x^p + \theta^p}, \tag{6}$$

$$\text{5PL} \quad z = \frac{1}{\left(1 + \left(\frac{x}{p_1}\right)^{p_2}\right)^{p_3}}, \tag{7}$$

$$\text{Sinus} \quad z = sin\left[p_1 \cdot \left(\pi x - \frac{\pi}{2}\right)\right], \tag{8}$$

$$\text{Michaelis–Menten} \quad z = \frac{x}{x + 0.01 + p_1}, \tag{9}$$

$$\text{CDF of normal distribution} \quad z = \frac{1}{2}\left(1 + erf\left(\frac{x - p_1}{\sqrt{2p_2^2}}\right)\right), \tag{10}$$

where $erf(x)$ is the error function $erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$

For completion, some more naive functions like a straight line and polynomials up to the third degree are also included. A full list of the 38 functions selected is given in the Table B.1 in Appendix B. Additional parameters describing scale (slope) and offset (baseline) were omitted from the core function parameters and considered as preprocessing parameters (see Section 2.1.3).

2.1.2. Choice of design for the computer experiments

Having chosen what functions to include, the next step was to generate the library of curves for each function, – its model phenome – by a computer simulation. The parameters are continuous for most of the functions (except a few number of them that have integer parameters). In the library only a finite set of curves may be computed and stored. Hence, the parameter range must be explored efficiently in terms of a finite number of curves. To span the relevant range of parameter space evenly, factorial designs with many levels of each factor were employed. It resulted in many curves for each function in Table B.1 (except for a few functions chosen to be implemented with no parameters except slope and offset — they are represented by only one curve). For functions with less than four parameters, a fine grid of values was chosen for all parameters, and curves for all combinations of values in the grid were generated. For functions with more than three parameters, combining all possible combinations of so many parameter levels will lead to a combinatorial explosion. Therefore, the multi-level binary replacement (MBR) method, a new method for reduced multi-level multi-factor designs [20,21] may be used for choosing a sensible combinations of the factor levels. The MBR design method was demonstrated here for the generalised logistic curve with four parameters (Model 26 in Table B.1), resulting in $256 = 2^8$ curves.

Thus, for every functional model $F_m(x; \mathbf{p}_m)$, $m = 1, 2, \ldots, 38$, there were generated N_m curves of a form

$$z_{j,m}(x) = F_m\left(x; \mathbf{p}_{j,m}\right), \tag{11}$$

where abscissa x is, e.g., time from 0.001 to 1; $\mathbf{p}_{j,m}$ is a parameter set for simulation j for model m. Vector $z_{j,\ m}(x) = [z_{j,\ m}(0.001), \ldots, z_{j,\ m}(1)]$ forms time series vector $z_{j,\ m}(1 \times K)$ with K data points.

2.1.3. Preprocessing

A certain standardisation step had to be adopted for the library curves, for two reasons. First of all, the preprocessing had to represent the curves in a way that allowed efficient bi-linear metamodelling. At the same time, the preprocessing should ensure that both the x and the z variables were modelled in a way that was independent of the unit in which they were originally given. All simulations were calculated for x in the interval $x \in [0.001;1]$. The interval value $x_{min} = 0.001$ was chosen so that $x = 0$ was ignored to avoid division by zero for some functions. The interval value $x_{max} = 1$ was chosen so that certain trigonometric curves gave the desired curvature type. Moreover, we chose to ensure that *z increases* with x in all curves; this required that decreasing curves were flipped in to increasing ones. And finally, we required that the $z \in [0;1]$. Hence, after the initial simulations, each curve was forced to start at (0:001;0), to have their functional values between 0 and 1 and to be increasing.

For simplicity, standardisation of the x-range was not implemented since only the interval [0.001;1] was considered for all simulations. However, a new curve may later be observed on another interval. An x-preprocessing must, therefore, be done to make it compatible with the curves in the data base. In case preprocessing of x is needed, one may do the following:

$$x = x_{min} + (x_{max} - x_{min}) \cdot \frac{x_{input} - x_{min_0}}{x_{max_0} - x_{min_0}}, \tag{12}$$

where x_{input} is whatever abscissa is originally used, with $[x_{min_0}, x_{max_0}]$ being the minimum and maximum of x_{input}, $[x_{min};x_{max}]$ is the desirable x-interval (in this case [0.001;1]).

After adjusting the x-interval, preprocessing of z is also needed:

$$y_{j,m} = \frac{z_{j,m} - off_{j,m}}{sl_{j,m}}, \tag{13}$$

where $off_{j,\ m}$ and $sl_{j,\ m}$ are offset and slope parameters such that $y_{j,m}(0.001) = 0$ and $y_{j,m}(1) = 1$. From now on, z will represent ordinates of simulated or measured input curves and y the corresponding of preprocessed curves.

All transformations mentioned above are linear, and, therefore, give no critical change of the shapes of the curves. Furthermore, all of them can be easily reconstructed given the values of the preprocessing parameters. An example of preprocessing of the logistic curve simulations can be seen in Fig. 1, right panel. Fig. 2 displays 100 randomly chosen preprocessed curves from the whole library of 38 different model types. It can be noticed here how different the curves are.

2.1.4. Compression of the model phenomes

After generating the curves for many different parameter value combinations for each of 38 function types, the total number of simulated curves was approximately 50,000. Since all preprocessed curves are increasing from 0 to 1, and the function types for given parameter values may give rise to similar curves, there is a lot of redundancy in the data. Therefore, the simulated data for each function were compressed by means of a bi-linear, mean-centred Principal Component Analysis (PCA), in such a way that a small set of basis vectors that spans the majority of the variation in the complete curve set. For instance, assuming that we have 1,000 simulated curves. Instead of saving all the curves, just a few “basal” curves (the mean curve and orthonormal loading vectors and score vectors) were stored. From them all the original curves may be reconstructed with only minimal loss. The number of basal curves to be stored was chosen as the minimum number to capture at least 99.9% of the variation in the original curve set. Usually the size of the basis is from three to fifteen curves (principle components, PCs) for those function types included in this study. The only exception in the library is the CDF of normal distribution with 27 basal curves. Number of PCs for every function in the library can be found in the Table B.1 in Appendix B.

The compression of data is made in the following way (Fig. 3):

$$Y_m = \overline{y}_m + T_m V_m' + E_m, \tag{14}$$

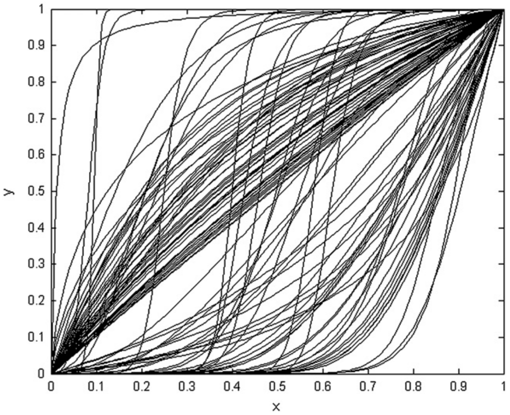


Fig. 2. Examples of z-preprocessed 100 randomly chosen curves from the whole set of 38 different curve-generating mathematical functions in the database. After preprocessing all curves are increasing, varying between 0 and 1 and having its minimum at $x = 0.001$ and its maximum at $x = 1$.

where Y_m is the collection of all N_m preprocessed curves of model m; $\bar{y}_m$ is the arithmetic mean of the curves; $T_m(N_m \times A_m)$ and $V_m(K \times A_m)$ are the scores and loadings for this A_m-dimensional metamodel, and E_m is the unmodelled residuals. For the metamodel to give adequate representation, the variance in E_m should amount to no more than, e.g., 0.001% of the initial variance in Y_m.

Hence, from the pictorial rendition of such data compression in Fig. 3, instead of having a data matrix of dimension 1,000 × 100, we obtain two matrices: a loadings-and-mean matrix of size 7 (e.g.) × 100 (basal curves, or PC's) and another matrix of size 1,000 × 6 (scores, or coordinates of original curves in the basis).

Thus, data are stored in the library as scores and loadings along with the corresponding parameter values (including mean centering values and preprocessing parameters). We will refer to this as the model phenome of the given function, expressing the "phenotypic" repertoire of the curves that the function type m may give rise to. We can use the term *phenome* here since we have a collection of observable characteristics or behaviours of these functions. The collection of function phenomes now comprises the complete look-up library of functions (Fig. 4).

The residual terms and residual variances are calculated for each curve j in the model m to assess goodness of models in future analysis:

$$e_{j,m} = \left(y_{j,m} - \bar{y}_m\right) - t_{j,m} V'_m, \tag{15}$$

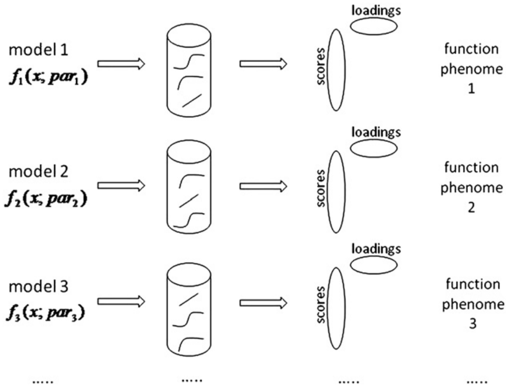


Fig. 4. Model phenomes as they are stored in the data base: represented by scores and loadings vectors.

$$s^2_{j,m} = \frac{e_{j,m} \cdot e'_{j,m}}{K}. \tag{16}$$

The variance $s^2_{j,\ m}$ represents the distance from the curve j to the model m. For every function phenome m they were sorted in ascending order and their maximal value was denoted as $s^2_{max,\ m}$. Then we found such a cut-off variance, $s^2_{95,\ m}$, so that 95% of the curves in that phenome have smaller variance than $s^2_{95,\ m}$. The same was done for 99% of the curves ($s^2_{99,\ m}$) and 99.9% ($s^2_{99.9,\ m}$), i.e., three confidence intervals for every model were defined. All these values for each model $m = 1, 2, \ldots, 38$, can be found in Table B.1 in Appendix B.

2.2. Look-up

2.2.1. Preprocessing

Let us assume now that a certain experiment yielded N observed curves $z_i^{obs}(x)$, $i = 1, 2, \ldots, N$, caused by an unknown function $f(x; \mathbf{p})$ with unknown parameter values $\mathbf{p}$. Those curves are to be parametrised. For each curve $z_i^{obs}(x)$, first, a proper function type $F(\cdot)$ has to be found, and then the parameter values that describe curves best have to be determined. For each z_i^{obs} the following form is sought:

$$z_i^{obs}(x) = a_i + b_i \cdot F_m(x; \mathbf{p}_i) + e_i(x), \tag{17}$$

where a_i and b_i are the offset and slope parameters respectively that define the difference in units in which the model phenome data base and the actual curves are given in.

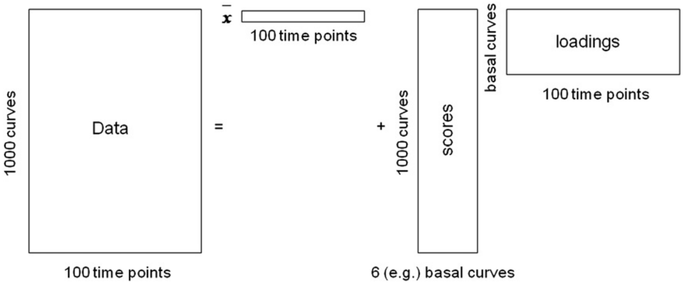


Fig. 3. Compression data with PCA. Here $\bar{x}$ is a mean curve.

Please cite this article as: J. Isaeva, et al., Nonlinear modelling of curvature by bi-linear metamodelling, Chemometr. Intell. Lab. Syst. (2011), doi:10.1016/j.chemolab.2011.04.010

In order to use the metamodel library for parameterising curves, the new input curves have to be compatible with the curves in the data base, i.e.:

- to be increasing;
- to have its ordinate values between 0 and 1;
- to be defined on the abscissa interval [0.001;1];
- to have 100 observation points of the same abscissa values as used in the model phenomes.

Therefore, the same procedure of preprocessing as for curves in the data base (Section 2.1.3, Eq. (13)) must be applied to the new curve:

$$y_i^{obs} = \frac{z_i^{obs} - off_i^{obs}}{sl_i^{obs}}, \tag{18}$$

where off_i^{obs} and sl_i^{obs} are calculated in such a way that $y_i^{obs}(0.001) = 0$ and $y_i^{obs}(1) = 1$.

This will result in an increasing curve on the [0.001;1]-interval with values from 0 to 1. To fulfil the last requirement, interpolation along the abscissa may be needed, e.g., if the new curve has fewer than 100 observation points. 1-D linear interpolation [25] can be used to find values of y at intermediate points inside the interval [0.001;1] so that those points are equally distributed in the interval.

2.2.2. Non-iterative curve fitting

After preprocessing, the new curves are projected onto each of the phenome metamodel subspaces $m = 1, 2, \ldots, 38$ in the library. Considering the way the curves were stored in the data base, this means that a score vector for a new curve is found using projection (linear OLS) on the loadings (basal curves) in the library from 2.1.1:

$$t_{i,m} = \left(y_i^{obs} - \bar{y}_m\right) \cdot V_m. \tag{19}$$

This is done within every function phenome in the library producing a score for each function type and a residual term is obtained:

$$e_{i,m} = \left(y_i^{obs} - \bar{y}_m\right) - t_{i,m} V'_m. \tag{20}$$

2.2.3. Finding the best models

After the projection of the new curves has been done and residuals were found by Eq. (20), a distance for every curve to each model $m = 1, 2, \ldots, 38$ is defined by the following formula for residual variance:

$$s^2_{e_{i,m}} = \frac{e_{i,m} \cdot e'_{i,m}}{K}. \tag{21}$$

To choose the closest models to each curve in terms of the smallest residual variance, the latter are sorted in ascending order, and in this paper those with $s_{e_{i,\ m}}{}^2 < s^2_{99,\ m}$ are considered for further parameter estimation. The choice of a confidence limit is up to the experimentalist since he or she should decide how precise models have to be.

2.2.4. Finding the best parameter set

Next step is to find which parameter values give the closest fit to the curve. This is here done by calculating the distance between the "unknown" curve and each individual curve in the given function phenome among those that are chosen in Section 2.2.3. The distance is calculated in the score space:

$$s_{t_{i,j,m}} = \sqrt{\left(t_{i,m} - t_{j,m}\right)\left(t_{i,m} - t_{j,m}\right)'}. \tag{22}$$

In this paper, when new curves are simulated without noise, the distance between scores is just a usual Euclidean distance. If new unknown curves are affected by noise from the experiment, Eq. (22) has to be modified (see Section 4 and [13]).

This distance measure is used to find the closest fit to the new curve in the score space within every function phenome. When all those distances $s_{t_{i,\ j,\ m}}$ for chosen plausible models are calculated, they are sorted over all chosen models, and parameter sets associated with, e.g., the ten best fits are considered as alternative solutions:

$$\begin{array}{lll}
\text{Alternative 1:} & m_1 & \hat{\mathbf{p}}_{i,1} = \mathbf{p}_{j,m_1} \\
\text{Alternative 2:} & m_2 & \hat{\mathbf{p}}_{i,2} = \mathbf{p}_{j,m_2} \\
 & \cdots & \cdots \\
\text{Alternative 10:} & m_{10} & \hat{\mathbf{p}}_{i,10} = \mathbf{p}_{j,m_{10}}
\end{array}$$

Here m_l may be equal to m_k for $l \neq k$, $l, k = 1, 2, \ldots, 10$, i.e., one functional model can appear several times among alternative solutions.

2.2.5. Removing the preprocessing

The only thing remained is to estimate the unknown offset and slope parameters a_i and b_i for each of the alternative suggestions. This is done by inserting Eqs. (17) and (18) into Eq. (13), which yields

$$\hat{a}_i = off_i^{obs} - off_{j,m} \cdot \frac{sl_i^{obs}}{sl_{j,m}}, \tag{23}$$

$$\hat{b}_i = \frac{sl_i^{obs}}{sl_{j,m}}. \tag{24}$$

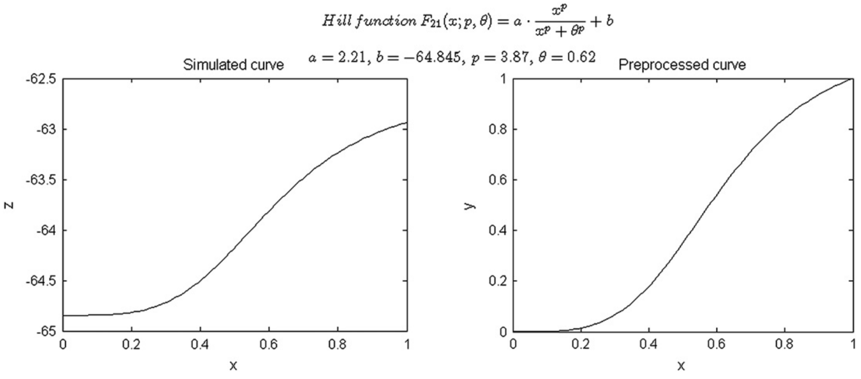


Fig. 5. Simulated curve to be projected onto the library before (left) and after (right) preprocessing.

Please cite this article as: J. Isaeva, et al., Nonlinear modelling of curvature by bi-linear metamodelling, Chemometr. Intell. Lab. Syst. (2011), doi:10.1016/j.chemolab.2011.04.010

The curve fitting is then completed to yield the fitted model:

$$\hat{z}_i^{obs}(x) = \hat{a}_i + \hat{b}_i \cdot F_m(x; \hat{\mathbf{p}}_i). \tag{25}$$

The final step of the look-up procedure is to present a list of the best curves (in descending order) along with the function parameters and the preprocessing parameters. Then it is up to the experimentalist to decide which curve he or she thinks describes the measurements in the most meaningful way. It may very well happen that the choice is not from the top of the list if the experimentalist has some prior knowledge ruling out certain function types. Still, the method presents a non-subjective list of good fitting function alternatives that may be enlightening for the user.

3. Results

To verify how the DLU method works, we illustrate the look-up procedure for a new noise-free curve. In the subsequent paper (Isaeva et al. [13]) the method is assessed and employed for massive amounts of noisy curves. A rather arbitrary choice of a function type and parameters was made. However, since it was desirable to show that the method is able to fit a model to an entire curve and not just to segments that can resemble a straight line, a sigmoidal function type – Hill function – was chosen. Fig. 5 shows the chosen curve and its transformation after preprocessing as well as the chosen parameters set.

No preprocessing of the *x*-axis was required since the curve was simulated on the same interval as the library curves. Therefore, only *z*-scaling was performed, to force the curve to lie between 0 and 1.

Distance measures after fitting the curve to all 38 function phenomes revealed that the 15 functions in Table 1 match the curve best (in ascending order of the residuals variance) with ${s_{e_{i,m}}}^2 < s_{99,m}^2$. It can be seen that many of the squared distances were quite small (between 10^{-9} and 10^{-6}) indicating that most of these models gave a reasonably good fit to this test curve.

Further, to find the best fit among all possible curves, the distances from the simulated curve to all library curves within all function phenomes from Table 1 were computed and sorted, and the parameter estimates for the 10 best suggestions were found. Fig. 6 shows the ten best solutions for the simulated curve and their errors. As it can be seen, all predicted curves are clearly close to the new curve, although not all of them are Hill-functions. The Hill function is among the best, although the curve with the best fit is of a 5PL function. Among other suggested models are log-logistic function, CDF of normal distribution, generalised logistic and Gompertz functions.

Simulating the scientist's final choice of model type based on background knowledge, we now focus on the function deemed most suitable: Fig. 7 shows curves (left panel) and the lack-of-fit residuals (right panel) for the ten best suggested solutions for the Hill function phenome. Corresponding parameter values can be found in Table 2.

It is apparent from Fig. 7 that the closest match to the new curve from the Hill function phenome is a very good fit, although the

Table 1
Parameters for the original curve (first line) followed by the distances and parameter estimates for the best solutions for Hill function. More densely sampled model Hill phenome.

#	m	Function	Formula, $F_m(x;\mathbf{p}_m) =$	${s_{e_{i,\ m}}}^2$	Score distance, 10^{-4}	Parameters				Preprocessing parameters	
						p_1	p_2	p_3	p_4	Slope	*offset*
1	18	Hyperbolic tangent	$\tanh(p_1x+p_2)$	$3.47e-09$	5.47	4.10	− 2.40			0.995	− 63.866
2	23	Error function	$erf(p_1x+p_2)$	$5.27e-09$	7.06	3.30	− 2.00			0.990	− 63.860
3	22	Gompertz function	$exp(p_1 \cdot exp(p_3-p_2x))$	$4.19e-08$	7.05	− 2.00	5.10	2.00		2.090	− 64.845
4	21	*Hillfunction*	$\frac{x^p}{x^p + \theta^p}$	$9.52e-08$	1.73	4.10	0.61			2.161	-64.845
5	32	CDF of normal distribution	$\frac{1}{2}\left(1 + erf\left(\frac{x-p_1}{\sqrt{2p_2^2}}\right)\right)$	$1.16e-07$	4.26	0.59	0.21			1.965	− 64.850
6	38	CDF of log-logistic distribution	$\frac{1}{1+\left(\frac{x}{p_1}\right)^{-p_2}}$	$1.60e-07$	5.74	0.61	4.10			2.161	− 64.850
7	14	Inverse tangent	$arctan(p_1x+p_2)$	$4.63e-07$	14.00	5.41	− 3.24			0.793	− 63.837
8	26	Generalised logistic function	$(1 + p_1 \cdot exp(p_2(p_3-x)))^{-\frac{1}{p_4}}$	$6.06e-07$	13.00	0.39	6.27	0.56	0.40	2.034	− 64.848
9	25	5PL function	$\frac{1}{1 + exp(-p_1x + p_2)}$	$1.17e-06$	0.66	0.50	− 3.00	1.80		2.361	− 64.845
10	24	Logistic function	$\frac{1}{\left(1+\left(\frac{x}{p_1}\right)^{p_2}\right)^{p_3}}$	$1.21e-06$	14.00	7.50	4.50			2.030	− 64.868
11	30	CDF of triangular distribution	$\begin{cases} \frac{x^2}{p_1}, & \text{for } x \le p_1 \\ 1-\frac{(1-x)^2}{1-p_1}, & \text{for } x > p_1 \end{cases}$	$1.44e-06$	160.00	0.71				1.910	− 64.845
12	27	CDF of beta distribution	$I_x(p_1, p_2)$	$3.63e-06$	18.00	8.10	6.10			1.910	− 64.845
13	29	CDF of Kumaraswamy distribution	$1-(1-x^{p_1})^{p_2}$	$3.82e-06$	8.59	3.10	3.10			1.910	− 64.845
14	37	CDF of gamma distribution	$\frac{\gamma\left(p_1, \frac{x}{p_2}\right)}{\Gamma(p_1)}$	$4.17e-05$	12.00	5.81	0.11			2.115	− 64.845
15	36	CDF of F-distribution	$I_{\frac{p_1x}{p_{1x}+p_2}}\left(\frac{p_1}{2}, \frac{p_2}{2}\right)$	$5.91e-05$	18.00	100.00	2.00			5.140	− 64.845

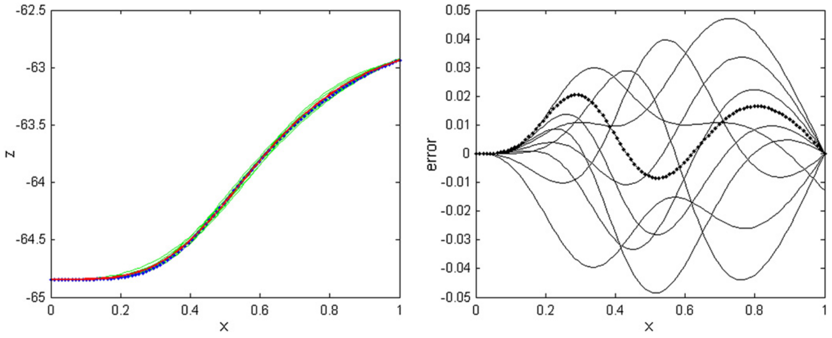


Fig. 6. The ten best fits to the new curve. The red curve is the new curve; green curves – the ten best solutions; blue dotted curve – the curve with the best fit; black dotted curve is the error of the best curve. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

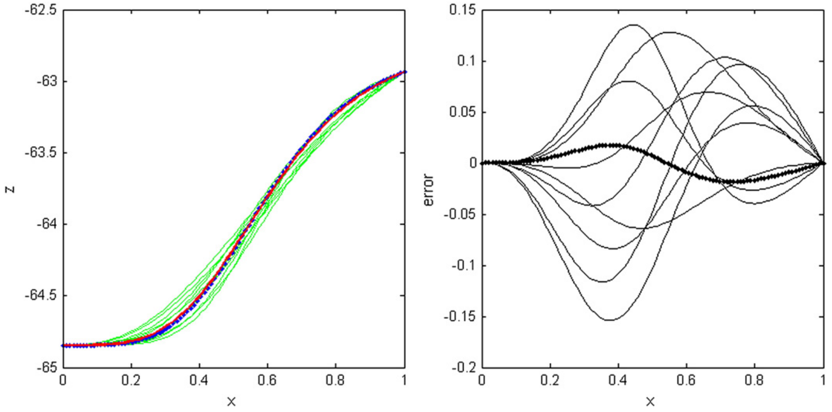


Fig. 7. Best solutions for the "unknown" Hill function: fits on the left panel and errors on the right panel. Red curve is the "unknown" curve; blue dotted line is the closest library curve in the Hill function phenome; black dotted line is the error of the best fit. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

estimation for parameter p should preferably be better:

	p	θ	slope	offset
true "unknown" curve(Hill)	**3.87**	**0.62**	**2.21**	**−64.845**
estimated curve (Hill)	4.1 ± 0.5	0.61 ± 0.05	2.16	−64.845

This lack of fit comes most likely from the fact that the model phenome was not sampled densely enough. To check whether this assumption is correct, it was decided to sample the parameter space more densely (198,740 curves instead of 1224) and fit a model to the curve again. This resulted in Fig. 8 and the best solution is:

	p	θ	slope	offset
true ″unknown″ curve	**3.87**	**0.62**	**2.21**	**−64.845**
estimated curve	3.85 ± 0.05	0.62 ± 0.01	2.2129	−64.845

Table 2
Adequate models for the simulated Hill function curve and their best corresponding parameter estimates. For notation of functions used here see Appendix A.

