2

Nonlinear Principal Component Analysis: Neural Network Models and Applications

Matthias Scholz[1], Martin Fraunholz[1], and Joachim Selbig[2]

[1] Competence Centre for Functional Genomics,
Institute for Microbiology, Ernst-Moritz-Arndt-University Greifswald,
F.-L.-Jahn-Str. 15, 17487 Greifswald, Germany,
Matthias.Scholz@uni-greifswald.de
Martin.Fraunholz@uni-greifswald.de

[2] Institute for Biochemistry and Biology, University of Potsdam,
c/o Max Planck Institute for Molecular Plant Physiology
Am Mühlenberg 1, 14424 Potsdam, Germany,
Selbig@mpimp-golm.mpg.de

Summary. *Nonlinear principal component analysis* (NLPCA) as a nonlinear generalisation of standard *principal component analysis* (PCA) means to generalise the principal components from straight lines to curves. This chapter aims to provide an extensive description of the autoassociative neural network approach for NLPCA. Several network architectures will be discussed including the hierarchical, the circular, and the inverse model with special emphasis to missing data. Results are shown from applications in the field of molecular biology. This includes metabolite data analysis of a cold stress experiment in the model plant *Arabidopsis thaliana* and gene expression analysis of the reproductive cycle of the malaria parasite *Plasmodium falciparum* within infected red blood cells.

2.1 Introduction

Many natural phenomena behave in a nonlinear way meaning that the observed data describe a curve or curved subspace in the original data space. Identifying such nonlinear manifolds becomes more and more important in the field of molecular biology. In general, molecular data are of very high dimensionality because of thousands of molecules that are simultaneously measured at a time. Since the data are usually located within a low-dimensional subspace, they can be well described by a single or low number of components. Experimental time course data are usually located within a curved subspace which requires a nonlinear dimensionality reduction as illustrated in Figure 2.1.

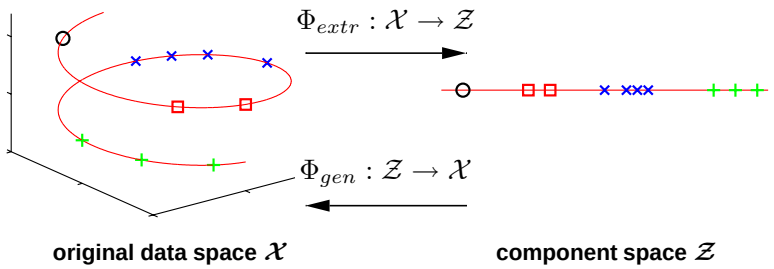

Fig. 2.1. Nonlinear dimensionality reduction. Illustrated are three-dimensional samples that are located on a one-dimensional subspace, and hence can be described without loss of information by a single variable (the component). The transformation is given by the two functions Φ_{extr} and Φ_{gen}. The extraction function Φ_{extr} maps each three-dimensional sample vector (left) onto a one-dimensional component value (right). The inverse mapping is given by the generation function Φ_{gen} which transforms any scalar component value back into the original data space. Such helical trajectory over time is not uncommon in molecular data. The horizontal axes may represent molecule concentrations driven by a circadian rhythm, whereas the vertical axis might represent a molecule with an increase in concentration

Visualising the data is one aspect of molecular data analysis, another important aspect is to model the mapping from original space to component space in order to interpret the impact of observed variables on the subspace (component space). Both the component values (scores) and the mapping function is provided by the neural network approach for nonlinear PCA.
Three important extensions of nonlinear PCA are discussed in this chapter: the hierarchical NLPCA, the circular PCA, and the inverse NLPCA. All of them can be used in combination. *Hierarchical NLPCA* means to enforce the nonlinear components to have the same hierarchical order as the linear components of standard PCA. This hierarchical condition yields a higher meaning of individual components. *Circular PCA* enables nonlinear PCA to extract circular components which describe a closed curve instead of the standard curve with an open interval. This is very useful for analysing data from cyclic or oscillatory phenomena. *Inverse NLPCA* defines nonlinear PCA as an inverse problem, where only the assumed data generation process is modelled, which has the advantage that more complex curves can be identified and NLPCA becomes applicable to incomplete data sets.

Bibliographic notes

Nonlinear PCA based on autoassociative neural networks was investigated in several studies [1, 2, 3, 4]. Kirby and Miranda [5] constrained network units to work in a circular manner resulting in a *circular PCA* whose components

are closed curves. In the fields of atmospheric and oceanic sciences, circular PCA is applied to oscillatory geophysical phenomena, for example, the ocean-atmosphere El Niño-Southern oscillation [6] or the tidal cycle at the German North Sea coast [7]. There are also applications in the field of robotics in order to analyse and control periodic movements [8]. In molecular biology, circular PCA is used for gene expression analysis of the reproductive cycle of the malaria parasite *Plasmodium falciparum* in red blood cells [9]. Scholz and Vigário [10] proposed a *hierarchical nonlinear PCA* which achieves a hierarchical order of nonlinear components similar to standard linear PCA. This hierarchical NLPCA was applied to spectral data of stars and to electromyographic (EMG) recordings of muscle activities. Neural network models for *inverse NLPCA* were first studied in [11, 12]. A more general Bayesian framework based on such inverse network architecture was proposed by Valpola and Honkela [13, 14] for a nonlinear factor analysis (NFA) and a nonlinear independent factor analysis (NIFA). In [15], such inverse NLPCA model was adapted to handle missing data in order to use it for molecular data analysis. It was applied to metabolite data of a cold stress experiment with the model plant *Arabidopsis thaliana*. Hinton and Salakhutdinov [16] have demonstrated the use of the autoassociative network architecture for visualisation and dimensionality reduction by using a special initialisation technique.

Even though the term nonlinear PCA (NLPCA) is commonly referred to as the autoassociative approach, there are many other methods which visualise data and extract components in a nonlinear manner. *Locally linear embedding* (LLE) [17, 18] and *Isomap* [19] were developed to visualise high dimensional data by projecting (embedding) them into a two or low-dimensional space, but the mapping function is not explicitly given. *Principal curves* [20] and *self organising maps* (SOM) [21] are useful for detecting nonlinear curves and two-dimensional nonlinear planes. Practically both methods are limited in the number of extracted components, usually two, due to high computational costs. *Kernel PCA* [22] is useful for visualisation and noise reduction [23].

Several efforts are made to extend independent component analysis (ICA) into a *nonlinear ICA*. However, the nonlinear extension of ICA is not only very challenging, but also intractable or non-unique in the absence of any *a priori* knowledge of the nonlinear mixing process. Therefore, special nonlinear ICA models simplify the problem to particular applications in which some information about the mixing system and the factors (source signals) is available, e.g., by using sequence information [24]. A discussion of nonlinear approaches to ICA can be found in [25, 26]. This chapter focuses on the less difficult task of nonlinear PCA. A perfect nonlinear PCA should, in principle, be able to remove all nonlinearities in the data such that a standard linear ICA can be applied subsequently to achieve, in total, a nonlinear ICA. This chapter is mainly based on [9, 10, 15, 27].

Data generation and component extraction

To extract components, linear as well as nonlinear, we assume that the data are determined by a number of factors and hence can be considered as being generated from them. Since the number of varied factors is often smaller than the number of observed variables, the data are located within a subspace of the given data space. The aim is to represent these factors by components which together describe this subspace. Nonlinear PCA is not limited to linear components, the subspace can be curved, as illustrated in Figure 2.1.
Suppose we have a data space $\mathcal{X}$ given by the observed variables and a component space $\mathcal{Z}$ which is a subspace of $\mathcal{X}$. Nonlinear PCA aims to provide both the subspace $\mathcal{Z}$ and the mapping between $\mathcal{X}$ and $\mathcal{Z}$. The mapping is given by nonlinear functions Φ_{extr} and Φ_{gen}. The *extraction* function $\Phi_{extr} : \mathcal{X} \rightarrow \mathcal{Z}$ transforms the sample coordinates $x = (x_1, x_2, \ldots, x_d)^T$ of the d-dimensional data space $\mathcal{X}$ into the corresponding coordinates $z = (z_1, z_2, \ldots, z_k)^T$ of the component space $\mathcal{Z}$ of usually lower dimensionality k. The *generation* function $\Phi_{gen} : \mathcal{Z} \rightarrow \hat{\mathcal{X}}$ is the inverse mapping which reconstructs the original sample vector x from their lower-dimensional component representation z. Thus, Φ_{gen} approximates the assumed data generation process.

