# 22

# Microarray Data Classified by Artificial Neural Networks

**Roland Linder, Tereza Richards, and Mathias Wagner**

**Summary**

Systems biology has enjoyed explosive growth in both the number of people participating in this area of research and the number of publications on the topic. The field of systems biology encompasses the *in silico* analysis of high-throughput data as provided by DNA or protein microarrays. Along with the increasing availability of microarray data, attention is focused on methods of analyzing the expression rates. One important type of analysis is the classification task, for example, distinguishing different types of cell functions or tumors. Recently, interest has been awakened toward artificial neural networks (ANN), which have many appealing characteristics such as an exceptional degree of accuracy. Nonlinear relationships or independence from certain assumptions regarding the data distribution are also considered. The current work reviews advantages as well as disadvantages of neural networks in the context of microarray analysis. Comparisons are drawn to alternative methods. Selected solutions are discussed, and finally algorithms for the effective combination of multiple ANNs are presented. The development of approaches to use ANN-processed microarray data applicable to run cell and tissue simulations may be slated for future investigation.

**Key Words:** Artificial neural network; comparison of methods; data analysis; microarray; multicategory classification; bibliometry.

## 1. Introduction

Systems biology is an emerging discipline focused on tackling the enormous technical and intellectual challenges associated with devising generally applicable methods of interpreting data in a way that will shed light on the complex relationships between multiple genes and their products in order to generate comprehensive understanding of how organisms are built and run. The term *systems biology* includes for example computer assisted mathematical modeling and data analysis. The latter encompasses dealing with high-throughput analyses

From: *Methods in Molecular Biology, vol. 382: Microarrays: Second Edition: Volume 2*
Edited by: J. B. Rampal © Humana Press Inc., Totowa, NJ

such as DNA microarrays. A DNA microarray is a two-dimensional array, typically mounted on a glass, filter, or silicon wafer, upon which genes or gene fragments otherwise referred to as complementary DNA or cDNA, are deposited or synthesized in a predetermined spatial order, which quantitatively measures corresponding mRNA sequences. These signals (expression rates) may be analyzed and are characterized by few observations and potentially thousands of predictor variables per microarray ***(1)***. The likely outcome of the analysis can be subdivided in three ways ***(2)***. This may be done by:

α. The detection of differences in the expression rates among diverse groups/populations.
β. Cluster analysis of genes or samples in order to detect groups or structures (*unsupervised learning*) ***(3)***.
γ. The classification of diseased entities (*supervised learning*) ***(4)***.

Concering α: it may be preferable to make the distinction between tumorous as opposed to sound tissue or to make a comparison of the expression rates of patients as against healthy controls. The rationale is the detection of features associated with the pathogenesis or manifestations of complex diseases as the point of departure for causal therapy concepts ***(5)***. This comparison of treated cell cultures against untreated may potentially help in improving therapy. Additionally, it is possible to improve therapy by comparing treated cell cultures against untreated.

Concering β: the cluster analysis can be useful in identifying groups or subgroups of genes that produce similar expression patterns ***(6,7)***. For example new tumor subtypes can be detected ***(8–10)***.

Concering γ: the classification of diseased entities may be one of the most promising possibilities of the microarray analysis. The classification of varied tumor subtypes can improve the diagnostic quality and so inform the selection of the most acceptable therapy. Researchers are continuously exploring gene expression profiles that are designed to give a pretherapeutic indicator concerning the success of certain chemotherapy ***(11)***. Another way in which this may be applied is in the categorization of breast tumor patients into high risk and low risk groups again with the objective of optimizing therapy ***(12)***. With respect to microbiology, classification may help to more accurately classify strains of bacteria ***(13,14)*** or to identify new strains ***(15)***. An advantage for the patient is that his or her infection can be identified more quickly and therefore treated more specifically ***(16,17)***.

The objectives previously outlined can be analyzed by artificial neural networks (ANN). ANNs (also termed "neural nets" or "connectionist models") are a series of nonlinear, interconnected mathematical equations, which tangentially resemble biological neuronal systems and which are used to calculate an

output variable on the basis of independent input variables. Neural network analysis is derived from artificial intelligence, however, it differs from expert systems in that, instead of being rule-based with preprogrammed constraints, rules, or conditions, it consists of neural networks, which "learn" and progressively develop meaningful reliable relationships between input and output variables. Unlike classical statistical models and correlative methods, neural networks comprise multiple indirect interconnections between input and output variables and employ nonlinear mathematical equations and statistical techniques to successively minimize the variance between actual and predicted outputs. This consequently produces a model, which can be subsequently applied to an independent data set, and in turn produces predicted outputs that reliably correspond to the actual observed values.

This work focuses on multilayered feed-forward networks using ANN as a synonym for this type of network. Even if this type of ANN is feasible to classify unsupervised networks (so-called *bottleneck networks)* ***(18)*** its primary effectiveness is the supervised classification as outlined above.

## 2. Methods

The history of ANNs dates back to as early as 1890, when a model was developed by an American psychologist to explain the capabilities of the brain in making associations ***(19)***. In 1958, a two-layered ANN known as "perceptron" was described ***(20)***. In 1986, a powerful learning algorithm was introduced: this algorithm could also handle hidden neurons ***(21)***, representing the starting point for the triumphant progress of ANN technology. Of no less importance to medicine, ANNs are used for supporting decisions in diagnosis, classification, early detection, prognosis, and quality control.

### *2.1. Bibliometric Analysis*

As revealed by a MEDLINE search for the MeSH term "Neural Networks Computer," there is a growing interest in ANNs, with the search having produced more than 7000 articles dealing with this topic, most of which focus on feed-forward networks (**Fig. 1**). Searching MEDLINE for "Neural Networks" (Computer) [MeSH] and "Microarray Analysis" [MeSH] retrieved 60 hits (as of 4th May, 2005), indicating that ANNs have been established for the analysis of microarray data (**Fig. 2**). The use of MeSH headings may be controversially discussed ***(22)***.

In light of the aforementioned results a simplified analysis of abstracts (without searching MeSH terms) was carried out to analyze the representation of the present topic in abstracts of the biomedical body of literature. The current approach was chosen because abstracts appear to be more frequently used to obtain a quick overview on a given topic ***(23)***. The bibliometric overview of

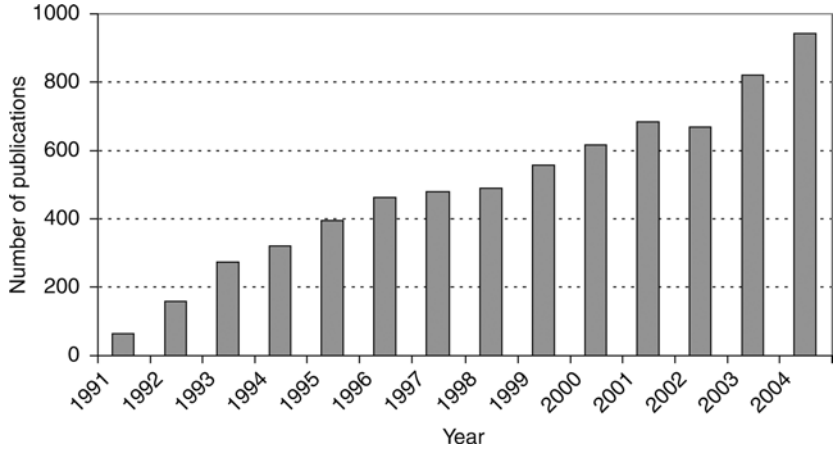


Fig. 1. MEDLINE search for the MeSH term "Neural Networks Computer" (as of April 7th, 2005).

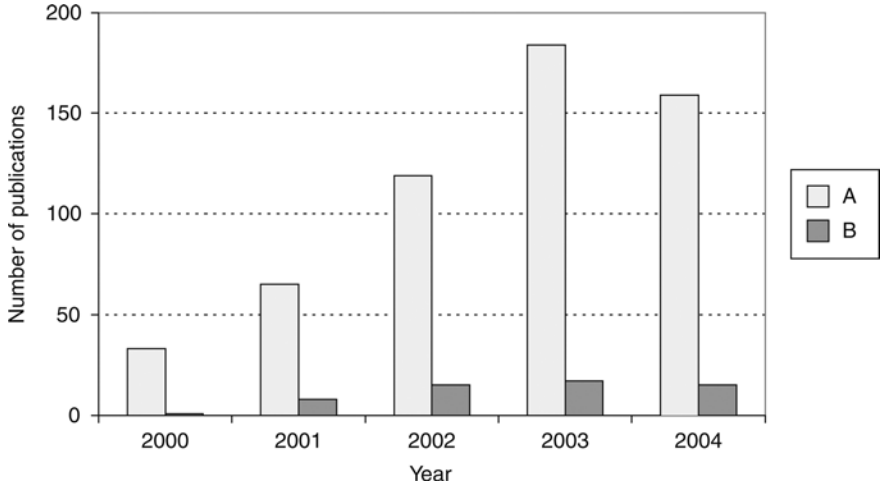


Fig. 2. MEDLINE search (as of May 5th, 2005) on methods involved in microarray analysis. Reviews have been omitted from the results of this search. Conventional statistical methods **(A)** vs ANN **(B)** are displayed; search algorithm for **(A)**: "Microarray Analysis[MeSH] AND "Medical Informatics" [MeSH] NOT "Neural Networks (Computer)" (MeSH); search algorithm for **(B)**: "Neural Networks (Computer)"[MeSH] AND "Microarray Analysis" [MeSH].