#	Score distance, 10^{-4}	p	θ	slope	offset
	True parameters (Hill)	3.87	0.62	2.21	− 64.845
1	1.73	4.1	0.61	2.16	− 64.845
2	5.20	3.6	0.61	2.23	− 64.845
3	5.63	3.6	0.66	2.34	− 64.845
4	5.82	4.6	0.61	2.11	− 64.845
5	6.52	3.1	0.66	2.44	− 64.845
6	8.34	3.1	0.71	2.57	− 64.845
7	9.68	5.1	0.61	2.06	− 64.845
8	11.00	2.6	0.76	2.85	− 64.845
9	11.00	4.1	0.66	2.26	− 64.845
10	12.00	2.6	0.71	2.69	− 64.845

Now, even after overall sorting of fits, only Hill functions were represented in the top-ten list of the best fits. The parameters were estimated much more precisely in comparison to the previous fit (see Table 3 for the best estimates for the Hill function curves). This means that in order to get better fits it is necessary either to sample the parameter space more densely, or to use a better method for finding the best fit than simply choosing the closest curve.

4. Discussion

In this paper a new approach, the DLU method, is presented for mathematical modelling of a given phenomenon (like curvature) in terms of a best function followed by a number of alternative plausible nonlinear models. A new and simpler method for fitting a nonlinear model or function to data is, thereby, obtained. An example with an

Please cite this article as: J. Isaeva, et al., Nonlinear modelling of curvature by bi-linear metamodelling, Chemometr. Intell. Lab. Syst. (2011), doi:10.1016/j.chemolab.2011.04.010

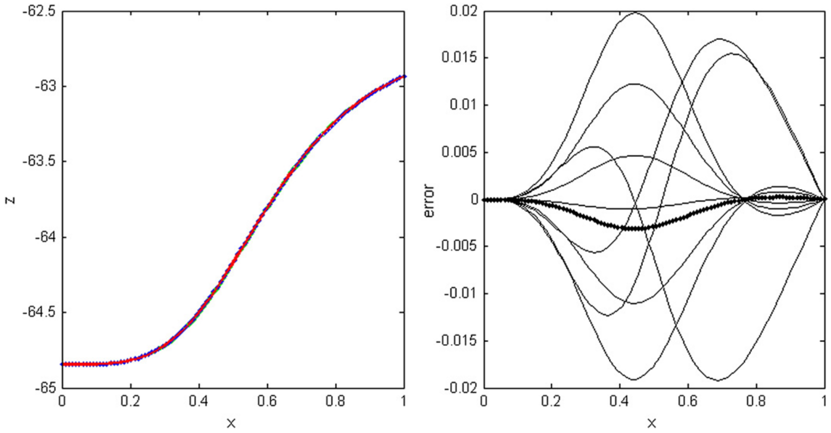


Fig. 8. Case with the more densed Hill function phenome. Best ten estimates for the simulated curve (left panel) and error of estimation (right panel). Red curve is the original curve. Dotted curve on the right panel is the error of the best fit. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

arbitrarily chosen sigmoidal curve was given as a proof of principle. This new curve was projected onto the data base with 38 function phenomes in order to identify a likely function type, along with a parameter set that describes the curve the best.

The model phenome described above is a collection of curves corresponding to a mathematical function. A comprehensive collection of model phenomes of a comprehensive set of functions describing a given phenomenon constitutes its modelome. The present preliminary modelome of line curvature can probably describe many processes in biology, chemistry and other fields of science. With a wide range of parameters the phenome captures a large variation of curves, from straight lines to approximate step functions. Each function phenome was sampled rather densely in order to provide reasonable estimation of parameters by simple look-up, without the need of local interpolation.

The example given in Section 3 showed that the method works well on noiseless data, predicting correctly the function type and parameter values. Since the look-up method also may give a list of alternative well-fitting functions, a scientist always has a choice to which model to select. Of course, this might again lead to subjectivity, but it might also open up for new models to be considered.

The library look-up method is, as far as we know, a new approach and is at an early stage of its development, so it needs further improvement. First of all, only an example using noiseless data has been given in this paper. Real data typically have noise from various sources, e.g., sampling errors, human errors, and other uncontrollable sources of variation. The DLU approach, therefore, has to deal with noise (possibly heteroscedastic) in order to be an alternative to the statistical approaches. Homoscedastic noise (constant variability across time) is not very critical. The look-up approach will still find a best fit in the library. However, it is beneficial to add a smoothing step (e.g., lowess smoothing) preceding the other preprocessing steps in order to reduce noise influence on the preprocessing variables and to improve the fit. Sometimes the noise is heteroscedastic and varying over time or with the level of the dependent variable. In such cases some parts of the data will be more informative than other parts. One way to deal with this is to define a vector of weights emphasising which parts of the curve that carry important information and which parts that are more noisy. This is an expert-opinion type of information, which the experimentalist often possesses. He or she may say in advance what level of noise is non-significant and can be allowed to be neglected. The weight vector may be used to modify the computation of score distance before look-up in the library [13].

Table 3
Distances and parameter estimates for the best solutions for the Hill function. The first line is the new “unknown” curve.

#	Score distance, 10^{-6}	p	θ	slope	offset
	True parameters (Hill)	3.87	0.62	2.21	−64.845
1	2.13	3.85	0.62	2.21	−64.845
2	3.16	3.90	0.62	2.21	−64.845
3	7.55	3.80	0.62	2.22	−64.845
4	8.34	3.95	0.62	2.20	−64.845
5	12.14	3.75	0.63	2.25	−64.845
6	12.41	3.70	0.63	2.26	−64.845
7	13.09	3.75	0.62	2.23	−64.845
8	13.41	4.00	0.62	2.19	−64.845
9	13.62	4.00	0.61	2.17	−64.845
10	13.87	4.05	0.61	2.17	−64.845

Another problem arises from the fact that in real life there are not so many processes that are described by a simple monotonous function, but rather a sum or even a product of such. In order to handle this in our proposed framework, high-level modelling (e.g., ANOVA, GAM or PARAFAC) may be required.

There is also an option for abscissa-preprocessing that consists in shifting curves along the *x*-axis. An example is to make each curve so that it passes through the point (0.5;0.5). This gives a further reduction of complexity, i.e., less number of components are needed at the step of PCA compression and less space is required to store the data.

As it was mentioned in Section 3, choosing the nearest neighbour curve works perfect in case when function phenomes are sufficiently densely sampled. However, the more curves there are in the library, the more time it takes to fit a model in PCA space. Hence, it may be preferably to use another technique for finding a parameter set that fits best to the original curve rather than storing more information in the library. This can be done by applying interpolation between, e.g., ten nearest neighbours. In our case, a simple averaging over the ten best curves did not give an improved fit. A more flexible approach, the HPLS (Hierarchical Partial Least Squares) (K. Tøndel and H. Martens, submitted, 2010), could be useful to predict parameter values for the new curve. This HPLS model, thus, serves as a metamodel connecting

Please cite this article as: J. Isaeva, et al., Nonlinear modelling of curvature by bi-linear metamodelling, Chemometr. Intell. Lab. Syst. (2011), doi:10.1016/j.chemolab.2011.04.010

score to parameter values and makes it possible to predict parameter values from scores as an alternative to using the average of the parameters of the nearest neighbours. Another possible solution of the longer performance problem with the more dense phenomes is to use the estimated parameter values from the "normal" modelome as starting values for a traditional approach, e.g., Hill climbing. Then the path to the true solution is much shorter; the chance of getting stuck in the local optima is lower, and, therefore, less time will be spent on the parameter estimation.

Statisticians may miss the usual significance tests possibilities and uncertainty measures of the parameter estimates in our proposed method. However, it should be straight forward to implement a parametric bootstrap routine to provide such statistics. Fitting, e.g., 1,000 bootstrap samples within the given function phenome (defined by the best fit) should be quick. The uncertainty of the parameters may then be computed directly from the bootstrap fits. It may be the case here that variability in the bootstrap samples is reduced because of the discrete sampling of the parameter space, but if an interpolation strategy between top fits or a metamodel approach is used, this should be a smaller problem.

Future application of the look-up approach will require that large sets of curves (thousands, millions) can be fitted quickly. Although it is relatively fast even at this stage, the look-up speed will be a topic for further development as we anticipate that the library will grow to incorporate more complex function types.

All weaknesses of the method and suggestions about its improvement mentioned above are being processed and some of the changes are shown in [13] (handling both homoscedastic and heteroscedastic noise, and working with large sets of data). Nevertheless, the database-method seems to work well even at this stage of development and can serve as a good alternative for finding model when having a time-series, stimulus–response or another type of data set.

Acknowledgements

Stig W. Omholt is thanked for constructive comments. This study was supported by the National Program for Research in Functional Genomics in Norway (FUGE) (RCN grant no. NFR151924/S10) and by the Norwegian eScience program (eVITA) (RCN grant no. NFR178901/V30).

Appendix A

List of functions used

$I_x(a;b)$	regularised incomplete beta function
$\Gamma(x)$	gamma function
$P(a;b)$	regularised gamma function
$\gamma(a;b)$	incomplete gamma function
$\mathrm{erf}(x)$	error function

Appendix B

Table B.1
All 38 models that are present at the current version of the library along with their parameter values range, number of curves in the model phenome, number of PC's. 95%, 99% and 99.9% confidence interval values are also given.

m	Function	Formula, $z=F_m(x;\mathbf{p}_m)$	NObj	p_1	p_2	p_3	p_4	PC's	$s^2_{95,\ m}$	$s^2_{99,\ m}$	$s^2_{999,\ m}$	$s^2_{max,\ m}$
1	Straight line	x	1					1	0	0	0	0
2	2nd degree polynomial	$(x+p_1)^2$	49	[0;2]				1	5.49e−032	6.64e−032	6.64e−032	6.64e−032
3	3rd degree polynomial	$(x+p_1)^3$	74	[0;2]				1	1.86e−007	5.04e−007	7.03e−007	7.03e−007
4	Inverse polynomial	$\frac{1}{x+p_1}$	47	[0.1;1]				2	1.54e−007	6.86e−007	6.86e−007	6.86e−007
5	inverse 2nd degree polynomial	$\frac{1}{(x+p_1)^2+p_2}$	935	[0;3]	[0.1;1]			2	7.09e−006	1.79e−005	4.58e−005	5.12e−005
6	Inverse 3rd degree polynomial	$\frac{1}{(x+p_1)^3+p_2}$	2,037	[0;2]	[0.1;1]			2	4.67e−005	1.17e−004	1.76e−004	1.77e−004
7		$\frac{x}{\sqrt{1+x^2}}$	1					1	x0	0	0	0
8	Sinus	$sin(p_1(\pi x-\frac{\pi}{2}))$	93	[0.01;1]				1	3.46e−008	1.21e−007	1.56e−007	1.56e−007
9	Cosinus	$cos(p_1\cdot\pi x)$	160	[−1;−0.01] [0.01;1]				1	1.77e−006	5.11e−006	6.87e−006	6.87e−006
10	Tangent	$tan(p_1\cdot\frac{\pi}{2}x)$	44	[0.01;0.5]				1	1.75e−007	5.55e−007	5.55e−007	5.55e−007
11	Cotangent	$cot(p_1\cdot\pi x)$	49	[0.01;0.5]				2	9.51e−006	3.22e−005	3.22e−005	3.22e−005
12	Inverse sinus	$arcsin(p_1(2x-1))$	42	[0.01;1]				2	5.90e−008	1.69e−007	1.69e−007	1.69e−007
13	Inverse cosinus	$arccos(p_1(2x-1))$	83	[0.01;1]				3	2.08e−008	2.54e−008	1.90e−007	1.890e−007
14	Inverse tangent	$arctan(p_1x+p_2)$	1,111	[0.01;10]	[−15;3]			5	2.02e−004	2.88e−004	3.20e−004	3.23e−004
15	Inverse cotangent	$\mathrm{arccot}(p_1x+p_2)$	164,000	[0.01;4]	[0;0.8] [1;4]			2	2.46e−006	1.43e−005	2.69e−005	2.70e−005
16	Hyperbolic sinus	$sinh(p_1x+p_2)$	397	[0.01;3]	[0;3]			1	2.74e−006	9.36e−006	1.67e−005	1.67e−005
17	Hyperbolic cosinus	$cosh(p_1x+p_2)$	110	[0.1;4]	[0;4]			2	1.54e−009	1.12e−008	1.13e−008	1.13e−008
18	Hyperbolic tangent	$tanh(p_1x+p_2)$	1,317	[0.1;15]	[−20;3]			8	2.88e−004	3.86e−004	4.67e−004	4.98e−004
19	Hyperbolic cotangent	$coth(p_1x+p_2)$	246	[0.01;1]	[0.01;0.1]			3	1.42e−006	3.30e−006	2.38e−005	2.38e−005
20	Michaelis–Mentenkinetics	$\frac{x}{x+0.01+p_1}$	101	$[10^{-6};10^{-3}]$				1	5.46e−011	1.90e−010	3.63e−010	3.63e−010
21	Hill function	$\frac{x^{p_1}}{x^{p_1}+p_2^{p_1}}$	1,224	[0.1;20]	[0.01;5]			11	4.09e−004	1.71e−004	2.26e−003	2.26e−003
22	Gompertz function	$exp(p_1\cdot exp(p_3-p_2x))$	1,021	[−10;−0.1]	[0.1;10]	[−10;3]		4	2.12e−004	4.06e−004	6.26e−004	1.06e−003
23	Error function	$\frac{2}{\sqrt{\pi}}\int_0^{p_1x+p_2}e^{-t^2}dt$	1,237	[0.5;10]	[−5;3]			6	1.92e−004	2.93e−004	4.02e−004	4.28e−004
24	Logistic function	$\frac{1}{1+exp(-p_1x+p_2)}$	654	[0.1;10]	[−6;5]			3	1.78e−004	2.90e004	3.58e−004	3.98e−004

Please cite this article as: J. Isaeva, et al., Nonlinear modelling of curvature by bi-linear metamodelling, Chemometr. Intell. Lab. Syst. (2011), doi:10.1016/j.chemolab.2011.04.010

Table B.1 (*continued*)

m	Function	Formula, $z = F_m(x;\mathbf{p}_m)$	NObj	p_1	p_2	p_3	p_4	PC's	$s^2_{95,\,m}$	$s^2_{99,\,m}$	$s^2_{999,\,m}$	$s^2_{max,\,m}$
25	5PL function	$\frac{1}{\left(1+\left(\frac{x}{p_1}\right)^{p_2}\right)^{p_3}}$	6,342	[0.05;0.9]	[−25;−2]	[0.3;1.8]		12	4.17e−004	6.64e−004	9.28e−004	2.13e−003
26	Generalised logistic function	$(1+p_1 \cdot exp(p_2(p_3-x)))^{-\frac{1}{p_4}}$	253	[0.3;1]	[3;10]	[0.05;0.9]	[0.1;1]	4	1.83e−004	3.87e−004	9.32e−004	9.32e−004
27	CDF of Beta distribution	$I_x(p_1, p_2)$	1,186	[0.1;100]	[0.1;100]			12	2.60e−004	5.24e−004	1.91e−003	7.59e−003
28	CDF of Student's t-distribution	$\frac{1}{2} \; + x \cdot \Gamma\left(\frac{\nu+1}{2}\right)$	106	[0.1;1.5] [2;40]				2	2.65e−007	3.00e−007	1.00e−006	1.00e−006
29	CDF of Kumaraswamy distribution	$1-(1-x^{p_1})^{p_2}$	1,198	[0.1;5] [6;50]	[0.1;25]			8	1.86e−004	6.34e−004	9.04e−004	9.29e−004
30	CDF of triangular distribution	$\begin{cases} \frac{x^2}{p_1}, & \text{for } x \le p_1 \\ 1-\frac{(1-x)^2}{1-p_1}, & \text{for } x \; > p_1 \end{cases}$	91	[0.1;1]				4	1.75e−005	4.65e−005	5.24e−005	5.24e−005
31	CDF of U-quadratic distribution	$4 \cdot ((x-0.5)^3 + 11.5^2)$	1					1	0	0	0	0
32	CDF of normal distribution	$\frac{1}{2}\left(1+erf\left(\frac{x-p_1}{\sqrt{2p_2^2}}\right)\right)$	481	[0.01;0.99]	[0.01;0.5]			27	5.34e−004	6.49e−004	1.14e−003	1.14e−003
33	CDF of Chi-square distribution	$\frac{1}{\Gamma\left(\frac{p_1}{2}\right)} \gamma\left(\frac{p_1}{2}, \frac{x}{2}\right)$	4	[1;10]				2	3.28e−006	3.28e−006	3.28e−006	3.28e−006
34	CDF of Chi distribution	$P\left(\frac{k}{2}, \frac{x^2}{2}\right)$	81	[0.5;15]				3	5.66e−005	8.50e−005	2.23e−003	2.23e−003
35	CDF of exponential distribution	$1-exp(-p_1 x)$	97	[10^{-2};10^1]				3	6.40e−006	1.27e−005	2.57e−005	2.57e−005
36	CDF of F-distribution	$I_{\frac{p_1 x}{p_{1x}+p_2}}\left(\frac{p_1}{2}, \frac{p_2}{2}\right)$	10,000	[1;100]	[1;100]			3	5.31e−005	1.56e−004	2.03e−004	2.63e−004
37	CDF of Gamma distribution	$\frac{\gamma\left(p_1, \frac{x}{p_2}\right)}{\Gamma(p_1)}$	352	[0.01;9]	[0.01;2.2]			4	1.36e−004	3.21e−004	4.83e−004	4.83e−004
38	CDF of log-logistic distribution	$\frac{1}{\left(1+\left(\frac{x}{p_1}\right)^{-p_2}\right)}$	215	[0.01;1] [2;3]	[0.1;10]			7	3.21e−004	6.09e−004	7.96e−004	7.96e−004

References

[1] D.C. Montgomery, E.A. Peck, Introduction to Linear Regression Analysis, 2nd edition Wiley, New York, USA, 1992.

[2] H. Martens, T. Naes, Multivariate Calibration, John Wiley & Sons Inc, 1989.

[3] J.R. Binder, S.M. Rao, T.A. Hammeke, J.A. Frost, P.A. Bandettini, J.S. Hyde, Effects of stimulus rate on signal response during functional magnetic resonance imaging of auditory cortex, Cognitive Brain Research 2 (1994) 31–38.

[4] D.J. Tolhurst, J.A. Movshon, I.D. Thompson, The dependence of response amplitude and variance of cat visual cortical neurones on stimulus contrast, Experimental Brain Research 41 (1981) 414–419.

[5] J. Warringer, D. Anevski, B. Liu, A. Blomberg, Chemogenetic fingerprinting by analysis of cellular growth dynamics, BMC Chemical Biology 8 (2008) 3–12.

[6] J.I. Steinfeld, J.S. Francisco, W.L. Hase, Chemical Kinetics and Dynamics, Prentice Hall Englewood Cliffs, New Jersey, 1989.

[7] F. Jacob, J. Monod, Genetic regulatory mechanisms in the synthesis of proteins+, Journal of Molecular Biology 3 (3) (1961) 318–356.

[8] M. Evans, N. Hastings, B. Peacock, Statistical Distributions, volume 12, 3rd edition IOP Publishing, 2001.

[9] D.J. Currie, Estimating Michaelis–Menten parameters: bias, variance and experimental design, Biometrics 38 (4) (1982) 907–919.

[10] J.C. Lagarias, J.A. Reeds, M.H. Wright, P.E. Wright, Convergence properties of the Nelder–Mead simplex algorithm in low dimensions, SIAM Journal of Optimization 9 (1996) 112–147.

[11] H.H. Rosenbrock, An automatic method for finding the greatest or least value of a function, The Computer Journal 3 (3) (1960) 175.

[12] K.E. Atkinson, An Introduction to Numerical Analysis, 2nd edition John Wiley and Sons, 1988.

[13] J. Isaeva, S. Sæbø, J.A. Wyller, S. Nhek, H. Martens, Fast and comprehensive fitting of complex mathematical models to massive amounts of empirical data, J. Chemometrics and Intelligent Laboratory Systems (2011), doi:10.1016/j.chemolab.2011.04.010.

[14] B. Becher, A.K. Knöfel, J. Peters, Time-based analysis of silver-stained proteins in acrylamide gels, Electrophoresis 27 (10) (2006) 1867–1873.

[15] P.G. Gottschalk, J.R. Dunn, The five-parameter logistic: a characterization and comparison with the four-parameter logistic, Analytical Biochemistry 343 (2005) 54–65.

[16] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, et al., Equation of state calculations by fast computing machines, The Journal of Chemical Physics 21 (6) (1953) 1087.

[17] W.K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika 57 (1) (1970) 97.

[18] A.E. Gelfand, A.F.M. Smith, Sampling-based approaches to calculating marginal densities, Journal of the American Statistical Associationt 85 (410) (1990) 398–409.

[19] M.K. Cowles, B.P. Carlin, Markov Chain Monte Carlo convergence diagnostics: a comparative review, Journal of the American Statistical Association 91 (434) (1996).

[20] H. Martens, I. Måge, K. Tøndel, J. Isaeva, A. Gjuvsland, M. Høy, S. Sæbø, Multi-level binary replacement (MBR) design for computer experiments in high-dimensional nonlinear systems, J. Chemometrics 24 (2010) 748–756.

[21] K. Tøndel, A. Gjuvsland, I. Måge, H. Martens, Screening design for computer experiments: metamodelling of a deterministic mathematical model of the mammalian circadian clock, J. Chemometrics 24 (2010) 738–747.

[22] S. Wold, M. Sjöström, SIMCA: a method for analyzing chemical data in terms of similarity and analogy, chapter 13, American Chemical Society Symposium Series 52, American Chemical Society, Wash., D.C, 1977, pp. 243–282.

[23] A. Kohler, U. Böcker, J. Warringer, A. Blomberg, S.W. Omholt, E. Stark, H. Martens, Reducing inter-replicate variation in Fourier transform infrared spectroscopy by extended multiplicative signal correction, Applied Spectroscopy 63 (3) (2009) 296–305.

[24] A. Kohler, M. Zimonja, V. Segtnan, H. Martens, Standard normal variate, Multiplicative Signal Correction Preprocessing in Biospectroscopy, volume 2, chapter 2.09, Elseveier, 2009, pp. 139–163.

[25] C.A. de Boor, A Practical Guide to Splines, Springer-Verlag, New York, 1978.

Paper III

Chemometrics and Intelligent Laboratory Systems xxx (2011) xxx–xxx

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemolab

Fast and comprehensive fitting of complex mathematical models to massive amounts of empirical data

Julia Isaeva [a,*], Solve Sæbo [a], John A. Wyller [b,c], Sarin Nhek [d], Harald Martens [b]

[a] Department of Chemistry, Biotechnology and Food Science, Norwegian University of Life Sciences, P.B. 5003, N-1432 Ås, Norway
[b] Centre for Integrative Genetics (CIGENE)/IMT, Norwegian University of Life Sciences, N-1432 Ås, Norway
[c] School of Mathematical Sciences, University of Nottingham, NG7 2RD, UK
[d] Nofima Mat AS, Osloveien 1, NO-1430 Ås, Norway

ARTICLE INFO

Article history:
Received 22 December 2010
Received in revised form 12 April 2011
Accepted 19 April 2011
Available online xxxx

Keywords:
Modelling
Heteroscedastic noise
Direct look-up
Iterative least squares
Metamodelling

ABSTRACT

The new method for parameterising a high number of observed curves in terms of nonlinear functions, presented by Isaeva et al. is here applied to noisy data and tested with respect to computational speed, ease of use and estimation precision. The method employs conventional least squares minimisation of the lack-of-fit residuals. But algorithmically it replaces traditional, time-consuming iterative hill-climbing (e.g., simplex optimisation) by a fast, non-iterative linear projection. Each nonlinear function is emulated by its multivariate metamodel (a low-dimensional bi-linear principal component analysis model of its behaviour), and yields parameter estimates by a simple projection plus a data base look-up.

For setting up a generic, fast modelling system for line curvature, a set of 38 widely different mathematical functions - most of them nonlinear - were selected for their ability to give sigmoid curves. For each model, its behavioural repertoire was established by designed computer simulation, and its multivariate metamodel was estimated. Then the new curve fitting approach was compared to conventional simplex optimisation, by fitting artificial, but noisy curves to the 38 curve-functions, in order to identify the correct function type and parameter values. Finally, the new method was adapted to heteroscedastic noise and employed for parameterisation of >170,000 sigmoid curves from time lapse monitoring of proteomic 2D Gel Electrophoresis (2DGE) image development.

The new method gave at least as precise parameter estimates as the simplex optimisation and worked well both for homoscedastic and heteroscedastic noise. It speeded up the parameter estimation in the nonlinear models by a factor of about 24 compared to the simplex optimisation.

Moreover, per definition it avoids the problems of having to select starting values and ending up in locally optimal solutions. And it reduced the problem of subjective, possibly erroneous choice of nonlinear model specification.

1. Introduction

Isaeva et al. [1] introduced a new method for fitting one or more nonlinear mathematical models to a large number of observed data, e.g., growth curves. The new method is based on an extension of a bi-linear modelling method first presented by Kohler et al. [2] for spectroscopic data and by Kohler et al. [3] for mass spectroscopic data. It consists in, first, preparing simple bi-linear metamodels that represent the individual, potentially relevant nonlinear models, and then fitting these metamodels to the observed input data. For each input curve, this fast, linear fitting and look-up in the metamodel reveals which of the nonlinear models are most plausible, and yields estimates of their model parameters and their uncertainties.

The metamodels are developed in the following way: for a given problem type (e.g., sigmoid growth curves), a set of nonlinear functions (e.g., a Hill function, a logistic curve and a cumulative normal distribution) are chosen by a user as potentially relevant. For each of these nonlinear functions, its so-called model phenome is established, once and for all, in terms of a large data set that represents all of the model's relevant "phenotypes" (output curves), by extensive, statistically designed computer simulations. This large set of curves is then preprocessed and compressed, e.g., by a singular value decomposition, into a bi-linear multivariate metamodel representing the nonlinear function.

Then, when a new set of measured curves is to be parameterised, each curve is fitted to the bi-linear metamodels for each of the potentially relevant nonlinear models. From the lack-of-fit to the metamodels, the models that fit that curve sufficiently well are found. Its parameter values for each of these plausible models are estimated by a look-up or a local interpolation within the bi-linear metamodel subspace. Each measured curve is, thus, emulated by the simulation curve in the model phenome data base that looks most like it. If several nonlinear models can generate curves that fit the empirical curve equally well, then it is up to the user to make the final choice of a

* Corresponding author. Tel.: +47 6496 6243.
E-mail address: julia.isaeva@umb.no (J. Isaeva).