2.2 Standard Nonlinear PCA

Nonlinear PCA (NLPCA) is based on a multi-layer perceptron (MLP) with an autoassociative topology, also known as an autoencoder, replicator network, bottleneck or sandglass type network. An introduction to multi-layer perceptrons can be found in [28].
The autoassociative network performs an identity mapping. The output $\hat{\boldsymbol{x}}$ is enforced to equal the input $\boldsymbol{x}$ with high accuracy. It is achieved by minimising the squared reconstruction error $E = \frac{1}{2} \parallel \hat{\boldsymbol{x}} - \boldsymbol{x} \parallel^2$.
This is a nontrivial task, as there is a 'bottleneck' in the middle: a layer of fewer units than at the input or output layer. Thus, the data have to be projected or compressed into a lower dimensional representation Z.
The network can be considered to consist of two parts: the first part represents the extraction function $\Phi_{extr} : \mathcal{X} \rightarrow \mathcal{Z}$, whereas the second part represents the inverse function, the generation or reconstruction function $\Phi_{gen} : \mathcal{Z} \rightarrow \hat{\mathcal{X}}$. A hidden layer in each part enables the network to perform nonlinear mapping functions. Without these hidden layers, the network would only be able to perform linear PCA even with nonlinear units in the component layer, as shown by Bourlard and Kamp [29]. To regularise the network, a *weight decay* term is added $E_{total} = E + \nu \sum_i w_i^2$ in order to penalise large network weights w. In most experiments, $\nu = 0.001$ was a reasonable choice.
In the following, we describe the applied network topology by the notation l_1-l_2-l_3...-l_S where l_s is the number of units in layer s. For example, 3-4-1-4-3 specifies a network of five layers having three units in the input and output

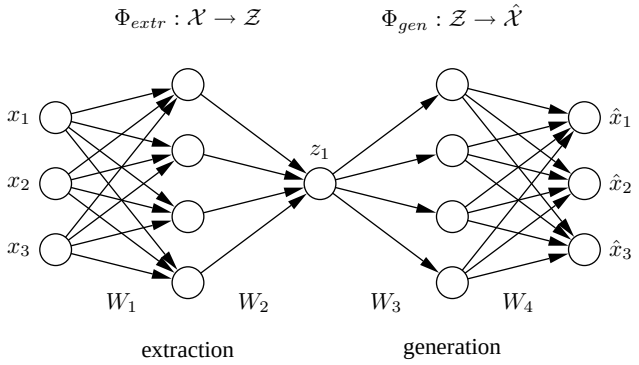

Fig. 2.2. Standard autoassociative neural network. The network output $\hat{\boldsymbol{x}}$ is required to be equal to the input $\boldsymbol{x}$. Illustrated is a 3-4-1-4-3 network architecture. Biases have been omitted for clarity. Three-dimensional samples $\boldsymbol{x}$ are compressed to one component value $\boldsymbol{z}$ in the middle by the extraction part. The inverse generation part reconstructs $\hat{\boldsymbol{x}}$ from $\boldsymbol{z}$. The sample $\hat{\boldsymbol{x}}$ is usually a noise-reduced representation of $\boldsymbol{x}$. The second and fourth layer, with four nonlinear units each, enable the network to perform nonlinear mappings. The network can be extended to extract more than one component by using additional units in the component layer in the middle

layer, four units in both hidden layers, and one unit in the component layer, as illustrated in Figure 2.2.

2.3 Hierarchical nonlinear PCA

In order to decompose data in a PCA related way, linearly or nonlinearly, it is important to distinguish applications of pure *dimensionality reduction* from applications where the identification and discrimination of unique and meaningful components is of primary interest, usually referred to as *feature extraction*. In applications of pure dimensionality reduction with clear emphasis on noise reduction and data compression, only a subspace with high descriptive capacity is required. How the individual components form this subspace is not particularly constrained and hence does not need to be unique. The only requirement is that the subspace explains maximal information in the mean squared error sense. Since the individual components which span this subspace, are treated equally by the algorithm without any particular order or differential weighting, this is referred to as symmetric type of learning. This includes the nonlinear PCA performed by the standard autoassociative neural network which is therefore referred to as s-NLPCA.
By contrast, *hierarchical nonlinear PCA (h-NLPCA)*, as proposed by Scholz and Vigário [10], provides not only the optimal nonlinear subspace spanned by components, it also constrains the nonlinear components to have the same hierarchical order as the linear components in standard PCA.

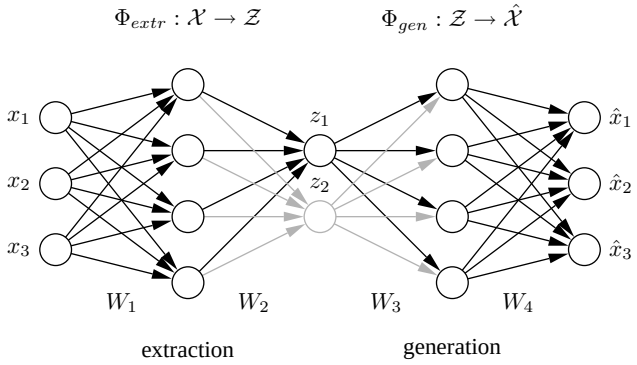

Fig. 2.3. Hierarchical NLPCA. The standard autoassociative network is hierarchically extended to perform the hierarchical NLPCA (h-NLPCA). In addition to the whole 3-4-**2**-4-3 network (grey+black), there is a 3-4-**1**-4-3 subnetwork (black) explicitly considered. The component layer in the middle has either one or two units which represent the first and second components, respectively. Both the error E_1 of the subnetwork with one component and the error of the total network with two components are estimated in each iteration. The network weights are then adapted at once with regard to the total hierarchic error $E = E_1 + E_{1,2}$

Hierarchy, in this context, is explained by two important properties: scalability and stability. Scalability means that the first n components explain the maximal variance that can be covered by a n-dimensional subspace. Stability means that the i-th component of an n component solution is identical to the i-th component of an m component solution.

A hierarchical order essentially yields uncorrelated components. Nonlinearly, this even means that h-NLPCA is able to remove complex nonlinear correlations between components. This can yield useful and meaningful components as will be shown by applications in Section 2.6. Additionally, by scaling the nonlinear uncorrelated components to unit variance, we obtain a complex nonlinear whitening (sphering) transformation [10]. This is a useful pre-processing step for applications such as regression, classification, or blind separation of sources. Since a nonlinear whitening removes the nonlinearities in the data, subsequently applied methods can be linear. This is particularly important for ICA which can be extended to a nonlinear approach by using this nonlinear whitening.

How can we achieve a hierarchical order? The naive approach to simply sort the symmetrically treated components by variance does not yield the required hierarchical order, neither linearly nor nonlinearly. In principle, hierarchy can be achieved by two strongly related ways: either by a constraint to the variance in the component space or by a constraint to the squared reconstruction error in the original space. Similar to linear PCA, the i-th component could be forced to account for the i-th highest variance. But nonlinearly, such constraint can be ineffective or non-unique without additional constraints to the

nonlinear transformation. In contrast, the reconstruction error can be much better controlled, since it is an absolute amount, invariant to any scaling in the transformation. A hierarchical constraint to the error is therefore much more effective. In the simple linear case, we can achieve hierarchically ordered components by a sequential (deflationary) approach in which the components are successively extracted one after the other on the remaining variance given by the squared error of the previous ones. However, this does not work in the nonlinear case, neither successively nor simultaneously by training several networks in parallel. The remaining variance cannot be interpreted by the squared error regardless of the nonlinear transformation [30]. The solution is to use only one network with a hierarchy of subnetworks as illustrated in Figure 2.3. This enables us to formulate the hierarchy directly in the error function [10]. For simplicity, we first restrict our discussion to the case of a two-dimensional component space, but all conclusions can then be generalised to any other dimensionality.