ANN and microarray was initiated by interrogating PubMed using the search strategies "artificial neural network" and microarray; "neural network" and microarray; and ANN and microarray (as of 3rd May, 2005). A total of 28 abstracts were found of which 23 (82.14%) were considered relevant on

perusal of the abstracts. Analysis of this data set, albeit a small one, gives a synoptic overview of how the abstracts serve to indicate the status of research activity and therefore give a microcosmic perspective of the state of research in that field as it is represented without including MeSH terms. The data set shows that of the 23 abstracts, one was published in 1999, whereas 22 (95.65%) came out between 2000 and 2005, and all corresponding articles had been written in English.

Lotka ***(24)*** states that the number of authors making $n$ contributions to the literature is about $1/n^2$ of those making one contribution. Using this premise as a point of departure, the data set shows that there are 127 distinct authors who contributed to the articles and of this number, 112 authors (88.18%) contributed to one article; 13 authors (10.23%) contributed to two articles; and 2 authors (1.57%) contributed to three articles.

The articles from which the abstracts have been analyzed were published across 18 journals and the level of productivity of these can best be assessed by considering Bradford ***(25)***, which contends that, if a set of articles is divided into three approximately equal "zones" of productivity such that the ratio $1:n:n^2$ will hold, where, 1 is the number of journals in the first zone and n is a proportional multiplier, then there is always a small "core" of journals, which contains a large number of the articles, usually about one-third of the total, a second larger group which accounts for another third and the last very large group of journals containing the final third. In applying Bradford's data set there are 16 journals (88.88%) having only one article each, one journal (5.55%) with three articles, and one journal (5.55%) with four articles. In this case the very small core consists of two journals *Bioinformatics* (four articles) and *BMC Bioinformatics* (three articles). Although mathematically the precise formulation might not "hold" in this case (which is a contention bibliometricians often cite in critiquing Bradford's Law), in principle it quite accurately represents the basic premise, that in most fields of research it is only a small number of "core" journals that are the most productive with decreasing productivity over the secondary and tertiary level journals.

There are eight countries which contributed to the research output (as represented by the abstracts excluding the MeSH terms) (**Fig. 3**).

Consistent with the literature of the sciences, journals are the primary vehicle of publication, with the small data set indicating the 23 articles were published in 18 different journals and with the small core of two journals being responsible for 7 of the 23 articles which is 30.43%. This approx ⅓ or 33% that Bradford's Law postulates that the first zone of "core" journals will contain. Increasing research interest is demonstrated, with over 95% of the articles having been published just for the period 2000–2005. A notable pattern of collaboration is shown through coauthorship of articles (**Fig. 4**).

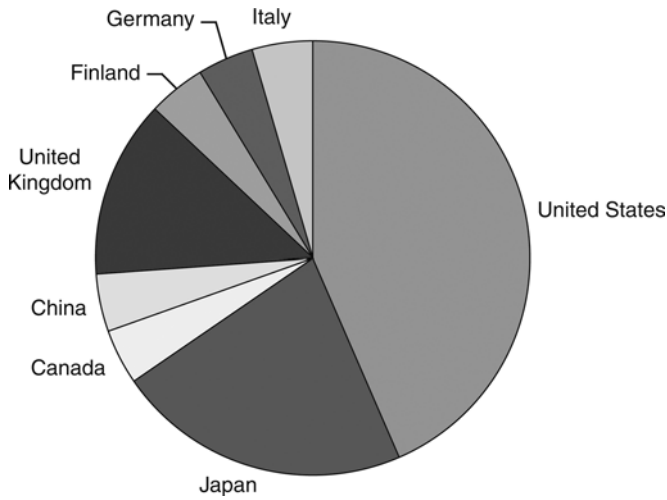


Fig. 3. Geographical location of research groups contributing relevant abstracts to MEDLINE in terms of ANN usage in microarray data analysis (as of May 3rd, 2005).

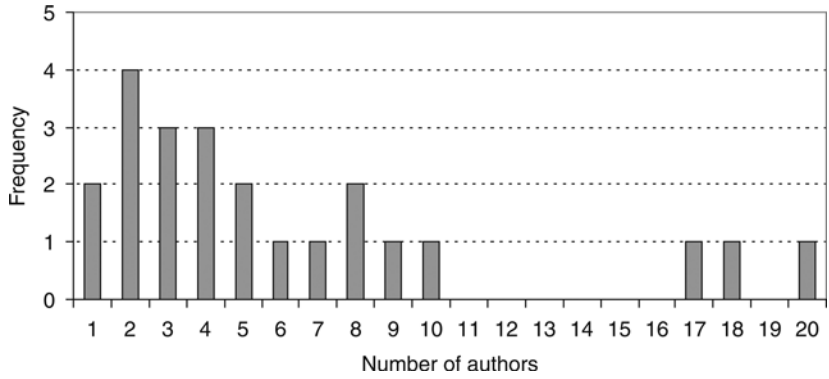


Fig. 4. Quantitative analysis of authors contributing to an ANN based analysis of microarray data (abstracts found in MEDLINE; as of May 3rd, 2005).

The mean number of authors collaborating per article is eight. There is a notable lack of articles written in languages other than English. This may be attributable to the assertion that it is usually more difficult to get cited when published in non-English language journals, resulting in English being the first language of choice, in which to get articles printed, as researchers will undoubtedly aspire to get their research results accepted for publication and so increase the likelihood of their articles being "acknowledged" by the academy and consequently cited in the literature.

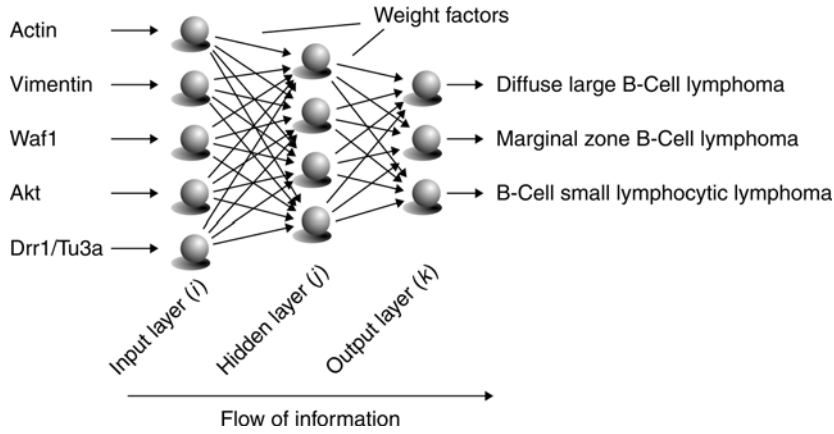


Fig. 5. An example for the structure of a feed-forward artificial neural network: Different types of lymphoma are distinguished by the expression rates of selected genes.

### *2.2. How Does an ANN Work?*

Generally, feed-forward networks consist of three or more layers of artificial neurons each with data entered in the input layer and further processed in the hidden and output layers. Through a learning process, which consists of a "training phase" and a "recall phase" ANNs use nonlinear mathematical equations to successively develop meaningful relationships between input and output variables. The relationships between the different input variables and the output variable(s) are established through adaptations of the weight factors assigned to the interconnections between the layers of the artificial neurons, and this takes place in the training phase. This adaptation is based on rules that are set in the learning algorithm. At the end of the training phase, the weight factors are fixed. Data from patterns not previously interpreted by the network are entered in the recall phase, and an output is calculated based on the previously mentioned and now fixed weight factors (**Fig. 5**). Output neurons usually produce what may be loosely referred to as "activity" between 0 and 1 when this occurs a pattern is assigned to the class with the highest neuronal activity which equates to the "winner takes all" rule.

Preprocessing (*see* **Note 1**) and gene selection (*see* **Note 2**) precede the training phase which will be initiated with random weights at the connections between the neurons. A training data set with known outcome is then entered at the input neurons. The ANN compares its own output values with the known outcome (e.g., 1 for "diffuse large B-cell lymphoma," 0 otherwise), calculates an error value, which will change as the weights at the connections change, with the ANN attempting to minimize the error by adjusting the weights according

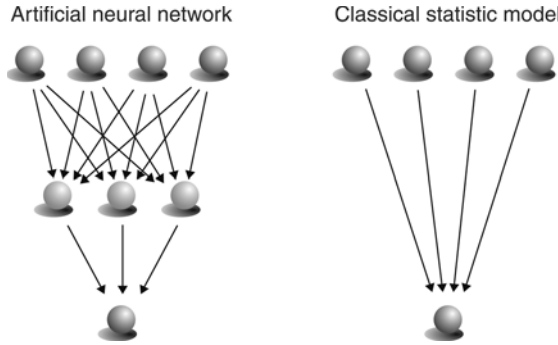


Fig. 6. This schematic comparison as described in **ref.** ***87*** reveals differences in architecture between a three layered artificial neural network and a classical statistic model.

to the learning algorithm. This process is repeated for a predefined number of times (or "epochs") and later, in the recall phase, the ANN can be tested on data with known outcome values (*see* **Note 3**).