0169-7439/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.chemolab.2011.04.009

model. This allows the user to discover unexpected modelling opportunities. But the choice can also be automatic, based on the user's prior specification of preference among the potential models.

The approach was shown by Isaeva et al. [1] to have several potential benefits over classical iterative estimation methods, in the way it can reduce computation time, modelling subjectivity and the risk of ending up in local optima. The look-up method was illustrated on a simple example of a sigmoid function (a Hill function) with noise-free data.

In the present paper we turn to more real-world problems where data are affected by noise and where the number of curves is very large. This introduces the need for some modifications to the basic method described by Isaeva et al. [1]. These will here be described and assessed on artificial data with known structure before they are applied to a large set of nonlinear time series—in this case 5-parameter logistic (5PL) curves.

Efficiency of the method will here also be tested on a real data set with a large number of curves in it highly affected by noise.

Massive sets of empirical curves are to be fitted to nonlinear models: the curves come from gel development in 2-Dimensional Gel Electrophoresis (2DGE). 2DGE as such is a standard method in proteomics, in which a number of different known or unidentified proteins, present in a mixture at low, but different and unknown concentrations, can be separated from each other in terms of their molecular charge and mass. The proteins are then revealed as spots in a two-dimensional image by a sensitive staining technique, and from the volume under the spots, the individual proteins are quantified and sometimes even identified. The use of silver nitrate for the staining of the protein spots in 2DGE, followed by photographic scanning the stained gel, is a standard procedure. Traditionally this has required that the staining development process is stopped at some point in time, after which the gel is scanned and represented a digital image.

This stopping is suboptimal because it is difficult to find a good compromise between over-developing (saturating) the proteins present at high concentrations and under-developing (being insensitive to) the proteins present at low concentrations in the mixture. Grove et al. [4] overcame this dilemma by time lapse photography, recording the temporal development process as a continuous video. For each pixel in the video image, a time series "growth curve" is thus obtained. After conversion of the raw image data to absorbance, the maximum slope of this curve was taken as a measure of the protein concentration.

However, it was later discovered that the colour development process for different proteins display qualitatively different "growth curves". Hence, we decided to parameterise each of the growth curves by a nonlinear function. In the present case just one of the 3 RGB colour camera channels, for just one single 2DGE gel, yielded several hundred thousand growth curves. To parameterise all of these in terms of a truly nonlinear function is a computational challenge.

In Section 2 we describe the methodological modifications of the direct look-up (DLU) method. Then in Section 3 we describe the data that have been used in the present article to demonstrate performance of the DLU. In Section 4.1 we compare the DLU method with Iterative Least Squares estimation (ILS) [5] for simulated data. In Section 4.2 we apply the data base approach to find a model for a very large real 2DGE data set. The article is closed by discussing the results and making some conclusions in Section 5.

2. Methods

2.1. Summary of the metamodelling method

Curve generating functions $m=1, 2, \ldots, 38$ were collected from different fields of science, and for each of them extensive simulations were performed with various parameter combinations:

$$z_{j,m} = F_m(x; \mathbf{p}_m), \tag{1}$$

where $j=1, 2, \ldots, N_m$ is the index of a simulation for model m, $m=1, 2, \ldots, 38$; F_m is the functions in the data base from Ref. [1] defined at $K=100$ x-values (e.g., time points) on the interval $x \in [0.001; 1]$.

These curves were then preprocessed to make them compatible between each other—the ordinates of the curves were forced to be between 0 and 1:

$$y_{j,m} = \frac{z_{j,m} - \mathrm{off}_{j,m}}{\mathrm{sl}_{j,m}}, \tag{2}$$

where $\mathrm{off}_{j,m}$ and $\mathrm{sl}_{j,m}$ are offset and slope parameters such that $y_{j,m}(0.001)=0$ and $y_{j,m}(1)=1$.

The table of simulated curve data $Y_m(N_m \times K)$ for each function was decided to be stored in a compressed way as a bi-linear principal component analysis (PCA) model

$$Y_m = \bar{y}_m + T_m V'_m + E_m, \tag{3}$$

with scores (T_m), loadings (V_m) and residuals E_m. The number of principal components (PCs) was chosen so that at least 99.9% of variance in Y_m was explained.

Then, when new curves $z_i^{\mathrm{obs}}(x)$, $i=1, 2, \ldots, N$, are obtained, they are first preprocessed in the same way as curves in the data base:

$$y_i^{\mathrm{obs}} = \frac{z_i^{\mathrm{obs}} - \mathrm{off}_i^{\mathrm{obs}}}{\mathrm{sl}_i^{\mathrm{obs}}}, \tag{4}$$

where $z_i^{\mathrm{obs}}(x)$ are new curves. Afterwards, when the new data are compatible with the curves in the data base (are defined on [0.001;1], have values between 0 and 1, and are increasing), they are to be projected onto loadings in every relevant function phenome m in order to find a score vector and a residual term for the new curves:

$$t_{i,m} = \left(y_i^{\mathrm{obs}} - \bar{y}_m\right) \cdot V_m, \tag{5}$$

$$e_{i,m} = \left(y_i^{\mathrm{obs}} - \bar{y}_m\right) - t_{i,m} V'_m. \tag{6}$$

Then, for every simulated curve i, the residual term gives a residual variance, i.e., distance from the curve i to every model m in the data base:

$$s^2_{e_{i,m}} = \frac{e_{i,m} \cdot e'_{i,m}}{K}, \tag{7}$$

where K is the number of observation points.

These $s^2_{e_{i,m}}$ values determine which models out of 38 present in the data base that have adequate fit to this curve i. It is defined by comparing them with the values for 99% confidence intervals for corresponding models (see [1]). Models with $s^2_{e_{i,m}} < s^2_{99,m}$ are considered as a good fit and are taken into account when estimating parameters.

To find $\hat{\mathrm{p}}_{i,m}$, the unknown parameter values of the new curve, its distances to each individual curve in the phenomes of plausible models are computed:

$$s_{t_{i,j,m}} = \sqrt{\left(t_{i,m} - t_{j,m}\right)\left(t_{i,m} - t_{j,m}\right)'}. \tag{8}$$

The closest fits are found by sorting $s_{t_{i,j,m}}$ over all good functional forms and taking the ten best with the smallest $s_{t_{i,j,m}}$. Here we then employ a simple direct look-up for parameters that correspond to the found curves.

Please cite this article as: J. Isaeva, et al., Fast and comprehensive fitting of complex mathematical models to massive amounts of empirical data, Chemometr. Intell. Lab. Syst. (2011), doi:10.1016/j.chemolab.2011.04.009

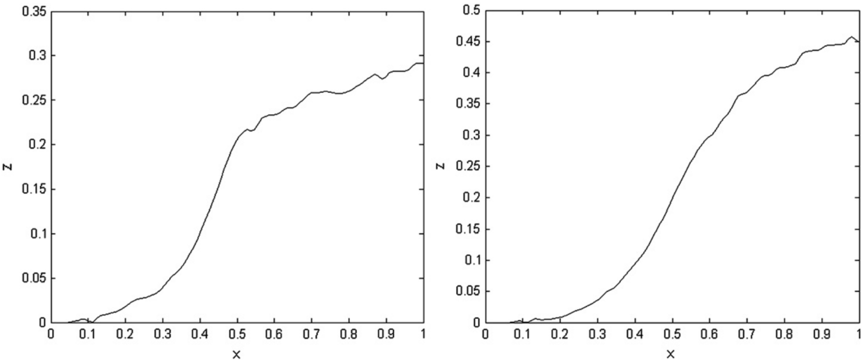


Fig. 1. Two examples of curves from 2D electrophoresis data. The curve on the left becomes noticeably noisy approximately after $x = 0.52$, whereas for the curve on the right hand side, the section part after $x = 0.73$ might be considered more noisy. (Note that here the curves are not yet preprocessed.)

For estimation of offset and slope parameters one has to apply the formulae:

$$\hat{a}_i = \text{off}_i^{\text{obs}} - \text{off}_{j,m} \cdot \frac{\text{sl}_i^{\text{obs}}}{\text{sl}_{j,m}}, \tag{9}$$

$$\hat{b}_i = \frac{\text{sl}_i^{\text{obs}}}{\text{sl}_{j,m}}. \tag{10}$$

Thus, the final estimate of the curve is obtained:

$$\hat{z}_i^{\text{obs}}(x) = \hat{a}_i + \hat{b}_i \cdot F_m(x; \hat{\mathrm{p}}_i). \tag{11}$$

For further details on the DLU method with the steps involved in making the data base of curves and estimating parameters of new curves see Ref. [1].

2.2. Method improvements

The methodology described in Isaeva et al. [1] was improved in the following way: for the speed-up needed for parameterising large sets of curves, the code was rewritten in such a way that number of "for" loops and calls for disc operation (including swapping due to memory problems) was reduced. A test-set with 1000 curves (noise-free) was used to check time performance of the program. The process of finding a model for all the curves took now approximately 30 seconds against almost 600 with the former version, used in Ref. [1], that is time was reduced by factor of 20.

2.3. Handling different noise structures in the input curves

In Ref. [1] it was demonstrated only how DLU works for noise-free data. During the simulations, no noise is expected in the obtained curve vectors z_j, $j = 1, 2, \ldots, N_m$, $m = 1, 2, \ldots, 38$, that are to be stored in the model phenome data base. So no special statistical precaution is needed when compressing the data base into the bi-linear metamodel. The lack-of-fit residuals in the metamodel generation are only due to minor nonlinearities left unmodelled. Even though the data themselves are error-free and deterministic, the presence and nature of these small residuals appear to us as if they are random.

However, real measured curve vectors z_i^{obs}, $i = 1, 2, \ldots, N$, are typically additionally affected by noise of some sort. Therefore, a few modifications to the data base look-up approach were introduced so that even noisy data may be fitted.

First of all, some degree of pre-smoothing of z_i^{obs} may be advantageous in order to facilitate the subsequent estimation of the preprocessing parameters. Secondly, the fitting of the preprocessed, but noisy curves y_i^{obs} to a metamodel V_m may require special attention. In case of homoscedastic noise, where the noise is expected to be independent and identically distributed across all time points, all parts of the curve are expected to carry equal amounts of information. In these cases no major modifications to the unweighted least squares

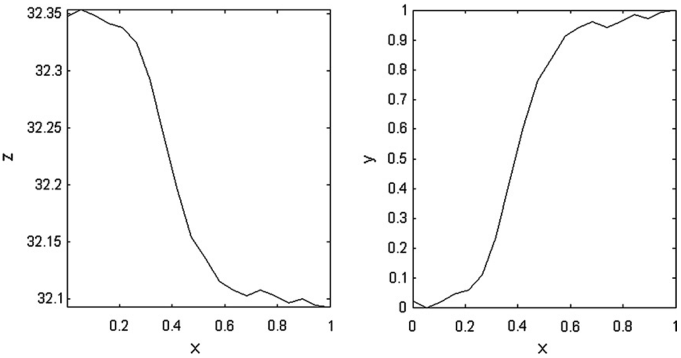


Fig. 2. An example of a simulated five-parameter logistic curve: before (left) and after (right) preprocessing.

Please cite this article as: J. Isaeva, et al., Fast and comprehensive fitting of complex mathematical models to massive amounts of empirical data, Chemometr. Intell. Lab. Syst. (2011), doi:10.1016/j.chemolab.2011.04.009

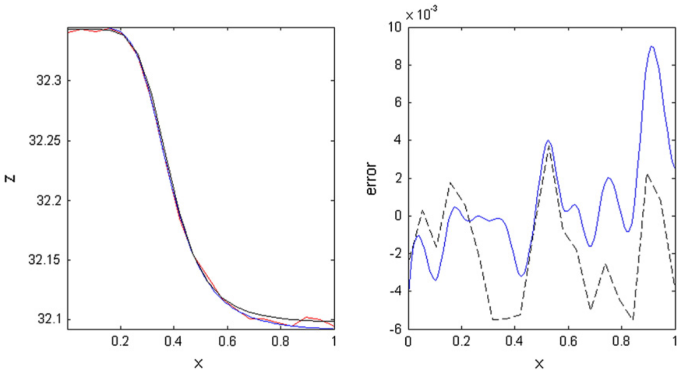


Fig. 3. The plot on the left panel shows an "unknown" curve (red) (a random curve from the set) and its fits from DLU (blue) and ILS (black). The lack of the fits can be seen on the right: blue solid line - for DLU and black dashed line - for ILS. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

method are needed. The look-up approach after the least squares projection of the preprocessed curves into the metamodel subspace will still find sensible estimates of the model parameters.

When it comes to heteroscedastic noise, where some parts of the data are known to be more affected by noise than other parts, more attention should be given to the most informative data. In this case weighted least squares (WLS) approach [6] is used, in order to emphasise those parts that are less effected by noise in the computation of scores and residuals. This is done by setting vectors of weights, w_i, of the same length as the data vector z_i^{obs}, which, for instance, may have ones as values at informative observation points and lower at more noisy points.

The scores for a new curve i (Eq. (5)) are now estimated by WLS regression on the loadings in the metamodel of model m over curve points $x = 0.001, \ldots, 1$:

$$t_{i,m} = \left(y_i^{obs} - \bar{y}_m\right) W_i V_m \left(V'_m W_i V_m\right)^{-1}, \tag{12}$$

where W_i is the diagonal matrix of weights w_i chosen for curve i. The residual lack-of-fit of curve i to metamodel m is estimated by Eq. (6) and the squared distance of curve i to metamodel m is found by

$$s^2_{e_{i,m}} = \frac{e_{i,m}\cdot W_i \cdot e'_{i,m}}{1\cdot W_i \cdot 1'}. \tag{13}$$

The distance of curve i to each of the individual simulated curves $j = 1, 2, \ldots, N_m$ is computed in the following way:

$$s_{t_{i,j,m}} = \sqrt{\frac{\left(t_{i,m} - t_{j,m}\right)\left(V'_m W_i V_m\right)^{-1}\left(t_{i,m} - t_{j,m}\right)'}{1\cdot\left(V'_m W_i V_m\right)^{-1}\cdot 1'}}. \tag{14}$$

The best model fit to input curve z_i^{obs} would then be expected among the simulated curves with lowest $s_{t_{i,j,m}}$ in the metamodel(s) m with lowest $s^2_{e_{i,m}}$. Eqs. (9) and (10) are subsequently applied in order to remove the effect of the offset and slope preprocessing yielding the final fit of the input curve number i (Eq. (11)).

The WLS method in Eqs. (12)–(14) ensures that the least informative parts of the input curves will have minimal contribution to the distance measures used in the library look-up process for choosing relevant metamodels and also for choosing nearest neighbour(s) within a chosen metamodel m.

Fig. 1 shows two examples of noise-affected curves from a dynamic 2D electrophoresis experiment representing the growth or development of silver colour over time (see Section 4.2 for details about the data). The curve on the left is by the experimentalist judged to be highly informative from $x = 0$ to $x = 0.52$, after which the noise level increases, whereas the curve on the right is found to be informative until $x = 0.73$. In these examples the last part of the curves were the noisy parts, but, of course, the noise may equally well be in the beginning or in the middle of a curve.

When a curve is noise-free or the noise is homoscedastic, weight vectors are just set equal to ones.

3. Data

3.1. Simulated noisy curves

As test data with known structure, a random 5PL curve (see [7]) with 20 observation points was simulated:

$$z = \text{slope}\cdot F(x; p_1, p_2, p_3) + \text{offset} = \text{slope}\cdot \frac{1}{\left(1 + \left(\frac{x}{p_1}\right)^{p_2}\right)^{p_3}} + \text{offset}, \tag{15}$$

where x is a vector with 20 equally distributed values between 0.001 and 1 values and

$$\begin{aligned} p_1 &= 0.35 \\ p_2 &= -5 \\ p_3 &= 1.45 \\ \text{slope} &= -0.2593 \\ \text{offset} &= 32.3497. \end{aligned}$$

Table 1
Parameter estimates obtained from DLU and ILS for the curve in Fig. 3. Numbers in parentheses show preceding and following values in the function phenome for given parameter estimates.

	p_1	p_2	p_3	Slope	Offset
True "unknown" parameters	0.35	−5.24	1.45	−0.25	32.34
DLU	0.35	−5.00	1.45	−0.250	32.350
	(0.30; 0.40)	(−4; −6)	(1.35; 1.45)		
ILS	0.34	−5.03	1.61	−0.250	32.350

Table 2
Comparison of fitting time and estimation errors for DLU and ILS.

	DLU, 38 functions	DLU, 3 functions	ILS
Time, seconds	28.80	5.04	120.61
Estimation error for p_1	$4.75e-05$	$4.75e-05$	0.0100
Estimation error for p_2	0.0728	0.0728	29.7866
Estimation error for p_3	0.0085	0.0085	0.7074
Estimation error for slope	$4.92e-05$	$4.92e-05$	$5.12e-05$
Estimation error for offset	$7.27e-05$	$7.27e-05$	$7.23e-05$
Lack-of-fit	0.0034	0.0034	0.0299

Then, a set of $N = 1000$ noisy replicates of this curve was generated by adding random noise e_i to this curve:

$$z_i = \text{slope}\cdot f(x;\mathbf{p}) + \text{offset} + e_i, \tag{16}$$

where e_i is independent identically (uniformly) distributed random errors with standard deviation of 0.01. Hence, even the perfectly fit of such a curve to a nonlinear model (or its bi-linear metamodel) is expected to have a distance (Eq. (14)) of 0.01.

3.2. Growth curves from dynamic proteomic imaging

For the present paper, a rather complex bovine serum albumin (BSA) protein sample was run on a 2-DE mini gel (XCell SureLock Mini-Cell, Invitrogen). Proteins were focused in IPG 5–8 and separated on 12.5% SDS-PAGE in the second dimension. The developer was added to the gel while the white translucent plastic tray was standing on a light box (qug/a2sl, 3×18W, DW Viewboxes). Pictures of the developing gel were taken at 5-second intervals for up to 17 minutes, whereafter it was considered that no further changes occurred.

Images were recorded with a Canon EOS 40D using a 28 mm lens at aperture f 9 and a shutter speed 1/50 seconds, and were saved in 14 bit raw colour format, and then converted and cropped into TIFF format without compression using Photoshop CS. The red RGB channel readings for each pixel were transformed into transmittance T (division by the readings at time zero for that pixel), and converted to approximate absorbance:

$$A = \log_{10}(1/T).$$

In order to correct for slight motions of the gels relative to the camera during the development, the whole-gel affine motion estimation and compensation was applied. The 174216 pixels in the cropped time lapse image data were then taken as $N = 174216$ input time series or “growth curves”.

4. Results

4.1. Simulated data: comparison with conventional iterative hill-climbing estimation

To check if DLU works any better than other very well-known and widely used methods for fitting nonlinear models to data, it was decided to test the method on noisy, but artificially created data with a known structure, as described above, and compare the DLU results with those from a typical representative of more classical estimation procedure. The method chosen for comparison was an ILS optimisation using the SIMPLEX implementation in Matlab [8]. Hence, the approach searches for the function and parameters minimising the least squares criterion:

$$S_{i,m}^2 = \frac{1}{K}\sum_{k=1}^{K}\left[z_i^{\text{obs}}(x_k) - F_m(x_k;\mathbf{p}_m)\right]^2, \tag{17}$$

where z_i are observed data; F_m is a model of type m to be fitted and K is the number of observations.

The functions $S_{i,m}^2$ were, thus, minimised with simplex optimisation for each candidate model type (see [8]).

In order to check the speed and precision of estimation for both methods, the set of 1000 curves that were obtained by adding random homoscedastic noise to the original curve was considered (Fig. 2 on the left shows one representative from the set).

To assess if the DLU gives computational compaction over the ILS, the following steps were performed:

- ILS estimation was run for each of the three functions:

$$\text{the Hill function}\quad F(x;\mathbf{p}) = \frac{x^{p_1}}{x^{p_1} + p_2^{p_1}}, \tag{18}$$

$$\text{the logistic function}\quad F(x;\mathbf{p}) = \frac{1}{1 + exp(-p_1 x + p_2)}, \tag{19}$$

$$\text{and 5PL function}\quad F(x;\mathbf{p}) = \frac{1}{\left(1 + \left(\frac{x}{p_1}\right)^{p_2}\right)^{p_3}}. \tag{20}$$

These functions were chosen since they are the best known sigmoids and often used in biology to describe microbial growth [9] or biological regulation ([10–12]). They were sampled at 20 abscissa points.

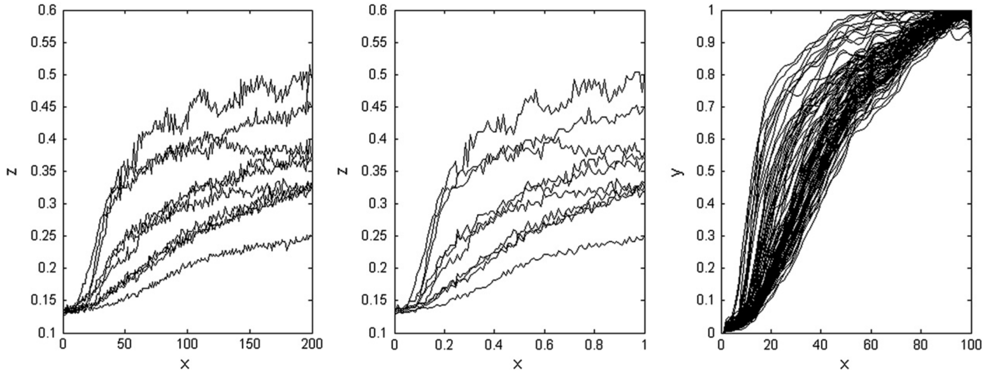


Fig. 4. Examples of curves from 2DGE data. On the left—ten randomly chosen original curves; in the middle—the same curves with only 100 observations considered shifted to the interval [0.001;1]; on the right—one hundred randomly chosen curves, preprocessed, smoothed and shifted to the interval [0.001;1].

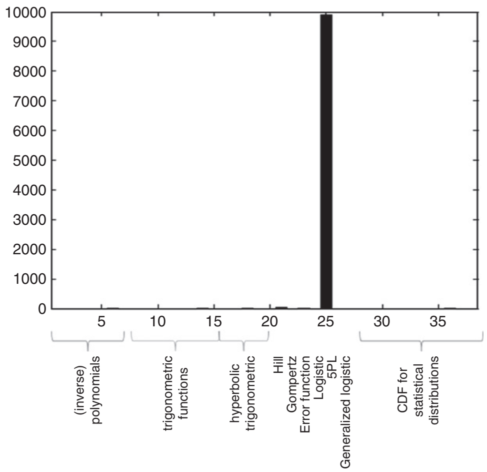


Fig. 5. Distribution of function types among fits to 2DGE data.

- Fitting with DLU was done twice: first checking all $M = 38$ available curvature models included in the "modelome of curvature" [1] (representing a situation when we do not know anything about the data and want to try many different mathematical models); and then considering only three functions out of 38 (the same as for ILS) in order to compare the time of fitting with such for ILS under similar conditions.
- A modification of DLU was used here which only searched for the single best solution instead of a list of alternatives. This was for the sake of having a fair comparison of fitting time for two different methods with similar outcomes.

To implement the look-up method here, the 20 time points of the abscissa x were first re-scaled and interpolated at 100 time points [13] in order to make them compatible with the curves in the library (i.e., to have 100 observation points in the [0.001;1] range). The interpolated curve ordinates z were preprocessed to ordinate y with a minimum of 0 and a maximum of 1. The preprocessed set of curve vectors $Y = [y_1, y_2, \ldots, y_{1000}]$ (Fig. 2 on the right) was projected onto the bi-linear metamodel of the model phenome of each of the potential nonlinear models.

Since the noise added to the curves was independent and identically distributed everywhere on the curves, the weights w were ignored because all the parts of the curves were considered equally important in terms of information.

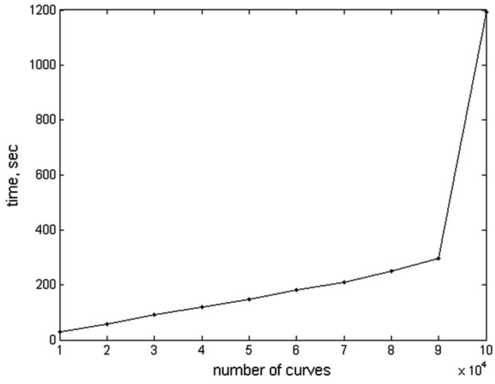


Fig. 6. Time spent by the look-up method to estimate sets of data of different sizes.

For each of the 1000 noise-contaminated replicates of the original curve, it was found – as expected – that the 5PL function gave the best bi-linear fit, showing that 5PL was the best nonlinear model of the noisy curves out of the 38 potentially relevant nonlinear functions.

The same procedure (projecting onto the library) was repeated again, but this time the search for the best solution was done only among three functions mentioned above. The result was, of course, expected to be absolutely the same as from within 38 models, but with correspondingly shorter estimation time.

For ILS estimation of parameters, there was no need for interpolation, and, therefore, curves with 20 observation points were used. The choice of functions to be fitted was due to the fact that we were aware of the type of the "unknown" curve (5PL). The other two functions were taken as the most typical alternative sigmoids.

The ILS parameter estimation encountered some problems when fitting these three models to the data. First of all, different sets of initial values for the parameters $\mathbf{p}_m$ sometimes led to different solutions and sometimes even implied divergence of solutions. Therefore, some of the termination conditions were relaxed, such as maximum number of iterations and function evaluations and termination tolerances with respect to function fit $(y_i - \hat{y}_i)$ and parameter estimate $(\hat{\mathrm{p}}_i)$ value. Given that all solutions converged. However, for the 5PL function, many of them had parameter estimates far from the true ones. This may either be due to the fact that the termination tolerances were not small enough, or that local minima were reached. Moreover, for approximately 30% of the curves the Simplex optimisation erroneously reported that the logistic function, rather than the 5PL, was the best model since it gave the smallest lack-of-fit in y:

$$S_i = \sqrt{\frac{1}{K}\sum_{k=1}^{K} \left[z_i^{\text{obs}}(x_k) - \hat{z}_i(x_k)\right]^2}. \tag{21}$$

Fig. 3 (left panel) shows an example of a curve from the set and its fits both from DLU and ILS estimation. Predicted curves are very close to the original one, and it is rather difficult to decide visually which fit is better. The right panel of Fig. 3 gives a slight notion that DLU provided with models with a smaller lack-of-fit, although it is not very clear. Therefore, let us compare parameter estimates (Table 1), estimation errors, lack-of-fit's and fitting time for all three cases (look-up with 38 and 3 functions and ILS) (Table 2).