2.3.1 The Hierarchical Error Function

E_1 and $E_{1,2}$ are the squared reconstruction errors when using only the first or both the first and the second component, respectively. In order to perform the h-NLPCA, we have to impose not only a small $E_{1,2}$ (as in s-NLPCA), but also a small E_1. This can be done by minimising the hierarchical error:

$$E_H = E_1 + E_{1,2} \; . \tag{2.1}$$

To find the optimal network weights for a minimal error in the h-NLPCA as well as in the standard symmetric approach, the *conjugate gradient descent* algorithm [31] is used. At each iteration, the single error terms E_1 and $E_{1,2}$ have to be calculated separately. This is performed in the standard s-NLPCA way by a network either with one or with two units in the component layer. Here, one network is the subnetwork of the other, as illustrated in Figure 2.3. The gradient ∇E_H is the sum of the individual gradients $\nabla E_H = \nabla E_1 + \nabla E_{1,2}$. If a weight w_i does not exist in the subnetwork, $\frac{\partial E_1}{\partial w_i}$ is set to zero.
To achieve more robust results, the network weights are set such that the sigmoidal nonlinearities work in the linear range, corresponding to initialise the network with the simple linear PCA solution.
The hierarchical error function (2.1) can be easily extended to k components ($k \leq d$):

$$E_H = E_1 + E_{1,2} + E_{1,2,3} + \cdots + E_{1,2,3,\ldots,k} \; . \tag{2.2}$$

The hierarchical condition as given by E_H can then be interpreted as follows: we search for a k-dimensional subspace of minimal mean square error (MSE) under the constraint that the $(k-1)$-dimensional subspace is also of minimal MSE. This is successively extended such that all $1, \ldots, k$ dimensional subspaces are of minimal MSE. Hence, each subspace represents the data with regard to its dimensionalities best. Hierarchical nonlinear PCA can therefore be seen as a true and natural nonlinear extension of standard linear PCA.

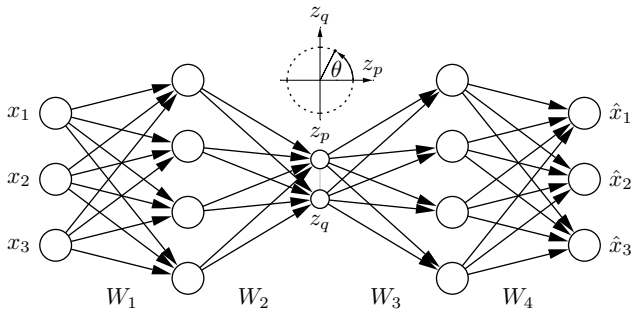

Fig. 2.4. Circular PCA network. To obtain circular components, the auto-associative neural network contains a circular unit pair (p, q) in the component layer. The output values z_p and z_q are constrained to lie on a unit circle and hence can be represented by a single angular variable θ

2.4 Circular PCA

Kirby and Miranda [5] introduced a circular unit at the component layer in order to describe a potential circular data structure by a closed curve. As illustrated in Figure 2.4, a circular unit is a pair of networks units p and q whose output values z_p and z_q are constrained to lie on a unit circle

$$z_p^2 + z_q^2 = 1 \; . \tag{2.3}$$

Thus, the values of both units can be described by a single angular variable θ.

$$z_p = \cos(\theta) \qquad \text{and} \qquad z_q = \sin(\theta) \; . \tag{2.4}$$

The *forward propagation* through the network is as follows: First, equivalent to standard units, both units are weighted sums of their inputs z_m given by the values of all units m in the previous layer.

$$a_p = \sum_m w_{pm} z_m \qquad \text{and} \qquad a_q = \sum_m w_{qm} z_m \; . \tag{2.5}$$

The weights w_{pm} and w_{qm} are of matrix W_2. Biases are not explicitly considered, however, they can be included by introducing an extra input with activation set to one.
The sums a_p and a_q are then corrected by the radial value

$$r = \sqrt{a_p^2 + a_q^2} \tag{2.6}$$

to obtain circularly constraint unit outputs z_p and z_q

$$z_p = \frac{a_p}{r} \qquad \text{and} \qquad z_q = \frac{a_q}{r} \; . \tag{2.7}$$

For *backward propagation*, we need the derivatives of the error function

$$E = \frac{1}{2} \sum_{n}^{N} \sum_{i}^{d} [\hat{\boldsymbol{x}}_i^n - \boldsymbol{x}_i^n]^2 \tag{2.8}$$

with respect to all network weights w. The dimensionality d of the data is given by the number of observed variables, N is the number of samples.
To simplify matters, we first consider the error e of a single sample $\boldsymbol{x}$, $e = \frac{1}{2} \sum_i^d [\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i]^2$ with $\boldsymbol{x} = (x_1, \ldots, x_d)^T$. The resulting derivatives can then be extended with respect to the total error E given by the sum over all n samples, $E = \sum_n e^n$.
While the derivatives of weights of matrices W_1, W_3, and W_4 are obtained by standard back-propagation, the derivatives of the weights w_{pm} and w_{qm} of matrix W_2, which connect units m of the second layer with the units p and q of the component layer, are obtained as follows: We first need the partial derivatives of e with respect to z_p and z_q:

$$\tilde{\sigma}_p = \frac{\partial e}{\partial z_p} = \sum_j w_{jp} \sigma_j \quad \text{and} \quad \tilde{\sigma}_q = \frac{\partial e}{\partial z_q} = \sum_j w_{jq} \sigma_j \ , \tag{2.9}$$

where σ_j are the partial derivatives $\frac{\partial e}{\partial a_j}$ of units j in the fourth layer.
The required partial derivatives of e in respect to a_p and a_q of the circular unit pair are

$$\sigma_p = \frac{\partial e}{\partial a_p} = (\tilde{\sigma}_p z_q - \tilde{\sigma}_q z_p) \frac{z_q}{r^3} \ \text{and} \ \sigma_q = \frac{\partial e}{\partial a_q} = (\tilde{\sigma}_q z_p - \tilde{\sigma}_p z_q) \frac{z_p}{r^3} \ . \tag{2.10}$$

The final back-propagation formulas for all n samples are

$$\frac{\partial E}{\partial w_{pm}} = \sum_n \sigma_p^n z_m^n \quad \text{and} \quad \frac{\partial E}{\partial w_{qm}} = \sum_n \sigma_q^n z_m^n \ . \tag{2.11}$$

2.5 Inverse Model of Nonlinear PCA

In this section we define nonlinear PCA as an inverse problem. While the classical forward problem consists of predicting the output from a given input, the inverse problem involves estimating the input which matches best a given output. Since the model or data generating process is not known, this is referred to as a *blind inverse problem*.
The simple linear PCA can be considered equally well either as a forward or inverse problem depending on whether the desired components are predicted as outputs or estimated as inputs by the respective algorithm. The autoassociative network models both the forward and the inverse model simultaneously. The forward model is given by the first part, the extraction

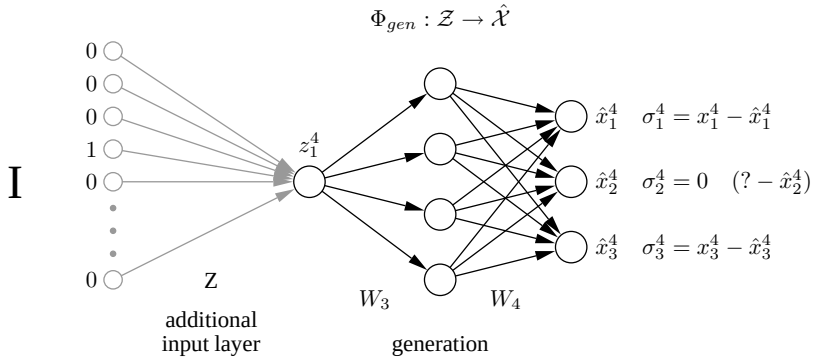

Fig. 2.5. The inverse NLPCA network. Only the second part of the autoassociative network (Figure 2.2) is needed, as illustrated by a 1-4-3 network (black). This generation part represents the inverse mapping Φ_{gen} which generates or reconstructs higher-dimensional samples $\boldsymbol{x}$ from their lower dimensional component representations $\boldsymbol{z}$. These component values $\boldsymbol{z}$ are now unknown inputs that can be estimated by propagating the partial errors σ back to the input layer $\boldsymbol{z}$. This is equivalent to the illustrated prefixed input layer (grey), where the weights are representing the component values $\boldsymbol{z}$. The input is then a (sample x sample) identity matrix I. For the 4th sample (n=4), as illustrated, all inputs are zero except the 4th, which is one. On the right, the second element x_2^4 of the 4th sample $\boldsymbol{x}^4$ is missing. Therefore, the partial error σ_2^4 is set to zero, identical to ignoring or non-back-propagating. The parameter of the model can thus be estimated even when there is missing data

function $\Phi_{extr} : \mathcal{X} \to \mathcal{Z}$. The inverse model is given by the second part, the generation function $\Phi_{gen} : \mathcal{Z} \to \hat{\mathcal{X}}$. Even though a forward model is appropriate for linear PCA, it is less suitable for nonlinear PCA, as it sometimes can be functionally very complex or even intractable due to a one-to-many mapping problem. Two identical samples $\boldsymbol{x}$ may correspond to distinct component values $\boldsymbol{z}$, for example, the point of self-intersection in Figure 2.6B.