### *2.3. What Are the Advantages of ANNs Compared to Alternative Methods?*

As shown earlier (**Fig. 1**), ANNs are increasingly being used as an "intelligent" alternative to the conventional statistical multivariate analysis methodology. The impetus behind the increasing interest in ANNs are primarily the consideration given to possible nonlinear connections (1) between the predictor variables (genes) and the output variables as well as (2) nonlinear interdependencies between the predictor variables. These correlations are biological realities and ought not to be ignored. ANNs consider such nonlinear relationships based on nonlinear mathematics, namely by sigmoid activation functions ***(26)*** and the multilayer concept; are independent from the partially rigid guidelines as they are known from conventional approaches like the *linear discriminant analysis*; and do not need to fulfill the assumptions of any well-defined distributions (e.g., the *Gaussian distribution*). There are different concepts of ANNs and linear statistical models (**Fig. 6**).

Based on these advantages ANNs generally produce better overall results in classification ***(27)*** and to some degree better results when compared to experts in the corresponding fields ***(28–33)***. ANNs may also outperform the *logistic regression* (LR), which is considered to be the *gold standard* in biomedical research (**Fig. 7**).

### *2.4. Statistical Approaches*

Standard statistical approaches, such as the *linear discriminant analysis* and LR, are easily implementable but assume independence among all inputs

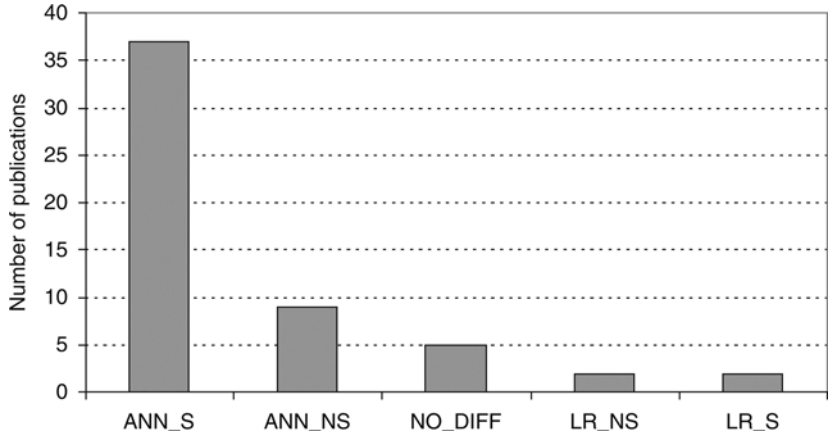


Fig. 7. MEDLINE search for "logistic regression neural network" comprising the years 2002 to 2004. In 11 publications the difference in accuracy was not significant or was not investigated regarding significance (ANN: $n = 9$, LR: $n = 2$). In 15 publications there was no comparison given (meta-analysis, reviews, and so on) and six publications were not available to the authors. The ANN classifies more accurately with significance (ANN_S) or without significance (ANN_NS). No difference between ANN and LR regarding accuracy (NO_DIFF). The LR classifies more accurately without significance (LR_NS) or with significance (LR_S).

and the existence of a linear relationship between input and output variables. Although *linear discriminant analysis* is incapable of handling interactions between predictor variables, using the LR approach, nonlinear input-output-relationships can be explicitly modeled and interactions between variables can be clearly defined. Based on the assumption of complexity in the relationships, the LR modeler might fail to consider all of the nonlinearities, although progress has been achieved through the modeling of nonlinear relationships through *fractional polynomials*, therefore these complexities as well as interactions are factors that can automatically be considered ***(34,35)***.

### 2.4.1. Classification Trees

*Classification trees* implicitly perform a step-by-step variable selection and—assuming binary predictor variables—are easy to interpret. Trees may be bushy when genes are used as input and *multisplit functions* are employed. Moreover, classification trees have a tendency to be unstable and lacking in accuracy. Their accuracy can be greatly improved by aggregation (*bagging* or *boosting*; *see* **Subheading 2.6.**). However, some simplicity is lost by aggregating trees.

### 2.4.2. K-Nearest Neighbor Classifier

*Nearest neighbor classifiers* are simple, intuitive and have remarkably low error rates when compared to more sophisticated classifiers and as such they are able to handle interactions between genes, they do so using a "black-box" approach and give very little insight into the structure of the data. As a disadvantage predictor variables are not weighted based on their discriminatory power, especially when using the most widespread *Euclidian distance*. The introduction and use of alternative distance measures, for example, the *Malahanobis distance*, classification could possibly fail if the expression levels of the (few) observations that are made are not distributed normally or if any one class comprises more than one cluster.

### 2.4.3. Support Vector Machines

SVMs are receiving increased attention and have been applied successfully to microarray gene expression cancer diagnosis ***(36–39)***, however they require more training than the methods previously discussed (e.g., choice of kernel function $K$ and scale factor $\lambda$). Additionally, SVMs are difficult to interpret if there are diverse support vectors (as is usually the case) and to date could not consistently be proved to be more accurate than alternative classification methods or even to solve problems where other methods failed ***(40)***.

In the case of DNA microarray analysis, a number of classification methods have been compared ***(41)***, and it was concluded that—beside the *k-nearest neighbor—backpropagation neural networks* are the best classifiers for that purpose. ANNs have been demonstrated to be nearly perfect in distinguishing different sets of lymphoma patients or predicting the long-term survival of individuals suffering from various lymphoproliferative conditions ***(42)***. When considered from a mathematical perspective the so-called "general function approximation theorem" would suggest that a three-layered ANN with appropriate weights could approximate any arbitrary nonlinear function ***(43)***. For this reason ANNs can serve as universal approximators.

## 2.5. What Are the Disadvantages and How to Overcome These Problems?

The criticism most frequently proffered is that because ANNs converge slowly, the learning phase may be stuck in local minima and so achieve poor accuracy; numerous learning parameters as well as the topology of the network have to be optimized by an expert; and the trained ANN resembles a "black box," thereby, hampering any interpretation of the classifier's response ***(44)***.

Since the late 1980s when ANNs began attracting increased attention, these problems have become well known and so various efforts have been undertaken

to overcome these drawbacks. Acceleration of this convergence may be facilitated by, numerous new learning algorithms being described in **refs. *45–47***; second order methods being introduced ***(26,48,49)***; the *standard logistic activation* function being replaced by the *tangent hyperbolic* ***(50)***; and approaches to eliminate flat spots in the derivation of the *activation function* ***(51)*** or modifications of the error function, for example, *cross-entropy* ***(52–55)***. To improve the generalization accuracy, local adaptive learning rates were established ***(56–58)***, weight initialization was optimized ***(59–61)***, and modular approaches were tested ***(62)***. To facilitate more intense learning data were *jittered* ***(63–65)*** or ensembles of ANNs were trained for majority or weighted voting. The body of literature comprises reviews of all these improvements ***(66–70)***. In the meanwhile most data analysis tools offer a neural network analysis, which considers some of the aforementioned learning strategies. A selection of these tools allows the user to implement further strategies ***(71)*** by programming additional routines. Even if one is not a computer-scientist there is a facet one can easily adjust in order to improve the accuracy, namely the size of the network structure, i.e., to increase considerably the number of hidden neurons. Although this may be considered time-consuming, 100 hidden neurons is determined a good choice, overcompensating for the network structure, however the current theory speaks to the efficacy of choosing the smallest admissible size that will provide a solution because many researchers believe that the simplest architecture is that which is most suitable for generalization. Notwithstanding this belief, several neural net empiricists have shown that surprisingly good generalization can be achieved with oversized feed-forward networks ***(72–76)***. From the experience of the authors, overcompensating the network architecture leads to a marked increase in the generalization performance although the network structure does not require additional incremental adjustments. Moreover, the capacity is sufficient to face almost every classification problem. There is however, one condition that must be adhered to: overtraining the ANN must be avoided.

*Overtraining* or *overfitting* can occur during the learning phase, when the error in the training set decreases more or less steadily, while the error in unseen patterns begins to deteriorate. This usually occurs in the later stages of learning. Before reaching this point, the network learns the general characteristics of the classes; afterwards, it takes advantage of some idiosyncrasies in the training data that aggravate generalization performance (**Fig. 8**). Several theoretical studies have evaluated the optimal stopping time ***(77–78)***. One common approach toward avoiding overfitting is *early stopping*, which consists of estimating the generalization performance during training (with an extra validation set removing some patterns from the training data) and stopping when it begins to decrease. This technique has been reported to be superior to other regularization methods in many cases ***(79)***. However, the real situation is somewhat more complex, in that

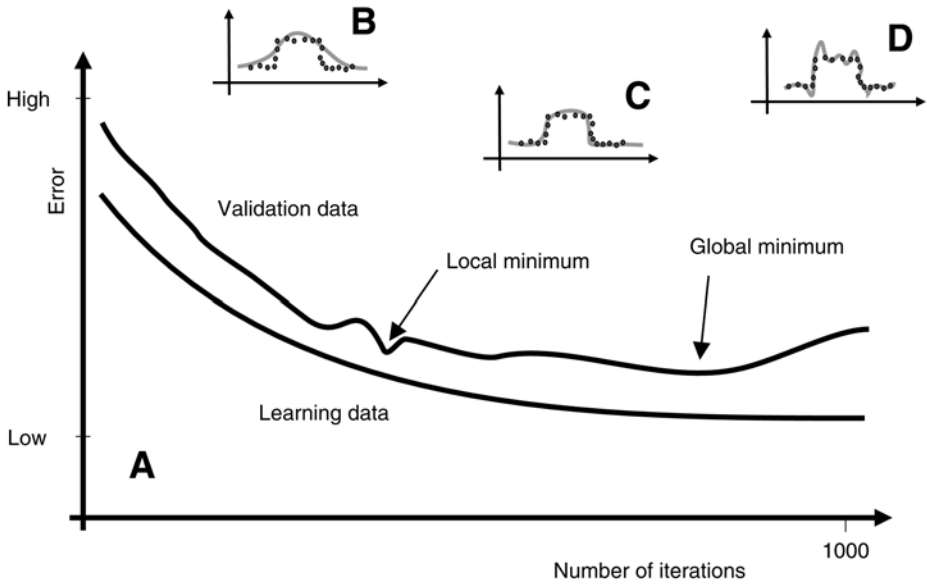


Fig. 8. Rectangle function as an example for *underfitting* and *overfitting*. Learning the rectangle function by an ANN **(A)**. Approximation accuracy is still inadequate **(B)**. The characteristic of the function has been learned **(C)**. All sampling points have been precisely fitted but the generalization performance has worsened **(D)**. The labeling of the axes applies to all graphs **(A–D)**.

generalization curves almost always have more than one local minimum. In light of this, 14 different automatic-stopping criteria have been developed ***(80)***.