For all practical purposes, the parameter estimates for this particular curve are equally good for both DLU and ILS, although with the experimental design presently used when establishing the 5PL model phenome, the DLU did estimate them slightly more correct. Nevertheless, estimation errors are dramatically different when calculated for the whole set of 1000 noisy curves (Table 2). For ILS they depend on whether optimisation function hits the desired global minimum or just a sub-optimal local minimum, whereas DLU always gives solutions close to the global minimum since its linear projection is non-iterative and, thus, requires no initial values and is only limited by the resolution of the experimental design behind the model phenome.

To get the average lack-of-fit for each method (DLU with 38 models, DLU with 3 models and ILS with three models), the minimal lack-of-fit's for each curve were taken:

$$S = \frac{1}{N}\sum_{i=1}^{N} S_i. \tag{22}$$

From Table 2 it can be noticed that the lack-of-fit, when using the DLU approach, is smaller than the one for the ILS approach. Besides, as it was mentioned before, in 282 cases out of 1000 logistic curve was found to fit the best. Therefore, the lack-of-fit becomes even larger when calculating it only for fits from 5PL function (0.0341).

Another important aspect of comparison of these two methods is the time that each of them needs to find an appropriate model and corresponding parameters. Fitting time for all three cases (DLU with 38 and 3 functions and ILS for 3 functions) are shown in Table 2. Of course, DLU needs considerable time to find the right function among all 38 possibilities and estimate the parameters in each, for all 1000 curves. But even this amount of time is much smaller than the time that ILS needs having only three functions. Further, if both methods are run under comparable conditions, with only three functions considered, the DLU wins against ILS in time with a huge difference: 5 seconds against 120 seconds.

Here each input curve-vector z_i^{obs} had only 20 observation points. Increasing this number implies increase of ILS working time (it took almost 167 seconds to fit a model to 1000 replicates of the curve with 100 observation points. In contrast, the DLU only used 5 seconds for the same curve resolution, see Table 2).

4.2. Real data

For an example of how the look-up method works on massive amounts of real data, dynamic developments of 2DGE data were taken. These growth curves represent time series for individual pixels in a high-resolution camera image, as described in Section 3.2.

The original data set consists of 174,216 curves of an unknown function with unknown parameters and 200 observation points. Left panel on Fig. 4 shows ten randomly chosen curve examples from the data set. To make them compatible with the curves in the data base, only $K = 100$ time points were employed, so only every second observation was taken into account, and abscissa was rescaled and shifted to the desired interval $x \in [0.001; 1]$ (Fig. 4 in the middle).

The rather large undulations in the uppermost curve is probably due to imperfect compensation of some small camera/gel motions, while the high-frequency noise is probably mostly due to random detector noise in the camera. Since the data were rather noisy, it was decided to smooth them first and then set up weight-vectors to apply WLS afterwards. Temporal smoothing was done by means of a twice moving average filtering with a half-width of three time points, to remove sharp features in the time series due to imperfect motion compensation and/or camera noise spikes (Fig. 4 on the right). It was noticed that in most of the cases the general noise level increased significantly once the absorbance ordinate reached above 0.3. In order to avoid having to redefine the projection matrices etc. (Eq. (12)) for each of many individual curves, a standard set of only ten weight vectors w_n, $n = 1, 2, \ldots 10$, were defined according to the formula:

$$w_n(x) = \begin{cases} 1 & \text{for } x < x_n \\ \dfrac{K-x}{K-x_n} & \text{for } x \geq x_n \end{cases}, \tag{23}$$

where x_n are predefined abscissa points dividing x-interval into ten equal parts. To determine which weight class a curve belongs to, one has to find an abscissa value of the curve corresponding to when it first time reaches 0.3 in its ordinate. The nearest on the left predefined x_n will indicate the weight class:

$$x_{n,i} = x\left(z_i^{\mathrm{obs}}(x) > 0.3\right).$$

These ten weight vectors were used then for generating ten weight versions of the projection matrices (Eq. (12)). When fitting the empirical curves, each weight version of the projection matrices was applied to the input curves classified into the corresponding weight group.

4.2.1. Finding a suitable nonlinear model

First it was decided to find out which model describes the data best. A sample of 1000 randomly selected curves was taken out of the whole set of 174216 curves and projected onto the modelome library of 38 curvature models. The ten best alternative solutions were saved for every curve and the respective function types were noted. Among the suggested functions were inverse polynomial of the third degree, inverse tangent, hyperbolic tangent, Hill function, error function and cumulative distribution function (CDF) of F-distribution. However, the histogram in Fig. 5 shows that the function type 25 (corresponding to 5PL function) clearly dominates among listed "best" models. Therefore, for the sake of saving time, only this function model was considered when estimating parameters for the whole set of curves. Unlike, e.g., the Hill function, the 5PL model cannot, to our knowledge, be linearised. Hence, to fit it to empirical data would traditionally require iterative nonlinear curve fitting.

4.2.2. Finding parameters

After the function type describing the data was found, each of the 174,216 curves was projected onto the model phenome data base of the 5PL model by WLS, and the parameter values for each of them

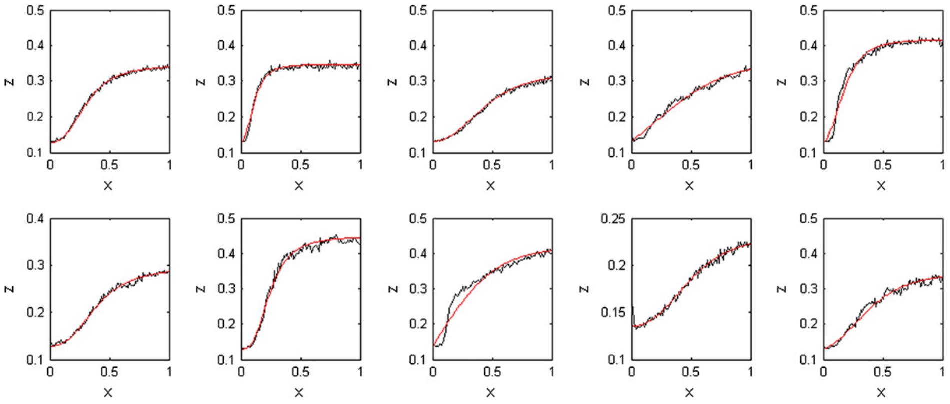


Fig. 7. Ten random curves (black) from the 2DGE data and their fits (red) with the DLU. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

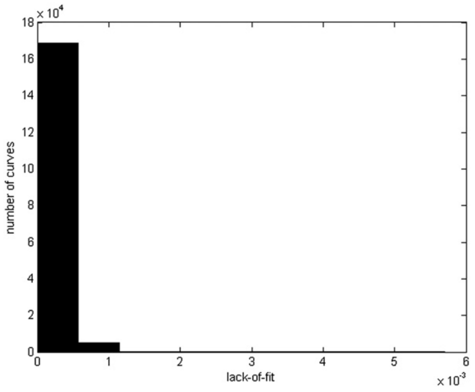


Fig. 8. The histogram showing a number of curves with a certain lack-of-fit.

were estimated. The time needed for fitting an increasing number of input curves was recorded and is shown in Fig. 6. Naturally, working time increases with the size of a data set. Hence, with the DLU one can know exactly how many floating point operations which are needed, and, hence, approximately how much CPU time it would take to analyse a given amount of data. For fitting large numbers of curves, disk swapping and other householding activities comes in addition.

Fig. 6 shows that a linear increase in fitting time for the DLU is observed for up to 90,000 curves at the same time, above which it increases, probably due to memory limitations causing disk swapping etc. Nevertheless, even such amount of time is still far less than ILS would spend to estimate parameters for 90,000 curves (90*120.61 = 10,854.90 seconds (Table 2) against 295.80 seconds with DLU). To escape the jump in fitting time due to disk swapping, it could be useful to increase memory capacity or divide the data set into several parts (up to 90,000 curves in each in our case) and use the DLU approach on each of it.

Fig. 7 shows DLU fits for ten randomly chosen curves from the data. Among these both good and poor fits can be observed. In order to prove that poor fitting is a seldom phenomenon, weighted lack-of-fit was calculated

$$S_i = \frac{1}{\sum_{l=1}^{K} w_i(x_l)} \sum_{l=1}^{K} \left[z_i^{\mathrm{obs}}(x_l) - \hat{z}_i(x_l) \right]^2 \cdot w_i(x_l) \tag{24}$$

and plotted as a histogram on Fig. 8. The histogram indicates that there is only a very small number of curves with poor fit and, hence, unreliable parameter estimates. Table 3 presents parameter estimates of the example curves plotted in Fig. 7.

After the parameters for the whole data set had been estimated, the curves were reconstructed from the 5PL function with the appropriate parameter estimates. It was then possible to get a movie of how the gel dying process was estimated to develop. While the input movie had some artefacts due to incomplete motion estimation, the reconstructed movie was without these or other visible artefacts. Fig. 9 compares the spatial configuration of the silver staining of the original and the reconstructed movies at the time point 44. This illustrates that the estimation of the parameters was quite satisfactory. More details on how the five different parameters from the 5PL model manifested themselves spatially in the 2DGE images are given in Nhek et al. (2011) (in preparation).

Table 3
Parameter estimates for the curves shown in Fig. 7. Numbers correspond to the plots in the following way: from the left to the right, from the top to the bottom.

	p_1	p_2	p_3	Slope	Offset
1	0.35	− 3.00	0.75	0.2194	0.1271
2	0.15	− 3.00	0.55	0.2201	0.1275
3	0.55	− 3.00	0.65	0.1968	0.1311
4	0.80	− 3.00	0.35	0.2327	0.1321
5	0.25	− 3.00	0.50	0.2904	0.1275
6	0.45	− 3.00	0.75	0.1682	0.1281
7	0.25	− 3.00	0.95	0.3207	0.1293
8	0.60	− 3.00	0.30	0.2902	0.1340
9	0.60	− 3.00	0.70	0.1006	0.1351
10	0.50	− 3.00	0.60	0.2541	0.1319

5. Discussion

In this paper the DLU method was improved in comparison to the version in Ref. [1] by introducing WLS instead of OLS in the estimation of metamodel scores and distances. It means that the whole modelome of curvature (38 curvature models [1]), or selected parts of it, can now be applied to large sets of sigmoid curve data, with both homoscedastic and heteroscedastic noise.

Table 2 shows that using the direct look-up data base of curves works at least as well as the ILS estimation, with respect to the precision of the parameter estimates. Moreover, it does not need any additional assumptions about function or parameter values (as many statistical methods do). In the first example it was easy to decide what kind of functions to explore, since we knew how the curves had been generated. But in the second example, the underlying chemical mechanism behind the silver staining development and its kinetics remains more or less unknown. In such cases, considering only one nonlinear function – or even only three functions – among a large number of possible causal mechanisms or usual function forms, may lead to a mistaken mechanistic interpretation or suboptimal functional form. In the present case, the 5PL function was generally found to give much higher frequency of acceptable fit than the other 37 curve models. The reason may be its ability to model also asymmetry in the sigmoid curves, as represented by its parameter p_3.

The performance of the data base approach was compared with an ILS approach in terms of computation time and parameter estimation accuracy (Section 4.1). When having equal conditions for both methods (the same number of curves and tested functions), the look-up method reduced the time needed for finding a model by a factor of 24 compared to the ILS estimation method in the cases when ILS found a solution. It is also very important that DLU in contrary to simplex does not depend on initial values. As we saw in Section 4.1, results from ILS are highly dependent on the point where we start. Choosing good starting values for every single curve may be very time-consuming when having a large data set, like, for example, in Section 4.2. With default settings in Matlab (both termination tolerance on the function and variable value equal to 1.0000e − 04), in 276 cases out of 1000 the ILS did not converge to a sensible solution, whereas the look-up method had no cases of non-sensible solutions. In the Discussion section of Ref. [1] it was mentioned the opportunity to use DLU and ILS together: first use DLU to identify function type and a good set of parameters, and then use these parameters as starting values for ILS. This would reduce the chance of getting stuck in the local optima and computation time is expected to be shorter than for ILS alone. The results from Section 4.1 showed that ILS needed 120 seconds when trying out three functions, whereas with the DLU approach three functions are checked in 5 seconds. Hence, in addition to avoiding the risk of local optima, there may be considerable time saving benefits from using DLU for pre-optimisation ahead of ILS.

In a practical solution, modelling with the DLU method alone may be by far more time effective than the ILS approach, which may give bad solutions requiring further actions from the analyst. Moreover, the speed and stability of the look-up method shown here are valuable for experiments giving a large number of curves to be analysed. Although the look-up method is relatively fast in comparison to ILS, it would be preferable to speed it up even more to perform parameters estimation

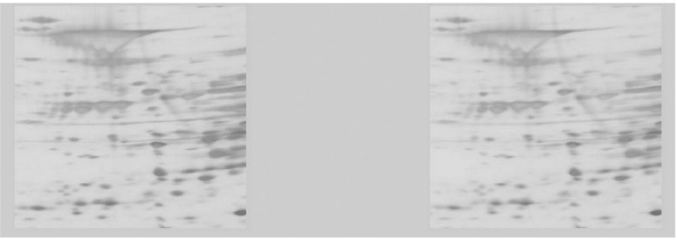


Fig. 9. Original and estimated gel at the time point 44.

on the fly. At the time being, new and more effective search algorithms are tested for the look-up step in our approach, and we anticipate a further reduction in computation time for newer versions of our methodology. This will be valuable if the method is to become practical for computational compaction in more complex models, such as high dimensional systems of nonlinear coupled differential equations.

It was also verified that the look-up method is able to work with data that are affected by heteroscedastic noise. It was tested on the 2DGE data (Section 4.2). For all curves tested the look-up approach predicted that the 5PL function describes the data best (which confirmed the initial guess of the experimentalist). It should be noticed that the fitting time was increasing linearly depending on the number of curves in the data set (when fitting less than 90,000 curves at the same time). Hence, the time needed for fitting a huge number of curves can be well predicted up front of the analysis. In order to fit larger sets of data (more than 90,000 curves), the program has to be further improved or, otherwise, the data set has to be divided into smaller parts. More than 97% of curves ($1.69e-05$) were fitted very precisely shapewise, with the lack-of-fit less than $1.0000e-03$. It means that setting *a priori* weight vectors enables the DLU approach to handle heteroscedastic noise.

Besides, the choice of starting values for the function parameters is often a problem in iterative hill-climbing, as we found here. With the wrong choice of starting values, the simplex optimisation function often ended up in a local minimum, which gave a bad curve fit and erroneous parameter value estimates.

There is still room for improvement of the metamodelling approach. One of these is to introduce an x-shifting forcing all the curves to pass through the point $(0.5; 0.5)$. This may reduce the number of components needed in the compression of the database and hence may reduce the storage space. Whether this gives a computation time benefit is unclear at the moment.

Another possible (and needed) improvement is to use a different way of selection the best parameter set. Calculating the distance for all possible solutions and them sorting them takes relatively much time, and it is believed that this time can be significantly reduced.

All pros for the DLU approach, despite cons existing at the moment, allow us to conclude that this method presented here works very well and may be highly competitive with other traditional methods in terms of speed and precision of work.

Acknowledgments

Stig W. Omholt and Kristin Tøndel are thanked for constructive comments. This study was supported by the National Program for Research in Functional Genomics in Norway (FUGE) (RCN grant no. NFR151924/S10) and by the Norwegian eScience program (eVITA) (RCN grant no. NFR178901/V30).

References

[1] J. Isaeva, S. Sæbø, J.A. Wyller, O. Wolkenhauer, and H. Martens. Nonlinear modelling of curvature by bi-linear metamodelling. J. Chemom. Intell. Lab. Syst. in press, doi:10.1016/j.chemolab.2011.04.010.

[2] A. Kohler, U. Böcker, J. Warringer, A. Blomberg, S.W. Omholt, E. Stark, H. Martens, Reducing inter-replicate variation in Fourier transform infrared spectroscopy by extended multiplicative signal correction, Appl. Spectrosc. 63 (3) (2009) 296–305.

[3] A. Kohler, M. Zimonja, V. Segtnan, H. Martens, Standard Normal Variate, Multiplicative Signal Correction Preprocessing in Biospectroscopy, volume 2, Elseveier, 2009, pp. 139–163, chapter 2.09.

[4] H. Grove, E.M. Faergestad, K. Hollung, H. Martens, Improved dynamic range of protein quantification in silver-stained gels by modelling gel images over time, Electrophoresis 30 (2009) 1856–1862.

[5] P.J. Bickel, K.A. Doksum, Mathematical Statistics: Basic Ideas and Selected Topics, Prentice-Hall, Inc., 1977.

[6] D.C. Montgomery, E.A. Peck, Introduction to Linear Regression Analysis, 2nd edition Wiley, New York, USA, 1992.

[7] P.G. Gottschalk, J.R. Dunn, The five-parameter logistic: a characterization and comparison with the four-parameter logistic, Anal. Biochem. 343 (2005) 54–65.

[8] J.C. Lagarias, J.A. Reeds, M.H. Wright, P.E. Wright, Convergence properties of the Nelder-Mead simplex algorithm in low dimensions, SIAM J. Opt. 9 (1996) 112–147.

[9] J. Warringer, D. Anevski, B. Liu, A. Blomberg, Chemogenetic fingerprinting by analysis of cellular growth dynamics, BMC Chem. Biol. 8 (2008) 3–12.

[10] L. Bintu, N.E. Buchler, H.G. Garcia, U. Gerland, T. Hwa, J. Kondev, R. Phillips, Transcriptional regulation by the numbers: models. Curr. Opin. Genet. Dev. 15 (2) (2005) 116–124.

[11] N. Rosenfeld, J.W. Young, U. Alon, P.S. Swain, M.B. Elowitz, Gene regulation at the single-cell level, Science, American Association for the Advancement of Science 307 (5717) (2005) 1962.

[12] R.A. Veitia, A sigmoidal transcriptional response: cooperativity, synergy and dosage effects, Biol. Rev. 78 (1) (2003) 149–170.

[13] C.A. de Boor, A Practical Guide to Splines, Springer-Verlag, New York, 1978.

Paper IV

The modelome of line curvature: Many nonlinear models approximated by a single bi-linear metamodel with verbal profiling

Julia Isaeva[1*], Magni Martens[2], Solve Sæbø[1], John A. Wyller[3,4] and Harald Martens[3]

6th June 2011

[1] Department of Chemistry, Biotechnology and Food Science, Norwegian University of Life Sciences, P.B. 5003, N-1432 Ås, Norway
[2] Nofima Mat AS, Osloveien 1, NO-1430 Ås, Norway
[3] Centre for Integrative Genetics (CIGENE)/IMT, Norwegian University of Life Sciences, N-1432 Ås, Norway
[4] School of Mathematical Sciences, University of Nottingham, NG7 2RD, UK

[*] Correspondence to: Julia Isaeva, Department of Chemistry, Biotechnology and Food Science, Norwegian University of Life Sciences, P.B. 5003, N-1432 Ås, Norway
E-mail: julia.isaeva@umb.no
Tel: 0047 6496 6243

Abstract

A generic mathematical phenomenon (line curvature) is described quantitatively and linguistically: a range of very different in form and representation models $z = F_m(x)$, $m = 1, 2, .., 38$, each yielding smooth, but curved relationships $z = f(x)$ with 0 or 1 inflection points, were collected from different fields of science, ranging from systems biology and statistics to trigonometry and psychophysics (Isaeva *et al.* (2011)). The behavioural repertoire of each of the models was realised by exhaustive statistically designed computer experiments, yielding a total of about 50,000 curves $z = f(x)$, each recorded at 100 x-values. A *modelome* of curvature was formed by this set of arched or sigmoid curves and was preprocessed and combined in a joint metamodel based on a bi-linear subspace analysis. To describe a total of 99.9% of the variability in the curves, 12 eigenvectors were needed. These 12 common curve descriptors were successfully related back to the original model input parameters in each of the individual models. Furthermore, to give verbal meaning to the *per se* meaningless axes in this 12-dimensional eigenvector space, a total of 64 curve images were selected by a statistical design, printed and submitted to descriptive sensory analysis, using a panel of ten trained judges. A quantitative map between the eigenvector space and the sensory space was successfully established and then used for predicting what the human descriptive profiling would be for each of the 50,000 curves. Thus, a first version of a complete "modelome" of the mathematical phenomenon "line curvature" has been established by multivariate metamodelling and described in terms of quantitative maps both to the original model parameters in the 38 individual models and to human verbal description of curve shapes.

Keywords: modelome, sensory science, human description, function parameters, curves

1 Introduction

1.1 The optimal model level: Detailed mechanism, crude approximation or general phenomenon?

This paper concerns the relationship between a generic relational phenomenon, its possible mathematical realisations and their causal interpretations. Mathematics is said to be the *lingua franca* of science. Applied mathematical modelling allows complicated conceptualisations to be formalised and tested. Mechanistic modelling allows specialists to describe their system of interest in light of their understanding, in a very efficient way, at the desired level of resolution. A model description that matches well to the available empirical evidence is a compact, concise, flexible and relevant representation of knowledge. Even if a mathematical formulation is uncertain, or causally wrong, it may still be useful, for crude approximation, as a functional building block in a larger system or as a tentative realisation of a thought experiment.

However, outputs from an uncertain formula must be interpreted with care; the concise nature of mathematical functions make them appear more objective or accurate than they really are. For instance, a chosen nonlinear function may, of course, give a good fit to empirical data if it has sufficiently many independent model parameters to be estimated. But the estimated parameter output may give problems, not just for statistical reasons (instability due to general over-parameterisation or specific collinearity), but – more seriously – model misspecification: a reasonable curve fit may appear to confirm a wrong mechanistic understanding.

Good modelling practice, therefore, lets the level of technical detail in a model correspond to a conscious choice of explanatory ambition. At one extreme, high-resolution mechanistic details may be modelled explicitly (usually requiring nonlinear formulations) as long as they reflect reliable knowledge or important postulates. At the other extreme, low-resolution crude linear or polynomial approximations may be preferable if the underlying mechanisms are unknown or irrelevant, and accuracy is not important.

But what about situations when a functional phenomenon needs to be parametrised accurately, but the underlying causal details are unknown or uncertain? Choosing to use wrong mechanistic model may give adequate curve fit, but lock the user's mind onto the wrong thought track. On the other hand, a simple straight line might give bad curve fit, while a more flexible polynomial might give meaningless model parameters and bad extrapolation properties. The idea in this paper is to develop and employ an intermediate-level, multivariate metamodel that describes a functional phenomenon at a generic level, with sufficient accuracy and focus but without unwarranted detailed mechanistic assumptions.

1.2 Line curvature: a functional phenomenon and its modelling

As an example of such a generic functional phenomenon, we here focus on simple line curvature, i.e., nonlinearity in a two-dimensional plane. Plane curves can be defined in different ways, e.g., in analytical geometry it is a set of points that satisfy the following equation [1]:

$$F(x, y) = 0 \text{ (implicit function)} \tag{1}$$

or, in the more familiar form of a function,

$$y = f(x) \text{ (explicit function).} \tag{2}$$

We restrict the phenomenon further, to *line curvature*, i.e., *monotonic bi-variate relationships with zero or one inflection point*. Even this simple phenomenon includes a wide range of curves, from a straight line via smooth arches and sigmoids and to near-step functions.

Monotonic, arched or sigmoid, relationships response curves are generated in many different natural systems, ranging from growth curves to cumulative statistical processes, with different causal mechanisms. Conversely, a wide range of mathematical functions can describe such line curvature. To choose the right mechanistic model in a given natural system can be difficult, while a wrong model may give misleading interpretation.

The present goal is to develop a metamodel of the *phenomenon of line curvature as such*, and to characterise this metamodel both by concise mathematical parameters and in more mundane human language. The generic line curvature model is intended to provide precise description of all such curves, without unwarranted assumptions, with simple parameter estimation. It is intended to encompass a wide range of specific curvature models, which, thereby, can be related to each other via the parameters of their common metamodel. The paper, thus, presents a first version of the "modelome of line curvature", combining a very wide choice of mathematical curvature models and functions, from a variety of sciences, into one single metamodel.

The metamodel is an approximation model intended to span all the variabilities in the output of all the chosen mathematical curvature models up to a chosen, high approximation accuracy. It is generated based on data from extensive computer simulations, summarised by a joint multivariate metamodel. This metamodel is obtained by an automatic preprocessing and subspace expansion. For each of the many explicit input models, its parameters are linked to the parameters in the metamodel by dedicated multivariate regression models.

1.3 Sensory description of the metamodel

Nonlinear mathematical modelling is increasingly used also by non-mathematicians, helped by standard computer software. The choice of model and the interpretation of their parameters may be difficult for non-mathematically oriented people. For instance, in the bio-sciences, specialists in biology, microbiology, physiology, biochemistry and biophysics collect a large amount of data that happen to display curvature. Due to the lack of advanced mathematical knowledge, some of them may face a problem of analysing their data and of communicating with mathematicians and physicists in scientific terms. Therefore, to be useful also for non-mathematicians, a generic metamodel of curvature must be interpretable.

Contrary to the parameters in the individual functions (Eqs. (1) and (2)), where the model parameters usually have explicit meaning, the parameters in a subspace model have no meaning *per se* – they only represent coordinates and directions in an orthogonal axis system. To provide meaning and allow interpretation, there is a need for additional characterisations of the metamodel, at least in its dominant subspace dimensions. This will here be done with two different methods of conceptualisation – visual ***prototype illustration*** and *verbal descriptive profiling.* The former is simple – finding and plotting representative curves for each of the main regions in the metamodel. The latter requires considerable interdisciplinary cooperation, within the framework of sensory science.

The majority of people probably do not think of curves in mathematical terms: they rather see them as resembling familiar objects: outlines of buildings, shapes of fruits, trajectories of movements etc. Simple words from every-day use are employed to describe contours and patterns: long, heavy, rigid, symmetric, gentle etc. On the other hand, scientists may prefer descriptors corresponding to their general understanding – delay phase, linear, saturation etc. From this, the idea arose of characterising the modelome of line curvature sensorically, by mapping between its formal representation with mathematical models and metamodel parameters on one hand, and human language on the other.