By contrast, modelling the inverse mapping $\Phi_{gen} : \mathcal{Z} \to \hat{\mathcal{X}}$ alone, provides a number of advantages: we directly model the assumed data generation process which is often much easier than modelling the extraction mapping. We also can extend the inverse NLPCA model to be applicable to incomplete data sets, since the data are only used to determine the error of the model output. And, it is more efficient than the entire autoassociative network, since we only have to estimate half of the network weights.

Since the desired components now are unknown inputs, the *blind inverse problem* is to estimate both the inputs and the parameters of the model by only given outputs. In the inverse NLPCA approach, we use one single error function for simultaneously optimising both the model weights $\boldsymbol{w}$ and the components as inputs $\boldsymbol{z}$.

2.5.1 The Inverse Network Model

Inverse NLPCA is given by the mapping function Φ_{gen}, which is represented by a multi-layer perceptron (MLP) as illustrated in Figure 2.5. The output $\hat{\boldsymbol{x}}$ depends on the input $\boldsymbol{z}$ and the network weights $w \in W_3, W_4$.

$$\hat{\boldsymbol{x}} = \Phi_{gen}(\boldsymbol{w}, \boldsymbol{z}) = W_4 g(W_3 \boldsymbol{z}) \; . \tag{2.12}$$

The nonlinear activation function g (e.g., $tanh$) is applied element-wise. Biases are not explicitly considered. They can be included by introducing extra units with activation set to one.
The aim is to find a function Φ_{gen} which generates data $\hat{\boldsymbol{x}}$ that approximate the observed data $\boldsymbol{x}$ by a minimal squared error $\parallel \hat{\boldsymbol{x}} - \boldsymbol{x} \parallel^2$. Hence, we search for a minimal error depending on $\boldsymbol{w}$ and $\boldsymbol{z}$: $\min\limits_{w,z} \parallel \Phi_{gen}(\boldsymbol{w}, \boldsymbol{z}) - \boldsymbol{x} \parallel^2$. Both the lower dimensional component representation $\boldsymbol{z}$ and the model parameters $\boldsymbol{w}$ are unknown and can be estimated by minimising the reconstruction error:

$$E(\boldsymbol{w}, \boldsymbol{z}) = \frac{1}{2} \sum_{n}^{N} \sum_{i}^{d} \left[\sum_{j}^{h} w_{ij} g \left(\sum_{i}^{m} w_{jk} z_k^n \right) - x_i^n \right]^2 \; , \tag{2.13}$$

where N is the number of samples and d the dimensionality.
The error can be minimised by using a gradient optimisation algorithm, e.g., *conjugate gradient descent* [31]. The gradients are obtained by propagating the partial errors σ_i^n back to the input layer, meaning one layer more than usual. The gradients of the weights $w_{ij} \in W_4$ and $w_{jk} \in W_3$ are given by the partial derivatives:

$$\frac{\partial E}{\partial w_{ij}} = \sum_n \sigma_i^n g(a_j^n) \quad ; \qquad \sigma_i^n = \hat{x}_i^n - x_i^n \; , \tag{2.14}$$

$$\frac{\partial E}{\partial w_{jk}} = \sum_n \sigma_j^n z_k^n \qquad ; \qquad \sigma_j^n = g'(a_j^n) \sum_i w_{ij} \sigma_i^n \; . \tag{2.15}$$

The partial derivatives of linear input units ($z_k = a_k$) are:

$$\frac{\partial E}{\partial z_k^n} = \sigma_k^n = \sum_j w_{jk} \sigma_j^n \; . \tag{2.16}$$

For circular input units given by equations (2.6) and (2.7), the partial derivatives of a_p and a_q are:

$$\frac{\partial E}{\partial a_p^n} = (\tilde{\sigma}_p^n z_q^n - \tilde{\sigma}_q^n z_p^n) \frac{z_q^n}{r_n^3} \qquad \text{and} \qquad \frac{\partial E}{\partial a_q^n} = (\tilde{\sigma}_q^n z_p^n - \tilde{\sigma}_p^n z_q^n) \frac{z_p^n}{r_n^3} \tag{2.17}$$

with $\tilde{\sigma}_p^n$ and $\tilde{\sigma}_q^n$ given by

$$\tilde{\sigma}_p^n = \sum_j w_{jp}\sigma_j^n \qquad \text{and} \qquad \tilde{\sigma}_q^n = \sum_j w_{jq}\sigma_j^n \ . \tag{2.18}$$

Biases can be added by using additional weights w_{i0} and w_{j0} and associated constants $z_0 = 1$ and $g(a_0) = 1$.
The weights $\boldsymbol{w}$ and the inputs $\boldsymbol{z}$ can be optimised simultaneously by considering $(\boldsymbol{w}, \boldsymbol{z})$ as one vector to optimise by given gradients. This would be equivalent to an approach where an additional input layer is representing the components $\boldsymbol{z}$ as weights, and new inputs are given by a (sample x sample) identity matrix, as illustrated in Figure 2.5. However, this layer is not needed for implementation. The purpose of the additional input layer is only to explain that the inverse NLPCA model can be converted to a conventionally trained multi-layer perceptron, with known inputs and simultaneously optimised weights, including the weights $\boldsymbol{z}$, representing the desired components. Hence, an alternating approach as used in [11] is not necessary. Besides providing a more efficient optimisation, it also avoids the risk of oscillations during training in an alternating approach.
A disadvantage of such an inverse approach is that we have no mapping function $\mathcal{X} \rightarrow \mathcal{Z}$ to map new data $\boldsymbol{x}$ to the component space. However, we can achieve the mapping by searching for an optimal input $\boldsymbol{z}$ to a given new sample $\boldsymbol{x}$. For that, the network weights $\boldsymbol{w}$ are set constant while the input $\boldsymbol{z}$ is estimated by minimising the squared error $\parallel \hat{\boldsymbol{x}}(\boldsymbol{z}) - \boldsymbol{x} \parallel^2$. This is only a low dimensional optimisation by given gradients efficiently performed by a gradient optimisation algorithm.
The inverse NLPCA is able to extract components of higher nonlinear complexity than the standard NLPCA, even self-intersecting components can be modelled, as shown in Figure 2.6B. Inverse NLPCA can be used to extract more than one component by increasing the number of units in the input layer.

2.5.2 NLPCA Models Applied to Circular Data

In Figure 2.6, a circular data structure is used to illustrate the behaviour of NLPCA models: the standard autoassociative network (NLPCA), the inverse model with standard units (NLPCA.inv), and the circular PCA (NLPCA.cir). The data are located on a unit circle, disturbed by Gaussian noise with standard deviation 0.1 . The standard autoassociative network is not able to describe the circular structure all-around due to the problem to map at least one point on the circle to two distinct component values. This problem does not occur in inverse NLPCA since it is only a mapping from component values to the data. The circular structure is approximated by a component that intersects with itself but it has still an open interval. Thus, the closed curve solution as provided by circular PCA gives a more useful description of the circular structure of the data. Circular PCA can also be used as inverse model to be more efficient and to handle missing data.