In order not to waste valuable data for the extra validation set, an ensemble technique can be used, with each ANN consisting of an ensemble of five modules. In this technique, all training data are divided into five equally sized sets, designated as "A," "B," "C," "D," and "E." The first module is trained by sets A, B, C, and D, and set E serves as a validation set; the second module is trained by sets A, B, C, and E, and set D is the validation set; this is continued until all the possible combinations are complete (**Fig. 9**). To obtain a single common result for each class, the outputs of the five modules can be (weighted) averaged. Early stopping can therefore be performed using an extra validation set and all training patterns can effectively be used for training.

Of central importance to ANNs is that the weights at the connections are "learned" during training of the network. "Experience" in the trained network is stored in these interconnection weights ***(32)***. To open the black box and therefore to give insight into the response behavior, one can either try to extract rules

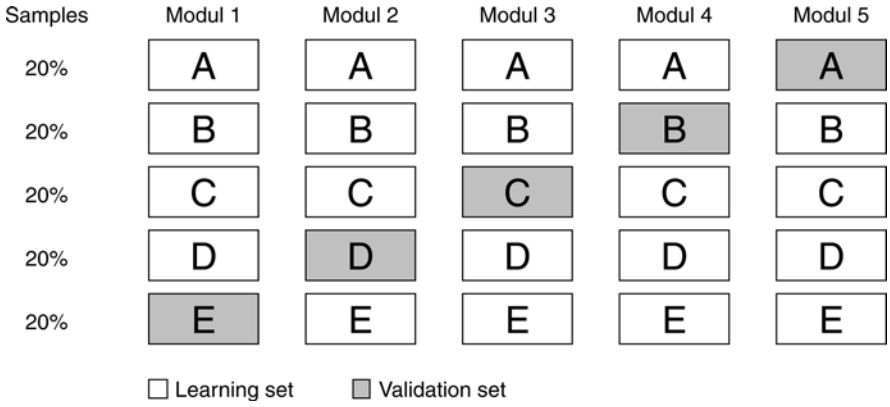


Fig. 9. A graphic representation of the learning set and the validation set within five modules.

from the weights ***(81–83)*** or perform so-called sensitivity analyses, i.e., systematically varying the input while observing the output. For instance, the ANN software tool *BrainMaker* ***(84)*** (California Scientific®, Nevada City, USA) offers a sensitivity analysis. Another approach, which is easily implementable and understood is the *Causal Index* introduced by Baba ***(85)*** or its modified version *MCI* ***(86)*** also considering the sigmoid curve progression of the activation function. The **Eqs. 1** and **2** indicate how the impact of an input neuron *i* regarding an output neuron *k* can be quantified as having a *Causal Index* or *MCI* value. Where, *j* denotes the hidden neurons, $w_{ij}$ is the weight from neuron *i* to neuron *j* (**Fig. 5**).

$$CI_{ik} = \sum_{j=1}^{n} w_{ij} \times w_{jk} \tag{1}$$

$$MCI_{ik} = \sum_{j=1}^{n} \tanh\,(w_{ij}) \times \tanh\,(w_{jk}) \tag{2}$$

Irrespective of the common preconception regarding the black box character of ANNs it has been stated that "(…) ANNs are a superior tool for digesting microarray data both with regard to making distinctions based on the data and with regard to providing very specific reference as to which genes were most important in making the correct distinction in each case," ***(42)*** (*see* **Note 4**).

Furthermore, detailed discussions about the advantages and disadvantages of the ANNs have been published previously ***(87–89)***.

### *2.6. How to Combine the Power of Multiple ANNs?*

In order to make learning more robust as well as to achieve a more accurate classification *committee* or *ensemble techniques* may be employed ***(90–91)***.

To attain this objective, several classifiers are specified for the same classification task and in this way produce one common result based on their collective voting. The ANNs involved can be differentiated based on their network architecture ***(92)***, different initialization procedures used ***(93)*** or learning data permutated as outlined above. Applying ensembles is not only restricted to ANNs, other learning algorithms can be grouped as ensembles, for example, SVMs ***(94)***. Different voting schemes can be used such as the *simple averaging method* ***(95–97)***, the *weighted averaging* ***(96,98,99)***, or the *majority voting scheme* or *plurality voting* ***(100,101)***. Further voting rules are the *Maximum Vote*, the *Borda Count* or the *Nash Vote* ***(102)***.

Moreover, generating perturbed versions of the learning set can be done by *bagging* and *boosting* ***(103,104)***. In the *bootstrap aggregating* or *bagging* procedure ***(105)***, perturbed learning sets of the same size as the original learning set are formed by forming bootstrap replicates of the learning set. In *boosting* ***(106)*** the data are resampled adaptively so that the weights in the resampling are increased for those cases most often misclassified. Hereby the aggregation of predictors is done by weighted voting.

Although an ensemble is a collection of different classifiers, for example, ANNs, specified for the same task, ANNs may also be trained for different subtasks. Those approaches are categorized into *mixture of experts*. For example, a two-level ANN has displayed superior performance over a single-level ANN ***(107)***. The task was to differentiate chest radiographs with lung nodules from those without lung nodules. To concentrate ANN-learning on patterns difficult to assign, first an ANN differentiated between patterns being "possible a nodule" from those being "probably no nodule." In the second step—when the "probably no nodule" patterns have been sorted out—the second ANN focused on the remaining patterns which were difficult to classify (**Fig. 10**).

A two-level ensemble architecture distinguishes normal cells from four types of malignant cells (adenocarcinoma, squamous cell carcinoma, small cell carcinoma, large cell carcinoma) using concepts of varying complexity ***(108)***. The first-level ANN ensemble classifies higher-level concepts (benign or malign), whereas the second-level ensemble is used to classify lower-level concepts (concrete malign types). Then a cell is only classified by the second ensemble if at least one individual ANN of the first ensemble concludes it is malign (**Fig. 11**).

Dealing with a multicategory classification problem comprising three classes, the same strategy can be employed. To predict the three possible states of the protein secondary structure ($\alpha$-helix, $\beta$-strand, or coil) from protein sequences, an ensemble of different neural networks for state prediction can be used in a first step ***(109)***. Positions where there was a full agreement between all ensemble members were taken as the final prediction. In case of no final outcome a second ANN obtained the final prediction (**Fig. 12**).

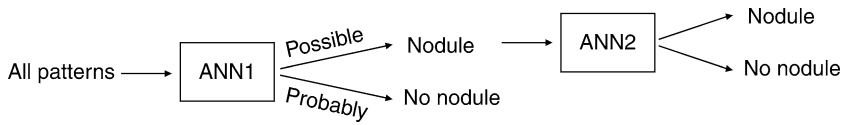


Fig. 10. Two-level architecture, two-class classification problem.

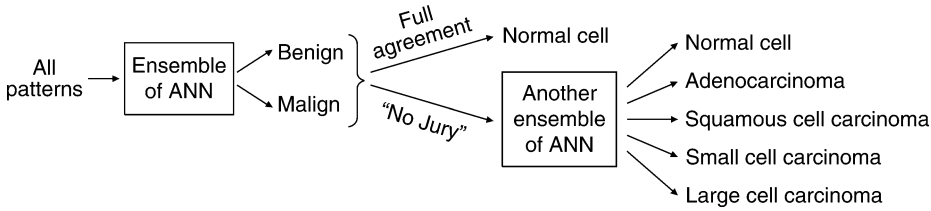


Fig. 11. Two-level architecture, classifying higher-level and lower-level concepts.

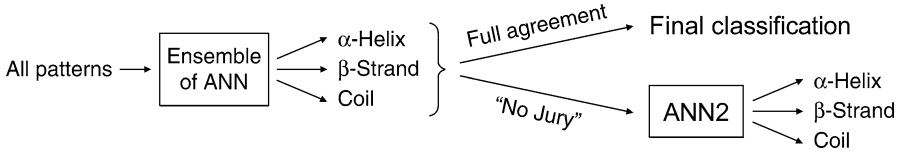


Fig. 12. Two-level architecture, multiclass classification problem.