There already exists a large amount of literature describing and studying curvature and its perception by human beings as for example in Refs. [2, 3, 4]. However, it was not clear whether the human perception of curves and mathematical expressions could be mapped into each other. For instance, to what degree is it possible, for a data set forming a given curve, to identify the reasonable function types and estimate their parameters from a verbal description of a curve using a list of attributes established beforehand? Such a verbal descriptor list, along with calibration scales, could play a role in cross-disciplinary communication: it could improve understanding, e.g., between bio-scientists and mathematicians, even though they speak different languages.

For this purpose, a sensory study was conducted. Descriptive sensory analysis is widely used in food science [5, 6] and consists in selecting a representative set of objects (in this case – individual curves) and profiling each of them (in this case the individually curves printed on paper) by a well trained panel of judges, using a predefined list of words-descriptors developed for the problem at hand. The sensory panel average profiles can then be mapped to external information about the same objects (in this case the model parameters or metamodel parameters of the chosen curves). In principle, it should then be possible to predict the sensory profile for new curves from their metamodel parameters, and, likewise, to predict their metamodel parameters from their sensory profile. The metamodel parameters are unique, but may, in turn, be linked to the model parameters in the individual curvature functions (of course, this may be a one-to-many mapping since the range of shapes from different functions will be partly overlapping). Hence, if successful, this combination of metamodelling and sensory profiling should allow both a mathematical and a verbal description of each and every parameter combination for each and every curvature function.

A previous use of sensory descriptive analysis in mathematical modelling was published by Martens *et al.* [7], Martens *et al.* [8] and Isaeva *et al.* [9]. In that case, one given, high-dimensional nonlinear dynamic model of cell differentiation was studied with respect to the effects of varying certain input parameters and initial conditions on the output cell pattern. The human sensory assessment of printouts of selected 2D solution patterns, combined with multivariate data mapping, was essential in the discovery of a new, systematic, but highly unexpected differentiation pattern. The same sensory profiling approach is here applied to mathematical models with simpler outputs (curves), but now for a whole class of models representing the same phenomenon (curvature), each with its parameter space probed in much higher resolution.

It is known that the visual cortex of the brain contains simple cells that easily can recognise a straight line [10, 11]. Besides, there exist hypercomplex cells that react on the curvature [10], but not that intensively as on the straightness. Perhaps, due to this separation of the perception cells, humans tend to distinguish straight lines from curves and consider them as two individual concepts. Nevertheless, according to the definition of a line (curve) in [1], a straight line is one of the curve types, so to say, a critical form, with the curvature equal to zero. Hence, another aim of the sensory analysis here was to verify whether a straight line would fit into the "community" of curves according to the sensory panel or not.

1.4 Model, metamodel and mapping

The term "model" is already familiar to scientists and means a simplification of the real world, e.g., by the use of mathematical "language" (functional forms and parameter values). For instance, having a large amount of data, one wants to know common properties of these data and implements various types of analyses to find a proper model describing most of the observed properties and reducing the data dimension. *Metamodel*, in turn, is a further simplification of data achieved by modelling of a model [12]. In our case, a particular metamodel is employed, based on a simple preprocessing and bi-linear eigenvector compression, followed by mapping back to the original nonlinear model parameters as well as external verbal descriptions.

The choice of curvature models, design of computer simulations and the general technique for a bi-linear metamodel development was explained in detail in Ref. [13]. A set of 38 functional models of line curvature was collected from different scientific fields. For each of them, extensive simulations were used for generating output curves, and a metamodel for each model was established by the principal component analysis (PCA) of the curves after a simple preprocessing. The compact eigenvector representation of the metamodel led to a significant reduction of dimensionality compared to the original data since all redundancy is compressed into joint metamodel parameters. The collection of 38 individual metamodels was used in Ref. [14] for finding the most adequate nonlinear model type (the five-parameter logistic curve) to describe proteomic growth curves based on their lack-of-fit. Moreover, the parameters of the optimal model, which was highly nonlinear, were estimated for a massive number ($>$ 170,000) of curves via its metamodel. Supervised use of conventional nonlinear iterative hill-climbing proved to be impractical due to long and unpredictable estimation time, and unsupervised use was deemed dangerous due to the risk of local minima. But since the metamodel was linear, the parameter estimation took only a fraction of the time, since each curve fitting consisted only in a simple linear projection followed by a local table look-up, apparently with no risk of finding local minima.

Instead of using individual metamodels for each mathematical model, in the present paper one joint metamodel for all the individual mathematical models is developed and described. Figure 1 outlines the comparison of conventional modelling and the multivariate metamodelling. A nonlinear model is constructed from prior knowledge, and matched to observed data (top plot). The corresponding metamodel of its model is developed (plot in the middle) by using prior knowledge to construct a statistical design to span the parameter space at a chosen resolution and range, then performing extensive computer simulations with high-dimensional monitoring of the outputs; the large tables of simulated data are compressed into a bi-linear metamodel, whose parameters are mapped back to the original parameters, or to external information (e.g., sensory profiling). The original model may be fitted to massive amounts of observed data (plot at the bottom) via its established metamodel, and the resulting metamodel parameter estimates may then predict the unknown parameter values.

In Section 2.1 of this paper we describe construction of a metamodel for this particular analysis that is described in Section 2.2. We present obtained results in Section 3 and close the paper by the discussion of them in Section 4.

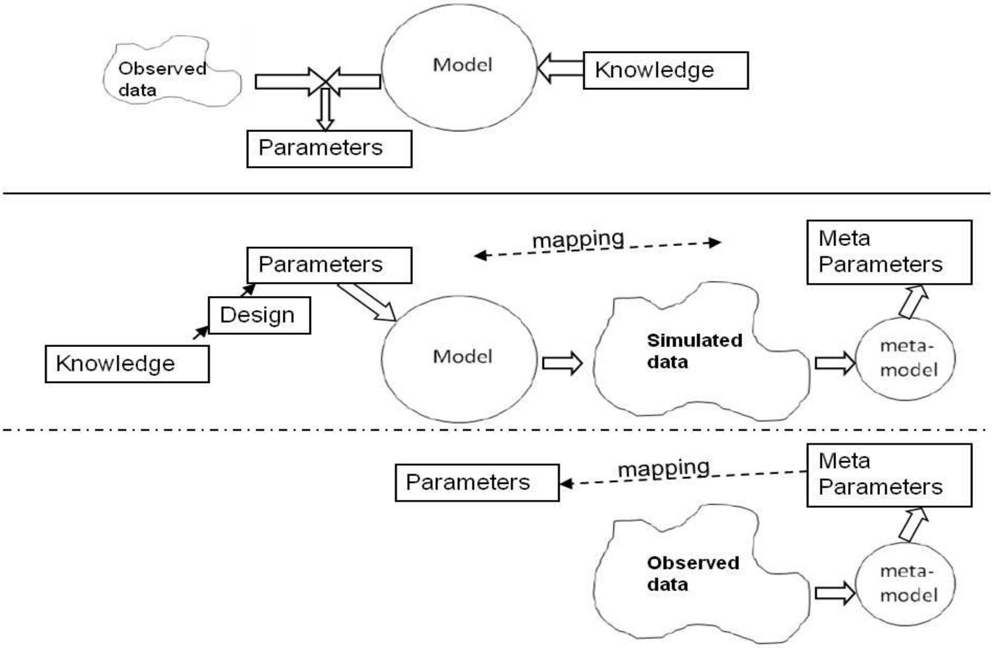


Figure 1: The relationship between conventional modelling and metamodelling. Top plot: Development of a nonlinear model. Middle plot: Development of a bilinear metamodel. Bottom: Fitting the model to massive amounts of observed data, via its metamodel.

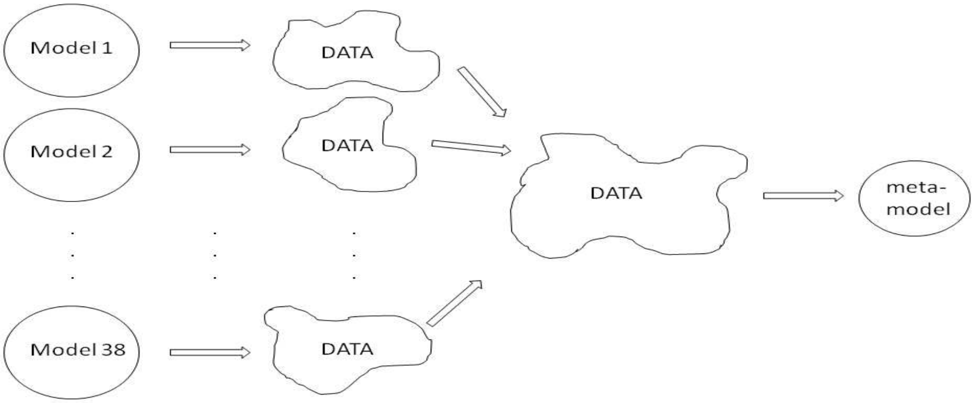


Figure 2: Construction of a global metamodel: data were generated for each model separately, then all of them were merged into one large data set and modelled altogether.

2 Methods

2.1 Metamodel

In Section 1.4, we mentioned a collection of realisations of 38 mathematical functions that has been formed for parameter estimation of each new curve obtained from an experiment. All curves in the collection were simulated on the x-interval of [0.001;1] with 100 observation points ($x = 0$ was omitted here to avoid division by zero for some of the functions). Curve shapes vary from a straight line to almost a step function showing that each function can give a wide spectrum of curves depending on parameter values, and the complete set of curves is hereby called *the modelome*, whereas a set of realisations for each single function type is called a *function phenome* [13]. Among functions represented in the modelome are polynomials, trigonometric and hyperbolic trigonometric functions, cumulative distribution functions for statistical distributions and others.

In order to verify that the modelome of curves captures the entire phenomenon of curvature, it was decided to make a joint metamodel for all the curves. In view of the fact that the original curves may have extremely different scales, all curves were preprocessed to be increasing functions on the [0;1] interval on the y-axis before they were put together and formed one modelome. The total number of curves was 47,840, hence, the size of the modelome was 47,840×100. These curves were then mean centred altogether, and, as expected, the global mean centre was a diagonal line. Further, PCA was used to compress the modelome of curves and to make a metamodel:

$$Z = \bar{z} + T_G \cdot P'_G + E, \tag{3}$$

where Z is the matrix of the preprocessed centred curves from all the 38 models; $\bar{z}$ is the global mean centre curve; T_G and P_G are scores and loadings of the modelome respectively; E is a matrix of the unmodelled residuals; and index G corresponds to *global* here. The set of scores and loadings formed the global metamodel that we are interested in. Figure 2 shows how the modelome of a generic phenomenon is developed, as a joint metamodel encompassing a whole set of alternative mathematical models of that phenomenon.

Such a compression ensures a minimal loss of information from the data set. It is a simple bi-linear transformation with a precision set in advance by an analyst. For now we just say that six principal components (PCs) were considered as an optimal number for building a data set for the sensory analysis since they explained 99.3% of the variability in the data; and more details about results of metamodelling can be found in Section 3.1.

2.2 Sensory data

2.2.1 First run

Our main interest was to check whether it is possible to map outcomes of pure mathematical expressions into the daily language used by people far from mathematics. With that end in view, a sensory panel was set up with ten well trained judges. The curve set consisted of 32 preprocessed curves, and the evaluation was run during two days, where the second day was a randomised repetition of the first one. In this way, two replicates for each sample were obtained giving an opportunity to assess reproducibility of the judges.

Curves for the sensory panel were chosen based on the requirement of their equal distribution in the curvature space to provide the judges with as many types of curves as possible. For this reason, multi-level binary replacement (MBR) design [15, 16] was applied here on the six-dimensional global score-space of the metamodel. A fractional factorial MBR design was employed to obtain 64 sample-curves for the sensory study. Further, values from T_G closest to the design levels were found, that is, 64 score vectors, equally distributed in the six-dimensional score space, were obtained. To reconstruct the curves corresponding to the chosen score vectors, Eq. (4) was applied:

$$Z_{recon} = \bar{z} + T_{G\_design} \cdot P'_G. \tag{4}$$

We wanted to make sure that curves for the sensory evaluation are accurately estimated by the global PCA model, and, therefore, it was decided that only curves that were monotonous after the reconstruction would be possible candidates for the sensory analysis. Since the conducted reconstruction brought us back to the curves after preprocessing, all the samples were supposed to be increasing. Here this requirement was relaxed a little since six PCs explained not exactly 100% of variability. A curve was considered to be increasing if

$$z(x_2) - z(x_1) \geq -0.005, \text{ where } x_2 > x_1. \tag{5}$$

Following this condition, 49 out of 64 curves turned out to be monotonous and were kept for further investigation. To get more curves, scores for all curves in the modelome for the first two PCs were plotted and colour-coded according to the monotonicity of the corresponding reconstructed curves; and another 20 plausible curves were randomly picked out from that plot in addition to already chosen 49.

Thus, a set of 69 potential candidates was formed. Corresponding original preprocessed curves from the modelome were printed out, and only 32 of them, representing as many curve types as possible, were selected for the sensory evaluation.

As was mentioned above, each curve was evaluated twice: the first day and the second day. However, mistakenly, curve number 3 got into the sample of the second day twice, forcing out curve number 21. Thereby, curve number 21 has no replicates at the end, whereas curve number 3 got three replicates.

The judges used 14 descriptors to evaluate each of the curves (see Table 1) on the unstructured scale from 1.0 to 9.0. At the end of each day, average across the judges was taken for all the attributes for every curve. Figure 3 demonstrates examples of the curves that got high values for some of the descriptors.

2.2.2 Second run

Sensory analysis was repeated with the same descriptors four months later. The size of the data set was the same (32 curves per day), but the content was slightly changed: some new models were added. Three psychophysical laws and their curve outputs were studied here: Weber's law, Fechner's law and Steven's law [17]. The formulations of the laws were rewritten in such a way that they would fit our criteria of curves in the modelome, namely, to depend only on one variable x. After omission of slope parameters we got:

$$\text{Weber's law} \quad y = x \tag{6}$$

$$\text{Fechner's law} \quad y = \ln(x) \tag{7}$$

$$\text{Steven's law} \quad y = x^p. \tag{8}$$

Name	Description	Low (1.0)	High (9.0)
Sigmoid	Degree of sigmoidness (*s*-curve)	Not sigmoid	Sigmoid
Arc	Degree of arc, **one** long arc	Small arc	Long arc
Symmetrical	Degree of symmetry	Asymmetrical	Symmetrical
Heavy	Degree of heaviness	Light	Heavy
Deviation from a straight line	Distance from the peak till the diagonal	Short distance	Long distance
Pliable	Degree of pliability	Little pliable (rigid)	Pliable
Initial phase	Length of the initial phase (at the bottom)	Short	Long
Lower arc	Degree of curvature	Small arc	Long arc
Steepness	Degree of steepness of the steepest part	Low steepness	High steepness
Upper arc	Degree of curvature	Small arc	Long arc
Stationary phase	Length of the stationary phase (at the bottom)	Short	Long
Harmonic	Feeling of harmony and balance	No harmony	Harmony
Elegant	Feeling of an elegant form	No elegance	Elegance
Associations	Degree of associations	No associations	Many associations

Table 1: Description of the sensory variables used for evaluation of the curves.

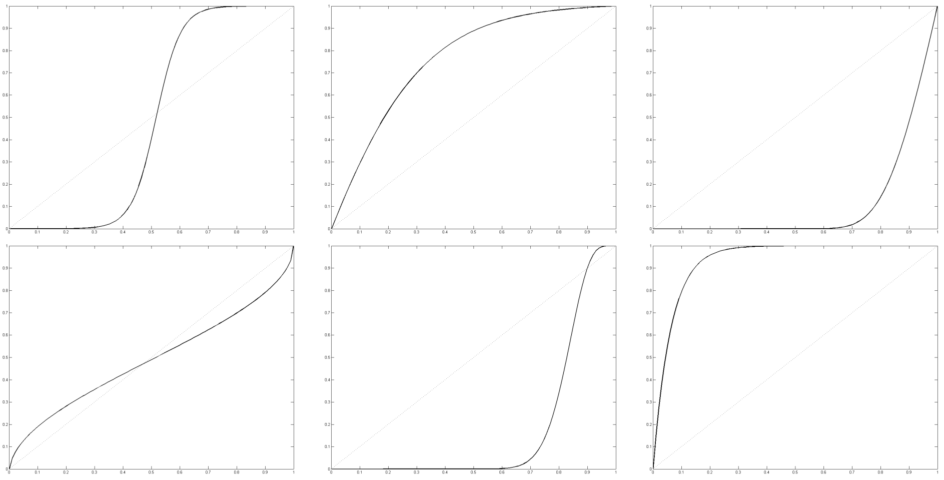


Figure 3: Examples of the curves with high values for *Sigmoid*, *Arc*, *Heavy*, *Pliable*, *Steepness* and *Stationary phase* respectively (from the left to the right, from the top to the bottom). Here the dotted diagonal line is a reference line for the sensory panel. It was decided to have it on the print outs for better visualisation of the curves behaviour with respect to the straight line.

In total, ten representatives of these new functions were taken to the sensory panel (one, two and seven curves for each of the laws respectively). Among them, a straight line happened to be twice: as an outcome of Weber's law and as one representative of Steven's law ($p = 0$).

Six other curves were obtained by simulations of a dynamic model of a gene regulatory network studied by Gjuvsland in Ref. [18]. Each model described there has three state variables, and for this paper model number 6, scaled and dimensionless, was randomly picked. Differential equations for the chosen model are following:

$$\begin{aligned} z_1' &= \alpha_1(1 - Y_1)Y_2 - \gamma_1 z_1, \\ z_2' &= \alpha_2(1 - Y_1) - \gamma_2 z_2, \\ z_3' &= \alpha_3 Y_1 Y_2 - \gamma_3 z_3, \end{aligned} \tag{9}$$

where z is a function of x; $\alpha_3 \in (0;1)$; all $\gamma_j \in (0;1)$, and $Y_j = z_j^p/(z_j^p + 1)$. Details see in Ref. [18].

Simulations were done with 125 different sets of starting values, and two of those that gave monotonous integral curves were picked out, that is, we obtained three curves for each set of starting values.

Further, for checking how good judges are in reproducing their own evaluation several months later, 14 random curves from the first run, of various shapes, were selected to be in the second round as well. And finally, to complete the data set (to have 32 curves), remaining two curves were chosen from those that were in the candidates' list for the first run but were not evaluated.

Hence, 32 curves were collected for the second run of the sensory evaluation. Among them are 16 curves from the new models, 14 old curves (old models, same parameters) and two curves of the old models but new parameter sets. Again, as during the first run of the sensory analysis, only preprocessed curves were considered here.

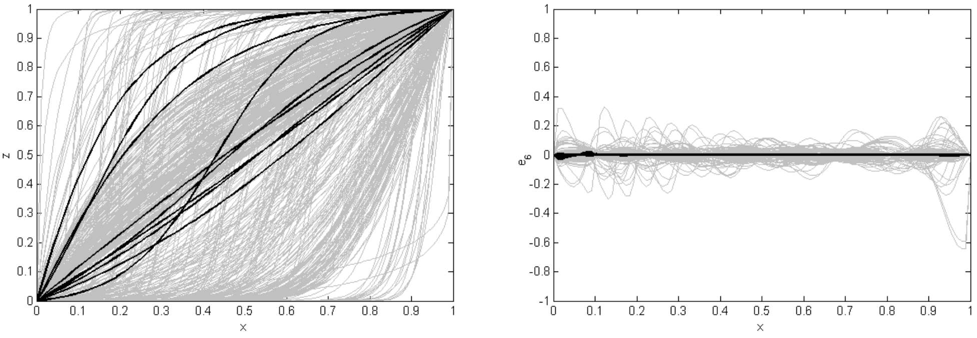
It was hoped that by the first curve set, the phenomenon of curvature would be explained quite extensively. In that case, having new models and projecting their curves onto the metamodel of the library would not give us any outliers or protruding results. Presence of a straight line made it even more interesting to see how the sensory panel would evaluate it in comparison to the other "typical" curves.

3 Results

3.1 Metamodel

Construction of a metamodel, as was mentioned above, leads to a simplified representation of the models in the modelome. For example, in Figure 4a one can see a wide variety of curves in the modelome represented by a subset of 500 random curves out of the 47,840. Six PCs of the global PCA model explained so much variability in the data that, after subtracting them from the latter, the residuals are rather small (Figure 4b). However, for building a more accurate model of the relationship between sensory values and function parameters, 12 PCs were taken into consideration, which explained 99.9% of the data variability (Figure 5). It means that having 12 basal curves (loadings) is enough to reconstruct each original curve with a precision deemed high enough for most practical purposes. Examples of the first six orthonormal loadings together with the global mean centre curve are presented in Figure 6b. It can be noticed that all basal curves have different degree of curvature, thereby providing various sigmoids by their linear combinations. These orthonormal prototype curves resemble sine waves with increasing frequencies, but contrary to a Fourier series, they are designed to describe as much variability in the curves as possible with as few terms as possible. Figure 6a, in its turn, shows the scores for all the curves for the first six PCs of the metamodel. Since the 38 different models are stored sequentially, it can be seen how the different models contribute to the different metamodel dimensions. Among them are polynomials, hyperbolic tangent, Hill function, five-parameter logistic function, generalised logistic function and cumulative distribution function for Student's t-distribution.

Hence, by building a global metamodel, we managed to represent the large original data matrix of size 47,840×100 by two, much smaller matrices of scores and loadings of size 47,840×12 and 12×100 respectively. The latter two are apparently easier to store and to handle. It is believed that this metamodel, comprising the 12 dimensional bi-linear model, captures the essential dimensionality of the line curvature phenomenon. It is expected that any new parameter combination of any of these 38 models will at the chosen resolution fall



(a)

(b)

Figure 4: 500 random curves from the joint modelome: (a) – original (locally mean centred) curves, (b) – shows residuals from the data after subtracting six PCs. Here black curves correspond to the logistic function type that is used in the first example in Section 3.4.

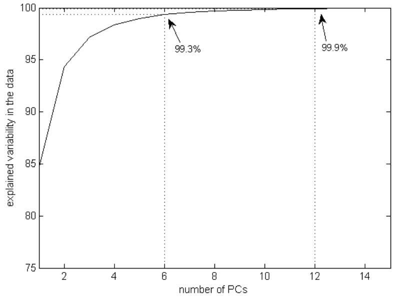
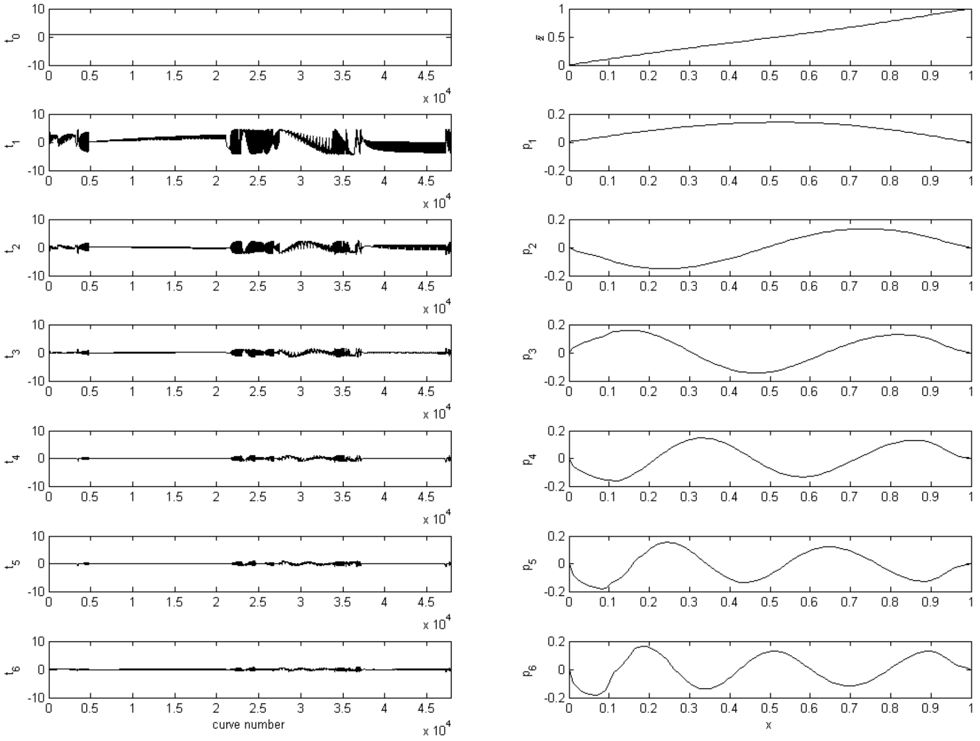


Figure 5: Number of PCs and corresponding percentage of explained variability in the data by the global metamodel. Six PCs are considered for making design for sensory evaluation, whereas for building a model of a further model 12 PCs are taken.



(a)

(b)

Figure 6: Scores (a) and loadings (b) of the global metamodel for the first six PCs. Top right plot in Figure (a) shows a vector of ones corresponding to the global mean centre curve (top left plot in Figure (b)). We denote it as t_0 here since mean centre can be considered as a zeroth PC in any PCA model.

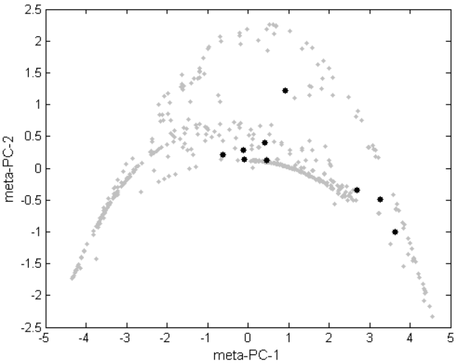


Figure 7: Metamodel scores for 500 random curves from the modelome: first meta-PC vs second meta-PC. Black dots here correspond to the curves of the logistic function type.

inside the 12-dimensional subspace of P_G. From its metamodel scores T_G (metascores), the mathematical form and the parameter value of the generating function may, therefore, be identified in that process. Of course, in the generic modelome metamodel, several different nonlinear models may yield more or less the same output curve, although with different model parameters. This is illustrated in Figure 7 for the first two metamodel dimensions: to avoid graphical cluttering, only a small, random subset of curves was selected in the score plot. Among them were some realisations of the logistic function type; these are marked explicitly. Clearly, the line curvatures produced by the simple logistic function are not unique, at least not in this projection. Given such many-to-few relationships, this is to be expected: for curves displaying simple, symmetric line curvature, a fit to this generic model may yield a range of alternative model types and parameter ranges. However, for curves with clear asymmetry, there are fewer model alternatives, as demonstrated by Isaeva *et al.* [14]. From now on, to avoid confusion, we will refer to the PCs of the metamodel as to meta-PCs.