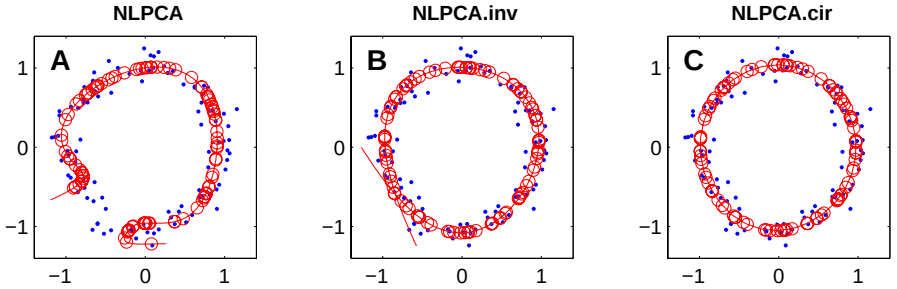

Fig. 2.6. Nonlinear PCA (NLPCA). Shown are results of different variants of NLPCA applied to a two-dimensional artificial data set of a noisy circle. The original data $\boldsymbol{x}$ ('·') are projected onto a nonlinear component (line). The projection or noised-reduced reconstruction $\hat{\boldsymbol{x}}$ is marked by a circle '○'. **(A)** The standard NLPCA cannot describe a circular structure completely. There is always a gap. **(B)** The inverse NLPCA can provide self-intersecting components and hence approximates the circular data structure already quite well. **(C)** The circular PCA is most suitable for a circular data structure, since it is able to approximate the data structure continuously by a closed curve

2.5.3 Inverse NLPCA for Missing Data

There are many methods for estimating missing values [32]. Some good approaches are based on maximum likelihood in conjunction with an expectation-maximisation (EM) algorithm [33]. To analyse incomplete data, it is common to estimate the missing values first in a separate step. But this can lead to problems caused by distinct assumptions in the missing data estimation step and the subsequent analysis. For example, a *linear* missing data estimation can run counter to a subsequent *nonlinear* analysis. Therefore, our strategy is to adapt the analysis technique to be applicable to incomplete data sets, instead of estimating missing values separately. Even though the aim is to extract nonlinear components directly from incomplete data, once the nonlinear mapping is modelled, the missing values can be estimated as well.
As shown in Figure 2.5, the inverse NLPCA model can be extended to be applicable to incomplete data sets [15]: If the ith element x_i^n of the nth sample vector $\boldsymbol{x}^n$ is missing, the corresponding partial error σ_i^n is omitted by setting to zero before back-propagating, hence it does not contribute to the gradients. The nonlinear components are extracted by using all available observations. By using these components, the original data can be reconstructed including the missing values. The network output $\hat{x}_i^n$ gives the estimation of the missing value x_i^n.
The same approach can be used to weight each value differently. This might be of interest when for each value an additional probability value p is available. Each partial error σ_i^n can then be weighted $\tilde{\sigma}_i^n = p * \sigma_i^n$ before back-propagating.

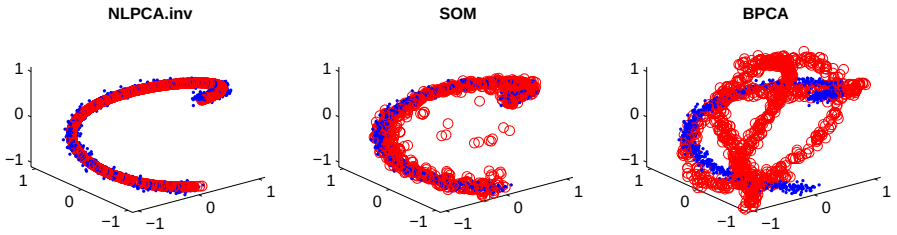

Fig. 2.7. Missing data estimation. Used is an artificial data set which describes a helical loop, plotted as dots ('·'). From each sample, one of the three values is rejected and have to be reconstructed by different missing data estimation algorithms. The reconstructed samples are plotted as circles ('○'). The inverse NLPCA identifies the nonlinear component best and hence gives a very good estimation of the missing values. SOM also gives a reasonably good estimation, but the linear approach BPCA fails on this nonlinear test set, see also Table 2.1

2.5.4 Missing Data Estimation

Even though an artificial data set does not reflect the whole complexity of real biological data, it is useful to illustrate the problem of missing data estimation in order to give a better understanding of how missing data are handled by different methods.

The inverse NLPCA approach is applied to an artificial data set and the results are compared to results of other missing value estimation techniques. This includes the nonlinear estimation by *self organising maps* (SOM) [21] implemented in the SOM TOOLBOX 2.0[3] [34]. Furthermore, we applied a linear PCA-based approach for missing value estimation, an adapted *Bayesian principal component analysis* (BPCA)[4] [35] based on [36].

The data $\boldsymbol{x}$ lie on a one-dimensional manifold (a helical loop) embedded in three dimensions, plus Gaussian noise η of standard deviation $\sigma = 0.05$, see Figure 2.7. 1,000 samples $\boldsymbol{x}$ were generated from a uniformly distributed factor t over the range [-1,1], t represents the angle:

$$\begin{aligned} x_1 &= \sin(\pi t) &+ \eta\,, \\ x_2 &= \cos(\pi t) &+ \eta\,, \\ x_3 &= t &+ \eta\,. \end{aligned}$$

From each three-dimensional sample, one value is randomly removed and regarded as missing. This generates a high missing value rate of 33.3 percent. However, if the nonlinear component (the helix) is known, the estimation of a missing value is exactly given by the two other coordinates, except at the first and last position of the helix loop, where in the case of missing vertical coordinate x_3, the sample can be assigned either to the first or to the last

[3] `http://www.cis.hut.fi/projects/somtoolbox/`
[4] `http://hawaii.aist-nara.ac.jp/~shige-o/tools/`

Table 2.1. Mean square error (MSE) of different missing data estimation techniques applied to the helical data (Figure 2.7). The inverse NLPCA model provides a very good estimation of the missing values. Although the model was trained with noisy data, the noise-free data were better represented than the noisy data, confirming the noise-reducing ability of the model. SOM also gives a good estimation on this nonlinear data, but the linear technique BPCA is only as good as the naive substitution by the mean over the residuals of each variable.

MSE of missing value estimation		
	noise	noise-free
NLPCA.inv	0.0021	0.0013
SOM	0.0405	0.0384
BPCA	0.4191	0.4186
mean	0.4429	0.4422

position. There are two valid solutions. Thus, missing value estimation is not always unique in the nonlinear case.
In Figure 2.7 and Table 2.1 it is shown that even if the data sets are incomplete for all samples, the inverse NLPCA model is able to detect the nonlinear component and provides a very accurate missing value estimation. The nonlinear technique SOM also achieves a reasonably good estimation, but the linear approach BPCA is unsuitable for this nonlinear application.

2.6 Applications

The purpose of nonlinear PCA is to identify and to extract nonlinear components from a given data set. The extracted components span a component space which is supposed to cover the most important information of the data. We would like to demonstrate this in examples of NLPCA applications. First, we discuss results of hierarchical NLPCA in order to illustrate the potential curvature of a components subspace. Then we describe two applications of NLPCA to experimental time courses from molecular biology. This includes both a non-periodic and a periodic time course. The periodic one demonstrates the use of circular PCA. In order to handle missing values, a frequent problem in molecular data, NLPCA is applied in the inverse mode. In both experiments nonlinear PCA is able to identify the time factor already with the first nonlinear component thereby confirming that time is the most important factor in the data. Since NLPCA models explicitly the nonlinear mapping between component space and original data space, it provides a model of the

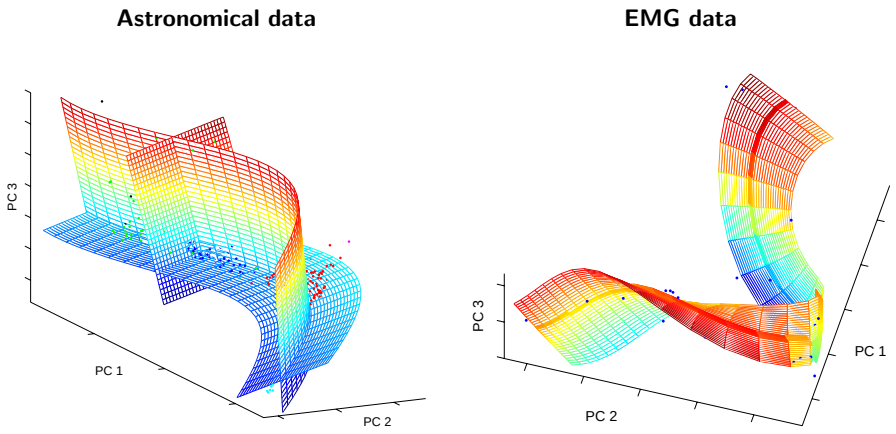

Fig. 2.8. Hierarchical nonlinear PCA is applied to a star spectral data set and to electromyographic (EMG) recordings. Both data sets show a clear nonlinear behaviour. The first three nonlinear components are visualised in the space of the first three PCA components. The grids represent the new coordinate system of the component space. Each grid is spanned by two of the three components while the third is set to zero

biological process which is used here to interpret the impact of individual molecules.