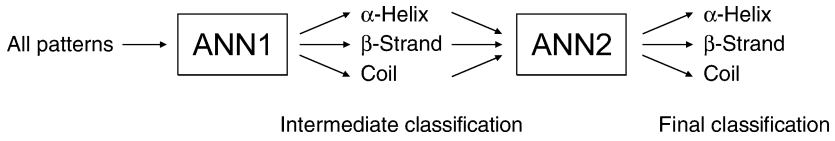


Fig. 13. Usage of output as input.

Another way to combine two ANNs is to simply use the output of the first ANN (protein secondary structure) as an input for a second ANN that finally makes the prediction (**Fig. 13**). Employing a consecutive structure–structure network is reported to result in a 2% improvement in prediction ***(110)***.

The *one-vs-all* approach builds $k$ (the number of classes) binary classifiers which distinguish one class from all other classes added together. A sample is assigned to the corresponding class label of the binary classifier achieving the greatest output activity (**Fig. 14**).

Similarly, the all-pairs approach builds $k(k-1)/2$ binary classifiers. For each class there are $k$–1 relevant binary classifiers, which distinguish it from the other classes. The output activities of those $k$–1 binary classifiers are summed up, and

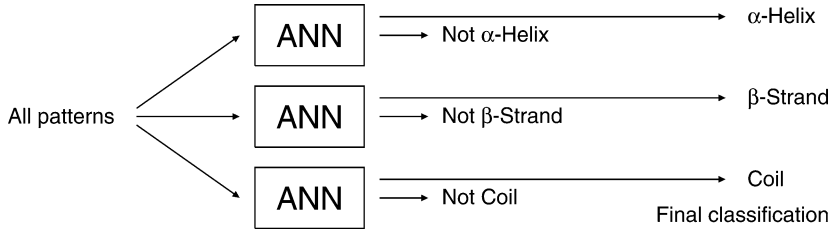


Fig. 14. *One-vs-all* approach.

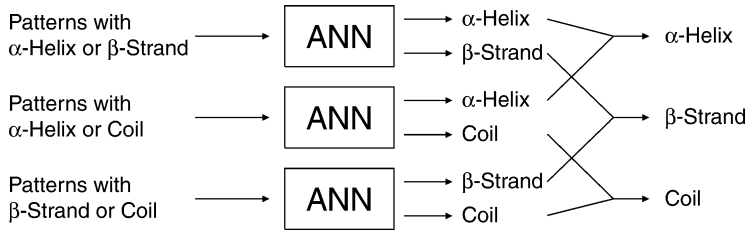


Fig. 15. *All-pairs* approach.

the class with the greatest overall activity is the winning class (**Fig. 15**). Moreover, there are more sophisticated approaches that somehow combine the one-vs-all approach with the all-pairs approach ***(111)***.

The *Subsequent ANN (SANN)* approach ***(112)*** is oriented to the human decision making process. In this situation, a process of exclusion occurs which is the first step where preferred choices are selected and included in the "narrowed-down choice," after which the final decision is made in a succeeding step. This means that the classification made by the first ANN is interpreted as a preselection to be followed by a final categorization by a successive, second application of ANN. Classification is therefore concentrated on the two primary classes, i.e., the two most preferred classes with the highest activities of the corresponding output neurons (**Fig. 16**).

When comparing these amalgamations of experts (as the authors have published elsewhere ***[112]***) the all-pairs method and the SANN approach clearly outperform the other methods, probably because they concentrate on classification problems comprising only of patterns of two adjacent classes. However, the all-pairs method leads to intense computing on both, classes which are difficult to separate, and classes which are readily separable. This shortcoming does not apply to the SANN approach calculating no. 2-class classification problems of classes easy to separate. Finally, it should be emphasized that mixtures of experts are—like the ensembles techniques—by no means restricted to ANNs. An example combining four different types of predictors has been reported previously ***(113)***.

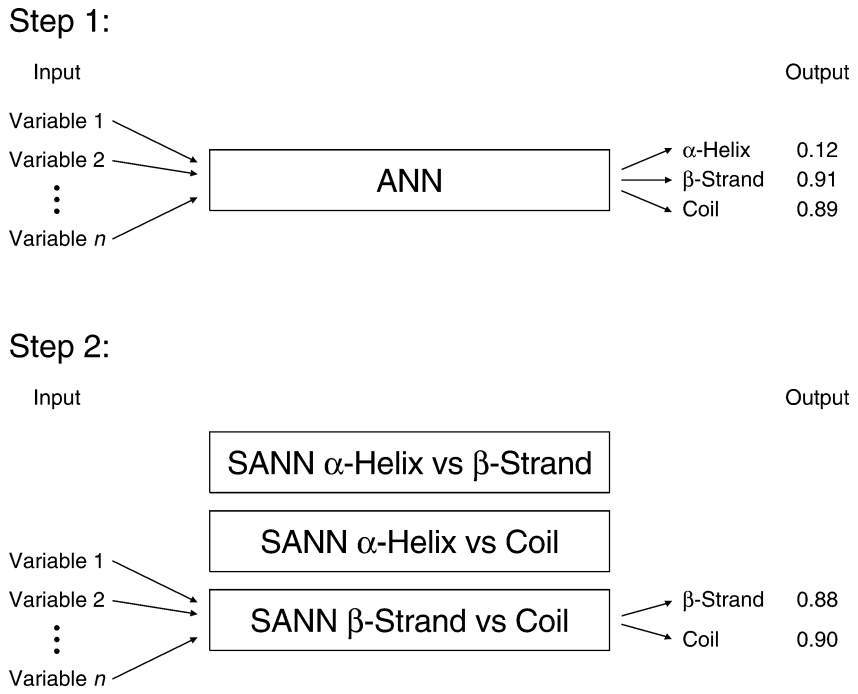


Fig. 16. Paradigmatic three-class problem. As an example (fictive values), the pattern, with parameter 1–*n,* will be primarily categorized as β-strand. A subsequent ANN (SANN) trained to discriminate between β-strand and coil conclusively classifies the pattern in the sense that the secondary structure will be a coil.

## 3. Closing Remarks

Whether the theoretical superiority of highly sophisticated methods as compared with more simple methods is really evident in practical applications is debatable. Comparisons of methods often do not consider important parameters such as population drift, sample selectivity bias, errors in class labels, arbitrariness in the class definition, or misleading optimization criteria and performance assessment ***(114)***. This is observed particularly in the analysis of gene expression data, where possible errors may occur with the marking of the mRNA by fluorescent nucleotides, when producing the chip, during the hybridization procedure, scanning the arrays or in the course of image processing ***(115)***. With this in mind, the importance in searching for only slightly more accurate algorithms makes this all relative. Deciding between two algorithms is mainly guided by the availability of a corresponding computer program and the experience to work with it, therefore, researchers who are experienced in applying a certain classification method will not really be affected by the views presented.

For researchers who are interested in a potent classification procedure, neural network technology might be a powerful alternative to other existing methods.

This also applies to the authors of the present report and other researchers in the field of systems biology who increasingly consider expression rates of genes as they are provided by the microarray analysis to modify their mathematical models ***(116,117)***. Simulating cell behavior, various multicategory classification problems have to be faced. The present work will hopefully be supportive with regards to this approach.

## 4. Notes

### *4.1. Preprocessing*

Preprocessing is as important a consideration as the subsequent classification procedure and includes a number of aspects. Although facets such as data cleaning, coding of nominal variables or feature extraction are less applicable to the analysis of microarray data, some essential steps are the imputation of missing data, the normalization, the logarithmic transformation, and the standardization of data.

#### *4.1.1. Imputation of Missing Data*

Some of the discriminatory methods are able to deal with missing data (e.g., *classification trees*); however, others require complete data and to this end there exists a comprehensive body of literature on how to input missing data ***(118)***. Interestingly, even ANNs are suited to the substitution of missing values ***(119,120)***. A very pragmatic approach is to use a simple $k$ nearest neighbor algorithm, in which the neighbors are the genes and the distance between neighbors is based on their correlation ***(121)***. For each gene with missing data: (1) compute its correlation with all other $n - 1$ genes, and (2) for each missing entry, identify the $k$ nearest genes having complete data for this entry and impute the missing entry by the average of the corresponding entries for the $k$ neighbors. Setting $k = 5$ has been proved to work successfully ***(121)***.

#### *4.1.2. Normalization*

In order for the comparison of gene expression levels of different microarray to be made, the signals must be normalized. To achieve this, *housekeeping genes* can be employed ***(122,123)*** or the average intensity of all genes (or a subset of the genes) can be used for scaling ***(124,125)***. Additionally, artificial transcripts can be added with a defined behavior suited for normalization ***(2)***.

#### 4.1.3. Logarithmic Transformation

It is customary that the expression signals are log-transformed and in that regard the base 10 logarithmic transformation ***(17)*** is as well used as the base 2 logarithm ***(8,126)***.

#### 4.1.4. Standardization

Conventional practice allows the use of the correlation between the gene expression profiles of two mRNA samples to measure their similarity ***(8,126,127)***. Consequently, observations are standardized to have mean 0 and variance 1 across variables (genes). With the data standardized in this manner, the distance between two mRNA samples may be measured by their Euclidean distance, for example, important for the *k nearest neighbor algorithm*.