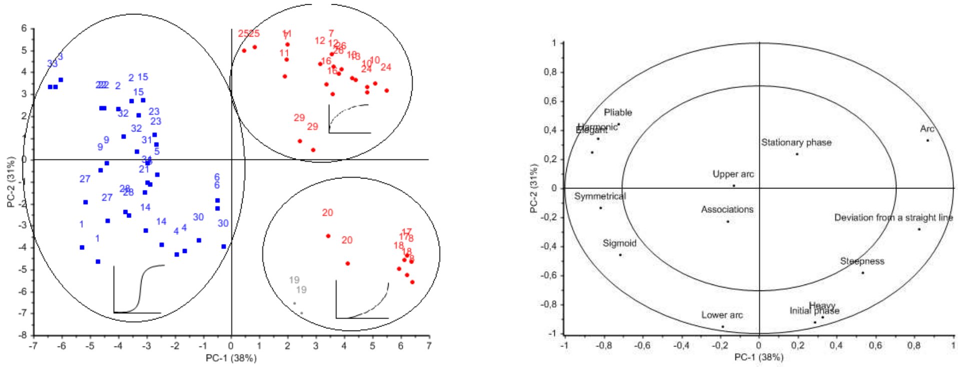
3.2 PCA

To get a primary overview of the data, PCA was run on the values of the sensory variables from the first trial. From the score plot (Figure 8b), one can see that the judges had a high repeatability from one day to another, i.e., scores for the replicates lie close to each other. Moreover, it was noticed that there is a clear grouping of samples along the first PC: the samples divided into ”sigmoid” and ”arc” groups. Second PC made a distinction between curves according to their location with respect to the line $y = x$: for the ”arcs” it was a clear separation of the curves into ”upper” and ”lower” arcs. Third PC (not shown on the figure) divided ”arcs” into those that have at least one long phase (initial or stationary) and those without phases at all. In total, five PCs explained 94.6% of the variability in the data. The model was validated by cross-validation with two segments formed by data from each of the two days. That allowed us to verify how different or how similar evaluations at each day were with respect to each other.

Loadings of the model were logically correlated, e.g., *Sigmoid* was negatively correlated to *Arc*; *Lower Arc* was negatively correlated to *Upper Arc* while positively correlated to *Initial Phase* and *Heavy*; naturally, *steep* curves *deviated* much *from the straight line* and were neither *elegant*, nor *harmonic*, nor *pliable* (Figure 8a).

To check whether the five-component PCA model on the curves from the first run was reasonable or not, sensory data from the second run was projected onto its score space. The variability in the new data was explained up to 82.8%. One might say that this prediction is not perfect and it is not. However, this fact is believed to be due to the presence of the straight line among other curves. Most likely, for the judges it was difficult to evaluate it since they were not trained on such a curve, or/and, as was mentioned above, a straight line is always separated from custom curves in a human mind.

Otherwise, scores for the curves that were present in both evaluations fell really close to each other, which



(a)

(b)

Figure 8: Score (a) and correlation loading (b) plots for the first and second PCs from the PCA model on curves from the first run. The scores are colour-coded according to their shapes: blue squares - sigmoidal curves, red circles - curves with one arc. Curve number 19 is not colour-coded here due to its unclearness: it is a one long arc, although it has a small tail like a sigmoid. The correlation loading plot shows relations between the sensory variables.

tells us that the evaluation was structural, not random, and did not depend on the time of evaluation. Even four months later they were able to give the same scores to the curves. This was also verified by comparison of the means and standard deviations of the sensory evaluation of the common curves of runs 1 and 2 (the results are shown in Figure 9). One can see here that the evaluation of the common curves from the second run for most of the descriptors did not vary significantly in comparison to such from the first run. The only variable that seemed to change much was *Associations*. It is a very unclearly defined descriptor and is rather subjective. Some judges may not associate a curve with anything, whereas others might see a shoulder, a wave or even a belly of a pregnant woman. Therefore, it was decided to eliminate *Associations* from the further analysis.

3.3 PLSR

In order to achieve the aim of finding a relationship between metascores of the curves in the global PCA model and the sensory descriptors, partial least squares regression (PLSR) analysis was implemented here [19]. Metascores for the first 12 meta-PCs for the curves from the first run were considered as predictors (forming a matrix X), whereas the response matrix Y held the sensory values for all the descriptors except *Associations*.

First, a model without interactions and square effects was studied, but this turned out to be too simple to yield good sensory predictions. Many variables were poorly predicted and some of them even showed a presence of nonlinearity, e.g., *Deviation from a straight line*, *Lower arc* and *Steepness*. Obviously, having just main effects (metascores) was not enough to build an appropriate model: only 73.5% of the Y-variability was explained by eight PCs. As a result of this, it was decided to extend the model with the interaction and square effects for the first six metascores-variables (since they were the most significant in the metamodel), so that the X- and Y-matrices were of size 64×33 and 64×13 respectively. Raising to the second power and multiplying values of scores for different meta-PCs with each other might lead to immense values of the new variables (interactions and square terms). Therefore, in order to avoid it, X-variables had to be weighted. Since the scale of the original metascores was different, their weighting was necessary as well. In this way, following weight vectors were imposed on the 12 main effects:

$$w_{main} = \frac{1}{std + 0.1} \tag{10}$$

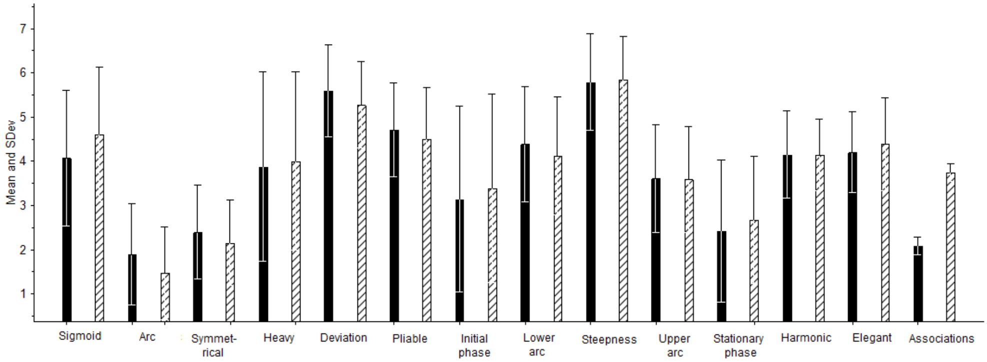


Figure 9: Mean and standard deviation of the sensory evaluation of the 14 common curves that were present in both runs of the sensory analysis: solid black bars correspond to the first run, hatched bars - to the second.

and on interaction and square terms:

$$w_{Ins\_Sqs} = \frac{0.11}{std + 0.1}, \tag{11}$$

where *std* denotes the standard deviation of the given variable. The number 0.1 in the denominator was introduced to prevent an inflation of the effects of the meta-PCs that are defined on a small scale and, therefore, have a small standard deviation; whereas the value 0.11 for the interaction and square terms was obtained by trial and error method aiming at supressing the effect of their large values. As in Section 3.2, a goodness of the model was tested by "leave-one-day-out" cross validation.

Naturally, adding more variables to the model led to an increase in the number of PCs needed: now 12 PCs seemed to be an optimal number, and they explained 89.4% of the Y-variability. It is much better than for the model with only main effects despite the increased complexity of the model. Besides, all the sensory variables were explained noticeably better: the smallest coefficient of determination for prediction r^2 was equal to 0.80 for the descriptor *Deviation from a straight line.* That was most likely due to the fact that some judges did not know how to evaluate steep sigmoids with respect to this attribute: such curves, despite their large deviation from a straight line, still intersect it. The rest of the variables were predicted very well (see examples in Figure 10), that is, the found PLSR model is suitable for explaining the relationship between metascores and sensory values and for prediction of sensory evaluation for new observations.

Projection of the curves from the second run onto this model gave rather good results indicating that the model found captured the main interdependences between sensory descriptors and metascores, and this dependence is nonlinear. However, it was again noticed here that the prediction of the sensory values struggled for the straight lines, meaning that there was a distinction in judges' minds between traditional curves and a straight line.

From the correlation loading plots in Figure 11, one can give interpretations to the meta-PCs by means of the sensory descriptors. It can be noted that the first meta-PC divides curves into "lower" and "upper" curves; the second meta-PC represents a division of curves into "sigmoids" and "arcs", whereas the fourth meta-PC seems to be negatively correlated to *Steepness.* Indeed, it is confirmed by the plots in Figure 12 showing a negative slope for the association between meta-PC-1 and the sensory descriptor *Initial phase*; whereas the descriptor *Sigmoid* has a positive associations to the meta-PC-2.

3.4 Interpretation of function parameters

Interpretation of parameters of various mathematical functions is a difficult task, especially if a function is not elementary. It gets next to impossible if one does not have advanced knowledge in mathematics. That is why, it is extremely important to make it accessible to a more general audience and help them in conducting

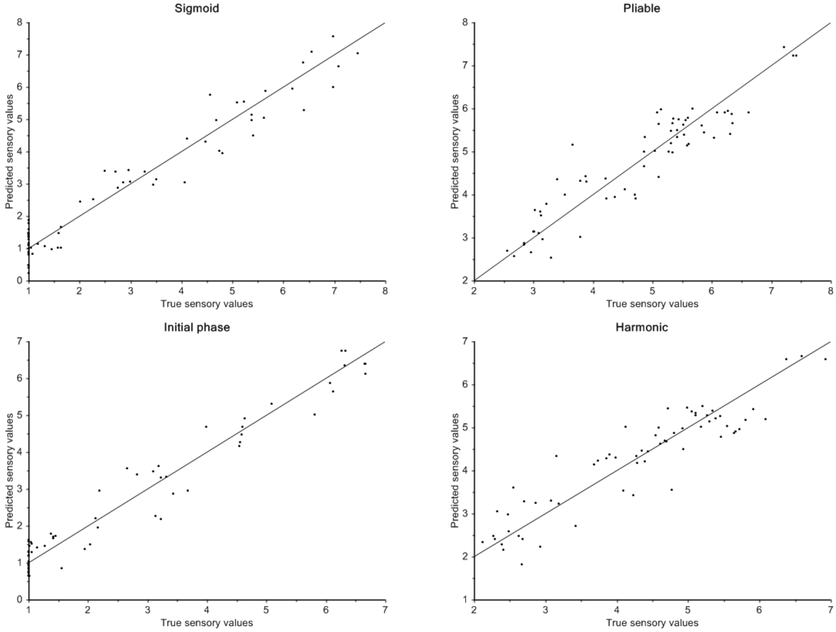
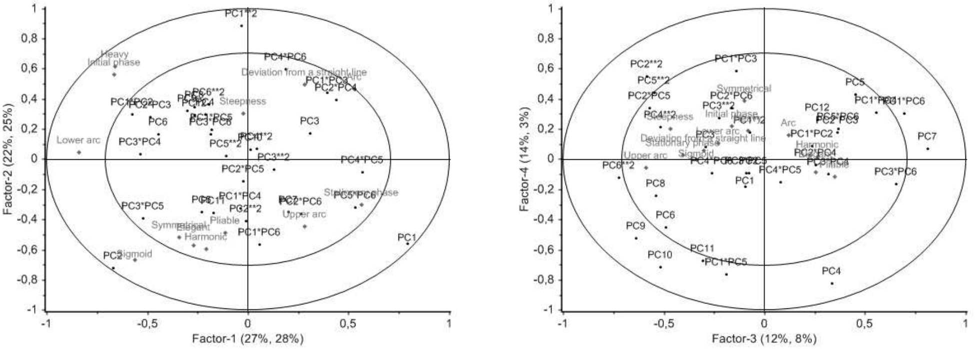


Figure 10: Predicted vs true values of the sensory evaluation for some sensory descriptors (from PLSR model on the curves form the first run). It can be seen that the points lie close to the straight line indicating that the prediction using the found model works reasonably good.



(a)

(b)

Figure 11: Correlation loading plots for the PLSR model for the curves from the first run: *(a)* – first PC vs second PC; *(b)* - third PC vs fourth PC. Here notation ”PC” on the plots correspond to the meta-PCs.

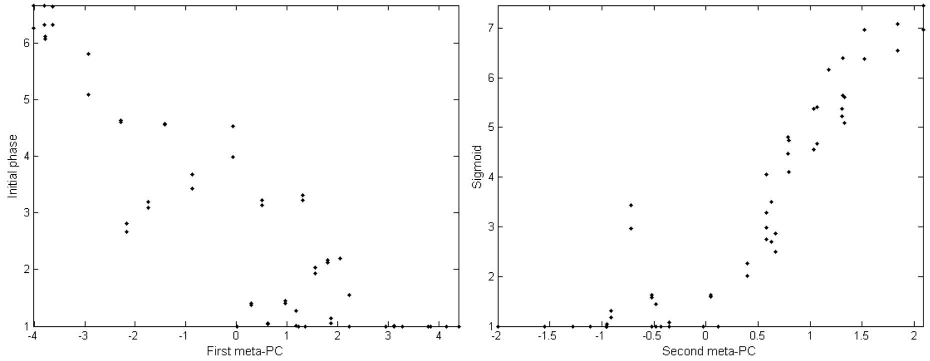


Figure 12: Relationship of the first and second meta-PC to *Initial phase* and *Sigmoid* respectively.

relevant experiments by giving a custom meaning to the complex mathematical parameters. A way to achieve this is to model the relationship between these parameters and the curve descriptors from Section 2.2. Here we show how it works on the example of the logistic and error functions from the modelome in Ref. [13].

First, a PLSR model based on the both runs of the sensory analysis was built. The model from Section 3.3 could, of course, also be used here, but it was decided to use all available data to build a more robust model for predicting new curves. The samples of the straight line were eliminated here due to reasons discussed in Section 3.3. That is, matrices X and Y were of size 124×33 and 124×13 respectively.

First, we study the example with the logistic function. Curves for that function phenome were simulated according to the formula

$$z(x; p_1, p_2) = \frac{1}{1 + \exp(-p_1 x + p_2)}, \tag{12}$$

where the x-interval was set to $[0.001; 1]$, p_1 varied from 0.1 to 10 in steps of 0.2 and p_2 had values from the interval $[-6; 5]$ in steps of 0.5.

The scores for these curves from the global PCA model were projected onto the PLSR model described above, and predictions of sensory values were obtained. The projection was very good explaining 93.3% of the variability of the matrix with the design parameters indicating that the chosen curve set (joint curves from both runs) represents the curvature phenomenon quite extensively. Then, predicted sensory values were used to find a relation to the parameter values for the logistic function. For this purpose, PLSR analysis was run with the predicted sensory values as an X-matrix and the functional design as a Y-matrix. The functional parameters were predicted extremely good with correlation coefficients r^2 equal to 0.93 and 0.94 between true and predicted values of p_1 and p_2, respectively (Figure 13).

Figure 14 shows a correlation loading plot. One can notice here that high values of both p_1 and p_2 will lead to high values of *Sigmoid* and *Symmetrical*, medium and high values of p_1 and p_2 respectively will result in high values of *Heavy* and *Initial phase*, whereas low values of p_2 are associated with high scores for *Deviation from a straight line* and *Arc*. These facts are confirmed by the plots in Figure 15.

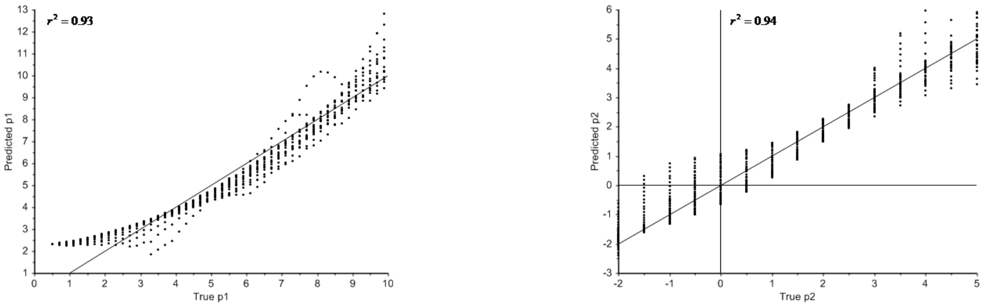
Thus, as a custom interpretation of the mathematical parameters p_1 and p_2, for instance, *Upper arc* and *Initial phase* can be used respectively.

Here prediction of function parameters was made in two steps:

1. Metascores → Sensory values

2. Sensory values → Function parameters

This was done with the purpose to show that function parameters can be interpreted and predicted from a human description of the curves provided a function type. Prediction of parameters directly from the metascores is presented in Ref. [13, 14], although by using local metamodels. Results of employing the global metamodel for this purpose are provided in the supplementary to this paper material.



(a)

(b)

Figure 13: True vs predicted parameter values for the logistic function.

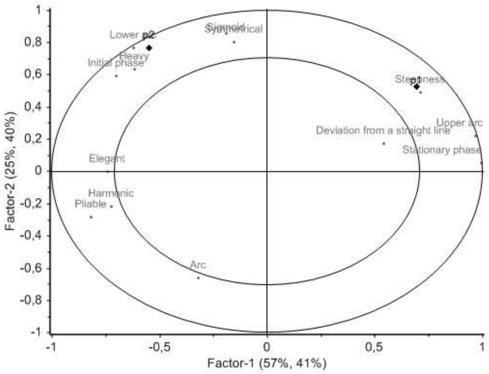


Figure 14: Correlation loading plot for the logistic function. Here black diamonds represent function parameters, whereas grey dots correspond to the sensory descriptors. It is easily seen from the plot how function parameters are associated with the sensory descriptors.

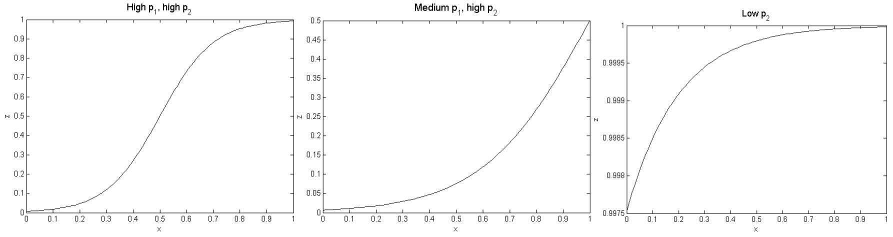


Figure 15: Plots of the logistic function with various parameter values. Shapes of curves are in agreement with the relation found between parameter values for this function and sensory evaluation.

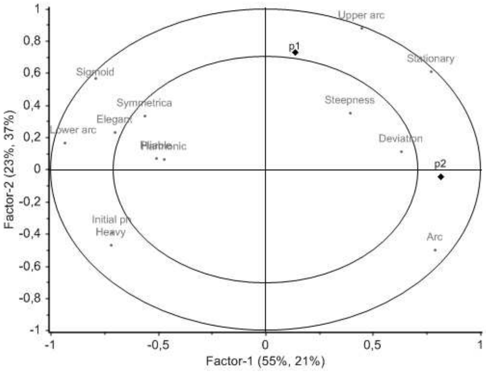


Figure 16: Correlation loading plot for the parameters of the error function and estimated sensory evaluation: black diamonds – function parameters, grey dots – sensory variables.

A similar procedure was also run for the error function:

$$z(x; p_1, p_2) = \frac{2}{\sqrt{\pi}} \int\limits_{0}^{p_1 x + p_2} \exp(-t^2) dt, \tag{13}$$

where x, as before, varied from 0.001 to 1; $p_1 \in [0.5 : 10]$ and $p_2 \in [-5 : 3]$ in steps of 0.2 for both. The error function is used in statistics and gives the probability that a measurement error will have a distance less than $p_1 x + p_2$ to the average value [20]. It is, in fact, the integral of the Gaussian curve and is often referred to as the cumulative Gaussian function. As can be seen from Eq. (13), the mathematical expression of this function looks rather complicated and intimidating for non-mathematicians (integral sign, exponential function, squaring). However, there was no problem to relate those parameters to the sensory descriptors by a PLSR model. Both parameters were predicted very well with the correlation coefficient between true and predicted values equal to 0.96 for each of them. Associations of p_1 and p_2 with the sensory variables can be seen in Figure 16. Here one can observe that with the increment of p_1 (parameter of the increasing speed), a curve becomes steeper since it grows faster; increase of p_2 gives a curve with a larger shift to the left and, consequently, larger deviation from a straight line, and reverse, decrease of p_2 (shift to the right) brings *Initial phase* to a curve and makes it *Heavy*.

4 Discussion

At a detailed mechanistic level, individual processes and relationships in nature can be understood by established physical laws which are described mathematically. These can, in turn, be combined into mathematical models summarising more complex systems, based on theory and evidence. Mechanistic mathematical modelling, based on sound physical principles, can reveal many phenomena about our reality, e.g., in biology, namely, how different factors of various processes influence each other.

But mechanistic modelling of complex systems has a strong subjective or inter-subjective component, for better or worse: it provides elegant, compact representation of otherwise overwhelming complexities, and, thus, simplifies thinking and communication. But it also has the danger of codifying incomplete or erroneous assumption to the extent that they become hard to correct. Consistently using an erroneous mechanistic model may work functionally to produce line curvature, but it may hamper the user's analysis and intuition. This paper, therefore, provides a precise, but pragmatic alternative to mechanistic modelling of line curvature, which gives the same functionality but with a less need for mechanistic assumptions.

This metamodelling can also have advantages in subsequent curve fitting; since the metamodel consists of linear projection followed by local mapping, the parameter estimation does not require an iterative search process.

More generally, we have focused on modelling a generic phenomenon line curvature and shown that, in spite of its wide range and high number of nonlinear model alternatives, the phenomenon *per se* has a rather low dimensionality.

Finally, the bi-linear metamodel dimensions themselves just represent an orthogonal axis system, and there is a need for naming the structure in this subspace, at least with respect to its main dimensions. We have, therefore, studied the phenomenon of nonlinearity in terms of the custom human language. The reason for that was a complexity of nonlinear systems and difficulties of their understanding by bio-scientists. Nonlinear models can be broadly seen in the world phenomena and are studied by a wide scientific society. Quite often it causes troubles even for mathematicians and physicists, who are used to such a complexity, to find out what each parameter in a formula stands for. There is no need to mention then what kind of a problem it is for bio-scientists to do that without advanced mathematical knowledge. The purpose of this paper was to show that this tangled phenomenon of nonlinearity can be related to simple words used in the everyday life. Here only nonlinearity on a plane was studied, that is, the main object was a variety of curves obtained by simulations of different mathematical functions [13].

First of all, a global metamodel of curvature was built based on the modelome from Ref. [13]. It has been shown that modelling a model gives a noticeable reduction of complexity and dimensionality. There was found a basis of 12 curves whose linear combinations would provide a large variety of curves.

Secondly, a sensory panel was set up to study a human perception of curves. The analysis was performed on two sets of curves, 32 curves in each. PCA on the results showed that the judges were able to reproduce their evaluation days and even months later indicating that they are well trained (similar scores for identical curves). It was shown that, by having a nonlinear model (containing interactions and square terms along with the main effects), it is possible to predict sensory values for a new curve (with a PLSR model). This is a big step towards reducing a gap between mathematics and non-mathematically oriented people. It is shown that it is indeed possible to describe outputs of mathematical expressions by the custom language, and this description is not random.

Appearance in the second run of the new models that were not presented in the modelome did not give any protruding results in the model construction. Their scores fell to the space as those for the old models, indicating thereby that the existing modelome captures so many curve types that any new curve will be inside the modelome's score-space.

When building a PLSR model, there were some problems with modelling straight line samples. It was seen that, even though a straight line is defined to be a curve as well, a human mind still distinguishes it from a traditional curve. Apparently, the judges had troubles with evaluation of the straight lines presented in the curves set from the second run. It was difficult to model those samples, and their elimination led to a better predictive ability of the model. Perhaps, if straight line samples had been included into the set of the curves for the first run, the results would have been different since judges would have trained themselves in evaluation of this critical form of curves.

With the obtained PLSR model (on curves from the first run) it became possible to give meaning to the meta-PCs. Before, they were only linear combinations of the original curves making a basis of the modelome. Now, by relating the metamodel to the sensory descriptors, we know what each meta-PC is responsible for.

Yet another important result of this work is the ability to map function parameters into the sensory attributes. It can enable many experimentalists to conduct their experiments better since they will know beforehand what property of a curve every parameter corresponds to. And since this knowledge does not require advanced mathematical skills, it is accessible to a wide audience. However, this has to be done carefully. This linkage of function parameters to the sensory descriptors involves projection of new samples onto the built PLSR model. It can well happen that scores for the new curves are lying close to the edge of the range tested for this model. In that case the distance to the centre of the model may be quite large and the predictions for these samples may be expected to be more uncertain, due to statistical estimation errors of various kinds.

It was shown that the relation of sensory evaluation of curves to their function parameters can be modelled by the PLSR. This can possibly be applied for estimation of function parameters from values of the sensory descriptors; namely, by knowing "grades" of a curve for all the sensory variables and given a function type, it would be possible to predict values for the parameters.

The obtained results might also be used in the future for comparing two incompatible functions that seem to be absolutely different in terms of mathematical expressions and may even have a different number of parameters, but have the same type of curves. It should be possible to find out by using established models what are the similarities of given functions and what are the differences.

To sum up, we have discussed the generic phenomenon of nonlinearity (in particular curvature) from a new perspective, namely building of its metamodel and the process of interpretation of this phenomenon by people not familiar with advanced mathematical notions and terms. We believe that this will contribute to make communication between different scientific communities easier.

Acknowledgements

Josefine Skaret and Jannike Olavesen are thanked for a great help in organising and conducting the sensory analysis. Kristin Tøndel is thanked for help with the metamodelling, and Stig Omholt is thanked for useful criticism of the manuscript. The Levy for research Agricultural Products is thanked for economic support. John A. Wyller wishes to thank the School of Mathematical Sciences, University of Nottingham for the kind hospitality during the stay. This research was supported by the Norwegian University of Life Sciences. The work has also been supported by The Research Council of Norway under the grant No. 178892 (eNEURO-multilevel modelling and simulation of the nervous system) and the grant No. 178901 (Bridging the gap: disclosure, understanding and exploitation of the genotype-phenotype map).

References

[1] M. Hazewinkel. *Encyclopaedia of Mathematics: An updated and annotated translation of the "Soviet Mathematical Encyclopaedia"*, volume 5. Kluwer Academic Pub, 1990.