2.6.1 Application of Hierarchical NLPCA

First, we illustrate the performance of hierarchical NLPCA on two separate data sets [10]. The first consists of 19-dimensional spectral information of 487 stars [37]. The second data set is based on electromyographic (EMG) recordings for different muscle activities (labelled as 0, 10, 30, 50 and 70% of maximal personal strength). The one-dimensional EMG signal is then embedded into a d-dimensional space and analysed as a recurrence plot [38]. The final data set then consists of 10 recurrence qualification analysis (RQA) variables for 35 samples, given by the 5 force levels of 7 subjects [39].
The nonlinear components are extracted by minimising the hierarchical error function $E_H = E_1 + E_{1,2} + E_{1,2,3}$. The autoassociative mappings are based on a 19-30-10-30-19 network for the star spectral data and a 10-7-3-7-10 network for the EMG data.
Figure 2.8 shows that both data sets have clear nonlinear characteristics. While in the star data set the nonlinearities seem moderate, this is clearly not the case for the EMG data. Furthermore, in the EMG data, most of the variance is explained by the first two components. The principal curvature given by the first nonlinear component is found to be strongly related to the force level [10]. Since the second component is not related to the force, the force

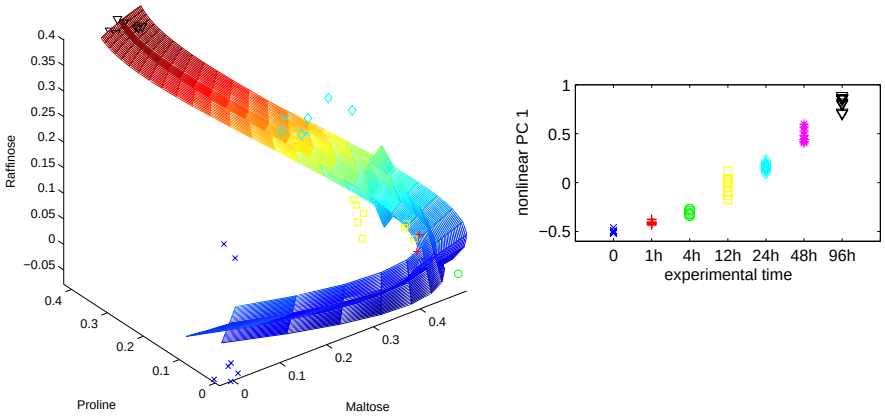

Fig. 2.9. Cold stress metabolite data. Left: The first three extracted nonlinear components are plotted into the data space given by the top three metabolites of highest variance. The grid represents the new coordinate system of the component space. The principal curvature given by the first nonlinear component represents the trajectory over time in the cold stress experiment as shown on the right by plotting the first component against the experimental time

information is supposed to be completely explained by the first component. The second component might be related to another physiological factor.

2.6.2 Metabolite Data Analysis

Cold stress can cause rapid changes of metabolite levels within cells. Analysed is the metabolite response to 4°C cold stress of the model plant *Arabidopsis thaliana* [15, 40]. Metabolite concentrations are measured by *gas chromatography / mass spectrometry (GC/MS)* at 7 time points in the range up to 96 hours. With 7-8 replicas at each time, we have a total number of 52 samples. Each sample provides the concentration levels of 388 metabolites. Precisely, we consider the relative concentrations given by the $\log_2$-ratios of absolute concentrations to a non-stress reference. In order to handle missing data, we applied NLPCA in the inverse mode. A neural network of a 3-20-388 architecture was used to extract three nonlinear components in a hierarchical order, shown in Figure 2.9. The extracted first nonlinear component is directly related to the experimental time factor. It shows a strong curvature in the original data space, shown in Figure 2.10. The second and third component are not related to time and the variance of both is much smaller and of similar amount. This suggests that the second and third component represent only the noise of the data. It also confirms our expectations that time is the major

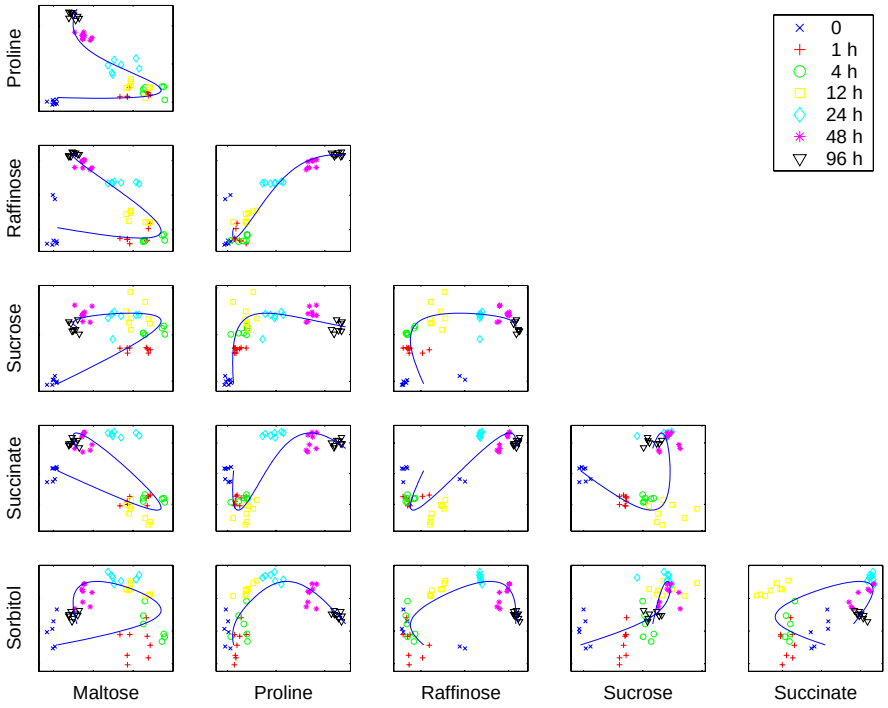

Fig. 2.10. Time trajectory. Scatter plots of pair-wise metabolite combinations of the top six metabolites of highest relative variance. The extracted time component (nonlinear PC 1), marked by a line, shows a strong nonlinear behaviour

factor in the data. NLPCA approximates the time trajectory of the data by use of the first nonlinear component which can therefore be regarded as time component.

In this analysis, NLPCA provides a model of the cold stress adaptation of *Arabidopsis thaliana*. The inverse NLPCA model gives us a mapping function $\mathcal{R}^1 \rightarrow \mathcal{R}^{388}$ from a time point t to the response $\boldsymbol{x}$ of all considered 388 metabolites $\boldsymbol{x} = (x_1, ..., x_{388})^T$. Thus, we can analyse the approximated response curves of each metabolite, shown in Figure 2.11. The cold stress is reflected in almost all metabolites, however, the response behaviour is quite different. Some metabolites have a very early positive or negative response, e.g., maltose and raffinose, whereas other metabolites only show a moderate increase.