### 4.2. Gene Selection

When administering an ANN, it makes for greater efficiency if gene selection is also done using an ANN, for example, using the *Neural Net Clamping Technique* ***(128)***. This technique exemplifies a kind of backward search, the process involves starting with $n$ input neurons; training the ANN only once, including all features, yet testing the ANN $n$ times; and setting (or clamping) the considered feature to its mean value over all test patterns. Should the feature with the smallest contribution to classification be omitted, the search continues with $n$-1 input neurons. This procedure is repeated until the best feature set is found and when compared to a standard backward search, the ANN only has to be trained once to leave out one feature.

Based on computational challenges experienced when starting the selection procedure with thousands of genes, a preselection has to be performed, for example, by a common *univariate method* (based on the *BSS/WSS criterion* ***[121]***, *S2N ratio* ***[17]***, *Wilcoxon test* ***[129]***, *t-Statistic* ***[121]***, or the *misclassification rate* ***[130]***). Increasingly, *multivariate approaches* are being reported ***(91,131,132)*** that—in contrast to the *univariate methods*—also consider possible interdependencies between the genes. Reviews on this issue have been previously published ***(133,134)***.

### 4.3. Statistical Evaluation

The disproportionately high number of possible predictor variables (genes) associated with only few observations may result in many false-positive findings. To avoid this, one should be careful to ensure that either multiple statistical testing can be performed ***(135,136)*** or validation is done by a $k$-fold cross-validation, which entails splitting the total amount of records into $k$ equally sized and representative subsets. This method requires deriving ANN

calculations using *(k-1)/k* records and evaluating the remaining *1/k* records, repeating this procedure *k* times to obtain a final result based on all of the records. The most commonly used values for *k* are 5 or 10. Setting $k = n$, cross-validation is also called the "*Leave One Out Method*," which is suited for very small sample sizes. For $n < 50$, some researchers have used *bootstrapping* or its variants ***(137)***. The most current variant is 632B+ ***(138)***.

### *4.4. Integration of Microarray Data Into Cell or Tissue Simulations (Mathematical Models)*

Pathway information based on microarray analyses may become increasingly vital for successful modeling of biological systems. Such integrated systems will greatly facilitate the constructive cycle of computational model building and experimental verification that lies at the heart of systems biology. One of the major issues is the modeling of transition, for example, normal cell to tumor cell. Mathematical models may be designed using the results of ANN-based microarray data analyses to describe transition. A simple equation can hence be formulated where $x_t$ represents a given condition at starting point t, and $x_{t+1}$ map to the condition at time point $t + 1$ while φ represents factors of influence **(Eq. 3)**.

$$x_{t+1} = f(x_t) + \varphi \tag{3}$$

The ANN-based classification may help model the transition function $f$ in which the corresponding microarray data are assigned to both conditions ($x_t$ and $x_{t+1}$).

## References

1. Schena, M. (ed.) (1999) *DNA Microarrays: A Practical Approach*. Oxford University Press, Oxford.
2. Victor, A., Klug, S., and Blettner, M. (2005) cDNA-microarrays—strategien zur bewältigung der datenflut. *Deutsches Ärzteblatt* **102,** 355–360.
3. Quackenbush, J. (2001) Computational analysis of microarray data. *Nat. Rev. Genet.* **2,** 418–427.
4. Ringner, M. and Peterson, C. (2003) Microarray-based cancer diagnosis with artificial neural networks. *Biotechniques Suppl.* 30–35.
5. Gu, C., Rao, D., Stormo, G., Hicks, C., and Province, M. (2002) Role of gene expression microarray analysis in finding complex disease genes. *Genet. Epidemiol.* **23,** 37–56.
6. Eisen, M., Spellman, P., Brown, P., and Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA* **95,** 14,863–14,868.
7. Tamayo, P., Slonim, D., Mesirov, J., et al. (1999) Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoetic differentiation. *Proc. Natl. Acad. Sci. USA* **96,** 2907–2912.

8. Alizadeh, A., Eisen, M., Davis, R., et al. (2000) Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature* **403,** 503–511.
9. Perou, C., Sørlie, T., Eisen, M., et al. (2000) Molecular portraits of human breast tumours. *Nature* **406,** 747–752.
10. Sørlie, T., Perou, C., Tibshirani, R., et al. (2001) Gene-expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proc. Natl. Acad. Sci. USA* **98,** 10,869–10,874.
11. Chang, J., Wooten, E., Tsimelzon, A., et al. (2003) Gene expression profiling for the prediction of therapeutic response to docetaxel in patients with breast cancer. *Lancet* **362,** 362–369.
12. van de Vijver, M., He, Y., van't Veer, L., et al. (2002) A gene-expression signature as a predictor of survival in breast cancer. *New Engl. J. Med.* **347,** 1999–2009.
13. Broekhuijsen, M., Larsson, P., Johansson, A., et al. (2003) Genome-wide DNA microarray analysis of *Francisella tularensis* strains demonstrates extensive genetic conservation within the species but identifies regions that are unique to the highly virulent F. tularensis subsp. tularensis. *J. Clin. Microbiol.* **41,** 2924–2931.
14. Li, J., Chen, S., and Evans, D. (2001) Typing and subtyping influenza virus using DNA microarrays and multiplex reverse transcriptase PCR. *J. Clin. Microbiol.* **39,** 696–704.
15. Bekal, S., Brousseau, R., Masson, L., et al. (2003) Rapid identification of *Escherichia coli* pathotypes by virulence gene detection with DNA-microarrays. *J. Clin. Microbiol.* **41,** 2113–2125.
16. Fukushima, M., Kakinuma, K., Hayashi, H., Nagai, H., Ito, K., and Kawaguchi, R. (2003) Detection and identification of *mycobacterium* species isolates by DNA microarray. *J. Clin. Microbiol.* **41,** 2605–2615.
17. Golub, T., Slonim, D., Tamayo, P., et al. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286,** 531–537.
18. Tafeit, E., Möller, R., Sudi, K., and Reibnegger, G. (1999) The determination of three subcutaneous adipose tissue compartments in non-insulin-dependent diabetes mellitus women with artificial neural networks and factor analysis. *Artif. Intell. Med.* **17,** 181–193.
19. James, W. (1890) The principles of psychology, in *Neurocomputing: Foundations of Research,* (Anderson, J. and Rosenfeld, E., eds.), Henry Holt and Co. New York, NY, USA.
20. Rosenblatt, F. (1958) The percepton: a probabilistic model for information storage and organization in the brain. *Psycholog. Rev.* **65,** 386–408.
21. Rumelhart, D., Hinton, G., and Williams, R. (1986) Learning representations by back-propagating errors. *Nature* **323,** 533–536.
22. Jenuwine, E. and Floyd, J. (2004) Comparison of medical subject headings and text-word searches in MEDLINE to retrieve studies on sleep in healthy individuals. *J. Med. Libr. Assoc.* **92,** 349–353.
23. Kuller, A., Wessel, C., Ginn, D., and Martin, T. (1993) Quality filtering of the clinical literature by librarians and physicians. *Bull. Med. Libr. Assoc.* **81,** 38–43. Erratum in *Bull. Med. Libr. Assoc.* **81,** 233.

24. Lotka, A. (1926) Frequency distribution of scientific productivity. *Journal of the Washington Academy of Sciences* **16,** 317–325.
25. Bradford, S. (ed.) (1953) *Documentation*. 2nd ed., Crosby Lockwood, London.
26. Bishop, C. (ed.) (1995) *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford.
27. Penny, W. and Frost, D. (1996) Neural networks in clinical medicine. *Med. Decis. Making* **16,** 386–398.
28. Baxt, W. and Skora, J. (1996) Prospective validation of artificial neural network trained to identify acute myocardial infarction. *Lancet* **347,** 12–15.
29. El-Solh, A., Hsiao, C. -B., Goodnough, S., Serghani, J., and Grant, B. (1999) Predicting active pulmonary tuberculosis using an artificial neural network. *Chest.* **116,** 968–973.
30. Bottaci, L., Drew, P., Hartley, J., et al. (1997) Artificial neural networks applied to outcome prediction for colorectal cancer patients in separate institutions. *Lancet* **350,** 469–472.
31. Burke, H., Goodman, P., Rosen, D., et al. (1997) Artificial neural networks improve the accuracy of cancer survival prediction. *Cancer* **79,** 857–862.
32. Geddes, C., Fox, J., Allison, M., Boulton-Jones, J., and Simpson, K. (1998) An artificial neural network can select patients at high risk of developing progressive IgA nephropathy more accurately than experienced nephrologists. *Nephrol. Dial. Transplant* **13,** 67–71.
33. Jiang, Y., Nishikawa, R., Wolverton, D., et al. (1996) Malignant and benign clustered microcalcifications: automated feature analysis and classification. *Radiology* **198,** 671–678.
34. Royston, P. and Sauerbrei, W. (2003) Stability of multivariable fractional polynomial models with selection of variables and transformations: a bootstrap investigation. *Stat. Med.* **22,** 639–659.
35. Royston, P. and Sauerbrei, W. (2004) A new approach to modeling interactions between treatment and continuous covariates in clinical trials by using fractional polynomials. *Stat. Med.* **23,** 2509–2525.
36. Statnikov, A., Aliferis, C., Tsamardinos, I., Hardin, D., and Levy, S. (2005) A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics* **21,** 631–643.
37. Lee, Y. and Lee, C. (2003) Classification of multiple cancer types by multicategory support vector machines using gene expression data. *Bioinformatics* **19,** 1132–1139.
38. Ramaswamy, S., Tamayo, P., Rifkin, R., et al. (2001) Multiclass cancer diagnosis using tumor gene expression signatures. *Proc. Natl. Acad. Sci. USA* **98,** 15,149–15,154.
39. Furey, T., Cristianini, N., Duffy, N., Bednarski, D., Schummer, M., and Haussler, D. (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* **16,** 906–914.
40. Hearst, M. (1998) Support vector machines. *IEEE Intell. Syst.* **13,** 18–28.