[2] R. Arnheim. *Art and Visual Perception: A Psychology of the Creative Eye*. Univ of California Press, 1954.

[3] S. Yantis. *Visual Perception: Essential Readings*. Psychology Pr, 2001.

[4] R.C. Yates. *Curves and Their Properties*. National Council of Teachers of Mathematics, Inc., 1906 Association Drive, Reston, Virginia 22091, 1974.

[5] J.R. Piggott. *Sensory Analysis of Foods*. Number Ed. 2. Elsevier Applied Science Publishers Ltd, 1988.

[6] J.M. Murray, C.M. Delahunty, and IA Baxter. Descriptive sensory analysis: past, present and future. *Food Research International*, 34(6):461–471, 2001.

[7] H. Martens, S.R. Veflingstad, E. Plahte, M. Martens, D. Bertrand, and S.W. Omholt. The genotype-phenotype relationship in multicellular pattern-generating models - the neglected role of pattern descriptors. *BMC Systems Biology*, 3(1):87, 2009.

[8] M. Martens, S.R. Veflingstad, E. Plahte, D. Bertrand, and H. Martens. A sensory scientific approach to visual pattern recognition of complex biological systems. *Food Quality and Preference*, 21(8):977–986, 2010.

[9] J. Isaeva, S. Sæbø, J.A. Wyller, K.H. Liland, E.M. Faergestad, R. Bro, and H. Martens. Using GEMANOVA to explore the pattern generating properties of the Delta-Notch model. *J. Chemom.*, 24:626–634, 2010.

[10] B. Crassini and R. Over. Masking, aftereffect, and illusion in visual perception of curvature. *Attention, Perception, & Psychophysics*, 17(4):411–416, 1975.

[11] P.O. Bishop and G.H. Henry. Striate neurons: receptive field concepts. *Investigative Ophthalmology & Visual Science*, 11(5):346, 1972.

[12] J.P.C. Kleijnen. *Design and Analysis of Simulation Experiments*. Springer Verlag, 2007.

[13] J. Isaeva, S. Sæbø, J.A. Wyller, O. Wolkenhauer, and H. Martens. Nonlinear modelling of curvature by bi-linear metamodelling. *Chemometrics and Intelligent Laboratory Systems*, doi: 10.1016/j.chemolab.2011.04.010, 2011.

[14] J. Isaeva, S. Sæbø, J.A. Wyller, S. Nhek, and H. Martens. Fast and comprehensive fitting of complex mathematical models to massive amounts of empirical data. *Chemometrics and Intelligent Laboratory Systems*, doi: 10.1016/j.chemolab.2011.04.009, 2011.

[15] H. Martens, I. Måge, K. Tøndel, J. Isaeva, A. Gjuvsland, M. Høy, and S. Sæbø. Multi-level binary replacement (MBR) design for computer experiments in high-dimensional nonlinear systems. *J. Chemom.*, 24:748–756, 2010.

[16] K. Tøndel, A. Gjuvsland, I. Måge, and H. Martens. Screening design for computer experiments: Metamodelling of a deterministic mathematical model of the mammalian circadian clock. *J. Chemom.*, 24:738–747, 2010.

[17] L.E. Krueger. Reconciling Fechner and Stevens: Toward a unified psychophysical law. *Behavioral and Brain Sciences*, 12(02):251–267, 1989.

[18] A.B. Gjuvsland, E. Plahte, and S.W. Omholt. Threshold-dominated regulation hides genetic variation in gene expression networks. *BMC Systems Biology*, 1(1):57, 2007.

[19] H. Martens and T. Naes. *Multivariate Calibration.* John Wiley & Sons Inc, 1989.

[20] N.L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Univariate Distributions*, volume 1. John Wiley & Sons, 2nd edition, 1994.

Paper V

Special Issue Article

Journal of CHEMOMETRICS

Received: 4 March 2010, Revised: 8 October 2010, Accepted: 17 October 2010, Published online in Wiley Online Library: 2010

(wileyonlinelibrary.com) DOI: 10.1002/cem.1366

Multi-level binary replacement (MBR) design for computer experiments in high-dimensional nonlinear systems

Harald Martens[a]*, Ingrid Måge[b], Kristin Tøndel[c], Julia Isaeva[d], Martin Høy[b] and Solve Sæbø[d]

Computer experiments are useful for studying a complex system, e.g. a high-dimensional nonlinear mathematical model of a biological or physical system. Based on the simulation results, an empirical "metamodel" may then be developed, emulating the behavior of the model in a way that is faster to compute and easier to understand. In modelometrics, the model phenome of a computer model is recorded, once and for all, by structured simulations according to a factorial design in the model inputs, and with high-dimensional profiling of its simulation outputs. A multivariate metamodel is then developed, by multivariate analysis of the input–output data, akin to how high-dimensional data are analyzed in chemometrics. To reveal strongly nonlinear input–output relationships, the factorial design must probe the design space at many different levels for each of the many input factors. A reduced factorial design method may be required if combinatorial explosion is to be avoided. In the multi-level binary replacement (MBR) design the levels of each input factor are represented as binary numbers, and all the individual binary factor bits are then combined in a fractional factorial (FF) design. The experiment size can thereby be greatly reduced at the price of some binary confounding. The MBR method is here described and then illustrated for the optimization of a nonlinear model of a microbiological growth curve with five design factors, for finding the relevant region in the design space, and subsequently for estimating the optimal design points in that space. Copyright © 2010 John Wiley & Sons, Ltd.

Keywords: binary replacement; combinatorial explosion; computer experiment; fractional factorial; multi-level design

1. INTRODUCTION

1.1. The importance of computer experiments

Computer models are increasingly used in many fields of science, ranging from global climate assessment, weather forecasting and process control, via design of mechanical devices in industry and defense, to quantum physics and the representation of genomic and metabolic regulation in systems biology and medicine. To represent a complex system for a given purpose, a "mechanistic" computer model is usually built in a bottom-up fashion, combining computational elements that mimic the individual mechanism thought to control the compositional, spatial, and temporal behavior of the system. Different types of computer models are used, involving nonlinear finite elements, cellular automata, or coupled nonlinear ordinary and partial differential equations. Irrespectively, such models usually have a number of inputs and can yield a number of outputs. Due to nonlinear feedback and sheer dimensionality, it is often difficult for scientists to assess the properties of such a model theoretically, e.g. to predict how variations in its inputs will affect its outputs or to foresee unexpected patterns of behavior or unexpected computational problems. For a complicated computer model, important properties therefore remain unknown to the user. And it can have dire consequences for the practical use of a computer model if the range in which its input gives acceptable model behavior without computational pitfalls is unknown. Likewise, not knowing which parameter values correspond to real-world conditions can make computer simulations misleading.

However, the behavior of a complex computer model may be studied empirically in computer experiments. An ideal computer experiment is a set of simulations that reveals how the model behaves under—more or less—all relevant input conditions. When each simulation is computationally demanding, it is necessary to reduce the size of the computer experiment. A structured experimental plan must then be employed, in order to

* *Correspondence to: H. Martens, Centre for Integrative Genetics (CIGENE), Dept. of Mathematical Sciences and Technology, Norwegian University of Life Sciences, P.O. Box 5003, N-1432 Ås, Norway.*
E-mail: harald.martens@umb.no

a *H. Martens*
CIGENE, Dept. of Mathematical Sciences and Technology, Norwegian University of Life Sciences, Norway

b *I. Måge, M. Høy*
Nofima Mat, N-1430 Ås, Norway

c *K. Tøndel*
CIGENE, Dept. of Mathematical Sciences and Technology, Norwegian University of Life Science, N-1432, Norway

d *J. Isaeva, S. Sæbø*
Institute of Chemistry, Biotechnology and Food Science, Norwegian University of Life Sciences, N-1432, Norway

get as much insight as possible with as few simulations as possible. That is the purpose of the statistical design method presented here. But before describing the new method itself, the context for which it is intended needs to be outlined.

1.2. Model phenomes and multivariate metamodels

For simulation with a computer model, the inputs, which will be controlled by the experimental design, may typically consist of model parameter values and/or initial values of state variables. The corresponding outputs may be of various kinds, but should be sufficiently informative to describe all the potentially important aspects of the model behavior. The output profile may e.g. consist of the final values of all the important computed state variables and their systematic temporal and spatial distribution patterns, in addition to high-level result summaries (e.g. cycling time) and descriptions of the simulation process itself (e.g. convergence rate).

An experimental design for studying a computer model defines where to probe a system in its input design space[1]. Not knowing the outcome from the experiment, one cannot expect to make a perfect choice of experimental design. But the design should reflect the investigator's prior knowledge or beliefs about the model, the range of simulation conditions expected to be of highest interest, the way each of the simulations is going to be characterized and how these outputs are going to be stored and subsequently analyzed.

The set of recorded outputs from designed computer experiments with a given computer model is here termed the "model phenome." It represents all the behaviors of the model, up to the chosen resolution of the inputs and the outputs. The model phenome may be considered to contain the same information as the computer model itself—but in a different domain and only to the said resolution. For a given computer model, the model phenome is established once and for all. Later, it may be used e.g. as a look-up table, to speed up the way the outputs are obtained from new inputs.

However, for this mass of data to be interpretable and validated in practice, statistical data analysis is required. A data model that links the inputs and the outputs of a computer model is called a "metamodel"[1]. Computer experiments with a high number of input- and output-variables call for multivariate data modeling, i.e. "multivariate metamodeling." Many of the output variables must be expected to be intercorrelated. The multivariate metamodeling then requires regression methods that handle collinearities, to simplify interpretation and to stabilize the parameter estimation. Since pre-processing and regression methods developed in the field of chemometrics and other '-metrics' fields appear particularly suitable for multivariate metamodeling, we here coin the term "modelometrics" for representing the multivariate metamodeling by typical chemometric methods.

1.3. Testing combinations of too many levels of too many factors?

In many complex computer models, representing e.g. living systems, the response variables may be expected to be strongly nonlinearly related to each other and to the input variables: In a certain parameter range, small changes in an input parameter may cause particularly large changes in computed output phenotypes. To be prepared for unknown, but possibly abrupt nonlinearities in the model behavior, it is important to be able to study many levels of each factor. For instance, in order to emulate a cumbersome nonlinear mathematical description of an aberration in infrared spectroscopy of individual cancer cells, Kohler *et al*.[2,3] sampled a certain optical parameter densely in order to develop a fast and simple PCA/EMSC-based pre-processing metamodel to render the infrared spectra interpretable.

On the other hand, it is usually important that the experimental design allows testing of many different combinations of the controllable input factors. This is traditionally attained by factorial experimental designs. For instance, Martens[4] analyzed time series data, obtained by computer simulation according to a full-factorial design, by nominal-level PLS regression, to identify and quantify various feedback structures in dynamic models of simple, but nonlinear regulatory systems.

For systems where each run is expensive, either due to high computational load or due to high cost of subsequent output characterization, it is important to reduce the number of runs, *N*, as much as possible. For instance, the behavioral repertoire of a dynamic, nonlinear, and spatially high-dimensional model of cell differentiation was studied[5], using a reduced simulation design: To be able to detect and quantify even unexpected patterns, sensory descriptive analysis of computer simulation outputs was employed, submitting paper print-outs of the computed cell patterns to the sensory panelists. To make this assessment cost-effective, a standard fractional factorial (FF) design (see below) was employed, combining $K = 7$ input parameters at two levels each, in a total of $N = 32$ runs. The input–output maps generated by PLS regression led to the discovery of new and unexpected pattern types that had not been foreseen by theoretical mathematical analysis of the dynamic model. However, probing only two levels of each factor was found to be an undesired limitation, given the locally nonlinear nature of the model behavior. Like in many other systems, the topology of the input–output map of the computer model proved later to be so complicated that particularly nonlinear model behavior only occurred at certain combinations of certain levels of the input variables.

To be prepared to detect unknown, locally nonlinear input–output topologies, all relevant input factors should ideally be tested at many levels each, and in all possible combinations. But this creates combinatorial explosion. For instance, even in a small system with only $K = 5$ input parameters, a full factorial design testing all combinations of the *K* design factors, at $M = 8$ levels each, would require $N = 32\ 768$ experiments. Complicated computer models of real-world relevance may easily have *K* equaling between 10 and 100 input parameters, whose effects need to be assessed at many levels, alone and in combination.

1.4. Factorial design of experiments

Unless an efficient Design Of Experiments (DOE) method is employed, multi-level multi-factor factorial designs make computer experiments in complex models studies prohibitively expensive. But till now, reduced design methods for cost-effective, but systematic testing of combinations of many factors at many levels are not well known, at least not within the fields of systems biology and chemometrics.

The general research on DOE dates back to the work of Fisher[6] in 1926, and since then, this important statistical issue has been studied and developed further by a long line of authors. The research on experimental design problems had a golden age in the 1960s and 1970s with numerous published papers, especially in the journal *Technometrics*. An excellent review of the achievements of this period was written by Steinberg and Hunter[7] in 1984. They present the development of experimental design from the early agricultural experiments with qualitative factors in the agricultural tradition, to later specialized designs such as response surface designs[8–10] and mixture designs[11] for continuous factors in chemical experiments.

Much effort has been put into constructing response surface designs, which were developed to be optimal with regard to certain optimality criteria (e.g. D- and G-optimality, resolution, and minimum aberration). However, these optimal designs have received criticism because of the apparent sensitivity of the optimality properties to the choice of the approximation model used for summarizing the results—in this case a surface model such as a second-order polynomial with interactions and square terms. Of course, higher-order polynomials may be more accurately estimated if a larger number of levels are run for each factor.

The problem of combinatorial explosion in multi-factor factorial designs was recognized already by Finney[12] in 1945, who introduced the concept of fractional factorial (FF) designs. In subsequent years the fractional designs were further developed, but mainly for two-level factorial designs. The so-called 2^{K-P} fractional factorial designs are powerful tools for investigating the main effects of factors, but at the expense of losing the possibility of assessing higher-order effects. If high-order response surface models are of main interest though, the two-level fractional designs are less appropriate unless a sequential experimental strategy is adopted. When it comes to experiments with many factors, each measured at multiple levels, some asymmetrical $4 \times 2^{K-P}$ FF designs are available, but Montgomery[13,14] advises that such designs should be used with caution, firstly, because central composite designs may be more optimal, and secondly, because the number of runs necessary for obtaining designs of minimum resolution IV is relatively high. However, as discussed above, the optimality properties may be sensitive to model choice, and new and more efficient measurement technologies may make experiments with relatively many runs more feasible. Computer experiments based on designed computer simulations, represent one such technology.

For designing computer simulation studies, Simpson *et al.*[15] reviewed literature and compared four sampling strategies: Latin hypercubes, Hammersley sequence sampling, orthogonal arrays, and uniform designs. The uniform designs may be described, for continuous design factors, as a type of FF design with an added uniformity property, akin to Latin hypercubes, but with n-dimensional uniformity.

We here present a similar approach, the 2^{K-P} FF designs with the so-called replacement method. The basic concept was introduced by Adelman[16] as early as in 1962, and has later been acknowledged as the “replacement method” in design literature. However, the original replacement method has its main purpose as a step toward constructing orthogonal asymmetric designs, e.g. the 3×2^k design. Our replacement method has another application, as a tool for constructing fractional designs recoded into multilevel factorial designs. As implemented here, we term the method *Multi-factor Binary Replacement* (MBR). The method represents a combination of elements from statistical design theory (FF design) and from signal processing (binary number representation), both having well-known theoretical properties.

1.5. Overview of this paper

The MBR design method will here be outlined and illustrated in a simple example involving a nonlinear system with five design factors:

1.5.1. *Initial range finding: where is the relevant search region?*

The MBR design will first be used for *initial range finding*. A problem in DOE, as mentioned by Steinberg and Hunter[7], is the necessity of initially defining the interesting region in the design space—usually defined by the experimental range for each factor. The region of interest is usually not known *a priori*, and the bias for the chosen model may become large if the experimenter chooses too extreme limits for the factors in order to accommodate for this unknown region problem. The reason is that the optimal designs tend to place many experimental runs at the extremes of the chosen region. At present there is no general method in classical design theory for initially finding the relevant ranges of design factors within which to apply the DOE; that is usually left to the domain expert, who often finds it difficult to balance the need to avoid irrelevant extremes against the need for spanning the design space.

The purpose of the present range finding experiment is to show how the MBR design can be used for finding transitions, possibly abrupt, which delineates the unknown region of interest in the design space from irrelevant regions outside. To find sharp limits between interesting and uninteresting regions, a high number (16) of levels for each of the five design factors will be tested. Still, to lower the risk of wasting resources on conditions found to be irrelevant, a reduced design (64 runs) will be used.

1.5.2. *Final optimization finding: Where is the optimal system behavior?*

Once the region of interest in the design space has been broadly identified based on relatively few runs, a new, more detailed experiment may be planned, in a more narrow design range. The MBR design method, although with different settings, will here be employed instead of more classical response surface designs: The purpose is now to show how the MBR design may be used for nonlinear response surface estimation—finding the point in the design space that is optimal with respect to a certain criterion. Since we now expect a simpler response, for which a smoother surface may be fitted, a lower resolution (8) for each factor is now accepted in the design.

In a follow-up paper, Tøndel *et al.*[17] optimize the MBR design method and compare it to some alternative design methods and apply it for efficient metamodeling of a high-dimensional nonlinear dynamic model from systems biology.

2. METHODS

2.1. The multi-level binary replacement design

The MBR design method combines binary recoding of multi-level design factor levels with fractional factorial designs in binary variables. Assume that a complex, unknown system has K quantitative design factors $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_k, \ldots \mathbf{x}_K]$, whose effects on the system are to be studied, individually and in combination. Some or all of these design factors need to be assessed at a number of levels $L(k)$, $k = 1,2,\ldots,K$, in order to reveal abrupt input–output changes.

If abrupt interaction effects are also to be revealed, factorial designs are required, at different levels of spatial resolution in the design space. With conventional full-factorial design of all factors at all levels, this would create combinatorial explosion: For a system with, for instance, $K = 3$ factors, a full factorial design at 16 levels each would require $N = 16^3 = 4096$ runs. For $K = 5$ factors, a full factorial design at 8 levels each, would require $N = 8^5 = 32\,768$ runs. The MBR design method, as used here, reduces these two experiments to $N = 2^6 = 64$ and $N = 2^5 = 32$ runs, respectively. Figure 1 illustrates the MBR design, which will now be defined:

2.2. Binary replacement of multi-level factors

With little loss of generality, the number of levels for each factor k, $L(k)$, may be chosen to be a multiple of 2; $L(k) = 2^{M(k)}$, e.g. 2, 4, 8, 16, ... For each quantitative design factor $\mathbf{x}_k$, its actual quantitative levels may take any values, at even (e.g. 0, 5, 10, 15, 20,...) or uneven (e.g. 0, 3, 4, 7, 10, 12, 14) steps. As usual, a factorial design is most easily attained if each design factor $\mathbf{x}_k$ is mapped into a decimal indexing variable $\mathbf{d}_k$ with equally spaced steps. The MBR design, the indexing representation of each recoded factor $\mathbf{d}_k$, is then further recoded into a binary (modulus 2) variable $\mathbf{f}_k$ ($N \times 1$) which has $M(k)$ factor bits $[\mathbf{f}_{k,1}, \ldots, \mathbf{f}_{k,M(k)}]$. For instance, if $\mathbf{d}_k$ has 8 levels (0, 1, 2, 3, 4, 5, 6, 7), this yields $M(k) = 3$ binary variables, each with two values (0 or 1), but representing (0 or 4), (0 or 2) and (0 or 1). A value $d_{i,k} = 5$ is thus written $f_{i,k} = 101$, yielding 3 individual factor bits $[f_{i,k,1}, \ldots, f_{i,k,M(k)}] = [1,0,1]$, and represents the 6th level of factor $\mathbf{x}_k$ Hence, the factor bits $[\mathbf{f}_{k,1}, \ldots, \mathbf{f}_{k,M(k)}]$, the binary variable $\mathbf{f}_k$, the indexing variable $\mathbf{d}_k$, and the original quantitative design factor $\mathbf{x}_k$ are equivalent representations of a given design factor:

$$
\begin{gathered}
\mathbf{d}_k : x_k = x_{d_k+1} \\
\mathbf{f}_k = \mathrm{mod2}(\mathbf{d}_k) \\
[\mathbf{f}_{k,1}, \mathbf{f}_{k,2}, \ldots, \mathbf{f}_{k,M(k)}] = \mathrm{bits}(\mathbf{f}_k) \\
\text{i.e.} \\
\mathbf{d}_k = \sum_{m=1}^{M(k)} 2^{m-1} \cdot \mathbf{f}_{k,m}
\end{gathered}
\qquad (1)
$$

The $M(k)$ bits in the binary factor $\mathbf{f}_k$ determine the granularity of the design: They allow us to probe different spatial resolutions in the design factor space for that factor k. With K multi-level factors $\mathbf{x}_1 \ldots, \mathbf{x}_k, \ldots, \mathbf{x}_K$ to be investigated, the total number of binary replacement factors is $\mathrm{M}_{\mathrm{tot}} = \sum_{k=1}^{K} \mathrm{M}(k)$.

2.3. Fractional factorial design in the binary replacement factors

In accordance with standard procedures in FF design, the factor bits $[\mathbf{f}_{k,1}, \ldots, \mathbf{f}_{k,M(k)}]$, each with values 0 or 1, are recoded into two-level replacement design factors $[\mathbf{g}_{k,1}, \mathbf{g}_{k,2}, \ldots, \mathbf{g}_{k,M(k)}]$ with

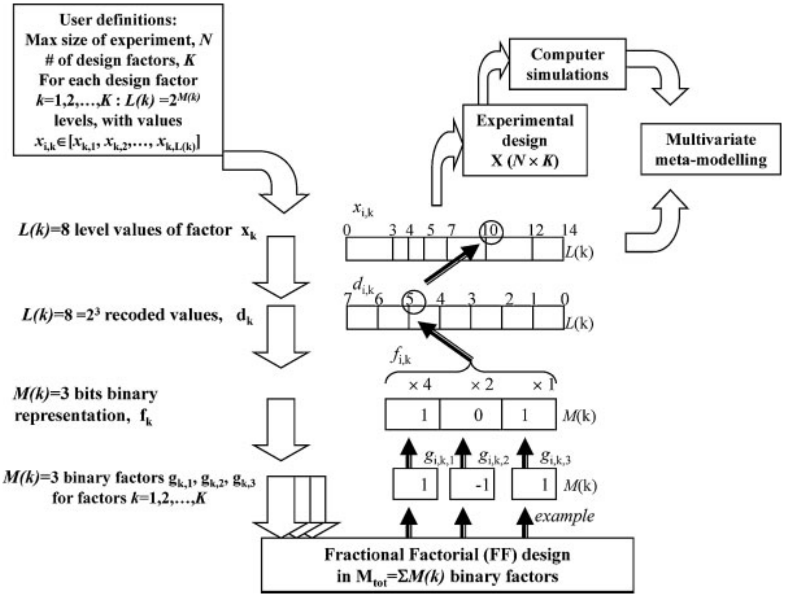

User definitions:
Max size of experiment, N
# of design factors, K
For each design factor
k=1,2,…,K : L(k) =2^M(k)
levels, with values
x_i,k ∈ [x_k,1, x_k,2,…, x_k,L(k)]
Computer simulations
Experimental design
X (N × K)
Multivariate meta-modelling
L(k)=8 level values of factor x_k
x_i,k
0 3 4 5 7 10 12 14
L(k)
L(k)=8 =2³ recoded values, d_k
d_i,k
7 6 5 4 3 2 1 0
L(k)
f_i,k
× 4 × 2 × 1
1 0 1
M(k)=3 bits binary representation, f_k
M(k)
g_i,k,1 g_i,k,2 g_i,k,3
1 -1 1
M(k)
M(k)=3 binary factors g_k,1, g_k,2, g_k,3 for factors k=1,2,…,K
example
Fractional Factorial (FF) design
in M_tot=ΣM(k) binary factors

Figure 1. Multi-level binary replacement (MBR) design for computer experiments. From user definitions of the size of the design, N, and the possible levels for each design factor, each design factor $\mathbf{x}_k, k = 1,2,\ldots,K$ is represented as binary numbers with M_k bits with values 0 or 1. The individual bits are recoded to values -1 or 1 and submitted to fractional factorial design with $M_{\mathrm{tot}} = \Sigma \mathrm{M}_k$ binary factors. From the obtained FF design, the M(k) bits for each individual design factor are then decoded and recombined to yield the quantitative value of the factor $\mathbf{x}_k$. The resulting experimental design is employed as input to experiments and mapped to the resulting outputs by multivariate data modeling.

values −1 or +1:

$$\left[g_{k,1}, g_{k,2}, \ldots, g_{k,M(k)}\right] = \left[f_{k,1}, f_{k,2}, \ldots, f_{k,M(k)}\right] \times 2 - 1 \quad (2)$$

Hence, $\mathbf{X}(N \times K) = [\mathbf{x}_1, \ldots, \mathbf{x}_K]$, $\mathbf{D}(N \times K) = [\mathbf{d}_1, \ldots, \mathbf{d}_K]$, $\mathbf{F}(N \times M_{tot}) = [\mathbf{f}_{1,1} \ldots, \mathbf{f}_{1,M(1)}, \ldots, \mathbf{f}_{K,1} \ldots, \mathbf{f}_{K,M(K)}]$ and $\mathbf{G}(N \times M_{tot}) = [\mathbf{g}_{1,1} \ldots, \mathbf{g}_{1,M(1)}, \ldots, \mathbf{g}_{K,1} \ldots, \mathbf{g}_{K,M(K)}]$ are equivalent representations of the design.

A full-factor experiment in **G** would still require $2^{M(\text{tot})}$ runs, since the full replacement design now has *M* (tot) design factors, each at two levels, creating $2^{M(\text{tot})}$ design dimensions—main effects, two-factor interactions, three-factor interactions, etc. The design size reduction is attained by applying standard fractional factorial (FF) design to the *M(*tot*)* replacement design factors in **G**, choosing only a reduced set of *M(*ind*)* design dimensions to vary independently of each other. This greatly limits the number of runs in the design, *N*. But this comes at a price: The remaining *M(*conf*)* = *M(*tot*)* − *M(*ind*)* dimensions are *confounded*—set to vary together with combinations of other dimensions. This procedure results in a $2^{M(\text{tot})-M(\text{conf})}$ FF design. The effect of this is that, in the end, all *M(*tot*)* individual binary design factors are confounded with one or more higher-order interactions, each being a product of one or more of other binary design factors.