In standard PCA, we can present the variables that are most important to a specific component by a rank order given by the absolute values of the corresponding eigenvector, sometimes termed *loadings* or *weights*. As the components are curves in nonlinear PCA, no global ranking is possible. The rank

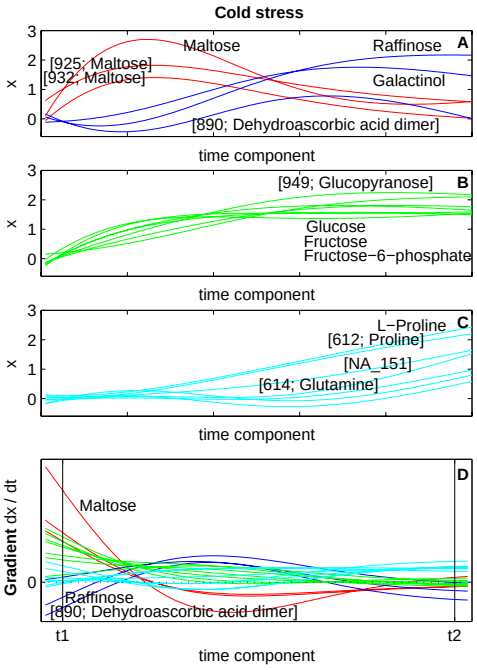

Fig. 2.11. The top three graphs show the different shapes of the approximated metabolite response curves over time. (**A**) Early positive or negative transients, (**B**) increasing metabolite concentrations up to a saturation level, or (**C**) a delayed increase, and still increasing at the last time point. (**D**) The gradient curves represent the influence of the metabolites over time. A high positive or high negative gradient means a strong relative change at the metabolite level. The results show a strong early dynamic, which is quickly moderated, except for some metabolites that are still unstable at the end. The top 20 metabolites of highest gradients are plotted. The metabolite rank order at early time t_1 and late time t_2 is listed in Table 2.2

order is different for different positions on the curved component, meaning that the rank order depends on time. The rank order for a specific time t is given by the values of the tangent vector $\boldsymbol{v} = \frac{d\hat{\boldsymbol{x}}}{dt}$ on the curve at this time. To compare different times, we use l_2-normalised tangents $\tilde{v}_i = v_i / \sqrt{\sum_i |v_i|^2}$ such that $\sum_i (\tilde{v}_i)^2 = 1$. Large absolute values $\tilde{v}_i$ correspond to metabolites of high relative changes on their concentration and hence may be of importance at the considered time. A list of the most important metabolites at an early time point t_1 and a late time point t_2 is given in Table 2.2. The dynamics over time are shown in Figure 2.11D.

2.6.3 Gene Expression Analysis

Many phenomena in biology proceed in a cycle. These include circadian rhythms, the cell cycle, and other regulatory or developmental processes such as the reproductive cycle of the malaria parasite *Plasmodium falciparum* in red blood cells (erythrocytes) which is considered here. Circular PCA is used to analyse this intraerythrocytic developmental cycle (IDC) [9]. The infection and persistence of red blood cells recurs with a periodicity of about 48 hours. The parasite transcriptome is observed by microarrays with a sampling rate of one hour. Two observations, at 23 and 29 hours, are rejected. Thus, the total number of expression profiles is 46, available at `http://malaria.ucsf.edu/`

Table 2.2. Candidate list. The most important metabolites are given for an early time t_1 of about 0.5 hours cold stress and a very late time t_2 of about 96 hours. The metabolites are ranked by the relative change on their concentration levels given by the tangent $\tilde{\boldsymbol{v}}(t)$ at time t on the component curve. As expected, maltose, fructose and glucose show a strong early response to cold stress, however, even after 96 hours there are still some metabolites with significant changes in their levels. Brackets '[...]' denote an unknown metabolite, e.g., [932; Maltose] denotes a metabolite with high mass spectral similarity to maltose.

$t_1 \sim$ 0.5 hours		$t_2 \sim$ 96 hours	
$\tilde{\boldsymbol{v}}$	metabolite	$\tilde{\boldsymbol{v}}$	metabolite
0.43	Maltose methoxyamine	0.24	[614; Glutamine]
0.23	[932; Maltose]	-0.20	[890; Dehydroascorbic acid dimer]
0.21	Fructose methoxyamine	0.18	[NA_293]
0.19	[925; Maltose]	0.18	[NA_201]
0.19	Fructose-6-phosphate	0.17	[NA_351]
0.17	Glucose methoxyamine	0.16	[NA_151]
0.17	Glucose-6-phosphate	0.16	L-Arginine
0.16	[674; Glutamine]	0.16	L-Proline
...		...	

[41, 42]. Each gene is represented by one or more oligonucleotides on this profile. In our analysis, we use the relative expression values of 5,800 oligonucleotides given by the $\log_2$-ratios of individual time hybridisations to a reference pool.

While a few hundred dimensions of the metabolite data set could still be handled by suitable regularisations in NLPCA, the very high-dimensional data space of 5,800 variables makes it very difficult or even intractable to identify optimal curved components by a given number of only 46 samples. Therefore, the 5,800 variables are linearly reduced to 12 principal components. To handle missing data, this linear PCA transformation was done by a linear neural network with two layers 12-5800 working in inverse mode similar to the nonlinear network in section 2.5. Circular PCA is then applied to the reduced data set of 12 linear components. A network of a 2-5-12 architecture is used with two units in the input layer constrained as circular unit pair (p, q).

Circular PCA identifies and describes the principal curvature of the cyclic data by a single component, as shown in Figure 2.12. Thus, circular PCA provides a noise-reduced model of the 48 hour time course of the IDC. The nonlinear 2-5-12 network and the linear 12-5800 network together provide a function $\hat{\boldsymbol{x}} = \Phi_{gen}(\theta)$ which maps any time point, represented by a angular value θ, to the 5,800-dimensional vector $\hat{\boldsymbol{x}} = (\hat{x}_1, ..., \hat{x}_{5800})^T$ of corresponding expression values. Again, this model can be used to identify candidate genes at specific times and to interpret the shape of gene expression curves [9].

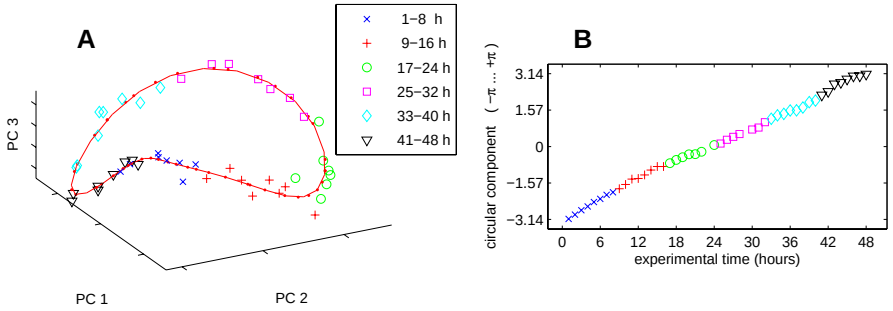

Fig. 2.12. Cyclic gene expression data. (A) Circular PCA describes the circular structure of the data by a closed curve – the circular component. The curve represents a one-dimensional component space as a subspace of a 5,800 dimensional data space. Visualised is the component curve in the reduced three dimensional subspace given by the first three components of standard (linear) PCA.
(B) The circular component (corrected by an angular shift) is plotted against the original experimental time. It shows that the main curvature, given by the circular component, explains the trajectory of the IDC over 48 hours

2.7 Summary

Nonlinear PCA (NLPCA) was described in several variants based on neural networks. This includes the hierarchical, the circular, and the inverse model. While standard NLPCA characterises the desired subspace only as a whole, hierarchical NLPCA enforces to describe this subspace by components arranged in a hierarchical order similar to linear PCA. Hierarchical NLPCA can therefore be seen as a natural nonlinear extension to standard PCA. To describe cyclic or oscillatory phenomena, we need components which describe a closed curve instead of a standard curve with open interval. These circular components can be achieved by circular PCA. In contrast to standard NLPCA which models both the forward component extraction and the inverse data generation, *inverse* NLPCA means to model the inverse mapping alone. Inverse NLPCA is often more efficient and better suited for describing real processes, since it directly models the assumed data generation process. Furthermore, such an inverse model offers the advantage to handle missing data. The idea behind solving the missing data problem was that the criterion of a missing data estimation does not always match the criterion of the subsequent data analysis. Our strategy was therefore to adapt nonlinear PCA to be applicable to incomplete data, instead of estimating the missing values in advance.
Nonlinear PCA was applied to several data sets, in particular to molecular data of experimental time courses. In both applications, the first nonlinear component describes the trajectory over time, thereby confirming our expectations and the quality of the data. Nonlinear PCA provides a noise-reduced model of the investigated biological process. Such computational model can

then be used to interpret the molecular behaviour over time in order to get a better understanding of the biological process.
With the increasing number of time experiments, nonlinear PCA may become more and more important in the field of molecular biology. Furthermore, nonlinearities can also be caused by other continuously observed factors, e.g., a range of temperatures. Even natural phenotypes often take the form of a continuous range [43], where the molecular variation may appear in a nonlinear way.