41. Cho, S. -B. and Won, H. (2003) *Machine Learning in DNA Microarray Analysis for Cancer Classification* in Chen, Y. -P. (ed.). First Asia-Pacific Bioinformatics Conference (APBC 2003). Adelaide, Australia: CRPIT 19 Australian Computer Society 2003, pp. 189–198.
42. O'Neill, M. and Song, L. (2003) Neural network analysis of lymphoma microarray data: prognosis and diagnosis near-perfect. *BMC Bioinformatics* **4,** 13.
43. Hornik, K., Stinchcombe, M., and White, H. (1989) Multilayer feedforward networks are universal approximators. *Neural Networks* **2,** 359–366.
44. Benítez, J., Castro, J., and Requena, I. (1997) Are artificial neural networks black boxes? *IEEE Transactions on Neural Networks* **8,** 1156–1164.
45. Riedmiller, M. and Braun, H. (1993) A direct adaptive method for faster backpropagation learning, in *The RPROP Algorithm* (Ruspini, E., ed.), IEEE International Conference on Neural Networks. San Francisco, CA, pp. 586–591.
46. Zimmermann, H. and Neuneier, R. (1998) The observer-observation dilemma in neuro-forecasting, in *Advances in Neural Information Processing Systems* (Jordan, M. I., Kearns, M. J., and Solla, S. A., eds.), MIT Press, pp. 992–998.
47. Fahlman, S. and Lebiere, C. (1990) The cascade-correlation learning architecture, in *Advances in Neural Information Processing Systems,* (Touretzky, D., ed.), Morgan Kaufmann, pp. 524–532.
48. Battiti, R. (1992) First- and second-order methods for learning: between steepest descent and Newton's method. *Neural Computation* **4,** 141–166.
49. Shepherd, A. (ed.) (1997) *Second-Order Methods for Neural Networks.* Springer, New York.
50. LeCun, Y., Bottou, L., Orr, G., and Müller, K. -R. (1998) Efficient BackProp, in *Neural Networks: Tricks of the Trade* (Orr, G., & Müller, K. R., eds.), Springer, Berlin, pp. 9–50.
51. Fahlman, S. (1988) *An Empirical Study of Learning Speed in Backpropagation*. Carnegie Mellon University.
52. Humpert, B. (1994) Improving back propagation with a new error function. *Neural Networks* **7,** 1191–1192.
53. Oh, S. (1997) Improving the error backpropagation algorithm with a modified error function. *IEEE Trans. Neural Networks* **8,** 799–803.
54. Solla, S., Levin, E., and Fleisher, M. (1988) Accelerated learning in layered neural networks. *Complex Syst.* **2,** 625–639.
55. van Ooyen, A. and Nienhuis, B. (1992) Improving the convergence of the backpropagation algorithm. *Neural Networks* **5,** 465–471.
56. Tollenaere, T. (1990) SuperSAB: fast adaptive back propagation with good scaling properties. *Neural Networks* **3,** 561–573.
57. Jacobs, R. (1988) Increased rates of convergence through learning rate adaptation. *Neural Networks* **1,** 295–307.
58. Linder, R., Wirtz, S., and Pöppl, S. (2000) Speeding up backpropagation learning by the APROP algorithm in *Proceedings of the Second International ICSC Symposium on Neural Computation*, (Bothe, H. and Rojas, R., eds.), Berlin, Germany: ICSC Academic Press, Millet, pp. 122–128.

59. Weymaere, N. and Martens, J. (1994) On the initialization and optimization of mutilayer perceptrons. *IEEE Trans. Neural Networks* **5,** 738–750.
60. Yam, Y., Chow, T., and Leung, C. (1997) A new method in determining initial weights of feedforward neural networks for training enhancement. *Neurocomputing* **16,** 23–32.
61. Lehtokangas, M., Saarinen, J., Kaski, K., and Huuhtanen, P. (1995) Initializing weights of a multilayer perception by using the orthogonal least squares algorithm. *Neural Computation* **7,** 982–999.
62. Anand, R., Mehrotra, K., Mohan, C., and Ranka, S. (1995) Efficient classification for multiclass problems using modular neural networks. *IEEE Trans. Neural Networks* **6,** 117–124.
63. Rögnvaldsson, T. (1994) On Langevin updating in multilayer perceptrons. *Neural Computation* **6,** 916–926.
64. Murray, A. and Edwards, P. (1993) Synaptic weight noise during multilayer perceptron training: Fault tolerance and training improvements. *IEEE Trans. Neural Networks* **4,** 722–725.
65. Grandvalet, Y., Canu, S., and Boucheron, S. (1997) Noise injection: theoretical prospects. *Neural Computation* **9,** 1093–1108.
66. Barnard, E. and Holm, J. (1994) A comparative study of optimization techniques for backpropagation. *Neurocomputing* **6,** 19–30.
67. Alpsan, D., Towsey, M., Ozdamar, O., Tsoi, A., and Ghista, D. (1995) Efficacy of modified backpropagation and optimisation methods on a real-world problem. *Neural Networks* **8,** 945–962.
68. Orr, G. M. K. -R. (ed.) (1998) *Neural Networks: Tricks of the Trade*. Springer, New York.
69. Looney, C. (1996) Stabilization and speedup of convergence in training feedforward neural networks. *Neurocomputing* **10,** 7–31.
70. Linder, R. and Pöppl, S. (2001) ACMD: a practical tool for automatic neural net based learning. *Lect. Notes Comp. Sci.* **2199,** 168–173.
71. Stuttgarter *Stuttgarter Neuronale Netze Simulator*. http://www-ra.informatik.uni-tuebingen.de/SNNS (as of May 5th, 2005).
72. Amirikian, B. and Nishimura, H. (1994) What size network is good for generalization of a specific task of interest? *Neural Networks* **7,** 321–329.
73. Murata, N. (1996) An integral representation of functions using three-layered networks and their approximation bounds. *Neural Networks* **9,** 947–956.
74. Kröse, B. and van der Smagt, P. (ed.) (1993) *An Introduction to Neural Networks*. 5, University of Amsterdam,
75. Bartlett, P. (1993) Vapnik-Chervonenkis dimension bounds for two- and three-layer networks. *Neural Computation* **5,** 371–373.
76. Lewicki, M. and Sejenowski, T. (2000) Learning overcomplete representations. *Neural Computation* **12,** 337–365.
77. Amari, S., Murata, N., Müller, K. -R., Finke, M., and Yang, H. (1997) Asymptotic statistical theory of overtraining and cross-validation. *IEEE Trans. Neural Networks* **8,** 985–996.

78. Wang, C., Venkatesh, S., and Judd, J. (1995) Optimal stopping and effective machine complexity in learning. *Adv. Neural Inf. Processing Syst.* **6,** 303–310.
79. Finoff, W., Hergert, F., and Zimmermann, G. (1993) Improving model selection by nonconvergent methods. *Neural Networks* **6,** 771–783.
80. Prechelt, L. (1998) Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks* **11,** 761–767.
81. Bologna, G. (1996) Rule extraction from the IMLP neural network: a comparative study. Proc. of the NIPS workshop of rule extraction from trained artifical neural networks. Snowmass, CO.
82. Setiono, R. and Liu, H. (1997) NeuroLinear: from neural networks to oblique decision rules. *Neurocomputing* **17,** 1–24.
83. Towell, G. and Shavlik, J. (1993) Extracting refined rules from knowledge based neural networks. *Machine Learning* **13,** 71–101.
84. Lawrence, J. and Frederickson, J. (eds.) (1993) *BrainMaker Professional User's Guide and Reference Manual*, 4*th*, California Scientific Software Press, Nevada City, CA.
85. Baba, K., Enbutu, I., and Yoda, M. (1990). Explicit representation of knowledge acquired from plant historical data using neural network in *Int. Joint Conf. on Neural Networks* (Caudill, M., ed.), San Diego, CA, pp. 155–160.
86. Linder, R., Theegarten, D., Mayer, S., et al. (2003) Der Einsatz eines Modifizierten Causal Index erleichtert die interpretation des Antwortverhaltens eines mit Daten einer Whole-Body Plethysmographie an einem Knock Out Mausmodell trainierten Artifiziellen Neuronalen Netzwerks (ANN). *Atemw. Lungenkrkh.* **29,** 340–343.
87. Chalfin, D. B. (1996) *Neural Networks: A New Tool for Predictive Models*, (Vincent, J. L, ed.) Springer, Berlin, Germany, pp. 816–829.
88. Tu, J. (1996) Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *J. Clin. Epidemiol.* **49,** 1225–1231.
89. Dreiseitl, S. and Ohno-Machado, L. (2002) Logistic regression and artificial neural network classification models: a methodology review. *J. Biomed. Informa.* **35,** 352–359.
90. Dimopoulos, I., Tsiros, I., Serelis, K., and Chronopoulou, A. (2004) Combining neural network models to predict spatial patterns of airborne pollutant accumulation in soils around an industrial point emission source. *J. Air. Waste Manag. Assoc.* **54,** 1506–1515.
91. Liu, B., Cui, Q., Jiang, T., and Ma, S. (2004) A combinational feature selection and ensemble neural network method for classification of gene expression data. *BMC Bioinforma.* **5,** 136.
92. Rogova, G. (1994) Combining the results of several neural network classifiers. *Neural Networks* **7,** 777–781.
93. Doyle, H., Parmanto, B., Munro, P., et al. (1995) Building clinical classifiers using incomplete observations—a neural network ensemble for hepatoma detection in patients with cirrhosis. *Methods of Inf. Med.* **34,** 253–258.