2.4. Confounding strategies

Depending on the chosen design resolution, each of the two-level design factors $\mathbf{g}_{k,m}$ will thereby be confounded with higher-order interactions of other two-level factors. However, there are hard choices to be made setting up such a confounding pattern.

The experimental design should be optimized and assessed in light of its intended use. Depending on the number of quantitative design factors and their chosen resolutions *L(k),k* = 1, 2, ..., *K*, there are many different ways to define a quality criterion for optimizing the FF confounding pattern. For systems with smoothly changing design responses, it is most important to distinguish simple main effects and two-factor interactions from each other; confoundings with more unlikely higher-order interactions are less damaging. In classical FF design with only *K* binary design variables, this is attained by choosing a design with maximum resolution at the given number of *N* and *K*. For instance, a design resolution III confounds the main effects with two-factor and higher-order interactions, while V confounds the main effects only with four- and higher-order interactions. Among the many possible founding patterns, one would select one with resolution V, not with IV or III, if possible, given *N*.

However, in MBR designs we have many more binary design variables in **G** because we expect the responses to change abruptly with small changes in a design factor $\mathbf{x}_k$, and differently so at different levels of other design factors $\mathbf{x}_{k^* \neq k}$. Hence, the classical FF resolution concept is not necessarily applicable. Since the different bit factors now represent the different spatial resolutions of the different factors $\mathbf{x}_k$ in the design space, many different quality criteria may be envisioned. In general, they should now be based on the values of quantitative design factors in **X**, not on the binary replacement factors themselves.

For simplicity, we here generated a number of alternative MBR confounding patterns and informally chose one that by graphical inspection (e.g. Figure 3) seemed to give adequate spatial coverage. In the follow-up paper[16] a formal optimization of the MBR design is presented.

2.5. The system to be studied: optimizing a growth curve

The MBR design will be illustrated by a computer-based simulation of the growth curve of a microorganism[18] under different conditions. A few parameters control the process, and a certain response is to be minimized—in this case the deviation of each growth curve from an ideal curve shape. Thus the example may be regarded as a designed computer experiment to study the model phenome of a highly nonlinear mathematical model, as well as an illustration of how the MBR design may be used for planning a physical study of a biological system with a highly nonlinear response.

Sigmoid growth curves of various shapes are here simulated by the logistic curve function[19] with *K* = 5 parameters:

$$y = f(t; \mathbf{x}) = f(t; x_1, x_2, x_3, x_4, x_5) = x_5 + \frac{x_4}{\left(1 + \left(\frac{t}{x_1}\right)^{x_2}\right)^{x_3}} \quad (3)$$

where *t*∈[0;100] represents time, parameter x_1 stands for time delay, x_2 for sigmoid steepness, x_3 for sigmoid asymmetry, x_4 for maximum growth, and x_5 for baseline offset. In a real biological experimental setting, parameters x_1, x_2,. . ., x_5 might instead represent five generic parameters affecting the growth curves, such as the composition of the growth medium, the sample temperature, the cell concentration added at time zero, and cells lost before counting.

First we employ the MBR design technique for initial range finding, to identify a relevant search range for the *K* = 3 first parameters, which are most difficult to assess *a priori*. Then, within the range found relevant, we seek to optimize the curve shape by a local design combining all *K* = 5 parameters.

In this presentation we focus on the MBR design method and intentionally down-play response measuring, data modeling methodology and realism in the chosen application. The follow-up paper[17] demonstrates its use for studying an actual, high-dimensional application.

The present design and simulation software was programmed in MATLAB® by the authors, and is available at www.specmod.org.

3. RESULTS

3.1. Initial range finding experiment

As in any design of experiment, there was first a need to find the interesting factor range to be investigated—in this case the relevant ranges for the five model parameters in the logistic curve model.

3.1.1. Response

Curves are here only considered interesting if they do not grow too fast or too slowly, as illustrated in Figure 2). Simulating a low-cost assessment, this response (1 = OK, 0 = not OK) was

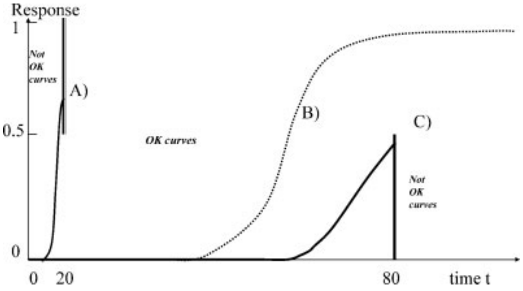


Figure 2. Range finding experiment: system response, y. The known ideal curve (B, dotted) and curves from two conditions (A, C; solid) deemed unacceptable by the simple range finding criteria: $y < 0.5$ at $t = 20$ and $y > 0.5$ at $t = 80$.

determined by the growth response y at only two points in time ($t = 20$ and $t = 80$).

3.1.2. *Design*

The simple offset and scale parameters x_4 and x_5 were considered easy to control and therefore kept constant at 1 and 0, respectively. A 2^{3*4-6} MBR design was defined, with 3 design factors, each at 16 levels (4 bits), requiring $N = 2^6 = 64$ runs. Recoded back to index design with levels 0, 1, ..., 15, the distributions of the chosen design factors [$\mathbf{d}_1$, $\mathbf{d}_2$, $\mathbf{d}_3$] are plotted pair-wise for the recoded integer factors in Figure 3. The confounding pattern was selected so that the factor space was sufficiently spanned.

Table I. Initial range finding design

Factor name	Range tested initially	Range found to be OK	Range found to give curves somewhat similar to the ideal
x_1	[0.01, 100]	[10,80]	[10,75]
x_2	[−100, 0.01]	[−100, −0.01]	[−20, −5]
x_3	[0.01, 100]	[0.01, 100]	[0.01, 2]
x_4	1	1	1
x_5	0	0	0

3.1.3. *Range*

The MBR design levels **D** were then mapped into the corresponding design matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]$ representing the three first parameters in Equation 3. Not knowing the effects of the parameters and their interactions on the response, the ranges were set rather wide (Table I, column 1).

3.1.4. *Evaluation*

The designed simulations were run to generate the 64 curves, and evaluated by the simple OK/not OK response criterion. Figure 4 shows the range finding results. The main effect of parameter x_2 is seen to have the most abrupt effect on the response criterion. But at high values of x_2, an interaction with

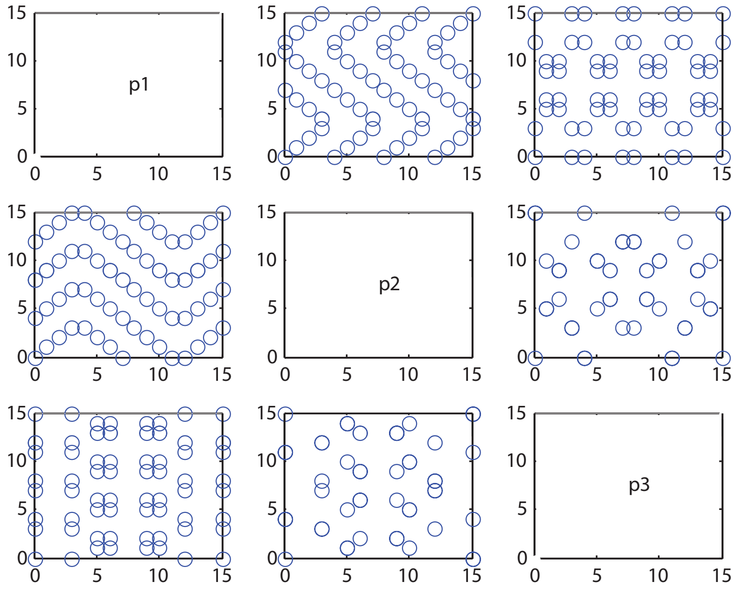


Figure 3. Range finding experiment: choosing an MBR design. Range finding design for $K = 3$ most important design factors $\mathbf{d}_1$, $\mathbf{d}_2$, and $\mathbf{d}_3$ representing the three first parameters $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ in the logistic model (Equation 3), each at $L(k) = 16$ levels (4 bits) in a design with $N = 64$ runs (i.e. a $2^{3 \times 4-6}$ MBR design).

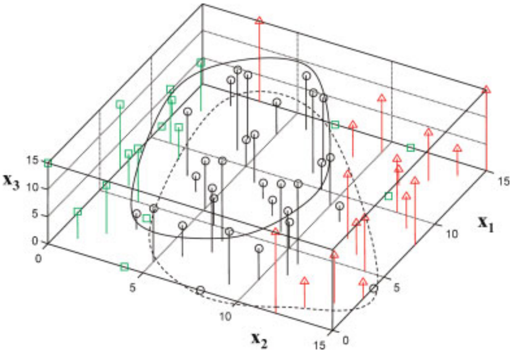


Figure 4. Range finding experiment: finding the acceptable design region. Design factors $\mathbf{d}_1$, $\mathbf{d}_2$, and $\mathbf{d}_3$ displayed in 3D; circles = OK curves (y = 1); triangles and squares = not OK curves (y = 0). Acceptable region of interest is roughly outlined at high and low levels of $\mathbf{d}_3$.

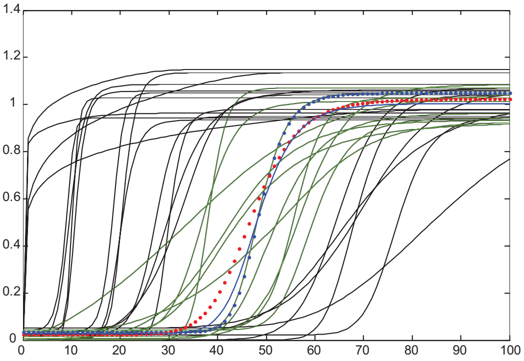


Figure 5. Optimization experiment: system optimization. The resulting curves from the optimization experiment. Red dotted line: the $\mathbf{y}_{\text{Ideal}}$, with "unknown" parameters. Solid curves: The 32 design samples' curves. With blue dots: $\mathbf{y}_{\text{DesignBest}}$. Green curves: the set of 2 × K surrounding neighbors circumscribing $\mathbf{y}_{\text{DesignBest}}$ in parameter space. Dashed blue curve: weighted average estimate of optimum $\mathbf{y}_{\text{WAvg}}$.

parameters x_1 and x_3 is also evident. Based on these results, the acceptable range for parameters x_1–x_3 was identified (Table I, column 2).

To illustrate an iterative way of using range finding designs, the MBR design was run twice with a purpose to find even better ranges of values for x_2 and x_3, but now with another response criterion—to find curves somewhat similar to the ideal one (Figure 2). It was now assumed that the ideal curve would have the property that it starts growing not long time before 1/2 of the maximum time, grow smoothly, and become stable at approximately 3/4 of the maximum time. This yielded the ranges given in Table I, column 3.

3.2. Final optimization experiment

Now that a sensible search range had been identified, a more detailed study was set up in order to identify the optimal values of all five parameters in the logistic curve model.

3.2.1. Response

A certain, predefined ideal curve $\mathbf{y}_{\text{Ideal}}$ (red, dotted curve in Figure 5) was generated from a set of parameter values (Table II, column 4), which are subsequently considered "unknown." Simulating a more expensive, but relevant quality, the criterion to be optimized is then the Euclidian distance of any curve to this ideal curve, measured at 100 time points over the time span $0 \leq t \leq 100$.

3.2.2. Design

If the experiment had been real, and not simulated, the cost of effectuating and profiling every run might be high, so the maximum number of runs was limited to $N = 32$. Still, since the response might display strongly nonlinear dependency on some of the $K = 5$ model parameters and some of their interactions, we wanted a design that spanned the main effects at high resolution ($L(k) = 8$ levels, i.e. 3 bits, for each factor), and at the same time also sampled the two- and three-factor interactions reasonably well at different resolutions. The upper triangle in Figure 6) shows the pair-wise level combinations $\mathbf{D}$ defined from the chosen $2^{5\times3\text{-}10}$ MBR design with only $N = 2^5 = 32$ runs.

3.2.3. Range

The MBR design levels $\mathbf{D}$ were then mapped into values of corresponding design factors $\mathbf{X}$ for all 5 parameters (Table II, column 1), with maximum and minimum values now defined by the acceptable ranges from the initial range finding design (Table I, column 3).

3.2.4. Evaluation

The designed simulations were run to generate the 32 curves $\mathbf{y}_1$–$\mathbf{y}_{32}$, which were evaluated in terms of their distance from the ideal curve $\mathbf{y}_{\text{Ideal}}$. Figure 5 shows this ideal curve $\mathbf{y}_{\text{Ideal}}$ (dotted), together with $\mathbf{y}_{\text{DesignBest}}$, the best-fitting alternative among the 32 runs (dashed, blue). Moreover, the figure shows the 2 × $K = 10$ surrounding neighbor curves (continuous, green), representing the best-fitting alternatives among $\mathbf{y}_1,\ldots,\mathbf{y}_{32}$ that, for each of the 5 parameters, have a design value just above or just below this very best-fitting design alternative $\mathbf{y}_{\text{DesignBest}}$. The lower left triangle in Figure 6 shows the parameters plotted pair-wise, but now with respect to the actual factor values $\mathbf{X}$. The red triangle represents the "unknown" parameters for $\mathbf{y}_{\text{Ideal}}$. The blue square represents the very best design point, yielding the curve $\mathbf{y}_{\text{DesignBest}}$ with smallest deviation from $\mathbf{y}_{\text{Ideal}}$. The green, filled circles represent the 10 neighbors circumscribing $\mathbf{y}_{\text{DesignBest}}$, while the black circles represent the rest of the 32 design points. The circle diameters are proportional to the distance between the curves $\mathbf{y}$ and $\mathbf{y}_{\text{Ideal}}$; small circles represent good fit to $\mathbf{y}_{\text{Ideal}}$.

Since the reduced design has limited resolution, it is useful to interpolate between the best design points, e.g. by local polynomial regression or local weighted averaging. The blue diamond, connected to $\mathbf{y}_{\text{DesignBest}}$, is $\mathbf{y}_{\text{WAvg}}$, the weighted average of the parameters corresponding to $\mathbf{y}_{\text{DesignBest}}$ and the 10 runs surrounding it, with weights defined as inversely proportional to the squared distance of each curve $\mathbf{y}$ from $\mathbf{y}_{\text{DesignBest}}$.

Table II summarizes the optimization results. It shows that the parameters for $\mathbf{y}_{\text{WAvg}}$, the weighted average of the parameters of

Table II. Final parameter levels for MBR response surface design

Factor Name	Range tested	$\mathbf{y}_{\mathrm{DesignBest}}$, the curve most similar to the ideal curve, ($\mathbf{y}_{\mathrm{Ideal}}$)	$\mathbf{y}_{\mathrm{WAvg}}$, the wgt. avg. of $\mathbf{y}_{\mathrm{DesignBest}}$ and its10 surrounding neighbors	$\mathbf{y}_{\mathrm{Ideal}}$, the ideal target curve with "unknown" parameters
x_1	[10,75]	47.1	46.9	47
x_2	[−20, −5]	−15.7	−11.9	−10
x_3	[0.01, 2]	1.43	1.30	1
x_4	[0.9, 1.1]	1.01	0.98	1
x_5	[0,0.05]	0.036	0.029	0.025
Distance from ideal curve		0.091	0.038	0

$1 + K$ nearest neighbors to the ideal curve $\mathbf{y}_{\mathrm{Ideal}}$, are quite close to the true, but "unknown" parameter values for the ideal curve.

4. DISCUSSION

We have presented a design method—the multi-level binary replacement MBR method in its basic form—that appears cost-effective for establishing a model phenome and thus for developing a suitable metamodel.

The simple example is intended to illustrate, generically, how the MBR design method may be applied to a system with relatively complex behavior. The chosen system may represent the growth of a microorganism or the dissolution of a pharmaceutical product under different conditions, as well as a computer model mimicking such a system. First the relevant region in the parameter space of the computer model was found by an initial range finding experiment. Then we illustrated how a reduced, but useful model phenome could be established and analyzed for optimizing the model with respect to a desired output profile. If $\mathbf{y}_{\mathrm{Ideal}}$ had happened to lie further away from the center of the design, the locally weighted interpolation $\mathbf{y}_{\mathrm{WAvg}}$ would have done likewise.

We believe that the MBR method is particularly useful for initial range finding—be it for computer experiments or for real world experiments. The system may be probed at many levels of many input factors with a limited number of runs. If the system behavior is then monitored by a fast or low-cost output characterization, the scientist can afford the risk of wasting many of the runs in regions of the design space that are afterwards

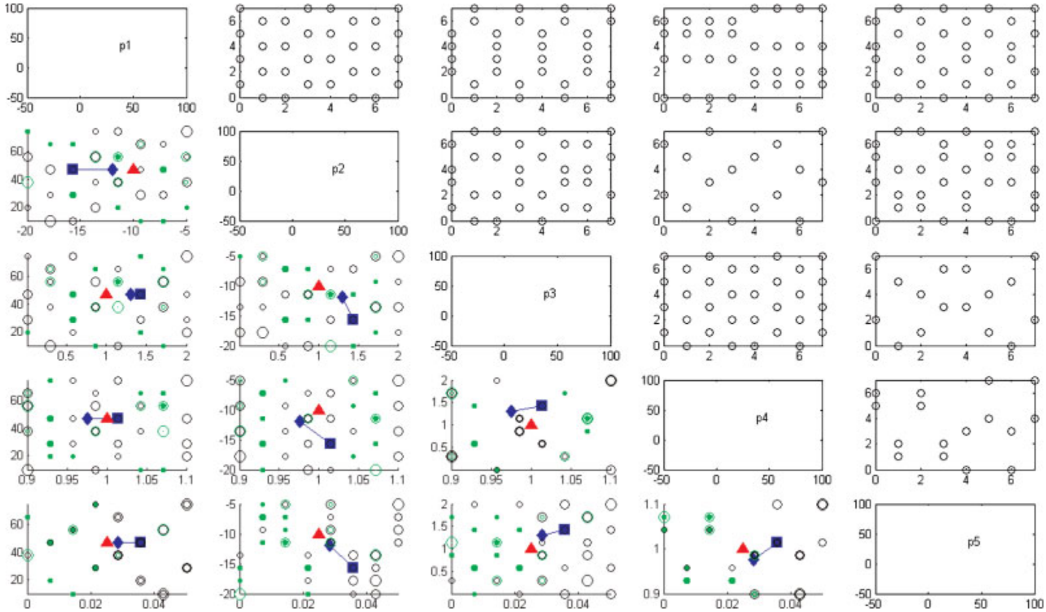


Figure 6. Optimization experiment: An MBR design and its analysis. Design for $K = 5$ factors, each with $L(k) = 8$ levels, studied in $N = 32$ samples, using a 2^{5*3-10} MBR design. Upper right triangle: Levels 0–7 for the five formal design factors $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_4$, and $\mathbf{d}_5]$, replacing the 2^{5*3-10} fractional factorial design $\mathbf{F}$. Lower left triangle: actual factor values $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$, and $\mathbf{x}_5]$ mapped linearly from $\mathbf{D}$. Red triangle = true, but "unknown" parameters of $\mathbf{y}_{\mathrm{Ideal}}$. Square: The parameter combination whose curve $\mathbf{y}_{\mathrm{DesignBest}}$ had the lowest Euclidian distance to $\mathbf{y}_{\mathrm{Ideal}}$. Filled circles: The set of $2^{*}K$ surrounding neighbors circumscribing $\mathbf{y}_{\mathrm{DesignBest}}$ in parameter space. Connected blue diamond = optimized parameter combination corresponding to $\mathbf{y}_{\mathrm{WAvg}}$. "Bubble" diameters correspond to distance from $\mathbf{y}_{\mathrm{Ideal}}$.

found to be unacceptable. Subsequently, a more informative, but demanding phenotyping may then be used, to characterize the output from computer experiments within the relevant parameter region.

To reduce a multi-factor multi-level design carries a risk, irrespective of design methodology: If an unknown, abrupt response change happens to occur only in a highly localized part of the design space, corresponding to a very high-order interaction of certain design factors at certain values, it may not be observed, because that region in the design space is not probed. But the alternative is even riskier—choosing too few factors or too few levels of each factor may cause abrupt effects to be overlooked. However, combining too many factors, with too many levels each, causes combinatorial explosion.

The MBR design is intended to reduce this risk, by detecting abrupt effects as long as they are not limited to very narrow, local design regions. But this remains to be verified. Also, there is a need to compare the MBR design to alternative design methods such as the uniform designs[15,20]. The MBR design method is still at an early stage of development. More work is needed in order to optimize the binary confounding strategy, theoretically or empirically. Since there is a well-defined quantitative relationship between the quantitative design factors **X** and the binary factors in which the fractional factorial design is defined, **G**, it is possible to choose confounding pattern in **G** based on an optimization criterion computed in **X**. In the ensuing modelometrics paper, Tøndel *et al.*[17] apply an optimized version of the MBR design method for computer experiments with a more realistic, complex and high-dimensional computer model from systems biology. To develop a full multivariate metamodel, a multivariate data modeling method from chemometrics is then employed.

Sequential use of the MBR design method calls for special consideration. For instance, when a conventional fractional factorial design with one factor bit per design factor is used, it is well known that follow-up experiments with just a few extra runs can resolve a given confounding structure. The effect of the binary confoundings does seem to be less drastic in the MBR design. Still, it may be useful to do a few follow-up runs even with the MBR method, but how to choose those remains to be elucidated.

5. CONCLUSION

Designed computer experiments are useful for studying the behavior of complicated mathematical or computational models. Based on computer simulation, the model phenome is established as an empirical representation of the behavioral repertoire of a complicated computer model. The information content of the model phenome is limited by the chosen design size and resolution. We have presented a design method—the multi-level binary replacement (MBR) method—that appears cost-effective for establishing a model phenome, since the different binary factor bits probe the parameters in the design space at different spatial resolutions simultaneously. The resulting model phenome, computed once and for all, may be used for developing data-driven multivariate metamodels. These may, in turn, be used for model optimization, computational compaction, and more confident modeling of the system at hand.

The MBR method may of course also be used for designing high-dimensional physical experiments. It was here illustrated for the optimization of a microbiological growth curve, represented by a non-linear function with five design factors, for finding the relevant region of potential interest in the design space, and subsequently for estimating the optimal design point in that space. This simple example is intended to illustrate, generically, how the MBR design method may be applied to a biological system with relatively complex behavior.

Acknowledgements

This study was supported by the National Program for Research in Functional Genomics in Norway (FUGE) (RCN grant no. NFR151924/S10), by the Norwegian eScience program (eVITA) (RCN grant no. NFR178901/V30) and by the Agricultural Food Research Foundation of Norway.

REFERENCES

1. Kleijnen JP. *Design and Analysis of Simulation Experiments*, (1st edn). Springer: 2007.
2. Kohler A, Sulé-Suso J, Sockalingum GD, Tobin M, Bahrami F, Yang Y, Pijanka J, Dumas P, Cotte M, van Pittius DG, Parkes G, Martens H. Estimating and correcting Mie scattering in synchrotron-based microscopic FTIR spectra by extended multiplicative signal correction (EMSC). *Appl. Spectrosc.* 2008; **62**(3): 259–266.
3. Bassan P, Kohler A, Martens H, Lee J, Byrne HJ, Dumas P, Gazi E, Brown M, Clarke N, Gardner P., Resonant Mie Scattering (RMieS) correction of infrared spectra from highly scattering biological samples. *Analyst* 2010; **135**, 268–277.
4. Martens H. 2009; Nonlinear multivariate dynamics modelled by PLSR. In *Proceedings of the 6th International Conference on Partial Least Squares and Related Methods, Beijing, 4–7 September 2009*, Vinzi VE, Tenenhaus M, Guan R, (eds). Publishing House of Electronics Industry: Beijing, China, 139–144. Available at: http://www.phei.com.cn
5. Martens H, Veflingstad SR, Plahte E, Martens M, Bertrand D, Omholt SW. 2009; The genotype-phenotype relationship in multicellular pattern-generating models—the neglected role of pattern descriptors. *BMC Syst. Biol.* **3**: 87. DOI: 10.1186/1752-0509-3-87
6. Fisher RA. The arrangement of field experiments. *J. Minist. Agric. (G.B.)* 1926; **33**: 503–513.
7. Steinberg DM, Hunter WG. Experimental design: review and comment. *Technometrics* 1984; **26**(2): 71–97.
8. Box GEP. The exploration and exploitation of response surfaces: Some general considerations and examples. *Biometrics* 1954; **10**: 16–60.
9. Box G, Behnken D. Some new three level designs for the study of quantitative variables. *Technometrics* 1960; **2**: 455–475.
10. Box GEP, Wilson KB. On the experimental attainment of optimum conditions. *J. Royal Stat. Soc., Ser. B* 1951; **13**: 1–45.
11. Cornell JA. *Experiments with Mixtures: DESIGNS, Models and the Analysis of Mixture Data*. Wiley: New York, 1981.
12. Finney DJ. Fractional replication of factorial arrangements. *Ann. Eugenics* 1945; **12**: 291–301.
13. Montgomery DC. *Design and Analysis of Experiments*, (6th edn). Wiley: New York, 2005.
14. Raymond H, Myers Douglas C. Montgomery: *Response Surface Methodology*. John Wiley and Sons: New York, 1995.
15. Simpson TW, Lin DKJ, Chen W. Sampling strategies for computer experiments: design and analysis. *Int. J. Reliab. Appl.* 2001; **2**(3): 209–240.
16. Addelman S. Orthogonal main-effects plans for asymmetrical factorial experiments. *Technometrics* 1962; **4**: 21–46.
17. Tøndel K, Gjuvsland AB, Måge I, Martens H. Screening design for computer experiments: metamodelling of a deterministic mathematical model of the mammalian circadian clock. *J. Chemom.* 2010; **23**: 1–11.
18. Warringer J, Anevski D, Liu B, Blomberg A. Chemogenetic fingerprinting by analysis of cellular growth dynamics. *MBC Chem. Biol.* **8**(3): 1–10.
19. Gottschalk PG, Dunn JR. The five-parameter logistic: A characterization and comparison with the four-parameter logistic. *Anal. Biochem.* 2005; **343**(1): 54–65.
20. Cela R, Phan Tan Luu R, Claeys-Bruno M. Screening strategies. In: Brown TR, Walczak B, (eds). *Comprehensive Chemometrics*. Elsevier: Oxford, 2009; 251–300.