Availability of Software

A MATLAB® implementation of nonlinear PCA including the hierarchical, the circular, and the inverse model is available at:
`http://www.NLPCA.org/matlab.html`.

Acknowledgement. This work is partially funded by the German Federal Ministry of Education and Research (BMBF) within the programme of *Centres for Innovation Competence* of the BMBF initiative *Entrepreneurial Regions* (Project No. ZIK 011).

References

1. Kramer, M.A.: Nonlinear principal component analysis using auto-associative neural networks. AIChE Journal, **37**(2), 233–243 (1991)
2. DeMers, D., Cottrell, G.W.: Nonlinear dimensionality reduction. In: Hanson, D., Cowan, J., Giles, L., eds.: Advances in Neural Information Processing Systems 5, San Mateo, CA, Morgan Kaufmann, 580–587 (1993)
3. Hecht-Nielsen, R.: Replicator neural networks for universal optimal source coding. Science, **269**, 1860–1863 (1995)
4. Malthouse, E.C.: Limitations of nonlinear PCA as performed with generic neural networks. IEEE Transactions on Neural Networks, **9**(1), 165–173 (1998)
5. Kirby, M.J., Miranda, R.: Circular nodes in neural networks. Neural Computation, **8**(2), 390–402 (1996)
6. Hsieh, W.W., Wu, A., Shabbar, A.: Nonlinear atmospheric teleconnections. Geophysical Research Letters, **33**(7), L07714 (2006)
7. Herman, A.: Nonlinear principal component analysis of the tidal dynamics in a shallow sea. Geophysical Research Letters, **34**, L02608 (2007)
8. MacDorman, K., Chalodhorn, R., Asada, M.: Periodic nonlinear principal component neural networks for humanoid motion segmentation, generalization, and generation. In: Proceedings of the Seventeenth International Conference on Pattern Recognition (ICPR), Cambridge, UK, 537–540 (2004)
9. Scholz, M.: Analysing periodic phenomena by circular PCA. In: Hochreiter, M., Wagner, R. (eds.) Proceedings BIRD conference. LNBI 4414, Springer-Verlag Berlin Heidelberg, 38–47 (2007)
10. Scholz, M., Vigário, R.: Nonlinear PCA: a new hierarchical approach. In: Verleysen, M., ed.: Proceedings ESANN, 439–444 (2002)

11. Hassoun, M.H., Sudjianto, A.: Compression net-free autoencoders. Workshop on Advances in Autoencoder/Autoassociator-Based Computations at the NIPS'97 Conference (1997)
12. Oh, J.H., Seung, H.: Learning generative models with the up-propagation algorithm. In: Jordan, M.I., Kearns, M.J., Solla, S.A., eds.: Advances in Neural Information Processing Systems. Vol. 10., The MIT Press, 605–611 (1998)
13. Lappalainen, H., Honkela, A.: Bayesian nonlinear independent component analysis by multi-layer perceptrons. In: Girolami, M. (ed.) Advances in Independent Component Analysis. Springer-Verlag, 93–121 (2000)
14. Honkela, A., Valpola, H.: Unsupervised variational bayesian learning of nonlinear models. In: Saul, L., Weis, Y., Bottous, L. (eds.) Advances in Neural Information Processing Systems, 17 (NIPS'04), 593–600 (2005)
15. Scholz, M., Kaplan, F., Guy, C., Kopka, J., Selbig, J.: Non-linear PCA: a missing data approach. Bioinformatics, **21**(20), 3887–3895 (2005)
16. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science, **313** (5786), 504–507 (2006)
17. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science, **290** (5500), 2323–2326 (2000)
18. Saul, L.K., Roweis, S.T.: Think globally, fit locally: Unsupervised learning of low dimensional manifolds. Journal of Machine Learning Research, **4** (2), 119–155 (2004)
19. Tenenbaum, J., de Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. Science, **290** (5500), 2319–2323 (2000)
20. Hastie, T., Stuetzle, W.: Principal curves. Journal of the American Statistical Association, **84**, 502–516 (1989)
21. Kohonen, T.: Self-Organizing Maps. 3rd edn. Springer (2001)
22. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, **10**, 1299–1319 (1998)
23. Mika, S., Schölkopf, B., Smola, A., Müller, K.R., Scholz, M., Rätsch, G.: Kernel PCA and de–noising in feature spaces. In: Kearns, M., Solla, S., Cohn, D., eds.: Advances in Neural Information Processing Systems 11, MIT Press, 536–542 (1999)
24. Harmeling, S., Ziehe, A., Kawanabe, M., Müller, K.R.: Kernel-based nonlinear blind source separation. Neural Computation, **15**, 1089–1124 (2003)
25. Jutten, C., Karhunen, J.: Advances in nonlinear blind source separation. In: Proc. Int. Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003), Nara, Japan, 245–256 (2003)
26. Cichocki, A., Amari, S.: Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications. Wiley, New York (2003)
27. Scholz, M.: Approaches to analyse and interpret biological profile data. PhD thesis, University of Potsdam, Germany (2006) URN: urn:nbn:de:kobv:517-opus-7839, URL: http://opus.kobv.de/ubp/volltexte/2006/783/.
28. Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press (1995)
29. Bourlard, H., Kamp, Y.: Auto-association by multilayer perceptrons and singular value decomposition. Biological Cybernetics, **59** (4-5), 291–294, (1988)
30. Scholz, M.: Nonlinear PCA based on neural networks. Master's thesis, Dep. of Computer Science, Humboldt-University Berlin (2002) (in German)

31. Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. Journal of Research of the National Bureau of Standards, **49**(6), 409–436 (1952)
32. Little, R.J.A., Rubin, D.B.: Statistical Analysis with Missing Data. 2nd edn. John Wiley & Sons, New York (2002)
33. Ghahramani, Z., Jordan, M.: Learning from incomplete data. Technical Report AIM-1509 (1994)
34. Vesanto, J.: Neural network tool for data mining: SOM toolbox. In: Proceedings of Symposium on Tool Environments and Development Methods for Intelligent Systems (TOOLMET2000), Oulu, Finland, Oulun yliopistopaino, 184–196 (2000)
35. Oba, S., Sato, M., Takemasa, I., Monden, M., Matsubara, K., Ishii, S.: A bayesian missing value estimation method for gene expression profile data. Bioinformatics, **19**(16), 2088–2096 (2003)
36. Bishop, C.: Variational principal components. In: Proceedings Ninth International Conference on Artificial Neural Networks, ICANN'99, 509–514 (1999)
37. Stock, J., Stock, M.: Quantitative stellar spectral classification. Revista Mexicana de Astronomia y Astrofisica, **34**, 143–156 (1999)
38. Webber Jr., C., Zbilut, J.: Dynamical assessment of physiological systems and states using recorrence plot strategies. Journal of Applied Physiology, **76**, 965–973 (1994)
39. Mewett, D.T., Reynolds, K.J., Nazeran, H.: Principal components of recurrence quantification analysis of EMG. In: Proceedings of the 23rd Annual IEEE/EMBS Conference, Istanbul, Turkey (2001)
40. Kaplan, F., Kopka, J., Haskell, D., Zhao, W., Schiller, K., Gatzke, N., Sung, D., Guy, C.: Exploring the temperature-stress metabolome of *Arabidopsis*. Plant Physiology, **136**(4), 4159–4168 (2004)
41. Bozdech, Z., Llinas, M., Pulliam, B., Wong, E., Zhu, J., DeRisi, J.: The transcriptome of the intraerythrocytic developmental cycle of *Plasmodium falciparum*. PLoS Biology, **1** (1), E5 (2003)
42. Kissinger, J., Brunk, B., Crabtree, J., Fraunholz, M., Gajria, et al., B.: The plasmodium genome database. Nature, **419** (6906), 490–492 (2002)
43. Fridman, E., Carrari, F., Liu, Y.S., Fernie, A., Zamir, D.: Zooming in on a quantitative trait for tomato yield using interspecific introgressions. Science, **305** (5691), 1786–1789 (2004)