94. Valentini, G., Muselli, M., and Ruffino, F. (2004) Cancer recognition with bagged ensembles of support vector machines. *Neurocomputing* **56,** 461–466.
95. Hansen, L. and Salamon, P. (1990) Neural networks ensembles. *IEEE Trans. Neural Networks* **12,** 993–1001.
96. Tumer, K. and Ghosh, J. (1995) Order statistics combiners for neural classifiers in *Worlds Congress on Neural Networks,* INNS Press, Washington, DC, pp. 31–34.
97. Munro, P. and Parmanto, B. (1997) Competition among networks improves committee performance, in *Advances in Neural Information Processing Systems,* (Mozer, M., Jordon, M., and Petsche, T., eds.), MIT Press, Cambridge, pp. 592–598.
98. Wolpert, D. (1992) Stacked generalization. *Neural Networks* **5,** 241–259.
99. Hashem, S. (1997) Optimal linear combinations of neural networks. *Neural Networks* **10,** 599–614.
100. Battiti, R. and Colla, A. (1994) Democracy in neural nets: voting schemes for classification. *Neural Networks* **7,** 691–707.
101. Lam, L. and Suen, C. (1995) Optimal combination of pattern classifiers. *Pattern Recognition Lett.* **16,** 945–954.
102. Wanas, N. and Kamel, M. (2001). Feature based decision fusion, in *ICAPR* (Singh, S., Murshed, N., and Kropatsch, W., eds.), Springer-Verlag, Berlin, Heidelberg, pp. 176–185.
103. Carney, J. and Cunningham, P. (1999) *The NeuralBAG* Algorithm: Optimizing Generalization Performance in Bagged Neural Networks in *Proceedings of the 7th European Symposium on Artificial Neural Networks* (Verleysen, M. ed.). pp. 3540.
104. Drucker, H., Schapire, R., and Simard, P. (1993). Improving Performance in Neural Networks Using a Boosting Algorithm, in *Advances in Neural Information Processing Systems* (Hanson, S., Cowen, J., and Giles, C. eds.), Morgan Kaufman, pp. 42–49.
105. Breiman, L. (1996) Bagging predictors. *Machine Learning* **24,** 123–140.
106. Schapire, R. (1990) The strength of weak learnability. *Machine Learning* **5,** 197–227.
107. Lin, J. -S., Lo, S. -C., Hasegawa, A., Freedman, M., and Mun, S. (1996) Reduction of false positives in lung nodule detection using a two-level neural classification. *IEEE Trans. Med. Imag.* **15,** 206–217.
108. Zhou, Z., Jiang, Y., Yang, Y. -B., and Chen, S. -F. (2002) Lung cancer cell identification based on artificial neural network ensembles. *Artif. Intell. Med.* **24**, 25–36.
109. Cuff, J. and Barton, G. (2000) Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins* **40,** 502–511.
110. Qian, N. and Sejenowski, T. (1988) Predicting the secondary structure of globular proteins using neural network models. *J. Molec. Biol.* **202,** 865–884.
111. Yeang, C. -H., Ramaswamy, S., Tamayo, P., et al. (2001) Molecular classification of multiple tumor types. *Bioinformatics* **17,** 316–322.

112. Linder, R., Dew, D., Sudhoff, H., Theegarten, D., Pöppl, S., and Wagner, M. (2004) The "subsequent artificial neural network" (SANN) approach might bring more classificatory power to ANN-based DNA microarray analyses. *Bioinformatics* **20,** 3544–3552.
113. Kittler, J., Hatef, M., Duin, R., and Matas, J. (1988) On combining classifiers. *IEEE Trans. Pattern Anal. Machine Intell.* **20(3),** 226–239.
114. Hand, D. (2004) Academic obsessions and classification realities: ignoring practicalities in supervised classification, in *Classification, Clustering, and Data Mining Applications,* (Banks, D., House, L., McMorris, F., Arabie, P., and Gaul, W., eds.), Springer, Berlin, Germany pp. 209–232.
115. Nguyen, D., Arpat, A., Wang, N., and Carroll, R. (2002) DNA microarray experiments: biological and technological aspects. *Biometrics* **58,** 701–717.
116. Dutilh, B. and Hogeweg, P. (1999) *Gene networks from microarray data: analysis of data from microarray experiments, the State of the art in gene network reconstruction*. *Bioinformatics*, Utrecht University.
117. Holter, N., Maritan, A., Cieplak, M., Fedoroff, N., and Banavar, J. (2001) Dynamic modeling of gene expression data. *Proc. Natl. Acad. Sci. USA* **98,** 1693–1698.
118. Little, R. and Rubin, D. (eds.) (2002) *Statistical Analysis with Missing Data*. 2, Wiley-Interscience, New York.
119. Yoon, S. -Y. and Lee, S. -Y. (1999) Training algorithm with incomplete data for feed-forward networks. *Neural Processing Lett.* **10,** 171–179.
120. Personen, E., Eskelinen, M., and Juhola, M. (1998) Treatment of missing data values in a neural network based decision support system for acute abdominal pain. *AI in Med.* **13,** 139–146.
121. Dudoit, S., Fridlyand, J., and Speed, T. (2002) Comparison of discrimination methods for classification of tumors using gene expression data. *J. Am. Stat. Assoc.* **97,** 77–87.
122. Beissbarth, T., Fellenberg, K., Brors, B., et al. (2000) Processing and quality control of DNA array hybridization data. *Bioinformatics* **16,** 1014–1022.
123. Schuchhardt, J., Beule, D., Malik, A., et al. (2000) Normalization strategies for cDNA microarrays. *Nucleic Acids Res.* **28,** E47.
124. Schadt, E., Li, C., Ellis, B., and Wing, H. (2001) Feature extraction and normalization algorithms for high-density oligonucleotide gene expression array data. *J. Cell Biochem. Suppl*. **37,** 120–125.
125. Yang, Y. H., Dudoit, S., Luu, P., Lin, D., Peng, V., Ngai, J., and Speed, T. (2002), Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res.* **30,** E15.
126. Ross, D., Scherf, U., Eisen, M., et al. (2000) Systematic variation in gene expression patterns in human cancer cell lines. *Nat. Genet.* **24,** 227–234.
127. Perou, C., Jeffrey, S., van de Rijn, M., et al. (1999) Distinctive gene expression patterns in human mammary epithelial cells and breast cancers. *Proc. Natl. Acad. Sci. USA* **96,** 9212–9217.

128. Wang, W., Jones, P., and Partridge, D. (1998) Ranking pattern recognition features for neural networks, in *Advances in Pattern Recognition,* (Singh, S., ed.), Springer, Berlin, Germany pp. 232–241.
129. Park, P., Pagano, M., and Bonetti, M. (2001) A nonparametric scoring algorithm for identifying informative genes from microarray data. *Pac. Symp. Biocomput.* **6,** 52–63.
130. Ben-Dor, A., Bruhn, L., Friedman, N., Nachman, I., Schummer, M., and Yakhini, Z. (2000) Tissue classification with gene expression profiles. *J. Comput. Biol.* **7,** 559–583.
131. Tsai, C., Chen, C., Lee, T., Ho, I., Yang, U., and Chen, J. (2004) Gene selection for sample classifications in microarray experiments. *DNA Cell Biol.* **23,** 607–614.
132. Bo, T. and Jonassen, I. (2002) New feature subset selection procedures for classification of expression profiles. *Genome Biol.* **3,** Research0017.
133. Guyon, I. and Elisseeff, A. (2003) An introduction to variable and feature selection. *J. Machine Learning Res.* **3,** 1157–1182.
134. Cho, S. -B. and Won, H. -H. (2003) Data mining for gene expression profiles from DNA microarray. *Int. J. Software Eng. & Knowledge Eng.* **13,** 593–608.
135. Dudoit, S., Shaffer, J., and Boldrick, J. (2003) Multiple hypothesis testing in microarray experiments. *Stat. Sci.* **18,** 71–103.
136. Dudoit, S., Yang, Y., Callow, M., and Speed, T. (2002) Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Stat. Sinica* **12,** 111–139.
137. Efron, B. and Tibshirani, R. (eds.) (1993) *An Introduction to the Bootstrap*. Chapman and Hill, London, UK.
138. Efron, B. and Tibshirani, R. (1997) Improvements on cross-validation the 632+ Bootstrap Method. *J. Am. Stat. Assoc.* **92,** 548–560.